Lattice-based Zero-Knowledge Proofs: New Techniques for Shorter and Faster Constructions and Applications

Muhammed F. Esgin[1,2], Ron Steinfeld[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,Joseph.Liu}@monash.edu
Dongxi.Liu@data61.csiro.au

Abstract. We devise new techniques for design and analysis of efficient lattice-based zero-knowledge proofs (ZKP). First, we introduce *one-shot* proof techniques for non-linear polynomial relations of degree $k \geq 2$, where the protocol achieves a negligible soundness error in a single execution, and thus performs significantly better in both computation and communication compared to prior protocols requiring multiple repetitions. Such proofs with degree $k \geq 2$ have been crucial ingredients for important privacy-preserving protocols in the discrete logarithm setting, such as Bulletproofs (IEEE S&P '18) and arithmetic circuit arguments (EUROCRYPT '16). In contrast, one-shot proofs in lattice-based cryptography have previously only been shown for the linear case ($k = 1$) and a very specific quadratic case ($k = 2$), which are obtained as a special case of our technique.

Moreover, we introduce two speedup techniques for lattice-based ZKPs: a CRT-packing technique supporting "inter-slot" operations, and "NTT-friendly" tools that permit the use of fully-splitting rings. The former technique comes at almost no cost to the proof length, and the latter one barely increases it, which can be compensated for by tweaking the rejection sampling parameters while still having faster computation overall.

To illustrate the utility of our techniques, we show how to use them to build efficient relaxed proofs for important relations, namely proof of commitment to bits, one-out-of-many proof, range proof and set membership proof. Despite their relaxed nature, we further show how our proof systems can be used as building blocks for advanced cryptographic tools such as ring signatures.

Our ring signature achieves a dramatic improvement in length over all the existing proposals from lattices at the same security level. The computational evaluation also shows that our construction is highly likely to outperform all the relevant works in running times. Being efficient in both aspects, our ring signature is particularly suitable for both small-scale and large-scale applications such as cryptocurrencies and e-voting systems. No trusted setup is required for any of our proposals.

Keywords: lattice-based cryptography, zero-knowledge proof, CRT packing, ring signature, one-out-of-many proof, range proof, set membership proof

1 Introduction

Zero-knowledge proofs (ZKP) are fundamental building blocks used in many privacy-preserving applications such as anonymous cryptocurrencies and anonymous credentials [15], and the underlying advanced cryptographic primitives such as ring signatures [35]. They enable a prover to convince a verifier that a certain statement regarding a secret is true with minimal secret information leakage. A core property of ZKPs is *soundness*, that is, a cheating prover should not be able to create a convincing "proof". In the context of proofs of knowledge (PoK), this means successful provers know a relevant secret (i.e., a *witness*), and this is usually proven by using an *extractor* that efficiently recovers the witness given two accepting protocol transcripts with the same initial message. We call this procedure *"basic" witness extraction* (also known as "2-special soundness", see Definition 3). A natural behaviour that is trivially observed in discrete logarithm (DL) based ZKPs is that they achieve a convincing soundness level (i.e., a negligible *soundness error*) in a single protocol run (i.e., they are *one-shot*). However, this natural behaviour turns out to be unexpectedly hard to achieve in lattice-based proofs. There are some works [28, 29, 10, 6, 30] that address this problem in lattice-based cryptography and provide one-shot proofs in the context of protocols that work with "basic" witness extraction. On the other hand, recent research in the DL setting [22, 11, 12, 14] has shown that it is possible to construct more efficient proofs that *require* a *"complex"* witness extraction involving more than two accepting protocol transcripts (and thus more than two challenges) for recovering prover's secret (i.e., the protocols are *many*-special sound). Such proofs rely on higher degree relations to obtain compact results, unlike the 2-special sound proofs that can only check linear (first degree) relations (we refer to the aforementioned works for the motivation behind proving high-degree relations). Again, in the DL setting, these proofs work smoothly and are easily one-shot. However, in the lattice setting, the situation is much more complicated, and, to the best of our knowledge, there is no one-shot witness extraction technique for non-linear relations.

1.1 Related work – Lattice-based zero-knowledge proofs

In being one-shot proofs, the most relevant works for our zero-knowledge proofs are [10] and [6], where the protocols explicitly make use of lattice-based commitments. In fact, the ideas date back to the works by Lyubashevsky [28, 29] introducing the "Fiat-Shamir with Aborts" technique in lattice-based cryptography. The advantage of these works is that the (underlying) protocols achieve a negligible soundness error in a single run, which makes them very efficient in practice. However, all these approaches are limited to working with "basic" witness extraction except for a specific multiplicative (second degree) relation in [10]. The multiplicative argument in [10] is to prove that the coefficient of a quadratic term is zero and no explicit witness extraction from this non-linear relation is provided (and, indeed, no witness extraction from this second degree relation is needed as witnesses are extracted from the linear relations). All these one-shot proofs introduce new complications (more precisely, *relaxations* in the relation being proved) as we discuss in detail in Section 3. One can get asymptotically efficient lattice-based proofs for arithmetic circuits when the circuit size is large compared to the security parameter λ using the amortization techniques from [5]. However, these techniques do not seem to be helpful in our case as the proved relations do not necessarily require a large circuit.

Another line of research makes use of *multi-shot* proofs that require multiple protocol repetitions to get a negligible soundness error. Stern-like combinatorial protocols [38] and proofs using binary challenges fall into this category, where one needs at least λ protocol repetitions for λ-bit security. Therefore, even though these approaches have a wide range of applications (e.g. logarithmic-sized group and ring signatures as in [26]), they currently seem to fall far behind practical expectations (see Table 1 for the concrete results of [26]).

In the ring $R = \mathbb{Z}[X]/(X^d + 1)$, it is possible to achieve a soundness error of $1/(2d)$ using the *monomial challenges* from [9]. Here the challenges are of the form X^i for some $0 \leq i < 2d$ (i.e., there are $2d$ possible challenges in total), and it is shown in [9] that doubled inverses of challenge differences are short (more precisely, $\left\|2(X^i - X^j)^{-1}\right\| \leq \sqrt{d}$ for $i \neq j$). Still proofs using monomial challenges require at least 10 repetitions for a typical ring dimension $d \leq 2048$. To summarize, for a soundness goal of $2^{-\lambda}$, all the above multi-shot approaches produce proofs of length $\widetilde{O}(\lambda^2)$, as a function of the security parameter λ.

1.2 Asymptotic costs of existing lattice-based ZKP techniques

First, let us assume that one relies on computational hardness assumptions, particularly, Module-SIS (M-SIS) and Module-LWE (M-LWE) for the security of a commitment scheme and let $d_{\text{SIS}}, d_{\text{LWE}}$ be the dimension parameters required for M-SIS and M-LWE security, respectively. It is known that one needs $d_{\text{SIS}} = O(\lambda \frac{\log^2 \beta_{\text{SIS}}}{\log q})$ for λ-bit security based on M-SIS where β_{SIS} is the norm of a valid M-SIS solution (see Appendix F.4 for more). Letting $\beta_{\text{SIS}} = q^{\varepsilon}$ for $0 < \varepsilon \leq 1$, we get $\log \beta_{\text{SIS}} = \varepsilon \log q$ and, for a balanced security,

$$d_{\text{LWE}} \approx d_{\text{SIS}} = O(\lambda \varepsilon^2 \log q). \tag{1}$$

In lattice-based cryptography, the most commonly used commitment schemes for algebraic proofs are Unbounded-Message Commitment (UMC) and Hashed-Message Commitment (HMC) (see Section 2.4). These commitment schemes have different tradeoffs as discussed in Appendix B.3. Let n, m, d, v be the module rank for M-SIS, the randomness vector dimension in a commitment, the polynomial ring dimension and the message vector dimension in a commitment, respectively. The commitment vector is of dimension $n + v$ for UMC and n for HMC, which means the space costs of a commitment are $(n + v)d \log q$ and $nd \log q$ for UMC and HMC, respectively. Letting κ be the number of protocol repetitions, we get the formulae for space costs in Table 2.

The commitment matrix dimensions are $(n + v) \times m$ for UMC and $n \times (m + v)$ for HMC, and both of the commitments are computed as a matrix-vector multiplication.[3] Therefore, we also get the formulae for the time costs as given in Table 2 assuming a degree-d polynomial multiplication can be performed in time $\widetilde{O}(d)$ (more precisely, $O(d \log d)$) using, e.g., FFT-like methods.

Further, we have $d_{\text{LWE}} = (m - n - v)d$ and thus $md > d_{\text{LWE}}$ for UMC, and $d_{\text{SIS}} = nd$ for both HMC and UMC. As a result, using (1), we get

$$md = O(\lambda\varepsilon^2 \log q) \text{ for UMC, and } \quad nd = O(\lambda\varepsilon^2 \log q) \text{ for UMC/HMC.} \qquad (2)$$

Now, suppose that we want to prove a relation that involves commitment to $k = O(\log q)$ messages (for example, to prove knowledge of $m_1, \ldots m_k$ such that $\sum_{i=1}^{k} \alpha_i m_i = 0$ for public values $\alpha_1, \ldots, \alpha_k$). Clearly, if we commit to these messages independently, then the overall cost of both time and space increase by a factor of k. Alternatively, we can pack multiple messages in a commitment by setting $v = k$ and hope that this gives a better performance. If an existing multi-shot technique such as Stern-based proofs, or those using binary or monomial challenges, is used, the number of protocol repetitions κ will be $\widetilde{O}(\lambda)$, and thus we get the asymptotic costs in the "multi-shot" column of Table 2 (using (2)). On the other hand, if one can make the proof one-shot, then we get the complexities in the "one-shot" column of Table 2, where there is a clear saving of $\widetilde{O}(\lambda)$.

1.3 Our contributions

One-shot proof techniques for non-linear polynomial relations via adjugate matrices. We introduce new techniques that provide the first solution to the problem of building efficient *one-shot* lattice-based ZKPs that require a "complex" witness extraction. In particular, we introduce witness extraction from non-linear polynomial relations of degree $k \geq 2$ (i.e., "$(k+1)$-special sound protocols", see Definition 3) while still having a one-shot proof. Our proofs reach a negligible soundness error in a single run of the protocol. In comparison to relevant multi-shot prior works such as [26, 19], we improve the asymptotic computation and communication costs by a factor of $\widetilde{O}(\lambda)$ for the security parameter λ (see Table 2), and also achieve a dramatic practical efficiency improvement in both costs (see Table 1). The previous one-shot ideas [28, 29, 10, 6] are obtained as a special case of our technique (see Section 3.2).

Speedup Technique 1: CRT-packing supporting inter-slot operations. Drawing inspiration from the CRT-packing techniques [36, 21] used in fully homomorphic encryption, we introduce the first CRT-packing technique in lattice-based ZKPs that supports "inter-slot" and a *complete* set of operations. That is, our technique supports operations between messages stored in separate CRT "slots", and gives the ability to commit to/encode multiple messages at once and then "extract" all the messages in a way that permits interoperability among extracted values. In its full potential, it provides an asymptotic improvement of $O(\log q)$ in computation costs of proofs involving $O(\log q)$ messages at no additional cost to the proof length (see Table 2).

[3] Here, we overlook the fact that some parts of the commitment matrix are zero or identity, but this does not change the asymptotic behaviour in Table 2.

Table 1: Size comparison of ring signatures for "post-quantum" 128-bit security with N ring participants (the challenge space size is 2^{256}). Signature lengths are in KB. See Appendix A for more details.

Ring Size (N) :	2	2^3	2^6	2^{12}	2^{21}	Security basis
[26]	23000	52000	94000	179000	306000	SIS
[19]	1000	1200	1600	2400	4100	M-LWE & M-SIS
[18]	236	477	839	1561	2645	LowMC (Sym-key)
[24]	?	?	~ 250	~ 456	?	LowMC (Sym-key)
This Work	**36**	**41**	**58**	**103**	**256**	M-LWE & M-SIS
[39]	> 38	> 124	> 900	61000	$> 2^{24}$	Ring-SIS
[7]	35	83	~ 600	40000	$> 2^{24}$	M-LWE & M-SIS

Speedup Technique 2: "NTT-friendly" tools for fully-splitting rings. An important obstacle to computational efficiency of lattice-based ZKPs is that one often requires invertibility of short elements in a ring. A common solution to meeting this criterion is to choose a modulus q of a special form (such as $q \equiv 5 \bmod 8$) at the cost of disabling the ring $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ to fully-split, and thus preventing the (full) use of fast computational algorithms such as Number Theoretic Transform (NTT). We introduce a new result (Lemma 7) that can be used as an alternative to enforcing invertibility, and show how it can be made used of while still supporting the use of NTT-like algorithms. The only requirement of our lemma is for the modulus q to be sufficiently large, without putting any assumptions on its "shape". One can see from, e.g., [31, Table 2] that full NTT provides a speedup of a factor between 6-8 in comparison to plain Karatsuba multiplication (with no FFT).

Design of shorter and faster lattice-based protocols. Our techniques enable the construction of communication and computation efficient lattice-based analogues of DL-based protocols for important applications, where there was previously no efficient lattice-based solutions known. To illustrate this utility of our techniques, we design an efficient range proof that uses speedup technique 1, and an efficient one-out-of-many proof that uses speedup technique 2, where our one-shot proof technique is also applied in both of the proofs.

Application to advanced cryptographic tools. Despite their relaxed nature, we show that our ZKPs are sufficient for important practical applications. Our one-out-of-many proof is used as a building block for lattice-based ring signatures, and our relaxed aggregated range proof is shown to be sufficient for an application in a form of privacy-preserving linkable anonymous credentials.

In Table 1, we compare our ring signature size results to the other potential post-quantum proposals.[4] Most of these schemes, including ours, are only ana-

[4] A concurrent work [27] has recently been put on ePrint, and it builds a linear-sized (linkable) ring signature. Even though "a less efficient version that is based on standard lattice problems" (in particular, SIS and Inhomogeneous SIS) is described, there are no concrete parameters provided for that scheme. The provided concrete instantiation, of size $1.3N$ KB for N ring members, relies on NTRU assumption and claims 103-bit security against quantum attackers. We restrict our comparison

lyzed in the classical random oracle model (ROM), and all the results provided in Table 1 are those in ROM. [18, 24] are recent proposals from symmetric-key primitives using LowMC cipher [2] and all the rest are lattice-based proposals. As can be seen from the table, we achieve a dramatic improvement in comparison to all these post-quantum solutions. Our scheme even reaches the same performance of the linear-sized proposals (bottom two rows), which are tailored to be efficient for small ring sizes, for the smallest possible ring size $N = 2$.[5]

As detailed in Appendix F.4, our ring signature achieves a signature length quasi-linear in the security parameter λ, and poly-logarithmic in the ring size N. In practice, the signature length is proportional to $\lambda \log^2 \lambda \log^c N$ for some constant $c \approx 1.67$. This improves on the quadratic dependence on λ in [26, 18, 24, 19].[6] In terms of the dependence on $\log N$, our scheme grows slightly faster, however, it still outperforms all these works for N as big as billions and beyond.

We further analyze the computational efficiency of our ring signature in Appendix F.5. The analysis based on reasonable assumptions shows that our construction also greatly improves the practical signing/verification times over the existing ring signature proposals with concrete computational efficiency results. For $N = 1024$, we estimate the signing/verification times of our scheme to be below 30 ms whereas [24] reports 2.8 seconds for both of the running times. Our ring signature as well as its underlying protocols, namely binary proof and one-out-of-many proof, do not require any assumption on the "shape" of the modulus q, and thus permit the use of NTT-like algorithms.

1.4 Our techniques

One-shot witness extraction for non-linear polynomial relations. The main challenge in designing *efficient* lattice-based ZKPs is that the extracted witness is required to be *short* as mandated by computational lattice problems (in particular, *Short* Integer Solution – SIS problem). Traditional witness extraction techniques involve the inverse of challenge differences as a multiplicative factor in extracted witnesses, and such an approach is problematic in lattice-based protocols as these inverse terms need not be short in general. This causes one to either resort to more inefficient techniques such as aforementioned multi-shot proofs or introduce relaxations in the proofs. Our solution falls into the latter.

The target problem reduces to the question of extracting useful information from a system of equations of the form $\boldsymbol{V} \cdot \boldsymbol{c} = \boldsymbol{b}$ where $\boldsymbol{V}$ is a matrix (a Vandermonde matrix in our case) constructed by challenges, $\boldsymbol{c}$ is a vector of commitments with unknown openings and $\boldsymbol{b}$ is a vector of commitments with

in Table 1 to those based on "standard lattice problems". Nevertheless, even the NTRU-based scheme produces longer ring signatures than ours when $N \geq 43$.

[5] Note that $N = 1$ would simply give an *ordinary* signature, and there is no reason for using a ring signature for that purpose.

[6] In [26], the soundness goal of $\lambda^{\omega(1)}$ is used and so the number of protocol repetitions for Stern's framework is taken to be $\omega(\log \lambda)$, which disappears in $\widetilde{O}(\cdot)$ notation. But, we consider a practice-oriented goal for the soundness error of $2^{-\lambda}$, and thus the number of protocol repetitions for Stern-based proofs must be $\Omega(\lambda)$. Also, it is stated in [24] that they have the same asymptotic signature growth with [26].

Table 2: The (minimal) asymptotic time and space complexities of lattice-based protocols involving commitment to $k = O(\log q)$ messages. β_{SIS}: M-SIS solution norm, q: modulus, κ: the number of protocol repetitions, n: module rank for M-SIS, v: message vector dimension in a commitment, d: polynomial ring dimension, m: randomness vector dimension in a commitment. Assume: $\log q < \log^2 \beta_{\text{SIS}}/2$ and degree-d polynomial multiplication costs $\widetilde{O}(d)$. To optimize both costs, one would set $n = v$ in all cases.

	Formula	**Multi-shot**[26, 19] $\kappa = \widetilde{O}(\lambda), v = k$	**One-shot** $\kappa = 1, v = k$	**One-shot + CRT** $\kappa = 1, v = O(1)$
Space UMC	$\kappa(n+v)d\log q$	$\widetilde{O}(\lambda^2 \log^2 \beta_{\text{SIS}})$	$\widetilde{O}(\lambda \log^2 \beta_{\text{SIS}})$	$\widetilde{O}(\lambda \log^2 \beta_{\text{SIS}})$
Time UMC	$\kappa(n+v)md$	$\widetilde{O}(\lambda^2 \log^2 \beta_{\text{SIS}})$	$\widetilde{O}(\lambda \log^2 \beta_{\text{SIS}})$	$\widetilde{O}(\lambda \log^2 \beta_{\text{SIS}}/\log q)$
Space HMC	$\kappa nd \log q$	$\widetilde{O}(\lambda^2 \log^2 \beta_{\text{SIS}})$	$\widetilde{O}(\lambda \log^2 \beta_{\text{SIS}})$	N/A
Time HMC	$\kappa n(m+v)d$	$\widetilde{O}(\lambda^2 \log^2 \beta_{\text{SIS}})$	$\widetilde{O}(\lambda \log^2 \beta_{\text{SIS}})$	N/A

known openings. Our idea is to introduce the use of *adjugate* matrices instead of inverse matrices in the "complex" witness extraction of lattice-based ZKPs. This technique, in one hand, enables us to extract *useful* information about the openings of the commitments in $\boldsymbol{c}$ without the involvement of inverse terms, and on the other hand, is the main cause of relaxations. Here, it is crucial that the relaxed proof proves a *useful* relation, is *sound*, and also *efficient*. These piece together nicely when the use of adjugate matrices is accompanied by a good choice of challenge space, and we provide an analysis of our technique with a family of commonly used challenge spaces. We emphasize that straightforward soundness proofs do not work, and one needs special tools such as those introduced in this work to overcome the complications. Our one-shot proof approach is detailed in Section 3 after introducing necessary preliminaries.

CRT-packing supporting inter-slot operations. Let $R = \mathbb{Z}[X]/(X^d + 1)$ and $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ for a usual choice of power-of-two d. It is known that $X^d + 1$ factors linearly (and thus R_q fully splits) for certain choices of q (e.g., a prime $q \equiv 1 \bmod 2d$) and, in that case, one can use NTT for polynomial multiplication in R_q in time $O(d \log d)$. Assume that we choose such an "NTT-friendly" q. For $1 \leq s \leq d$ where s is a power of two, let $R_q^{(0)}, \ldots, R_q^{(s-1)}$ be the polynomial rings of dimension d/s such that $R_q = R_q^{(0)} \times \cdots \times R_q^{(s-1)}$ and $R_q^{(i)} = \mathbb{Z}_q[X]/(P^{(i)}(X))$ for some polynomial $P^{(i)}(X)$ of degree d/s for all $0 \leq i < s$ (which is obtained by the Chinese Remainder Theorem – CRT). We use these CRT "slots" to store s messages in a single ring element. Thus, if we have k messages in total, we can set the message vector dimension in a commitment as $v = k/s$ (instead of $v = k$ in previous approaches).

This initial part of the CRT-packing idea seems easy, and indeed a possible application of CRT in lattice-based ZKPs is mentioned in [31] to perform parallel proofs where there is no interaction between the messages in different slots. We are, on the other hand, interested in applications such as range proofs requiring "inter-slot" operations between messages in separate CRT slots, and get a *complete* set of operations (see [21] for a discussion in the context of FHE).

First thing to note about the CRT-packing technique is that even if the messages to be stored in CRT slots are short, the resulting element in R_q representing s messages need not be so. This makes the technique inapplicable to HMC, which require short message inputs (at least in the general case). More importantly, there are two crucial hurdles we need to overcome: 1) it is not clear how to enable inter-slot operations and make the ZKP work in this setting, and 2) we need to make the proof one-shot in order not to lose the factor λ gained.

Let us write $m = \langle m_0, \ldots, m_{s-1} \rangle$ where $m \in R_q$ and $m_i \in R_q^{(i)}$ for $0 \leq i < s$ if m maps to $(m_0, \ldots, m_{s-1})$ under the CRT-mapping. In general, to prove knowledge of a message b, the prover in the protocol needs to send some "encoding" of the message as $f = \mathrm{Enc}_x(b) = x \cdot b + \rho$ where x is a challenge and ρ is a random masking value. Clearly, we do not want to send k encodings in R_q as it does not result in any savings. Instead, our idea is to send k/s elements in R_q, each encoding s messages, *in a way* that enables the verifier to "extract" all k messages out of them. When the prover sends $f = x \cdot m + \rho$ (there may be multiple such f's), for each $0 \leq i < s$, the verifier can compute $f_i = f \bmod (q, P^{(i)}(X)) = x_i \cdot m_i + \rho_i$ as the extracted encodings where $x = \langle x_0, \ldots, x_{s-1} \rangle$ and $\rho = \langle \rho_0, \ldots, \rho_{s-1} \rangle$. The main problem here is now that f_i's are encodings of m_i's, but under possibly *different* x_i's, which circumvents interoperability of distinct f_i's. For example, the sum $f_i + f_j$ for $i \neq j$ does not result in an encoding of the sum of messages under a common challenge x if $x_i \neq x_j$.

To overcome this problem, our idea is to choose the challenge $x = \langle x, \ldots, x \rangle$ for $x \in \bigcap_{i=0}^{s-1} R_q^{(i)}$ such that all extracted encodings are under the same challenge x. This means x must be of degree smaller than d/s and thus the challenge space size is possibly greatly decreased.[7] To make the proof one-shot, we choose the challenges to be polynomials of degree at most $d/s - 1$ with coefficients in $\mathbb{Z}_p$ such that $p^{d/s} = 2^{2\lambda}$ (i.e., there are $2^{2\lambda}$ challenges in total).[8] Therefore, we need $d/s \cdot \log p = 2\lambda$, which is satisfied by choosing $d/s = \lambda\varepsilon^2$ and $\log p = 2/\varepsilon^2$. We should also ensure $\log q > \log p = 2/\varepsilon^2 = 2\log^2 q / \log^2 \beta_{\mathrm{SIS}}$. This holds assuming $\log q < \log^2 \beta_{\mathrm{SIS}}/2$, which is easily satisfied in most of the practical applications.

To have fast computation, we also set $d = d_{\mathrm{SIS}} = O(\lambda\varepsilon^2 \log q)$, and hence get $s = O(\log q)$. Recall that we have k messages in total and s slots in a single ring element. As a result, for $k = O(\log q)$, it is enough to have $v = k/s = O(1)$. Overall, we end up with the asymptotic costs in the last column of Table 2, where our technique has a factor $\log q$ saving in asymptotic computational time in comparison to previous approaches *without* any compromise in communication.

An attractive example in practice where one would need a commitment to $k = O(\log q)$ messages is a range proof on $[0, 2^k - 1]$. Let us take a range proof on $\ell \in [0, 2^{64} - 1]$ as a running example. In this case, our proof proceeds as follows. We allow R_q to split into at least 64 factors, and thus use a *single* R_q element to commit to all the bits of ℓ (so committing to all the bits of ℓ only

[7] We remark that earlier works [37, 10] also considered choosing a challenge of degree d/s for some $s > 1$ for the purpose of invertibility of challenges. However, our motivation here is to make sure that x has the same element in all CRT slots.

[8] In this work, we consider a challenge space size of $2^{2\lambda}$ for λ-bit post-quantum security.

Table 3: Comparison of non-interactive range proof sizes (in KB). "Ideal w/o CRT" is a hypothetical scheme optimized for proof length. FFT denotes the maximum number of FFT levels supported. Our proof sizes can be slightly reduced at the cost of reducing the FFT levels. The full parameter setting details are given in Appendix C.

range width (N)	$N = 2^{32}$				$N = 2^{64}$			
# of batched proofs (ψ)	1	5	10	(d, FFT)	1	5	10	(d, FFT)
with "norm-optimal" challenges from [31]	161	745	1484	(256, 1)	443	2131	4274	(256, 2)
Ideal w/o CRT	52	113	180	(32, 5)	86	201	302	(16, 4)
Our Work: CRT-packed	58	130	202	(512, 5)	93	216	319	(512, 6)

cost a single commitment with message vector dimension $v = 1$). In its initial move, the prover sends some commitments and gets a challenge from the verifier. Then, the prover responds with a *single* encoding in R_q (or 64 small encodings that costs as much as a single element in R_q). From here, the verifier extracts the encodings of all the bits, reconstructs the masked integer value ℓ and checks whether it matches the input commitment to ℓ. In this setting, it is clear that we require operability between different slots, and thus set the encodings of all the bits to be under the same challenge x. For a ring dimension $d = 512$, the infinity norm of a challenge can be as large as 2^{31}, which seems quite large.

An alternative to this approach is to use "norm-optimal" challenges from [31] (named "optimal" in [31]) such that the infinity norm of a challenge is set to 1, and thus the overall Euclidean norm of a challenge is minimized. In this case, one needs to set the ring dimension $d \geq 256$ to get a challenge space size of at least 2^{256}. However, this results in significantly longer proofs as shown in Table 3. The reason behind this phenomenon is that one needs to encode 64 values and with the "norm-optimal" challenges the cost of these encodings and the commitments grow too much. The use of challenges with larger (even much larger) norm does not seem to cause significant increase in the proof length, which can be explained as follows. To do a range proof on 64-bit range, the modulus q must be at least 2^{64}. Using UMC, where the message part does not affect the hardness of finding binding collisions (in particular, M-SIS hardness), such a large q already makes the M-SIS very hard and M-LWE very easy. Therefore, having a challenge with a large norm only brings the hardness level of M-SIS to that of M-LWE, and results in a very compact proof.

We also add for comparison a hypothetical idealized range proof scheme optimized for proof length in Table 3, where for this scheme we only check two conditions: (1) $q \geq N$ and (2) M-SIS and M-LWE root Hermite factors are less than or equal to 1.0045. More specifically, we go over all the values of the ring dimension $d \in \{8, 16, \ldots, 1024\}$, $\log q \in \{\log N, \ldots, 100\}$ and initial noise distribution $\mathcal{U}(\{-\mathcal{B}, \ldots, \mathcal{B}\})$ for $\mathcal{B} \in \{1, 2, 3\}$, and set the remaining parameters so that the above security condition (2) is satisfied. Therefore, for the "ideal w/o CRT" scheme we do not check whether the soundness proof of the protocol works with the parameters set. Even with this advantage given, we see from Table 3 that our range proof, as expected, has approximately the same proof length as "ideal w/o CRT", and also achieves a significant speedup as the ring dimension as well as the number of FFT levels supported is higher. One can see from [31, Table 2] that going from 2 levels of FFT to 6 levels of FFT alone results in a speedup of a factor more than 3.

When we allow the ring R_q to split into more than 64 factors, then the 64 subrings in which the message bits are encoded will not be fields and the structure of R_q in these subring is lost. We are currently unable to make the soundness proof of the binary proof go through in these subrings, whose structure is unclear. On the other hand, we can make the binary proof work both in R_q using our new result (Lemma 7) and in any field. Thus, we allow R_q to split into exactly $\log N$ *fields* for a range proof of width N, which also gives the invertibility of challenges and challenge differences at no cost. The reason why the scheme with "norm-optimal" challenges cannot split into more than $2^2 = 4$ factors is because the invertibility of polynomials with coefficients as large as 2^{16} is required when one relies solely on the results of [31].

"NTT-friendly" tools for fully-splitting rings. [31] studies in detail how cyclotomic rings split and the required invertibility conditions for short ring elements. A main motivation in [31] for the invertibility of short elements can be sketched as follows. In the hope of proving knowledge of a secret s (which is usually a message-randomness pair (m, r)) that satisfies a certain relation $g(s) = t$ for public homomorphic function g and public t, one-shot proofs can only convince the verifier of knowledge of $\bar{s}$ such that $g(\bar{s}) = \bar{x}t$, where $\bar{x} = x - x'$ for some (distinct) challenges x, x'. If g is a commitment scheme and one later opens t to a valid s' such that $g(s') = t$, then one can show that $s' = \bar{s}/\bar{x}$ using the binding property of the commitment scheme provided that $\bar{x}$ is invertible. In our protocols, however, the relaxed relation proves knowledge of a secret *message* m such that

$$g'(\bar{x}m) = \bar{x}t'$$

where g' and t' are the parts dependent on the message (see Definitions 4 and 6). When one gets two relaxed openings $(\bar{x}_0, m_0)$ and $(\bar{x}_1, m_1)$, we have

$$\begin{matrix} g'(\bar{x}_0 m_0) = \bar{x}_0 t' \\ g'(\bar{x}_0 m_1) = \bar{x}_1 t' \end{matrix} \implies \begin{matrix} g'(\bar{x}_1 \bar{x}_0 m_0) = \bar{x}_1 \bar{x}_0 t' \\ g'(\bar{x}_0 \bar{x}_1 m_1) = \bar{x}_0 \bar{x}_1 t' \end{matrix} \implies \bar{x}_1 \bar{x}_0 m_0 = \bar{x}_0 \bar{x}_1 m_1, \quad (3)$$

due to the binding property of the commitment scheme. On contrary to the invertibility requirement, if the norm of each term is small relative to q, which is often the case, we use our new result Lemma 7 to show that,

$$\bar{x}_0 \bar{x}_1 (m_0 - m_1) = 0 \text{ in } \mathbb{Z}_q[X]/(X^d + 1) \qquad \implies \qquad m_0 = m_1. \quad (4)$$

That is, we can conclude the equality of two message openings even for non-invertible challenge differences. The lemma only requires q to be sufficiently large without putting any condition on its "shape", and thus enables the use of an "NTT-friendly" modulus q.

Open Problems. Our CRT technique only allows us to gain an improvement in terms of computation. A very interesting result would be to also have an asymptotic/practical advantage in communication costs, which remains as an open problem. Another interesting question is whether one can make the binary proof work while having a fully-splitting R_q. This would allow us to exploit the full potential of our CRT technique in its application to range proofs.

Roadmap. Section 3 is devoted to the introduction of the one-shot proof technique for non-linear polynomial relations. Our CRT-packing technique and other new tools that enable faster proofs are detailed in Section 4, followed by an application to range proofs. We apply our one-shot proof techniques to build efficient ZKPs of useful relations such as one-out-of-many proofs in Section 5. Further applications to advanced cryptographic tools such as ring signatures (detailed in Appendix F) and anonymous credentials (detailed in Appendix G) are discussed under Section 6. Some formal definitions, further discussions and proofs of lemmas/theorems are deferred to appendices due to limited space.

2 Preliminaries

We use standard notations as detailed in Appendix B.1. Additionally, for a vector of polynomials $\boldsymbol{p}$, $\mathsf{HW}(\boldsymbol{p})$ denotes the Hamming weight of the coefficient vector of $\boldsymbol{p}$, and D_σ^r denotes the discrete normal distribution with center zero and standard deviation σ over $\mathbb{Z}^r$. The formal definition and the norm bounds of normal distribution, and relations between different norms are recalled in Appendix B. We summarize the rejection sampling [29], used to make prover's responses independent of secret information, in Algorithm 1 and its statement in Lemma 11 in Appendix B.

Algorithm 1 $\mathrm{Rej}(\boldsymbol{z}, \boldsymbol{c}, \phi, T)$

1: $\sigma = \phi T$; $\mu(\phi) = e^{12/\phi + 1/(2\phi^2)}$; $u \leftarrow [0, 1)$
2: **if** $u > (\frac{1}{\mu(\phi)}) \cdot \exp\left(\frac{-2\langle \boldsymbol{z}, \mathbf{c}\rangle + \|\mathbf{c}\|^2}{2\sigma^2}\right)$ **then return** 0 ▷ means abort in the protocols.
3: **else return** 1

2.1 Vandermonde matrices and some basics of Linear Algebra

We recall some basics about Vandermonde matrices and from Linear Algebra relevant to our discussions (see e.g. [23] for more details). We denote the n-dimensional identity matrix by $\boldsymbol{I}_n$, and assume that the matrices are defined over a ring $\mathfrak{R}$. Let $\boldsymbol{A}$ be a $n \times n$ square matrix and $\det(\boldsymbol{A})$ denote its determinant. The adjugate $\mathrm{adj}(\boldsymbol{A})$ of $\boldsymbol{A}$, defined as the transpose of the cofactor matrix of $\boldsymbol{A}$, satisfies the following property

$$\mathrm{adj}(\boldsymbol{A}) \cdot \boldsymbol{A} = \boldsymbol{A} \cdot \mathrm{adj}(\boldsymbol{A}) = \det(\boldsymbol{A}) \cdot \boldsymbol{I}_n. \tag{5}$$

Therefore, if $\boldsymbol{A}$ is non-singular, $\mathrm{adj}(\boldsymbol{A}) = \det(\boldsymbol{A}) \cdot \boldsymbol{A}^{-1}$. A $(k+1)$-dimensional Vandermonde matrix $\boldsymbol{V}$ is defined as below for some $x_0, \ldots, x_k \in \mathfrak{R}$, with its determinant satisfying the following property

$$\boldsymbol{V} = \begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^k \\ 1 & x_1 & x_1^2 & \cdots & x_1^k \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_k & x_k^2 & \cdots & x_k^k \end{pmatrix}, \qquad \text{and} \qquad \det(\boldsymbol{V}) = \prod_{0 \le i < j \le k} (x_j - x_i). \tag{6}$$

The following is an easy consequence of (6).

Fact 1 *The Vandermonde determinant* $\det(\boldsymbol{V})$ *has* $\binom{k+1}{2}$ *multiplicands of the form* $x_j - x_i$ *with* $j \neq i$.

As given in [19], the Vandermonde matrix inverse $\boldsymbol{V}^{-1}$, when it exists, has the following structure

$$\begin{pmatrix} \frac{*}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_k)} & \frac{*}{(x_0-x_1)(x_1-x_2)\cdots(x_1-x_k)} & \cdots & \frac{*}{(x_0-x_k)(x_1-x_k)\cdots(x_{k-1}-x_k)} \\ \frac{*}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_k)} & \frac{*}{(x_0-x_1)(x_1-x_2)\cdots(x_1-x_k)} & \cdots & \frac{*}{(x_0-x_k)(x_1-x_k)\cdots(x_{k-1}-x_k)} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_k)} & \frac{-1}{(x_0-x_1)(x_1-x_2)\cdots(x_1-x_k)} & \cdots & \frac{(-1)^k}{(x_0-x_k)(x_1-x_k)\cdots(x_{k-1}-x_k)} \end{pmatrix}, \quad (7)$$

where $*$ denotes some element in the ring $\mathfrak{R}$, computed as a function of x_i's. It is clear from this structure that $\boldsymbol{V}^{-1}$ exists over $\mathfrak{R}$ if and only if the differences $x_i - x_j$ for $0 \leq i < j \leq k$ are invertible over $\mathfrak{R}$. The structure in (7) helps us to visualize the structure of $\text{adj}(\boldsymbol{V})$ using the fact that $\text{adj}(\boldsymbol{V}) = \det(\boldsymbol{V}) \cdot \boldsymbol{V}^{-1}$ if $\boldsymbol{V}$ is non-singular. In particular, we have the following fact.

Fact 2 *Let* $(\Gamma_0, \ldots, \Gamma_k)$ *be the last row of* $\text{adj}(\boldsymbol{V})$. *Then,*

$$\Gamma_i = (-1)^{i+k} \prod_{0 \leq l < j \leq k \,\wedge\, j,l \neq i} (x_j - x_l),$$

and Γ_i *has* $\left[\binom{k+1}{2} - k\right] = \frac{k(k-1)}{2}$ *multiplicands for all* $0 \leq i \leq k$.

Fact 2 follows by observing that k multiplicands in $\det(\boldsymbol{V})$ are cancelled out by the corresponding denominator in $\boldsymbol{V}^{-1}$.

2.2 Module-SIS and Module-LWE problems

Our schemes' security relies on the hardness of Module-SIS (M-SIS) (defined in "Hermite normal form" as in [6]) and Module-LWE (M-LWE) problems [25].

Definition 1 (M-SIS$_{n,m,q,\beta_{\text{SIS}}}$). *Given* $\boldsymbol{A} = [\, \boldsymbol{I}_n \,\|\, \boldsymbol{A}' \,]$ *with* $\boldsymbol{A}' \leftarrow \mathcal{U}(R_q^{n\times(m-n)})$, *the goal is to find* $\boldsymbol{z} \in R_q^m$ *such that* $\boldsymbol{A}\boldsymbol{z} = \boldsymbol{0} \bmod q$ *and* $0 < \|\boldsymbol{z}\| \leq \beta_{\text{SIS}}$.

Definition 2 (M-LWE$_{n,m,q,\chi}$). *Let* χ *be a distribution over* R_q *and* $\boldsymbol{s} \leftarrow \chi^n$ *be a secret key. Define* $\texttt{LWE}_{q,\boldsymbol{s}}$ *as the distribution obtained by sampling* $\boldsymbol{a} \leftarrow R_q^n$, $e \leftarrow \chi$ *and outputting* $(\boldsymbol{a}, \langle \boldsymbol{a}, \boldsymbol{s} \rangle + e)$. *The goal is to distinguish between* m *given samples from either* $\texttt{LWE}_{q,\boldsymbol{s}}$ *or* $\mathcal{U}(R_q^n, R_q)$.

The above definition is a standard variant of decision M-LWE problem where the secret is sampled from the error distribution. More discussion about the security aspects is given in Appendix F.3 when we instantiate our ring signature with concrete parameters.

2.3 Σ-protocols

Σ-protocols are a type of interactive proof systems between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. It is 3-move as in Protocol 1. A protocol transcript is *accepting* if it is accepted by the verifier. Σ-protocols are defined for a relation $\mathcal{R}$, and for a $(v, w) \in \mathcal{R}$, the quantity w is said to be a witness for v. We use the generalized definition of Σ-protocols from [19] that extends the one in [9].

Definition 3 ([19, Definition 4]). *For relations $\mathcal{R}, \mathcal{R}'$ with $\mathcal{R} \subseteq \mathcal{R}'$, $(\mathcal{P}, \mathcal{V})$ is called a Σ-protocol for $\mathcal{R}, \mathcal{R}'$ with completeness error α, a challenge space $\mathcal{C}$, public-private inputs (v, w), if the following properties are satisfied.*

- ***Completeness:*** *An interaction between an honest prover and an honest verifier is accepted with probability at least $1 - \alpha$ whenever $(v, w) \in \mathcal{R}$.*
- $(k+1)$***-special soundness:*** *There exists an efficient PPT* extractor $\mathcal{E}$ *that computes w' satisfying $(v, w') \in \mathcal{R}'$ given $(k+1)$ accepting protocol transcripts $(a, x_0, z_0), \ldots, (a, x_k, z_k)$ with distinct x_i's for $0 \leq i \leq k$. We refer to this process as* witness extraction.
- ***Special honest-verifier zero-knowledge (SHVZK):*** *There exists an efficient PPT* simulator $\mathcal{S}$ *that outputs (a, z) given v in the language of $\mathcal{R}$ and $x \in \mathcal{C}$ such that (a, x, z) is indistinguishable from an accepting transcript produced by a real run of the protocol.*

As seen from above, the special soundness is *relaxed* in the sense the verifier is only convinced of the proof of knowledge of a witness for the relation $\mathcal{R}'$. This is usually referred to as the *soundness gap*. This relaxation is necessary for efficient algebraic proofs and such relaxed proofs are sufficient for our applications.

2.4 Commitment schemes

We define the commitment schemes UMC (Unbounded-Message Commitment) [10, 6] and HMC (Hashed-Message Commitment) (see, e.g., [19, 6]). Both hiding and binding properties are computational (see Appendix B.3 for formal definitions of commitments, their properties and more discussion). Let $n, m, \mathcal{B}, q$ be positive integers, and assume that we commit to v-dimensional vectors over R_q for $v \geq 1$. As in [10, 6], the opening algorithm Open is *relaxed* in the sense that there is an additional input $y \in R_q$, called *relaxation factor*, to Open algorithm along with a message-randomness pair $(\boldsymbol{m}', \boldsymbol{r}')$ such that Open checks if $y \cdot C = \mathrm{Com}_{ck}(\boldsymbol{m}'; \boldsymbol{r}')$. The instantiation of HMC with $m > n$ is as follows.

- $\mathsf{CKeygen}(1^\lambda)$: Pick $\boldsymbol{G}_r' \leftarrow R_q^{n \times (m-n)}$ and $\boldsymbol{G}_m \leftarrow R_q^{n \times v}$. Output $ck = \boldsymbol{G} = [\,\boldsymbol{G}_r \,\|\, \boldsymbol{G}_m\,] \in R_q^{n \times (m+v)}$ where $\boldsymbol{G}_r = [\,\boldsymbol{I}_n \,\|\, \boldsymbol{G}_r'\,]$. We assume that Commit and Open takes ck as an input implicitly.
- $\mathsf{Commit}(\boldsymbol{m})$: Pick $\boldsymbol{r} \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$. Output

$$\mathrm{Com}_{ck}(\boldsymbol{m}; \boldsymbol{r}) = \boldsymbol{G} \cdot (\boldsymbol{r}, \boldsymbol{m}) = \boldsymbol{G}_r \cdot \boldsymbol{r} + \boldsymbol{G}_m \cdot \boldsymbol{m}.$$

- $\mathsf{Open}(C, (y, \boldsymbol{m}', \boldsymbol{r}'))$: If $\mathrm{Com}_{ck}(\boldsymbol{m}'; \boldsymbol{r}') = yC$ and $\|(\boldsymbol{r}', \boldsymbol{m}')\| \leq \gamma_{\mathrm{com}}$, return 1. Otherwise, return 0.

Lemma 1. *If M-LWE$_{m-n,n,q,\mathcal{U}(\{-\mathcal{B},\ldots,\mathcal{B}\}^d)}$ problem is hard, then HMC is computationally hiding. If M-SIS$_{n,m+v,q,2\gamma_{\mathrm{com}}}$ is hard, then HMC is computationally strong γ_{com}-binding with respect to the same relaxation factor y.*

The instantiation of UMC is also similar and defined as below for $m > n + v$.

- $\mathsf{CKeygen}(1^\lambda)$: Pick $\boldsymbol{G}_1' \leftarrow R_q^{n \times (m-n)}$ and $\boldsymbol{G}_2' \leftarrow R_q^{v \times (m-n-v)}$. Set $\boldsymbol{G}_1 = [\,\boldsymbol{I}_n \,\|\, \boldsymbol{G}_1'\,]$ and $\boldsymbol{G}_2 = [\,\boldsymbol{0}^{v \times n} || \boldsymbol{I}_v \,\|\, \boldsymbol{G}_2'\,]$. Output $ck = \boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_1 \\ \boldsymbol{G}_2 \end{bmatrix} \in R_q^{(n+v) \times m}$. We assume that Commit and Open takes ck as an input implicitly.

- $\mathsf{Commit}(\boldsymbol{m})$: Pick $\boldsymbol{r} \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$. Output

$$\mathrm{Com}_{ck}(\boldsymbol{m};\, \boldsymbol{r}) = \boldsymbol{G} \cdot \boldsymbol{r} + (\boldsymbol{0}^n, \boldsymbol{m}).$$

- $\mathsf{Open}(C, (y, \boldsymbol{m}', \boldsymbol{r}'))$: If $\mathrm{Com}_{ck}(\boldsymbol{m}';\, \boldsymbol{r}') = yC$ and $\|\boldsymbol{r}'\| \leq \gamma_{\mathrm{com}}$, return 1. Otherwise, return 0.

Observe from the above definition that only the norm of $\boldsymbol{r}'$ is checked in the Open algorithm of UMC whereas that of $(\boldsymbol{m}', \boldsymbol{r}')$ is checked in HMC. Also, our definition of Open for UMC is slightly different than that in [6] because we do not multiply the relaxation factor with the message as the invertibility of the relaxation factor is not guaranteed in our case.

Lemma 2 ([6]). *If M-LWE$_{m-n-v,n+v,q,\mathcal{U}(\{-\mathcal{B},\ldots,\mathcal{B}\}^d)}$ problem is hard, then UMC is computationally hiding. If M-SIS$_{n,m,q,2\gamma_{\mathrm{com}}}$ is hard, then UMC is computationally γ_{com}-binding with respect to the same relaxation factor y.*

We use the same notation for both of the commitment schemes and will clarify in the relevant sections which specific instantiation is used. We say that $(y, \boldsymbol{m}', \boldsymbol{r}')$ is a *valid* opening of C if $\mathsf{Open}(C, (y, \boldsymbol{m}', \boldsymbol{r}')) = 1$. A valid opening $(y, \boldsymbol{m}', \boldsymbol{r}')$ with $y = 1$ is called an *exact valid* opening. We call the message part $\boldsymbol{m}'$ of an opening as *message opening*, and if $(y, \boldsymbol{m}', \boldsymbol{r}')$ is a valid opening such that $yC = \mathrm{Com}_{ck}(y\boldsymbol{m}';\, \boldsymbol{r}')$, then we call $\boldsymbol{m}'$ a *relaxed message opening* with relaxation factor y. It is also straightforward that both UMC and HMC satisfy the following homomorphic properties: $\mathrm{Com}_{ck}(\boldsymbol{m}_0;\, \boldsymbol{r}_0) + \mathrm{Com}_{ck}(\boldsymbol{m}_1;\, \boldsymbol{r}_1) = \mathrm{Com}_{ck}(\boldsymbol{m}_0 + \boldsymbol{m}_1;\, \boldsymbol{r}_0 + \boldsymbol{r}_1)$ and $c \cdot \mathrm{Com}_{ck}(\boldsymbol{m};\, \boldsymbol{r}) = \mathrm{Com}_{ck}(c \cdot \boldsymbol{m};\, c \cdot \boldsymbol{r})$ for $c \in R_q$.

3 One-Shot Proofs for Non-Linear Polynomial Relations

In this section, we focus on lattice-based zero-knowledge proofs in a general framework using homomorphic commitments, and introduce our techniques to get efficient proofs. Even though such a setting is also mostly shared with DL-based Σ-protocols using homomorphic commitments, the main challenges described here are not encountered in those cases. Since our main concern is about the soundness of the protocol, in this section, we omit the discussion about the zero-knowledge property, which is later obtained using a standard rejection sampling technique. We always consider homomorphic commitments when referring to "commitment" and assume that all the elements are in a ring $\mathfrak{R}$.

3.1 The case for linear relations (2-special soundness)

If we investigate the (underlying) one-shot Σ-protocols from [28, 29, 10, 6], we see the following. The common input of the protocol is a commitment C_1 to the prover's witness and the prover sends an initial commitment C_0.[9] Then, the verifier sends a random challenge $x \leftarrow \mathcal{C}$, which is responded by the prover as $(\boldsymbol{f}, \boldsymbol{z})$, and $(\boldsymbol{f}, \boldsymbol{z})$ is used by the verifier as a message-randomness pair for a

[9] The reason behind indexing becomes clear in what follows.

commitment computation.[10] More precisely, the verification checks if $C_0 + xC_1 = \mathrm{Com}_{ck}(\boldsymbol{f}; \boldsymbol{z})$ holds and $\boldsymbol{f}, \boldsymbol{z}$ have small norm. This is equivalent to the structure represented in Protocol 1 for $k = 1$. From here, when the extractor gets two valid protocol transcripts $(C_0, x_0, \boldsymbol{f}_0, \boldsymbol{z}_0), (C_0, x_1, \boldsymbol{f}_1, \boldsymbol{z}_1)$ using the same initial message C_0, and different challenges x_0 and x_1, the extractor obtains

$$\begin{matrix} C_0 + x_0 C_1 = \mathrm{Com}_{ck}(\boldsymbol{f}_0; \boldsymbol{z}_0) \\ C_0 + x_1 C_1 = \mathrm{Com}_{ck}(\boldsymbol{f}_1; \boldsymbol{z}_1) \end{matrix} \implies (x_1 - x_0)C_1 = \mathrm{Com}_{ck}(\boldsymbol{f}_1 - \boldsymbol{f}_0; \boldsymbol{z}_1 - \boldsymbol{z}_0). \quad (8)$$

At this stage, it is not possible to obtain a *valid exact* opening of C_1 unless $(x_1 - x_0)^{-1}$ is guaranteed to be short due to the shortness requirements of valid openings for lattice-based commitment schemes.[11] Unless ensured by design, there is no particular reason why the inverse term $(x_1 - x_0)^{-1}$ would be short. In the current state of affairs, the largest set of challenges with short challenge difference inverses is monomial challenges [9] used with ring variants of lattice assumptions. Here, only $2(x_1 - x_0)^{-1}$ is guaranteed to be short and thus the extractor can only get the openings of $2C_1$. As discussed previously, for a ring dimension of d, the cardinality of the monomial challenge space is only $2d$, which is typically smaller than 2^{12} in practice. This small challenge space problem causes major efficiency drawbacks in terms of both computation and communication as the protocol is required to be repeated many times to get a negligible soundness error (that is, the same computation and communication steps are repeated multiple times, resulting in a multi-fold increase in both computation and communication). The situation is even worse in terms of the number of repetitions when binary challenges or Stern's framework [38] is used where the protocol is required to be repeated at least λ times for λ-bit security.

The idea for a one-shot proof is to make use of (8) without any inverse computation by observing that $(\boldsymbol{f}_1 - \boldsymbol{f}_0, \boldsymbol{z}_1 - \boldsymbol{z}_0)$ is a valid opening of $(x_1 - x_0)C_1$ as long as $\boldsymbol{f}_1 - \boldsymbol{f}_0$ and $\boldsymbol{z}_1 - \boldsymbol{z}_0$ are short, which is ensured by norm checks on $\boldsymbol{f}, \boldsymbol{z}$ in each verification. If one can prove that having this *relaxed* case is sufficient and also violates the binding property of the commitment (i.e., that it allows one to solve a computationally hard problem), then the soundness of the protocol is achieved (with a relaxed relation $\mathcal{R}'$ as in Definition 3) with no challenge difference inverses involved. This eliminates the need for challenge differences to have short inverses and enables one to use exponentially large challenge spaces, resulting in *one-shot* proofs. The main technical difficulty here is handling soundness gap, where the extractor only obtains an exact opening of $(x_1 - x_0)C_1$ (rather than C_1, which is the commitment to the prover's witness).

3.2 Generalization to degree $k > 1$ ($(k+1)$-special soundness)

As can be seen from (8), the 2-special sound case is quite restrictive as it only allows witness extraction from linear (first degree) relations. On the other hand,

[10] In certain proofs, the use of UMC allows the prover to respond only with the randomness part $\boldsymbol{z}$. In such a case, $\boldsymbol{f}$ need not be transmitted and can be assumed to be set appropriately by the verifier.

[11] Recall that UMC allows an unbounded *message* opening, but still the randomness is required to be short.

$\mathcal{P}((ck, C_0, \ldots, C_k), (\boldsymbol{m}_k, \boldsymbol{r}_k))$		$\mathcal{V}(ck, C_0, \ldots, C_k)$
	$\overset{x}{\longleftarrow}$	$x \leftarrow \mathcal{C}$
$\boldsymbol{f}, \boldsymbol{z} \leftarrow F(\cdots)$	$\overset{\boldsymbol{f}, \boldsymbol{z}}{\longrightarrow}$	
		$\\|\boldsymbol{f}\\| \overset{?}{\leq} \mathcal{T}_f,\ \\|\boldsymbol{z}\\| \overset{?}{\leq} \mathcal{T}_z$
		$C_0 + xC_1 + \cdots + x^k C_k \overset{?}{=} \mathrm{Com}_{ck}(\boldsymbol{f};\, \boldsymbol{z})$

Protocol 1: Structure of a $(k+1)$-special sound Σ-protocol. $\mathcal{T}_f, \mathcal{T}_z \in \mathbb{R}^+$ are some pre-determined values that vary among different proofs.

the ability to work with non-linear relations is a must in recent efficient proofs [22, 11, 12, 14], which renders the existing lattice-based one-shot techniques inapplicable. Therefore, we generalize our setting, and suppose that we have a degree-k polynomial relation ($(k+1)$-special sound Σ-protocol), $k \geq 1$, with the structure given in Protocol 1. Note that since the extractor only knows that verification steps hold, unaware of how any component is generated, other steps but those in the verification is not important. Therefore, we write all the C_i's as a common input whereas in the actual protocol a subset of them can be generated during a protocol run. The commitment to the prover's witness $(\boldsymbol{m}_k, \boldsymbol{r}_k)$ is C_k.

The witness extraction, in this case, works by the extractor obtaining $k+1$ accepting protocol transcripts for distinct challenges $x_0, \ldots, x_k$ with the same input $(C_0, \ldots, C_k)$, and responses $(\boldsymbol{f}_0, \boldsymbol{z}_0), \ldots, (\boldsymbol{f}_k, \boldsymbol{z}_k)$, represented as below.

$$\begin{pmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^k \\ 1 & x_1 & x_1^2 & \cdots & x_1^k \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_k & x_k^2 & \cdots & x_k^k \end{pmatrix} \cdot \begin{pmatrix} C_0 \\ C_1 \\ \vdots \\ C_k \end{pmatrix} = \begin{pmatrix} \mathrm{Com}_{ck}(\boldsymbol{f}_0;\, \boldsymbol{z}_0) \\ \mathrm{Com}_{ck}(\boldsymbol{f}_1;\, \boldsymbol{z}_1) \\ \vdots \\ \mathrm{Com}_{ck}(\boldsymbol{f}_k;\, \boldsymbol{z}_k) \end{pmatrix}. \tag{9}$$

We have seen that using the aforementioned *relaxed* opening approach, one can extract a witness from a linear relation (8) in *one shot*. Now a natural generalization is to ask "Can we extract a witness from a non-linear relation (9) as in Protocol 1 in *one shot*?"

Naive approach and previous *multi-shot* approach. Denoting (9) as $\boldsymbol{V} \cdot \boldsymbol{c} = \boldsymbol{b}$, the matrix $\boldsymbol{V}$ is a Vandermonde matrix. A straightforward idea to obtain the openings of C_i's is to multiply both sides of (9) by $\boldsymbol{V}^{-1}$, which gives $\boldsymbol{c} = \boldsymbol{V}^{-1} \cdot \boldsymbol{b}$. From here, using the homomorphic properties of the commitment scheme, we can get *potential* "openings" of C_i's. However, one needs to make sure that $\boldsymbol{V}^{-1}$ exists over $\mathfrak{R}$ and that it has *short* entries so that these "openings" are valid. The way [19] deals with this issue is by making use of monomial challenges from [9]. Using the structure of $\boldsymbol{V}^{-1}$ in (7), it is argued in [19] that the entries in $2^k \boldsymbol{V}^{-1}$ are short by the fact that doubled inverse of challenge differences (i.e., $2(x_j - x_i)^{-1}$) are short *when* monomial challenges are used. Thus, this approach still maintains the drawback of requiring multiple protocol repetitions to achieve a negligible soundness error, and does not address our question.

Our *one-shot* solution. Now, let us see how we develop a one-shot proof technique for non-linear relations. Using (5), we multiply both sides of (9) by $\mathrm{adj}(\boldsymbol{V})$, and obtain

$$\mathrm{adj}(\boldsymbol{V}) \cdot \boldsymbol{V} \cdot \boldsymbol{c} = \mathrm{adj}(\boldsymbol{V}) \cdot \boldsymbol{b} \quad \Longrightarrow \quad \det(\boldsymbol{V}) \cdot \boldsymbol{c} = \mathrm{adj}(\boldsymbol{V}) \cdot \boldsymbol{b}. \tag{10}$$

Note that $\det(\boldsymbol{V})$ is just some scalar in $\mathfrak{R}$, and we obtain *potential relaxed* "openings" of C_i's as a result of the multiplication $\mathrm{adj}(\boldsymbol{V}) \cdot \boldsymbol{b}$. In particular, for the commitment C_k of the *witness*, we have

$$\det(\boldsymbol{V}) \cdot C_k = \sum_{i=0}^{k} \Gamma_i \cdot \mathrm{Com}_{ck}(\boldsymbol{f}_i;\, \boldsymbol{z}_i) = \mathrm{Com}_{ck}(\sum_{i=0}^{k} \Gamma_i \cdot \boldsymbol{f}_i;\, \sum_{i=0}^{k} \Gamma_i \cdot \boldsymbol{z}_i), \tag{11}$$

where $\Gamma_i = (-1)^{i+k} \prod_{0 \le l < j \le k \wedge j, l \neq i} (x_j - x_l)$ by Fact 2. As a result, we get a *relaxed* opening of C_k, or more precisely, an *exact* opening of $\det(\boldsymbol{V}) \cdot C_k$ as $(\hat{\boldsymbol{m}}_k, \hat{\boldsymbol{r}}_k) = \left(\sum_{i=0}^{k} \Gamma_i \boldsymbol{f}_i, \sum_{i=0}^{k} \Gamma_i \boldsymbol{z}_i\right)$. Provided that the norms of $\hat{\boldsymbol{m}}_k$ and $\hat{\boldsymbol{r}}_k$ are small, this gives a *valid* opening and thus can be related to a hard lattice problem (M-SIS, in particular). It is important to observe here that $\hat{\boldsymbol{m}}_k$ and $\hat{\boldsymbol{r}}_k$ do not involve any inverse term and can be guaranteed to be short by ensuring that Γ_i's are short. The opening of other C_i's can also be recovered in a similar fashion, but the case for C_k is sufficient for our applications.

When $k = 1$, i.e., when the protocol is 2-special sound, $\det(\boldsymbol{V}) = (x_1 - x_0)$ and $(\Gamma_0, \Gamma_1) = (-1, 1)$. Therefore, we exactly obtain (8) as a special case of (11) with $k = 1$. That is, we get the results of the previous approaches from [28, 29, 10, 6] as a special case of ours.

3.3 New tools for compact proofs

Let us analyze our generalized solution and introduce our new tools to get compact proofs. The results can be easily used in other protocols that use a challenge space of the form defined in (12) as they are independent of the low-level details of a protocol. Since the most commonly used challenge spaces (e.g., in [6, 7, 17, 30, 31]) for one-shot proofs are special cases of (12), our results are widely applicable. Let $\mathfrak{R} = R = \mathbb{Z}[X]/(X^d + 1)$ and $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ for $q \in \mathbb{Z}^+$. For $w \le d$ and $p \le q/2$, let $\mathcal{C}^d_{w,p}$ be the challenge space defined as

$$\mathcal{C}^d_{w,p} = \{\, x \in \mathbb{Z}[X] \,:\, \deg(x) = d - 1 \,\wedge\, \mathsf{HW}(x) = w \,\wedge\, \|x\|_\infty = p \,\}. \tag{12}$$

It is easy to observe that $\|x\|_1 \le pw$ for any $x \in \mathcal{C}^d_{w,p}$ and $|\mathcal{C}^d_{w,p}| = \binom{d}{w} \cdot (2p)^w$, which is, for example, larger than 2^{256} for $(d, w, p) = (256, 60, 1)$. We define $\Delta\mathcal{C}^d_{w,p}$ to be the set of challenge differences excluding zero.

Bound on the product of challenge differences.

Lemma 3. *For any $y_1, \ldots, y_n \in \Delta\mathcal{C}^d_{w,p}$, the following holds*

$$\|\prod_{i=1}^{n} y_i\|_\infty \le (2p)^n \cdot w^{n-1}, \quad \textit{and} \quad \|\prod_{i=1}^{n} y_i\| \le \sqrt{d} \cdot (2p)^n \cdot w^{n-1}.$$

Bound on the relaxation factor: $\det(\boldsymbol{V})$**.**

Lemma 4. *Let $\kappa = \binom{k+1}{2} = \frac{k(k+1)}{2}$. For the $(k+1)$-dimensional Vandermonde matrix $\boldsymbol{V}$ defined in (9) using the challenge space $\mathcal{C}^d_{w,p}$ in (12),*

$$\|\det(\boldsymbol{V})\|_\infty \leq (2p)^\kappa \cdot w^{\kappa-1}.$$

Proof. By Fact 1, $\det(\boldsymbol{V})$ has $\kappa = \binom{k+1}{2}$ multiplicands where each multiplicand is in $\Delta\mathcal{C}^d_{w,p}$. The result follows from Lemma 3. □

Bound on the extracted witness norm: $\mathrm{adj}(\boldsymbol{V}) \times$ (**openings of** $\boldsymbol{b}$)**.**

Lemma 5. *For $k \geq 1$ and $(\hat{\boldsymbol{m}}_k, \hat{\boldsymbol{r}}_k) = \left(\sum_{i=0}^k \Gamma_i \boldsymbol{f}_i, \sum_{i=0}^k \Gamma_i \boldsymbol{z}_i\right)$ where $\Gamma_i = \prod_{0 \leq l < j \leq k \wedge j, l \neq i}(x_j - x_l)$, the following holds, for $\kappa' = k(k-1)/2$,*

- $\|\hat{\boldsymbol{m}}_k\| \leq (k+1) \cdot d \cdot (2p)^{\kappa'} \cdot w^{\kappa'-1} \cdot \max_i \|\boldsymbol{f}_i\|$, *and*
- $\|\hat{\boldsymbol{r}}_k\| \leq (k+1) \cdot d \cdot (2p)^{\kappa'} \cdot w^{\kappa'-1} \cdot \max_i \|\boldsymbol{z}_i\|$.

The proofs of Lemma 3 and Lemma 5 are provided in Appendix H.

Reducing extracted witness norm in proofs with non-linear relations. In some proofs with non-linear polynomial relations such as our one-out-of-many proof, the extractor obtains an opening with a relaxation factor y of some component that is witness of a sub-protocol. Since the invertibility of y is not ensured, when this opening is used in the non-linear polynomial relation, the relaxation factor also gets exponentiated by the degree $k > 1$. In the end, instead of getting $\det(\boldsymbol{V})$ as the overall relaxation factor, we end up with relaxation factor $y^k \cdot \det(\boldsymbol{V})$. We use the lemma below to show that even though we cannot completely eliminate the extra term y^k, we can eliminate its exponent k. This results in obtaining an extracted witness with a smaller norm, and in turn, helps in getting shorter proofs. The proof of the lemma below is given in Appendix H.

Lemma 6. *Let $f, g \in R = \mathbb{Z}[X]/(X^d+1)$. If $f \cdot g^k = 0$ in $R_q = \mathbb{Z}_q[X]/(X^d+1)$ for some $k \in \mathbb{Z}^+$, then $f \cdot g = 0$ in R_q.*

4 New Techniques for Faster Lattice-based Proofs

In this section, we go into the details of our new techniques to get computation-efficient proofs. We first show a lemma that enables one to prove that if a product of polynomials is equal to zero in R_q and the norm of each factor is sufficiently small, then there must be a factor which is exactly equal to zero. This result works for any sufficiently large q, enabling the use of a modulus suitable for fast computation such as an "NTT-friendly" modulus.

Lemma 7. *Let $f_1, \ldots, f_n \in R$ for some $n \geq 1$. If $\prod_{i=1}^n f_i = 0$ in R_q and $q/2 > \|f_1\|_\infty \cdot \prod_{i=2}^n \|f_i\|_1$, then there exists $1 \leq j \leq n$ such that $f_j = 0$.*

Proof (Lemma 7). Using Lemma 8 and the assumption on q, we have

$$\|\prod_{i=1}^s f_i\|_\infty \leq \|f_1\|_\infty \cdot \prod_{i=2}^n \|f_i\|_1 < q/2.$$

Therefore, $\prod_{i=1}^n f_i = 0$ holds over R. Since $X^d + 1$ is irreducible over $\mathbb{Q}$, (at least) one of the multiplicand f_i's must be zero. □

Note that Lemma 7 requires all the multiplicands to have bounded norm whereas there is no such requirement in Lemma 6. Therefore, we are unable to use Lemma 7 for the purpose of the use of Lemma 6 described previously as there is no norm-bound on a multiplicand in the place Lemma 6 is used (see how these lemmas are used in the soundness proofs for more details). Lemma 7 is used in the binary proof to argue that $y_0 y_1 y_2 \hat{b}(y - \hat{b}) = 0$ in R_q for some (non-zero) challenge differences y, y_0, y_1, y_2 implies $\hat{b} = yb$ for a bit $b \in \{0, 1\}$ without requiring invertibility of any challenge difference (see Section 5.1).

4.1 Supporting inter-slot operations on CRT-packed messages

Now, we can go into the details of our CRT packing technique. Define $f = \mathrm{Enc}_x(m) = x \cdot m + \rho \in R_q$ as an encoding of a message m under a challenge x. This encoding is widely used in proofs of knowledge as a "masked" response to a challenge x. An important advantage of this encoding over a commitment is that the storage cost of an encoding is at most $d \log q$ whereas that of a commitment is $nd \log q$ for HMC and $(n + v)d \log q$ for UMC. Therefore, for a typical module rank of, say, 4, a commitment is $4\times$ more costly than an encoding.

There are known methods to choose a modulus q such that $X^d + 1$ splits into s factors, in which case, R_q splits into s fields and we get $R_q = R_q^{(0)} \times \cdots \times R_q^{(s-1)}$. In the case that $X^d + 1$ splits into more than s factors, but we only want to use s slots, we still have $R_q = R_q^{(0)} \times \cdots \times R_q^{(s-1)}$ where $R_q^{(i)} = \mathbb{Z}_q[X]/(P^{(i)}(X))$ for some polynomial $P^{(i)}(X)$ of degree d/s. However, $R_q^{(i)}$'s are not a field in that case as $P^{(i)}(X)$'s are not irreducible over $\mathbb{Z}_q$.

As discussed previously, when we use these s slots to pack s messages in a single ring element, we have

$$f = \mathrm{Enc}_x(m) = x \cdot m + \rho = \langle x_0 m_0 + \rho_0, \ldots, x_{s-1} m_{s-1} + \rho_{s-1} \rangle, \tag{13}$$

where $x = \langle x_0, \ldots, x_{s-1} \rangle$, $m = \langle m_0, \ldots, m_{s-1} \rangle$ and $\rho = \langle \rho_0, \ldots, \rho_{s-1} \rangle$ in the CRT-packed representation. In this case, parallel additions are easy as

$$\mathrm{Enc}_x(\langle m_0, \ldots, m_{s-1} \rangle) + \mathrm{Enc}_x(\langle m'_0, \ldots, m'_{s-1} \rangle) = \mathrm{Enc}_x(\langle m_0 + m'_0, \ldots, m_{s-1} + m'_{s-1} \rangle).$$

Parallel multiplication is also possible as $\mathrm{Enc}_x(m) \cdot \mathrm{Enc}_x(m') = m \cdot m' \cdot x^2 + c_1 x + c_0$ for c_0, c_1 only dependent on m, m', ρ, ρ', all of which are known to the prover in advance of his first move. Therefore, the prover can prove that the coefficient of x^2 is the product of m and m', and thus proving the relation in parallel for all CRT slots.[12] Addition and multiplication alone, however, do not provide a complete set of operations (see [21] for a discussion in the context of FHE). Given an encoding of m, our main requirement is to have the ability to extract all encodings in the CRT slots of m in a way that allows further operations among extracted encodings. That is, all extracted encodings must be under the same challenge x, which translates to requiring $x = \langle x, \ldots, x \rangle$ for $x \in \bigcap_{i=0}^{s-1} R_q^{(i)}$. As a result, when we use s slots, the degree of a challenge can be at most $d/s - 1$. With

[12] We believe this is the application of CRT mentioned in [31].

this, from an encoding $f = \mathrm{Enc}_x(\langle m_0, \ldots, m_{s-1}\rangle)$, anyone can extract encodings by computing

$$f_i = \mathrm{Enc}_x(m_i) = f \bmod (q, P^{(i)}(X)) = x \cdot m_i + \rho_i = \mathrm{Enc}_x(m_i)$$

for all $0 \leq i \leq s-1$. Conversely, given encoding $\mathrm{Enc}_x(m_i)$'s for all $0 \leq i \leq s-1$, anyone can compute an encoding $\mathrm{Enc}_x(\langle m_0, \ldots, m_{s-1}\rangle)$.

Even more, with this choice of the challenge $x = \langle x, \ldots, x\rangle$ for $x \in \bigcap_{i=0}^{s-1} R_q^{(i)}$, we get invariance of the challenge under *any* permutation σ on CRT slots. That is, for any permutation σ, we have $\sigma(\mathrm{Enc}_x(m)) = \mathrm{Enc}_x(\sigma(m))$. From here, one can perform any inter-slot operation, and may even not require packing/unpacking of the messages in some applications. In our application to the range proof, extraction of the slots is sufficient and we refer to [21] for more on permutations. In our approach, an encoding and a commitment per message slot costs, respectively, at most $d \log q/s$ bits and $(n+v)\log q/s$ bits, which are much cheaper than a commitment to a single message.

4.2 Using CRT-packed inter-slot operations in relaxed range proof

In this section, we introduce the first application of our ideas to Σ-protocols where the proof is *relaxed* as described in Section 2.3. In all of our protocols, the prover aborts if any rejection sampling step (Algorithm 1) returns 0, and our protocols are honest-verifier zero-knowledge for *non-aborting* interactions. For most of the practical applications, the protocol is made non-interactive, and thus having only non-aborting protocols with the zero-knowledge property does not cause an issue. Nevertheless, the protocols can be easily adapted to be zero-knowledge for the aborting cases using a standard technique from [9].

Our first application is a range proof that allows an efficient aggregation in the sense that the prover can prove that a set of committed values packed in a *single* commitment falls within a set of certain ranges. Let $\psi \in \mathbb{Z}^+$, $\ell^{(i)} \in [0, N_i)$ be prover's values for $1 \leq i \leq \psi$ and $N_i = 2^{k_i}$ with $k = k_1 + \cdots + k_\psi$, and s be the smallest power of two such that $s \geq \max\{k_1, \ldots, k_\psi\}$. For simplicity, we use base $\beta = 2$, but the result can be generalized to other base values β. Binary case gives the the most compact proofs in practice. Assume that $R_q = \mathbb{Z}_q[X]/(X^d+1)$ splits into exactly s fields such that $R_q = R_q^{(0)} \times \cdots \times R_q^{(s-1)}$ and $R_q^{(i)} = \mathbb{Z}_q[X]/(P^{(i)}(X))$ for some *irreducible* polynomial $P^{(i)}(X)$ of degree d/s for all $0 \leq i < s$. Write $\ell^{(i)} = (b_0^{(i)}, \ldots, b_{k_i-1}^{(i)})$ in the binary representation and define $\ell_{\mathrm{crti}}^{(i)} = \langle b_0^{(i)}, \ldots, b_{k_i-1}^{(i)}\rangle$. The exact relations proved by our "simultaneous" range proof is given in Definition 4. We show in Appendix G that the relaxed range proof is sufficient for an application in anonymous credentials. Such a "simultaneous" range proof is useful when showing a credential that a set of attributes such as age, expiry date, residential postcode etc. fall into some respective ranges, and this can be achieved with a single commitment and a single proof using our techniques.

Definition 4. *The following defines the relations for Protocol 2 for* $\mathcal{T}, \hat{\mathcal{T}} \in \mathbb{R}^+$.

$$\mathcal{R}_{\text{range}}(\mathcal{T}) = \left\{ \begin{array}{c} ((ck, V), (\ell^{(1)}, \ldots, \ell^{(\psi)}, \boldsymbol{r})) \;:\; \|\boldsymbol{r}\| \leq \mathcal{T} \;\wedge \\ V = \mathrm{Com}_{ck}(\ell^{(1)}, \ldots, \ell^{(\psi)}; \boldsymbol{r}) \;\wedge\; \ell^{(i)} \in [0, N_i) \; \forall 1 \leq i \leq \psi \end{array} \right\},$$

$$\mathcal{R}'_{\text{range}}(\hat{\mathcal{T}}) = \left\{ \begin{array}{l} ((ck, V), (\bar{x}, \ell^{(1)}, \ldots, \ell^{(\psi)}, \hat{\boldsymbol{r}})) \;:\; \|\hat{\boldsymbol{r}}\| \leq \hat{\mathcal{T}} \;\wedge\; \bar{x} \in \Delta\mathcal{C}_{w,p}^{d/s} \;\wedge \\ \bar{x}V = \mathrm{Com}_{ck}(\bar{x}\ell^{(1)}, \ldots, \bar{x}\ell^{(\psi)}; \hat{\boldsymbol{r}}) \wedge \ell^{(i)} \in [0, N_i) \; \forall 1 \leq i \leq \psi \end{array} \right\}.$$

The full description of the range proof is given in Protocol 2 where the commitment scheme is instantiated with UMC and ϕ_1, ϕ_2 are parameters determining the rejection sampling rate. The first part of the proof (Steps 4 and 5 in the verification, and its relevant components) uses the binary proof idea from [11, 19] to show that $f_j^{(i)}$'s are encodings of bits, but the proof is done in parallel CRT slots. Observe in Protocol 2 that $f^{(i)} = x \cdot \langle b_0^{(i)}, \ldots, b_{k_i-1}^{(i)}, \mathbf{0}^{s-k_i} \rangle + \langle a_0^{(i)}, \ldots, a_{k_i-1}^{(i)}, \mathbf{0}^{s-k_i} \rangle = x \cdot \ell_{\text{crti}}^{(i)} + a_{\text{crti}}^{(i)}$ where $\mathbf{0}^{s-k_i}$ denotes a zero vector of dimension $s - k_i$. Therefore, we have, for each $1 \leq i \leq \psi$,

$$f^{(i)}(x - f^{(i)}) = x^2 \cdot \ell_{\text{crti}}^{(i)}(1 - \ell_{\text{crti}}^{(i)}) + x \cdot a_{\text{crti}}^{(i)}(1 - 2\ell_{\text{crti}}^{(i)}) - (a_{\text{crti}}^{(i)})^2$$

Since there is no x^2 term (i.e., the coefficient of x^2 is zero) on the left hand side of Step 5 in the verification, we get $\ell_{\text{crti}}^{(i)}(1 - \ell_{\text{crti}}^{(i)}) = 0$ when Step 5 is satisfied for 3 distinct challenges x. This gives us

$$\langle b_0^{(i)}(1-b_0^{(i)}), \ldots, b_{k_i-1}^{(i)}(1-b_{k_i-1}^{(i)}), \mathbf{0}^{s-k_i} \rangle = 0 \implies b_j^{(i)}(1-b_j^{(i)}) = 0 \text{ in } R_q^{(j)} \quad (14)$$

for each $0 \leq j < s - k_i$. This fact is then used to prove that $b_j^{(i)}$'s are binary. However, since the proof is relaxed, we need to deal with more complicated issues and give the full details in the proofs of Theorem 1.

The second part of the proof is a standard argument to show that the bits $b_0^{(i)}, \ldots, b_{k_i-1}^{(i)}$ construct a value $\ell^{(i)}$ for each $1 \leq i \leq \psi$. We assumed N_i's are of the form $N_i = 2^{k_i}$ for $k_i \geq 1$. This can be extended to work for any range as described in Appendix C, where we also discuss about the practical aspects of the range proof. The following states the properties of Protocol 2.

Theorem 1. *Let* $\gamma_{\text{range}} = 4\sqrt{3}\phi_2 pw\mathcal{B}md$. *Assume* $q > \max\{N_1, \ldots, N_\psi\}$, $d \geq 128$,[13] R_q *splits into exactly s fields and UMC is hiding and* γ_{range}*-binding. Then, Protocol 2 is a* 3*-special sound Σ-protocol (as in Definition 3) for the relations* $\mathcal{R}_{\text{range}}(\mathcal{B}\sqrt{md})$ *and* $\mathcal{R}'_{\text{range}}(\gamma_{\text{range}})$ *with a completeness error* $1 - 1/(\mu(\phi_1)\mu(\phi_2))$ *for* $\mu(\cdot)$ *defined in Lemma 11.*

Proof (Theorem 1). Completeness and SHVZK proofs are given in Appendix H.
3-special soundness: Given 3 accepting protocol transcripts, we have $(A, B, C, D, E, x, \boldsymbol{f}_{\text{crt}}, \boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z})$, $(A, B, C, D, E, x', \boldsymbol{f}'_{\text{crt}}, \boldsymbol{z}'_b, \boldsymbol{z}'_c, \boldsymbol{z}')$, $(A, B, C, D, E, x'', \boldsymbol{f}''_{\text{crt}}, \boldsymbol{z}''_b, \boldsymbol{z}''_c, \boldsymbol{z}'')$, with $\boldsymbol{f} = (f^{(1)}, \ldots, f^{(\psi)})$, $\boldsymbol{f}' =$

[13] The assumption $d \geq 128$ is put merely to use a constant factor of 2 as in Lemma 10 when bounding the Euclidean norm of a vector following normal distribution.

$\mathcal{P}_{\text{range}}((ck, V), (\ell^{(1)}, \ldots, \ell^{(\psi)}; \boldsymbol{r}))$ $\qquad\qquad$ $\mathcal{V}_{\text{range}}(ck, V)$

$$\boldsymbol{r}_b, \boldsymbol{r}_c \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$$

$$\boldsymbol{r}_a, \boldsymbol{r}_d, \boldsymbol{r}_e \leftarrow D^{md}_{\phi_2 T_2} \text{ for } T_2 = pw\mathcal{B}\sqrt{3md}$$

$$\textbf{for } i = 1, \ldots, \psi \textbf{ do}$$

$$\quad a_0^{(i)}, \ldots, a_{k_i-1}^{(i)} \leftarrow D^{d/s}_{\phi_1 T_1} \text{ for } T_1 = p\sqrt{kw}$$

$$\quad a_{\text{crti}}^{(i)} = \texttt{CRT}^{-1}(a_0^{(i)}, \ldots, a_{k_i-1}^{(i)}, \mathbf{0}^{s-k_i})$$

$$\quad \ell_{\text{crti}}^{(i)} = \texttt{CRT}^{-1}(b_0^{(i)}, \ldots, b_{k_i-1}^{(i)}, \mathbf{0}^{s-k_i})$$

$$B = \text{Com}_{ck}(\ell_{\text{crti}}^{(1)}, \ldots, \ell_{\text{crti}}^{(\psi)}; \boldsymbol{r}_b)$$

$$A = \text{Com}_{ck}(a_{\text{crti}}^{(1)}, \ldots, a_{\text{crti}}^{(\psi)}; \boldsymbol{r}_a)$$

$$C = \text{Com}_{ck}(a_{\text{crti}}^{(1)}(1 - 2\ell_{\text{crti}}^{(1)}), \ldots, a_{\text{crti}}^{(\psi)}(1 - 2\ell_{\text{crti}}^{(\psi)}); \boldsymbol{r}_c)$$

$$D = \text{Com}_{ck}(-(a_{\text{crti}}^{(1)})^2, \ldots, -(a_{\text{crti}}^{(\psi)})^2; \boldsymbol{r}_d)$$

$$E = \text{Com}_{ck}(\boldsymbol{e}; \boldsymbol{r}_e) \qquad \xrightarrow{\quad A, B, C, D, E \quad}$$

$$\xleftarrow{\quad x \quad} \qquad x \leftarrow \mathcal{C}^{d'}_{w,p} \text{ for } d' = d/s$$

$$\textbf{for } i \in [1, \psi], j \in [0, k_i) \textbf{ do}$$

$$\quad f_j^{(i)} = x \cdot b_j^{(i)} + a_j^{(i)}$$

$$\boldsymbol{f}_{\text{crt}} := (f_0^{(1)}, \ldots, f_{k_\psi - 1}^{(\psi)}), \boldsymbol{b} := (b_0^{(1)}, \ldots, b_{k_\psi - 1}^{(\psi)})$$

$$\text{Rej}(\boldsymbol{f}_{\text{crt}}, x\boldsymbol{b}, \phi_1, p\sqrt{kw})$$

$$\boldsymbol{z}_b = x \cdot \boldsymbol{r}_b + \boldsymbol{r}_a, \ \boldsymbol{z}_c = x \cdot \boldsymbol{r}_c + \boldsymbol{r}_d$$

$$\boldsymbol{z} = x \cdot \boldsymbol{r} + \boldsymbol{r}_e$$

$$\text{Rej}((\boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}), x(\boldsymbol{r}_b, \boldsymbol{r}_c, \boldsymbol{r}), \phi_2, T_2)$$

If aborted, return $\perp$. $\qquad \xrightarrow{\quad \boldsymbol{f}_{\text{crt}}, \boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z} \quad}$

$$\textbf{for } i = 1, \ldots, \psi \textbf{ do}$$

$$\quad f^{(i)} = \texttt{CRT}^{-1}(f_0^{(i)}, \ldots, f_{k_i-1}^{(i)}, \mathbf{0}^{s-k_i})$$

$$\|\boldsymbol{z}_b\|, \|\boldsymbol{z}_c\|, \|\boldsymbol{z}\| \stackrel{?}{\leq} 2\phi_2 T_2 \sqrt{md}$$

$$xB + A \stackrel{?}{=} \text{Com}_{ck}(f^{(0)}, \ldots, f^{(\psi)}; \boldsymbol{z}_b)$$

$$\boldsymbol{g} := (f^{(0)}(x - f^{(0)}), \ldots, f^{(\psi)}(x - f^{(\psi)}))$$

$$xC + D \stackrel{?}{=} \text{Com}_{ck}(\boldsymbol{g}; \boldsymbol{z}_c)$$

$$xV + E \stackrel{?}{=} \text{Com}_{ck}(\boldsymbol{v}; \boldsymbol{z})$$

Protocol 2: Σ-protocol for $\mathcal{R}_{\text{range}}$ and $\mathcal{R}'_{\text{range}}$. The vectors $\boldsymbol{e}$ and $\boldsymbol{v}$ are defined below.
$\boldsymbol{e} := \left(\sum_{j=0}^{k_1-1} 2^j a_j^{(1)}, \ldots, \sum_{j=0}^{k_\psi-1} 2^j a_j^{(\psi)}\right), \boldsymbol{v} := \left(\sum_{j=0}^{k_1-1} 2^j f_j^{(1)}, \ldots, \sum_{j=0}^{k_\psi-1} 2^j f_j^{(\psi)}\right)$ over R_q.

$(f'^{(1)}, \ldots, f'^{(\psi)})$ and $\boldsymbol{f}'' = (f''^{(1)}, \ldots, f''^{(\psi)})$ computed as in the verification. We split the proof into two parts: binary proof and range proof.
Binary proof. By Step 4 in the verification, we have

$$xB + A = \mathrm{Com}_{ck}(\boldsymbol{f};\, \boldsymbol{z}_b), \tag{15}$$
$$x'B + A = \mathrm{Com}_{ck}(\boldsymbol{f}';\, \boldsymbol{z}'_b), \tag{16}$$
$$x''B + A = \mathrm{Com}_{ck}(\boldsymbol{f}'';\, \boldsymbol{z}''_b). \tag{17}$$

Subtracting (16) from (15), we get $(x - x') \cdot B = \mathrm{Com}_{ck}(\boldsymbol{f} - \boldsymbol{f}';\, \boldsymbol{z}_b - \boldsymbol{z}'_b)$. Thus, for $y := x - x'$, we get exact valid openings of yB such that

$$yB = \mathrm{Com}_{ck}(\boldsymbol{f} - \boldsymbol{f}';\, \boldsymbol{z}_b - \boldsymbol{z}'_b) =: \mathrm{Com}_{ck}(\hat{\boldsymbol{b}};\, \hat{\boldsymbol{r}}_b). \tag{18}$$

Note that $\|\hat{\boldsymbol{r}}_b\| = \|\boldsymbol{z}_b - \boldsymbol{z}'_b\| \le 4\sqrt{3}\phi_2 pw\mathcal{B}md = \gamma_{\mathrm{range}}$, proving the claimed bound for $\mathcal{R}'_{\mathrm{range}}$. Multiplying (15) by y and using (18) gives

$$\begin{aligned} yA &= \mathrm{Com}_{ck}(y\boldsymbol{f};\, y\boldsymbol{z}_b) - xyB = \mathrm{Com}_{ck}(y\boldsymbol{f} - x\hat{\boldsymbol{b}};\, y\boldsymbol{z}_b - x\hat{\boldsymbol{r}}_b) \\ &= \mathrm{Com}_{ck}(x\boldsymbol{f}' - x'\boldsymbol{f};\, x\boldsymbol{z}'_b - x'\boldsymbol{z}_b) =: \mathrm{Com}_{ck}(\hat{\boldsymbol{a}};\, \hat{\boldsymbol{r}}_a). \end{aligned} \tag{19}$$

Observe that $y\boldsymbol{f} = x\hat{\boldsymbol{b}} + \hat{\boldsymbol{a}}$ by the definition of $\hat{\boldsymbol{a}}$. By the Chinese Remainder Theorem, the equality holds in each CRT slot. Using Step 5 of the verification in a similar manner, we get exact message openings $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{d}}$ of yC and yD such that $y\boldsymbol{g} = x\hat{\boldsymbol{c}} + \hat{\boldsymbol{d}}$. Writing these equations coordinate-wise in each CRT slot, we have the following for all $1 \le i \le \psi$ and $0 \le j \le s-1$

$$yf_j^{(i)} = x\hat{b}_j^{(i)} + \hat{a}_j^{(i)} \quad \text{in } R_q^{(j)}, \text{ and} \tag{20}$$
$$yg_j^{(i)} = yf_j^{(i)}(x - f_j^{(i)}) = x\hat{c}_j^{(i)} + \hat{d}_j^{(i)} \quad \text{in } R_q^{(j)}, \tag{21}$$

since all the challenges and their differences are the same in each CRT slot. Now, by the γ_{range}-binding property of UMC, except with negligible probability, the PPT prover cannot output a new valid exact opening of yA, yB, yC or yD in any of its rewinds. Thus, except with negligible probability, responses with respect to x' and x'' will have the same form. That is, the following holds

$$\begin{aligned} yf_j'^{(i)} &= x'\hat{b}_j^{(i)} + \hat{a}_j^{(i)}, & yf_j'^{(i)}(x' - f_j'^{(i)}) &= x'\hat{c}_j^{(i)} + \hat{d}_j^{(i)}, \\ yf_j''^{(i)} &= x''\hat{b}_j^{(i)} + \hat{a}_j^{(i)}, & yf_j''^{(i)}(x'' - f_j''^{(i)}) &= x''\hat{c}_j^{(i)} + \hat{d}_j^{(i)}, \end{aligned} \quad \text{in } R_q^{(j)}. \tag{22}$$

Now, multiplying (21) by y and using (20), we get

$$\begin{aligned} y \cdot \left(x \cdot \hat{c}_j^{(i)} + \hat{d}_j^{(i)}\right) &= y \cdot \left(yf_j^{(i)}(x - f_j^{(i)})\right) = yf_j^{(i)}(yx - yf_j^{(i)}) \\ &= (x\hat{b}_j^{(i)} + \hat{a}_j^{(i)})(yx - x\hat{b}_j^{(i)} - \hat{a}_j^{(i)}) = (x\hat{b}_j^{(i)} + \hat{a}_j^{(i)})(x(y - \hat{b}_j^{(i)}) - \hat{a}_j^{(i)}) \\ &= x^2\left[\hat{b}_j^{(i)}(y - \hat{b}_j^{(i)})\right] + x\left[\hat{a}_j^{(i)}(y - 2\hat{b}_j^{(i)})\right] - (\hat{a}_j^{(i)})^2, \end{aligned} \tag{23}$$

and thus

$$x^2\left[\hat{b}_j^{(i)}(y - \hat{b}_j^{(i)})\right] + x\left[\hat{a}_j^{(i)}(y - 2\hat{b}_j^{(i)}) - y\hat{c}_j^{(i)}\right] - (\hat{a}_j^{(i)})^2 - y\hat{d}_j^{(i)} = 0 \quad \text{in } R_q^{(j)}. \tag{24}$$

Repeating the same steps of (23) with the equations in (22), we get two copies of (24) where x is replaced with x' in one and with x'' in the other. That is, we have the following system

$$\begin{pmatrix} 1 & x & x^2 \\ 1 & x' & x'^2 \\ 1 & x'' & x''^2 \end{pmatrix} \cdot \begin{pmatrix} -(\hat{a}_j^{(i)})^2 - y\hat{d}_j^{(i)} \\ \hat{a}_j^{(i)}(y - 2\hat{b}_j^{(i)}) - y\hat{c}_j^{(i)} \\ \hat{b}_j^{(i)}(y - \hat{b}_j^{(i)}) \end{pmatrix} = \mathbf{0} \qquad \text{in } R_q^{(j)}. \tag{25}$$

Since $R_q^{(j)}$ is a field, the Vandermonde matrix on the left is invertible for distinct challenges, and we get $\hat{b}_j^{(i)}(y - \hat{b}_j^{(i)}) = 0$, which implies $\hat{b}_j^{(i)} \in \{0, y\}$ in a field, i.e.,

$$\hat{b}_j^{(i)} = yb_j^{(i)} \quad \text{for } b_j^{(i)} \in \{0, 1\}. \tag{26}$$

The range proof part is rather easier and is given in Appendix H. □

Remark 1. The first rejection sampling at Step 14 of Protocol 2 is not necessary as UMC allows unbounded-length messages. However, when rejection sampling is done, the bitsize of $f_j^{(i)}$'s are smaller (about a factor 3) than $d \log q/s$, which is the bitsize of a random element in $R_q^{(j)}$. Further, there is no mod q reduction in the prover's response, and also no mod $P^{(j)}(X)$ at Step 13 of Protocol 2 since $b_j^{(i)}$'s are binary.

5 Efficient One-Shot Proofs for Useful Relations

5.1 Relaxed proof of commitment to sequences of bits

Using our new techniques, we extend the multi-shot proof of commitment to bits from [19] to a one-shot proof. Our protocol proves a weaker relation but, the relaxation is tailored in a way that the soundness proof of the higher level proofs (Protocol 3) still work. The protocol proves that a commitment B opens to sequences of binary values such that there is a single 1 in each sequence, i.e., Hamming weight of each sequence is exactly 1. The relations of our binary proof are defined in Definition 5 where $\boldsymbol{b} = (b_{0,0}, \ldots, b_{k-1,\beta-1})$ for $k \geq 1, \beta \geq 2$.

Definition 5. *The following defines the relations for Protocol 4 for* $\mathcal{T}, \hat{\mathcal{T}} \in \mathbb{R}^+$.

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

$$\mathcal{R}'_{\text{bin}}(\hat{\mathcal{T}}) = \left\{ \begin{matrix} ((ck, B), (y, \boldsymbol{b}, \hat{\boldsymbol{r}})) : \|\hat{\boldsymbol{r}}\| \leq \hat{\mathcal{T}} \; \wedge \; (b_{j,i} \in \{0,1\} \; \forall j, i) \; \wedge \\ y \in \Delta\mathcal{C}_{w,p}^d \; \wedge \; yB = \text{Com}_{ck}(y\boldsymbol{b}; \hat{\boldsymbol{r}}) \; \wedge \; (\sum_{i=0}^{\beta-1} b_{j,i} = 1 \; \forall j) \end{matrix} \right\}.$$

The idea of the binary proof (combined with the CRT-packing technique) is already used in Protocol 2. The condition on the Hamming weight is the difference to Protocol 2 and is handled with a small modification. We defer its full description to Appendix D and show below the crucial part in making the binary proof work in a fully-splitting ring R_q.

Handling binary proof for NTT-friendly modulus q. As in (25) in the soundness proof of Theorem 1, we get the same system of equations below in the soundness proof of Protocol 4

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

where $\hat{b}_{j,i}$ are the values we want to prove to be of the form $\hat{b}_{j,i} = yb_{j,i}$ for $b_{j,i} \in \{0,1\}$. The difference now is that all equations now hold in R_q, and we cannot use any invertibility argument. Multiplying both sides of the above system by $\text{adj}(\boldsymbol{V})$ where $\boldsymbol{V}$ is the Vandermonde matrix on the left, we get

$$\det(\boldsymbol{V})\hat{b}_{j,i}(y - \hat{b}_{j,i}) = (x'' - x')(x' - x)(x'' - x)\hat{b}_{j,i}(y - \hat{b}_{j,i}) = 0 \quad \text{in } R_q. \quad (27)$$

We show in the proof of Theorem 2 that $\|(x''-x')(x'-x)(x''-x)\hat{b}_{j,i}(y-\hat{b}_{j,i})\|_\infty \leq 2^7\phi_1^2p^5w^3d^2k\beta$. Therefore assuming $q/2 > 2^7\phi_1^2p^5w^3d^2k\beta$, one of the factors in (27) must be zero by Lemma 7. As challenge differences are non-zero, this gives either $\hat{b}_{j,i}$ or $y - \hat{b}_{j,i}$ is zero. Thus, we get $\hat{b}_{j,i} \in \{0, y\}$. That is, $\hat{b}_{j,i} = yb_{j,i}$ for $b_{j,i} \in \{0,1\}$ as needed for $\mathcal{R}'_{\text{bin}}$. We state the results in the theorem below, and defer its full proof to Appendix H.

Theorem 2. *Let* $\gamma_{\text{bin}} = 2p\sqrt{dw}\left(16\phi_1^4p^4d^3k^3w^2\beta(\beta+1) + 12\phi_2^2p^2w^2\mathcal{B}^2m^2d^2\right)^{1/2}$. *Assume that* $d \geq 128$, $q/2 > 2^7\phi_1^2p^5w^3d^2k\beta$ *and HMC is hiding and* γ_{bin}*-binding. Then, Protocol 4 is a* 3*-special sound* Σ*-protocol (as in Definition 3) for the relations* $\mathcal{R}_{\text{bin}}(\mathcal{B}\sqrt{md})$ *and* $\mathcal{R}'_{\text{bin}}(4\sqrt{2}\phi_2pw\mathcal{B}md)$ *with a completeness error* $1 - 1/(\mu(\phi_1)\mu(\phi_2))$ *for* $\mu(\cdot)$ *defined in Lemma 11.*

5.2 Relaxed one-out-of-many proof

Our one-out-of-many proof has the same structure as in [19], which combines some ideas from [22, 11]. The main differences of our proof from the one in [19] are the use of an exponentially large challenge set, enabling one-shot proofs, the relation the verifier is convinced of and some tweaks to the rejection sampling. The challenging parts here is the soundness proof of the protocol. We use our new tools, namely Lemmas 3, 5 and 6, from Section 3 to make the soundness proof work.

Let $\boldsymbol{L} = \{P_0, \ldots, P_{N-1}\}$ be a set of public commitments for some $N \geq 1$. The prover's goal is to show that he knows an opening of one of these P_i's. In common with the previous works [22, 11, 19], we assume that $N = \beta^k$, which can be easily satisfied by adding dummy values to $\boldsymbol{L}$ when needed. Suppose that the prover's commitment is P_ℓ for some $0 \leq \ell < N$. Observe that $\sum_{i=0}^{N-1} \delta_{\ell,i}P_i = P_\ell$. The idea for the proof is then to prove knowledge of the index ℓ with $\sum_{i=0}^{N-1} \delta_{\ell,i}P_i$ being a commitment to zero. Writing $\ell = (\ell_0, \ldots, \ell_{k-1})$ and $i = (i_0, \ldots, i_{k-1})$ as the representations in base β, we have $\delta_{\ell,i} = \prod_{j=0}^{k-1} \delta_{\ell_j,i_j}$. The prover first commits to the sequences $(\delta_{\ell_j,0}, \ldots, \delta_{\ell_j,\beta-1})$ for all $0 \leq j \leq k-1$, and then uses Protocol 4 to show that they are well-formed (i.e., they construct an index in the range $[0, N)$ as in the range proof). Let us define the proved relations next.

Definition 6. *The following defines the relations for Protocol 3 for $\mathcal{T}, \hat{\mathcal{T}} \in \mathbb{R}^+$.*

$$\mathcal{R}_{1/\mathrm{N}}(\mathcal{T}) = \left\{ \begin{array}{c} ((ck, (P_0, \ldots, P_{N-1})), (\ell, \boldsymbol{r})) \ : \\ \ell \in [0, N) \ \wedge \ \|\boldsymbol{r}\| \leq \mathcal{T} \ \wedge \ P_\ell = \mathrm{Com}_{ck}(\boldsymbol{0}; \boldsymbol{r}) \end{array} \right\},$$

$$\mathcal{R}'_{1/\mathrm{N}}(\hat{\mathcal{T}}) = \left\{ \begin{array}{c} ((ck, (P_0, \ldots, P_{N-1})), (y, \ell, \hat{\boldsymbol{r}})) \ : \ \ell \in [0, N) \ \wedge \ \|\hat{\boldsymbol{r}}\| \leq \hat{\mathcal{T}} \ \wedge \\ yP_\ell = \mathrm{Com}_{ck}(\boldsymbol{0}; \hat{\boldsymbol{r}}) \ \wedge \ y \textit{ is a product of elements in } \Delta\mathcal{C}^d_{w,p} \end{array} \right\}.$$

From Protocol 4, the prover's response contains $f_{j,i} = x\delta_{\ell_j,i} + a_{j,i}$ for a challenge x. Considering the product $p_i(x) := \prod_{j=0}^{k-1} f_{j,i_j}$, we see that, for all $i \in [0, N-1]$,

$$p_i(x) = \prod_{j=0}^{k-1} \left(x\delta_{\ell_j,i_j} + a_{j,i_j}\right) = \prod_{j=0}^{k-1} x \cdot \delta_{\ell_j,i_j} + \sum_{j=0}^{k-1} p_{i,j}x^j = \delta_{\ell,i}x^k + \sum_{j=0}^{k-1} p_{i,j}x^j, \quad (28)$$

for some ring element $p_{i,j}$'s as a function of ℓ and $a_{j,i}$'s (independent of the challenge x). Since ℓ and $a_{j,i}$'s are known to the prover before receiving a challenge, he can compute $p_{i,j}$'s prior to sending the initial commitment. Since p_ℓ is the only such polynomial of degree k, in his first move, the prover sends some E_j's that are tailored to cancel out the coefficients of the terms $1, x, \ldots, x^{k-1}$, and the coefficient of x^k is set to the prover's commitment P_ℓ using $\sum_{i=0}^{N-1} \delta_{\ell,i}P_i$. The full description is given in Protocol 3. In Appendix E, we show how our one-out-of-many proof can be extended to a set membership proof.

Theorem 3. *Let $\gamma_{1/\mathrm{N}} = (k+1)2^{\kappa'+2}\sqrt{3}\phi_2\mathcal{B}md^2w^\kappa p^{\kappa+1}$ for $\kappa' = k(k-1)/2$ and $\kappa = k(k+1)/2$. Assume $d \geq 128$, $q > 2^7\phi_1^2p^5w^3d^2k\beta$ and HMC is hiding and γ-binding for $\gamma = \max\{\gamma_{\mathrm{bin}}, \gamma_{1/\mathrm{N}}\}$. For $\mu(\cdot)$ defined in Lemma 11, Protocol 3 is a $(k'+1)$-special sound Σ-protocol (as in Definition 3) for the relations $\mathcal{R}_{1/\mathrm{N}}(\mathcal{B}\sqrt{md})$ and $\mathcal{R}'_{1/\mathrm{N}}(\gamma_{1/\mathrm{N}})$ with a completeness error $1 - 1/(\mu(\phi_1)\mu(\phi_2))$ where $k' = \max\{2, k\}$.*

Proof (Theorem 3). Completeness and SHVZK proofs are given to Appendix H. $(k+1)$**-special soundness:** Given $(k+1)$ distinct challenges $x_0, \ldots, x_k$, we have $(k+1)$ accepting responses with the same $(A, B, C, D, E_0, \ldots, E_{k-1})$. Assume that $k > 1$ and $(\boldsymbol{f}_1^{(0)}, \boldsymbol{z}^{(0)}), \ldots, (\boldsymbol{f}_1^{(k)}, \boldsymbol{z}^{(k)})$ are part of the responses with respect to challenges $x_0, \ldots, x_k$, respectively. Setting $y = x_1 - x_0$, we first use 3-special soundness of Protocol 4 to extract exact valid message openings $\hat{b}_{j,i}$ and $\hat{a}_{j,i}$ of yB and yA, respectively. We know that $\hat{b}_{j,i} = yb_{j,i}$ for $b_{j,i} \in \{0, 1\}$ and only a single one of $\{b_{j,0}, \ldots, b_{j,\beta-1}\}$ is 1 for each $j \in \{0, \ldots, k-1\}$. Now, we construct the representation of ℓ in base β as follows. For each $0 \leq j \leq k-1$, the j-th digit ℓ_j is the integer c such that $b_{j,c} = 1$. It is easy to construct the index ℓ from here using its digit ℓ_j's.

Recalling equations in (39) from the soundness proof of Protocol 4 that use γ_{bin}-binding property of the commitment scheme, we have, for all $0 \leq \eta \leq k-1$,

$$yf_{j,i}^{(\eta)} = x_\eta\hat{b}_{j,i} + \hat{a}_{j,i} = x_\eta \cdot yb_{j,i} + \hat{a}_{j,i}.$$

Now compute $\hat{p}_i(x_\eta) = y^k \prod_{j=0}^{k-1} f_{j,i_j}^{(\eta)} = \prod_{j=0}^{k-1} yf_{j,i_j}^{(\eta)} = \prod_{j=0}^{k-1} \left(yx_\eta b_{j,i_j} + \hat{a}_{j,i_j}\right)$ for each $i = 0, \ldots, N-1$. By the construction of ℓ, $\hat{p}_\ell(x_\eta)$ is the only polynomial of

$\mathcal{P}_{1/\mathrm{N}}((ck, (P_0, \ldots, P_{N-1})), (\ell, \boldsymbol{r}))$ $\qquad\qquad$ $\mathcal{V}_{1/\mathrm{N}}(ck, (P_0, \ldots, P_{N-1}))$

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: $A, C, D \leftarrow \mathcal{P}_{\mathrm{bin}}((ck, B), (\boldsymbol{\delta}, \boldsymbol{r}_b))$

5: $\boldsymbol{\rho}_0 \leftarrow D^{md}_{\phi_2 T_2}$ for $T_2 = \mathcal{B}p^k w^k \sqrt{3md}$

6: **for** $j = 0, \ldots, k-1$ **do**

7: $\quad \boldsymbol{\rho}_j \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$ if $j \neq 0$

8: $\quad E_j = \sum_{i=0}^{N-1} p_{i,j} P_i + \mathrm{Com}_{ck}(\boldsymbol{0};\ \boldsymbol{\rho}_j)$

using $p_{i,j}$'s from (28) $\quad \xrightarrow{A, B, C, D, E_0, \ldots, E_{k-1}}$

$\xleftarrow{\quad x \quad}$ $\qquad x \leftarrow \mathcal{C}^d_{w,p}$

9: $\boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z}_c \leftarrow \mathcal{P}_{\mathrm{bin}}(x)$

10: $\mathrm{Rej}(\boldsymbol{f}_1, x\boldsymbol{\delta}_1, \phi_1, p\sqrt{kw})$ for $\boldsymbol{\delta}_1 := (\delta_{\ell_0,1}, \ldots, \delta_{\ell_{k-1},\beta-1})$

11: $\boldsymbol{z} = x^k \boldsymbol{r} - \sum_{j=0}^{k-1} x^j \boldsymbol{\rho}_j$

12: $\mathrm{Rej}((\boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}), (x\boldsymbol{r}_b, x\boldsymbol{r}_c, x^k \boldsymbol{r} - \sum_{j=1}^{k-1} x^j \boldsymbol{\rho}_j), \phi_2, T_2)$

If aborted, return $\perp$. $\quad \xrightarrow{\boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}}$

1: $\mathcal{V}_{\mathrm{bin}}(ck, B, x, A, C, D, \boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z}_c) \stackrel{?}{=} 1$

2: $\|\boldsymbol{z}\|, \|\boldsymbol{z}_b\|, \|\boldsymbol{z}_c\| \stackrel{?}{\leq} 2\sqrt{3}\phi_2 \mathcal{B} m d p^k w^k$

3: $\sum_{i=0}^{N-1} \left(\prod_{j=0}^{k-1} f_{j,i_j} \right) P_i - \sum_{j=0}^{k-1} E_j x^j \stackrel{?}{=} \mathrm{Com}_{ck}(\boldsymbol{0};\ \boldsymbol{z})$

Protocol 3: Σ-protocol for $\mathcal{R}_{1/\mathrm{N}}$ and $\mathcal{R}'_{1/\mathrm{N}}$. Step 5 of the verification (norm checks on $\boldsymbol{z}_b, \boldsymbol{z}_c$) in Protocol 4 is skipped when $\mathcal{V}_{\mathrm{bin}}(ck, B, x, A, C, D, \boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z}_c)$ is run.

degree k in x_η for all $0 \leq \eta \leq k-1$. Then, we can multiply the both sides of the last verification step by y^k and re-write it as below

$$\sum_{i=0}^{N-1} \hat{p}_i(x_\eta) P_i - \sum_{j=0}^{k-1} y^k E_j x_\eta^j = x_\eta^k \cdot y^k P_\ell + \sum_{j=0}^{k-1} \tilde{E}_j x_\eta^j = \mathrm{Com}_{ck}(\boldsymbol{0};\ y^k \boldsymbol{z}^{(\eta)}), \quad (29)$$

where $\tilde{E}_j$'s are the terms multiplied by the monomials x_η^j's of degree at most $k-1$ and are independent of x_η. Equation (29) is exactly the case described in (9) and the verification of Protocol 1 in Section 3 with $C_k = y^k P_\ell$. By the discussion in Section 3, we obtain exact openings of $\det(\boldsymbol{V})y^k P_\ell$ as $(\mathbf{0}, y^k \hat{\boldsymbol{r}})$ where $\hat{\boldsymbol{r}} = \sum_{i=0}^{k} \Gamma_i \boldsymbol{z}^{(i)}$ for $\Gamma_i = (-1)^{i+k} \prod_{0 \le l < j \le k \wedge j, l \ne i} (x_j - x_l)$, i.e., we have

$$\begin{aligned} \det(\boldsymbol{V})y^k P_\ell = \mathrm{Com}_{ck}(\mathbf{0};\, y^k \hat{\boldsymbol{r}}) &\implies y^k \cdot (\det(\boldsymbol{V})P_\ell - \mathrm{Com}_{ck}(\mathbf{0};\, \hat{\boldsymbol{r}})) = 0 \\ \text{(by Lemma 6)} &\implies y \cdot (\det(\boldsymbol{V})P_\ell - \mathrm{Com}_{ck}(\mathbf{0};\, \hat{\boldsymbol{r}})) = 0 \\ &\implies \det(\boldsymbol{V})yP_\ell = \mathrm{Com}_{ck}(\mathbf{0};\, y\hat{\boldsymbol{r}}). \end{aligned} \tag{30}$$

In the end, we have an exact opening of $\det(\boldsymbol{V})yP_\ell$ as $(\mathbf{0}, y\hat{\boldsymbol{r}})$. This randomness opening is a factor $y \in \Delta\mathcal{C}^d_{w,p}$ larger than what we have in Lemma 5. Thus, using Lemma 3 and Lemma 5, we conclude, for $\kappa' = k(k-1)/2$ and $\kappa = k(k+1)/2$,

$$\begin{aligned} \|y\hat{\boldsymbol{r}}\| &\le (k+1)d(2p)^{\kappa'+1} w^{\kappa'} \max_i \|\boldsymbol{z}^{(i)}\| \le (k+1)d(2p)^{\kappa'+1} w^{\kappa'} \cdot 2\sqrt{3}\phi_2 \mathcal{B} m d w^k p^k \\ &\le (k+1)2^{\kappa'+2}\sqrt{3}\phi_2 \mathcal{B} m d^2 w^{\kappa} p^{\kappa+1}. \end{aligned}$$

Recall that we assumed $k > 1$. When $k = 1$, Protocol 4 still needs 3 challenges for its soundness property. As a result, Protocol 3 is at least 3-special sound. □

6 Applications of Relaxed ZKPs to Advanced Tools

The relaxed range proof combined with a relaxed proof of knowledge results in a form of efficient anonymous credentials as detailed in Appendix G. To prove relations on a set of attributes, a single use of our range proof is sufficient and we show how the relaxation is handled. Our second construction is a ring signature that builds on the relaxed one-out-of-many proof.

Ring Signature. The construction of ring signature from one-out-of-many proof follows the same strategy as in [22, 11, 19]. The users commit to their secret keys and these commitments represent the public keys. A set of public keys is then used as the set of public commitments in the one-out-of-many proof. The prover proves knowledge of an opening of one of the commitments (i.e., knowledge of a secret key corresponding to one of the public keys used to construct the signature). The main difference from [22, 11, 19] is that we show that our relaxed proof is still sufficient. The formal definitions of a ring signature, our full construction, analysis of asymptotic signature length and computational efficiency are provided in Appendix F. The concrete instantiation of the parameters is given in Table 4 and we refer to Appendix F.3 for more details.

Table 4: Parameter setting of our ring signature with a root Hermite factor ≤ 1.0045 for both M-LWE and M-SIS. $\mathcal{B} = 1, \phi_1 = \phi_2 = 15$ for all cases.

N	2	8	64	2^{12}	2^{21}
$(d,\, w,\, p)$	$(256, 60, 1)$	$(256, 60, 1)$	$(128, 66, 2)$	$(128, 66, 2)$	$(128, 66, 2)$
$(n,\, m)$	$(4, 12)$	$(4, 13)$	$(10, 28)$	$(13, 32)$	$(22, 46)$
$(k,\, \beta)$	$(1, 2)$	$(1, 8)$	$(1, 64)$	$(2, 64)$	$(3, 128)$
q	$\approx 2^{53}$	$\approx 2^{58}$	$\approx 2^{59}$	$\approx 2^{60}$	$\approx 2^{77}$
Signature Length (KB)	36	41	58	103	256
Public Key Length (KB)	6.63	7.25	9.22	12.19	26.47
Secret Key Length (KB)	0.38	0.41	0.44	0.50	0.72

References

[1] M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

[2] M. R. Albrecht, C. Rechberger, T. Schneider, T. Tiessen, and M. Zohner. Ciphers for MPC and FHE. In *EUROCRYPT*, LNCS, pages 430–454. Springer, 2015.

[3] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum key exchange-a new hope. In *USENIX Security Symposium*, 2016.

[4] S. Arora and R. Ge. New algorithms for learning in presence of errors. In *ICALP*, pages 403–415. Springer, 2011.

[5] C. Baum, J. Bootle, A. Cerulli, R. del Pino, J. Groth, and V. Lyubashevsky. Sub-linear lattice-based zero-knowledge arguments for arithmetic circuits. In *CRYPTO*, volume 10992 of *LNCS*, pages 669–699. Springer, 2018.

[6] C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More efficient commitments from structured lattice assumptions. In *SCN*, pages 368–385. Springer, 2018.

[7] C. Baum, H. Lin, and S. Oechsner. Towards practical lattice-based one-time linkable ring signatures. In *ICICS*, LNCS, pages 303–322. Springer, 2018.

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

[9] 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*, pages 551–572. Springer, 2014.

[10] 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.

[11] J. Bootle, A. Cerulli, P. Chaidos, E. Ghadafi, J. Groth, and C. Petit. Short accountable ring signatures based on DDH. In *ESORICS*, pages 243–265. Springer, 2015.

[12] J. Bootle, A. Cerulli, P. Chaidos, J. Groth, and C. Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In *EUROCRYPT*, pages 327–357. Springer, 2016.

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

[14] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *IEEE Symposium on Security and Privacy*, pages 315–334. IEEE, 2018.

[15] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.

[16] C. Chuengsatiansup. Private communication, 2018.

[17] R. del Pino, V. Lyubashevsky, and G. Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In *ACM CCS*, pages 574–591. ACM, 2018.

[18] D. Derler, S. Ramacher, and D. Slamanig. Post-quantum zero-knowledge proofs for accumulators with applications to ring signatures from symmetric-key primitives. In *PQCrypto*, pages 419–440. Springer, 2018. (Extended version at https://eprint.iacr.org/2017/1154).

[19] M. F. Esgin, R. Steinfeld, A. Sakzad, J. K. Liu, and D. Liu. Short lattice-based one-out-of-many proofs and applications to ring signatures. Cryptology ePrint

Archive, Report 2018/773, 2018. https://eprint.iacr.org/2018/773 (To appear ACNS 2019).

[20] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194. Springer, 1986.

[21] C. Gentry, S. Halevi, and N. P. Smart. Fully homomorphic encryption with polylog overhead. In *EUROCRYPT*, volume 7237 of *LNCS*, pages 465–482. Springer, 2012.

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

[23] R. A. Horn, R. A. Horn, and C. R. Johnson. *Matrix analysis*. Cambridge university press, 1990.

[24] J. Katz, V. Kolesnikov, and X. Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In *ACM CCS*, pages 525–537. ACM, 2018.

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

[26] 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.

[27] X. Lu, M. H. Au, and Z. Zhang. Raptor: A practical lattice-based (linkable) ring signature. Cryptology ePrint Archive, Report 2018/857, 2018. (To appear ACNS 2019).

[28] V. Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, pages 598–616. Springer, 2009.

[29] V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, pages 738–755. Springer, 2012. (Full version).

[30] V. Lyubashevsky and G. Neven. One-shot verifiable encryption from lattices. In *EUROCRYPT*, pages 293–323. Springer, 2017.

[31] 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.

[32] D. Micciancio and C. Peikert. Hardness of sis and lwe with small parameters. In *Advances in Cryptology–CRYPTO 2013*, pages 21–39. Springer, 2013.

[33] D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

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

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

[36] N. P. Smart and F. Vercauteren. Fully homomorphic SIMD operations. *Des. Codes Cryptography*, 71(1):57–81, 2014.

[37] D. Stehlé, R. Steinfeld, K. Tanaka, and K. Xagawa. Efficient public key encryption based on ideal lattices. In *ASIACRYPT*, pages 617–635. Springer, 2009.

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

[39] W. A. A. Torres, R. Steinfeld, A. Sakzad, J. K. Liu, V. Kuchta, N. Bhattacharjee, M. H. Au, and J. Cheng. Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (lattice ringct v1. 0). In *ACISP*, pages 558–576. Springer, 2018.

[40] R. K. Zhao, R. Steinfeld, and A. Sakzad. FACCT: Fast, compact, and constant-time discrete gaussian sampler over integers. Cryptology ePrint Archive, Report 2018/1234, 2018. https://eprint.iacr.org/2018/1234.

A More Details about Compared Works

We provide more details about our comparison in Table 1 with the related works. First, all the very large values are rounded for readability and we consider an overall challenge space of size 2^{256} for 128-bit post-quantum security. As mentioned, the root-Hermite factor δ (for both M-SIS and M-LWE) is at most 1.0045 for our parameters. The results of [26] are obtained using the same parameters given in [26] for 80-bit security. However, we increase the number of protocol repetitions to match 2^{256} elements in total for the challenge space. For [19], we use the formula provided in their paper, set the root Hermite factor for M-SIS to be 1.0045 and again match the total challenge space size to 2^{256}.

The ring signatures in [39, 7] are linear-sized, and thus their sizes increase very quickly. [39] builds a linkable ring signature and sets the parameters for 100-bit security and a root-Hermite factor of 1.007. We take the parameters in Column IV of Table 1 in [39] (the current latest 10-Nov-2018 version of their paper on ePrint) as that ring dimension matches our results and the one in [7]. Then, we scale up their results by a factor of 1.28 (assuming that the signature length scales linearly with the security parameter at best).

We are not always able to compute the exact results for [7, 24] as no explicit formula is provided. Instead, we approximate (whenever we can) the lengths based on the results provided in those papers. The root-Hermite factor considered in [7] is 1.0030 and 128-bit security is aimed at. The values for [18, 24] are those obtained using Davies-Meyer transform (i.e., the smaller signature lengths), and the improved results in the extended version of [18] is used with the formula provided in their work. We also note that [18, 24] is based on a symmetric-key primitive called LowMC [2], specifically designed to have small number of multiplication gates and its security is not well understood currently. If one instead uses a standard symmetric-key primitive such as AES, then the signature lengths of [18, 24] grow by a factor more than 5.

B More on Preliminaries

B.1 Notation

The main security parameter is denoted by λ where we use $\lambda = 128$ for concrete parameter estimates. A function $\nu(\lambda)$ is said to be negligible (denoted by $\nu = \text{negl}(\lambda)$) if $\nu(\lambda) \leq 2^{-\lambda}$. $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$ 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 integer. We define the rings $R = \mathbb{Z}[X]/(X^d+1)$ and $R_q = \mathbb{Z}_q[X]/(X^d+1)$ where $d > 1$ is a power of 2. We use bold-face lower-case letters such as $\boldsymbol{v}$ and bold-face capital letters such as $\boldsymbol{A}$ to denote column vectors and matrices, respectively. Commitments are denoted by capital letters such as C (even though they may be vectors). $(\boldsymbol{v}, \boldsymbol{w})$ denotes appending the vector $\boldsymbol{w}$ to the vector $\boldsymbol{v}$. For a vector $\boldsymbol{v} = (v_0, \ldots, v_{n-1})$, the norms are defined as $\|\boldsymbol{v}\| = \sqrt{\sum_{i=0}^{n-1} v_i^2}$, $\|\boldsymbol{v}\|_\infty = \max_i |v_i|$ and $\|\boldsymbol{v}\|_1 = \sum_{i=0}^{n-1} |v_i|$. For a polynomial p, the corresponding norms are defined analogously on the coefficient vector of p. For a vector $\boldsymbol{p} = (p_0, \ldots, p_{s-1})$ of polynomials, $\|\boldsymbol{p}\| = \sqrt{\sum_{i=0}^{s-1} \|p_i\|^2}$, $\|\boldsymbol{p}\|_1 = \sum_{i=0}^{s-1} \|p_i\|_1$, $\|\boldsymbol{p}\|_\infty = \max_i \|p_i\|_\infty$ and $\mathsf{HW}(\boldsymbol{p})$ is the Hamming weight of the (whole) coefficient vector of $\boldsymbol{p}$. $\mathcal{U}(S)$ denotes the uniform distribution on a set S. We use $a \leftarrow \mathcal{S}$ to denote sampling a from a distribution $\mathcal{S}$, or uniformly sampling from a set $\mathcal{S}$. We write $\mathcal{S}^{md}$ to indicate that a total of md coefficients are sampled to generate m polynomials of degree d. Unless specified otherwise, logarithms are base 2, and $[a, b] = \{a, \ldots, b\}$, $[a, b) = \{a, \ldots, b-1\}$ for $a < b \in \mathbb{Z}$.

The next lemma summarizes some results regarding different norms.

Lemma 8. *For any $f, g \in R = \mathbb{Z}[X]/(X^d + 1)$, we have the following relations*

1. $\|f\| \leq \sqrt{d} \cdot \|f\|_\infty$ *and* $\|f\| \leq \|f\|_1 \leq \sqrt{d}\,\|f\|$,
2. $\|f \cdot g\| \leq \sqrt{d} \cdot \|f\| \cdot \|g\|$,
3. $\|f \cdot g\|_\infty \leq \|f\| \cdot \|g\|$,
4. $\|f \cdot g\|_\infty \leq \|f\|_1 \cdot \|g\|_\infty$,
5. $\|\prod_{i=1}^n f_i\|_\infty \leq \left(\prod_{i=1}^{n-1} \|f_i\|_1\right) \cdot \|f_n\|_\infty$ *where* $f_i \in R$ *for all* $1 \leq i \leq n$.

Proof (Lemma 8). The first 4 relations are standard and we only provide a proof for the last one. If $n = 2$, the result is clear by the forth relation. Assume that the result holds for all $s < n$, and we want to show that it holds for $n > 2$.

$$\|\prod_{i=1}^{n} f_i\|_\infty \leq \|f_1\|_1 \cdot \|\prod_{i=2}^{n} f_i\|_\infty \leq \left(\prod_{i=1}^{n-1} \|f_i\|_1\right) \cdot \|f_n\|_\infty,$$

where the first inequality holds due to the forth relation and the second one follows by the inductive assumption. □

B.2 Discrete Normal Distribution and Sum of Discrete Normal Variables

The discrete normal distribution is defined formally as below.

Definition 7. *Let $\rho_{\mathbf{c},\sigma}^s(\boldsymbol{x}) = \left(1/(\sqrt{2\pi}\sigma)\right)^s e^{-\frac{\|\boldsymbol{x}-\mathbf{c}\|^2}{2\sigma^2}}$ be the continuous normal distribution centered at $\boldsymbol{c}$ with standard deviation σ over $\mathbb{R}^s$ for $s \geq 1$. The discrete normal distribution over $\mathbb{Z}^s$ centered at $\boldsymbol{c}$ with standard deviation σ is defined as $D_{\mathbf{c},\sigma}^s(\boldsymbol{x}) = \rho_{\mathbf{c},\sigma}^s(\boldsymbol{x})/\rho_{\mathbf{c},\sigma}^s(\mathbb{Z}^s)$ where $\rho_{\mathbf{c},\sigma}^s(\mathbb{Z}^s) = \sum_{\boldsymbol{z}\in\mathbb{Z}^s} \rho_{\mathbf{c},\sigma}^s(\boldsymbol{z})$.*

In our protocols, we sometimes deal with sum of independent vectors from discrete normal distribution. To study the behaviour of such sums, we make use the following lemma.

Lemma 9 (Special case of [32, Theorem 3.3]). *Let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s$ be independent vectors with distribution D_σ^d for $d \geq 1$. If $\sigma \geq \eta(\mathbb{Z}^d)/\sqrt{\pi}$ for the smoothing parameter $\eta(\mathbb{Z}^d)$ of $\mathbb{Z}^d$, then the distribution of $\boldsymbol{z} := \boldsymbol{y}_1 + \cdots + \boldsymbol{y}_s$ is statistically close to $D_{\sigma\sqrt{s}}^d$.*

The smoothing parameter (defined in [33]) $\eta(\mathbb{Z}^d)$ of $\mathbb{Z}^d$ can be easily upper-bounded using Lemma 3.3 of [33]. In particular, $\eta(\mathbb{Z}^d) \leq 6$ for $d \leq 2^{13}$ (when taking 2^{-128} as a negligible function). We do not go into much technical details here and refer to [33]. For our purposes, the important point is that since the standard deviations in our protocols are always much larger than 6, discrete normal variables behave as its continuous counterpart when multiple samples are summed over. The lemma below summarizes the "tail-cut" bounds on discrete normal distribution.

Lemma 10 ([29, Lemma 4.4]).
– *For any* $c > 0$, $\Pr[|z| > c\sigma \ : \ z \leftarrow D_\sigma] \leq 2e^{-c^2/2}$,
– *For any* $c > 1$, $\Pr[\|\boldsymbol{z}\| > c\sigma\sqrt{s} \ : \ \boldsymbol{z} \leftarrow D_\sigma^s] < c^s e^{\frac{1-c^2}{2}s}$.
Therefore, we have, in particular, $\Pr[|z| > 12\sigma \ : \ z \leftarrow D_\sigma] < 2^{-100}$, *and* $\Pr[\|\boldsymbol{z}\| > 2\sigma\sqrt{s} \ : \ \boldsymbol{z} \leftarrow D_\sigma^s] < 2^{-128}$ *if* $s \geq 110$.

Lemma 11 ([29]). *Let h be a probability distribution over $V \subseteq \mathbb{Z}^s$ $(s \geq 1)$ where all the elements have norm less than T. Let $\boldsymbol{c} \leftarrow h$ and $\phi > 0$, and consider the algorithm $\mathcal{F}$ that samples $\boldsymbol{y} \leftarrow D_\sigma^s$ and outputs Rej($\boldsymbol{z}, \boldsymbol{c}, \phi, T$) (Algorithm 1) for $\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{c}$. The probability that $\mathcal{F}$ outputs 1 is within 2^{-100} of $1/\mu(\phi)$ for $\mu(\phi) = e^{12/\phi + 1/(2\phi^2)}$, and conditioned on the output being 1, the statistical distance between distribution of $\boldsymbol{z}$ and D_σ^s is at most 2^{-100}.*

B.3 Commitments: Definitions and More Discussion

A commitment scheme consist of three algorithms as below.

$\mathsf{CKeygen}$ is a PPT algorithm that, on input security parameter 1^λ, outputs the public parameters pp together with the specifications of a message space $\mathcal{S}_M$, a randomness space $\mathcal{S}_R$ and a commitment space $\mathcal{S}_C$.

Commit is a PPT algorithm that, on input public parameters pp and a message $M \in \mathcal{S}_M$, outputs a commitment $C \in \mathcal{S}_C$.

Open is a deterministic polynomial-time algorithm that, on input public parameters pp, a tuple $(M, r; C) \in \mathcal{S}_M \times \mathcal{S}_R \times \mathcal{S}_C$, outputs a bit b, indicating 'accept' when $b = 1$, and 'reject' otherwise.

Some lattice-based commitment schemes, as in our case, have an additional input y, called the relaxation factor, to the Open algorithm, which is also parametrized by a norm bound γ_{com}. *Computational hiding* and *computational strong γ_{com}-binding* properties are defined, for all PPT algorithms $\mathcal{A}$, respectively as

$$\Pr\left[\begin{matrix} pp \leftarrow \mathsf{CKeygen}(1^\lambda); (M_0, M_1) \leftarrow \mathcal{A}^{\mathsf{CKeygen}(pp)} \\ b \leftarrow \{0,1\};\ C \leftarrow \mathsf{Commit}_{pp}(M_b) \end{matrix} : \mathcal{A}(C) = b\right] \approx 1/2, \quad \text{and}$$

$$\Pr\left[\begin{matrix} pp \leftarrow \mathsf{CKeygen}(1^\lambda); \\ (C, \boldsymbol{t}_0, \boldsymbol{t}_1) \leftarrow \mathcal{A}(pp) \end{matrix} : \begin{matrix} (M_0, r_0) \neq (M_1, r_1) \wedge \\ \mathsf{Open}_{pp}(C, \boldsymbol{t}_0) = \mathsf{Open}_{pp}(C, \boldsymbol{t}_1) = 1 \end{matrix}\right] \approx 0,$$

where $\boldsymbol{t}_i = (y_i, M_i, r_i)$ for $i = 0, 1$ and the norm bound parameter in Open is γ_{com}. In the case of *computational binding*, the requirement $(M_0, r_0) \neq (M_1, r_1)$ in strong binding is replaced with $M_0 \neq M_1$.

The commitment schemes UMC and HMC offer different tradeoffs. UMC allows one to commit to messages of unbounded length but the commitment vector dimension increases linearly with the message vector dimension in a commitment. For HMC, on the other hand, one can only commit to messages of bounded length (when binding is based on M-SIS) but the height of the commitment vector is independent of the message vector dimension (thus, one can commit to big chunks of messages without significantly increasing the commitment vector size).

For HMC, it is easy to see that when $(y, \boldsymbol{m}_0, \boldsymbol{r}_0)$ and $(y, \boldsymbol{m}_1, \boldsymbol{r}_1)$ gives a binding collision pair such that $(\boldsymbol{m}_0, \boldsymbol{r}_0) \neq (\boldsymbol{m}_1, \boldsymbol{r}_1)$, we get $\boldsymbol{G} \cdot (\boldsymbol{m}_0, \boldsymbol{r}_0) = \boldsymbol{G} \cdot (\boldsymbol{m}_1, \boldsymbol{r}_1)$, which implies $\boldsymbol{G} \cdot (\boldsymbol{m}_0 - \boldsymbol{m}_1, \boldsymbol{r}_0 - \boldsymbol{r}_1) = 0$. Since $(\boldsymbol{m}_0, \boldsymbol{r}_0), (\boldsymbol{m}_1, \boldsymbol{r}_1) \leq \gamma_{\text{com}}$, $(\boldsymbol{m}_0 - \boldsymbol{m}_1, \boldsymbol{r}_0 - \boldsymbol{r}_1)$ gives a solution to $\text{M-SIS}_{n,m+v,q,2\gamma_{\text{com}}}$. Also, observe that $\boldsymbol{G}_r \cdot \boldsymbol{r} = \boldsymbol{r}_0 + \boldsymbol{G}'_r \cdot \boldsymbol{r}_1$ where $\boldsymbol{r} = (\boldsymbol{r}_0, \boldsymbol{r}_1)$. This gives n M-LWE samples for $\chi = \mathcal{U}(\{-\mathcal{B}, \ldots, \mathcal{B}\}^d)$ where the secret key $\boldsymbol{s} = \boldsymbol{r}_1$ is in R_q^{m-n}. As a result, the commitment scheme defined above is also computationally hiding if $\text{M-LWE}_{m-n,n,q,\mathcal{U}(\{-\mathcal{B},\ldots,\mathcal{B}\}^d)}$ problem is hard. See also Lemma 6 and Lemma 9 of [6] for the hiding property.

For HMC, we see that M-SIS security increases with n whereas M-LWE security increases with $m - n$. On the other hand, for UMC, M-SIS security increases with n while M-LWE security increases with $m - n - v$. We balance these two security aspects when setting the concrete parameters. Moreover, for a scheme using UMC, the parameter setting is done using the results of Lemma 2. Similarly, for a scheme using HMC, the parameter setting is done using the results of Lemma 1. To make sure that the commitment scheme is γ-binding, we use the methodology in [34] and set the parameters so that

$$\min\{q, 2^{2\sqrt{nd \log q \log \delta}}\} > 2\gamma \tag{31}$$

is satisfied for a root Hermite factor δ.

C More Discussion on the Range Proof

Practical aspects of the range proof. Let $N = \beta^k$, and assume we want to prove knowledge of an opening ℓ of V such that $V = \text{Com}(\ell)$ and $\ell \in [0, N)$. The generic way for such a range proof works as follows. The prover publishes the commitments $\text{Com}(\ell_j)$ to the digit ℓ_j's of ℓ, and proves that each digit is in $\{0, \ldots, \beta - 1\}$, namely a *set membership proof*. The last step is then to use the homomorphic properties of the commitment to check that these digits construct V, i.e., $\text{Com}(\ell) = \sum_j \beta^j \text{Com}(\ell_j)$. Such a proof involves sending at least k commitments and k masked randomnesses. The β value needs to be small, otherwise the set membership proof becomes cumbersome (especially in the case of lattice-based proofs), and in general $\beta = 2$ is set and thus $k = \log N$. Doing a range proof for ψ values, at best, would multiply the number of commitments by ψ. Therefore, the overall cost would be proportional to at least $\psi \log N nd \log q$ bits since a commitment is of size at least $nd \log q$ bits.

In our proof, on the other hand, the number of commitments and randomnesses communicated is constant. More precisely, always 2 commitments[14] and 3 randomnesses are sent, but their dimensions may vary. In total, the range proof length is $(2(n+v)d\log q + \psi \log N(d/s)B_f + 3mdB_z)$ bits where $B_f, B_z < \log q$ are the bit-lengths of $f_j^{(i)}$'s and $\boldsymbol{z}$'s, respectively. We have $v \approx \psi$ and $m \approx 2n+\psi$. Therefore, the overall proof length growth is slower in comparison to the generic approach. In Table 3, we provide a comparison of our CRT-packed range proof with an idealized scheme and one that uses the "norm-optimal" challenges with infinity norm 1 [31]. We can see easily that our range proof provides much better computational efficiency *without* any significant compromise in communication size. We also have the invertibility of the challenge differences in $\Delta\mathcal{C}_{w,p}^{d/s}$. In Tables 5, 6 and 7, we provide the full parameter setting of the compared range proof methods.

Extension to arbitrary ranges. We assumed that a range is of the form $[0, N)$ for $N = 2^k$. Our range proof can be extended to work for arbitrary ranges using standard techniques as follows. For simplicity, let us assume that $\psi = 1$, i.e., V is a commitment to a single value ℓ. Our discussion easily generalizes to the case of committing to a set of values. Suppose that we want to prove $\ell \in [a, b)$ for $b > a + 1$. First, using $V' = V - \mathrm{Com}_{ck}(a; 0)$ in the protocol proves that $\ell - a \in [0, N)$, i.e., $\ell \in [a, N + a)$ (this implies that we can "shift" the range in any suitable way). Now, if $b - a$ can be set so that $b - a = N = 2^k$, then we are done. Otherwise, we set $2^k = N > b - a$, and run another range proof for $V'' = \mathrm{Com}_{ck}(b; 0) - V$. This proves that $b - \ell \in [0, N)$, i.e., $\ell \in [b - N, b)$. As a result, ℓ must be in the intersection of $[a, N + a)$ and $[b - N, b)$, i.e., $\ell \in [a, b)$. Note that the proved relations are relaxed as in $\mathcal{R}'_{\text{range}}$, but they indeed work in this sense. Suppose that we have a range proof for $V_1 = V - \mathrm{Com}_{ck}(a; 0)$ for the range $[0, N_1)$ and another range proof for $V_2 = V - \mathrm{Com}_{ck}(b; 0)$ for the range $[0, N_2)$. Then, these prove knowledge of $(\bar{x}_1, \ell_1, \hat{\boldsymbol{r}}_1)$ and $(\bar{x}_2, \ell_2, \hat{\boldsymbol{r}}_2)$ such that

$$\mathrm{Com}_{ck}(\bar{x}_1\ell_1; \hat{\boldsymbol{r}}_1) = \bar{x}_1 V_1 = \bar{x}_1(V - \mathrm{Com}_{ck}(a; 0)) \ \wedge\ \ell_1 \in [0, N_1),$$
$$\mathrm{Com}_{ck}(\bar{x}_2\ell_2; \hat{\boldsymbol{r}}_2) = \bar{x}_2 V_2 = \bar{x}_2(V - \mathrm{Com}_{ck}(b; 0)) \ \wedge\ \ell_2 \in [0, N_2).$$

Therefore, we have

$$\bar{x}_1 V = \mathrm{Com}_{ck}(\bar{x}_1(\ell_1 + a); \hat{\boldsymbol{r}}_1) \quad \Longrightarrow \bar{x}_2\bar{x}_1 V = \mathrm{Com}_{ck}(\bar{x}_2\bar{x}_1(\ell_1 + a); \hat{\boldsymbol{r}}_1),$$
$$\bar{x}_2 V = \mathrm{Com}_{ck}(\bar{x}_2(\ell_2 + b); \hat{\boldsymbol{r}}_2) \quad \Longrightarrow \bar{x}_1\bar{x}_2 V = \mathrm{Com}_{ck}(\bar{x}_1\bar{x}_2(\ell_2 + b); \hat{\boldsymbol{r}}_2).$$

By the binding property on $\bar{x}_1\bar{x}_2 V$, the committed messages $\bar{x}_2\bar{x}_1(\ell_1 + a)$ and $\bar{x}_1\bar{x}_2(\ell_2 + b)$ must be the same. That is, $\bar{x}_2\bar{x}_1(\ell_1 + a) = \bar{x}_1\bar{x}_2(\ell_2 + b)$, which implies that $\ell := \ell_1 + a = \ell_2 + b$ since $\bar{x}_1, \bar{x}_2 \in \Delta\mathcal{C}_{w,p}^{d/s}$ and thus are invertible when R_q splits into exactly s fields. Hence, $\ell \in [a, N_1 + a) \cap [b, N_2 + b)$, which concludes that the combination of the proofs behave in an expected manner.

[14] Note that this happens in the non-interactive case where 5 commitments reduce to 2 commitments. It is standard to exclude from the proof output the commitments $(A, D, E$ in our case) that are uniquely determined by the remaining components.

Table 5: The parameter setting of our range proof on $[0, 2^{\log N} - 1]$ with CRT-packing for 128-bit security. $\mathcal{B} = 1$ for all cases. The root Hermite factor for LWE varies in between 1.00399 and 1.00493, and for SIS is ≈ 1.0035. M-SIS and M-LWE dimension parameters are nd and $(m-n-v)d$, respectively, for Tables 5, 6 and 7. Here it is harder to balance the security as the dimension parameters increase as multiples of 512.

	$\log N$	32	32	32	64	64	64
	ψ	1	5	10	1	5	10
Range Proof Size (KB)		58	130	202	93	216	319
module rank for M-SIS	n	2	3	3	2	4	4
com. randomness vector dimension	m	7	12	17	9	14	19
poly. ring dimension	d	512	512	512	512	512	512
num. of slots in CRT	s	16	32	32	32	64	64
com. message vector dimension	v	2	5	10	2	5	10
modulus	q	$\approx 2^{43}$	$\approx 2^{46}$	$\approx 2^{47}$	$\approx 2^{67}$	$\approx 2^{66}$	$\approx 2^{67}$
Hamming weight of challenges	w	32	16	16	16	8	8
max. abs. coefficient of challenges	p	128	32768	32768	32768	2^{31}	2^{31}

Table 6: The parameter setting of "Ideal w/o CRT" range proof on $[0, 2^{\log N} - 1]$ for 128-bit security. $\mathcal{B} = 1$ and $v = \psi \log N$ for all cases. The root Hermite factor for SIS and LWE are ≈ 1.0045. There is no additional assumption on the parameters to make sure that the binary proof works. The only assumption is $q \geq 2^{\log N}$ and the parameters are set to make UMC hiding and γ_{range}-binding.

$\log N$	32	32	32	64	64	64
ψ	1	5	10	1	5	10
Range Proof Size (KB)	52	113	180	86	201	302
n	34	89	92	52	210	213
m	107	345	508	275	847	1170
d	32	16	16	16	8	8
q	$\approx 2^{32}$	$\approx 2^{38}$	$\approx 2^{38}$	$\approx 2^{64}$	$\approx 2^{64}$	$\approx 2^{64}$
w	32	16	16	16	8	8
p	128	32768	32768	32768	2^{31}	2^{31}

Table 7: The parameter setting of range proof on $[0, 2^{\log N} - 1]$ using "norm-optimal" challenges with infinity norm 1 for 128-bit security. $\mathcal{B} = 1$ and $v = \psi \log N$ for all cases. The root Hermite factor for LWE is ≈ 1.0045. For SIS, the root Hermite factor is in between 1.0030 and 1.0045. When the invertibility results of [31] are used, q may actually need to be larger to ensure that $\hat{b}(y - \hat{b}) = 0$ in R_q implies $\hat{b} \in \{0, y\}$. But, we ignore this in favor of the method.

$\log N$	32	32	32	64	64	64
ψ	1	5	10	1	5	10
Range Proof Size (KB)	161	745	1484	443	2131	4274
n	3	4	2	2	4	3
m	40	169	332	76	332	653
d	256	256	256	256	256	256
q	$\approx 2^{32}$	$\approx 2^{32}$	$\approx 2^{32}$	$\approx 2^{64}$	$\approx 2^{64}$	$\approx 2^{64}$
w	60	60	60	60	60	60
p	1	1	1	1	1	1

D Full Description of Relaxed Binary Proof

$\mathcal{P}_{\text{bin}}((ck, B), (\boldsymbol{b}; \boldsymbol{r}))$ $\qquad\qquad$ $\mathcal{V}_{\text{bin}}(ck, B)$

1: $a_{0,1}, \ldots, a_{k-1,\beta-1} \leftarrow D^{d}_{\phi_1 T_1}$ for $T_1 = p\sqrt{kw}$

2: $\boldsymbol{r}_c \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$

3: $\boldsymbol{r}_a, \boldsymbol{r}_d \leftarrow D^{md}_{\phi_2 T_2}$ for $T_2 = pw\mathcal{B}\sqrt{2md}$

4: **for** $j = 0, \ldots, k-1$ **do**

5: $\quad a_{j,0} = -\sum_{i=1}^{\beta-1} a_{j,i}$

6: $A = \text{Com}_{ck}(\{a_{j,i}\}_{j,i=0}^{k-1,\beta-1}; \boldsymbol{r}_a)$

7: $C = \text{Com}_{ck}(\{a_{j,i}(1-2b_{j,i})\}_{j,i=0}^{k-1,\beta-1}; \boldsymbol{r}_c)$

8: $D = \text{Com}_{ck}(\{-(a_{j,i})^2\}_{j,i=0}^{k-1,\beta-1}; \boldsymbol{r}_d)$

$\xrightarrow{\quad A, C, D \quad}$

$\xleftarrow{\quad x \quad}$ $\qquad$ $x \leftarrow \mathcal{C}^{d}_{w,p}$

9: **for** $j \in [0, k), i \in [1, \beta)$ **do**

10: $\quad f_{j,i} = x \cdot b_{j,i} + a_{j,i}$

$\boldsymbol{f}_1 := (f_{0,1}, \ldots, f_{k-1,\beta-1}), \boldsymbol{b}_1 := (b_{0,1}, \ldots, b_{k-1,\beta-1})$

11: $\text{Rej}(\boldsymbol{f}_1, x\boldsymbol{b}_1, \phi_1, T_1)$

12: $\boldsymbol{z}_b = x \cdot \boldsymbol{r} + \boldsymbol{r}_a$

13: $\boldsymbol{z}_c = x \cdot \boldsymbol{r}_c + \boldsymbol{r}_d$

14: $\text{Rej}((\boldsymbol{z}_b, \boldsymbol{z}_c), x(\boldsymbol{r}, \boldsymbol{r}_c), \phi_2, T_2)$

If aborted, return $\perp$. $\qquad$ $\xrightarrow{\quad \boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z}_c \quad}$

1: **for** $j = 0, \ldots, k-1$ **do**

2: $\quad f_{j,0} = x - \sum_{i=1}^{\beta-1} f_{j,i}$

3: $\|f_{j,i}\| \overset{?}{\leq} 2\phi_1 T_1 \sqrt{d} \;\; \forall j, \forall i \neq 0$

4: $\|f_{j,0}\| \overset{?}{\leq} 2\phi_1 T_1 \sqrt{\beta d} \;\; \forall j$

5: $\|\boldsymbol{z}_b\|, \|\boldsymbol{z}_c\| \overset{?}{\leq} 2\phi_2 T_2 \sqrt{md}$

$\quad \boldsymbol{f} := \{f_{j,i}\}_{j,i=0}^{k-1,\beta-1}$

$\quad \boldsymbol{g} := \{f_{j,i}(x - f_{j,i})\}_{j,i=0}^{k-1,\beta-1}$

6: $xB + A \overset{?}{=} \text{Com}_{ck}(\boldsymbol{f}; \boldsymbol{z}_b)$

7: $xC + D \overset{?}{=} \text{Com}_{ck}(\boldsymbol{g}; \boldsymbol{z}_c)$

Protocol 4: Σ-protocol for $\mathcal{R}_{\text{bin}}$ and $\mathcal{R}'_{\text{bin}}$.

E Relaxed Set Membership Proof

Suppose the prover has a commitment C and wants to prove knowledge of a message opening $\boldsymbol{m}$ of C such that $\boldsymbol{m} \in \mathcal{S} = \{\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{N-1}\}$. Such a *set membership proof* can be constructed from one-out-of-many proof as in [22]. The protocol works as follows. Both the prover and the verifier set $P_i = C - \mathrm{Com}_{ck}(\boldsymbol{v}_i; 0)$ in the one-out-of-many proof, and run that protocol. This proves knowledge of $(y, \ell, \hat{\boldsymbol{r}})$ such that $yP_\ell = \mathrm{Com}_{ck}(\boldsymbol{0}; \hat{\boldsymbol{r}})$, which gives

$$
\begin{aligned}
& yP_\ell = y(C - \mathrm{Com}_{ck}(\boldsymbol{v}_\ell; 0)) = \mathrm{Com}_{ck}(\boldsymbol{0}; \hat{\boldsymbol{r}}), \\
\Longrightarrow \quad & yC = \mathrm{Com}_{ck}(\boldsymbol{0}; \hat{\boldsymbol{r}}) + y\mathrm{Com}_{ck}(\boldsymbol{v}_\ell; 0) = \mathrm{Com}_{ck}(y\boldsymbol{v}_\ell; \hat{\boldsymbol{r}}).
\end{aligned} \tag{32}
$$

As a result, our set membership proof convinces the verifier of the following statement, for some $\hat{\mathcal{T}} \in \mathbb{R}^+$,

$$
\mathcal{R}'_{\mathrm{mem}}(\hat{\mathcal{T}}) = \left\{ \begin{matrix} ((ck, (\boldsymbol{v}_0, \ldots, \boldsymbol{v}_{N-1}), C), (y, \ell, \hat{\boldsymbol{r}})) : \ell \in [0, N) \ \wedge \ \|\hat{\boldsymbol{r}}\| \leq \hat{\mathcal{T}} \ \wedge \\ yC = \mathrm{Com}_{ck}(y\boldsymbol{v}_\ell; \hat{\boldsymbol{r}}) \ \wedge y \text{ is a product of elements in } \Delta\mathcal{C}^d_{w,p} \end{matrix} \right\}. \tag{33}
$$

F Additional Material about Ring Signature

Ring signatures, introduced by Rivest, Shamir and Tauman-Kalai [35], offer a way for anonymous signature generation in that the signer's identity is hidden within a set of identities. That is, the outside world only knows that one of the *ring* members generated the signature, unable to determine which one exactly. The formal definitions of ring signatures were established in [8], and we use variants of those.

F.1 Formal Definitions

A ring signature consists of four algorithms (**RSetup**, **RKeygen**, **RSign**, **RVerify**) defined as follows.

- $pp \leftarrow \mathbf{RSetup}(1^\lambda)$: On input a security parameter λ, outputs the public parameters pp, which are available to everyone.
- $(pk, sk) \leftarrow \mathbf{RKeygen}(pp)$: Given pp, generates a public-secret key pair (pk, sk).
- $\sigma \leftarrow \mathbf{RSign}_{pp,sk}(M, \boldsymbol{L})$: On input a message M and a ring $\boldsymbol{L}$ of public keys and for a secret key sk generated by **RKeygen** and its corresponding public key $pk \in \boldsymbol{L}$, outputs a signature σ on M with respect to $\boldsymbol{L}$.
- $\{0, 1\} \leftarrow \mathbf{RVerify}_{pp}(M, \boldsymbol{L}, \sigma)$: On input a purported signature σ, a message M and a ring $\boldsymbol{L}$, checks if σ is a valid signature on M with respect to $\boldsymbol{L}$. Outputs 1 when it is valid, and outputs 0 otherwise.

Definition 8 (Correctness). *A ring signature scheme has statistical correctness if the following holds for any $pp \leftarrow \mathbf{RSetup}(1^\lambda)$, any $(pk, sk) \leftarrow \mathbf{RKeygen}(pp)$, any $\boldsymbol{L}$ with $pk \in \boldsymbol{L}$, and any $M \in \{0, 1\}^*$,*

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

Definition 9 (Anonymity). *A ring signature scheme has statistical anonymity if the following holds for any PPT adversary $\mathcal{A}$*

$$\Pr\left[\begin{matrix} pp \leftarrow \mathbf{RSetup}(1^\lambda); (M, j_0, j_1, \boldsymbol{L}) \leftarrow \mathcal{A}^{\mathbf{RKeygen}(pp)} \\ b \leftarrow \{0,1\};\ \sigma \leftarrow \mathbf{RSign}_{pp,sk_{j_b}}(M, \boldsymbol{L});\ b' \leftarrow \mathcal{A}(\sigma) \end{matrix} : b' = b\right] = \frac{1}{2} + \mathrm{negl}(\lambda),$$

where $(pk_{j_0}, sk_{j_0}), (pk_{j_1}, sk_{j_1}) \leftarrow \mathbf{RKeygen}(pp)$ *and* $pk_{j_0}, pk_{j_1} \in \boldsymbol{L}$.

Definition 10 (Unforgeability w.r.t. insider corruption). *A ring signature scheme is unforgeable with respect to insider corruption if the following holds for all PPT adversary $\mathcal{A}$*

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

where

- PKGen*: on the i-th query, runs* $(pk_i, sk_i) \leftarrow \mathbf{RKeygen}(pp)$ *and returns* pk_i.
- $\mathrm{Sign}(i, M, \boldsymbol{L})$*: returns* $\sigma \leftarrow \mathbf{RSign}_{pp,sk_i}(M, \boldsymbol{L})$ *if* $(pk_i, sk_i) \leftarrow$ PKGen *and* $pk_i \in \boldsymbol{L}$. *Otherwise, returns* $\perp$.
- $\mathrm{Corrupt}(i)$*: returns* sk_i *if* $(pk_i, sk_i) \leftarrow$ PKGen. *Otherwise, returns* $\perp$.
- *For $\mathcal{A}$'s output* $(M, \boldsymbol{L}, \sigma)$, $\mathrm{Sign}(\cdot, M, \boldsymbol{L})$ *has never been queried, all public keys in* $\boldsymbol{L}$ *are generated by* PKGen *and no public key in* $\boldsymbol{L}$ *is corrupted.*

F.2 Construction

Let $N = \beta^k$ for $2 \leq \beta \leq N$, $k \geq 1$ and $q, \mathcal{B}$ with $\mathcal{B} < q$ be positive integers. Let $\mathsf{CMT} = (A, B, C, D, E_0, \ldots, E_{k-1})$ and $\mathsf{RSP} = (\{f_{j,i}\}_{j=0,i=1}^{k-1,\beta-1}, \boldsymbol{z}, \boldsymbol{z}_b, \boldsymbol{z}_c)$ be the initial commitment and prover's response from Protocol 3, respectively. Also, denote $\mathcal{C}^d_{w,p}$ as the challenge space defined in (12) and $\mathsf{CMT}^* = (B, C, E_1, \ldots, E_{k-1})$ as a subset of CMT. Our ring signature uses Fiat-Shamir heuristic [20] to make Protocol 3 non-interactive and is defined as follows.

- $\mathbf{RSetup}(1^\lambda)$: Run $\boldsymbol{G} \leftarrow \mathsf{CKeygen}(1^\lambda)$ of HMC and pick a hash function $H : \{0,1\}^* \rightarrow \mathcal{C}$. Return the commitment key $ck = \boldsymbol{G}$ and H as $pp = (ck, H)$.
- $\mathbf{RKeygen}(pp)$: Sample $\boldsymbol{r} \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$ and compute $P = \mathrm{Com}_{ck}(\boldsymbol{0}; \boldsymbol{r})$ for the all-zero vector $\boldsymbol{0}$. Return $(pk, sk) = (P, \boldsymbol{r})$.
- $\mathbf{RSign}_{pp,sk}(M, \boldsymbol{L})$: Parse $\boldsymbol{L} = (P_0, \ldots, P_{N-1})$ with $P_\ell = \mathrm{Com}_{ck}(\boldsymbol{0}; sk)$ for $\ell \in [0, N)$. Proceed as follows.
  1. Generate CMT by running $\mathcal{P}_{1/\mathrm{N}}((ck, (P_0, \ldots, P_{N-1})), (\ell, sk))$.
  2. Compute $x = H(ck, M, \boldsymbol{L}, \mathsf{CMT})$.
  3. Compute RSP by running $\mathcal{P}_{1/\mathrm{N}}(x)$ with CMT.
  4. If $\mathsf{RSP} = \perp$, go to Step 1.
  5. Otherwise, return $\sigma = (\mathsf{CMT}^*, x, \mathsf{RSP})$.
- $\mathbf{RVerify}_{pp}(M, \boldsymbol{L}, \sigma)$: Parse $\boldsymbol{L} = (P_0, \ldots, P_{N-1})$, $\sigma = (\mathsf{CMT}^*, x, \mathsf{RSP})$ and $\mathsf{CMT}^* = (B, C, E_1, \ldots, E_{k-1})$. Continue as follows.
  1. Compute A, D and E_0 so that Steps 6 and 7 in Protocol 4, and Step 3 in Protocol 3 are satisfied.
  2. Set $\mathsf{CMT} = (A, B, C, D, E_0, \ldots, E_{k-1})$.

3. If $x \neq H(ck, M, \boldsymbol{L}, \mathsf{CMT})$, return 0.
4. Return the output of $\mathcal{V}_{1/\mathrm{N}}(ck, (P_0, \ldots, P_{N-1}), (\mathsf{CMT}, x, \mathsf{RSP}))$.

Correctness and anonymity properties of the ring signature follows easily from the completeness and SHVZK of Protocol 3, respectively. In particular, for $\phi_1 = \phi_2 = 15$, we have $1/(\mu(\phi_1)\mu(\phi_2)) > 1/5$. Therefore, an accepting transcript is produced by Protocol 3 with probability at least 1/5. Thus, the expected number of iterations in **RSign** is 5, which is $O(1)$, for $\phi_1 = \phi_2 = 15$. Unforgeability of the ring signature is stated in Theorem 4.

Theorem 4. *If HMC is hiding and γ-binding where $\gamma = \max\{\gamma_{\mathrm{bin}}, \gamma_{1/\mathrm{N}}\}$ for γ_{bin} and $\gamma_{1/\mathrm{N}}$ defined in Theorem 2 and Theorem 3, respectively, then the ring signature scheme described by* $(\mathbf{RSetup}, \mathbf{RKeygen}, \mathbf{RSign}, \mathbf{RVerify})$ *is unforgeable with respect to insider corruption in the random oracle model.*

Proof (Sketch for Theorem 4). The idea for the proof is similar to that in [19], but the challenge space is exponentially large in our case and no parallel repetition of the underlying protocol is required in the ring signature. Assume that there exists a PPT adversary $\mathcal{F}$ that can forge a ring signature in polynomial time and non-negligible probability. This gives rise to an adversary $\mathcal{A}$ which can break the binding property of the commitment scheme, and thus solve M-SIS problem.

$\mathcal{A}$ creates an invalid public key pk_ℓ such that $pk_\ell = \mathrm{Com}_{ck}(1, 0, \ldots, 0; \boldsymbol{r})$ for $\boldsymbol{r} \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$, which cannot be detected by $\mathcal{F}$ due to the hiding property of the commitment scheme. Then, it runs $\mathcal{F}$ until $k+1$ forgeries in total with distinct challenges are obtained where CMT^* part of the signature (and thus CMT) is the same for all forgeries and pk_ℓ is not corrupted. This can be done in polynomial time using the Forking Lemma in [13]. Then, $\mathcal{A}$ runs the extractor of Protocol 3 to get an *exact* valid opening $(\mathbf{0}; \boldsymbol{s})$ of $y \cdot pk_i$ for some public key pk_i where y is the relaxation factor in Definition 6. With $1/\mathrm{poly}(\lambda)$ probability, $i = \ell$ as $\mathcal{F}$ can only make polynomially many queries to PKGen. As a result, $((y, 0, \ldots, 0; y\boldsymbol{r}), (\mathbf{0}; \boldsymbol{s}))$ is binding collision pair for the commitment scheme since $(y, 0, \ldots, 0) \neq \mathbf{0}$. □

F.3 Concrete parameters

Firstly, it is clear from their definitions that M-SIS problem gets harder as β_{SIS} gets smaller, and M-LWE problem gets harder when the error is sampled from a wider distribution (i.e., a distribution with larger standard deviation). Thus, we set our parameters to make the easiest cases hard in practice. We aim for 128-bit "post-quantum" security $\lambda = 128$. We set (d, w, p) so that $|\mathcal{C}^d_{w,p}| > 2^{256}$. Similar to recent lattice-based proposals [6, 30, 17], we too consider $\chi = \mathcal{U}(\{-1, 0, 1\}^d)$ (i.e., $\mathcal{B} = 1$) for M-LWE problem. Currently, this seems to affect the hardness of the problem only when the number of samples is cubic in the overall dimension parameter (i.e., $((m-n)d)^3$ samples are needed in our case) due to the attack in [4]. The number of samples in our work is always significantly smaller than this value. In any case, it is straightforward to increase $\mathcal{B}$, and even change the error distribution χ to a Gaussian distribution, both of which have a minor effect on the parameter setting. We set $\phi_1 = \phi_2 = 15$ so that the acceptance rate of the rejection sampling is more than 1/5.

For the M-SIS security, we follow the methodology from [34], which is commonly used (e.g., [19, 6, 29]), and estimate the security using root Hermite factor δ_{SIS}. To that end, we make sure that the commitment scheme is γ_{bin}-binding and $\gamma_{1/\text{N}}$-binding. For the M-LWE security estimation, we use Albrecht et al.'s estimator [1] under both sieving and enumeration techniques. It shows that a "post-quantum" root Hermite factor δ_{LWE} of around 1.0045 offers a post-quantum security of about 128 bits. Therefore, we set all the parameters so that both δ_{SIS} and δ_{LWE} are smaller than 1.0045. The parameter setting (except for those that have already been fixed) is given in Table 4 for various ring sizes.

F.4 Asymptotic signature length of our ring signature

We neglect the $\log\log$ terms throughout our analysis in this section and work with the challenge space $\mathcal{C}^d_{w,1}$. Security of β_{SIS}-M-SIS in dimension nd and modulus q with $\beta_{\text{SIS}} \approx q$ against BKZ-reduction attack with root Hermite factor δ [34] requires $nd \geq \log q/(4 \log \delta)$. Using the BKZ root Hermite factor from [3] with $\log \delta = \Omega(1/\lambda)$ for security parameter λ, we get $nd = \Omega(\lambda \log q)$. Balancing the same security level for "dual" attack on LWE [3], we get $m = O(n)$ and $md = \Omega(\lambda \log q)$ (recall that the LWE dimension parameter is proportional to $(m - n) \cdot d$). Take $k = O((\log N)/t)$ for a parameter $1 \leq t \leq \log N$ to be optimized. As a result, we have $\beta = N^{1/k} = 2^{\log N/k} = O(2^t)$. To be a one-shot proof, we require $2^w \binom{d}{w} = 2^{O(\lambda)}$. Then, choosing $d = O(\lambda)$ and $w = O(\lambda/\log \lambda)$ is sufficient. Finally, we need $q = O(w^{k^2} kmd^2)$ for M-SIS security[15]. Therefore, $\log q = O(k^2 \log w + \log(md)) = O(((\log N)/t)^2 \log(\lambda/\log \lambda) + \log(\lambda \log q)) = O((\log^2 N \log \lambda)/t^2)$. Using these, we can also find

$$\log(kw) = O(\log(\log N\lambda/(t \log \lambda))) = O(\log \lambda), \text{ and}$$
$$\log(wmd) = O(\log(\lambda^2/(\log \lambda \log q)) = O(\log \lambda)$$

Now, for the signature size $|\sigma|$, we have

$$\begin{aligned}
|\sigma| &= O(knd \log q + k\beta d \log(kw) + md \log(wmd)) \\
&= O(\lambda \log q(\log N \log q/t + \log \lambda) + \lambda \log \lambda \log N 2^t/t) \\
&= O(\lambda \log N(\log^2 q/t + 2^t \log \lambda/t)) \\
&= O(\lambda \log N(\log^4 N \log^2 \lambda/t^5 + 2^t \log \lambda/t)) \\
&= O(\lambda \log \lambda \log N(\log^4 N \log \lambda/t^5 + 2^t/t))
\end{aligned}$$

Taking $t = O(1)$, we can get $|\sigma|$ to be quasi-linear in λ and poly-logarithmic in N, i.e., $|\sigma| = O(\lambda \log^2 \lambda \log^5 N)$. However, if we set $t = (\log N)^{2/3}$, then $\log^4 N \log \lambda/t^5 = \log \lambda(\log N)^{2/3}$ and this term is roughly of the same size as or larger than $2^t/t$ for all practical N values such as $N \leq 2^{30}$. Therefore, we can say that the signature length in practice is proportional to $\lambda \log^2 \lambda \log^c N$ where

[15] This is due to $\gamma_{1/\text{N}}$, which grows asymptotically faster than γ_{bin}.

$c \approx 1.67$ for $N \leq 2^{30}$. Hence, for a fixed security level, the signature length grows slightly faster than logarithmic in N, which also matches the signature length growth for the values provided in Table 4.

F.5 Computational efficiency of our ring signature

To estimate the computational efficiency of the ring signature, we look at that of the one-out-of-many proof in Section 5.2, and consider the efficiency in terms of degree-256 polynomial multiplications in R_q, denoted by `poly256_mult`. We assume that a standard PC has a CPU running at 3 GHz.

Let us take a medium-sized number of ring participants as $N = 2^{10}$. Our ring signature in this case can be instantiated with $(d, w, p) = (256, 60, 1)$, $n = 6$, $m = 15$, $q \approx 2^{56}$, $k = 2$, $\beta = 32$, $\mathcal{B} = 1$, $\phi_1 = \phi_2 = 15$ (signature length is 89 KB). Then, the commitment key dimensions are 6×79 where the first 6×6 part is the identity matrix. Therefore, each commitment computation requires at most $6 \cdot 73 = 438$ `poly256_mult` (this can be further optimized when the committed message is zero or binary-valued).

Offline signing. To compute A, B, C, D, there will be $4 \cdot 438 = 1752$ `poly256_mult` in total. To compute E_j's, we need to perform around $k \cdot n \cdot N + k \cdot n \cdot m = 2 \cdot 6 \cdot 2^{10} + 2 \cdot 6 \cdot 15 = 12468$ `poly256_mult` (the computation of $p_{i,j}$ takes at most 1 `poly256_mult` for $k = 2$). Setting $\phi_1 = \phi_2 = 15$, the expected number of iterations due to rejection sampling will be 5. Therefore, the initial step for the prover is dominated by $5 \cdot (1752 + 12468) \approx 2^{16}$ `poly256_mult` on average. According to the NTT implementation of [16], for polynomial degree 256 and 51-bit modulus, NTT transformation costs about 8000 cycles and pointwise multiplication costs about 1000 cycles. Note that the user public keys and the commitment matrix can be stored in the NTT domain, and thus the number of NTT transformations are much less than that of the pointwise multiplications in our scheme. In particular, the number of NTT transformations is in the order of $k \cdot (m + \beta)$ ($k\beta$ transformations for $a_{0,0}, \ldots, a_{k-1,\beta-1} \in R_q$'s and $(k + 4)m$ transformations for the randomnesses $\boldsymbol{r}_a, \boldsymbol{r}_b, \boldsymbol{r}_c, \boldsymbol{r}_d, \rho_0, \ldots, \rho_{k-1} \in R_q^m$) whereas that of pointwise multiplications is in the order of $kn(N + m)$ (due to Step 8 of the prover). Therefore, the cost of pointwise multiplication is the dominant part in the computational cost of signing, and it can be done in about $1000 \cdot 2^{16} \approx 2^{26}$ cycles, i.e., in about 20 ms on a standard PC. This phase can be easily computed offline.

Note that we need to sample $3md$ coefficients from a wide discrete normal distribution to construct the vectors $\boldsymbol{\rho}_0, \boldsymbol{r}_a, \boldsymbol{r}_d \in R_q^m$, which is repeated 5 times on average due to rejection sampling. For the concrete parameters with $N = 2^{10}$, this means sampling less than 58000 coefficients in total with a standard deviation around $2^{22.5}$. Figure 3 and Table 4 in [40] show that one can sample 1024 Gaussian coefficients in less than 2^{18} cycles *independent* of the standard deviation. Thus, Gaussian sampling with a cost of about $58 \cdot 2^{18} < 2^{24}$ cycles will not be a bottleneck for offline running time.

Online signing. The response phase of the prover requires about $(k+2)m$ polynomial multiplications, which is only $4 \cdot 15 = 60$ `poly256_mult`. When repeated

5 times on average, it would take only around 100 μs on a standard PC. This phase can be treated as the online signing phase and is very fast.

Verification. The verification time of the ring signature is dominated by the last verification step in Protocol 3, which takes around $(k-1)\cdot n\cdot N+n\cdot(k-1)+n\cdot m=6240$ `poly256_mult`. Note that there is no additional factor due to rejection sampling here. This would take about 2 ms with the same assumptions.

The reported signing/verification running times of [24] with the same $N=2^{10}$ is 2.8 seconds. Also, extrapolating the computational efficiency results of NTRU-based ring signature from [27] (without linkability), the running time for signing/verification would be around 700-800 ms (where the signature length is about 14 times larger than ours). Therefore, our ring signature scheme also outperforms [24] and [27], which are the only two works providing concrete running times, by a large margin in terms of computational efficiency for medium-sized rings.

For smaller ring sizes, the scheme in [24] does not seem to get noticeably faster. For example, for $N=2^7$, the running times of signing and verification go down to 2 seconds, i.e., not even reduced by a factor of 2 over the case $N=2^{10}$. In our case, the offline signing and verification times would be reduced by a factor of more than 8 as N is reduced by a factor 8 and k would be set to 1.

G Application to Privacy-Preserving Credentials

In an anonymous credential system, there are three entities: organizations, that are able to issue credentials, users, who can obtain and show credentials, and verifiers, who verify the user credentials. Our goal here is to enable an *efficient* way for users to get a credential containing a set of attributes and later use it to prove that some of the attributes satisfy certain properties without revealing the attribute itself. We provide privacy for credential attributes by revealing only that they satisfy a certain relation, but we don't provide unlinkability between multiple showings or issuing, which is left as an interesting open problem for future work. In fact, linkability is a desired property in some applications such as e-voting and e-cash systems. Also, user anonymity can be obtained to some degree by using pseudonyms. Let us a give a simple example scenario. The user, Alice, wants to apply for a job that only considers applicants of age between 18 and 33, and from a European country. Then, she obtains a credential from her state government with these (and possibly more) attributes. The goal in this scenario is for Alice to convince the employer that she is eligible for the application without revealing the full details as she may not end up getting the job. The privacy of the credential attributes is achieved by revealing only the fact that some attributes satisfy a certain relation. Let us first describe our security model and then see how we can tackle this problem using our tools.

Let $\mathsf{RPoK}(C)$ be a *relaxed* proof of knowledge where the verifier is convinced that the prover knows $(y,\hat{\boldsymbol{r}})$ such that $yC=\mathrm{Com}_{ck}(\mathbf{0};\hat{\boldsymbol{r}})$ and y is invertible. Further, let $\mathit{\Pi}_{\mathbb{P}}(C)$ be a protocol that proves knowledge of $(y,\hat{\boldsymbol{m}},\hat{\boldsymbol{r}})$ such that $yC=\mathrm{Com}_{ck}(y\hat{\boldsymbol{m}};\hat{\boldsymbol{r}})$, y is invertible and $\hat{\boldsymbol{m}}$ satisfies some property $\mathbb{P}$, denoted by $\hat{\boldsymbol{m}} \in \mathbb{P}$ (recall that the relaxation factor in our range proof is invertible). Now, the game between a prover and a challenger works as follows.

1. The prover sends a message $\boldsymbol{m}$ and a commitment C along with $\mathsf{RPoK}(C')$ where $C' = C - \mathrm{Com}_{ck}(\boldsymbol{m}; \boldsymbol{0})$.
2. Then, the challenger chooses a property $\mathbb{P}$ that can be proven by some protocol $\Pi_{\mathbb{P}}$, and sends $\mathbb{P}$ to the prover.
3. Finally, the prover sends $\Pi_{\mathbb{P}}(C)$.

The prover wins if $\boldsymbol{m} \notin \mathbb{P}$.

Theorem 5. *If the commitment scheme* $\mathrm{Com}_{ck}(*; *)$ *is* γ*-binding (for appropriately set* γ*) and* $\mathsf{RPoK}(C')$ *and* $\Pi_{\mathbb{P}}(C)$ *are sound with an invertible relaxation factor, then no PPT adversary can win the above game with a non-negligible probability.*

Proof (Theorem 5). Let $\mathcal{A}$ be a PPT adversary that plays the above game and sends $\boldsymbol{m}$ and C in the first move. Let $C' = C - \mathrm{Com}_{ck}(\boldsymbol{m}; \boldsymbol{0})$. By the soundness of $\mathsf{RPoK}(C')$, $\mathcal{A}$ must know $(y, \hat{\boldsymbol{r}})$ such that

$$yC' = \mathrm{Com}_{ck}(\boldsymbol{0}; \hat{\boldsymbol{r}}) \implies y(C - \mathrm{Com}_{ck}(\boldsymbol{m}; \boldsymbol{0})) = \mathrm{Com}_{ck}(\boldsymbol{0}; \hat{\boldsymbol{r}}) \implies yC = \mathrm{Com}_{ck}(y\boldsymbol{m}; \hat{\boldsymbol{r}}). \quad (34)$$

Now, by the soundness of $\Pi_{\mathbb{P}}(C)$, $\mathcal{A}$ must also know $(y', \hat{\boldsymbol{m}}', \hat{\boldsymbol{r}}')$ such that $\hat{\boldsymbol{m}}' \in \mathbb{P}$ and

$$y'C = \mathrm{Com}_{ck}(y'\hat{\boldsymbol{m}}'; \hat{\boldsymbol{r}}'). \quad (35)$$

Multiplying (34) by y' and (35) by y, we get

$$y' \cdot (yC) = \mathrm{Com}_{ck}(y'y\boldsymbol{m}; y'\hat{\boldsymbol{r}}) \quad \text{and} \quad y \cdot (y'C) = \mathrm{Com}_{ck}(yy'\hat{\boldsymbol{m}}'; y\hat{\boldsymbol{r}}') \implies \mathrm{Com}_{ck}(y'y\boldsymbol{m}; y'\hat{\boldsymbol{r}}) = \mathrm{Com}_{ck}(yy'\hat{\boldsymbol{m}}'; y\hat{\boldsymbol{r}}'). \quad (36)$$

By the binding property of the commitment scheme, $y'y\boldsymbol{m} = yy'\hat{\boldsymbol{m}}'$, and thus $\boldsymbol{m} = \hat{\boldsymbol{m}}' \in \mathbb{P}$ since y, y' are invertible by assumption. □

Remark 2. A similar argument as in Theorem 5 can be used to strengthen the proved relations in our protocols as follows. A relaxed proof for a property $\mathbb{P}$ combined with a proof of *exact* knowledge (i.e., proving knowledge of $(\boldsymbol{m}, \boldsymbol{r})$ such that $C = \mathrm{Com}_{ck}(\boldsymbol{m}; \boldsymbol{r})$) proves that the prover knows an *exact* valid opening of C and that this opening (without any relaxation) satisfy $\mathbb{P}$. However, such lattice-based proofs of exact knowledge are not currently very efficient. Furthermore, the invertibility assumption of the relaxation factor can be circumvented using Lemma 7 provided that the relaxation factor and the message $\boldsymbol{m}$ have bounded norm, i.e., we can infer $\boldsymbol{m} = \boldsymbol{m}'$ from $yy'(\boldsymbol{m} - \boldsymbol{m}') = 0$ in R_q using Lemma 7.

Going back our scenario, $\boldsymbol{m}$ in the above game represents the set of attributes. Alice applies to get a credential with the set of attributes in $\boldsymbol{m}$, and obtains a signature on C after passing the relaxed proof of knowledge. In her application for the job, she first shows that C is signed by an authority and that her age attribute is the range $[18, 33)$, and the expiry date and country code attributes are in some valid ranges (using a single relaxed range proof). Here, the ranges for all the other attributes but these three are set so that they are trivially satisfied (for example, the range is $[0, 2^{32})$ if they are unsigned and represented by 32 bits). Seeing a signature on C and the range proof, the employer is convinced that Alice is eligible to apply.

All of our proofs except for one-out-of-many proof has the structure represented by $\Pi_{\mathbb{P}}$ where the property $\mathbb{P}$ changes for each proof. For example, a similar idea can be also used with the set membership proof. In that case, the public set S in the protocol needs to be set so that it covers all the possible options for the attributes that Alice does not want to reveal any information about.

H Remaining Lemma and Theorem Proofs

Lemma 3

Proof (Lemma 3). Since $\|x\|_\infty \leq p$ and $\|x\|_1 \leq pw$ for all $x \in \mathcal{C}^d_{w,p}$, we have $\|y\|_\infty \leq 2p$ and $\|y\|_1 \leq 2pw$ for all $y \in \Delta\mathcal{C}^d_{w,p}$. Therefore, using Lemma 8, we get

$$\|\prod_{i=1}^{n} y_i\|_\infty \leq \prod_{i=1}^{n-1} \|y_i\|_1 \cdot \|y_n\|_\infty \leq (2p)^n \cdot w^{n-1}.$$

Therefore, we also have

$$\left\|\prod_{i=1}^{n} y_i\right\| \leq \sqrt{d} \cdot \|\prod_{i=1}^{n} y_i\|_\infty \leq \sqrt{d} \cdot (2p)^n \cdot w^{n-1}.$$ □

Lemma 5

Proof (Lemma 5). Let $\kappa' = \frac{k(k-1)}{2}$. We have

$$\begin{aligned}\|\hat{\boldsymbol{m}}_k\| = \left\|\sum_{i=0}^{k} \Gamma_i \boldsymbol{f}_i\right\| &\leq (k+1) \cdot \max_i \|\Gamma_i \boldsymbol{f}_i\| \leq (k+1) \cdot \sqrt{d} \cdot \max_i \|\Gamma_i\| \cdot \max_i \|\boldsymbol{f}_i\| \\ &\leq (k+1) \cdot d \cdot (2p)^{\kappa'} \cdot w^{\kappa'-1} \cdot \max_i \|\boldsymbol{f}_i\|, \quad \text{(by Fact 2 and Lemma 3)}.\end{aligned}$$

The bound for $\hat{\boldsymbol{r}}_k$ follows in a similar manner. □

Lemma 6

Proof (Lemma 6). If $k = 1$, then the result is clear. Assume that $k \geq 2$. Suppose that $X^d + 1$ factors into $n \leq d$ irreducible polynomials $\alpha_1, \ldots, \alpha_n$ modulo q. Let S be the set of indices i such that $g^k = 0 \bmod (q, \alpha_i)$ and $\varepsilon = |S|$. (Note that S may be an empty set and $\varepsilon = 0$).

(1) From the definition of S and the fact that $f \cdot g^k = 0$ over R_q, we have $f = 0 \bmod (q, \alpha_j)$ for all $j \notin S$.

(2) For any $i \in S$, $g^k = 0 \bmod (q, \alpha_i)$ by the definition of S. Since α_i is irreducible modulo q, it is impossible to have this property without having $g = 0 \bmod (q, \alpha_i)$.

Thus, for all $i \notin S$, $f \cdot g = 0 \bmod (q, \alpha_i)$ by (1). And, for all $i \in S$, $f \cdot g = 0 \bmod (q, \alpha_i)$ by (2). By the Chinese Remainder Theorem, $f \cdot g = 0$ over R_q. □

Theorem 1

Proof (Theorem 1). Let $k = k_1 + \cdots + k_\psi$.
Completeness: The prover responds with probability $1/(\mu(\phi_1)\mu(\phi_2)) + \varepsilon$ for $|\varepsilon| \leq 2 \cdot 2^{-100}$ by Lemma 11. Since there is at most $k_1 + \cdots + k_s = k$-many 1's in $\boldsymbol{b}$ and the rest is zero, there will be at most kw non-zero coefficients in $x\boldsymbol{b}$ where each coefficient is in $\{-p, \ldots, p\}$. Thus, we have

$$\|x\boldsymbol{b}\| = \left\| x(b_0^{(1)}, \ldots, b_{k_\psi - 1}^{(\psi)}) \right\| \leq p\sqrt{kw} = T_1.$$

Also, we have, using Lemma 8,

$$\|x(\boldsymbol{r}_b, \boldsymbol{r}_c, \boldsymbol{r})\| \leq \|x(\boldsymbol{r}_b, \boldsymbol{r}_c, \boldsymbol{r})\|_\infty \sqrt{3md} \leq \|x\|_1 \|(\boldsymbol{r}, \boldsymbol{r}_c)\|_\infty \sqrt{3md} \leq pw\mathcal{B}\sqrt{3md} = T_2.$$

Hence, by Lemma 11, the distributions of $f_j^{(i)}$'s are statistically close to $D^{d/s}_{\phi_1 T_1}$ and those of $\boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}$ are statistically close to $D^{md}_{\phi_2 T_2}$ (within statistical distance 2^{-100} in both cases). Therefore, since $d \geq 128$, by Lemma 10 except with probability at most 2^{-128}, we have, for all $1 \leq i \leq \psi$,

$$\|\boldsymbol{z}_b\|, \|\boldsymbol{z}_c\|, \|\boldsymbol{z}\| \leq 2 \cdot \phi_2 pw\mathcal{B}\sqrt{3md} \cdot \sqrt{md} = 2\sqrt{3}\phi_2 pw\mathcal{B}md.$$

Step 4 and 5 of the verification follows by a straightforward investigation. For the last step of the verification, we have, for each $1 \leq i \leq \psi$,

$$\sum_{j=0}^{\psi} 2^j f_j^{(i)} = x \sum_{j=0}^{\psi} 2^j b_j^{(i)} + \sum_{j=0}^{\psi} 2^j a_j^{(i)} = x \cdot \ell^{(i)} + \sum_{j=0}^{\psi} 2^j a_j^{(i)}. \tag{37}$$

Therefore, we get

$$\mathrm{Com}_{ck}(\boldsymbol{v}; \boldsymbol{z}) - E = \mathrm{Com}_{ck}(\boldsymbol{v} - \boldsymbol{e}; \boldsymbol{z} - \boldsymbol{r}_e) = \mathrm{Com}_{ck}(x\ell^{(1)}, \ldots, x\ell^{(\psi)}; x\boldsymbol{r}) = xV.$$

SHVZK: Assume that the protocol is not aborted. The simulator sets $C = \mathrm{Com}_{ck}(\boldsymbol{0}; \boldsymbol{r}_c)$ and $B = \mathrm{Com}_{ck}(\boldsymbol{0}; \boldsymbol{r}_b)$ for $\boldsymbol{r}_c, \boldsymbol{r}_b \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$. Then, it picks $f_j^{(i)} \leftarrow D^{d/s}_{\phi_1 T_1}$ for all $1 \leq i \leq \psi$ and $0 \leq j \leq k_i - 1$, and also $\boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z} \leftarrow D^{md}_{\phi_2 T_2}$. Then, it computes $\boldsymbol{f}$ and $\boldsymbol{f}_{\mathrm{crt}}$ as in the soundness proof. Finally, it computes $A = \mathrm{Com}_{ck}(\boldsymbol{f}; \boldsymbol{z}_b) - xB$, $D = \mathrm{Com}_{ck}(\boldsymbol{g}; \boldsymbol{z}_c) - xC$ and $E = \mathrm{Com}_{ck}(\boldsymbol{v}; \boldsymbol{z}) - xV$ where $\boldsymbol{g}, \boldsymbol{v}$ are set as in Protocol 2. It outputs the simulated transcript $((A, B, C, D, E), x, (\boldsymbol{f}_{\mathrm{crt}}, \boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}))$.

The distribution of simulated $(\boldsymbol{f}_{\mathrm{crt}}, \boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z})$ is statistically close to the real distribution by Lemma 11 as argued in the completeness proof. Conditioned on $(\boldsymbol{f}_{\text{crt}}, \boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}, x)$ and (B, C, V), simulated (A, D, E)'s distribution is exactly the same as in the real case. Finally, the distribution of simulated (B, C) is computationally indistinguishable from the real one by the hiding property of the commitment scheme (i.e., due to M-LWE).

Special Soundness: *Range proof.* By Step 6 of the verification, we have $yV = \mathrm{Com}_{ck}(\boldsymbol{v} - \boldsymbol{v}'; \boldsymbol{z} - \boldsymbol{z}')$. Multiplying $\boldsymbol{v} - \boldsymbol{v}'$ by y, we also know that

$$\begin{aligned} y \cdot (\boldsymbol{v} - \boldsymbol{v}') &= \left(\sum_{j=0}^{k_1-1} 2^j (y f_j^{(1)} - y f_j'^{(1)}), \ldots, \sum_{j=0}^{k_\psi - 1} 2^j (y f_j^{(\psi)} - y f_j'^{(\psi)}) \right) \\ &= \left(\sum_{j=0}^{k_1-1} 2^j (x - x') \hat{b}_j^{(i)}, \ldots, \sum_{j=0}^{k_\psi - 1} 2^j (x - x') \hat{b}_j^{(\psi)} \right) \quad \text{(by (20) and (22))}, \\ &= \left(\sum_{j=0}^{k_1-1} 2^j y^2 b_j^{(i)}, \ldots, \sum_{j=0}^{k_\psi - 1} 2^j y^2 b_j^{(\psi)} \right) \qquad \text{(by (26))}. \end{aligned}$$

Let us focus on a coordinate $y\hat{\ell}^{(i)}$ of $y(\boldsymbol{v} - \boldsymbol{v}')$ for any $1 \le i \le \psi$. Since $y = \langle y, \ldots, y \rangle$ and it is invertible in all $R_q^{(j)}$'s, it is invertible in R_q. Then, we have

$$y\hat{\ell}^{(i)} = y^2 \underbrace{\sum_{j=0}^{k_i-1} 2^j b_j^{(i)}}_{\in [0, N_i - 1]} \quad \Longrightarrow \quad \hat{\ell}^{(i)} = y \ell^{(i)} \text{ for some } \ell^{(i)} \in [0, N_i - 1]. \tag{38}$$

As a result, $yV = \mathrm{Com}_{ck}(y\ell^{(1)}, \ldots, y\ell^{(\psi)}; \boldsymbol{z} - \boldsymbol{z}')$ where $\ell^{(1)}, \ldots \ell^{(\psi)}$ are in the ranges $[0, N_1), \ldots, [0, N_\psi)$, respectively. Note that since $q > \max\{N_1, \ldots, N_\psi\}$ there is no modular reduction performed when computing $\sum_{j=0}^{k_i-1} 2^j b_j^{(i)}$. □

Theorem 2

Proof (Theorem 2). **Completeness:** The main difference from Protocol 2 is that there is a sum of $f_{j,i}$'s, which follow a normal distribution. By the discussion in Appendix B.2, the sum of discrete normal variables behaves as its continuous counterpart. That is, the distribution of $\sum_{i=1}^{\beta-1} f_{j,i}$ is statistically close to $D^d_{\phi_1 T_1 \sqrt{\beta - 1}}$. Hence, we have

$$\begin{aligned} \|f_{j,i}\| &\le 2 \cdot \phi_1 p \sqrt{kw} \cdot \sqrt{d} = 2\phi_1 p \sqrt{dkw}, \qquad \forall j \in [0, k),\ \forall i \in [0, \beta),\ \text{and} \\ \|f_{j,0}\| &= \left\| x - \sum_{i=1}^{\beta-1} f_{j,i} \right\| \le \|x\| + \left\| \sum_{i=1}^{\beta-1} f_{j,i} \right\| \\ &\le \sqrt{w} + 2 \cdot \phi_1 p \sqrt{kw(\beta - 1)} \cdot \sqrt{d} \approx 2\phi_1 p \sqrt{\beta dkw}. \end{aligned}$$

The rest is analogous to the completeness proof of Protocol 2.

SHVZK: Assume that the protocol is not aborted. The simulator sets $C =$

$\mathrm{Com}_{ck}(\mathbf{0}; \boldsymbol{r}_c)$ for $\boldsymbol{r}_c \leftarrow \{-\mathcal{B}, \ldots, \mathcal{B}\}^{md}$, and picks $f_{j,i} \leftarrow D^d_{\phi_1 T_1}$ for all $0 \leq j \leq k-1$ and $1 \leq i \leq \beta - 1$, and also $\boldsymbol{z}_b, \boldsymbol{z}_c \leftarrow D^{md}_{\phi_2 T_2}$. Then, given x, it sets $f_{j,0} = x - \sum_{i=1}^{\beta-1} f_{j,i}$ for all $j = 0, \ldots, k-1$. Finally, it computes $A = \mathrm{Com}_{ck}(\boldsymbol{f}; \boldsymbol{z}_b) - xB$ and $D = \mathrm{Com}_{ck}(\boldsymbol{g}; \boldsymbol{z}_c) - xC$ where $\boldsymbol{f}, \boldsymbol{g}$ are set as in Protocol 4. It outputs the simulated transcript $((A, C, D), x, (\boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z}_c))$ where $\boldsymbol{f}_1$ is set as in Protocol 4. The indistinguishability argument is as in SHVZK of Theorem 1.

3-special soundness: The proof proceeds almost identical to the soundness proof of Theorem 1 up to Equation (25) except that when the commitment scheme is instantiated with HMC, we need to bound the norm of the message-randomness opening pair together to use the binding property argument. The exact opening of yD has the largest norm-bound of

$$\gamma_{\mathrm{bin}} = 2p\sqrt{dw}\left(16\phi_1^4 p^4 d^3 k^3 w^2 \beta(\beta+1) + 12\phi_2^2 p^2 w^2 \mathcal{B}^2 m^2 d^2\right)^{1/2}$$

as shown in Lemma 13. As in (20), (21), (22), we have the following equations, but now holding in R_q

$$\begin{aligned} y f_{j,i} &= x\hat{b}_{j,i} + \hat{a}_{j,i}, & y f_{j,i}(x - f_{j,i}) &= x\hat{c}_{j,i} + \hat{d}_{j,i}, \\ y f'_{j,i} &= x'\hat{b}_{j,i} + \hat{a}_{j,i}, & y f'_{j,i}(x' - f'_{j,i}) &= x'\hat{c}_{j,i} + \hat{d}_{j,i}, \quad \text{in } R_q. \\ y f''_{j,i} &= x''\hat{b}_{j,i} + \hat{a}_{j,i}, & y f''_{j,i}(x'' - f''_{j,i}) &= x''\hat{c}_{j,i} + \hat{d}_{j,i}, \end{aligned} \tag{39}$$

Then, by the γ_{bin}-binding property of HMC, we get below the same system of equations as in (25) of the soundness proof of Theorem 1

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

Note that again all equations now hold in R_q, and there is no use of any invertibility argument. Now, multiplying both sides of the above system of equations by $\mathrm{adj}(\boldsymbol{V})$ where $\boldsymbol{V}$ is the Vandermonde matrix on the left, we get

$$\det(\boldsymbol{V})\hat{b}_{j,i}(y - \hat{b}_{j,i}) = (x'' - x')(x' - x)(x'' - x)\hat{b}_{j,i}(y - \hat{b}_{j,i}) = 0 \quad \text{in } R_q. \tag{40}$$

We also know that

$$\begin{aligned} \|\hat{b}_{j,i}\|_1 &= \|f_{j,i} - f'_{j,i}\|_1 \leq 2 \cdot \sqrt{d} \cdot 2\phi_1 p\sqrt{\beta dkw} = 4\phi_1 pd\sqrt{\beta kw}, \text{ and} \\ \|y - \hat{b}_{j,i}\|_1 &\leq \|y\|_1 + \|\hat{b}_{j,i}\|_1 \leq pw + 4\phi_1 pd\sqrt{\beta kw} \approx 4\phi_1 pd\sqrt{\beta kw}. \end{aligned}$$

From here, we can further get a bound as

$$\begin{aligned} \|(x'' - x')(x' - x)(x'' - x)\hat{b}_{j,i}(y - \hat{b}_{j,i})\|_\infty &\leq \\ \|x'' - x'\|_\infty \cdot \|x' - x\|_1 \cdot \|x'' - x\|_1 \cdot \|\hat{b}_{j,i}\|_1 \cdot \|y - \hat{b}_{j,i}\|_1 & \\ &\leq 2p \cdot (2pw)^2 \cdot (4\phi_1 pd\sqrt{\beta kw})^2 \\ &= 2^7 \phi_1^2 p^5 w^3 d^2 k\beta. \end{aligned} \tag{41}$$

Since $q/2 > 2^7\phi_1^2 p^5 w^3 d^2 k\beta$, one of the factors in (40) must be zero by Lemma 7. As challenge differences are non-zero, this gives either $\hat{b}_{j,i}$ or $y - \hat{b}_{j,i}$ is zero. Thus, we get $\hat{b}_{j,i} \in \{0, y\}$. That is, $\hat{b}_{j,i} = yb_{j,i}$ for $b_{j,i} \in \{0,1\}$ as needed for $\mathcal{R}'_{\text{bin}}$.

Finally, for all $j = 0, \dots, k-1$, by multiplying Step 2 of the verification by y, we have the following

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

This holds for 2 distinct challenges x and x', and therefore

$$\left(\sum_{i=0}^{\beta-1} b_{j,i} - 1\right) y(x - x') = \left(\sum_{i=0}^{\beta-1} b_{j,i} - 1\right) y^2 = 0 \quad \text{in } R_q.$$

Using Lemma 7 as above (the condition on the size of q here is much weaker), we get $\sum_{i=0}^{\beta-1} b_{j,i} = 1$ for all $0 \le j \le k-1$ as required. □

Theorem 3

Proof (Theorem 3). **Completeness:** Step 1 of verification follows from the completeness of Protocol 4. For bounding the maximum norm of masked randomnesses in Step 11 of prover's computation, we have

$$\begin{aligned}
\left\| x^k \boldsymbol{r} - \sum_{j=1}^{k-1} x^j \boldsymbol{\rho}_j \right\| &\le \left\| x^k \boldsymbol{r} \right\| + \sum_{j=1}^{k-1} \left\| x^j \boldsymbol{\rho}_j \right\| \le \sqrt{md} \cdot \left(\| x^k \boldsymbol{r} \|_\infty + \sum_{j=1}^{k-1} \| x^j \boldsymbol{\rho}_j \|_\infty \right) \\
&\le \sqrt{md} \cdot \left(\|x\|_1^k \|\boldsymbol{r}\|_\infty + \sum_{j=1}^{k-1} \|x\|_1^j \|\boldsymbol{\rho}_j\|_\infty \right) \\
&\le \sqrt{md} \cdot \left((pw)^k \mathcal{B} + \sum_{j=1}^{k-1} (pw)^j \mathcal{B} \right) = \mathcal{B}\sqrt{md} \sum_{j=1}^{k} (pw)^j.
\end{aligned}$$

Denote $\boldsymbol{r'} = x^k \boldsymbol{r} - \sum_{j=1}^{k-1} x^j \boldsymbol{\rho}_j$. Then, we have

$$\begin{aligned}
\|(x\boldsymbol{r}_b, x\boldsymbol{r}_c, \boldsymbol{r'})\| &= \left(\|(x\boldsymbol{r}_b, x\boldsymbol{r}_c)\|^2 + \|\boldsymbol{r'}\|^2\right)^{1/2} \\
&\leq \left(\left(pw\mathcal{B}\sqrt{2md}\right)^2 + \left(\mathcal{B}\sqrt{md}\sum_{j=1}^{k}(pw)^j\right)^2\right)^{1/2} \\
&\leq \left(2w^2p^2\mathcal{B}^2md + \mathcal{B}^2md\left(\sum_{j=1}^{k}(pw)^j\right)^2\right)^{1/2} \\
&\leq \left(\mathcal{B}^2md \cdot \left[2w^2p^2 + \left(\sum_{j=1}^{k}(pw)^j\right)^2\right]\right)^{1/2} \leq \mathcal{B}w^kp^k\sqrt{3md}.
\end{aligned}$$

Therefore, for $T_2 = \mathcal{B}w^kp^k\sqrt{3md}$, the distribution of $\boldsymbol{z}, \boldsymbol{z}_b, \boldsymbol{z}_c$ are statistically close to $D^{md}_{\phi_2 T_2}$ by Lemma 11. Hence, by Lemma 10, we have

$$\|\boldsymbol{z}\|, \|\boldsymbol{z}_b\|, \|\boldsymbol{z}_c\| \leq 2 \cdot \phi_2\mathcal{B}w^kp^k\sqrt{3md} \cdot \sqrt{md} = 2\sqrt{3}\phi_2\mathcal{B}mdw^kp^k.$$

For the last verification, we have, for $P_\ell = \mathrm{Com}_{ck}(\boldsymbol{0};\, \boldsymbol{r})$,

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

SHVZK: Assume that the protocol is not aborted. $A, B, C, D, \boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}$ are simulated as in the case of Protocol 2 where $\boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z}$ are sampled from $D^{md}_{\phi_2 T_2}$ for $T_2 = \mathcal{B}p^kw^k\sqrt{3md}$. The simulator also samples $E_1, \ldots, E_{k-1} \leftarrow \mathcal{U}(R_q^n)$ and computes E_0 in the way to ensure that the last verification step is satisfied. Then, the simulated transcript is output as below

$$((A, B, C, D, \{E_j\}_{j=0}^{k-1}), x, (\boldsymbol{f}_1, \boldsymbol{z}_b, \boldsymbol{z}_c, \boldsymbol{z})).$$

The simulation of $E_1, \ldots, E_{k-1}$ is computationally indistinguishable from the real case by M-LWE assumption. The rest of the indistinguishability argument is the same as in SHVZK of Protocol 2. $\square$

Lemma 12. *The vector $\boldsymbol{g}$ defined in Protocol 4 satisfies the following*

$$\|\boldsymbol{g}\|^2 \leq 16\phi_1^4 p^4 d^3 k^3 w^2 \beta(\beta+1).$$

Proof. We use the bounds on the norm of $f_{j,i}$'s in the sequel (see Protocol 4 definition). For simplicity, we bound $\|x - f_{j,0}\|$ by the bound on $\|f_{j,0}\|$ as $\|x\|$ is much smaller in comparison.

$$\begin{aligned}
\|\boldsymbol{g}\|^2 &= \sum_{j=0}^{k-1}\sum_{i=0}^{\beta-1} \|f_{j,i}(x - f_{j,i})\|^2 = \sum_{j=0}^{k-1}\sum_{i=1}^{\beta-1} \|f_{j,i}(x - f_{j,i})\|^2 + \sum_{j=0}^{k-1} \|f_{j,0}(x - f_{j,0})\|^2 \\
&\leq \sum_{j=0}^{k-1}\sum_{i=1}^{\beta-1} d\,\|f_{j,i}\|^2\,\|x - f_{j,i}\|^2 + \sum_{j=0}^{k-1} d\,\|f_{j,0}\|^2\,\|x - f_{j,0}\|^2 \\
&\leq dk(\beta-1)\left(2\phi_1 p\sqrt{dkw}\right)^4 + dk\left(2\phi_1 p\sqrt{\beta dkw}\right)^4 \\
&\leq dk\left(2\phi_1 p\sqrt{dkw}\right)^4 \cdot \left[(\beta-1) + \beta^2\right] \leq 16\phi_1^4 p^4 d^3 k^3 w^2 \beta(\beta+1).
\end{aligned}$$

□

Lemma 13. *The exact opening $(\hat{\boldsymbol{d}}, \hat{\boldsymbol{r}}_{\boldsymbol{d}})$ of yD in the soundness proof of Theorem 2 satisfies the following*

$$\left\|(\hat{\boldsymbol{d}}, \hat{\boldsymbol{r}}_{\boldsymbol{d}})\right\| \leq 2p\sqrt{dw}\left(16\phi_1^4 p^4 d^3 k^3 w^2 \beta(\beta+1) + 12\phi_2^2 p^2 w^2 \mathcal{B}^2 m^2 d^2\right)^{1/2}.$$

Proof. For $y = x - x'$, we have

$$yC = \mathrm{Com}_{ck}(\boldsymbol{g} - \boldsymbol{g}';\, \boldsymbol{z}_c - \boldsymbol{z}_c'). \tag{42}$$

Then, the exact opening $(\hat{\boldsymbol{d}}, \hat{\boldsymbol{r}}_{\boldsymbol{d}})$ of yD is obtained as follows

$$\begin{aligned}
yD &= \mathrm{Com}_{ck}(y\boldsymbol{g};\, y\boldsymbol{z}_c) - xyC = \mathrm{Com}_{ck}(y\boldsymbol{g};\, y\boldsymbol{z}_c) - x\mathrm{Com}_{ck}(\boldsymbol{g} - \boldsymbol{g}';\, \boldsymbol{z}_c - \boldsymbol{z}_c') \\
&= \mathrm{Com}_{ck}(x\boldsymbol{g} - x'\boldsymbol{g};\, x\boldsymbol{z}_c - x'\boldsymbol{z}_c) - \mathrm{Com}_{ck}(x\boldsymbol{g} - x\boldsymbol{g}';\, x\boldsymbol{z}_c - x\boldsymbol{z}_c') \\
&= \mathrm{Com}_{ck}(x\boldsymbol{g}' - x'\boldsymbol{g};\, x\boldsymbol{z}_c' - x'\boldsymbol{z}_c).
\end{aligned} \tag{43}$$

Without loss of generality, assume that $\|(x\boldsymbol{g}', x\boldsymbol{z}_c')\| \geq \|(x'\boldsymbol{g}, x'\boldsymbol{z}_c)\|$.

$$\begin{aligned}
&\left\|(\hat{\boldsymbol{d}}, \hat{\boldsymbol{r}}_{\boldsymbol{d}})\right\| = \|(x\boldsymbol{g}' - x'\boldsymbol{g}, x\boldsymbol{z}_c' - x'\boldsymbol{z}_c)\| \leq 2\,\|(x\boldsymbol{g}', x\boldsymbol{z}_c')\| \leq 2\sqrt{d}\,\|x\| \cdot \|(\boldsymbol{g}', \boldsymbol{z}_c')\| \\
&\leq 2p\sqrt{dw} \cdot \|(\boldsymbol{g}', \boldsymbol{z}_c')\| = 2p\sqrt{dw}\left(\|\boldsymbol{g}'\|^2 + \|\boldsymbol{z}_c'\|^2\right)^{1/2} \\
&\leq 2p\sqrt{dw}\left(16\phi_1^4 p^4 d^3 k^3 w^2 \beta(\beta+1) + \left(2\sqrt{3}\phi_2 p w \mathcal{B} m d\right)^2\right)^{1/2} \quad \text{(by Lemma 12)} \\
&= 2p\sqrt{dw}\left(16\phi_1^4 p^4 d^3 k^3 w^2 \beta(\beta+1) + 12\phi_2^2 p^2 w^2 \mathcal{B}^2 m^2 d^2\right)^{1/2}.
\end{aligned} \tag{44}$$

□

Remark 3. Note that, when bounding $\|\boldsymbol{z}_c\|$ in the proof of Lemma 13, we use the norm bound in Protocol 2's verification, which is a stronger case than that in Protocol 4. The norm bound of $\boldsymbol{z}_c$ in Protocol 3 is the largest one, and (44) becomes $2p\sqrt{dw}\left(16\phi_1^4 p^4 d^3 k^3 w^2 \beta(\beta+1) + 12\phi_2^2 p^{2k} \mathcal{B}^2 m^2 d^2 w^{2k}\right)^{1/2}$ using that bound. We consider the strongest bound when instantiating the parameters for the ring signature.