Short Lattice-based One-out-of-Many Proofs and Applications to Ring Signatures

Muhammed F. Esgin[1,2], Ron Steinfeld[1], Amin Sakzad[1], Joseph K. Liu[1], and Dongxi Liu[2]

[1] Faculty of Information Technology, Monash University, Australia
[2] Data61, CSIRO, Australia
{Muhammed.Esgin,Ron.Steinfeld,Amin.Sakzad,Joseph.Liu}@monash.edu
Dongxi.Liu@data61.csiro.au

Abstract. In this work, we construct a short one-out-of-many proof from (Module-SIS) lattices, allowing one to prove knowledge of a secret associated with one of the public values in a set. The proof system follows a combination of ideas from the efficient proposals in the discrete logarithm setting by Groth and Kohlweiss (EUROCRYPT '15) and Bootle et al. (ESORICS '15), can have logarithmic communication complexity in the set size and does not require a trusted setup.
Our work resolves an open problem mentioned by Libert et al. (EUROCRYPT '16) of how to efficiently adapt the above discrete logarithm proof techniques to the lattice setting. To achieve our result, we introduce technical tools for design and analysis of algebraic lattice-based zero-knowledge proofs, which may be of independent interest.
Using our proof system as a building block, we design a short lattice-based ring signature scheme. Our scheme offers post-quantum security and practical usability in cryptocurrencies and e-voting systems. Even for a very large ring size such as 1 billion, our ring signature size is only 4.5 MB for 100-bit security level compared to 166 MB in the best existing lattice-based result by Libert et al. (EUROCRYPT '16).

Keywords: lattice-based cryptography, zero-knowledge proof, ring signature

1 Introduction

In the last decade, lattice-based cryptography has seen a great interest with many new applications developed rapidly. Although it offers solutions even to problems which long seemed elusive, there is still a gap in some areas where lattice-based cryptographic proposals are not efficient enough for practical use and even fall far behind their number theoretic counterparts in terms of efficiency. One important example for such a case is zero-knowledge proofs (ZKPs). It seems that lattice-based cryptography does not agree well with ZKPs and the adaptation of the existing number theoretic proposals to the lattice setting is quite challenging.

A particular example is one-out-of-many proofs where the prover's goal is to convince the verifier that he knows the openings of a commitment within a

set of commitments without revealing which one he has. Groth and Kohlweiss [13] and Bootle et al. [7] gave very efficient constructions based on decisional Diffie-Hellman (DDH) assumption resulting in protocols with logarithmic (log) communication complexity in the size of the set of commitments. Their protocols also lead to very efficient ring signatures without trusted setup[3], where a signatory signs a message on behalf of a group of users (referred as a *ring*). The idea behind obtaining a ring signature from a one-out-of-many proof works as follows. Users commit to their secret keys, resulting in the users' public keys. Then, the signatory proves (in a non-interactive fashion using Fiat-Shamir heuristic) that he knows an opening (i.e., the secret key) of one of the commitments (i.e., corresponding public keys) used to create the ring signature. Ring signatures are important tools used in e-voting systems and cryptocurrencies to provide anonymity. Especially in the case of cryptocurrencies, an important aspect is the ring signature size, which makes the schemes in [13, 7] very attractive on a large scale. However, these proposals in [13, 7] do not offer post-quantum security as they are in the discrete logarithm (DL) setting.

On the side of lattice-based cryptography, offering post-quantum security, efficient designs targeting the same problems do not currently exist. There has not been a successful extension of the ideas in [13, 7] to the lattice setting, and other approaches proposed so far resulted in very inefficient schemes that are far from offering practical usability. To illustrate, while [7] gives constructions in the order of a few KB even for very large ring sizes, the current shortest log-sized ring signature from lattices by Libert et al. [17] results in a ring signature of size exceeding 58MB for around a thousand ring members and a security level of 100 bits. It is therefore tempting to realise the ideas in [13, 7] using lattice-based techniques, but, as we discuss next, this is far from trivial. In this work, we tackle this problem and design short one-out-of-many proofs and ring signatures from lattices following a combination of ideas from [13] and [7].

1.1 Technical difficulties

The starting point of our protocol is the works by Groth and Kohlweiss [13] and Bootle et al. [7]. The latter work borrows ideas from the former, and, in a way, generalises them. The ideas in those works are instantiated based on DDH problem and using Pedersen commitment as a core ingredient. From here, a natural question arises: "how can one extend the schemes [13, 7] to a lattice setting?" As also noted in [17] and [2], the answer is not so clear and it is not straightforward to design lattice-based one-out-of-many proofs and ring signatures from the ideas in [13, 7]. One can see [28] for an attempt to design a linkable lattice-based ring signature based on [13]. The authors of [28] claim that the anonymity and unforgeability of their scheme follow from the framework of [13], provided that a perfectly hiding and computationally binding commitment scheme is used. However, as we show here, there are many issues to be addressed if one aims

[3] There are some constructions of ring signatures that give a constant size signature but require a trusted setup.

to use the ideas from [13, 7] in the lattice setting, whereas [28] did not go into details of how these issues are to be solved. To begin with, the *valid* input space of lattice-based commitment schemes is a proper subset of $\mathbb{Z}_q^v$ for some $v \geq 1$ (or the underlying polynomial ring $R_q^v = \mathbb{Z}_q[X]/(X^d + 1)$ in the case of ring variants) consisting of vectors of *small* elements unlike their number-theoretic counterparts such as Pedersen commitment accepting any element in $\mathbb{Z}_q^v$. This restriction prevents straightforward adaptation of number-theoretic results, and in fact there is a crucial difference between the relations of the lattice-based and DDH-based one-out-of-many proofs (see Remark 1 in Section 4). Furthermore, extending [13] alone does not yield *efficient* lattice-based ring signatures even if the security issues in the lattice setting are addressed properly.

Let us briefly mention the technical difficulties one may face in extending [13, 7] to the lattice setting. As we move forward, we will explain how and why these difficulties are encountered, and how they are solved. We denote the public set size for the one-out-of-many proof (or the ring size for the ring signature) by N, and $C = \mathrm{Com}_{ck}(m\ ;\ r)$ as a commitment to a message m with randomness r using a commitment key ck. A pair of *acceptable* values (m', r') such that $C = \mathrm{Com}_{ck}(m'\ ;\ r')$ is called an opening of C. The reader unfamiliar with the general concepts of Σ-protocols is referred to Section 2.3.

1. **Growth of extracted witness size:** As mentioned previously, lattice-based commitment schemes accept only elements of bounded size as valid openings. It will turn out that the sizes of extracted witnesses, which will be openings of some commitments, grow rapidly with the size of challenge difference inverses in our protocol (see Section 3.2). In particular, we show that if one works over a ring $R_q = \mathbb{Z}_q[X]/(X^d + 1)$, the growth can be made to be of the form $\Gamma = d^{\log N}$. Letting $d = 2^{10}$ with $N = 2^{20}$ users, Γ (and, in turn, q) reaches 200 bits without any additional considerations.
2. **Challenge space size:** In connection with the above difficulty, we need to find a challenge space where the sizes of challenge difference inverses are guaranteed to be small. Unfortunately, we cannot find such a space with exponentially many elements, restricting us to a small challenge space (see Section 2.1). A simple (commonly used) possible option is to use binary challenges. However, the scheme presented in [7] requires at least 3 distinct challenges to extract a witness, making that option ineligible. In fact, the main protocols in [13, 7] even require up to $\log_2 N + 1$ challenges for witness extraction.
3. **Proof of commitment to a binary value over R_q:** When working over the ring $R_q = \mathbb{Z}_q[X]/(X^d + 1)$, the following statement, which is typically used to prove that a value is binary, does not necessarily hold: $x(x-1) = 0 \implies x \in \{0, 1\}$. This is because there exist zero divisors in R_q (unlike the field $\mathbb{Z}_q$ used in DL-based schemes). Hence, straightforward proofs of $x(x-1) = 0$ does not guarantee that x is binary (see Section 3.1).
4. **Simulation of witness-dependent prover responses:** Similar to some other lattice-based proofs (e.g., [5, 1]), the distribution of the prover's responses in our protocol depends on some secret values as the responses cannot be uniformly distributed in the domain because their norm must be small.

Since, in case of a simulation, these values are unknown, one cannot easily simulate the protocol, and thus cannot prove zero-knowledge property straightforwardly (see Section 3.3).

5. **Soundness gap:** In common with some other lattice-based proofs, our protocol has the so-called *soundness gap*[4] unlike DL-based schemes. That is, the extractor recovers the openings of $\gamma \cdot \mathrm{Com}_{ck}(m\ ;\ r)$ instead of the actual commitments of the form $\mathrm{Com}_{ck}(m\ ;\ r)$. This makes things more complicated in the soundness proofs (see Theorem 1 and 2) and may cause problems in protocol's application to a ring signature as the extractor is never guaranteed to recover the openings of the actual commitments used in the protocol.
6. **Unforgeability proof of signature with a small challenge space:** A general idea for the unforgeability proof of a signature based on a Σ-protocol is to show that if one gets multiple successful forgeries, then he can also extract a witness for the underlying protocol. Due to the small challenge space, a single round of our protocol does not provide negligible soundness error, and thus the protocol is repeated multiple times to achieve it. Furthermore, since our main protocol is $(k+1)$-special sound where k may be as large as $\log_2 N$, the forgery algorithm is required to get many successful forgeries, not just 2 (i.e., several *forkings* are needed). These make things much harder in the unforgeability proof. In some previous works (for example, in [17]), it is assumed that challenges used by the forgery algorithm are uniformly distributed *conditioned on* successful forgeries. However, it is unclear if this assumption holds as these challenges are restricted only to those yielding successful forgeries. Additionally, a commonly used Forking Lemma from [10] requires the number of challenges needed for witness extraction to be much smaller than the cardinality of the challenge space, which is not trivially satisfied in our case. An unforgeability proof with a small challenge space that does not rely on these assumptions forms another problem we address (see Section 3.4), which may be of independent interest for other lattice-based schemes.

The last two difficulties come into play rather in the protocol's application to a ring signature, whereas the first four are crucial in the protocol design. These technical difficulties are referred by Difficulty i in the rest of the paper.

1.2 Related work

A ring signature enables one to sign a message on behalf of an ad hoc group, called *ring*, of users without revealing the actual signatory. The ring is formed by gathering public keys and no consent is required from the users to involve them in the process. Ring signatures were introduced by Rivest, Shamir and Tauman-Kalai [24] and the rigorous security notions were established in the work of Bender, Katz and Morselli [4]. The work in this area is relatively scarce and currently, the only log-size (in the number of ring members) ring signatures

[4] This is also closely related to the notion of *slack*, which describes the ratio between the norm of an extracted witness and that of a witness which can be used by an honest prover (see, for example, [3] for a discussion).

based on number-theoretic assumptions are due to [13] and [7], where the main ideas in the latter are borrowed from the former. In [13, 7], the authors first describe efficient (in terms of communication complexity) one-out-of-many proofs, enabling a prover to convince a verifier that he knows an opening of a commitment contained in public set of commitments. Then, this tool enables them to design short ring signatures in the DL setting. On the side of lattice setting, most of the existing ring signature schemes (e.g., [9, 14, 21, 2]) have linear size.

As mentioned earlier, [28] attempts to extend Groth-Kohlweiss' scheme [13] by replacing Pedersen commitment with a lattice-based commitment scheme. It is claimed that the security requirements for the instantiation with this lattice-based commitment follows from the results of [13]. We found that this does not hold true without addressing the issues detailed in Section 1.1, which is also hinted in the works [17, 2] by noting that Groth-Kohlweiss' scheme does not easily extend to the lattice setting. We provide a brief discussion about [28] in Appendix A. Moreover, even if the security issues were to be solved, [28] leads to inefficient parameters without our techniques.

This leaves us with the work of Libert et al. [17] (and a follow-up by [27], adding linkability to [17]) as the only log-sized ring signature from lattices. In [17], the authors first design an accumulator through a Merkle tree using SIS-based hash function. Zero-knowledge membership arguments are then built for this accumulator. Having these building blocks, the authors propose ring and group signatures, both of which are log-sized in the number of users involved. We therefore focus on [17] for efficiency comparison purposes.

Most of the existing lattice-based zero-knowledge proofs requiring an extraction of a small witness make use of either binary challenges or use Stern-type protocols [25], providing soundness errors of $1/2$ and $2/3$, respectively. These approaches inherently require more than 100 repetitions to achieve a negligible soundness error, say 2^{-100}. Benhamouda et al. [5] introduced a different challenge space in the Ring-LWE setting consisting of *monomial challenges* of the form $X^i \in R = \mathbb{Z}[X]/(X^d + 1)$ and proved that the (Euclidean) norm of the doubled inverse differences of such challenges is at most $\sqrt{d}$ (i.e., $\left\|2(X^i - X^j)^{-1}\right\| < \sqrt{d}$). If we consider a ring dimension $d = 2^{10}$, this approach requires only 10 repetitions to achieve the same soundness error of 2^{-100}.

1.3 Our contributions

Design of short lattice-based one-out-of-many proofs and ring signatures. By now, it is clear that extending the works [13, 7] to the lattice setting is far from being trivial, which was indeed stated as an open problem in [17, 2]. Our main contribution in this work is, by carefully crafting various techniques from lattice-based cryptography, to design short one-out-of-many proofs and sublinear size ring signature schemes from (module) lattices. It is worth emphasising that our proposal is not a direct adaptation of either [13] or [7], but rather carefully combines ideas from both in a way suitable in the lattice scenario. We introduce new methods and blend them with the existing ones to overcome the problems in such algebraic protocols (see Section 3).

The flexibility of our scheme offers a tradeoff between the constant overhead and signature size growth with respect to ring size. This stems from the ability to choose varying base representations for user indices and results in different asymptotic growths of signature length, and also enables us to get better practical efficiency. Our results show the suitability of our scheme in a post-quantum scenario.[5] Table 1 shows that, in terms of ring signature size, we achieve an improvement of a factor between 37 and 50 in comparison to the current shortest log-sized ring signature from lattices by Libert et al. [17].

Table 1: Comparison of ring signature sizes for a security level $\lambda = 100$ with $2^{\log N}$ ring participants. The sizes are rounded to the nearest integer. For the results of [17], we use the same system parameters given in [17] for a soundness error of 2^{-80}, but only increase the number of protocol repetitions to have 2^{-100} soundness error.

$\log N$	6	8	10	12	16	20	30
[17] (sign. size in KB)	37022	48094	59166	70238	92382	114526	169887
Our Work (sign. size in KB)	930	1132	1409	1492	1814	2604	4511

A series of previous proposals of group and ring signatures (for example, [17, 16, 18]) relied on combinatorial Stern-like protocols [25]. Even though these protocols offer a range of functionalities, all of them have very long signature sizes that seem too large for practical use. Our technical tools developed in Section 3 introduce new directions for efficient applications of algebraic lattice-based techniques to areas where lattice-based proposals fall behind their number-theoretic counterparts. In fact, one may make use of our techniques to fix the issues in [28], however it seems unlikely that the revised scheme would have better practical efficiency than the current work.

We show how to improve efficiency of our algebraic ZKP using a packed message commitment technique that preserves the homomorphic commitment property. A packing technique is also used in [17], but the homomorphic properties are not preserved, which makes that packing method unsuitable for efficient algebraic arguments as we use in this paper, unlike the combinatorial Stern-like arguments in [17].

Exploiting module variants of standard lattice assumptions for efficiency purposes. Another important contribution of our work is to show that the use of Module-SIS problem [15] (over SIS or Ring-SIS) opens the door for significant efficiency improvements by allowing us to tradeoff extracted witness size growth (and hence signature length) against computational efficiency. To

[5] Our scheme is only analyzed in the classical random oracle model (ROM) (rather than quantum ROM). At this point, we do not consider for a further analysis in quantum ROM, and refer the reader to Section 1.3 of [12] for a discussion.

the best of our knowledge, this is the first time a lattice-based ZKP has been instantiated based on Module-SIS to gain such an efficiency improvement[6].

Interestingly, the idea of monomial challenges from [5] has not seen much application. In Lemma 7, we prove a general statement about an important issue encountered in algebraic protocols as ours that relates the use of monomial challenges and module lattice dimension. We believe that the combination of using monomial challenges together with Module-SIS to fine-tune the parameters for efficiency purposes holds great potential to be investigated through further research in lattice-based cryptography. Our successful incorporation of these various techniques achieving a dramatic efficiency improvement can be seen as an indication of this potential.

Novel unforgeability proof for lattice-based proofs with small challenge space. Last but not the least, we give a new proof that relaxes an assumption made in some earlier works (for example, in [17]), namely uniformity of the distribution of challenges conditioned on successful forgeries (recall Difficulty 6). The new proof applies to a wider range of parameters by removing another assumption in the commonly-used Forking Lemma of [10] and may be of independent interest for other lattice-based proofs with small challenge spaces.

2 Preliminaries

Throughout the manuscript, bold-face lower-case letters like $\boldsymbol{x}$ are used to denote column vectors and bold-face capital letters like $\boldsymbol{A}$ to denote matrices. $(\boldsymbol{x}, \boldsymbol{y})$ denotes appending the vector $\boldsymbol{y}$ to the vector $\boldsymbol{x}$. $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ denotes the ring of integers modulo q represented by the range $\left[-\frac{q-1}{2}, \frac{q-1}{2}\right]$ where q is an odd positive integer. We further assume that $q \equiv 5 \bmod 8$ is prime throughout the manuscript. We usually work with the Euclidean norm denoted by $\|\cdot\|$ unless otherwise stated. For a vector $\boldsymbol{x} = (x_0, \ldots, x_{n-1})$ and a polynomial $p(X) = a_0 + a_1 X + \cdots + a_{n-1} X^{n-1}$ in variable X, the Euclidean norm is defined as $\|\boldsymbol{x}\| = \sqrt{\sum_{i=0}^{n-1} x_i^2}$ and $\|p\| = \sqrt{\sum_{i=0}^{n-1} a_i^2}$, respectively. The infinity norm of p is $\|p\|_\infty = \max_i |a_i|$. For a vector $\boldsymbol{p} = (p_0, \ldots, p_{m-1})$ of polynomials, $\|\boldsymbol{p}\| = \sqrt{\sum_{i=0}^{m-1} \|p_i\|^2}$. We let $R = \mathbb{Z}[X]/(X^d+1)$ and $R_q = \mathbb{Z}_q[X]/(X^d+1)$ where $d > 1$ is a power of 2. Also, we denote the main security parameter by λ and adapt $\lambda = 100$ when instantiating parameters. $a \leftarrow \mathcal{Z}$ means a is chosen uniformly from a set $\mathcal{Z}$. We use the same notation to sample a from a distribution $\mathcal{Z}$. In the case that $\mathcal{Z}$ is an algorithm, the same notation is used to denote that the algorithm outputs a. $D_{v,\sigma}$ denotes the discrete Normal distribution (see Definition 3) centered at v with standard deviation σ. If $v = 0$, we simply write

[6] Module-SIS is used mostly (e.g. as in [11]) to fix the polynomial ring dimension and to avoid the need for a change of this dimension to accommodate new security parameters. It does not have a significant effect on the efficiency due to extracted witness norm unlike in our case.

D_σ. Logarithms are base 2 unless explicitly specified otherwise. We say that a function $\nu(\lambda)$ is negligible (denoted by $\nu = \text{negl}(\lambda)$) if $\nu(\lambda) < 1/\lambda^c$ for any $c > 0$ and all sufficiently large λ. $[a, b]$ denotes the set of integers $\{a, a+1, \ldots, b-1, b\}$.

2.1 Module-SIS problem and commitment scheme

To tackle Difficulty 1 and 2, we work over a ring R_q and use monomial challenges $X^i \in R$ for $0 \leq i \leq 2d-1$. This ensures $2d$ distinct elements in the challenge space and allows us to bound the norm of extracted witnesses. The security of our schemes is based on the hardness of Module-SIS problem [15] defined below.

Definition 1 (Module-SIS$_{n,m,q,\theta}$). *Let $R_q = \mathbb{Z}_q[X]/(X^d+1)$. Given $\boldsymbol{A} \in R_q^{n \times m}$ where each component is chosen independently from the uniform distribution, find $\boldsymbol{z} \in R_q^m$ such that $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{0} \bmod q$ and $0 < \|\boldsymbol{z}\| \leq \theta$.*

Note that when $n = 1$, the Module-SIS problem is equivalent to the Ring-SIS problem. The hardness of Module-SIS is related to the dimension of the corresponding module lattice given by $n_L = n \cdot d$.

We use the following lattice-based commitment scheme that allows commitment to multiple messages, and is additively homomorphic. Following the standard notions, hiding property requires that it is hard to distinguish between commitments to two distinct message-randomness pairs and binding property dictates that it is hard to find two distinct *valid* openings of a commitment.

Definition 2. *Let $R_q = \mathbb{Z}_q[X]/(X^d+1)$, $S_r(\gamma_r) = \left\{\boldsymbol{r} \in R_q^m \ : \ \|\boldsymbol{r}\|_\infty \leq \gamma_r\right\}$ be the randomness domain with χ_r as the probability distribution of $\boldsymbol{r}$ on $S_r(\gamma_r)$ for a positive real number γ_r, and $S_M(\gamma_M) = \left\{\boldsymbol{m} \in R_q^v \ : \ \|\boldsymbol{m}\|_\infty \leq \gamma_M\right\}$ be the message domain for a positive real number γ_M for $m, v \in \mathbb{Z}^+$. The commitment of a message vector $\boldsymbol{m} = (m_1, \ldots, m_v) \in S_M(\gamma_M)$ using a randomness $\boldsymbol{r} \in S_r(\gamma_r)$ (treated as a column vector) is given as*

$$\text{Com}_{ck}(\boldsymbol{m} \ ; \ \boldsymbol{r}) = \text{Com}_{ck}(m_1, \ldots, m_v \ ; \ \boldsymbol{r}) = \boldsymbol{G} \cdot \begin{pmatrix} \boldsymbol{r} \\ m_1 \\ \vdots \\ m_v \end{pmatrix} \in R_q^n,$$

where $ck = \boldsymbol{G} \in R_q^{n \times (m+v)}$ is a public matrix chosen uniformly at random, and it is used as the commitment key. One may assume that $\boldsymbol{G}$ is in Hermite normal form, i.e., the first part of $\boldsymbol{G}$ is the $n \times n$ identity matrix.

We use a special case of Corollary 1.2 in [20] that is obtained by putting $k = 2$ in that corollary as below, and also prove the hiding/binding properties.

Lemma 1 ([20, Corollary 1.2]). *Let $q \equiv 5 \bmod 8$ be prime. Then, any non-zero polynomial $z \in R_q$ with $2\|z\|_\infty^2 < q$ or $\|z\|^2 < q$ is invertible in R_q.*

Lemma 2. *For a security parameter λ and prime $q \equiv 5 \bmod 8$, the commitment scheme in Definition 2 is statistically hiding if $2\gamma_r < \sqrt{q/2}$ and the min-entropy of χ_r is greater than $n \log q/m + 2\lambda/(md)$, and computationally binding*

if Module-SIS$_{n,m+v,q,\theta}$ problem for $\theta = 2\sqrt{\gamma_r^2 md + \gamma_M^2 vd}$ is hard. In particular, the statistical distance between the distribution of $\mathrm{Com}_{ck}(\boldsymbol{m}\ ;\ \boldsymbol{r})$ *(with respect to* $\boldsymbol{r}$ *for any fixed* $\boldsymbol{m}$*) and the uniform distribution on* R_q^n *is at most* $2^{-\lambda}$*.*

Proof. Given $(\boldsymbol{r}, \boldsymbol{m})$ and $(\boldsymbol{r}', \boldsymbol{m}')$ such that $\mathrm{Com}_{ck}(\boldsymbol{m}\ ;\ \boldsymbol{r}) = \mathrm{Com}_{ck}(\boldsymbol{m}'\ ;\ \boldsymbol{r}')$ and $(\boldsymbol{r}, \boldsymbol{m}) \neq (\boldsymbol{r}', \boldsymbol{m}')$, we have $\boldsymbol{G} \cdot (\boldsymbol{r}, \boldsymbol{m}) = \boldsymbol{G} \cdot (\boldsymbol{r}', \boldsymbol{m}')$, which implies $\boldsymbol{G} \cdot (\boldsymbol{r} - \boldsymbol{r}', \boldsymbol{m} - \boldsymbol{m}') = 0$. Therefore, $(\boldsymbol{r} - \boldsymbol{r}', \boldsymbol{m} - \boldsymbol{m}')$ is a solution to Module-SIS$_{n,m+v,q,\theta}$ problem for $\theta = 2\sqrt{\gamma_r^2 md + \gamma_M^2 vd}$ since $\|(\boldsymbol{r} - \boldsymbol{r}', \boldsymbol{m} - \boldsymbol{m}')\| \leq 2\sqrt{\gamma_r^2 md + \gamma_M^2 vd}$.

We can write $\mathrm{Com}_{ck}(\boldsymbol{m}\ ;\ \boldsymbol{r}) = \boldsymbol{G}_0 \cdot \boldsymbol{r} + \boldsymbol{G}_1 \cdot \boldsymbol{m}$ where $\boldsymbol{G} = [\boldsymbol{G}_0 || \boldsymbol{G}_1]$ with $\boldsymbol{G}_0 \in R_q^{n \times m}$ and $\boldsymbol{G}_1 \in R_q^{n \times v}$. Focusing on $\boldsymbol{G}_0 \cdot \boldsymbol{r}$ part, the statistical hiding property follows from Lemma 4 of [1] (that uses Leftover Hash Lemma and [20, Corollary 1.2]) where we replace 256 by 2λ and set $d = 2$ in that lemma. □

Note that for the binding property, the maximum Euclidean norms on the message and randomness domains are important as Module-SIS is defined in terms of the Euclidean norm. It is clear that as θ gets smaller, Module-SIS problem becomes harder. That's why for the computational binding of the commitment, the norm of valid openings are required to be small, and thus γ_r and γ_M must be much smaller than q. Similar to the previous results such as [19], when instantiating parameters, we estimate the computational hardness of the Module-SIS problem based on the results of [22]. They show that state-of-the-art lattice reduction algorithms find a non-zero vector of length

$$\min\left\{q, 2^{2\sqrt{n_L \log q \log \delta}}\right\}, \tag{1}$$

where n_L is the dimension of the lattice and δ is the root Hermite factor depending on the quality of the lattice reduction algorithm. Similar to [19], we adapt a security level of $\lambda = 100$ and set $\delta = 1.007$. In order to tie the security requirements to Module-SIS, we make sure that the largest norm of an extracted witness is strictly smaller than (1). Also, observe that the following homomorphic properties hold: $\mathrm{Com}_{ck}(\boldsymbol{a}\ ;\ \boldsymbol{r}_1) + \mathrm{Com}_{ck}(\boldsymbol{b}\ ;\ \boldsymbol{r}_2) = \mathrm{Com}_{ck}(\boldsymbol{a} + \boldsymbol{b}\ ;\ \boldsymbol{r}_1 + \boldsymbol{r}_2)$ and $\gamma \cdot \mathrm{Com}_{ck}(\boldsymbol{a}\ ;\ \boldsymbol{r}) = \mathrm{Com}_{ck}(\gamma \cdot \boldsymbol{a}\ ;\ \gamma \cdot \boldsymbol{r})$ for any $\gamma \in R_q$.

2.2 Technical definitions and general lemmas

In this section, we review some technical definitions and lemmas. We start with a definition of discrete Normal distribution and statements regarding the bounds on the norms of a vector following such a distribution.

Definition 3. *The discrete normal distribution over* $\mathbb{Z}^t$ *for a positive integer* t *centered at* $\boldsymbol{v}$ *with standard deviation* σ *is defined by the probability mass function* $D^t_{\boldsymbol{v},\sigma}(\boldsymbol{x}) = \rho^t_{\boldsymbol{v},\sigma}(\boldsymbol{x})/\rho^t_\sigma(\mathbb{Z}^t)$ *where* $\rho^t_{\boldsymbol{v},\sigma}(\boldsymbol{x}) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^t e^{-\frac{\|\boldsymbol{x}-\boldsymbol{v}\|^2}{2\sigma^2}}$ *is the continuous normal distribution over* $\mathbb{R}^t$*, and* $\rho^t_\sigma(\mathbb{Z}^t) = \sum_{\boldsymbol{z} \in \mathbb{Z}^t} \rho^t_\sigma(\boldsymbol{z})$ *is a normalisation factor needed to obtain a probability distribution.*

Lemma 3 ([19, Lemma 4.4]).

1. *For any* $\alpha > 0$, $\Pr[|z| > \alpha \cdot \sigma \ : \ z \leftarrow D_\sigma] \leq 2 \cdot \exp(-\frac{\alpha^2}{2})$,
2. *For any* $\alpha > 1$, $\Pr[\|\boldsymbol{z}\| > \alpha\sigma\sqrt{t} \ : \ \boldsymbol{z} \leftarrow D_\sigma^t] < \alpha^t e^{\frac{1-\alpha^2}{2}t}$. *In particular,*

- $\Pr[|z| > 12\sigma \ : \ z \leftarrow D_\sigma] < 2^{-100}$,
- $\Pr[\|\boldsymbol{z}\| > 2\sigma\sqrt{t} \ : \ \boldsymbol{z} \leftarrow D_\sigma^t] < 2^{-100}$ *if* $t \geq 86$, *and*
- $\Pr[\|\boldsymbol{z}\| > 5\sigma\sqrt{t} \ : \ \boldsymbol{z} \leftarrow D_\sigma^t] < 2^{-100}$ *if* $t \geq 7$.

We also summarise some known results related to the norms in the next lemma.

Lemma 4. *For* $a, b \in R_q = \mathbb{Z}_q[X]/(X^d + 1)$, *we have the following relations*

$$\|a\| \leq \sqrt{d} \cdot \|a\|_\infty, \quad \|a \cdot b\| \leq \sqrt{d} \cdot \|a\| \cdot \|b\|, \ \textit{and} \quad \|a \cdot b\|_\infty \leq \|a\| \cdot \|b\|.$$

2.3 Σ-protocols

Σ-protocols are a type of zero-knowledge proofs between two parties: the prover and the verifier. A language $\mathcal{L} \subseteq \{0,1\}^*$ is said to have a witness relationship $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ provided $v \in \mathcal{L}$ if and only if there exists $w \in \{0,1\}^*$ such that $(v, w) \in \mathcal{R}$. The quantity w is referred to as a witness for v. The definition of Σ-protocols from [5] generalises the well-known notion of Σ-protocols. We further extend it to allow $(k+1)$-special soundness as in [13, 7].

Definition 4 (Extension of Definition 2.5 in [5]). *Let* $(\mathcal{P}, \mathcal{V})$ *be a two-party protocol where* $\mathcal{V}$ *is a PPT algorithm, and* $\mathcal{L}, \mathcal{L}'$ *be languages with witness relations* $\mathcal{R}, \mathcal{R}'$ *with* $\mathcal{R} \subseteq \mathcal{R}'$. *Then,* $(\mathcal{P}, \mathcal{V})$ *is called a* Σ*-protocol for* $\mathcal{R}, \mathcal{R}'$ *with completeness error* α, *a challenge set* $\mathcal{C}$, *public input* v *and private input* w, *if it satisfies the following conditions:*

- ***Three-move form:*** *The protocol has the following form. On input* (v, w), $\mathcal{P}$ *computes initial commitment* t *and sends it to* $\mathcal{V}$. *On input* v, $\mathcal{V}$ *draws a challenge* $x \leftarrow \mathcal{C}$ *and sends it to* $\mathcal{P}$. *The prover sends a response* s *to* $\mathcal{V}$. *The verifier accepts or rejects depending on the protocol transcript* (t, x, s). *The transcript* (t, x, s) *is called accepting if the verifier accepts the protocol run.*
- ***Completeness:*** *Whenever* $(v, w) \in \mathcal{R}$, *the honest verifier accepts with probability at least* $1 - \alpha$ *when interacting with an honest prover.*
- $(k+1)$***-special soundness:*** *There exists a PPT algorithm* $\mathcal{E}$ *(called the extractor) which takes* $(k+1)$ *accepting transcripts* $(t, x_0, s_0), \ldots, (t, x_k, s_k)$ *with pairwise distinct* x_i*'s* $(0 \leq i \leq k)$ *as inputs, and outputs* w' *satisfying* $(v, w') \in \mathcal{R}'$. *We call this procedure* witness extraction*, and say that the protocol has a soundness error* $\frac{k}{|C|}$.[7]
- ***Special honest-verifier zero-knowledge (SHVZK):*** *There exists a PPT algorithm* $\mathcal{S}$ *(called the simulator) that takes* $v \in \mathcal{L}$ *and* $x \in \mathcal{C}$ *as inputs, and outputs* (t, s) *such that* (t, x, s) *is indistinguishable from an accepting protocol transcript generated by a real protocol run.*

The *soundness gap* aforementioned stems from the fact that the verifier is only convinced that the prover knows a witness for the relation $\mathcal{R}'$ whereas the prover's privacy is guaranteed when he knows a witness for $\mathcal{R} \subseteq \mathcal{R}'$.

[7] We refer to Section 2.2 of [6] for further discussion on soundness error.

3 Technical Tools for Lattice-based Proofs

In this section, we present a collection of technical tools we use in our construction of algebraic lattice-based proofs. These tools may be of independent interest for future works on algebraic lattice-based zero-knowledge proofs and signatures.

3.1 Proof of commitment to a binary value over R_q (Difficulty 3)

We start with a lemma that enables to us prove that a value is binary over a ring R_q. It will be particularly useful for the proofs of Protocol 1.

Lemma 5. *Let $\hat{b} \in R_q$ where q is a prime with $q \equiv 5 \bmod 8$. If $\hat{b}(2-\hat{b}) = 0$ over R_q and $2\|\hat{b}\|_\infty^2 < q$, then $b \in \{0,1\}$ for $b = 2^{-1}\hat{b}$.*

Proof. By Lemma 1, if $\hat{b}$ is non-zero and $2\|\hat{b}\|_\infty^2 < q$, we know that it must be invertible. Thus, $2 - \hat{b} = 0$ and $\hat{b} = 2$ in that case. As a result, $\hat{b} \in \{0, 2\}$. This gives us that $b = 2^{-1}\hat{b} \in \{0, 1\}$. □

3.2 The size of Vandermonde matrix inverse entries (Difficulty 1)

Consider a Σ-protocol where the prover's initial commitments are $a_0, a_1, \ldots, a_k$ ($k \geq 1$), and he responds with $z_x = (f_x, r_x)$ for a given challenge x by the verifier. Then, the verifier checks whether $a_0 + a_1 x + a_2 x^2 + \cdots + a_k x^k = \mathrm{Com}(z_x)$ holds where Com is a homomorphic commitment scheme. Now, suppose a_k is the commitment of prover's witness and that the extractor obtains $k+1$ accepting protocol transcripts for the same initial commitments, represented as follows.

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^k \\ 1 & x_1 & x_1^2 & \cdots & x_1^k \\ : & : & : & : & : \\ 1 & x_k & x_k^2 & \cdots & x_k^k \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ : \\ a_k \end{pmatrix} = \begin{pmatrix} \mathrm{Com}(z_{x_0}) \\ \mathrm{Com}(z_{x_1}) \\ : \\ \mathrm{Com}(z_{x_k}) \end{pmatrix}$$

Here, the matrix on the very left is a Vandermonde matrix $\boldsymbol{V}$, and the extractor can recover a *possible* opening of a_k via multiplying both sides by $\boldsymbol{V}^{-1}$, if exists, due to the homomorphic properties of the commitment scheme. We observe from [26] that the inverse matrix $\boldsymbol{V}^{-1}$ has the following form:

$$\begin{pmatrix} \frac{*}{(x_1-x_0)(x_2-x_0)\cdots(x_k-x_0)} & \frac{*}{(x_0-x_1)(x_2-x_1)\cdots(x_k-x_1)} & \cdots & \frac{*}{(x_0-x_k)(x_1-x_k)\cdots(x_{k-1}-x_k)} \\ \frac{*}{(x_1-x_0)(x_2-x_0)\cdots(x_k-x_0)} & \frac{*}{(x_0-x_1)(x_2-x_1)\cdots(x_k-x_1)} & \cdots & \frac{*}{(x_0-x_k)(x_1-x_k)\cdots(x_{k-1}-x_k)} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{(x_1-x_0)(x_2-x_0)\cdots(x_k-x_0)} & \frac{1}{(x_0-x_1)(x_2-x_1)\cdots(x_k-x_1)} & \cdots & \frac{1}{(x_0-x_k)(x_1-x_k)\cdots(x_{k-1}-x_k)} \end{pmatrix} \quad (2)$$

where $*$ denotes some element in the domain. Our protocol as well as the protocols in [13, 7] have this structure and, therefore, the Vandermonde matrix inverse plays a crucial role in the witness extraction. Thus, we need to make sure that it exists in the first place, which follows from the invertibility of pairwise differences of challenges. What is more important in the case of lattice-based proofs is that the sizes of the entries in $\boldsymbol{V}^{-1}$ must be small so that extracted witness is a *valid* opening. To that end, we first state the following lemma.

Lemma 6 ([5, Lemma 3.1]). *Let $R = \mathbb{Z}[X]/(X^d + 1)$ where $d > 1$ is a power of 2, and $0 < i, j < 2d - 1$. Then, all the coefficients of $2(X^i - X^j)^{-1} \in R$ are in $\{-1, 0, 1\}$. This implies that $\left\|2(X^i - X^j)^{-1}\right\| \leq \sqrt{d}$.*

Let us take the first entry in the last row of $\boldsymbol{V}^{-1}$ as an example, and call it α_0. We have

$$2^k \alpha_0 = \frac{2^k}{(x_1 - x_0)(x_2 - x_0)\cdots(x_k - x_0)} = \frac{2}{x_1 - x_0} \cdot \frac{2}{x_2 - x_0} \cdots \frac{2}{x_k - x_0}.$$

For monomial challenges, we have $x_i = X^{\omega_i}$ for some $0 \leq \omega_i \leq 2d-1$, and hence

$$\begin{aligned}
\left\|2^k \alpha_0\right\| &= \left\| \prod_{i=1}^{k} \frac{2}{x_i - x_0} \right\| = \left\| \prod_{i=1}^{k} 2(X^{\omega_i} - X^{\omega_0})^{-1} \right\| \\
&\leq \left(\sqrt{d}\right)^{k-1} \prod_{i=1}^{k} \left\|2(X^{\omega_i} - X^{\omega_0})^{-1}\right\| \quad \text{(by Lemma 4)} \\
&\leq \left(\sqrt{d}\right)^{k-1} \left(\sqrt{d}\right)^{k} = d^{k-0.5} \quad \text{(by Lemma 6).}
\end{aligned}$$

Since all the entries in the last row have a similar form and the bound does not depend on the particular values of ω_i's, the same bound holds for all entries in the last row of $\boldsymbol{V}^{-1}$. Note that $\boldsymbol{V}^{-1}$ exists over R_q for odd q (though may not have small entries) since 2 is invertible for such q. We summarise these results in the following lemma, whose proof follows from the above discussion.

Lemma 7. *Let $x_i = X^{\omega_i} \in R = \mathbb{Z}[X]/(X^d + 1)$ for $0 \leq \omega_i \leq 2d - 1$ and $0 \leq i \leq k$. Define the Vandermonde matrix $\boldsymbol{V}$ of dimension $k+1$ where i-th row is the vector $(1, x_i, x_i^2, \ldots, x_i^k)$. Then, $\boldsymbol{V}$ is invertible over R_q for odd q, and for any entry α_j in the last row of $\boldsymbol{V}^{-1}$, we have $\left\|2^k \alpha_j\right\| \leq d^{k-0.5}$.*

3.3 Rejection sampling (Difficulty 4)

As mentioned in Difficulty 4, the distribution of the prover's response in our protocol is shifted depending on some secret values. This prevents the protocol from being zero-knowledge as it cannot be simulated. To tackle this problem, we make use of the rejection sampling technique from [19]. The idea is to output the response with a certain probability, and abort the protocol otherwise. The following lemma is a corollary of Theorem 4.6 in [19].

Lemma 8 ([19, Theorem 4.6]). *Let V be a subset of $\mathbb{Z}^d$ where all the elements have norms less than T, and h be a probability distribution over V. Define the following algorithms:*

$$\mathcal{A}: \ \boldsymbol{v} \leftarrow h; \quad \boldsymbol{z} \leftarrow D^d_{v,\sigma}; \quad \text{output } (\boldsymbol{z}, \boldsymbol{v}) \text{ with probability } \min\left\{\frac{D^d_\sigma(\boldsymbol{z})}{M D^d_{\boldsymbol{v},\sigma}(\boldsymbol{z})}, 1\right\},$$

$$\mathcal{F}: \ \boldsymbol{v} \leftarrow h; \quad \boldsymbol{z} \leftarrow D^d_\sigma; \quad \text{output } (\boldsymbol{z}, \boldsymbol{v}) \text{ with probability } 1/M,$$

where $\sigma = 12T$ *and* $M = e^{1+\frac{1}{288}}$*. Then, the output of algorithm* $\mathcal{A}$ *is within statistical distance* $2^{-100}/M$ *of the output of* $\mathcal{F}$*. Moreover, the probability that* $\mathcal{A}$ *outputs something is more than* $\frac{1-2^{-100}}{M}$*.*

The rejection sampling in our protocol is applied in two (parallel) steps due to a careful observation to be described in Remark 2. This helps relax the condition on the size of q and provides better efficiency results. Furthermore, in the construction of the ring signature scheme, we apply the rejection sampling on concatenated vectors to minimise the computational cost in the signing algorithm and the resulting effect on the signature size. Otherwise, either the signing algorithm would require unreasonably many iterations or the effect on the signature size in practice would be much larger.

3.4 A remark on Difficulties 5 and 6

So far, we have introduced the main tools for addressing the first four difficulties. Lemma 5 also helps deal with the soundness gap (Difficulty 5) in Protocol 1 given in Section 4. Moreover, we still tie the unforgeability of the ring signature to the binding property of the commitment scheme in Theorem 4 even though the underlying protocol has a soundness gap.

As mentioned, multiple repetitions of our protocol is required to get a negligible soundness error and witness extraction works with $k+1$ accepting protocol transcripts. Each repetition using a random challenge x_j means that the overall protocol with negligible soundness error deals with challenge *vectors* of the form $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,r})$ (instead of single challenges). Then, in the unforgeability proof of the ring signature, we try to get multiple *successful* forgeries using challenge vectors, say, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_s$. Hence, we have a matrix of challenges yielding successful forgeries as follows:

$$\boldsymbol{S} := \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,r} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,r} \\ \vdots & \vdots & \vdots & \vdots \\ x_{s,1} & x_{s,2} & \cdots & x_{s,r} \end{pmatrix}.$$

Now, in order to tie the unforgeability of the ring signature scheme to the special soundness of the one-out-of-many protocol, we need to find a column of $\boldsymbol{S}$ containing $k+1$ pairwise distinct challenges.[8] As mentioned, some works assume that $x_{i,j}$'s are uniformly distributed conditioned on successful forgeries. We will show in the unforgeability proof of our ring signature scheme that one does not need to such an assumption. Without making any assumptions on the distribution of such challenges, we prove via a pigeonhole argument in Claim 1 in the proof of Theorem 4 that, with a high probability, there exists a set of distinct challenges given sufficiently many successful forgeries. Moreover, our proof does not require the assumption that the number $(k+1)$ of required

[8] Note that when this happens, $k+1$ accepting protocol transcripts with the same initial commitment and pairwise distinct challenges are obtained, meaning that the extractor can extract a witness.

challenges to extract a witness is much smaller than the cardinality $|\mathcal{C}|$ of the challenge space. For example, in the Forking Lemma of [10], it is assumed that $(k+1)^2 < |\mathcal{C}|$, which is a very weak condition in the discrete logarithm setting as $|\mathcal{C}|$ is exponentially large. However, it is not always trivially satisfied by lattice-based schemes as $|\mathcal{C}|$ can be very small (as it happens in our case).

4 Σ-protocol for Commitment to a Sequence of Bits

In this section, we describe a lattice-based Σ-protocol showing that a commitment B opens to sequences of binary values where the Hamming weight of each sequence is exactly one. We first fix the notations given in Table 2, and define the relations to be proved in Definition 5.

Definition 5. *For positive real numbers $\mathcal{T}$ and $\hat{\mathcal{T}}$, we define the following relations to be used in Protocol 1.*

$$\mathcal{R}_{\text{bin}}(\mathcal{T}) = \left\{ \begin{array}{c} ((ck, B), (b_{0,0}, \ldots, b_{k-1,\beta-1}, \boldsymbol{r})) \ : \\ \|\boldsymbol{r}\| \leq \mathcal{T} \ \wedge \ B = \text{Com}_{ck}(b_{0,0}, \ldots, b_{k-1,\beta-1} \ ; \ \boldsymbol{r}) \wedge \\ (b_{j,i} \in \{0,1\} \ \forall j, i) \ \wedge \ (\sum_{i=0}^{\beta-1} b_{j,i} = 1 \ \forall j) \end{array} \right\}.$$

$$\mathcal{R}'_{\text{bin}}(\hat{\mathcal{T}}) = \left\{ \begin{array}{c} ((ck, B), (\hat{b}_{0,0}, \ldots, \hat{b}_{k-1,\beta-1}, \hat{\boldsymbol{r}})) \ : \\ \|\hat{\boldsymbol{r}}\| \leq \hat{\mathcal{T}} \ \wedge \ 2B = \text{Com}_{ck}(\hat{b}_{0,0}, \ldots, \hat{b}_{k-1,\beta-1} \ ; \ \hat{\boldsymbol{r}}) \wedge \\ (\hat{b}_{j,i} \in \{0,2\} \ \forall j, i) \ \wedge \ (b_{j,i} := \frac{\hat{b}_{j,i}}{2}, \ \sum_{i=0}^{\beta-1} b_{j,i} = 1 \ \forall j) \end{array} \right\}.$$

Remark 1. The conditions on the norms of $\boldsymbol{r}$ and $\hat{\boldsymbol{r}}$ in the relations $\mathcal{R}_{\text{bin}}$ and $\mathcal{R}'_{\text{bin}}$ play a very crucial role, and is one of the main differences of a lattice-based zero-knowledge proof over its number-theoretic counterpart. Without that control, one cannot easily tie the security of the protocol to a hard lattice problem.

Table 2: A summary of identifiers.

Notation	Explanation
λ	security parameter
$N = \beta^k$	the number of commitments for one-out-of-many proof (or the ring size for the ring signature)
β	base for the representation of user indices
q	a prime modulus with $q \equiv 5 \bmod 8$
d	ring dimension (i.e., $R_q = \mathbb{Z}_q[X]/(X^d + 1)$)
$k \cdot \beta$	the number of packed messages in a commitment
m	the dimension of randomness in a commitment (i.e., $\boldsymbol{r} \in R_q^m$)
$n \times (m + k\beta)$	public commitment key dimensions (i.e., $\boldsymbol{G} \in R_q^{n \times (m+k\beta)}$)
$n \times 1$	commitment dimensions
$\mathcal{B}$	maximum absolute coefficient of a uniformly chosen fresh randomness
r	the number of protocol repetitions to achieve negligible soundness error
ℓ	prover's index with $0 \leq \ell \leq N - 1$
$\mathcal{C}$	challenge space with $\mathcal{C} = \{X^\omega \ : \ 0 \leq \omega \leq 2d - 1\}$

In the protocol, we first prove that each value in the sequences is binary, and then that the sum of each sequence equals one. This guarantees that there is only a single 1 in each sequence. The idea behind proving a value binary works as follows. Let b be the value we want to prove binary. Given a challenge x, the value b is multiplied by x and the resulting value is masked by a as $f = x \cdot b + a$ in the protocol (Step 10 in Protocol 1). Now observe that $f \cdot (x - f) = b(1-b) \cdot x^2 + a(1-2b) \cdot x - a^2$ and proving that the coefficient of x^2 is zero implies that $b(1-b) = 0$. Then, using the discussion in Section 3.1, we show in the special soundness proof that for a sufficiently large q, this statement over R_q implies that b is binary.

Similar to [5], we make use of an auxiliary commitment scheme aCom (which is assumed to be hiding and binding) in order to be able to simulate aborts in the proof of zero-knowledge property.[9] One can treat aCom as a random oracle. However, if aCom is computationally binding, then the soundness of the protocol holds under the respective assumption and similarly if it is computationally hiding [5]. The protocol is described in Protocol 1, which will later be used in the one-out-of-many proof.

Theorem 1. *Using the notation in Protocol 1, let* $\boldsymbol{f}_1 := (f_{0,1}, \ldots, f_{k-1,\beta-1})$, $\boldsymbol{b}_1 := (b_{0,1}, \ldots, b_{k-1,\beta-1})$, *$M$ be the constant defined in Lemma 8. Assume that* $d \geq 7$, $md \geq 86$, $q > \max\left\{2\left(120d\sqrt{k(\beta-1)}\right)^2, 8\left(144\mathcal{B}\sqrt{2md}\right)^2\right\}$ *and* $2\mathcal{B} \geq q^{n/m} 2^{2\lambda/(md)}$. *Protocol 1 with*

$$p_{bin} = \frac{D^{k(\beta-1)d}_{12\sqrt{k}}(\boldsymbol{f}_1)}{MD^{k(\beta-1)d}_{x\cdot\boldsymbol{b}_1, 12\sqrt{k}}(\boldsymbol{f}_1)} \cdot \frac{D^{2md}_{12\mathcal{B}\sqrt{2md}}((\boldsymbol{z}_b, \boldsymbol{z}_c))}{MD^{2md}_{x\cdot(\boldsymbol{r}_b, \boldsymbol{r}_c), 12\mathcal{B}\sqrt{2md}}((\boldsymbol{z}_b, \boldsymbol{z}_c))} \tag{3}$$

is a 3-special sound Σ-protocol (as given in Definition 4) for the relations $\mathcal{R}_{\text{bin}}(\mathcal{B}\sqrt{md})$ *and* $\mathcal{R}'_{\text{bin}}(48\sqrt{2}\mathcal{B}md^2)$ *with soundness error* $\frac{1}{d}$ *and a completeness error* $1 - \frac{1}{M^2}$.

Remark 2. In Protocol 1, the rejection sampling is applied in parallel steps: 1) on $\boldsymbol{f}_1$ and 2) on the whole vector $(\boldsymbol{z}_b, \boldsymbol{z}_c)$. This is because if the rejection sampling is done on $(\boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z}_c)$ all together, the sizes of $f_{j,i}$'s would be unnecessarily large, making the condition on the size of q stronger. Therefore, we apply the rejection sampling this way, which will result in a slight computational cost in the signing algorithm of the ring signature construction to be described in Section 6.2. But, this cost can be compensated for as discussed in Appendix C.

Proof (Theorem 1). **Completeness:** By Lemma 8, the prover responds with probability $\frac{1}{M^2}$, and the distributions of $f_{j,i}$'s ($i \neq 0$) are statistically close to $D^d_{12\sqrt{k}}$ and the distributions of $\boldsymbol{z}_b, \boldsymbol{z}_c$ are statistically close to $D^{md}_{12\mathcal{B}\sqrt{2md}}$ since

$$\|(x \cdot b_{0,1}, \ldots, x \cdot b_{k-1,\beta-1})\| \leq \sqrt{k}, \quad \text{and} \quad \|(x \cdot \boldsymbol{r}, x \cdot \boldsymbol{r}_c)\| \leq \mathcal{B}\sqrt{2md}.$$

[9] In the application of protocol to a ring signature (and for other applications in general), simulation of aborts is not needed as the protocol is made non-interactive.

$\mathcal{P}_{\text{bin}}(ck, B, (\{b_{j,i}\}_{j,i=0}^{k-1,\beta-1}; \boldsymbol{r}))$		$\mathcal{V}_{\text{bin}}(ck, B)$
$a_{0,1}, \ldots, a_{k-1,\beta-1} \leftarrow D_{12\sqrt{k}}^{d}$		
$\boldsymbol{r}_c \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$		
$\boldsymbol{r}_a, \boldsymbol{r}_d \leftarrow D_{12\mathcal{B}\sqrt{2md}}^{md}$		
for $j = 0, \ldots, k-1$ **do**		
$a_{j,0} = -\sum_{i=1}^{\beta-1} a_{j,i}$		
$A = \text{Com}_{ck}(a_{0,0}, \ldots, a_{k-1,\beta-1}\ ;\ \boldsymbol{r}_a)$		
$C = \text{Com}_{ck}(\{a_{j,i}(1-2b_{j,i})\}_{j,i=0}^{k-1,\beta-1}\ ;\ \boldsymbol{r}_c)$		
$D = \text{Com}_{ck}(-a_{0,0}^2, \ldots, -a_{k-1,\beta-1}^2\ ;\ \boldsymbol{r}_d)$		
$(c_a, d_a) = \text{aCom}(A, C, D)$		
	$\xrightarrow{c_a}$	
		$\omega \leftarrow \{0, \ldots, 2d-1\}$
	$\xleftarrow{x := X^{\omega}}$	
$f_{j,i} = x \cdot b_{j,i} + a_{j,i}\ \forall j, \forall i \neq 0$		
$\boldsymbol{z}_b = x \cdot \boldsymbol{r} + \boldsymbol{r}_a$		
$\boldsymbol{z}_c = x \cdot \boldsymbol{r}_c + \boldsymbol{r}_d$		
abort with prob. $(1 - p_{bin})$ from (3)		
Return $\perp$ if aborted.	$\xrightarrow{\substack{f_{0,1}, \ldots, f_{k-1,\beta-1}, \\ d_a, A, C, D, \boldsymbol{z}_b, \boldsymbol{z}_c}}$	
		for $j = 0, \ldots, k-1$ **do**
		$f_{j,0} = x - \sum_{i=1}^{\beta-1} f_{j,i}$
		$c_a \stackrel{?}{=} \text{aCom}(A, C, D)$
		$\|f_{j,i}\| \stackrel{?}{\leq} 60\sqrt{dk}\ \ \forall j, \forall i \neq 0$
		$\|f_{j,0}\| \stackrel{?}{\leq} 60\sqrt{dk(\beta-1)}\ \ \forall j$
		$\|\boldsymbol{z}_b\|, \|\boldsymbol{z}_c\| \stackrel{?}{\leq} 24\sqrt{2}\mathcal{B}md$
		$\boldsymbol{f} := (f_{0,0}, \ldots, f_{k-1,\beta-1})$
		$\boldsymbol{g} := \{f_{j,i}(x - f_{j,i})\}_{j,i=0}^{k-1,\beta-1}$
		$xB + A \stackrel{?}{=} \text{Com}_{ck}(\boldsymbol{f}\ ;\ \boldsymbol{z}_b)$
		$xC + D \stackrel{?}{=} \text{Com}_{ck}(\boldsymbol{g}\ ;\ \boldsymbol{z}_c)$

Protocol 1: Lattice-based Σ-protocol for $\mathcal{R}_{\text{bin}}$.

Also, since $f_{j,0} = x - \sum_{i=1}^{\beta-1} f_{j,i}$ for all $0 \le j \le k-1$, the distributions of $f_{j,0}$'s are statistically close to $D^d_{12\sqrt{k(\beta-1)}}$ by Lemma 8. Therefore, if the prover does not abort, and since $d \ge 7$ and $md \ge 86$,[10] by Lemma 3 except with probability at most 2^{-100}, we have,

$$\|f_{j,i}\| \le 5 \cdot 12\sqrt{k} \cdot \sqrt{d} = 60\sqrt{dk}, \qquad \forall j \in [0, k-1], \forall i \in [1, \beta-1],$$
$$\|f_{j,0}\| \le 5 \cdot 12\sqrt{k(\beta-1)} \cdot \sqrt{d} = 60\sqrt{dk(\beta-1)}, \qquad \forall j \in [0, k-1],$$

and $\|\boldsymbol{z}_b\|, \|\boldsymbol{z}_c\| \le 2 \cdot 12\mathcal{B}\sqrt{2md} \cdot \sqrt{md} = 24\sqrt{2}\mathcal{B}md$, proving the bounds on the norms. The other verification steps follow via straightforward investigation.
SHVZK: Given a challenge x, the simulator outputs $(\mathrm{aCom}(0), x, \perp)$ indicating an abort with probability $1 - \frac{1}{M^2}$. Otherwise, it picks $C \leftarrow R_q^n$, $f_{j,i} \leftarrow D^d_{12\sqrt{k}}$ for all $0 \le j \le k-1$ and $1 \le i \le \beta - 1$, and also $\boldsymbol{z}_b, \boldsymbol{z}_c \leftarrow D^{md}_{12\mathcal{B}\sqrt{2md}}$. Then, it sets $f_{j,0} = x - \sum_{i=1}^{\beta-1} f_{j,i}$ for all $j = 0, \ldots, k-1$. Finally, it computes $A = \mathrm{Com}_{ck}(\boldsymbol{f}\ ;\ \boldsymbol{z}_b) - xB$, $D = \mathrm{Com}_{ck}(\{f_{j,i}(x - f_{j,i})\}_{j,i}\ ;\ \boldsymbol{z}_c) - xC$ and $(c_a, d_a) = \mathrm{aCom}(A, C, D)$ where $\boldsymbol{f} = (f_{0,0}, \ldots, f_{k-1,\beta-1})$. It outputs the simulated transcript $(c_a, x, (d_a, \{f_{j,i}\}_{j=0,i=1}^{k-1,\beta-1}, A, C, D, \boldsymbol{z}_b, \boldsymbol{z}_c))$.

Note that the largest absolute coefficient in any of the randomness is at most $144\mathcal{B}\sqrt{2md}$ except with probability at most 2^{-100} by Lemma 3. Therefore, by Lemma 2, all of the commitments computed are guaranteed to be statistically hiding due to the bounds on q and $2\mathcal{B}$, ensuring all of the sampled randomnesses have enough min-entropy. Hence, if the protocol is not aborted, the real and simulated transcripts are indistinguishable by Lemma 8 and the hiding property of the commitment scheme. If an abort occurs, then the indistinguishability is satisfied due to hiding property of aCom and the fact that the probability of having an abort is the same for all x.

3-special soundness: Given 3 accepting transcripts, by the binding property of aCom, we have the tuples $(A, C, D, x, f_{0,1}, \ldots, f_{k-1,\beta-1}, \boldsymbol{z}_b, \boldsymbol{z}_c)$, $(A, C, D, x', f'_{0,1}, \ldots, f'_{k-1,\beta-1}, \boldsymbol{z}'_b, \boldsymbol{z}'_c)$, $(A, C, D, x'', f''_{0,1}, \ldots, f''_{k-1,\beta-1}, \boldsymbol{z}''_b, \boldsymbol{z}''_c)$. Let $\boldsymbol{f} = (f_{0,0}, \ldots, f_{k-1,\beta-1})$, $\boldsymbol{f}' = (f'_{0,0}, \ldots, f'_{k-1,\beta-1})$, $\boldsymbol{f}'' = (f''_{0,0}, \ldots, f''_{k-1,\beta-1})$ where $f_{j,0}, f'_{j,0}, f''_{j,0}$'s are computed in the way done by the verifier. Then, by Step 7 in the verification, we have $xB + A = \mathrm{Com}_{ck}(\boldsymbol{f}\ ;\ \boldsymbol{z}_b)$ and $x'B + A = \mathrm{Com}_{ck}(\boldsymbol{f}'\ ;\ \boldsymbol{z}'_b)$. By subtracting the equations and multiplying both sides by $2(x - x')^{-1}$,

$$2B = \mathrm{Com}_{ck}(2(x-x')^{-1}(\boldsymbol{f} - \boldsymbol{f}')\ ;\ 2(x-x')^{-1}(\boldsymbol{z}_b - \boldsymbol{z}'_b)).$$

This gives us openings of $2B$ as $\hat{\boldsymbol{b}} = (\hat{b}_{0,0}, \ldots, \hat{b}_{k-1,\beta-1})$ and $\hat{\boldsymbol{r}}_b$. Note that

$$\begin{aligned}\|\hat{\boldsymbol{r}}_b\| &= \left\|2(x-x')^{-1}(\boldsymbol{z}_b - \boldsymbol{z}'_b)\right\| \le \sqrt{d} \cdot \left\|2(x-x')^{-1}\right\| \cdot \|(\boldsymbol{z}_b - \boldsymbol{z}'_b)\| \\ &\le d \cdot \|(\boldsymbol{z}_b - \boldsymbol{z}'_b)\| \le d \cdot 2 \cdot 24\sqrt{2}\mathcal{B}md = 48\sqrt{2}\mathcal{B}md^2,\end{aligned}$$

which proves the required norm-bound on the extracted randomness.

[10] These conditions also hold for our concrete parameters in Table 3.

We can also recover openings of $2A$ by computing $\hat{a}_{j,i} = 2f_{j,i} - x \cdot \hat{b}_{j,i}$ and $\hat{\boldsymbol{r}}_a = 2\boldsymbol{z}_b - x \cdot \hat{\boldsymbol{r}}_b$. Similarly, we get openings $\hat{c}_{j,i}$ and $\hat{d}_{j,i}$ of $2C$ and $2D$, respectively, by Step 8 of the verification. Now, by Step 8 of the verification, we have

$$\begin{aligned} 2 \cdot \left(x \cdot \hat{c}_{j,i} + \hat{d}_{j,i} \right) &= 2 \cdot (2f_{j,i}(x - f_{j,i})) = 2f_{j,i}(2x - 2f_{j,i}) \\ &= x^2 \left[\hat{b}_{j,i}(2 - \hat{b}_{j,i}) \right] + x \left[2\hat{a}_{j,i}(1 - \hat{b}_{j,i}) \right] - \hat{a}_{j,i}^2, \end{aligned}$$

which implies $x^2 \left[\hat{b}_{j,i}(2 - \hat{b}_{j,i}) \right] + x \left[2\hat{a}_{j,i}(1 - \hat{b}_{j,i}) - 2\hat{c}_{j,i} \right] - \hat{a}_{j,i}^2 - 2\hat{d}_{j,i} = 0$. If this equality holds for 3 distinct challenges x, x' and x'', then we can write this system of equations as

$$\begin{pmatrix} 1 & x & x^2 \\ 1 & x' & x'^2 \\ 1 & x'' & x''^2 \end{pmatrix} \cdot \begin{pmatrix} -\hat{a}_{j,i}^2 - 2\hat{d}_{j,i} \\ 2\hat{a}_{j,i}(1 - \hat{b}_{j,i}) - 2\hat{c}_{j,i} \\ \hat{b}_{j,i}(2 - \hat{b}_{j,i}) \end{pmatrix} = \mathbf{0} \qquad \text{over } R_q.$$

The left-most matrix is a Vandermonde matrix $\boldsymbol{V}$, which is invertible by Lemma 7. Therefore, we get $\hat{b}_{j,i}(2 - \hat{b}_{j,i}) = 0$ over R_q. Further, we have

$$\begin{aligned} \|\hat{b}_{j,i}\|_\infty &= \|2(x - x')^{-1}(f_{j,i} - f'_{j,i})\|_\infty \leq \left\|2(x - x')^{-1}\right\| \cdot \left\|f_{j,i} - f'_{j,i}\right\| \\ &\leq \sqrt{d} \cdot \left\|f_{j,i} - f'_{j,i}\right\| \leq \sqrt{d} \cdot 2 \cdot \left(60\sqrt{dk(\beta - 1)}\right) = 120d\sqrt{k(\beta - 1)}. \end{aligned}$$

Since $q > 2\left(120d\sqrt{k(\beta - 1)}\right)^2 \geq 2\|\hat{b}_{j,i}\|_\infty^2$, message openings of B satisfy $b_{j,i} = 2^{-1}\hat{b}_{j,i} \in \{0, 1\}$ by Lemma 5. Moreover, by construction, for all $i = 0, \ldots, k - 1$,

$$2x = \sum_{j=0}^{\beta-1} 2f_{j,i} = x \cdot \sum_{j=0}^{\beta-1} 2b_{j,i} + \sum_{j=0}^{\beta-1} \hat{a}_{j,i} = 2x \cdot \sum_{j=0}^{\beta-1} b_{j,i} + \sum_{j=0}^{\beta-1} \hat{a}_{j,i}.$$

If this is true for 2 distinct challenges x and x', then $\sum_{i=0}^{\beta-1} b_{j,i} = 1$ for all $j = 0, \ldots, k - 1$ as desired. □

5 Lattice-based One-out-of-Many Protocol

We are now ready to describe our main protocol. Let $\delta_{j,i}$ denote the Kronecker's delta such that $\delta_{j,i} = 1$ if $j = i$, and $\delta_{j,i} = 0$ otherwise. The prover's goal in the protocol is to show that he knows the randomness within a commitment to zero among a list of N commitments. Similar to the previous works [13, 7], we assume that the number of commitments satisfy $N = \beta^k$, which can be realised by using the same commitment multiple times until such an N is reached. Let c_ℓ be the prover's commitment for $0 \leq \ell \leq N - 1$, and $L = \{c_0, \ldots, c_{N-1}\}$ be the list of all commitments. The main idea is to prove knowledge of the index ℓ such that $\sum_{i=0}^{N-1} \delta_{\ell,i} c_i$ is a commitment to zero. Note that $\delta_{\ell,i} = \prod_{j=0}^{k-1} \delta_{\ell_j, i_j}$ where $\ell = (\ell_0, \ldots, \ell_{k-1})$ and $i = (i_0, \ldots, i_{k-1})$ are representations in base β. The relations for the protocol are given in Definition 6.

Definition 6. *For positive real numbers $\mathcal{T}$ and $\hat{\mathcal{T}}$, we define the following relations to be used in Protocol 2.*

$$\mathcal{R}_{1/\mathrm{N}}(\mathcal{T}) = \left\{ \begin{array}{l} ((ck,(c_0,\ldots,c_{N-1})),(\ell,\boldsymbol{r})) \ : \ (c_i \in R_q^n \ \ \forall i \in [0,N-1]) \wedge \\ \quad \ell \in \{0,\ldots,N-1\} \ \wedge \ \|\boldsymbol{r}\| \leq \mathcal{T} \ \wedge \ c_\ell = \mathrm{Com}_{ck}(\boldsymbol{0}\ ;\ \boldsymbol{r}) \end{array} \right\}.$$

$$\mathcal{R}'_{1/\mathrm{N}}(\hat{\mathcal{T}}) = \left\{ \begin{array}{l} ((ck,(c_0,\ldots,c_{N-1})),(\ell,\hat{\boldsymbol{r}})) \ : \ (c_i \in R_q^n \ \ \forall i \in [0,N-1]) \wedge \\ \quad \ell \in \{0,\ldots,N-1\} \ \wedge \ \|\hat{\boldsymbol{r}}\| \leq \hat{\mathcal{T}} \ \wedge \ 2^k c_\ell = \mathrm{Com}_{ck}(\boldsymbol{0}\ ;\ \hat{\boldsymbol{r}}) \end{array} \right\}.$$

For each $0 \leq j \leq k-1$, the prover commits to a sequence $(\delta_{\ell_j,0},\ldots,\delta_{\ell_j,\beta-1})$ and proves that it is a binary sequence with Hamming weight one using Protocol 1. As given in Protocol 1, the prover responds with $f_{j,i} = x \cdot \delta_{\ell_j,i} + a_{j,i}$ upon receiving a challenge x. Now, let us concentrate on the product $\prod_{j=0}^{k-1} f_{j,i_j} =: p_i(x)$. Observe that for all $i \in \{0,\ldots,N-1\}$,

$$p_i(x) = \prod_{j=0}^{k-1} \left(x \cdot \delta_{\ell_j,i_j} + a_{j,i_j}\right) = \prod_{j=0}^{k-1} x \cdot \delta_{\ell_j,i_j} + \sum_{j=0}^{k-1} p_{i,j} x^j = \delta_{\ell,i} x^k + \sum_{j=0}^{k-1} p_{i,j} x^j, \ (4)$$

for some coefficients $p_{i,j}$'s depending on ℓ and $a_{j,i}$, which means that $p_{i,j}$'s can be computed by the prover before receiving a challenge. Now, since $\delta_{\ell,i} = 1$ if and only if $i = \ell$, the only p_i of degree k is p_ℓ. Then, the idea is to send some E_j's in the initial message, which will later be used by the verifier to cancel out the coefficients of low order terms $1, x, \ldots, x^{k-1}$, and the coefficient of x^k will be $\sum_{i=0}^{N-1} \delta_{\ell,i} c_i = c_\ell$, which corresponds to the prover's commitment. The full protocol is described in Protocol 2.

Theorem 2. *Using the notations in Protocol 1 and Protocol 2, let $\boldsymbol{f}_1 := (f_{0,1},\ldots,f_{k-1,\beta-1})$, $\boldsymbol{\delta}_1 := (\delta_{\ell_0,1},\ldots,\delta_{\ell_{k-1},\beta-1})$, M be the constant defined in Lemma 8. Assume that $d \geq 7$, $md \geq 86$, $2\mathcal{B} \geq q^{n/m} 2^{\frac{2\lambda}{md}}$ and $q > \max\left\{2\left(120d\sqrt{k(\beta-1)}\right)^2, 8\left(144\mathcal{B}\sqrt{3md}\right)^2\right\}$. Protocol 2 with*

$$p_{1/N} = \frac{D^{k(\beta-1)d}_{12\sqrt{k}}(\boldsymbol{f}_1)}{M D^{k(\beta-1)d}_{x\cdot\boldsymbol{\delta}_1,12\sqrt{k}}(\boldsymbol{f}_1)} \cdot \frac{D^{3md}_{12\mathcal{B}\sqrt{3md}}((\boldsymbol{z}_b,\boldsymbol{z}_c,\boldsymbol{z}))}{M D^{3md}_{x\cdot(\boldsymbol{r}_b,\boldsymbol{r}_c,\boldsymbol{r}),12\mathcal{B}\sqrt{3md}}((\boldsymbol{z}_b,\boldsymbol{z}_c,\boldsymbol{z}))} \quad (5)$$

is a $(k'+1)$-special sound Σ-protocol (as given in Definition 4) for the relations $\mathcal{R}_{1/\mathrm{N}}(\mathcal{B}\sqrt{md})$ and $\mathcal{R}'_{1/\mathrm{N}}(24\sqrt{3}m\mathcal{B}\cdot(k+1)\cdot d^{k+1})$ with a soundness error $\frac{k'}{2d}$ and a completeness error $1 - \frac{1}{M^2}$ where $k' = \max\{2,k\}$.

Remark 3. In Protocol 2, the rejection sampling is applied in two parallel steps: 1) on $\boldsymbol{f}_1$ and 2) on the whole vector $(\boldsymbol{z}_b,\boldsymbol{z}_c,\boldsymbol{z})$. Furthermore, $\boldsymbol{r}_a$ and $\boldsymbol{r}_d$ in $\mathcal{P}_{\mathrm{bin}}(ck,B,(\boldsymbol{\delta},\boldsymbol{r}_b))[1-8]$ are drawn from $D^{md}_{12\mathcal{B}\sqrt{3md}}$ instead of $D^{md}_{12\mathcal{B}\sqrt{2md}}$ as the rejection sampling is now done on a $(3md)$-dimensional vector.

Proof. The completeness and SHVZK properties can be proven in a similar way as done in the proof of Protocol 1, and are deferred to Appendix B.

$\mathcal{P}(ck, (c_0, \ldots, c_{N-1}), (\ell, \boldsymbol{r}))$	$\mathcal{V}(ck, (c_0, \ldots, c_{N-1}))$
1: $\boldsymbol{r}_b \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$	
2: $\boldsymbol{\delta} = (\delta_{\ell_0,0}, \ldots, \delta_{\ell_{k-1},\beta-1})$	
3: $B = \mathrm{Com}_{ck}(\boldsymbol{\delta}\ ;\ \boldsymbol{r}_b)$	
4: $A, C, D \leftarrow \mathcal{P}_{\mathrm{bin}}(ck, B, (\boldsymbol{\delta}, \boldsymbol{r}_b))[1-8]$	
5: **for** $j = 0, \ldots, k-1$ **do**	
6: $\quad \boldsymbol{\rho}_j \leftarrow D^{md}_{12\mathcal{B}\sqrt{3md/k}}$	
7: $\quad E_j = \sum_{i=0}^{N-1} p_{i,j} c_i + \mathrm{Com}_{ck}(\boldsymbol{0}\ ;\ \boldsymbol{\rho}_j)$	
using $p_{i,j}$'s from (5)	
8: $(c_a, d_a) = \mathrm{aCom}(A, B, C, D, \{E_j\})$	
$\xrightarrow{\quad c_a \quad}$	
	$\omega \leftarrow \{0, \ldots, 2d-1\}$
$\xleftarrow{\quad x = X^{\omega} \quad}$	
9: $\boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z}_c \leftarrow \mathcal{P}_{\mathrm{bin}}(x)[10-12]$	
10: $\boldsymbol{z} = x^k \cdot \boldsymbol{r} - \sum_{j=0}^{k-1} x^j \cdot \boldsymbol{\rho}_j$	
abort with prob. $(1 - p_{1/N})$ from (5)	
Return $\perp$ if aborted.	$\xrightarrow{\quad d_a, \boldsymbol{f}_1, B, \boldsymbol{z}, \{E_j\}_{j=0}^{k-1}, \boldsymbol{R} := (A, C, D, \boldsymbol{z}_b, \boldsymbol{z}_c) \quad}$
	1: $\mathcal{V}_{\mathrm{bin}}(ck, B, x, \boldsymbol{f}_1, \boldsymbol{R})[1,2,6,7] \stackrel{?}{=} 1$
	2: $c_a \stackrel{?}{=} \mathrm{aCom}(A, B, C, D, \{E_j\})$
	3: $\lVert f_{j,i} \rVert \stackrel{?}{\leq} 60\sqrt{dk}\ \ \forall j, \forall i \neq 0$
	4: $\lVert f_{j,0} \rVert \stackrel{?}{\leq} 60\sqrt{dk(\beta-1)}\ \ \forall j$
	5: $\lVert \boldsymbol{z} \rVert, \lVert \boldsymbol{z}_b \rVert, \lVert \boldsymbol{z}_c \rVert \stackrel{?}{\leq} 24\sqrt{3}\mathcal{B}md$
	6: $\sum_{i=0}^{N-1} \left(\prod_{j=0}^{k-1} f_{j,i_j} \right) c_i - \sum_{j=0}^{k-1} E_j x^j$ $\stackrel{?}{=} \mathrm{Com}_{ck}(\boldsymbol{0}\ ;\ \boldsymbol{z})$ for $i = (i_0, \ldots, i_{k-1})$.

Protocol 2: Lattice-based Σ-protocol for $\mathcal{R}_{1/N}$.
$\mathcal{P}_{\mathrm{bin}}(ck, B, (\boldsymbol{\delta}, \boldsymbol{r}_b))[1-8]$ denotes running the same steps from 1 to 8 done by $\mathcal{P}_{\mathrm{bin}}$ in Protocol 1. Similar notation is used for $\mathcal{V}_{\mathrm{bin}}$.

$(k+1)$**-special soundness:** Given $(k+1)$ distinct challenges $x_0, \ldots, x_k$, by the binding property of aCom, we have $(k+1)$ accepting responses with the same $(A, B, C, D, \{E_j\})$. Suppose that $((f_{j,i}^{(0)}, \boldsymbol{z}^{(0)}), \ldots, (f_{j,i}^{(k)}, \boldsymbol{z}^{(k)}))$ are produced and $k > 1$. We first use 3-special soundness of Protocol 1 to extract openings $\hat{b}_{j,i}$ and $\hat{a}_{j,i}$ of $2B$ and $2A$, respectively. We can also obtain $b_{j,i}$ such that $\hat{b}_{j,i} = 2b_{j,i}$ as discussed previously, and it is guaranteed that $b_{j,i} \in \{0, 1\}$ and $\sum_{i=0}^{\beta-1} b_{j,i} = 1$. From here, we can obtain the digits ℓ_j by choosing $\ell_j = i^*$ for which $b_{j,i^*} = 1$. Then, we construct the index ℓ as $\ell = \sum_{j=0}^{k-1} \beta^j \ell_j$.

Using $2b_{j,i}$ and $\hat{a}_{j,i}$, we can compute $\hat{p}_i(x) = 2^k \prod_{j=0}^{k-1} f_{j,i_j} = \prod_{j=0}^{k-1} 2f_{j,i_j} = \prod_{j=0}^{k-1}(x \cdot 2b_{j,i_j} + \hat{a}_{j,i_j})$. Note that $\hat{p}_\ell(x)$ is the only such polynomial of degree k in x. Thus, the last verification step, when both sides are multiplied by 2^k, can be rewritten as $\sum_{i=0}^{N-1} \hat{p}_i(x)c_i - \sum_{j=0}^{k-1} 2^k E_j x^j = \mathrm{Com}_{ck}(\mathbf{0}\ ;\ 2^k \boldsymbol{z})$. Separating the term of degree k with respect to x, we get

$$x^k \cdot 2^k c_\ell + \sum_{j=0}^{k-1} \tilde{E}_j x^j = \mathrm{Com}_{ck}(\mathbf{0}\ ;\ 2^k \boldsymbol{z}), \tag{6}$$

where $\tilde{E}_j$'s are the coefficients of the monomials x^j of degree strictly less than k. Now, we know that Equation 6 holds for distinct challenges $x_0, \ldots, x_k$, which can be represented as a system of equations where $x_0, \ldots, x_k$ form a Vandermonde matrix $\boldsymbol{V}$ as in Section 3.2. From the discussion in Section 3.2, $\boldsymbol{V}$ is invertible and we can obtain a linear combination $\alpha_0, \ldots, \alpha_k$ of copies of Equation 6 with respect to different challenges that produces the vector $(0, \ldots, 0, 1)$. This gives

$$2^k c_\ell = \sum_{e=0}^{k} \alpha_e \left(x_e^k \cdot 2^k c_\ell + \sum_{j=0}^{k-1} \tilde{E}_j x_e^j \right) = \mathrm{Com}_{ck}(\mathbf{0}\ ;\ 2^k \sum_{e=0}^{k} \alpha_e \boldsymbol{z}^{(e)}). \tag{7}$$

An opening of $2^k c_\ell$ to the message $\mathbf{0}$ with randomness $\boldsymbol{r}_{ext} = 2^k \sum_{e=0}^{k} \alpha_e \boldsymbol{z}^{(e)}$ is obtained. The claimed bound on the norm of $\boldsymbol{r}_{ext}$ is proved in Lemma 9.

Finally, we assumed that $k > 1$. If $k = 1$, then we still need at least 3 challenges to be able to prove special soundness due to the 3-special soundness of Protocol 1. In this case, Protocol 2 is also 3-special sound. □

Lemma 9. *Let $N = \beta^k$ (i.e., $k = \log_\beta N$) for a given base $2 \leq \beta \leq N$. For the extracted randomness $\boldsymbol{r}_{ext}$ from $2^k c_\ell$, we have $\|\boldsymbol{r}_{ext}\| \leq 24\sqrt{3}m\mathcal{B} \cdot (k+1) \cdot d^{k+1}$.*

Proof.
$$\begin{aligned}
\|\boldsymbol{r}_{ext}\| = \left\| 2^k \textstyle\sum_{e=0}^{k} \alpha_e \boldsymbol{z}^{(e)} \right\| &\leq (k+1) \cdot \max_e \left\| 2^k \alpha_e \boldsymbol{z}^{(e)} \right\| \quad \text{(by Lemma 4)} \\
&\leq (k+1) \cdot \sqrt{d} \cdot \max_e \|2^k \alpha_e\| \cdot \max_e \|\boldsymbol{z}^{(e)}\| \quad \text{(by Lemma 4)} \\
&\leq (k+1) \cdot \sqrt{d} \cdot d^{k-0.5} \cdot \max_e \|\boldsymbol{z}^{(e)}\| \qquad \text{(by Lemma 7)} \\
&\leq (k+1) \cdot d^k \cdot (24\sqrt{3}\mathcal{B}md) \qquad \text{(guaranteed by Protocol 2)} \\
&= 24\sqrt{3}m\mathcal{B} \cdot (k+1) \cdot d^{k+1}. \quad \square
\end{aligned}$$

It is easy to see from Lemma 9 that the norm of the extracted randomness, and thus the size of q, grows with $d^k = d^{\log_\beta N}$. If one is to rely on the Ring-SIS problem and use a base $\beta = 2$, then this growth would be very rapid, yielding a very inefficient scheme. This justifies our choice of working with Module-SIS problem and choosing large base values β as we shall see in Section 6.3.

6 Lattice-based Ring Signature

Building on top of the lattice-based one-out-of-many proof described in Section 5, we introduce a short lattice-based ring signature in this section.

6.1 Definitions

We recall the standard definitions and properties of a ring signature, which consists of four algorithms (**RSetup**, **RKeygen**, **RSign**, **RVerify**) as follows.

- $pp \leftarrow$**RSetup**(1^λ)**:** On input a security parameter λ, generates the public parameters pp, which are assumed to be made available to everyone.
- $(pk, sk) \leftarrow$**RKeygen**(pp)**:** Given pp, outputs a public-secret key pair (pk, sk).
- $\sigma \leftarrow$**RSign**$_{pp,sk}(\mathcal{M}, L)$**:** On input a message $\mathcal{M}$ and a ring L of public keys, outputs a signature σ on $\mathcal{M}$ with respect to L. It is required that sk is generated by **RKeygen**(pp), and the corresponding public key pk is in L.
- $\{0,1\} \leftarrow$**RVerify**$_{pp}(\mathcal{M}, L, \sigma)$**:** On input a purported signature σ on a message $\mathcal{M}$ with respect to a ring L, checks the validity of σ. If it is valid, outputs 1 or outputs 0 otherwise.

Definition 7 (Correctness). *A ring signature (**RSetup**, **RKeygen**, **RSign**, **RVerify**) provides statistical correctness if for any $pp \leftarrow$ **RSetup**, any $(pk, sk) \leftarrow$**RKeygen**(pp), any L such that $pk \in L$, and any $\mathcal{M} \in \{0,1\}^*$, the following is satisfied*

$$\Pr[\ \mathbf{RVerify}_{pp}(\mathcal{M}, L, \mathbf{RSign}_{pp,sk}(\mathcal{M}, L)) = 1\ ] = 1 - \mathrm{negl}(\lambda).$$

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

$$\Pr\left[\begin{matrix} pp \leftarrow \mathbf{RSetup}(1^\lambda); (\mathcal{M}, i_0, i_1, L) \leftarrow \mathcal{A}^{\mathbf{RKeygen}(pp)} \\ b \leftarrow \{0,1\};\ \sigma \leftarrow \mathbf{RSign}_{pp}(sk_{i_b}, \mathcal{M}, L) \end{matrix} : \mathcal{A}(\sigma) = b\right] = \frac{1}{2} + \mathrm{negl}(\lambda),$$

*where $pk_{i_0}, pk_{i_1} \in L$ and $(pk_{i_0}, sk_{i_0}), (pk_{i_1}, sk_{i_1}) \leftarrow$ **RKeygen**(pp).*

Definition 9 (Unforgeability w.r.t. insider corruption). *A ring signature (**RSetup**, **RKeygen**, **RSign**, **RVerify**) is unforgeable with respect to insider collusion if for all PPT adversary $\mathcal{A}$*

$$\Pr\left[\begin{matrix} pp \leftarrow \mathbf{RSetup}(1^\lambda); \\ (\mathcal{M}, L, \sigma) \leftarrow \mathcal{A}^{\mathrm{PKGen,Sign,Corrupt}}(pp) \end{matrix} : \mathbf{RVerify}(\mathcal{M}, L, \sigma) = 1\right] = \mathrm{negl}(\lambda),$$

where

– PKGen*: on the i-th query, picks a randomness ρ_i, runs $(pk_i, sk_i) \leftarrow \mathbf{RKeygen}(pp; \rho_i)$ and returns pk_i.*
– Sign*(i, M, L): returns $\sigma \leftarrow \mathbf{RSign}_{pp,sk_i}(\mathcal{M}, L)$, provided (pk_i, sk_i) has been generated by* PKGen.
– Corrupt*(i): returns ρ_i (enabling the computation of sk_i), provided (pk_i, sk_i) has been generated by* PKGen.
– *$\mathcal{A}$ outputs $(\mathcal{M}, L, \sigma)$ such that* Sign*$(\cdot, \mathcal{M}, L)$ has not been queried and L only contains pk_i's generated by* PKGen *where* Corrupt*(i) has not been queried.*

6.2 Construction

First, we summarise the assumptions on the parameters as follows.

Assumption 1. *Assume that $d \geq 7$, $md \geq 86$, $2\mathcal{B} \geq q^{n/m} \cdot 2^{2\lambda/(md)}$ and $q > \max\left\{2\left(120d\sqrt{kr(\beta-1)}\right)^2, 8\left(144\mathcal{B}\sqrt{3mdr}\right)^2\right\}$ with $q \equiv 5 \bmod 8$.*

Let $N = \beta^k$ for $2 \leq \beta \leq N$, and n, m be fixed positive integers. Suppose that r non-aborting runs of Protocol 2 is enough to get negligible soundness error in λ. Further, assume that Assumption 1 holds. Let $CMT = (A, B, C, D, \{E_j\}_{j=0}^{k-1})$ and $RSP = (\{f_{j,i}\}_{j=0,i=1}^{k-1,\beta-1}, \boldsymbol{z}, \boldsymbol{z}_b, \boldsymbol{z}_c)$ be the corresponding values from Protocol 2. We now give our lattice-based ring signature construction.

– **RSetup(1^λ):** Pick $\boldsymbol{G} \leftarrow R_q^{n\times(m+k\beta)}$ and a hash function $H : \{0,1\}^* \to \mathcal{C}^r$ for challenge space $\mathcal{C}$. Return the commitment key $ck = \boldsymbol{G}$ and H as $pp = (ck, H)$.
– **RKeygen(pp):** Pick $\boldsymbol{r} \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$ and compute $c = \mathrm{Com}_{ck}(\mathbf{0}\ ;\ \boldsymbol{r})$ using ck where $\mathbf{0}$ is the all-zero vector. Return $(pk, sk) = (c, \boldsymbol{r})$.
– **RSign$_{pp,sk}(\mathcal{M}, L)$:** Parse $L = (c_0, \ldots, c_{N-1})$ with $c_\ell = \mathrm{Com}_{ck}(\mathbf{0}\ ;\ sk)$ where $\ell \in \{0, \ldots, N-1\}$. Continue as follows.
  1. Generate $(CMT_1, \ldots, CMT_r)$ by running $\mathcal{P}(ck, (c_0, \ldots, c_{N-1}), (\ell, sk))[1-7]$ r-times in parallel with the modifications given in Remark 4.
  2. Compute $\boldsymbol{x} = (x_1, \ldots, x_r) = H(ck, \mathcal{M}, L, (CMT_1, \ldots, CMT_r))$.
  3. Compute RSP_i by running $\mathcal{P}(x_i)[9, 10]$ with CMT_i for all $i \in \{1, \ldots, r\}$.
  4. If $(RSP_i) \neq \perp$ for all $i \in \{1, \ldots, r\}$, then return
$$\sigma = (\{CMT_i\}_{i=1}^r, \boldsymbol{x}, \{RSP_i\}_{i=1}^r)\,.$$
  5. Otherwise go to Step 1 (repeat at most $\frac{-\lambda}{\log(1-1/M^2)}$ times).
  6. If the maximum number of iterations $\frac{-\lambda}{\log(1-1/M^2)}$ is reached, return $\perp$. [11]
– **RVerify$_{pp}(\mathcal{M}, L, \sigma)$:** If $\sigma = \perp$, return 0. Otherwise, parse $L = (c_0, \ldots, c_{N-1})$, $\sigma = (\{CMT_i\}_{i=1}^r, \boldsymbol{x}, \{RSP_i\}_{i=1}^r)$, and $\boldsymbol{x} = (x_1, \ldots, x_r)$. Proceed as follows.

[11] We note that a check on the number of repetitions is put merely to make sure that the algorithm terminates with probability exactly 1. Otherwise, in common with all other schemes using rejection sampling, there is always a non-zero probability that all rejection sampling steps fail for any finite number of iterations. We show in Theorem 3 that the probability is negligibly small (i.e., $2^{-\lambda}$) for the given maximum number of iterations. We refer to Appendix C for further discussion.

1. If $\boldsymbol{x} \neq H(ck, \mathcal{M}, L, (CMT_1, \ldots, CMT_r))$, return 0.
2. For each $i \in \{1, \ldots, r\}$:
   (a) Check all the verification steps of Protocol 2 using CMT_i, x_i and RSP_i except the step involving aCom.
   (b) If verification fails, return 0.
3. Return 1.

Remark 4. In **RSign**, the rejection sampling is applied to r-concatenated vectors at once. That is, it is applied on $(\boldsymbol{f}_1^1, \ldots, \boldsymbol{f}_1^r)$ and $(\boldsymbol{z}^1, \boldsymbol{z}_b^1, \boldsymbol{z}_c^1, \ldots, \boldsymbol{z}^r, \boldsymbol{z}_b^r, \boldsymbol{z}_c^r)$. This means that $f_{j,i} \in D^d_{12\sqrt{kr}}$ $(i \neq 0)$ and $\boldsymbol{z}, \boldsymbol{z}_b, \boldsymbol{z}_c \in D^{md}_{12\mathcal{B}\sqrt{3mdr}}$, and hence we require $q > \max\left\{2\left(120d\sqrt{kr(\beta-1)}\right)^2, 8\left(144\mathcal{B}\sqrt{3mdr}\right)^2\right\}$. Also, the prover's non-abort probability changes as

$$p_{1/N} = \frac{D^{k(\beta-1)dr}_{12\sqrt{kr}}(\boldsymbol{f}_1)}{MD^{k(\beta-1)dr}_{x\cdot\boldsymbol{\delta}_1, 12\sqrt{kr}}(\boldsymbol{f}_1)} \cdot \frac{D^{3mdr}_{12\mathcal{B}\sqrt{3mdr}}((\boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}))}{MD^{3mdr}_{x\cdot(\boldsymbol{r}_b, \boldsymbol{r}_c, \boldsymbol{r}), 12\mathcal{B}\sqrt{3mdr}}((\boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}))}.$$

Furthermore, since the extracted randomness will be larger, the relation $\mathcal{R}'_{1/\mathrm{N}}$ becomes $\mathcal{R}'_{1/\mathrm{N}}(24\sqrt{3r} \cdot m\mathcal{B} \cdot (k+1) \cdot d^{k+1})$.

In the ring signature scheme, we need to have r accepting transcripts as part of the signature to obtain a negligible soundness error where the challenge in these transcripts come from the hash of the protocol commitments together. Therefore, we cannot simply generate many commitments, then hash them all and choose the non-rejected ones to constitute the signature. Because then the verification will not succeed as the hash values will not match. Of course, we can put all the initial messages (including the ones with $\perp$ as a response) in the signature, enabling the signature verification to succeed, but that would make the scheme too inefficient in practice, increasing the signature size by factor of at least M^2.

Another (inefficient) approach would be doing the same **RSign** process without the tweaks in Remark 4. The reason why this does not work is that we need to have r independent accepting responses where each one has a $1/M^2$ chance of being accepted. This means that the probability that the `if` statement in Step 4 of **RSign** is successful becomes $1/M^{2r}$, which is exponentially small in λ.

Instead, we introduce a much more efficient approach through the tweaks in Remark 4, yielding a very small overhead in the signature size and allowing us to get a constant overall completeness error of r-repeated Protocol 2 at $1 - 1/M^2$. Note that these tweaks do not affect the soundness error of individual protocol runs as the extraction still works with $k+1$ accepting transcripts. Only the extracted witness norm is increased as given in Remark 4 since the bound on $\|\boldsymbol{z}\|$ changes from $24\sqrt{3}\mathcal{B}md$ to $24\sqrt{3r}\mathcal{B}md$ in Protocol 2 and Lemma 9.

Theorem 3. *If Assumption 1 holds, the ring signature scheme described by (**RSetup**, **RKeygen**, **RSign**, **RVerify**) as above provides statistical correctness with a negligible correctness error and statistical anonymity. The expected number of iterations for **RSign** is $M^2 = O(1)$.*

Proof. **Correctness:** If Step 4 in **RSign** returns a signature, then **RVerify** will return 1 with probability 1, and thus a correctness error happens only if the underlying protocol is aborted for all iterations in **RSign**. Note that Protocol 2, when run r times in parallel with the given modifications in Remark 4, is aborted with probability $1 - \frac{1}{M^2}$. Therefore, for $\frac{-\lambda}{\log(1-1/M^2)}$ iterations, we have

$$\Pr[\text{ all iterations fail }] = \left(1 - 1/M^2\right)^{\frac{-\lambda}{\log(1-1/M^2)}} = 2^{-\lambda},$$

which is the probability that **RSign** returns $\sigma = \perp$. Therefore, with probability $1 - 2^{-\lambda}$, **RSign** produces an accepting protocol transcript, which will also be accepted by **RVerify**. Note that a single iteration in **RSign** produces an accepting signature with probability $1/M^2$ and it is terminated as soon as it does so. Hence, the expected number of iterations for **RSign** is $M^2 < 7.5$, which is $O(1)$.

Anonymity: Recall that the prover's responses in Protocol 2 is made independent of the secret values using rejection sampling. In particular, Lemma 8 implies that the distributions of the real and simulated transcripts are statistically close for any given secret key sk. Therefore, by the triangle inequality, the distributions of the real transcripts using sk_1 and sk_2 are also statistically close for any given sk_1 and sk_2. Hence, it is infeasible to distinguish which secret has been used to generate the ring signature, proving anonymity (the same argument can be used to prove statistical witness-indistinguishability of Protocol 2). □

Theorem 4. *If Assumption 1 holds and the commitment scheme defined in Definition 2 is binding with respect to message and randomness domains with maximum Euclidean norms 2^k and $24\sqrt{3r} \cdot m\mathcal{B} \cdot (k+1) \cdot d^{k+1}$, respectively, then the ring signature scheme described by (**RSetup**, **RKeygen**, **RSign**, **RVerify**) is unforgeable with respect to insider corruption in the random oracle model.*

Proof. We prove the unforgeability by showing if there exists a PPT adversary with a polynomial running time and a non-negligible success probability, then one can break the binding property of the commitment scheme for message and randomness domains with maximum Euclidean norms 2^k and $24\sqrt{3r} \cdot m\mathcal{B}(k+1)d^{k+1}$, respectively. This implies that one can find a solution to Module-SIS$_{n,m+k\beta,q,\theta}$ problem for $\theta = 2\sqrt{\left(24\sqrt{3r} \cdot m\mathcal{B}(k+1)d^{k+1}\right)^2 + 2^{2k}}$ by Lemma 2.

Let $\mathcal{C}^r$ be the range of H (i.e., each output component of H is in $\mathcal{C}$), Ψ be the set of all random tapes that could be used by a PPT adversary $\mathcal{A}$, and Φ be the set of all random tapes defining the random oracle H. Let $\boldsymbol{x}_j = (\boldsymbol{x}_{j,1}, \ldots, \boldsymbol{x}_{j,r})$ be the output of j-th random oracle query. We partition Φ into Φ_{j_-}, $\boldsymbol{x}_j$ and Φ_{j_+} so that Φ_{j_-}, Φ_{j_+} represent the sets of random tapes defining the random oracle outputs up to j-th query (i.e., $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{j-1}$) and after j-th query (i.e., $\boldsymbol{x}_{j+1}, \ldots, \boldsymbol{x}_Q$), respectively. Therefore, the tuple $(\phi_{j_-}, \boldsymbol{x}_j, \phi_{j_+})$ defines all the random oracle outputs. Further, assume that $\mathcal{A}$ makes q_P, q_S, q_H queries to PKGen, Sign and the random oracle, respectively. Hence, $\mathcal{A}$ makes at most $Q = q_S + q_H$ random oracle queries in total. Suppose that $\mathcal{A}$ has running time $T_A = \text{poly}(\lambda)$ and a probability $\varepsilon = 1/\text{poly}(\lambda) > 4Q\eta$ of generating a successful forgery where $\eta = (k/|\mathcal{C}|)^r$.

We construct an adversary $\mathcal{D}$ against the binding property of the commitment scheme with a running time $T_B = \text{poly}(\lambda)$ and non-negligible success probability $\varepsilon_B = 1/\text{poly}(\lambda)$. On input a commitment key ck, $\mathcal{D}$ works as follows.

1. Pick $t \leftarrow \{1, \ldots, q_P\}$.
2. Set $pk_t = \text{Com}_{ck}(\mathbf{1} \; ; \; \boldsymbol{r}_t)$ for some randomness $\boldsymbol{r}_t \in \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$ where $\mathbf{1} = (1, 0, \ldots, 0) \in \{0,1\}^k$ (observe that $\|\boldsymbol{r}_t\| \leq \mathcal{B}\sqrt{md}$).
3. Pick $j \leftarrow \{1, \ldots, Q\}$.
4. Pick $\psi \leftarrow \Psi$.
5. Pick $(\phi_{j_-}, \boldsymbol{x}_j, \phi_{j_+}) \leftarrow \Phi_{j_-} \times \mathcal{C} \times \Phi_{j_+}$
6. Run 0: run $\mathcal{A}(\psi, \phi_{j_-}, \boldsymbol{x}_j, \phi_{j_+})$ with access to the oracles PKGen, Sign, Corrupt and the random oracle $H(\phi_{j_-}, \boldsymbol{x}_j, \phi_{j_+})$ simulated as follows. Whenever $\mathcal{A}$ queries PKGen, $\mathcal{D}$ answers as in the real case except for t-th query where pk_t is returned. If $\mathcal{A}$ ever queries Corrupt(t), $\mathcal{D}$ aborts (abort Type I). If $\mathcal{A}$ queries Sign($t, \mathcal{M}, L$), it picks a random challenge vector $\boldsymbol{x}$ and uses SHVZK simulator of Protocol 2 to simulate the proof $(\{CMT_i\}_{i=1}^r, \{RSP_i\}_{i=1}^r)$ (note that only the simulation of non-aborted protocols is used here). Then, the random oracle is programmed as $H(ck, \mathcal{M}, L, \{CMT_i\}_{i=1}^r) = \boldsymbol{x}$, except if $(ck, \mathcal{M}, L, \{CMT_i\}_{i=1}^r)$ has been queried before (abort Type II).
   (a) If $\mathcal{A}$ outputs a forgery σ^0 using j-th random oracle query output $\boldsymbol{x}_j^0$, fix ψ and ϕ_{j_-}.
   (b) Otherwise, abort.
7. Pick $\phi'_1, \ldots, \phi'_{\mathcal{N}} \leftarrow \Phi_{j_+}$
8. Run i (for $i \in \{1, \ldots, \mathcal{N}\}$ where $\mathcal{N}$ is defined below in the analysis): run $\mathcal{A}(\psi, \phi_{j_-}, \boldsymbol{x}_j^i, \phi'_i)$ with access to the oracles PKGen, Sign, Corrupt and the random oracle $H(\phi_{j_-}, \boldsymbol{x}_j^i, \phi'_i)$ where $\boldsymbol{x}_j^i$ is the response of the j-th random oracle query at iteration i.
   (a) $\mathcal{A}$ outputs a forgery σ^i. We say that Run i is j-successful if σ^i was forged with respect to $\boldsymbol{x}_j^i$.
9. If there exists $i^* \in [1, r]$ and $S^* \subseteq \{0, \ldots, \mathcal{N}\}$ with $|S^*| = k + 1$ such that $\mathcal{G}^* := \{x_{j,i^*}^u \; : \; u \in S^*\}$ contains $k + 1$ distinct challenges and σ^u is j-successful for all $u \in S^*$, then run $(k+1)$-special soundness extractor $\mathcal{E}$ of Protocol 2 on input $\{\sigma^u\}_{u \in S^*}$ to extract an opening of $2^k pk_{t'}$ to $(\mathbf{0}, \boldsymbol{s}_{t'})$ for some $1 \leq t' \leq q_P$ where $\|\boldsymbol{s}_{t'}\| \leq 24\sqrt{3r} \cdot m\mathcal{B} \cdot (k+1) \cdot d^{k+1}$.
10. If $t = t'$, return $((2^k \cdot \mathbf{1}, 2^k \cdot \boldsymbol{r}_t), (\mathbf{0}, \boldsymbol{s}_{t'}))$ as a binding collision pair for the commitment scheme. Note that multiplication of $(\mathbf{1}, \boldsymbol{r}_t)$ by 2^k gives a valid opening of $2^k pk_t$, because $d^{k+1} > 2^k$ since $d \geq 7$.
11. Otherwise, abort.

Note that when $\mathcal{D}$ returns a binding collision, there cannot be Type I aborts as the forged signature must be for a ring comprised only of uncorrupted users.

Now, let us analyse this procedure in more details. First, we observe that in each run of $\mathcal{A}$, the view of $\mathcal{A}$ is simulated by $\mathcal{D}$ with the same distribution as in the real attack except for:

– pk_t is a commitment to $\mathbf{1}$ in the simulation by $\mathcal{D}$ whereas it is a commitment to $\mathbf{0}$ in the real attack. By the $2^{-\lambda}$-statistical hiding of the commitment scheme, this reduces the success probability of $\mathcal{A}$ by at most $2^{-\lambda}$.

– There is a statistical distance of at most $O(q_S \cdot 2^{-\lambda})$ between the distribution of signing oracle simulator and that of the real signing oracle.
– A Type II abort occurs during a signing oracle query with probability at most $Q \cdot 2^{-\lambda}$.

By the simulation statistical distance argument above, each run of $\mathcal{A}$ with pk_t and signing oracle simulated by $\mathcal{D}$ succeeds with probability $\tilde{\varepsilon} \geq \varepsilon - O(Q \cdot 2^{-\lambda})$. We say that $(\psi, \phi_{j_-}, \boldsymbol{x}_j, \phi_{j_+}, j)$ is 'winning' if $\mathcal{A}(\psi, \phi_{j_-}, \boldsymbol{x}_j, \phi_{j_+})$ outputs a valid forgery using $\boldsymbol{x}_j$ after Q random oracle queries. Note that there exists a $j^* \in \{1, \ldots, Q\}$ such that $\Pr[\ (\psi, \phi_{j^*_-}, \boldsymbol{x}_{j^*}, \phi_{j^*_+}, j^*) \text{ winning }] \geq \tilde{\varepsilon}/Q$. By the Splitting Lemma (Lemma 7 of [23]), there exists a subset $S \subseteq \Psi \times \Phi_{j^*_-}$ such that

$$\Pr_{\psi \in \Psi, \phi_{j^*_-} \in \Phi_{j^*_-}}[(\psi, \phi_{j^*_-}) \in S] \geq \tilde{\varepsilon}/(2Q), \text{ and}$$

$$\varepsilon' := \Pr_{\boldsymbol{x}_{j^*} \in \mathcal{C}, \phi_{j^*_+} \in \Phi_{j^*_+}}[(\psi, \phi_{j^*_-}, \boldsymbol{x}_{j^*}, \phi_{j^*_+}, j^*) \text{ winning }] \geq \tilde{\varepsilon}/(2Q) \qquad \forall (\psi, \phi_{j^*_-}) \in S.$$

Now, for $(\psi, \phi_{j^*_-}) \in S$, $c \in \mathcal{C}$ and $1 \leq i \leq r$, define $p_i(c)$ as the probability with respect to $\boldsymbol{x}_{j^*} \in \mathcal{C}$ and $\phi_{j^*_+} \in \Phi_{j^*_+}$ that $(\psi, \phi_{j^*_-}, \boldsymbol{x}_{j^*}, \phi_{j^*_+}, j^*)$ is winning and $\boldsymbol{x}_{j^*} = (x_{j^*,1}, \ldots, x_{j^*,r})$ with $x_{j^*,i} = c$.

Claim 1 *If $\varepsilon' > (k/|\mathcal{C}|)^r$, then there exists an $i^* \in [1, r]$ and $\mathcal{G} \subseteq \mathcal{C}$ with $|\mathcal{G}| = k + 1$ such that*

$$p_{i^*}(c) \geq \frac{\varepsilon' - (k/|\mathcal{C}|)^r}{(|\mathcal{C}| - k) \cdot r} =: p \qquad \forall c \in \mathcal{G}.$$

If the claim holds, then a sample of $\mathcal{N} := (k+1) \cdot p^{-1}$ independent and identically distributed winning tuples $(\psi, \phi_{j_-}, \boldsymbol{x}_j, \phi_{j_+}, j)$ will yield a set $\left\{\boldsymbol{x}_j^1, \ldots, \boldsymbol{x}_j^{k+1}\right\}$ such that $\mathcal{G} = \{x_{j,i^*}^1, \ldots, x_{j,i*}^{k+1}\}$ with a probability at least $1 - (k+1)e^{-(k+1)}$, which is greater than $7/10$ for $k \geq 1$ (this comes from the fact that the probability that $\mathcal{N}$ samples do not contain c for some $c \in \mathcal{G}$ is at most $(k+1) \cdot (1-p)^{\mathcal{N}}$). That is, after $\mathcal{N}/\varepsilon'$ rewindings, we obtain a set of $(k+1)$ distinct challenge values of Protocol 2 with respect to the same initial commitment with a high probability.

Now, $\mathcal{N} = \text{poly}(\lambda)$ if $k, |\mathcal{C}|, r = \text{poly}(\lambda)$ and $(\varepsilon' - (k/|\mathcal{C}|)^r)^{-1} \leq \text{poly}(\lambda)$. It is easy to see that the first requirement holds since $|\mathcal{C}| = 2d$, $r = \frac{\lambda}{\log(2d) - \log k}$ and k is a small constant. For the second requirement, we have

$$(\varepsilon' - (k/|\mathcal{C}|)^r)^{-1} = (\varepsilon' - \eta)^{-1} \leq (\varepsilon' - \varepsilon'/2)^{-1} = 2/\varepsilon' \leq \text{poly}(\lambda),$$

where the first inequality holds since $\varepsilon' > 2\eta$. Now, by $(k+1)$-special soundness of Protocol 2, we can use the set $\mathcal{G}$ to extract an opening of $2^k pk_{t'}$ to $(\mathbf{0}, \boldsymbol{s}_{t'})$ for some $t' \in \{1, \ldots, q_P\}$. By the $2^{-\lambda}$-statistical hiding property of the commitment scheme, $t' = t$ with probability at least $\frac{1}{q_P} - 2^{-\lambda}$. Also, $j = j^*$ with probability $\frac{1}{Q}$. Hence, $\mathcal{D}$ succeeds to output a binding collision pair with probability

$$\Pr[j = j^*] \cdot \Pr[(\psi, \phi_{j_-}) \in S] \cdot \Pr\left[\begin{matrix}\mathcal{N} \text{ runs contain } k+1 \\ j\text{-successful distinct challenges}\end{matrix}\right] \cdot \Pr[t = t'] \geq \frac{1}{Q} \cdot \frac{\tilde{\varepsilon}}{2Q} \cdot \frac{7}{10} \cdot \left(\frac{1}{q_P} - 2^{-\lambda}\right) = \frac{1}{\text{poly}(\lambda)}.$$

This leaves us with the proof of the claim, which is based on a pigeonhole argument. For each $i \in [1, r]$, let M_i with $|M_i| = k$ be the set of $c \in \mathcal{C}$ such that $p_i(c') \leq p_i(c)$ for all $c' \notin M_i$ and all $c \in M_i$. Further, let B be the set of $(\boldsymbol{x}_j, \phi_{j_+}) \in \mathcal{C}^r \times \Phi_{j_+}$ for $\boldsymbol{x}_j = (x_{j,1}, \ldots, x_{j,r})$ such that $x_{j,i} \in M_i$ for all $i \in [1, r]$. Since $|M_i| = k$,

$$\Pr[(\boldsymbol{x}_j, \phi_{j_+}) \in B] \leq \Pr[x_{j,i} \in M_i \quad \forall i \in [1, r]] \leq (k/|\mathcal{C}|)^r .$$

For each $(\boldsymbol{x}_j, \phi_{j_+}) \in S \setminus B$, there exists $i \in [1, r]$ and $c \in \mathcal{C} \setminus M_i$ such that $x_{j,i} = c$. This implies that

$$\begin{aligned}\sum_{i=1}^{r} \sum_{c \in \mathcal{C} \setminus M_i} p_i(c) \geq \Pr[(\boldsymbol{x}_j, \phi_{j_+}) \in S \setminus B] &\geq \Pr[(\boldsymbol{x}_j, \phi_{j_+}) \in S] - \Pr[(\boldsymbol{x}_j, \phi_{j_+}) \in B] \\ &\geq \varepsilon' - (k/|\mathcal{C}|)^r .\end{aligned}$$

From here, we can deduce that there exists $i^* \in [1, r]$ and $c^* \in \mathcal{C} \setminus M_i$ such that $p_{i^*}(c^*) \geq \frac{\varepsilon' - (k/|\mathcal{C}|)^r}{(|\mathcal{C}| - k) \cdot r}$. Hence, for all $c \in \mathcal{G} := M_{i^*} \cup \{c^*\}$, $p_{i^*}(c) \geq \frac{\varepsilon' - (k/|\mathcal{C}|)^r}{(|\mathcal{C}| - k) \cdot r}$, proving the claim. □

6.3 Parameter setting

First of all, by Lemma 2 (and Assumption 1), we have the following condition $2\mathcal{B} \geq q^{n/m} 2^{2\lambda/(md)}$, which is equivalent to

$$\log(2\mathcal{B}) \geq \frac{n \log q}{m} + \frac{2\lambda}{md}. \tag{8}$$

Assumption 1 also requires that

$$q > \max\left\{2\left(120d\sqrt{kr(\beta - 1)}\right)^2, 8\left(144\mathcal{B}\sqrt{3mdr}\right)^2\right\}. \tag{9}$$

Recalling the extracted witness norm bound from Section 6.2, to make Module-SIS secure against lattice attacks (see Section 2.1), we ensure the following holds

$$\min\left\{q,\ 2^{2\sqrt{n \cdot d \cdot \log q \log \delta}}\right\} > 24\sqrt{3r} \cdot m\mathcal{B} \cdot (k + 1) \cdot d^{k+1}. \tag{10}$$

To summarise, we make sure that the inequalities (8), (9) and (10) are satisfied when setting parameters. Table 3 shows several instances with respect to different ring sizes where we restrict $\log q \leq 64$. Note that since r is rounded up, the security parameter λ may be slightly larger than 100. We refer to Appendix C for more discussion on parameter setting, and a clarification for (10).

References

[1] C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More efficient commitments from structured lattice assumptions. In *Security and Cryptography for Networks*, 2018.

Table 3: Parameters and sizes of our lattice-based ring signature scheme for a root Hermite factor $\delta = 1.007$. The signature sizes are rounded to the nearest integer.

$\log N$	6	8	10	12	16	20	30
n	8	16	12	12	23	42	45
m	54	108	75	75	136	262	334
d	64	32	64	64	32	16	16
$\log q$	60	54	59	59	56	53	63
β	64	256	32	64	256	1024	1024
k	1	1	2	2	2	2	3
$\log(2\mathcal{B})$	8.95	8.06	9.48	9.48	9.52	8.54	8.53
r	17	20	17	17	20	25	30
λ	102.0	100.0	102.0	102.0	100.0	100.0	102.5
Signature Size (KB)	930	1132	1409	1492	1814	2604	4511
User PK Size (KB)	3.75	3.38	5.53	5.53	5.03	4.35	5.54
User SK Size (KB)	3.77	3.40	5.56	5.56	5.06	4.37	5.56

[2] C. Baum, H. Lin, and S. Oechsner. Towards practical lattice-based one-time linkable ring signatures. Cryptology ePrint Archive, Report 2018/107, 2018. https://eprint.iacr.org/2018/107.

[3] C. Baum and V. Lyubashevsky. Simple amortized proofs of shortness for linear relations over polynomial rings. Cryptology ePrint Archive, Report 2017/759, 2017. https://eprint.iacr.org/2017/759.

[4] A. Bender, J. Katz, and R. Morselli. Ring signatures: Stronger definitions, and constructions without random oracles. *Journal of Cryptology*, 22(1):114–138, 2009.

[5] 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*, pages 551–572. Springer, 2014.

[6] F. Benhamouda, S. Krenn, V. Lyubashevsky, and K. Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In *ESORICS*, pages 305–325. Springer, 2015.

[7] J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, J. Groth, and C. Petit. Short accountable ring signatures based on DDH. In *European Symposium on Research in Computer Security*, pages 243–265. Springer, 2015.

[8] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. Crystals – Kyber: A cca-secure module-lattice-based kem. In *EuroS&P*, pages 353–367, April 2018.

[9] Z. Brakerski and Y. T. Kalai. A framework for efficient signatures, ring signatures and identity based encryption in the standard model. *IACR Cryptology ePrint Archive*, 2010:86, 2010.

[10] E. Brickell, D. Pointcheval, S. Vaudenay, and M. Yung. Design validations for discrete logarithm based signature schemes. In *PKC*, pages 276–292. Springer, 2000.

[11] R. Del Pino, V. Lyubashevsky, G. Neven, and G. Seiler. Practical quantum-safe voting from lattices. In *CCS*, pages 1565–1581. ACM, 2017.

[12] L. Ducas, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. Crystals – Dilithium: Digital signatures from module lattices. Cryptology ePrint Archive, Report 2017/633, 2017. https://eprint.iacr.org/2017/633.

[13] J. Groth and M. Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In *EUROCRYPT (2)*, volume 9057, pages 253–280. Springer Berlin Heidelberg, 2015.

[14] A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *ASIACRYPT*, volume 5350, pages 372–389. Springer, 2008.

[15] A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, 2015.

[16] B. Libert, S. Ling, F. Mouhartem, K. Nguyen, and H. Wang. Signature schemes with efficient protocols and dynamic group signatures from lattice assumptions. In *ASIACRYPT 2016*, pages 373–403. Springer, 2016.

[17] B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based accumulators: logarithmic-size ring signatures and group signatures without trapdoors. In *EUROCRYPT*, pages 1–31. Springer, 2016.

[18] S. Ling, K. Nguyen, H. Wang, and Y. Xu. Lattice-based group signatures: Achieving full dynamicity with ease. In *ACNS*, pages 293–312. Springer, 2017.

[19] V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, pages 738–755. Springer, 2012. (Full version at https://eprint.iacr.org/2011/537).

[20] V. Lyubashevsky and G. Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In *EUROCRYPT*, pages 204–224. Springer, 2018.

[21] C. A. Melchor, S. Bettaieb, X. Boyen, L. Fousse, and P. Gaborit. Adapting Lyubashevsky's signature schemes to the ring signature setting. In *AFRICACRYPT*, pages 1–25, 2013.

[22] D. Micciancio and O. Regev. Lattice-based cryptography. In *Post-quantum cryptography*, pages 147–191. Springer, 2009.

[23] D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of cryptology*, 13(3):361–396, 2000.

[24] R. Rivest, A. Shamir, and Y. Tauman. How to leak a secret. *Advances in Cryptology ASIACRYPT 2001*, pages 552–565, 2001.

[25] J. Stern. A new paradigm for public key identification. *IEEE Transactions on Information Theory*, 42(6):1757–1768, 1996.

[26] L. R. Turner. Inverse of the vandermonde matrix with applications. Technical Report NASA-TN-D-3547, Lewis Research Center, NASA, 1966.

[27] R. Yang, M. H. Au, J. Lai, Q. Xu, and Z. Yu. Lattice-based techniques for accountable anonymity: Composition of abstract sterns protocols and weak prf with efficient protocols from lwr. Cryptology ePrint Archive, Report 2017/781, 2017. https://eprint.iacr.org/2017/781.

[28] H. Zhang, F. Zhang, H. Tian, and M. H. Au. Anonymous post-quantum cryptocash. Cryptology ePrint Archive, Report 2017/716, 2017. https://eprint.iacr.org/2017/716 (To appear in FC 2018).

Auxiliary Supporting Material

A A Brief Discussion on the Protocol in [28]

In this section, we list some of the issues in the security proofs given in [28]. In an earlier version of [28], it was claimed without providing an explicit definition of the underlying protocol that the anonymity and unforgeability of their ring signature scheme followed directly from the results of [13], provided that a perfectly hiding and computationally binding commitment scheme is used. Later, the authors revised their claims, and provided an explicit one-out-of-many protocol. However, we see that still not all security issues are addressed properly. Since the issues are related to the details of the protocol used as a building block for the ring signature, we focus on Appendix B of [28] (01-Apr-2018 version on IACR's eprint archive).

The authors first claim that for a prime q and a power-of-two n, $X^n + 1$ is irreducible in $\mathbb{Z}_q[X]$ and $\mathbb{Z}_q[X]/(X^n + 1)$ is a field, which clearly does not hold. Furthermore, the distribution of the prover's responses in their protocol depends on secret witness values (similar to our case). It is not mentioned at all how this issue is tackled as the simulator is unaware of these values and straightforward simulation (without using a technique such as rejection sampling) does not work.

Moreover, the authors also claim that the invertibility of a Vandermonde matrix formed by challenges $x_0, \ldots, x_k$ over R_q follows when x_i's are distinct and invertible in R_q. As we have clearly shown, we need the *differences* of challenges, $(x_i - x_j)$, to be invertible in R_q. The authors do not consider anything about the invertibility of the challenge differences with respect to the challenge space they use. In addition, in the special soundness proof for their one-out-of-many protocol, the authors assume that the accepting transcripts used for witness extraction are well-formed (as it would happen in an honest run of the protocol). No assumption on how the accepting transcripts are generated can be made as they are provided by a (possibly) cheating prover.

B Remaining Proofs of Theorem 2

Proof (Proof of Theorem 2). **Completeness:** By Lemma 8, the prover responds with probability $\frac{1}{M^2}$, and the distributions of $f_{j,i}$'s $(i \neq 0)$ are statistically close to $D^d_{12\sqrt{k}}$ and the distributions of $\boldsymbol{z}, \boldsymbol{z}_b, \boldsymbol{z}_c$ are statistically close to $D^{md}_{12\mathcal{B}\sqrt{3md}}$ since

$$\left\|(x \cdot \delta_{\ell_0,1}, \ldots, x \cdot \delta_{\ell_{k-1},\beta-1})\right\| \leq \sqrt{k}, \quad \text{and} \quad \|(x \cdot \boldsymbol{r}, x \cdot \boldsymbol{r}_b, x \cdot \boldsymbol{r}_c)\| \leq \mathcal{B}\sqrt{3md}.$$

Note that $\sum_{j=0}^{k-1} x^j \boldsymbol{\rho}_j \in D^{md}_{12\mathcal{B}\sqrt{3md}}$. From here the bounds on the norms of each component follow similar to the completeness proof of Theorem 1.

All the remaining but the last verification steps also follow straightforwardly. To prove that the last verification step holds for honestly generated values, we

have, for $c_\ell = \mathrm{Com}_{ck}(\boldsymbol{m}_\ell \ ; \ \boldsymbol{r})$,

$$\begin{aligned}
\sum_{i=0}^{N-1}\left(\prod_{j=0}^{k-1} f_{j,i_j}\right)c_i - \sum_{j=0}^{k-1} E_j x^j &= \sum_{i=0}^{N-1} p_i(x)c_i - \sum_{j=0}^{k-1}\left(\sum_{i=0}^{N-1} p_{i,j}c_i + \mathrm{Com}_{ck}(\mathbf{0} \ ; \ \boldsymbol{\rho}_j)\right)x^j \\
&= \sum_{i=0}^{N-1} p_i(x)c_i - \sum_{j=0}^{k-1}\sum_{i=0}^{N-1} p_{i,j}c_i x^j - \sum_{j=0}^{k-1} x^j \cdot \mathrm{Com}_{ck}(\mathbf{0} \ ; \ \boldsymbol{\rho}_j) \\
&= \sum_{i=0}^{N-1} c_i\left(p_i(x) - \sum_{j=0}^{k-1} p_{i,j}x^j\right) - \sum_{j=0}^{k-1} x^j \cdot \mathrm{Com}_{ck}(\mathbf{0} \ ; \ \boldsymbol{\rho}_j) \\
&= \sum_{i=0}^{N-1} c_i\delta_{\ell,i}x^k - \sum_{j=0}^{k-1} x^j \cdot \mathrm{Com}_{ck}(\mathbf{0} \ ; \ \boldsymbol{\rho}_j) \\
&= x^k \cdot c_\ell - \sum_{j=0}^{k-1} x^j \cdot \mathrm{Com}_{ck}(\mathbf{0} \ ; \ \boldsymbol{\rho}_j) \\
&= \mathrm{Com}_{ck}(x^k \cdot \boldsymbol{m}_\ell \ ; \ x^k \cdot \boldsymbol{r} - \sum_{j=0}^{k-1} x^j \cdot \boldsymbol{\rho}_j) \\
&= \mathrm{Com}_{ck}(x^k \cdot \boldsymbol{m}_\ell \ ; \ \boldsymbol{z}) = \mathrm{Com}_{ck}(\mathbf{0} \ ; \ \boldsymbol{z}) \qquad \text{if } \boldsymbol{m}_\ell = \mathbf{0}.
\end{aligned}$$

SHVZK: Given a challenge x, the simulator outputs $(\mathrm{aCom}(0), x, \perp)$ indicating an abort with probability $1 - \frac{1}{M^2}$. Otherwise, it picks $B, C, E_1, \ldots, E_{k-1} \leftarrow R_q^n$ and $f_{j,i} \leftarrow D^d_{12\sqrt{k}}$ for all $0 \le j \le k-1$ and $1 \le i \le \beta - 1$, and also picks $\boldsymbol{z}, \boldsymbol{z}_b, \boldsymbol{z}_c \leftarrow D^{md}_{12\mathcal{B}\sqrt{3md}}$. Then, it calculates $f_{j,0} = x - \sum_{i=1}^{\beta-1} f_{j,i}$ for all $0 \le j \le k-1$, and computes E_0 so as to ensure that the last verification equation is satisfied. Similarly, it computes A and D so that the corresponding verification equations are satisfied. Then, it calculates $(c_a, d_a) = \mathrm{aCom}(A, B, C, D, \{E_j\}_{j=0}^{k-1})$ and outputs the simulated transcript

$$(c_a, x, (d_a, \{f_{j,i}\}_{i\neq 0}, A, B, C, D, \{E_j\}_{j=0}^{k-1}, \boldsymbol{z}, \boldsymbol{z}_b, \boldsymbol{z}_c)).$$

Note that the largest absolute coefficient in any randomness is at most $144\mathcal{B}\sqrt{3md}$ except with probability at most 2^{-100}. Therefore, by Lemma 2, all of the commitments computed are statistically hiding due to the bounds on q and $2\mathcal{B}$, ensuring all of the sampled randomnesses have enough min-entropy. Hence, if the protocol is not aborted, the real and simulated transcripts are indistinguishable by Lemma 8, the hiding property of the commitment scheme and the fact that E_0 is uniquely determined by the last verification equation given all the other components in both the real proof and the simulation. If an abort occurs, then the indistinguishability is satisfied due to hiding property of aCom and the fact that the probability of having an abort is the same for all x. □

C Further discussion on parameter setting

We first note that the parameters we provide in the paper is only an example setting. The same signature and key lengths can be obtained by different sets

of parameters due to the following observation. If $n \cdot \log q$ is fixed and all the parameters but n and $\log q$ remain the same, the signature and key lengths also remain the same. Therefore, any $(n, \log q)$ pair for a fixed $n \cdot \log q$ and a larger $\log q$ than those provided satisfies the requirements in Section 6.3 and could be used to get the same signature and key lengths. We opt to provide the results with $\log q \leq 64$.

The commitment key size for the set of parameters given in Table 3 can be as large as 18 MB. However, one can generate the commitment key from a small seed similar to [8]. Alternatively, for example, even for the extreme case $N = 2^{30}$, we can set parameters so that the signature size becomes 5232 KB and the commitment key size becomes 5892 KB.

Also it is clear that the running time of **RSign** is affected by the acceptance rate of rejection sampling (which is $1/M^2$ for the parameters provided in Table 3). For computational efficiency, one can increase the standard deviation of the masking randomnesses (i.e., $\boldsymbol{r}_a, \boldsymbol{r}_d, \boldsymbol{\rho}_j$ in Protocol 2) to increase the acceptance rate and thus reduce the signing time. For example, choosing a larger standard deviation (i.e., setting $\sigma = 24T$ in Lemma 8) to increase the acceptance rate in **RSign** to $1/M \approx 1/e$ results in less than a 10% increase in the signature size for all the results given in Table 3 (the expected number of iterations in **RSign** is less than 3 in this case). We again emphasize that this probabilistic behaviour is common to all schemes using rejection sampling (for example, [19, 5, 11, 12, 1]).

In the proof of Theorem 4, we relate the unforgeability of the ring signature scheme to Module-SIS$_{n,m+k\beta,q,\theta}$ problem by Lemma 2 for

$$\theta = 2\sqrt{\left(24\sqrt{3r} \cdot m\mathcal{B} \cdot (k+1) \cdot d^{k+1}\right)^2 + 2^{2k}}.$$

However, we can actually find a slightly better bound for θ as follows. We know that binding collision pair is $((2^k \cdot \mathbf{1}, 2^k \cdot \boldsymbol{r}_t), (\mathbf{0}, \boldsymbol{s}_{t'}))$ where $\mathbf{1} = (1, 0, \ldots, 0)$ and

$$\begin{aligned} \|\boldsymbol{s}_{t'}\| &\leq 24\sqrt{3r} \cdot m\mathcal{B} \cdot (k+1) \cdot d^{k+1}, \\ \|\boldsymbol{r}_t\| &\leq \mathcal{B}\sqrt{md}. \end{aligned}$$

From here, for the solution $(2^k \cdot \boldsymbol{r}_t - \boldsymbol{s}_{t'}, 2^k \cdot \mathbf{1})$ to Module-SIS (recall Lemma 2),

$$\begin{aligned} \left\|(2^k \cdot \boldsymbol{r}_t - \boldsymbol{s}_{t'}, 2^k \cdot \mathbf{1})\right\| &\leq \left\|(2^k \cdot \boldsymbol{r}_t, 2^k \cdot \mathbf{1})\right\| + \|(\boldsymbol{s}_{t'}, \mathbf{0})\| \\ &\leq \sqrt{\left(2^k\mathcal{B}\sqrt{md}\right)^2 + 2^{2k}} + 24\sqrt{3r} \cdot m\mathcal{B} \cdot (k+1) \cdot d^{k+1} \\ &\leq 24\sqrt{3r} \cdot m\mathcal{B} \cdot (k+1) \cdot d^{k+1} + \underbrace{2^{k+1}\mathcal{B}\sqrt{md}}_{=:\zeta} =: \theta' \end{aligned}$$

Therefore, we can use the bound θ' and rely on Module-SIS$_{n,m+k\beta,q,\theta'}$ problem. The additive term ζ is very small in comparison to the first part as $d \geq 7$. We have verified that the same results in the parameter setting are obtained whether ζ is ignored. Therefore, for simplicity, Inequality (10) neglects ζ.

Table 4: Calculation of parameters and sizes for the ring signature in Section 6.2.

	Formula	**Notes**
Security parameter	λ	
Dimension of randomness vector	$m \approx \frac{n \log q}{9}$	Chosen based on experimental results
Soundness error	$\eta = \frac{\max\{2,\, \log_\beta N\}}{2d}$	Recall that $k = \log_\beta N$
Number of protocol repetitions	$r = \lceil -\frac{\lambda}{\log \eta} \rceil$	
Number of commitments	$N_c = 4 + k$	$A, B, C, D, E_0, \ldots, E_{k-1}$
Number of $f_{j,i}$ values	$N_f = k \cdot (\beta - 1)$	$f_{0,1}, \ldots, f_{k-1,\beta-1} \in D_{12\sqrt{kr}}$
Number of randomness	$N_R = 3$	$\boldsymbol{z}, \boldsymbol{z}_b, \boldsymbol{z}_c \in D^{md}_{12\mathcal{B}\sqrt{3mdr}}$
Ring Signature size	$r \cdot [N_c \cdot (n \cdot d \log q)$ $+N_f \cdot d \cdot \log(144\sqrt{kr})$ $+N_R \cdot (md \cdot \log(144\mathcal{B}\sqrt{3mdr}))]$	
Commitment key size	$n(m + k \cdot \beta) \cdot d \log q$	An element in $R_q^{n \times (m + k \cdot \beta)}$
User public key size	$nd \log q$	A commitment in $R_q^{n \times 1}$
User secret key size	$md \cdot \log(2\mathcal{B})$	A randomness vector in a commitment.

Table 5: Parameters and sizes of our lattice-based ring signature scheme for a root Hermite factor $\delta = 1.005$ and $\lambda \approx 128$. The signature sizes are rounded to the nearest integer, and $\log q$ is upper-bounded by 64.

$\log N$	6	8	10	12	16	20	30
n	3	11	5	18	29	58	70
m	22	81	33	108	203	368	450
d	256	64	256	64	32	16	16
$\log q$	64	62	61	57	63	57	61
β	64	256	32	64	256	1024	1024
k	1	1	2	2	2	2	3
$\log(2\mathcal{B})$	8.77	8.47	9.27	9.54	9.04	9.03	9.52
r	16	22	16	22	26	32	38
λ	128.0	132.0	128.0	132.0	130.0	128.0	129.8
Signature Size (KB)	1593	2035	2490	2726	3221	4317	7287
User PK Size (KB)	6.00	5.33	9.53	8.02	7.14	6.46	8.34
User SK Size (KB)	6.03	5.36	9.56	8.05	7.17	6.49	8.37