Practical Product Proofs for Lattice Commitments*

Thomas Attema[1,2,3], Vadim Lyubashevsky[4], and Gregor Seiler[4,5]

[1] CWI – Amsterdam, The Netherlands
[2] Leiden University, The Netherlands
[3] TNO – The Hague, The Netherlands
[4] IBM Research – Zurich, Switzerland
[5] ETH Zurich, Switzerland

Abstract. We construct a practical lattice-based zero-knowledge argument for proving multiplicative relations between committed values. The underlying commitment scheme that we use is the currently most efficient one of Baum et al. (SCN 2018), and the size of our multiplicative proof (8KB) is only slightly larger than the 7KB required for just proving knowledge of the committed values. We additionally expand on the work of Lyubashevsky and Seiler (Eurocrypt 2018) by showing that the above-mentioned result can also apply when working over rings $\mathbb{Z}_q[X]/(X^d+1)$ where X^d+1 splits into low-degree factors, which is a desirable property for many applications (e.g. range proofs, multiplications over $\mathbb{Z}_q$) that take advantage of packing multiple integers into the NTT coefficients of the committed polynomial.

1 Introduction

Commitment schemes, and their associated zero-knowledge proofs of knowledge (ZKPoK) of committed messages, form an important ingredient in the construction of generalized zero-knowledge proofs and advanced cryptographic primitives. An additional feature that's often desirable is being able to prove algebraic relationships among committed values. Very efficient constructions of such primitives exist based on the discrete logarithm problem (e.g. [BBB+18]), but the state of affairs is rather different when it comes to quantum-safe assumptions, with the maid difficulty being proving multiplicative relations.

There exist generic PCP-type proof techniques [Kil92, Mic00, BBHR18, BCR+19], which even have asymptotically logarithmic-size proofs, but these proofs have a fixed cost of outputting paths to a Merkle tree in the range of $100-200$KB. One could also think about using fully-homomorphic encryption, which would allow the verifier himself to create additive and multiplicative relations of his choice, thus foregoing the need for a zero-knowledge proof. The main issue with this approach is that one would need to prove that the initial ciphertexts are well-formed, and these proofs are also currently on the order of a few hundred kilobytes (either using generic techniques or lattice-based proofs [BLS19, YAZ+19]). There have also been lattice-based approaches proposed for this type of problem (e.g. [BKLP15, LLNW18]), but they result in proofs that are orders of magnitude longer.

1.1 Results Overview.

While there aren't yet any practical lattice-based commitment schemes for proving multiplicative relations among committed values, the commitment scheme in [BDL+18] has a ZK proof that is fairly efficient for proving linear relations among committed polynomials over the ring $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d+1)$, where q is prime. The main result of our work is building an efficient ZKPoK of multiplicative relations for messages committed under this scheme. Our construction is very efficient with the communication complexity of our protocol being essentially the same as that in [BDL+18] for just proving knowledge of the message. An additional result of this work is lifting the restriction from [BDL+18] that required that the ring $\mathcal{R}_q$ be chosen such that all

* This research was supported by the SNSF ERC starting transfer grant FELICITY and the EU H2020 project No 780701 (PROMETHEUS).

polynomials with small coefficients (which corresponds to the set of all possible difference of challenges) be invertible in the underlying ring $\mathcal{R}_q$. By [LS18], one could conclude that elements with small coefficients are invertible in $\mathcal{R}_q$ if the polynomial $X^d + 1$ does not split into too many factors over q. Removing this aforementioned restriction is particularly useful because if $X^d + 1$ splits into *distinct* linear (or very low-degree) factors, then it allows one to commit to (and independently operate on) many elements in $\mathbb{Z}_q$ by packing them into the NTT coefficients of the committed message. One particular example where this is handy is range proofs where we commit to a number written in binary and want to prove that it is in the range $[0, 2^j)$. We sketch the idea below.

Proving that a vector $\vec{v} = v_0 v_1 \dots v_{d-1} \in \{0,1\}^d$ is binary and the integer represented by it is less than 2^j is equivalent to the statement

$$\begin{bmatrix} v_0 \\ \cdots \\ v_{j-1} \\ v_j \\ \cdots \\ v_{d-1} \end{bmatrix} \circ \begin{bmatrix} 1 - v_0 \\ \cdots \\ 1 - v_{j-1} \\ v_j \\ \cdots \\ v_{d-1} \end{bmatrix} = 0 \bmod q, \tag{1}$$

where $\circ$ is the component-wise product. Thus if we create a commitment to $\vec{v}$ by putting the coefficients of $\vec{v}$ into the NTT coefficients of some polynomial $\boldsymbol{m}$ and can create the polynomial $\boldsymbol{m}'$ corresponding to the right multiplicand in (1), then the proof that $\boldsymbol{m}\boldsymbol{m}' = 0$ would be exactly the range proof we would like since multiplication of NTT slots is component-wise. Note that the number of NTT slots is the logarithm of the largest integer that can be committed to. Using our multiplicative proofs, range proofs for 32-bit numbers are approximately 5.6KB in size (see Section 5.3). This is about an order of magnitude longer than the discrete logarithm based proofs (c.f. [BBB+18, Table 2]), but is shorter than any quantum-safe proof system (e.g. [BCR+19, KKW18, ESLL19]). It should be pointed out that the proofs in [BBB+18, BCR+19] grow logarithmically in the number of instances, while our proof grows linearly. The results of the current work are thus best suited for non-batched use cases where one wishes to prove knowledge about single instances over $\mathcal{R}_q$ (which actually could be up to d instances over $\mathbb{Z}_q$ when taking advantage of NTT packing.)

1.2 Techniques.

We will now provide a somewhat technical overview of the main results of the paper. Prior to getting into them, we recall the commitment scheme of [BDL+18] and its zero-knowledge proof.

Overview of [BDL+18]. The scheme of [BDL+18] commits to a message vector $\vec{\boldsymbol{m}} \in \mathcal{R}_q^k$ by choosing a vector $\vec{\boldsymbol{r}}$ with small coefficients and then outputting the commitment

$$\boldsymbol{B}_0 \vec{\boldsymbol{r}} = \vec{\boldsymbol{t}}_0 \tag{2}$$

$$\boldsymbol{B}_1 \vec{\boldsymbol{r}} + \vec{\boldsymbol{m}} = \vec{\boldsymbol{t}}_1. \tag{3}$$

The intuition is that if the opening proof can show that $\vec{\boldsymbol{r}}$ is short, then (2) binds the committer to the short $\vec{\boldsymbol{r}}$ (based on the hardness of the SIS problem), and then the message is uniquely determined from (3). Unfortunately, there do not exist very efficient proofs allowing a prover to prove knowledge of such a short $\vec{\boldsymbol{r}}$ satisfying (2), but one can instead give a rather efficient ZKPoK of a vector $\bar{\vec{\boldsymbol{z}}}$ with coefficients somewhat larger than those of $\vec{\boldsymbol{r}}$, and a polynomial $\bar{\boldsymbol{c}}$ with very small coefficients satisfying

$$\boldsymbol{B}_0 \bar{\vec{\boldsymbol{z}}} = \bar{\boldsymbol{c}} \vec{\boldsymbol{t}}_0. \tag{4}$$

The proof is a Σ-protocol where the prover picks a small-coefficient masking vector $\vec{\boldsymbol{y}}$ and sends $\vec{\boldsymbol{w}} = \vec{\boldsymbol{B}}_0 \vec{\boldsymbol{y}}$ to the verifier in the first step. The verifier then selects a challenge polynomial $\boldsymbol{c}$ from the challenge set (which should consist of polynomials with very small coefficients), and the prover responds with $\vec{\boldsymbol{z}} = \vec{\boldsymbol{y}} + \boldsymbol{c}\vec{\boldsymbol{r}}$. Using

standard rejection sampling techniques [Lyu09, Lyu12], the prover can make the vector $\vec{\boldsymbol{z}}$ independent of $\vec{\boldsymbol{r}}$ to preserve zero-knowledge. The verifier checks that $\boldsymbol{B}_0\vec{\boldsymbol{z}} = \vec{\boldsymbol{w}} + \boldsymbol{c}\vec{\boldsymbol{t}}_0$ and that $\vec{\boldsymbol{z}}$ has small coefficients. If both of these are satisfied (and $\boldsymbol{c}$ comes from a large-enough domain), then a standard rewinding (where the extractor sends a fresh $\boldsymbol{c}'$ and receives another valid $\vec{\boldsymbol{z}}'$) allows the extractor to obtain $\bar{\vec{\boldsymbol{z}}} = \vec{\boldsymbol{z}} - \vec{\boldsymbol{z}}'$ and $\bar{\boldsymbol{c}} = \boldsymbol{c} - \boldsymbol{c}'$ satisfying (4).

Combining this with the proof that, unless SIS is easy, there can only be a unique opening $(\bar{\vec{\boldsymbol{z}}}, \vec{\boldsymbol{m}}, \bar{\boldsymbol{c}})$ where $\bar{\boldsymbol{c}}$ is invertible in $\mathcal{R}_q$ satisfying (4) and

$$\boldsymbol{B}_1\bar{\vec{\boldsymbol{z}}} + \vec{\boldsymbol{m}}\bar{\boldsymbol{c}} = \bar{\boldsymbol{c}}\vec{\boldsymbol{t}}_1, \tag{5}$$

it implies that the ZKPoK of (4) uniquely determines $\vec{\boldsymbol{m}}$. It is furthermore shown in [BDL$^+$18] (also see [dPLS18]) that one can prove that a commitment is to some $\vec{\boldsymbol{m}}$ satisfying $\boldsymbol{U}\vec{\boldsymbol{m}} = \vec{\boldsymbol{v}}$, where $\boldsymbol{U}$ and $\vec{\boldsymbol{v}}$ are an arbitrary public matrix and vector over $\mathcal{R}_q$. Interestingly, this latter proof does not require any extra communication over the basic opening proof, and both the proof and commitment are comfortably under 10KB for some simple lattice relations (see Table 2 of [BDL$^+$18]).

Distribution of the NTT Coefficients. To show that $\bar{\boldsymbol{c}}$ is invertible, it was proposed in [BDL$^+$18] to set the modulus q to a prime such that the polynomial $X^d + 1$ does not split too much modulo q – then by the result in [LS18], it would imply that all elements in the ring with small coefficients are invertible.

In the current paper we show that one no longer needs such a restriction q to be any prime. In particular, the prime q can be chosen to allow $X^d + 1$ to fully split into d linear factors. The observation is that we do not need $\bar{\boldsymbol{c}}$ to always be invertible – it suffices for it to be invertible with high probability.

An element in $\mathcal{R}_q$ is invertible if and only if all of its NTT coefficients are non-zero. To show that $\bar{\boldsymbol{c}} = \boldsymbol{c} - \boldsymbol{c}'$ is invertible, it would therefore suffice to show that the probability that a random $\boldsymbol{c}$ from the challenge set hits a particular NTT coefficient is smaller than the targeted soundness error. If $\boldsymbol{c}$ were uniformly random in $\mathcal{R}_q$, then this probability would be easy to calculate as each of its NTT coefficients has a $1/q$ probability of being any element in $\mathbb{Z}_q$. But $\boldsymbol{c}$ is chosen from a challenge set that has small coefficients and so the distribution of its NTT coefficients requires different techniques to compute.

As an example, suppose that $X^d + 1 = \prod_{i=1}^{d}(X - r_i) \bmod q$ and that we choose an element $\boldsymbol{c} = \sum_{j=0}^{d-1} c_i X^i$ from $\mathbb{Z}_q[X]/(X^d + 1)$ where $c_i \leftarrow \{-1, 0, 1\}$ with equal probability. Then

$$\Pr[\boldsymbol{c} \text{ is invertible}] = \Pr[\boldsymbol{c}(r_1) \neq 0 \wedge \ldots \wedge \boldsymbol{c}(r_d) \neq 0].$$

Observe that for any r, $\boldsymbol{c}(r)$ can be written as

$$\sum_{j=0}^{d-1} c_j r^j = c_0 + r\left(c_1 + r\left(c_2 + \ldots + r\left(c_{d-2} + rc_{d-1}\right)\right)\ldots\right),$$

and so the distribution of $\boldsymbol{c}(r)$ is equivalent to the distribution of the random variable Y_0 in the stochastic process $(Y_d, Y_{d-1}, Y_{d-2}, \ldots, Y_0)$ where $Y_d = 0$ and $Y_i = c_i + rY_{i+1}$ for $i < d$. Fourier analysis is often a useful technique for analyzing certain properties (e.g. min entropy, mixing time, etc.) of stochastic processes, and we show how to efficiently calculate $\max_{y \in \mathbb{Z}_q}[Y_0 = y]$.[6] Calculating the exact probability (or putting a very good bound on it) would require computing sums consisting of q terms, which may be prohibitive when q is on the order of billions, so we furthermore show how certain algebraic symmetries allow us to significantly speed up the computation.

[6] In [CLS16], the same techniques were used to show that the statistical distance of Ring-LWE errors is statistically-close to uniform modulo the NTT coefficients. The slight differences are in the distribution of the original polynomial (for our application, it only makes sense to consider polynomials whose coefficients have various distributions over $\{-1, 0, 1\}$) and that we do not need statistical closeness for our application, and obtain tight bounds for a different quantity. We provide more details in Section 3.

In our applications, we will actually be more interested in a more general case of proving that for a factorization

$$X^d + 1 = \prod_{i=1}^{d/k} (X^k - r_i), \text{for } r_i \in \mathbb{Z}_q, \tag{6}$$

the value $\boldsymbol{c} \bmod (X^k - r_i)$ is not concentrated on any particular polynomial $c'_0 + c'_1 X + \ldots + c'_{k-1} X^{k-1}$. But proving this is a simple extension of the above case where we were computing $\boldsymbol{c}(r) = \boldsymbol{c} \bmod (X - r)$ because each of the k coefficients $c'_i X^i$ of $c \bmod X^k - r_i$ is only dependent on the coefficients c_{jk+i} for $0 \leq j < d/k$ (i.e. the k coefficients are mutually independent). So the distribution of c'_i has the distribution of the same stochastic process as above, except it consists of d/k steps rather than d.

Proofs of Multiplicative Relations. We now sketch some of the new ingredients of our main result – being able to prove multiplicative relations among committed messages in the commitment scheme defined by (2) and (3). In its most basic form, this involves proving that $\boldsymbol{m}_1 \boldsymbol{m}_2 = \boldsymbol{m}_3$, where $\vec{\boldsymbol{m}} = \begin{bmatrix} \boldsymbol{m}_1 & \boldsymbol{m}_2 & \boldsymbol{m}_3 \end{bmatrix}^T$.

We first make a series of observations that show that one can extract more than just (4) from the prover that produces valid transcripts $(\vec{\boldsymbol{w}}, \boldsymbol{c}, \vec{\boldsymbol{z}})$ following the protocol of [BDL+18]. If we assume, for the moment, that $\bar{\boldsymbol{c}}$ is invertible, then the extractor can extract a unique $\vec{\boldsymbol{r}}^* = \bar{\vec{\boldsymbol{z}}}/\bar{\boldsymbol{c}}$, not necessarily with small coefficients, satisfying

$$\boldsymbol{B}\vec{\boldsymbol{r}}^* = \vec{\boldsymbol{t}}. \tag{7}$$

The reason for the uniqueness is that for any small-norm $\left(\bar{\vec{\boldsymbol{z}}}_1, \bar{\boldsymbol{c}}_1\right), \left(\bar{\vec{\boldsymbol{z}}}_2, \bar{\boldsymbol{c}}_2\right)$ satisfying

$$\boldsymbol{B}\bar{\vec{\boldsymbol{z}}}_1 = \bar{\boldsymbol{c}}_1 \vec{\boldsymbol{t}} \qquad\qquad \boldsymbol{B}\bar{\vec{\boldsymbol{z}}}_2 = \bar{\boldsymbol{c}}_2 \vec{\boldsymbol{t}}, \tag{8}$$

if $\bar{\vec{\boldsymbol{z}}}_1/\bar{\boldsymbol{c}}_1 \neq \bar{\vec{\boldsymbol{z}}}_2/\bar{\boldsymbol{c}}_2$, then (4) implies that

$$\boldsymbol{B}\left(\bar{\boldsymbol{c}}_2 \bar{\vec{\boldsymbol{z}}}_1 - \bar{\boldsymbol{c}}_1 \bar{\vec{\boldsymbol{z}}}_2\right) = 0. \tag{9}$$

where the vector being multiplied by $\boldsymbol{B}$ has small coefficients. By the assumption, this vector in additionally non-zero, and so it's a solution to SIS. The next observation (see Section 4) crucial for keeping our product proof short is that as soon as the (successful) Prover sends $\vec{\boldsymbol{w}}$, he has also committed to a $\vec{\boldsymbol{y}}^*$ satisfying $\boldsymbol{B}\vec{\boldsymbol{y}}^* = \vec{\boldsymbol{w}}$. Furthermore, for a challenge $\boldsymbol{c}$, his response $\vec{\boldsymbol{z}}$ will always be

$$\vec{\boldsymbol{z}} = \vec{\boldsymbol{y}}^* + \boldsymbol{c}\vec{\boldsymbol{r}}^*. \tag{10}$$

This is important because of how the product proof works. For simplicity, we will explain how this would result in immediate improvements in the particular product proofs implicit in [BLS19, YAZ+19] which essentially prove that the pointwise product of $\vec{\boldsymbol{r}}$ and $\vec{\boldsymbol{1}} - \vec{\boldsymbol{r}}$ is a zero vector, which implies that $\vec{\boldsymbol{r}}$ is a 0/1 polynomial. Our scenario is different and we relegate the details to Section 5, but the core approach for proving multiplicative relations is quite similar, and so the reason for the efficiency gain is the same. In particular, the main idea is to convince the verifier that the prover has set up a quadratic equation in which the highest-degree term is exactly the relation that we would like to prove is 0 (in our case, it would be $\boldsymbol{m}_1 \boldsymbol{m}_2 - \boldsymbol{m}_3$). Then the prover convinces the verifier that this equation is actually linear, which implies that the the quadratic coefficient is indeed 0.

In [BLS19, YAZ+19], one makes the observation that if the response can indeed be written as $\vec{\boldsymbol{z}} = \vec{\boldsymbol{y}} + \boldsymbol{c}\vec{\boldsymbol{r}}$, then one can write

$$\vec{\boldsymbol{z}} \circ (\vec{\boldsymbol{z}} - \boldsymbol{c}\vec{\boldsymbol{1}}) = \boldsymbol{c}^2 \vec{\boldsymbol{r}} \circ (\vec{\boldsymbol{r}} - \vec{\boldsymbol{1}}) + \boldsymbol{c}\vec{\boldsymbol{y}} \circ (2\vec{\boldsymbol{r}} - \vec{\boldsymbol{1}}) + \vec{\boldsymbol{y}} \circ \vec{\boldsymbol{y}},$$

where $\circ$ corresponds to component-wise multiplication. After additionally committing to "garbage terms" $\vec{\boldsymbol{y}} \circ (2\vec{\boldsymbol{r}} - \vec{\boldsymbol{1}})$ and $\vec{\boldsymbol{y}} \circ \vec{\boldsymbol{y}}$, the prover proceeds to show that the above equation is linear in $\boldsymbol{c}$, which means that the $\vec{\boldsymbol{r}} \circ (\vec{\boldsymbol{r}} - \vec{\boldsymbol{1}})$ term is 0, and thus all the coefficients of $\vec{\boldsymbol{r}}$ are 0/1. In order for this proof to go through, it's crucial for the $\vec{\boldsymbol{y}}$ to be fixed by the prover. In [BLS19, YAZ+19], this was done via an additional commitment and proof to $\vec{\boldsymbol{y}}$, which essentially doubled the size of the total proof.

An almost immediate consequence of our work would therefore result in a significant reduction of the proofs of [BLS19, YAZ+19]. We do not discuss this direction further, because with additional techniques, it is shown in a parallel submission [ENS20] how one can use the full product proof of commitments from the current paper to produce an even shorter proof. For this application (and others) we would need to consider the case where $X^d + 1$ fully splits into linear terms in $\mathcal{R}_q$, and therefore we can no longer assume that $\bar{\boldsymbol{c}}$ is invertible. So we continue to describe the ingredients needed here.

If $\bar{\boldsymbol{c}}$ is not invertible, then some NTT coefficient of $\bar{\boldsymbol{c}}$ is 0. In this case we would need to run the protocol in parallel to obtain extractions $(\bar{\boldsymbol{c}}_1, \vec{\bar{\boldsymbol{z}}}_1), \ldots, (\bar{\boldsymbol{c}}_\ell, \vec{\bar{\boldsymbol{z}}}_\ell)$ such that for every NTT coefficient, some $\bar{\boldsymbol{c}}_i$ in non-zero in that NTT coefficient. In this case, we can again prove that a valid prover knows a unique $\vec{\boldsymbol{r}}^*$ satisfying (7), and every $\vec{\boldsymbol{w}}$ is similarly a commitment to a $\vec{\boldsymbol{y}}^*$ satisfying (10). One could obtain such $\bar{\boldsymbol{c}}_i$ by sending several challenges in parallel, but for technical reasons (described in Section 5) having the challenges $\boldsymbol{c}_i$ related via specific algebraic particular automorphism operations results in smaller proofs. We now explain how the automorphisms are chosen.

When $X^d + 1$ splits into linear terms, one can also write $X^d + 1$ as in (6) where the multiplicative terms $X^k - r_i$ are not irreducible. In particular, we would like to consider such a factorization where $q^k \approx 2^{128}$ to have approximately 128 bits of soundness in the protocol. Then using the results on the distribution of $\boldsymbol{c} \bmod X^k - r_i$, we obtain that except with 2^{-128} probability, two $\boldsymbol{c}, \boldsymbol{c}'$ will not be equivalent modulo $X^k - r_i$. Since $X^k - r_i$ can be further factored as $X^k - r_i = \prod_{j=1}^{k}(X - r_j)$, this directly implies that one of these NTT coefficients will be distinct – in particular $(\boldsymbol{c} \neq \boldsymbol{c}' \bmod X - r_j)$ for some j. Then we define the automorphisms to be exactly those that cycle through the NTT coefficients represented by $X - r_j$, for $j = 1$ to k, and therefore for every NTT coefficient, one of the k automorphisms will result in $\bar{\boldsymbol{c}}$ being non-zero there.

The combination of these techniques, along with several key optimizations that minimize the number of necessary "garbage terms", results in a proof (described in Section 5) that is only a kilobyte longer (see Section 5.3) than just the opening proof in [BDL+18]. Furthermore, if one would like to prove many multiplicative relations, the size of the proof even further approaches the size of the proof from [BDL+18] because the extra elements needed in the proof amortize over all the proofs.

2 Preliminaries

2.1 Notation

As is often the case in ring-based lattice cryptography, computation will be performed in the ring $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d + 1)$, which is the quotient ring of the ring of integers $\mathcal{R}$ of the power-of-two $2d$-th cyclotomic number field modulo a rational prime $q \in Z$.

We use bold letters $\boldsymbol{f}$ for polynomials in $\mathcal{R}$ or $\mathcal{R}_q$, arrows for integer vectors $\vec{v}$ over $\mathbb{Z}_q$, bold letters with arrows $\vec{\boldsymbol{b}}$ for vectors of polynomials over $\mathcal{R}$ or $\mathcal{R}_q$ and capital letters A and $\boldsymbol{A}$ for integer and polynomial matrices, respectively. We write $x \xleftarrow{\$} S$ when $x \in S$ is sampled uniformly at random from the set S and similarly $x \xleftarrow{\$} D$ when x is sampled according to the distribution D.

For $\boldsymbol{f}, \boldsymbol{g} \in \mathcal{R}$, we have the coefficient norm

$$\|\boldsymbol{f}\|_2 = \left(\sum_{i=1}^{n} |f_i|^2\right)^{\frac{1}{2}}.$$

The norm is extended to vectors $\vec{\boldsymbol{v}} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k)$ of polynomials in the natural way,

$$\|\vec{\boldsymbol{v}}\|_2 = \left(\sum_{i=1}^{k} \|\boldsymbol{v}_i\|_2^2\right)^{\frac{1}{2}}.$$

2.2 Prime Splitting and Galois Automorphisms

Let l be a power of two dividing d and suppose $q-1 \equiv 2l \pmod{4l}$. Then, $\mathbb{Z}_q$ contains primitive $2l$-th roots of unity but no elements with order a higher power of two, and the polynomial X^d+1 factors into l irreducible binomials $X^{d/l} - \zeta$ modulo q where ζ runs over the $2l$-th roots of unity in $\mathbb{Z}_q$ [LS18, Theorem 2.3].

The ring $\mathcal{R}_q$ has a group of automorphisms $\mathsf{Aut}(\mathcal{R}_q)$ that is isomorphic to $\mathbb{Z}_{2d}^\times$,

$$i \mapsto \sigma_i \colon \mathbb{Z}_{2d}^\times \to \mathsf{Aut}(\mathcal{R}_q),$$

where σ_i is defined by $\sigma_i(X) = X^i$. In fact, these automorphisms come from the Galois automorphisms of the $2d$-th cyclotomic number field which factor through $\mathcal{R}_q$.

The group $\mathsf{Aut}(\mathcal{R}_q)$ acts transitively on the prime ideals $(X^{d/l} - \zeta)$ in $\mathcal{R}_q$ and every σ_i factors through field isomorphisms

$$\mathcal{R}_q/(X^{d/l} - \zeta) \to \mathcal{R}_q/(\sigma^i(X^{d/l} - \zeta)).$$

Concretely, for $i \in \mathbb{Z}_{2d}^\times$ it holds that

$$\sigma_i(X^{d/l} - \zeta) = (X^{id/l} - \zeta) = (X^{d/l} - \zeta^{i^{-1}})$$

To see this, observe that the roots of $X^{d/l} - \zeta^{i^{-1}}$ (in an appropriate extension field of $\mathbb{Z}_q$) are also roots of $X^{id/l} - \zeta$. Then, for $f \in \mathcal{R}_q$,

$$\sigma_i\left(f \bmod (X^{d/l} - \zeta)\right) = \sigma_i(f) \bmod (X^{d/l} - \zeta^{i^{-1}}).$$

The cyclic subgroup $\langle 2l+1 \rangle < \mathbb{Z}_{2d}^\times$ has order d/l [LS18, Lemma 2.4] and stabilizes every prime ideal $(X^{d/l} - \zeta)$ since ζ has order $2l$. The quotient group $\mathbb{Z}_{2d}^\times / \langle 2l+1 \rangle$ has order l and hence acts simply transitively on the l prime ideals. Therefore, we can index the prime ideals by $i \in \mathbb{Z}_{2d}^\times / \langle 2l+1 \rangle$ and write

$$(X^d + 1) = \prod_{i \in \mathbb{Z}_{2d}^\times / \langle 2l+1 \rangle} \left(X^{d/l} - \zeta^i\right)$$

Now, the product of the $k \mid l$ prime ideals $(X^{d/l} - \zeta^i)$ where i runs over $\langle 2l/k+1 \rangle / \langle 2l+1 \rangle$ is given by the ideal $(X^{kd/l} - \zeta^k)$. So, we can partition the l prime ideals into l/k groups of k ideals each, and write

$$(X^d + 1) = \prod_{j \in \mathbb{Z}_{2d}^\times / \langle 2l/k+1 \rangle} \left(X^{kd/l} - \zeta^{jk}\right) = \prod_{j \in \mathbb{Z}_{2d}^\times / \langle 2l/k+1 \rangle} \prod_{i \in \langle 2l/k+1 \rangle / \langle 2l+1 \rangle} \left(X^{\frac{d}{l}} - \zeta^{ij}\right).$$

Another way to write this, which we will use in our protocols, is to note that $\mathbb{Z}_{2d}^\times / \langle 2l/k+1 \rangle \cong \mathbb{Z}_{2l/k}^\times$ and the powers $(2l/k+1)^i$ for $i = 0, \ldots, k-1$ form a complete set of representatives for $\langle 2l/k+1 \rangle / \langle 2l+1 \rangle$. So, if $\sigma = \sigma_{2l/k+1} \in \mathsf{Aut}(\mathcal{R}_q)$, then

$$(X^d + 1) = \prod_{j \in \mathbb{Z}_{2l/k}^\times} \prod_{i=0}^{k-1} \sigma^i \left(X^{\frac{d}{l}} - \zeta^j\right),$$

and the prime ideals are indexed by $(i, j) \in I = \{0, \ldots, k-1\} \times \mathbb{Z}_{2l/k}^\times$.

2.3 Module SIS/LWE

We employ the computationally binding and computationally hiding commitment scheme from [BDL+18] in our protocols, and rely on the well-known Module-LWE (MLWE) and Module-SIS (MSIS) [PR06, LM06, LPR10, LS15] problems to prove the security of our constructions. Both problems are defined over a ring $\mathcal{R}_q$ for a positive modulus $q \in \mathbb{Z}^+$.

Definition 2.1 ($\mathsf{MSIS}_{n,m,\beta_{\mathrm{SIS}}}$)**.** *The goal in the* Module-SIS *problem with parameters $n, m > 0$ and $0 < \beta_{\mathrm{SIS}} < q$ is to find, for a given matrix $\boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{n\times m}$, $\vec{\boldsymbol{x}} \in \mathcal{R}_q^m$ such that $\boldsymbol{A}\vec{\boldsymbol{x}} = \vec{\boldsymbol{0}}$ over $\mathcal{R}_q$ and $0 < \|\vec{\boldsymbol{x}}\|_2 \leq \beta_{\mathrm{SIS}}$. We say that a PPT adversary $\mathcal{A}$ has advantage ϵ in solving $\mathsf{MSIS}_{n,m,\beta_{\mathrm{SIS}}}$ if*

$$\Pr\left[0 < \|\vec{\boldsymbol{x}}\|_2 \leq \beta_{\mathrm{SIS}} \wedge \boldsymbol{A}\vec{\boldsymbol{x}} = \vec{\boldsymbol{0}} \text{ over } \mathcal{R}_q \,\middle|\, \boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{n\times m};\ \vec{\boldsymbol{x}} \leftarrow \mathcal{A}(\boldsymbol{A})\right] \geq \epsilon.$$

Definition 2.2 ($\mathsf{MLWE}_{n,m,\chi}$)**.** *In the* Module-LWE *problem with parameters $n, m > 0$ and an error distribution χ over $\mathcal{R}$, the PPT adversary $\mathcal{A}$ is asked to distinguish $(\boldsymbol{A}, \vec{\boldsymbol{t}}) \xleftarrow{\$} \mathcal{R}_q^{m\times n} \times \mathcal{R}_q^m$ from $(\boldsymbol{A}, \boldsymbol{A}\vec{\boldsymbol{s}} + \vec{\boldsymbol{e}})$ for $\boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{m\times n}$, a secret vector $\vec{\boldsymbol{s}} \xleftarrow{\$} \chi^n$ and error vector $\vec{\boldsymbol{e}} \xleftarrow{\$} \chi^m$. We say that $\mathcal{A}$ has advantage ϵ in solving $\mathsf{MLWE}_{n,m,\chi}$ if*

$$\begin{aligned}\Big|\Pr\left[b = 1 \,\middle|\, \boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{m\times n};\ \vec{\boldsymbol{s}} \xleftarrow{\$} \chi^n;\ \vec{\boldsymbol{e}} \xleftarrow{\$} \chi^m;\ b \leftarrow \mathcal{A}(\boldsymbol{A}, \boldsymbol{A}\vec{\boldsymbol{s}} + \vec{\boldsymbol{e}})\right] \\ - \Pr\left[b = 1 \,\middle|\, \boldsymbol{A} \xleftarrow{\$} \mathcal{R}_q^{m\times n};\ \vec{\boldsymbol{t}} \xleftarrow{\$} \mathcal{R}_q^m;\ b \leftarrow \mathcal{A}(\boldsymbol{A}, \vec{\boldsymbol{t}})\right]\Big| \geq \epsilon.\end{aligned} \tag{11}$$

For our practical security estimations of these two problems against known attacks, the parameter m in both of the problems does not play a crucial role. Therefore, we sometimes simply omit m and use the notations $\mathsf{MSIS}_{n,B}$ and $\mathsf{MLWE}_{n,\chi}$. The parameters κ and λ denote the *module ranks* for MSIS and MLWE, respectively.

2.4 Error Distribution, Discrete Gaussians and Rejection Sampling

For sampling randomness in the commitment scheme that we use, and to define the particular variant of the Module-LWE problem that we use, we need to specify the error distribution χ^d on $\mathcal{R}$. In general any of the standard choices in the literature is fine. So, for example, χ can be a narrow discrete Gaussian distribution or the uniform distribution on a small interval. In the numerical examples in Section 5.3 we assume that χ is the computationally simple centered binomial distribution on $\{-1, 0, 1\}$ where ± 1 both have probability $5/16$ and 0 has probability $6/16$. This distribution is chosen (rather than the more "natural" uniform one) because it is easy to sample given a random bitstring by computing $a_1 + a_2 - b_1 - b_2 \bmod 3$ with uniformly random bits a_i, b_i.

Rejection Sampling. In our zero-knowledge proof, the prover will want to output a vector $\vec{\boldsymbol{z}}$ whose distribution should be independent of a secret randomness vector $\vec{\boldsymbol{r}}$, so that $\vec{\boldsymbol{z}}$ cannot be used to gain any information on the prover's secret. During the protocol, the prover computes $\vec{\boldsymbol{z}} = \vec{\boldsymbol{y}} + \boldsymbol{c}\vec{\boldsymbol{r}}$ where $\vec{\boldsymbol{r}}$ is the randomness used to commit to the prover's secret, $\boldsymbol{c} \xleftarrow{\$} C$ is a challenge polynomial, and $\vec{\boldsymbol{y}}$ is a "masking" vector. To remove the dependency of $\vec{\boldsymbol{z}}$ on $\vec{\boldsymbol{r}}$, we use the rejection sampling technique by Lyubashevsky [Lyu09, Lyu12]. In the two variants of this technique the masking vector is either sampled uniformly from some bounded region or using a discrete Gaussian distribution. In the high dimensions we will encounter, the Gaussian variant is far superior as it gives acceptable rejection probabilities for much narrower distributions. We first define the discrete Gaussian distribution and then state the rejection sampling algorithm in Figure 1, which plays a central role in Lemma 2.4.

Definition 2.3. *The* discrete Gaussian distribution *on $\mathcal{R}^\ell$ centered around $\vec{\boldsymbol{v}} \in \mathcal{R}^\ell$ with standard deviation $\mathfrak{s} > 0$ is given by*

$$D_{\vec{\boldsymbol{v}},\mathfrak{s}}^{\ell d}(\vec{\boldsymbol{z}}) = \frac{e^{-\|\vec{\boldsymbol{z}}-\vec{\boldsymbol{v}}\|_2^2/2\mathfrak{s}^2}}{\sum_{\vec{\boldsymbol{z}}'\in\mathcal{R}^\ell} e^{-\|\vec{\boldsymbol{z}}'\|_2^2/2\mathfrak{s}^2}}.$$

When it is centered around $\vec{\boldsymbol{0}} \in \mathcal{R}^\ell$ we write $D_{\mathfrak{s}}^{\ell d} = D_{\vec{\boldsymbol{0}},\mathfrak{s}}^{\ell d}$

Lemma 2.4 (Rejection Sampling). *Let $V \subseteq \mathcal{R}^\ell$ be a set of polynomials with norm at most T and $\rho\colon V \to [0,1]$ be a probability distribution. Also, write $\mathfrak{s} = 11T$ and $M = 3$. Now, sample $\vec{\boldsymbol{v}} \xleftarrow{\$} \rho$ and $\vec{\boldsymbol{y}} \xleftarrow{\$} D^{\ell d}_{\mathfrak{s}}$, set $\vec{\boldsymbol{z}} = \vec{\boldsymbol{y}} + \vec{\boldsymbol{v}}$, and run $b \leftarrow \mathsf{Rej}(\vec{\boldsymbol{z}}, \vec{\boldsymbol{v}}, \mathfrak{s})$ Then, the probability that $b = 0$ is at least $(1 - 2^{-100})/M$ and the distribution of $(\vec{\boldsymbol{v}}, \vec{\boldsymbol{z}})$, conditioned on $b = 0$, is within statistical distance of $2^{-100}/M$ of the product distribution $\rho \times D^{\ell d}_{\mathfrak{s}}$.*

$\underline{\mathrm{Rej}(\vec{\boldsymbol{z}}, \vec{\boldsymbol{v}}, \mathfrak{s})}$
01 $u \xleftarrow{\$} [0,1)$
02 If $u > \frac{1}{M} \cdot \exp\left(\frac{-2\langle \vec{\boldsymbol{z}}, \vec{\boldsymbol{v}}\rangle + \|\vec{\boldsymbol{v}}\|^2}{2\mathfrak{s}^2}\right)$
03 return 0
04 Else
05 return 1

Fig. 1. Rejection Sampling [Lyu12].

We will also use the following tail bound, which follows from [Ban93, Lemma 1.5(i)].

Lemma 2.5. *Let $\vec{\boldsymbol{z}} \xleftarrow{\$} D^{\ell d}_{\mathfrak{s}}$. Then*

$$\Pr\left[\|\vec{\boldsymbol{z}}\|_2 \leq \mathfrak{s}\sqrt{2\ell d}\right] \geq 1 - 2^{-\log(e/2)\ell d/4}.$$

2.5 Commitment Scheme

In our protocol, we use a variant of the commitment scheme from [BDL+18] which commits to a vector of messages in $\mathcal{R}_q$. Our basic proof of knowledge of multiplicative relations will prove that $\boldsymbol{m}_1\boldsymbol{m}_2 = \boldsymbol{m}_3$, so for simplicity, we just describe the commitment scheme for three messages.

The public parameters are a uniformly random matrix $\boldsymbol{B}_0 \in \mathcal{R}_q^{\mu \times (\lambda+\mu+3)}$ and uniform vectors $\vec{\boldsymbol{b}}_1, \ldots, \vec{\boldsymbol{b}}_3 \in \mathcal{R}_q^{\lambda+\mu+3}$. To commit to $\vec{\boldsymbol{m}} = (\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3)^T \in \mathcal{R}_q^3$, we choose a random short polynomial vector $\vec{\boldsymbol{r}} \xleftarrow{\$} \chi^{(\lambda+\mu+3)d}$ from the error distribution and output the commitment

$$\begin{aligned}
\vec{\boldsymbol{t}}_0 &= \boldsymbol{B}_0\vec{\boldsymbol{r}},\\
\boldsymbol{t}_1 &= \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{r}}\rangle + \boldsymbol{m}_1,\\
\boldsymbol{t}_2 &= \langle \vec{\boldsymbol{b}}_2, \vec{\boldsymbol{r}}\rangle + \boldsymbol{m}_2,\\
\boldsymbol{t}_3 &= \langle \vec{\boldsymbol{b}}_3, \vec{\boldsymbol{r}}\rangle + \boldsymbol{m}_3.
\end{aligned}$$

The commitment scheme is computationally hiding under the Module-LWE assumption and computationally binding under the Module-SIS assumption; see [BDL+18]. Moreover, the scheme is not only binding for the opening $(\vec{\boldsymbol{r}}, \vec{\boldsymbol{m}})$ known by the prover, but also binding with respect to a relaxed opening $(\bar{\boldsymbol{c}}, \vec{\boldsymbol{r}}*, \vec{\boldsymbol{m}}^*)$. The relaxed opening also includes a short polynomial $\bar{\boldsymbol{c}}$, the randomness vector $\vec{\boldsymbol{r}}^*$ is longer than $\vec{\boldsymbol{r}}$, and the following equations hold,

$$\begin{aligned}
\bar{\boldsymbol{c}}\vec{\boldsymbol{t}}_0 &= \boldsymbol{B}_0\vec{\boldsymbol{r}}^*,\\
\bar{\boldsymbol{c}}\boldsymbol{t}_1 &= \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{r}}^*\rangle + \bar{\boldsymbol{c}}\boldsymbol{m}_1^*,\\
\bar{\boldsymbol{c}}\boldsymbol{t}_2 &= \langle \vec{\boldsymbol{b}}_2, \vec{\boldsymbol{r}}^*\rangle + \bar{\boldsymbol{c}}\boldsymbol{m}_2^*,\\
\bar{\boldsymbol{c}}\boldsymbol{t}_3 &= \langle \vec{\boldsymbol{b}}_3, \vec{\boldsymbol{r}}^*\rangle + \bar{\boldsymbol{c}}\boldsymbol{m}_3^*.
\end{aligned}$$

The notion of relaxed opening is important since there is an efficient protocol for proving knowledge of a relaxed opening. We do not go into details here since we will define a new notion of a binding relaxed opening and provide a proof of knowledge protocol.

The utility of the commitment scheme for zero-knowledge proof systems stems from the fact that one can compute module homomorphisms on committed messages. For example, let $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ be from $\mathcal{R}_q$. Then

$$\boldsymbol{a}_1\boldsymbol{t}_1 + \boldsymbol{a}_2\boldsymbol{t}_2 = \langle \boldsymbol{a}_1\vec{\boldsymbol{b}}_1 + \boldsymbol{a}_2\vec{\boldsymbol{b}}_2, \vec{\boldsymbol{r}}\rangle + \boldsymbol{a}_1\boldsymbol{m}_1 + \boldsymbol{a}_2\boldsymbol{m}_2$$

is a commitment to the message $\boldsymbol{a}_1\boldsymbol{m}_1+\boldsymbol{a}_2\boldsymbol{m}_2$ with matrix $\boldsymbol{a}_1\vec{\boldsymbol{b}}_1+\boldsymbol{a}_2\vec{\boldsymbol{b}}_2$. This module homomorphic property together with a proof that a commitment is a commitment to the zero polynomial allows to prove linear relations among committed messages over $\mathcal{R}_q$.

3 Distribution in the NTT

In this section we present a way to construct challenge sets $\mathcal{C} \subset \mathcal{R}_q$ so as to be able to compute the (almost exact) probability that $\boldsymbol{c} - \boldsymbol{c}'$ is invertible in $\mathcal{R}_q$, when $\boldsymbol{c}$ and $\boldsymbol{c}'$ are sampled from some distribution C over $\mathcal{C}$. Recall that $d \geq l$ are powers of 2. Moreover,

$$\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d+1) \cong \prod_{i\in\mathbb{Z}_{2l}^\times} \mathbb{Z}_q[X]/(X^{d/l} - \zeta^i), \tag{12}$$

where $\zeta \in \mathbb{Z}_q$ is a $2l$-th root of unity (in this section, the factors $X^{d/l} - \zeta^i$ are not necessarily irreducible as this doesn't really matter for the results here). The challenge set is defined as all degree d polynomials with coefficients in $\{-1, 0, 1\}$, i.e., $\mathcal{C} = \{-1, 0, 1\}^d \subset \mathcal{R}_q$. The coefficients of a challenge $\boldsymbol{c} \in \mathcal{C}$ are independently and identically distributed, where 0 has probability p and ± 1 both have probability $(1 - p)/2$. For the resulting distribution over $\mathcal{C}$ we write C, and sampling a challenge $\boldsymbol{c}$ from this distribution is written as $\boldsymbol{c} \leftarrow C$.

In the remainder of this section we use Fourier analysis to study the distribution of $\boldsymbol{c} \mod X^{d/l} - \zeta^i$ for $\boldsymbol{c} \leftarrow C$ and $i \in \mathbb{Z}_q^\times$. Lemma 3.1 shows that this distribution does not depend on i.

In [CLS16] a similar analysis is performed. The main differences with our approach is that they sample the coefficients from a binomial distribution centered at 0. In particular, our coefficient distribution with $p = 1/2$ corresponds to a special case of the binomial distribution considered in [CLS16]. For our application it makes sense to consider various distributions over $\{-1, 0, 1\}$. The binomial distribution does allow for the derivation of an elegant upper bound on the maximum probability of $\boldsymbol{c} \mod X^{d/l} - \zeta^i$. However, this upper bound is only applicable when $\sqrt{q} \leq 2d$. For this reason we derive a less elegant but much tighter upper bound on various distributions over $\{-1, 0, 1\}$, that are also applicable when $\sqrt{q} > 2d$.

Lemma 3.1. *Let $\boldsymbol{x} \in \mathcal{R}_q$ be a random polynomial with coefficients independently and identically distributed. Then $\mathcal{R}_q/(X^{d/l} - \zeta^i) \cong \mathcal{R}_q/(X^{d/l} - \zeta^j)$, and $\boldsymbol{x} \bmod (X^{d/l} - \zeta^i)$ and $\boldsymbol{x} \bmod (X^{d/l} - \zeta^j)$ are identically distributed for all $i, j \in \mathbb{Z}_{2l}^\times$.*

Proof. First suppose that $X^{d/l} - \zeta^i$ is irreducible for all $i \in \mathbb{Z}_{2l}^\times$. Then $\mathfrak{q}_i = (q, X^{d/l} - \zeta^i)$ is prime in $K = \mathbb{Q}[X]/(X^d+1)$ and for all $i, j \in \mathbb{Z}_{2l}^\times$ there exists an automorphism $\sigma \in \mathrm{Gal}\,(K/\mathbb{Q})$ such that $\sigma(\mathfrak{q}_i) = \mathfrak{q}_j$. Hence, σ induces an isomorphism between the finite fields $\mathcal{R}_q/(X^{d/l} - \zeta^i)$ and $\mathcal{R}_q/(X^{d/l} - \zeta^j)$.

Since the coefficients of $\boldsymbol{x}$ are i.i.d., it holds that $\sigma(\boldsymbol{x})$ follows the same distribution over R_q as $\boldsymbol{x}$. Hence, $\boldsymbol{x} \mod (X^{d/l} - \zeta^i)$ follows the same distribution as $\sigma(\boldsymbol{x} \mod (X^{d/l} - \zeta^i)) = \sigma(\boldsymbol{x}) \mod (X^{d/l} - \zeta^j)$ and as $\boldsymbol{x} \mod (X^{d/l} - \zeta^j)$ which proves the lemma for this case.

Now suppose that $X^{d/l} - \zeta^i$ is reducible in $\mathbb{Z}_q$, then so is $X^{d/l} - \zeta^j$. Moreover, since K is Galois both these polynomials split in the same number irreducible factors and for every pair $f(X), g(X)$ of irreducible factors there exists an automorphism $\sigma \in \mathrm{Gal}(K/\mathbb{Q})$ such that $\sigma\left((q, f(X))\right) = (q, g(X))$. Using these automorphisms the lemma follows in an analogous manner.

Let us now consider the coefficients of the polynomial $\boldsymbol{c} \mod (X^{d/l} - \zeta)$ for $\boldsymbol{c} \leftarrow C$. Clearly all coefficients follow the same distribution over $\mathbb{Z}_q$. Let us write Y for the random variable over $\mathbb{Z}_q$ that follows this distribution. The following lemma gives an upper bound on the maximum probability of Y.

Lemma 3.2. *Let the random variable Y over $\mathbb{Z}_q$ be defined as above. Then for all $x \in \mathbb{Z}_q$,*

$$\Pr(Y = x) \leq M := \frac{1}{q} + \frac{1}{q} \sum_{j \in \mathbb{Z}_q^\times} \prod_{k=0}^{l-1} \left| p + (1-p)\cos(2\pi j \zeta^k / q) \right|. \tag{13}$$

Proof. From Fourier analysis (see, e.g., [Dia88]) we find that

$$\begin{aligned} P(x) &:= \Pr(Y = x), \\ &= \frac{1}{q} + \frac{1}{q} \sum_{j \in \mathbb{Z}_q^\times} \widehat{P}(j) \exp\left(-2\pi i j x / q\right), \end{aligned} \tag{14}$$

where $\widehat{P}$ is the Fourier transform of $P : \mathbb{Z}_q \to [0,1]$. Moreover, the probability distribution P is the convolution of the distributions μ_k $(0 \leq k \leq l-1)$ with corresponding Fourier transforms $\widehat{\mu}_k$, where

$$\begin{aligned} \mu_k(0) = p, \quad \mu_k(\zeta^k) = \mu_k(-\zeta^k) = (1-p)/2, \\ \widehat{\mu}_k : \mathbb{Z}_q \to \mathbb{C}, \quad j \mapsto p + (1-p)\cos\left(2\pi j \zeta^k / q\right). \end{aligned} \tag{15}$$

Hence, from Fourier theory, it follows that

$$\widehat{P}(j) = \prod_{k=0}^{l-1} \widehat{\mu}_k(j), \tag{16}$$

and therefore that

$$P(x) = \frac{1}{q} + \frac{1}{q} \sum_{j \in \mathbb{Z}_q^\times} \prod_{k=0}^{l-1} \widehat{\mu}_k(j) \exp\left(-2\pi i j x / q\right), \tag{17}$$

Taking absolute values on both sides and applying the triangle inequality now proves the lemma.

The following lemma shows that, by utilizing certain algebraic symmetries, we can reduce the number of terms in the summation of Lemma 3.2 by a factor $2l$, thereby allowing the maximum probability to be computed more efficiently.

Lemma 3.3. *Let the random variable Y over $\mathbb{Z}_q$ be defined as above. Then for all $x \in \mathbb{Z}_q$,*

$$\Pr(Y = x) \leq M := \frac{1}{q} + \frac{2l}{q} \sum_{j \in \mathbb{Z}_q^\times / \langle\zeta\rangle} \prod_{k=0}^{l-1} \left| p + (1-p)\cos(2\pi j y \zeta^k / q) \right|. \tag{18}$$

Proof. Let $a, b \in \mathbb{Z}_q^\times$ such that $ab^{-1} \in \langle\zeta\rangle$, i.e., $a = b\zeta^m$ for some m. Now note that $\{1, \zeta, \ldots, \zeta^{l-1}\} = \langle\zeta\rangle / \pm 1 = \zeta^m \langle\zeta\rangle / \pm 1$ for all $m \in \mathbb{Z}$. Since $\cos(x)$ is an even function it therefore follows that $\widehat{P}(a) = \widehat{P}(b)$, from which the lemma immediately follows.

The random variable $Y = Y_l$ corresponds to a random walk of length l over $\mathbb{Z}_q$ defined as follows

$$Y_0 = 0, \quad Y_n = \zeta Y_{n-1} + b_n, \tag{19}$$

where b_n are i.i.d. with distribution $\mu(0) = p$ and $\mu(1) = \mu(-1) = (1-p)/2$. Random walks of this type have been studied extensively [CDG87, Dia88, Hil90, Hil06, BV19] and convergence is expected in time $O(\log q / H_2(\mu))$ [BV19], where

$$H_2(\mu) := -\log\left(\sum_{x\in\mathbb{Z}_q} \mu(x)^2\right). \tag{20}$$

However, there exist random walks of this form for which convergence only occurs in time $O(\log q \log\log q)$ [Dia88, Hil06].

Let us consider the following example. Let q be the 32-bit prime $4294962689 =$ mod 1 mod 512 and $d \mid 256$ the dimension of the ring $\mathcal{R}$. Then, for any d, q splits completely in $\mathbb{Z}[X]/(X^d+1)$, hence in this case $l = d$. Moreover, suppose that the coefficients of challenges are sampled from a uniform distribution over $\{-1, 0, 1\}$, i.e., $p = 1/3$. Table 1 shows a bound M on the maximum probability $\max_{x\in\mathbb{Z}_q} |\Pr(Y = x)|$, as defined in Lemma 3.2 and Lemma 3.3.

Table 1. Maximum probability for the coefficients of challenges $\boldsymbol{c} \leftarrow C$ when reduced modulo $(X-\zeta)$ ($q = 4294962689$ and $p = 1/3$).

Dimension d	1	2	4	8	16	32	64
$\log_2(M)$	-1.06	-2.13	-4.25	-8.50	-17.01	-31.69	$\approx -\log_2(q)$

4 Opening Proof

Suppose the prover knows an opening to the commitment

$$\begin{aligned}\vec{\boldsymbol{t}}_0 &= \boldsymbol{B}_0\vec{\boldsymbol{r}},\\ \boldsymbol{t}_1 &= \langle\vec{\boldsymbol{b}}_1, \vec{\boldsymbol{r}}\rangle + \boldsymbol{m}.\end{aligned}$$

The standard protocol for proving this, stemming from [BDL+18], works by giving an approximate proof for the first equation $\vec{\boldsymbol{t}}_0 = \boldsymbol{B}_0\vec{\boldsymbol{r}}$. So, the prover commits to a short masking vector $\vec{\boldsymbol{y}}$ from a discrete Gaussian distribution by sending $\vec{\boldsymbol{w}} = \boldsymbol{B}_0\vec{\boldsymbol{y}}$. Then the verifier sends a short challenge polynomial $\boldsymbol{c} \in \mathcal{C} \subset \mathcal{R}$ and the prover replies with the short vector $\vec{\boldsymbol{z}} = \vec{\boldsymbol{y}} + \boldsymbol{c}\vec{\boldsymbol{r}}$. Here rejection sampling is used to make the distribution of $\vec{\boldsymbol{z}}$ independent from $\vec{\boldsymbol{r}}$. The verifier checks that $\vec{\boldsymbol{z}}$ is short, i.e. $\|\vec{\boldsymbol{z}}\|_2 \le \beta$, and the equation $\boldsymbol{B}_0\vec{\boldsymbol{z}} = \vec{\boldsymbol{w}} + \boldsymbol{c}\vec{\boldsymbol{t}}_0$.

For suitable instantiations this proves knowledge of a commitment opening because it is possible to extract two prover replies $\vec{\boldsymbol{z}}$ and $\vec{\boldsymbol{z}}'$ for two challenges $\boldsymbol{c}$ and $\boldsymbol{c}'$, respectively, and a message $\boldsymbol{m}^* \in \mathcal{R}_q$ such that

$$\begin{aligned}\bar{\boldsymbol{c}}\vec{\boldsymbol{t}}_0 &= \boldsymbol{B}_0(\vec{\boldsymbol{z}} - \vec{\boldsymbol{z}}'),\\ \bar{\boldsymbol{c}}\boldsymbol{t}_1 &= \langle\vec{\boldsymbol{b}}_1, \vec{\boldsymbol{z}} - \vec{\boldsymbol{z}}'\rangle + \bar{\boldsymbol{c}}\boldsymbol{m}^*,\end{aligned}$$

where $\bar{\boldsymbol{c}} = \boldsymbol{c} - \boldsymbol{c}'$ is the difference of the challenges. In fact, it can be shown [BDL+18] that the commitment scheme is binding with respect to the message $\boldsymbol{m}^*$ under the Module-SIS assumption if we have the additional property that $\bar{\boldsymbol{c}}$ is invertible in the ring $\mathcal{R}_q$. Then, it must be that $\boldsymbol{m}^* = \boldsymbol{m}$, unless the prover knows a Module-SIS solution for $\boldsymbol{B}_0$. The invertibility property is crucial in all previous works that study zero-knowledge proofs for the commitment scheme. It is enforced by choosing the set $\mathcal{C}$ of challenges such that the difference of every two distinct elements is invertible. Unfortunately, depending on how much the prime q splits in the ring $\mathcal{R}$, there will not be sufficiently large sets with this property, and even less so large sets consisting of *short* polynomials. For instance, for both theoretical and practical reasons one often wants q to split completely, but then there can be at most q polynomials which are pairwise different modulo one of the

degree 1 prime divisors of q. Even if we let q split slightly less, say in degree 4 prime ideals, then we do not know of large sets of short polynomials that do not collide modulo one of the divisors. This severely restricts the soundness of the protocol and the protocol has to be repeated several times to boost soundness, which blows up the proof size. See [LS18] for more details about this problem.

The results from Section 3 present a way to construct larger challenge sets with the weaker property that $\bar{\boldsymbol{c}}$ is non-invertible only with negligible probability. We generalize the proof further and explain how it is possible to make use of challenge sets where the difference of two elements is non-invertible with non-negligible probability.

So, in the extraction, we drop the assumption that for a pair of accepting transcripts with different challenges $\boldsymbol{c}$ and $\boldsymbol{c}'$, the difference $\bar{\boldsymbol{c}} = \boldsymbol{c} - \boldsymbol{c}'$ is invertible. This essentially means that we can not uniquely interpolate the prover replies $\vec{\boldsymbol{z}}$ and $\vec{\boldsymbol{z}}'$, and obtain vectors $\vec{\boldsymbol{y}}^*$ and $\vec{\boldsymbol{r}}^*$ such that

$$\vec{\boldsymbol{z}} = \vec{\boldsymbol{y}}^* + \boldsymbol{c}\vec{\boldsymbol{r}}^* \quad \text{and} \quad \vec{\boldsymbol{z}}' = \vec{\boldsymbol{y}}^* + \boldsymbol{c}'\vec{\boldsymbol{r}}^*. \tag{21}$$

But we can restore the interpolation by piecing together several transcript pairs that we interpolate locally modulo the various prime ideals dividing q.

Let $X^d + 1 \equiv \boldsymbol{\varphi}_1 \dots \boldsymbol{\varphi}_l \pmod q$ be the factorization of $X^d + 1$ into irreducible polynomials modulo q. Thus, our ring $\mathcal{R}_q$ is the product of the corresponding residue fields $\kappa_i = \mathbb{Z}_q[X]/(\boldsymbol{\varphi}_i)$, i.e.

$$\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d + 1) = \mathbb{Z}_q[X]/(\boldsymbol{\varphi}_1) \times \cdots \times \mathbb{Z}_q[X]/(\boldsymbol{\varphi}_l).$$

Now, what is needed specifically is that for every i there is an accepting transcript pair with nonzero challenge difference $\bar{\boldsymbol{c}}$ modulo $\boldsymbol{\varphi}_i$. So, concretely, suppose the extractor $\mathcal{E}$ has obtained l pairs $(\vec{\boldsymbol{z}}_i, \vec{\boldsymbol{z}}'_i)$, $i = 1, \dots, l$, of replies from the prover $\mathcal{P}$ for the challenge pairs $(\boldsymbol{c}_i, \boldsymbol{c}'_i)$, respectively, such that

$$\bar{\boldsymbol{c}}_i = \boldsymbol{c}_i - \boldsymbol{c}'_i \not\equiv 0 \pmod{\boldsymbol{\varphi}_i}.$$

We also assume that all transcripts contain the same prover commitment $\vec{\boldsymbol{w}}$ and are accepting; that is, in particular, $\boldsymbol{B}_0\vec{\boldsymbol{z}}_i = \vec{\boldsymbol{w}} + \boldsymbol{c}_i\vec{\boldsymbol{t}}_0$ and $\boldsymbol{B}_0\vec{\boldsymbol{z}}'_i = \vec{\boldsymbol{w}} + \boldsymbol{c}'_i\vec{\boldsymbol{t}}_0$ for all i. From this data $\mathcal{E}$ computes the local interpolations

$$\vec{\boldsymbol{z}}_i \equiv \vec{\boldsymbol{y}}^*_i + \boldsymbol{c}_i\vec{\boldsymbol{r}}^*_i \quad \text{and} \quad \vec{\boldsymbol{z}}'_i \equiv \vec{\boldsymbol{y}}^*_i + \boldsymbol{c}'_i\vec{\boldsymbol{r}}^*_i \pmod{\boldsymbol{\varphi}_i}.$$

Concretely, we set

$$\vec{\boldsymbol{r}}^*_i = \frac{\vec{\boldsymbol{z}}_i - \vec{\boldsymbol{z}}'_i}{\bar{\boldsymbol{c}}_i} \bmod \boldsymbol{\varphi}_i, \text{ and}$$
$$\vec{\boldsymbol{y}}^*_i = \frac{\boldsymbol{c}_i\vec{\boldsymbol{z}}'_i - \boldsymbol{c}'_i\vec{\boldsymbol{z}}_i}{\bar{\boldsymbol{c}}_i} \bmod \boldsymbol{\varphi}_i.$$

Now, let $\vec{\boldsymbol{r}}^*$ and $\vec{\boldsymbol{y}}^*$ over $\mathcal{R}_q$ be the CRT lifting of the $\vec{\boldsymbol{r}}^*_i$ and $\vec{\boldsymbol{y}}^*_i$. We show it must hold that

$$\vec{\boldsymbol{z}}_i = \vec{\boldsymbol{y}}^* + \boldsymbol{c}_i\vec{\boldsymbol{r}}^* \quad \text{and} \quad \vec{\boldsymbol{z}}'_i = \vec{\boldsymbol{y}}^* + \boldsymbol{c}'_i\vec{\boldsymbol{r}}^*$$

for all i. This restores the global interpolations as in Equation 21. In fact, we show more than this. Namely that in every accepting transcript with commitment $\vec{\boldsymbol{w}}$, the prover reply must be precisely of the form in Equation 21. Also the vectors $\vec{\boldsymbol{r}}^*$ and $\vec{\boldsymbol{y}}^*$ are preimages of $\vec{\boldsymbol{t}}_0$ and $\vec{\boldsymbol{w}}$, respectively, which is what we suspect. So the prover really is committed to $\vec{\boldsymbol{r}}^*$ and $\vec{\boldsymbol{y}}^*$ by $\vec{\boldsymbol{t}}_0$ and $\vec{\boldsymbol{w}}$.

Lemma 4.1. *If we have obtained l pairs of accepting transcripts with commitment $\vec{\boldsymbol{w}}$ as in the preceding paragraph, then every accepting transcript $(\vec{\boldsymbol{w}}, \boldsymbol{c}, \vec{\boldsymbol{z}})$ with commitment $\vec{\boldsymbol{w}}$ must be such that $\vec{\boldsymbol{z}} = \vec{\boldsymbol{y}}^* + \boldsymbol{c}\vec{\boldsymbol{r}}^*$ where $\vec{\boldsymbol{y}}^*$ and $\vec{\boldsymbol{r}}^*$ are the vectors computed above independently from $\boldsymbol{c}$, or we obtain an $\mathsf{MSIS}_{\mu,8\kappa\beta}$ solution for $\boldsymbol{B}_0$. Moreover, we have $\boldsymbol{B}_0\vec{\boldsymbol{r}}^* = \vec{\boldsymbol{t}}_0$ and $\boldsymbol{B}_0\vec{\boldsymbol{y}}^* = \vec{\boldsymbol{w}}$.*

Proof. Define $\vec{\boldsymbol{y}}^{*\prime}$ by $\vec{\boldsymbol{z}} = \vec{\boldsymbol{y}}^{*\prime} + \boldsymbol{c}\vec{\boldsymbol{r}}^{*}$. Fix some $i \in \{1, \dots, l\}$. Since all transcripts are accepting we get from subtracting the verification equations,

$$\begin{aligned}\boldsymbol{B}_0(\vec{\boldsymbol{z}}_i - \vec{\boldsymbol{z}}_i') &= \bar{\boldsymbol{c}}_i\vec{\boldsymbol{t}}_0, \text{ and}\\ \boldsymbol{B}_0(\vec{\boldsymbol{z}} - \vec{\boldsymbol{z}}_i) &= (\boldsymbol{c} - \boldsymbol{c}_i)\vec{\boldsymbol{t}}_0.\end{aligned}$$

Now, cross-multiplying by $\bar{\boldsymbol{c}}_i$ and $\boldsymbol{c} - \boldsymbol{c}_i$ and subtracting shows that we either have an $\mathsf{MSIS}_{\mu,8\kappa\beta}$ solution for $\boldsymbol{B}_0$, or

$$\bar{\boldsymbol{c}}_i(\vec{\boldsymbol{z}} - \vec{\boldsymbol{z}}_i) = (\boldsymbol{c} - \boldsymbol{c}_i)(\vec{\boldsymbol{z}}_i - \vec{\boldsymbol{z}}_i').$$

Suppose the latter case is true. Then we reduce modulo $\boldsymbol{\varphi}_i$ and substitute the local expressions for $\vec{\boldsymbol{z}}$, $\vec{\boldsymbol{z}}_i$ and $\vec{\boldsymbol{z}}_i'$, which shows

$$\begin{aligned}&\bar{\boldsymbol{c}}_i(\vec{\boldsymbol{y}}^{*\prime} - \vec{\boldsymbol{y}}_i^{*} + (\boldsymbol{c} - \boldsymbol{c}_i)\vec{\boldsymbol{r}}_i^{*}) \equiv (\boldsymbol{c} - \boldsymbol{c}_i)\bar{\boldsymbol{c}}_i\vec{\boldsymbol{r}}_i^{*} \pmod{\boldsymbol{\varphi}_i}\\ &\Leftrightarrow \bar{\boldsymbol{c}}_i(\vec{\boldsymbol{y}}^{*\prime} - \vec{\boldsymbol{y}}_i^{*}) \equiv 0 \pmod{\boldsymbol{\varphi}_i}.\end{aligned}$$

Since $\bar{\boldsymbol{c}}_i \bmod \boldsymbol{\varphi}_i \neq 0$, $\vec{\boldsymbol{y}}^{*\prime} \equiv \vec{\boldsymbol{y}}_i^{*} \equiv \vec{\boldsymbol{y}}^{*}$ modulo $\boldsymbol{\varphi}_i$. This holds for all i and hence it follows that $\vec{\boldsymbol{y}}^{*\prime} = \vec{\boldsymbol{y}}^{*}$.

We come to the statements $\boldsymbol{B}_0\vec{\boldsymbol{r}}^{*} = \vec{\boldsymbol{t}}_0$ and $\boldsymbol{B}_0\vec{\boldsymbol{y}}^{*} = \vec{\boldsymbol{w}}$. From the construction of $\vec{\boldsymbol{r}}^{*}$ and the verification equations it follows that

$$\begin{aligned}\boldsymbol{B}_0\vec{\boldsymbol{r}}^{*} &\equiv \boldsymbol{B}_0\vec{\boldsymbol{r}}_i^{*}\\ &\equiv \boldsymbol{B}_0\frac{\vec{\boldsymbol{z}}_i - \vec{\boldsymbol{z}}_i'}{\bar{\boldsymbol{c}}_i}\\ &\equiv \vec{\boldsymbol{t}}_0 \pmod{\boldsymbol{\varphi}_i}\end{aligned}$$

for all i. Similarly, for $\vec{\boldsymbol{y}}^{*}$,

$$\begin{aligned}\boldsymbol{B}_0\vec{\boldsymbol{y}}^{*} &\equiv \boldsymbol{B}_0\vec{\boldsymbol{y}}_i^{*}\\ &\equiv \boldsymbol{B}_0\frac{\boldsymbol{c}_i\vec{\boldsymbol{z}}_i' - \boldsymbol{c}_i'\vec{\boldsymbol{z}}_i}{\bar{\boldsymbol{c}}_i}\\ &\equiv \vec{\boldsymbol{w}} \pmod{\boldsymbol{\varphi}_i}.\end{aligned}$$

The statements in the lemma follow from the Chinese remainder theorem. □

Finally, the extracted vector $\vec{\boldsymbol{r}}^{*}$ can be used to define a binding notion of opening for the commitment scheme where the extracted message $\boldsymbol{m}^{*}$ is simply set to fulfill

$$\boldsymbol{t}_1 = \langle\vec{\boldsymbol{b}}_1, \vec{\boldsymbol{r}}^{*}\rangle + \boldsymbol{m}^{*}.$$

Then we have found an instance of the following definition.

Definition 4.2. *A* weak opening *for the commitment* $\vec{\boldsymbol{t}} = \vec{\boldsymbol{t}}_0 \parallel \boldsymbol{t}_1$ *consists of* l *polynomials* $\bar{\boldsymbol{c}}_i \in \mathcal{R}_q$*, a randomness vector* $\vec{\boldsymbol{r}}^{*}$ *over* $\mathcal{R}_q$ *and a message* $\boldsymbol{m}^{*} \in \mathcal{R}_q$ *such that*

$$\begin{aligned}&\|\bar{\boldsymbol{c}}_i\|_1 \leq 2\kappa \text{ and } \bar{\boldsymbol{c}}_i \bmod \boldsymbol{\varphi}_i \neq 0 \text{ for all } 1 \leq i \leq l,\\ &\|\bar{\boldsymbol{c}}_i\vec{\boldsymbol{r}}^{*}\|_2 \leq 2\beta \text{ for all } 1 \leq i \leq l,\\ &\boldsymbol{B}_0\vec{\boldsymbol{r}}^{*} = \vec{\boldsymbol{t}}_0,\\ &\langle\vec{\boldsymbol{b}}_1, \vec{\boldsymbol{r}}^{*}\rangle + \boldsymbol{m}^{*} = \boldsymbol{t}_1.\end{aligned}$$

It is easy to show that the commitment scheme is binding with respect to these weak openings.

Lemma 4.3. *The commitment scheme is binding with respect to weak openings if* $\mathsf{MSIS}_{\mu,8\kappa\beta}$ *is hard. More precisely, from two different weak openings* $((\bar{\boldsymbol{c}}_i), \vec{\boldsymbol{r}}^{*}, \boldsymbol{m}^{*})$ *and* $((\bar{\boldsymbol{c}}_i), \vec{\boldsymbol{r}}^{*\prime}, \boldsymbol{m}^{*\prime})$ *with* $\boldsymbol{m}^{*} \neq \boldsymbol{m}^{*\prime}$ *one can immediately compute a Module-SIS solution for* $\boldsymbol{B}_0$ *of length at most* $8\kappa\beta$*.*

Proof. Suppose there are two weak openings $((\bar{\boldsymbol{c}}_i), \vec{\boldsymbol{r}}^*, \boldsymbol{m}^*)$ and $((\bar{\boldsymbol{c}}'_i), \vec{\boldsymbol{r}}^{*\prime}, \boldsymbol{m}^{*\prime})$ with $\boldsymbol{m}^* \neq \boldsymbol{m}^{*\prime}$. Then, $\langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{r}}^* \rangle + \boldsymbol{m}^* = \boldsymbol{t}_1 = \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{r}}^{*\prime} \rangle + \boldsymbol{m}^{*\prime}$ implies $\vec{\boldsymbol{r}}^* \neq \vec{\boldsymbol{r}}^{*\prime}$. Therefore, there exists an $i \in \{1, \ldots, l\}$ such that $\vec{\boldsymbol{r}}^* \not\equiv \vec{\boldsymbol{r}}^{*\prime} \pmod{\boldsymbol{\varphi}_i}$. Consequently, $\bar{\boldsymbol{c}}_i \bar{\boldsymbol{c}}'_i (\vec{\boldsymbol{r}}^* - \vec{\boldsymbol{r}}^{*\prime}) = \bar{\boldsymbol{c}}'_i \bar{\boldsymbol{c}}_i \vec{\boldsymbol{r}}^* - \bar{\boldsymbol{c}}_i \bar{\boldsymbol{c}}'_i \vec{\boldsymbol{r}}^{*\prime} \neq 0$ since the polynomials $\boldsymbol{c}_i$ and $\boldsymbol{c}'_i$ are non-zero modulo $\boldsymbol{\varphi}_i$. Hence,

$$\boldsymbol{B}_0 \bar{\boldsymbol{c}}_i \bar{\boldsymbol{c}}'_i (\vec{\boldsymbol{r}}^* - \vec{\boldsymbol{r}}^{*\prime}) = 0$$

is a non-trivial Module-SIS solution for $\boldsymbol{B}_0$ of length at most $8\kappa\beta$. □

It remains to explain how we make it possible to arrive at the transcript pairs that we want to piece together. Suppose $\mathcal{R}_q$ factors in the following way,

$$\mathcal{R}_q = \prod_{i \in \mathbb{Z}_{2l}^\times} \mathbb{Z}_q[X]/(X^{\frac{d}{l}} - \zeta^i)$$

with l irreducible $\boldsymbol{\varphi}_i = X^{d/l} - \zeta^i$ and ζ a primitive $2l$-th root of unity. Let $\mathcal{C} = \{-1, 0, 1\}^d \subset \mathcal{R}$ and $\boldsymbol{c} \in \mathcal{C}$ be a random element from $\mathcal{C}$ where each coefficient is independently identically distributed with $\Pr(0) = 1/2$ and $\Pr(-1) = \Pr(1) = 1/4$. Then the d/l coefficients of $\boldsymbol{c} \bmod \boldsymbol{\varphi}_i$ for a fixed i are mutually independent and Lemma 3.3 gives a bound on their maximum probability over $\mathbb{Z}_q$. We will set parameters such that the maximum probability is not much bigger than $1/q$. Then the probability that a cheating prover can get away with only answering challenges with a particular value modulo $\boldsymbol{\varphi}_i$ is about $q^{-d/l}$. If this probability is negligible, then, although the projections $\boldsymbol{c} \bmod \boldsymbol{\varphi}_i$ for varying i are not independent, we can get several transcript pairs where for each i at least one $\bar{\boldsymbol{c}} \bmod \boldsymbol{\varphi}_i$ is non-zero. This works by rewinding the prover l times, once for every i, and sending a challenge that differs from a previous successful challenge modulo $\boldsymbol{\varphi}_i$. If otherwise the probability $q^{-d/l}$ is not negligible we can run several, say k, copies of the protocol in parallel and reduce the cheating probability to $q^{-kd/l}$. Then there are k prover commitments $\vec{\boldsymbol{w}}_i$ in the first flow and there won't be l accepting transcript pairs for each of them. Hence this requires a slightly more general analysis than what we have provided in the overview in this section. We handle this case in the security proof of our protocol given in Figure 2. It turns out that it is still possible to extract unique preimages $\vec{\boldsymbol{y}}_i$ for all commitments $\vec{\boldsymbol{w}}_i$.

In the k parallel repetitions we do not sample the challenges independently. The reason is that when proving relations on the messages and specifically in our product proof we will need more structure. Let $\sigma = \sigma_{2l/k+1} \in \mathrm{Aut}(\mathcal{R}_q) \cong \mathbb{Z}_{2d}^\times$ be the automorphism of order kd/l that stabilizes the ideals

$$\left(X^{\frac{kd}{l}} - \zeta^{jk}\right) = \prod_{i=0,\ldots,k-1} \sigma^i \left(X^{\frac{d}{l}} - \zeta^j\right) = \prod_{i \in \langle 2l/k+1 \rangle / \langle 2l+1 \rangle} \left(X^{\frac{d}{l}} - \zeta^{ij}\right)$$

for $j \in \langle -1, 5 \rangle / \langle 2l/k + 1 \rangle \cong \mathbb{Z}_{2l/k}^\times$. Now, we let the challenges in the k parallel executions be the images $\sigma^i(\boldsymbol{c})$, $i = 0, \ldots, k-1$, of a single polynomial $\boldsymbol{c} \in \mathcal{C}$. If parameters are such that the maximum probability of each of the mutually independent coefficients of $\boldsymbol{c} \bmod (X^{kd/l} - \zeta^{jk})$ is essentially $1/q$, and thus the maximum probability of $\boldsymbol{c} \bmod (X^{kd/l} - \zeta^{jk})$ is essentially $q^{-kd/l}$, and this is negligible, then the prover must answer two $\boldsymbol{c}, \boldsymbol{c}'$ that differ modulo $X^{kd/l} - \zeta^{jk}$. Hence, $\bar{\boldsymbol{c}} = \boldsymbol{c} - \boldsymbol{c}'$ is non-zero modulo at least one of the divisors, say $(X^{d/l} - \zeta^j)$. Therefore, for every other divisor $\sigma^i(X^{d/l} - \zeta^j)$ we have

$$\sigma^i(\boldsymbol{c}) \bmod \sigma^i \left(X^{\frac{d}{l}} - \zeta^j\right) = \sigma^i \left(\boldsymbol{c} \bmod \left(X^{\frac{d}{l}} - \zeta^j\right)\right) \neq 0.$$

So we are in the situation where we have an accepting transcript pair with non-zero $\bar{\boldsymbol{c}}$ modulo every prime divisor of $(X^{kd/l} - \zeta^{jk})$. By repeating the argument for every $j \in \mathbb{Z}_{2l/k}^\times$, we see that we can get an extraction with non-vanishing $\bar{\boldsymbol{c}}$ modulo every prime divisor of $(X^d + 1)$.

The final protocol is given in Figure 2.

Theorem 4.4. *The protocol in Figure 2 is complete, statistical honest verifier zero-knowledge and computational special sound under the Module-SIS assumption. More precisely, let p be the maximum probability over $\mathbb{Z}_q$ of the coefficients of $\boldsymbol{c} \bmod X^{kd/l} - \zeta^k$ as in Lemma 3.3.*

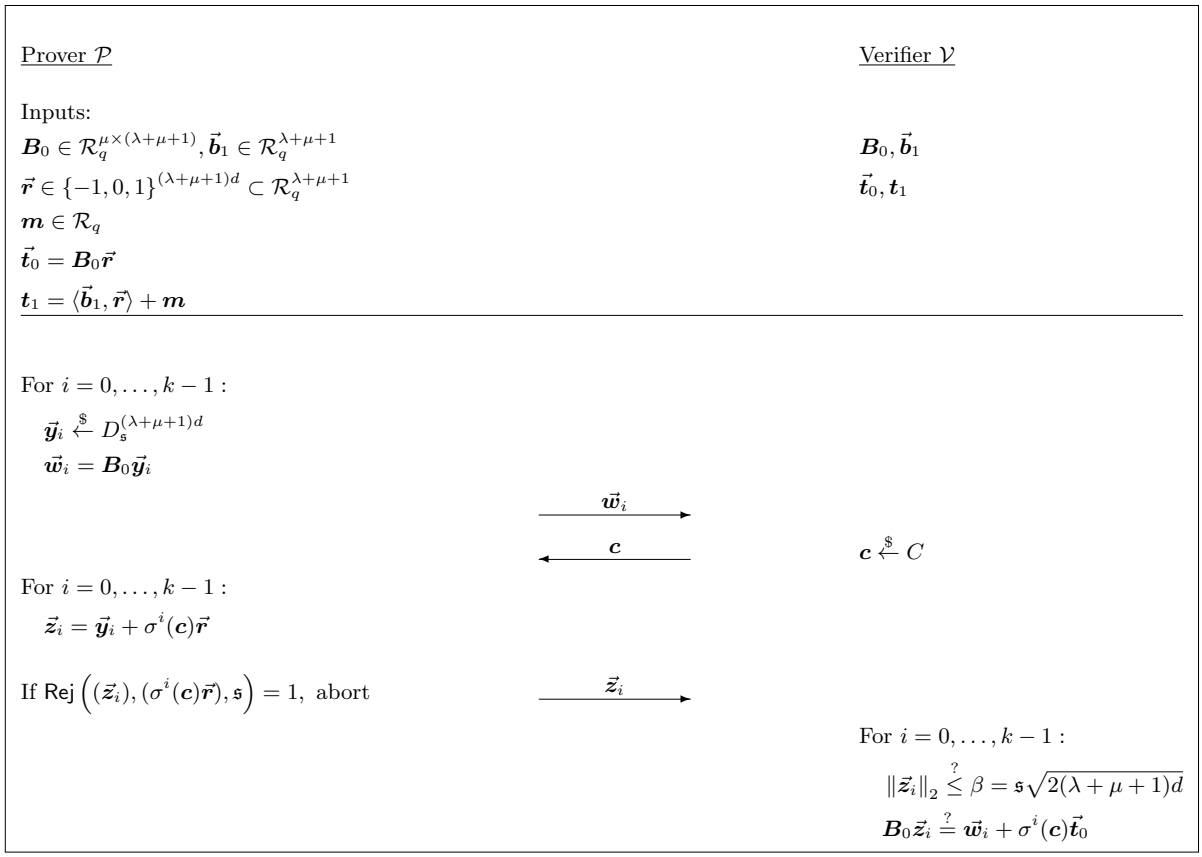

Fig. 2. Automorphism opening proof for the commitment scheme. We assume l, k are powers of two such that $k < l \leq d$, $q - 1 \equiv 2l \pmod{4l}$, and $\sigma = \sigma_{2l/k+1} \in \mathsf{Aut}(\mathcal{R}_q)$. Furthermore, C is the challenge distribution over $\mathcal{R}$ where each coefficient is independently identically distributed with $\Pr(0) = 1/2$ and $\Pr(-1) = \Pr(1) = 1/4$, κ is a bound on the ℓ_1-norm of $\boldsymbol{c}$, i.e. $\|\boldsymbol{c}\|_1 \leq \kappa$ with overwhelming probability for $\boldsymbol{c} \xleftarrow{\$} C$, and $D_{\mathfrak{s}}$ is the discrete Gaussian distribution on $\mathbb{Z}$ with standard deviation $\mathfrak{s} = 11k\kappa\|\vec{\boldsymbol{r}}\|_2$.

Then, for completeness, unless the honest prover $\mathcal{P}$ aborts due to the rejection sampling, it convinces the honest verifier $\mathcal{V}$ with overwhelming probability.

For zero-knowledge, there exists a simulator $\mathcal{S}$, that, without access to secret information, outputs a simulation of a non-aborting transcript of the protocol between $\mathcal{P}$ and $\mathcal{V}$ which has statistical distance at most 2^{-100} to the actual interaction.

For knowledge-soundness, there is an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a deterministic prover $\mathcal{P}^*$ that convinces $\mathcal{V}$ with probability $\varepsilon > p^{kd/l}$, $\mathcal{E}$ either outputs a weak opening for the commitment $\vec{\boldsymbol{t}}$ or a $\mathsf{MSIS}_{\mu,8\kappa\beta}$ solution for $\boldsymbol{B}_0$ in expected time at most $1/\varepsilon + (l/k)(\varepsilon - p^{kd/l})^{-1}$ when running $\mathcal{P}^*$ once is assumed to take unit time.

Moreover, the weak opening can be extended to also include k vectors $\vec{\boldsymbol{y}}_i^* \in \mathcal{R}_q^{\lambda+\mu+1}$ such that $\boldsymbol{B}_0\vec{\boldsymbol{y}}_i^* = \vec{\boldsymbol{w}}_i$, where $\vec{\boldsymbol{w}}_i$ are the prover commitments sent by $\mathcal{P}^*$ in the first flow. Furthermore, for every accepting transcript of an interaction with $\mathcal{P}^*$, the prover replies are given by $\vec{\boldsymbol{z}}_i = \vec{\boldsymbol{y}}_i^* + \sigma^i(\boldsymbol{c})\vec{\boldsymbol{r}}^*$.

Proof. Completeness. The vectors $\vec{\boldsymbol{z}}_i$ sent by $\mathcal{P}$ are independent and their distribution has statistical distance at most 2^{-100} from $D_{\mathfrak{s}}^{(\lambda+\mu+1)d}$ by Lemma 2.4. Lemma 2.5 implies that the bounds $\|\vec{\boldsymbol{z}}_i\|_2 \leq \beta = \mathfrak{s}\sqrt{2(\lambda+\mu+1)d}$ are true with overwhelming probability. It is easy to see that all of the other verification equations are always true for the messages sent by $\mathcal{P}$.

Zero-Knowledge. We can simulate a non-aborting transcript between the honest prover and the honest verifier in the following way. First, in a non-aborting honest transcript the $\vec{\boldsymbol{z}}_i$ are statistically close to $D_{\mathfrak{s}}^{(\lambda+\mu+1)d}$ by Lemma 2.4. So the simulator can just sample $\vec{\boldsymbol{z}}_i \xleftarrow{\$} D_{\mathfrak{s}}^{(\lambda+\mu+1)d}$. Next, again by Lemma 2.4, we know that $\sigma^i(\boldsymbol{c})\vec{\boldsymbol{r}}$ is independent of $\vec{\boldsymbol{z}}_i$ for all i, and hence $\boldsymbol{c}$ is independent of the $\vec{\boldsymbol{z}}_i$. So, the simulator picks $\boldsymbol{c} \xleftarrow{\$} C$ like the honest verifier. Now, the remaining messages $\vec{\boldsymbol{w}}_i$ are uniquely determined by the verification equations in an honest transcript because of completeness. We see that if the simulator computes these messages so that the verification equations become true, then the resulting transcript is statistically close to an honest transcript.

Soundness. The extractor $\mathcal{E}$ repeatedly runs $\mathcal{P}$ with freshly sampled challenges until it hits an accepting transcript. Let $\vec{\boldsymbol{w}}_i$, $\boldsymbol{c}$ and $\vec{\boldsymbol{z}}_i$ be the prover commitments, challenge and prover replies in this transcript, respectively. Then, $\mathcal{E}$ wants to get l/k more accepting transcripts such that for each of the l/k ideals $(X^{kd/l} - \zeta^{jk})$, $j \in \mathbb{Z}_{2l/k}^{\times}$, there is a transcript whose challenge differs from $\boldsymbol{c}$ modulo the ideal. Moreover, these transcripts need all contain the same prover commitments $\vec{\boldsymbol{w}}_i$ as in the first accepting transcript. To this end, for every j, $\mathcal{P}$ repeatedly rewinds the prover to just after the first flow and sends a random challenge that is different from $\boldsymbol{c}$ modulo $(X^{kd/l} - \zeta^{jk})$ until the resulting transcript with challenge $\boldsymbol{c}_j$ and replies $\vec{\boldsymbol{z}}_{ij}$ is accepting. We write $\bar{\boldsymbol{c}}_j = \boldsymbol{c} - \boldsymbol{c}_j$ for the challenge differences. By construction, $\bar{\boldsymbol{c}}_j \bmod (X^{kd/l} - \zeta^{jk}) \neq 0$.

The expected runtime for the whole process is as follows. The first transcript takes expected time $1/\varepsilon$. Next, when restricting to challenges that are different modulo one of the ideals $(X^{kd/l} + \zeta^{jk})$, the remaining success probability is at least $\varepsilon - p^{kd/l}$. So in expected time at most

$$\frac{1}{\varepsilon} + \frac{l}{k}\frac{1}{\varepsilon - p^{kd/l}}$$

the extractor has the $1 + l/k$ accepting transcripts.

Now fix an index $(e, f) \in I = \{0, \ldots, k-1\} \times \mathbb{Z}_{2l/k}^{\times}$ and consider the associated prime ideal $\mathfrak{p}_{ef} = \sigma^e(X^{d/l} - \zeta^f)$ dividing $(X^{kd/l} - \zeta^{fk})$. One of the permutations of $\bar{\boldsymbol{c}}_f$ is nonzero modulo $\mathfrak{p}_{ef}$. So there exists at least one $e' = e'(e, f) \in \{0, \ldots, k-1\}$ such that $\sigma^{e'}(\bar{\boldsymbol{c}}_f) \bmod \mathfrak{p}_{ef} \neq 0$. Now, we set

$$\vec{\boldsymbol{r}}^*_{ef} = \frac{\vec{\boldsymbol{z}}_{e'} - \vec{\boldsymbol{z}}_{e'f}}{\sigma^{e'}(\bar{\boldsymbol{c}}_f)} \bmod \sigma^e\left(X^{\frac{d}{l}} - \zeta^f\right).$$

Next, let $\vec{\boldsymbol{r}}^* \in \mathcal{R}_q^{\lambda+\mu+1}$ be such that $\vec{\boldsymbol{r}}^* \equiv \vec{\boldsymbol{r}}^*_{ef} \pmod{\sigma^e(X^{d/l} - \zeta^f)}$ for all $(e, f) \in I$. We claim $\sigma^i(\bar{\boldsymbol{c}}_j)\vec{\boldsymbol{r}}^* = \vec{\boldsymbol{z}}_i - \vec{\boldsymbol{z}}_{ij}$ for all $(i, j) \in I$, unless we find a Module-SIS solution for $\boldsymbol{B}_0$. From the verification equations we have

$$\boldsymbol{B}_0(\vec{\boldsymbol{z}}_i - \vec{\boldsymbol{z}}_{ij}) = \sigma^i(\bar{\boldsymbol{c}}_j)\vec{\boldsymbol{t}}_0 \tag{22}$$

for all $(i, j) \in I$. Therefore, either

$$\sigma^{e'}(\bar{\boldsymbol{c}}_f)(\vec{\boldsymbol{z}}_i - \vec{\boldsymbol{z}}_{ij}) = \sigma^i(\bar{\boldsymbol{c}}_j)(\vec{\boldsymbol{z}}_{e'} - \vec{\boldsymbol{z}}_{e'f}),$$

or we have found a non-trivial Module-SIS solution for $\boldsymbol{B}_0$ of length at most $8\kappa\beta$. We assume the former is true. Then,

$$\begin{aligned}
\sigma^i(\bar{\boldsymbol{c}}_j)\vec{\boldsymbol{r}}^* &\equiv \sigma^i(\bar{\boldsymbol{c}}_j)\vec{\boldsymbol{r}}^*_{ef} \\
&\equiv \sigma^i(\bar{\boldsymbol{c}}_j)\frac{\vec{\boldsymbol{z}}_{e'} - \vec{\boldsymbol{z}}_{e'f}}{\sigma^{e'}(\bar{\boldsymbol{c}}_f)} \\
&\equiv \vec{\boldsymbol{z}}_i - \vec{\boldsymbol{z}}_{ij} \pmod{\sigma^e(X^{\frac{d}{l}} - \zeta^f)},
\end{aligned}$$

and the claim follows from the Chinese remainder theorem. It holds $\boldsymbol{B}_0\sigma^i(\bar{\boldsymbol{c}}_j)\vec{\boldsymbol{r}}^* = \sigma^i(\bar{\boldsymbol{c}}_j)\vec{\boldsymbol{t}}_0$ for all $(i, j) \in I$ and this implies

$$\boldsymbol{B}_0\vec{\boldsymbol{r}}^* = \vec{\boldsymbol{t}}_0.$$

Finally, we compute the extracted message $\boldsymbol{m}^*$ which we set to fulfill the equation

$$\boldsymbol{t}_1 = \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{r}}^* \rangle + \boldsymbol{m}^*.$$

We conclude that the extractor has obtained a weak opening $(\sigma^i(\bar{\boldsymbol{c}}_j), \vec{\boldsymbol{r}}^*, \boldsymbol{m}^*)$ for the commitment $\vec{\boldsymbol{t}}$. In particular, it is true that $\left\|\sigma^i(\bar{\boldsymbol{c}}_j)\vec{\boldsymbol{r}}^*\right\|_2 \leq 2\beta$ for all $(i,j) \in I$.

We turn to the $\vec{\boldsymbol{y}}_i^*$. Set them to be the vectors defined by

$$\vec{\boldsymbol{z}}_i = \vec{\boldsymbol{y}}_i^* + \sigma^i(\boldsymbol{c})\vec{\boldsymbol{r}}^*.$$

Clearly, $\boldsymbol{B}_0\vec{\boldsymbol{y}}_i^* = \boldsymbol{B}_0(\vec{\boldsymbol{z}}_i - \sigma^i(\boldsymbol{c})\vec{\boldsymbol{r}}^*) = \vec{\boldsymbol{w}}_i$. Consider an arbitrary accepting transcript with the same prover commitments $\vec{\boldsymbol{w}}_i$ as above, but possibly a different challenge $\boldsymbol{c}'$ and different last messages $\vec{\boldsymbol{z}}_i'$. Then, for a moment write $\vec{\boldsymbol{z}}_i' = \vec{\boldsymbol{y}}_i^{*\prime} + \sigma^i(\boldsymbol{c}')\vec{\boldsymbol{r}}^*$. We aim to show $\vec{\boldsymbol{y}}_i^* = \vec{\boldsymbol{y}}_i^{*\prime}$. From the verification equations for $\vec{\boldsymbol{z}}_i$ and $\vec{\boldsymbol{z}}_i'$,

$$\boldsymbol{B}_0(\vec{\boldsymbol{z}}_i - \vec{\boldsymbol{z}}_i') = \sigma^i(\bar{\boldsymbol{c}})\vec{\boldsymbol{t}}_0$$

for all $i \in \{0, \ldots, k-1\}$ where $\bar{\boldsymbol{c}} = \boldsymbol{c} - \boldsymbol{c}'$. Combining this with Equation (22), unless we find a Module-SIS solution for $\boldsymbol{B}_0$,

$$\sigma^{e'}(\bar{\boldsymbol{c}}_f)(\vec{\boldsymbol{z}}_i - \vec{\boldsymbol{z}}_i') = \sigma^i(\bar{\boldsymbol{c}})(\vec{\boldsymbol{z}}_{e'} - \vec{\boldsymbol{z}}_{e'f}),$$

This implies, since $\vec{\boldsymbol{z}}_{e'} - \vec{\boldsymbol{z}}_{e'f} = \sigma^{e'}(\bar{\boldsymbol{c}}_f)\vec{\boldsymbol{r}}^*$,

$$\sigma^{e'}(\bar{\boldsymbol{c}}_f)(\vec{\boldsymbol{y}}_i^* - \vec{\boldsymbol{y}}_i^{*\prime}) = 0.$$

Recall $\sigma^{e'}(\bar{\boldsymbol{c}}_f) \not\equiv 0 \pmod{\mathfrak{p}_{ef}}$. Hence, $\vec{\boldsymbol{y}}_i^* \equiv \vec{\boldsymbol{y}}_i^{*\prime} \pmod{\mathfrak{p}_{ef}}$, and thus $\vec{\boldsymbol{y}}_i^* = \vec{\boldsymbol{y}}_i^{*\prime}$. □

.

5 Product Proof

In this section we present an efficient protocol for proving multiplicative relations between committed messages. Suppose the prover knows an opening to a commitment $\vec{\boldsymbol{t}}$ to three secret polynomials $\boldsymbol{m}_1, \boldsymbol{m}_2, \boldsymbol{m}_3 \in \mathcal{R}_q$,

$$\begin{aligned}
\vec{\boldsymbol{t}}_0 &= \boldsymbol{B}_0\vec{\boldsymbol{r}}, \\
\boldsymbol{t}_1 &= \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{r}} \rangle + \boldsymbol{m}_1, \\
\boldsymbol{t}_2 &= \langle \vec{\boldsymbol{b}}_2, \vec{\boldsymbol{r}} \rangle + \boldsymbol{m}_2, \\
\boldsymbol{t}_3 &= \langle \vec{\boldsymbol{b}}_3, \vec{\boldsymbol{r}} \rangle + \boldsymbol{m}_3.
\end{aligned}$$

His goal is to prove the multiplicative relation $\boldsymbol{m}_1\boldsymbol{m}_2 = \boldsymbol{m}_3$ in $\mathcal{R}_q$. We recall a simple technique for this, which for example was used in [BLS19, YAZ+19]. The prover commits to uniformly random masking polynomials $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3 \in \mathcal{R}_q$ and two so-called "garbage polynomials",

$$\begin{aligned}
\vec{\boldsymbol{t}}_0' &= \boldsymbol{B}_0'\vec{\boldsymbol{r}}', \\
\boldsymbol{t}_1' &= \langle \vec{\boldsymbol{b}}_1', \vec{\boldsymbol{r}}' \rangle + \boldsymbol{a}_1, \\
\boldsymbol{t}_2' &= \langle \vec{\boldsymbol{b}}_2', \vec{\boldsymbol{r}}' \rangle + \boldsymbol{a}_2, \\
\boldsymbol{t}_3' &= \langle \vec{\boldsymbol{b}}_3', \vec{\boldsymbol{r}}' \rangle + \boldsymbol{a}_3, \\
\boldsymbol{t}_4' &= \langle \vec{\boldsymbol{b}}_4', \vec{\boldsymbol{r}}' \rangle + \boldsymbol{a}_1\boldsymbol{m}_2 + \boldsymbol{a}_2\boldsymbol{m}_1 + \boldsymbol{a}_3, \\
\boldsymbol{t}_5' &= \langle \vec{\boldsymbol{b}}_5', \vec{\boldsymbol{r}}' \rangle + \boldsymbol{a}_1\boldsymbol{a}_2.
\end{aligned}$$

Then $\mathcal{P}$ replies to a challenge polynomial $\boldsymbol{x} \in \mathcal{R}_q$ with masked openings $\boldsymbol{f}_i = \boldsymbol{a}_i + \boldsymbol{x}\boldsymbol{m}_i$ of the messages $\boldsymbol{m}_i$. Now $\mathcal{P}$ shows that the $\boldsymbol{f}_i$ really open to the committed messages by proving that $\boldsymbol{t}_i' + \boldsymbol{x}\boldsymbol{t}_i - \boldsymbol{f}_i$ is a commitment

to zero for $i = 1, 2, 3$. Concretely, in addition to the standard opening proof for all of the commitments where the prover sends

$$\begin{aligned}\vec{\boldsymbol{w}} &= \boldsymbol{B}_0\vec{\boldsymbol{y}},\\ \vec{\boldsymbol{w}}' &= \boldsymbol{B}_0'\vec{\boldsymbol{y}}',\\ \vec{\boldsymbol{z}} &= \vec{\boldsymbol{y}} + \boldsymbol{c}\vec{\boldsymbol{r}},\\ \vec{\boldsymbol{z}}' &= \vec{\boldsymbol{y}}' + \boldsymbol{c}\vec{\boldsymbol{r}}',\end{aligned}$$

it will also send

$$\begin{aligned}\boldsymbol{v}_1 &= \langle\vec{\boldsymbol{b}}_1', \vec{\boldsymbol{y}}'\rangle + \boldsymbol{x}\langle\vec{\boldsymbol{b}}_1, \vec{\boldsymbol{y}}\rangle,\\ \boldsymbol{v}_2 &= \langle\vec{\boldsymbol{b}}_2', \vec{\boldsymbol{y}}'\rangle + \boldsymbol{x}\langle\vec{\boldsymbol{b}}_2, \vec{\boldsymbol{y}}\rangle,\\ \boldsymbol{v}_3 &= \langle\vec{\boldsymbol{b}}_3', \vec{\boldsymbol{y}}'\rangle + \boldsymbol{x}\langle\vec{\boldsymbol{b}}_3, \vec{\boldsymbol{y}}\rangle.\end{aligned}$$

The verifier then checks the equations

$$\begin{aligned}\boldsymbol{B}_0\vec{\boldsymbol{z}} &= \vec{\boldsymbol{w}} + \boldsymbol{c}\vec{\boldsymbol{t}}_0,\\ \boldsymbol{B}_0'\vec{\boldsymbol{z}}' &= \vec{\boldsymbol{w}}' + \boldsymbol{c}\vec{\boldsymbol{t}}_0',\\ \langle\vec{\boldsymbol{b}}_1', \vec{\boldsymbol{z}}'\rangle + \boldsymbol{x}\langle\vec{\boldsymbol{b}}_1, \vec{\boldsymbol{z}}\rangle &= \boldsymbol{v}_1 + \boldsymbol{c}(\boldsymbol{t}_1' + \boldsymbol{x}\boldsymbol{t}_1 - \boldsymbol{f}_1),\\ \langle\vec{\boldsymbol{b}}_2', \vec{\boldsymbol{z}}'\rangle + \boldsymbol{x}\langle\vec{\boldsymbol{b}}_2, \vec{\boldsymbol{z}}\rangle &= \boldsymbol{v}_2 + \boldsymbol{c}(\boldsymbol{t}_2' + \boldsymbol{x}\boldsymbol{t}_2 - \boldsymbol{f}_2),\\ \langle\vec{\boldsymbol{b}}_3', \vec{\boldsymbol{z}}'\rangle + \boldsymbol{x}\langle\vec{\boldsymbol{b}}_3, \vec{\boldsymbol{z}}\rangle &= \boldsymbol{v}_3 + \boldsymbol{c}(\boldsymbol{t}_3' + \boldsymbol{x}\boldsymbol{t}_3 - \boldsymbol{f}_3).\end{aligned}$$

This convinces the verifier that the $\boldsymbol{f}_i$ open to the secret messages $\boldsymbol{m}_i$. Next, consider the commitment

$$\boldsymbol{\tau} = \boldsymbol{t}_5' + \boldsymbol{x}\boldsymbol{t}_4' - (\boldsymbol{f}_1\boldsymbol{f}_2 - \boldsymbol{x}\boldsymbol{f}_3). \tag{23}$$

The verifier knows that the $\boldsymbol{f}_i$ are of the form $\boldsymbol{f}_i = \boldsymbol{a}_i^* + \boldsymbol{x}\boldsymbol{m}_i^*$ where the $\boldsymbol{a}_i^*$ and $\boldsymbol{m}_i^*$ are the (extracted) messages in the commitments $\boldsymbol{t}_i'$, $\boldsymbol{t}_i$. Therefore, $\mathcal{V}$ knows that $\boldsymbol{\tau}$ is a commitment to the message

$$\begin{aligned}\boldsymbol{\mu} &= \boldsymbol{m}_5^* + \boldsymbol{x}\boldsymbol{m}_4^* - (\boldsymbol{a}_1^*\boldsymbol{a}_2^* + \boldsymbol{x}(\boldsymbol{a}_1^*\boldsymbol{m}_2^* + \boldsymbol{a}_2^*\boldsymbol{m}_1^*) + \boldsymbol{x}^2\boldsymbol{m}_1^*\boldsymbol{m}_2^* - \boldsymbol{x}\boldsymbol{a}_3^* - \boldsymbol{x}^2\boldsymbol{m}_3^*)\\ &= (\boldsymbol{m}_5^* - \boldsymbol{a}_1^*\boldsymbol{a}_2^*) + \boldsymbol{x}(\boldsymbol{m}_4^* - \boldsymbol{a}_1^*\boldsymbol{m}_2^* - \boldsymbol{a}_2^*\boldsymbol{m}_1^* + \boldsymbol{a}_3^*) + \boldsymbol{x}^2(\boldsymbol{m}_3^* - \boldsymbol{m}_1^*\boldsymbol{m}_2^*)\end{aligned}$$

where $\boldsymbol{m}_4^*$, $\boldsymbol{m}_5^*$ are the extracted messages from the two garbage commitments. Now the prover completes the product proof by proving that $\boldsymbol{\tau}$ is a commitment to zero. We explain why this suffices. The message $\boldsymbol{\mu}$ can be viewed as a quadratic polynomial in $\boldsymbol{x}$ with coefficients that are independent from $\boldsymbol{x}$. If the prover is able to answer three challenges $\boldsymbol{x}$ such that their pairwise differences are invertible, then the polynomial must be the zero polynomial. In particular, the interesting term $\boldsymbol{m}_1^*\boldsymbol{m}_2^* - \boldsymbol{m}_3^*$, which is separated from the other terms as the leading coefficient in the challenge $\boldsymbol{x}$, must be zero.

There are two main problems with the technique:

1. The prover needs to send a large commitment $\vec{\boldsymbol{t}}'$ consisting of six uniform polynomials together with an opening proof for it, and also the three uniform masked openings $\boldsymbol{f}_i$.
2. Similarly as in the opening proof, the prover can cheat unless it is forced to be able to answer several challenges $\boldsymbol{x}$ with invertible differences. Unlike for the challenge $\boldsymbol{c}$ there is no shortness requirement associated to $\boldsymbol{x}$. Still, if q splits completely, the soundness error is restricted to $1/q$ even for uniformly random $\boldsymbol{x} \in \mathcal{R}_q$. Repetition is particularly expensive in the case of $\boldsymbol{x}$ since the masking polynomials $\boldsymbol{a}_i$ and corresponding commitments $\boldsymbol{t}_i'$ can not be reused. In fact, sending $\boldsymbol{f}_i = \boldsymbol{a}_i + \boldsymbol{x}\boldsymbol{m}_i$ for different $\boldsymbol{x}$ would break zero-knowledge. This even further increases the cost of the masking and garbage commitment and its opening proof.

Both problems result in concretely quite large communication sizes. We provide solutions to both problems and hereby drastically reduce the proof size.

First Problem. Instead of making the prover send the masked openings $\boldsymbol{f}_i$ and prove their well-formedness by committing to the $\boldsymbol{a}_i$, we let the verifier compute the $\boldsymbol{f}_i$ from the commitments $\boldsymbol{t}_i$. Then the proper relation to the messages $\boldsymbol{m}_i$ follows by construction. This is made possible by the results from Section 4. Recall that the verifier will be convinced that the vector $\vec{\boldsymbol{z}}$ in the opening proof is of the form $\vec{\boldsymbol{z}} = \vec{\boldsymbol{y}}^* + \boldsymbol{c}\vec{\boldsymbol{r}}^*$ where $\vec{\boldsymbol{y}}^*, \vec{\boldsymbol{r}}^*$ are independent from $\boldsymbol{c}$ and $\boldsymbol{t}_i = \langle \vec{\boldsymbol{b}}_i, \vec{\boldsymbol{r}}^* \rangle + \boldsymbol{m}_i^*$ with binded $\boldsymbol{m}_i^*$. Hence, the verifier will be convinced that

$$\boldsymbol{f}_i = \langle \vec{\boldsymbol{b}}_i, \vec{\boldsymbol{z}} \rangle - \boldsymbol{c}\boldsymbol{t}_i = \langle \vec{\boldsymbol{b}}_i, \vec{\boldsymbol{y}}^* \rangle - \boldsymbol{c}\boldsymbol{m}_i^*.$$

But this exactly is a masked opening of $\boldsymbol{m}_i^*$ with challenge $\boldsymbol{c}$ and masking polynomial $\boldsymbol{a}_i^* = \langle \vec{\boldsymbol{b}}_i, \vec{\boldsymbol{y}}^* \rangle$.

Now, when we compute the quadratic relation $\boldsymbol{f}_1\boldsymbol{f}_2 - \boldsymbol{c}\boldsymbol{f}_3$ we need to get rid of the garbage terms. It seems we need to linear combine the garbage commitments $\boldsymbol{t}_4'$ and $\boldsymbol{t}_5'$ with the challenge $\boldsymbol{c}$ and hereby construct a new commitment with commitment matrix $\boldsymbol{b}_4' + \boldsymbol{c}\boldsymbol{b}_5'$ depending on $\boldsymbol{c}$. If we went down this path we would need to send a second fresh opening proof with new challenge to show that $\boldsymbol{t}_4' + \boldsymbol{c}\boldsymbol{t}_5' - (\boldsymbol{f}_1\boldsymbol{f}_2 + \boldsymbol{c}\boldsymbol{f}_3)$ is a commitment to zero. This would be particularly bad if the garbage commitments are part of the commitment to the messages as one wants to have it in applications.

Instead, we use a new proof technique to achieve the same goal without two-layered opening proof and only one garbage commitment. In a nutshell, we use the masked opening $\boldsymbol{f}_4' = \langle \vec{\boldsymbol{b}}_4', \vec{\boldsymbol{z}}' \rangle - \boldsymbol{c}\boldsymbol{t}_4'$ of the garbage term to reduce $\boldsymbol{f}_1\boldsymbol{f}_2 + \boldsymbol{c}\boldsymbol{f}_3$ to the polynomial $\boldsymbol{f}_1\boldsymbol{f}_2 + \boldsymbol{c}\boldsymbol{f}_3 - \boldsymbol{f}_4'$ that is constant in $\boldsymbol{c}$. Then we show that the prover can just send this polynomial divided by $\boldsymbol{c}$ before seeing $\boldsymbol{c}$ without destroying zero-knowledge. The resulting verification equation, which is quadratic in the commitments, can be handled in the extraction proof by making repeated use of the interpolations of $\vec{\boldsymbol{z}}$, $\vec{\boldsymbol{z}}'$ and the associated expressions for the commitments.

As a further smaller improvement, notice the vector $\vec{\boldsymbol{w}} = \boldsymbol{B}_0\vec{\boldsymbol{y}}$ serves as a commitment to the randomness vector $\vec{\boldsymbol{y}}$. So this randomness vector can also be used for commitments to messages. Hence, instead of sampling a fresh randomness vector $\vec{\boldsymbol{r}}'$ and committing to it by sending $\vec{\boldsymbol{t}}_0'$, we directly commit to a garbage polynomial by sending

$$\boldsymbol{t}_1' = \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{y}} \rangle + \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{y}} \rangle \langle \vec{\boldsymbol{b}}_2, \vec{\boldsymbol{y}} \rangle = \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{y}} \rangle + \boldsymbol{a}_1\boldsymbol{a}_2.$$

This is a commitment to the constant polynomial in $\boldsymbol{f}_1\boldsymbol{f}_2 - \boldsymbol{c}\boldsymbol{f}_3$. So by subtracting $\boldsymbol{f}_1' = \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{z}} \rangle - \vec{\boldsymbol{t}}_1'$ we arrive at a simple multiple of $\boldsymbol{c}$.

For concreteness we state the resulting protocol in Figure 3. It has negligible soundness error when $\bar{\boldsymbol{c}}$ is invertible with overwhelming probability. Otherwise the protocol could be repeated to boost the soundness. Instead, we present a better solution.

Second Problem. As explained in Section 4, we set up parameters so that, for some $j \in \mathbb{Z}_{2l/k+1}^{\times}$, the prover can guess the challenge $\boldsymbol{c}$ modulo each of the k prime ideals $\sigma^i(X^{d/l} - \zeta^j)$, $i = 0, \ldots, k-1$, with non-negligible independent probability of about $1/q^{d/l}$. This means with the above method the prover will prove

$$\boldsymbol{m}_1\boldsymbol{m}_2 \equiv \boldsymbol{m}_3 \pmod{\sigma^i(X^{d/l} - \zeta^j)}$$

only with non-negligible soundness error. We solve this problem by linear combining all the permutations $\sigma^i(\boldsymbol{m}_1\boldsymbol{m}_2 - \boldsymbol{m}_3)$ with independent uniformly random challenge polynomials $\boldsymbol{\alpha}_i$. So we set out to prove

$$\sum_{i=0}^{k-1} \boldsymbol{\alpha}_i \sigma^i(\boldsymbol{m}_1\boldsymbol{m}_2 - \boldsymbol{m}_3) = 0.$$

Then our proof will show

$$\sum_{i=0}^{k-1} \boldsymbol{\alpha}_i \sigma^i(\boldsymbol{m}_1\boldsymbol{m}_2 - \boldsymbol{m}_3) \equiv 0 \pmod{\sigma^{i'}(X^{d/l} - \zeta^j)}$$

with independent cheating probability for $i' = 0, \ldots, k-1$. But the last equation for a single i' proves

$$\begin{aligned} &\sigma^i(\boldsymbol{m}_1\boldsymbol{m}_2 - \boldsymbol{m}_3) \equiv 0 \pmod{\sigma^{i'}(X^{d/l} - \zeta^j)} \\ \Rightarrow\ &\boldsymbol{m}_1\boldsymbol{m}_2 - \boldsymbol{m}_3 \equiv 0 \pmod{\sigma^{i'-i}(X^{d/l} - \zeta^j)} \end{aligned}$$

for all $i = 0, \ldots, k-1$ with cheating probability $1/q^{d/l}$ by the Schwartz-Zippel Lemma. A careful analysis will show the success probability of a cheating prover will be reduced to essentially at most

$$\varepsilon = \left(\frac{3}{q^{d/l}}\right)^k.$$

Now we derive the corresponding equation for the masked message openings. Here is where we need the randomness openings $\vec{\boldsymbol{z}}_i$ with the permutations $\sigma^i(\boldsymbol{c})$ of the challenge. The verifier can compute k masked openings for every message with challenges $\sigma^i(\boldsymbol{c})$ by setting

$$\boldsymbol{f}_j^{(i)} = \langle \vec{\boldsymbol{b}}_j, \vec{\boldsymbol{z}}_i \rangle - \sigma^i(\boldsymbol{c})\boldsymbol{t}_j.$$

In the extraction we will have the expressions

$$\boldsymbol{f}_j^{(i)} = \langle \vec{\boldsymbol{b}}_j, \vec{\boldsymbol{y}}_i^* \rangle - \sigma^i(\boldsymbol{c})\boldsymbol{m}_j^*.$$

Therefore, it follows that

$$\begin{aligned}
&\sum_{i=0}^{k-1} \boldsymbol{\alpha}_i \sigma^{-i} \left(\boldsymbol{f}_1^{(i)} \boldsymbol{f}_2^{(i)} - \sigma^i(\boldsymbol{c}) \boldsymbol{f}_3^{(i)} \right) \\
&= \sum_{i=0}^{k-1} \boldsymbol{\alpha}_i \sigma^{-i} \left(\langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{y}}_i^* \rangle \langle \vec{\boldsymbol{b}}_2, \vec{\boldsymbol{y}}_i^* \rangle \right) \\
&\quad + \boldsymbol{c} \sum_{i=0}^{k-1} \boldsymbol{\alpha}_i \sigma^{-i} \left(\boldsymbol{m}_1 \langle \vec{\boldsymbol{b}}_2, \vec{\boldsymbol{y}}_i^* \rangle + \boldsymbol{m}_2 \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{y}}_i^* \rangle - \langle \vec{\boldsymbol{b}}_3, \vec{\boldsymbol{y}}_i^* \rangle \right) \\
&\quad + \boldsymbol{c}^2 \left(\sum_{i=0}^{k-1} \boldsymbol{\alpha}_i \sigma^{-i} \left(\boldsymbol{m}_1^* \boldsymbol{m}_2^* - \boldsymbol{m}_3^* \right) \right)
\end{aligned}$$

We fold the constant coefficient into the coefficient of $\boldsymbol{c}$ by subtracting $\boldsymbol{f}_1' = \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{z}}_0 \rangle - \boldsymbol{t}_1'$ computed from the garbage commitment

$$\boldsymbol{t}_1' = \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{y}}_0 \rangle - \sum_{i=0}^{k-1} \boldsymbol{\alpha}_i \sigma^{-i} \left(\langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{y}}_i \rangle \langle \vec{\boldsymbol{b}}_2, \vec{\boldsymbol{y}}_i \rangle \right).$$

Then we arrive at

$$\begin{aligned}
&\sum_{i=0}^{k-1} \boldsymbol{\alpha}_i \sigma^{-i} \left(\boldsymbol{f}_1^{(i)} \boldsymbol{f}_2^{(i)} - \sigma^i(\boldsymbol{c}) \boldsymbol{f}_3^{(i)} \right) - \boldsymbol{f}_1' \\
&= \sum_{i=0}^{k-1} \boldsymbol{\alpha}_i \sigma^{-i} \left(\langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{y}}_i^* \rangle \langle \vec{\boldsymbol{b}}_2, \vec{\boldsymbol{y}}_i^* \rangle \right) + \boldsymbol{m}_1^{*\prime} \\
&\quad + \boldsymbol{c} \left(\sum_{i=0}^{k-1} \boldsymbol{\alpha}_i \sigma^{-i} \left(\boldsymbol{m}_1 \langle \vec{\boldsymbol{b}}_2, \vec{\boldsymbol{y}}_i^* \rangle + \boldsymbol{m}_2 \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{y}}_i^* \rangle - \langle \vec{\boldsymbol{b}}_3, \vec{\boldsymbol{y}}_i^* \rangle \right) - \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{r}}^* \rangle \right) \\
&\quad + \boldsymbol{c}^2 \left(\sum_{i=0}^{k-1} \boldsymbol{\alpha}_i \sigma^{-i} \left(\boldsymbol{m}_1^* \boldsymbol{m}_2^* - \boldsymbol{m}_3^* \right) \right).
\end{aligned}$$

The verifier checks that this is equal to $\boldsymbol{cv}$ using the polynomial $\boldsymbol{v}$ that it has received before sending the challenge.

It is important to note that we have departed from a straight-forward repetition of the protocol in Figure 3. The main advantage being that there is still only one garbage commitment necessary.

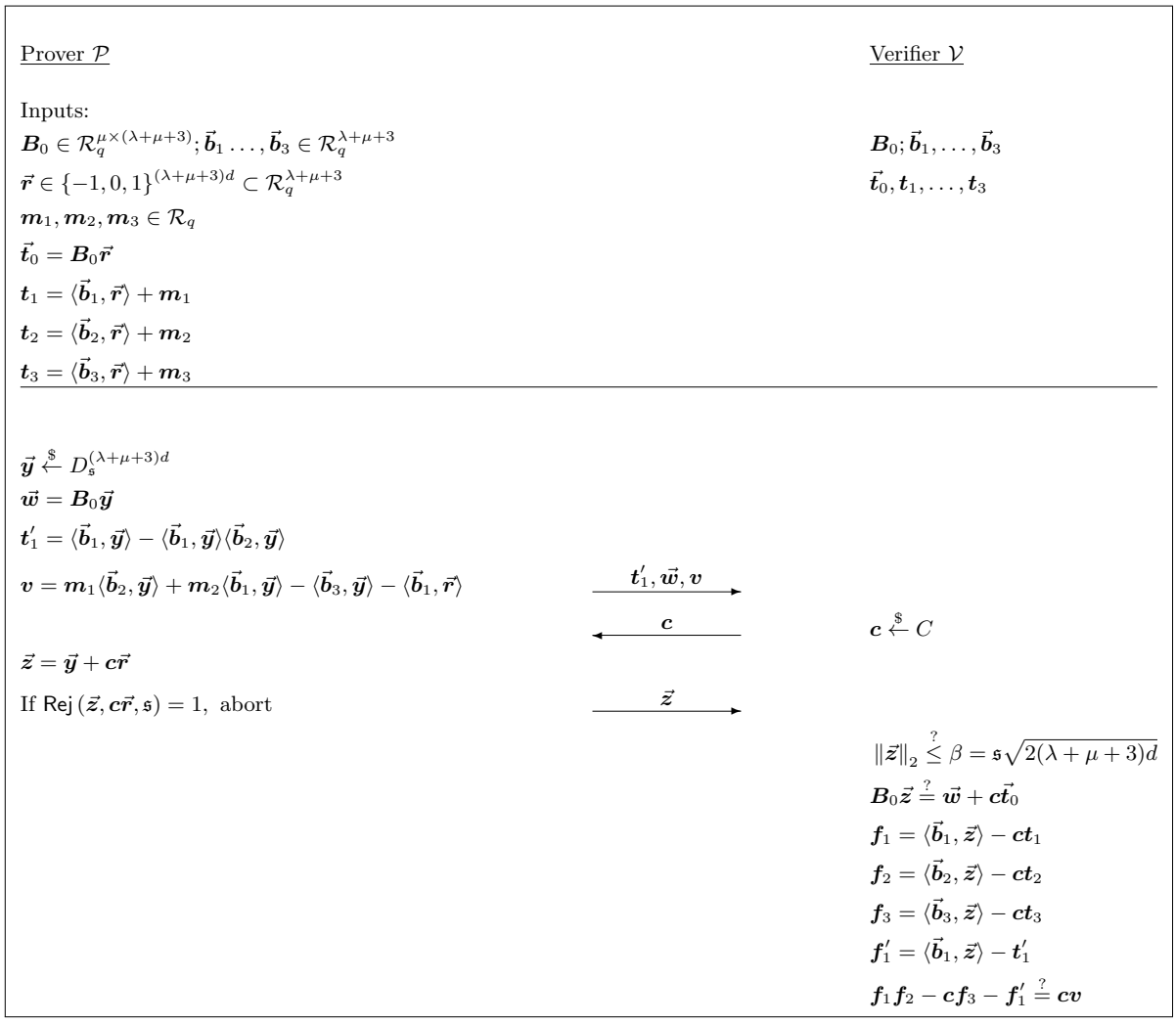

Fig. 3. Simple proof of multiplicative relation.

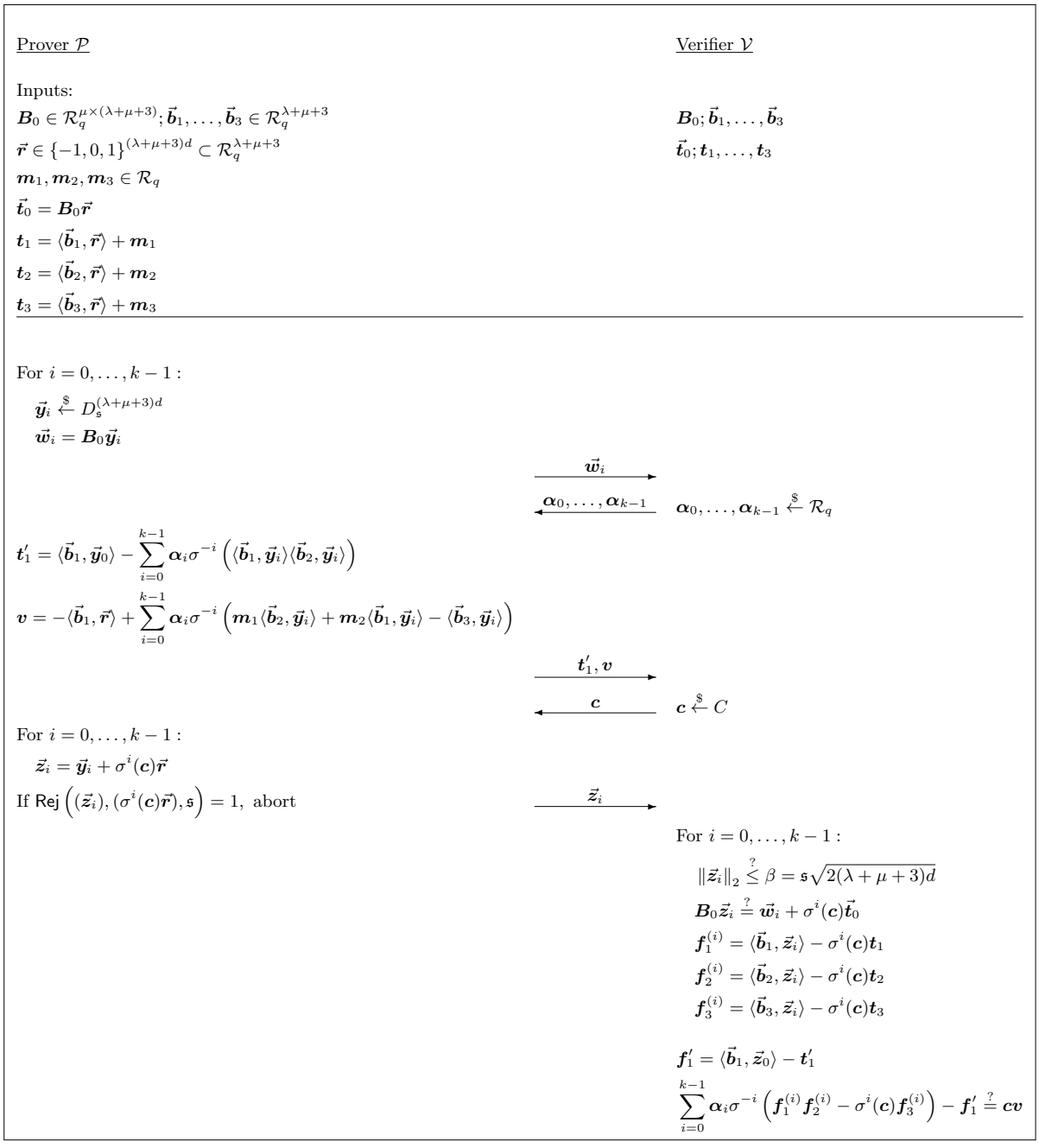

Fig. 4. Automorphism proof of multiplicative relation for automorphism $\sigma \in \mathrm{Aut}(\mathcal{R}_q)$ of order kd/l.

5.1 The Protocol

The final protocol is given in Figure 4. Its security is stated in Theorem 5.1.

Theorem 5.1. *The protocol in Figure 4 is complete, computational honest verifier zero-knowledge under the Module-LWE assumption and computational special sound under the Module-SIS assumption. More precisely, let p be the maximum probability over $\mathbb{Z}_q$ of the coefficients of $\boldsymbol{c} \bmod X^{kd/l} - \zeta^k$ as in Lemma 3.3.*

Then, for completeness, in case the honest prover $\mathcal{P}$ does not abort due to rejection sampling, it convinces the honest verifier $\mathcal{V}$ with overwhelming probability.

For zero-knowledge, there exists a simulator $\mathcal{S}$, that, without access to secret information, outputs a simulation of a non-aborting transcript of the protocol between $\mathcal{P}$ and $\mathcal{V}$. Then for every algorithm $\mathcal{A}$ that has advantage ε in distinguishing the simulated transcript from an actual transcript, there is an algorithm $\mathcal{A}'$ with the same running time that has advantage $\varepsilon - 2^{-100}$ in distinguishing $\mathsf{MLWE}_{\lambda,\chi}$.

For soundness, there is an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a deterministic prover $\mathcal{P}^*$ that convinces $\mathcal{V}$ with probability $\varepsilon \geq (3p^{d/l})^k$, $\mathcal{E}$ either outputs a weak opening for the commitment $\vec{\boldsymbol{t}}$ with messages $\boldsymbol{m}_1^*$, $\boldsymbol{m}_2^*$ and $\boldsymbol{m}_3^*$ such that $\boldsymbol{m}_1^*\boldsymbol{m}_2^* = \boldsymbol{m}_3^*$, or a $\mathsf{MSIS}_{\mu,8\kappa\beta}$ solution for $\boldsymbol{B}_0$ in expected time at most $1/\varepsilon + (l/k)(\varepsilon - p^{kd/l})^{-1}$ when running $\mathcal{P}^*$ once is assumed to take unit time.

Proof. Completeness. The vectors $\vec{\boldsymbol{z}}_i$ sent by $\mathcal{P}$ are independent and their distribution has statistical distance at most 2^{-100} from $D_{\mathfrak{s}}^{(\lambda+\mu+3)d}$ by Lemma 2.4. Lemma 2.5 implies that the bounds $\|\vec{\boldsymbol{z}}_i\|_2 \leq \beta = \mathfrak{s}\sqrt{2(\lambda+\mu+3)d}$ are true with overwhelming probability. It follows from careful inspection that all of the other verification equations are always true for the messages sent by $\mathcal{P}$.

Zero-Knowledge. We can simulate a non-aborting transcript between the honest prover and the honest verifier in the following way. First, in a non-aborting transcript the vectors $\vec{\boldsymbol{z}}_i$ are statistically close to $D_{\mathfrak{s}}^{(\lambda+\mu+3)d}$ by Lemma 2.4. So the simulator can just sample $\vec{\boldsymbol{z}}_i \xleftarrow{\$} D_{\mathfrak{s}}^{(\lambda+\mu+3)d}$. Next, again by Lemma 2.4, we know that $\sigma^i(\boldsymbol{c})\vec{\boldsymbol{r}}$ is independent from $\vec{\boldsymbol{z}}_i$, and hence $\boldsymbol{c}$ is independent from the $\vec{\boldsymbol{z}}_i$. So, the simulator picks $\boldsymbol{c} \xleftarrow{\$} C$ like the honest verifier. The commitment $\boldsymbol{t}_1'$ is computationally indistinguishable from a uniformly random polynomial if MLWE_λ is hard. In fact, the construction of the commitment scheme is such that $\boldsymbol{t}_1'$ contains an additive term that is precisely an MLWE_λ sample. So the simulator can just take a uniformly random $\boldsymbol{t}_1' \in \mathcal{R}_q$. Now, in an honest transcript, the remaining messages $\vec{\boldsymbol{w}}_i$ and $\boldsymbol{v}$ are all uniquely determined by the verification equations because of completeness. We see that if the simulator computes these messages so that the verification equations become true, then the resulting transcript is indistinguishable from the honest transcript. More precisely, a simulated transcript has statistical distance at most 2^{-100} from a distribution which differs from the actual transcripts only in that $\boldsymbol{t}_1'$ is distributed differently. Therefore, if there is an algorithm $\mathcal{A}$ that has advantage ε in distinguishing a simulated transcript from an actual transcript, then this algorithm must be able to distinguish MLWE_λ samples from random with advantage $\varepsilon - 2^{-100}$.

Soundness. Firstly, the extractor opens the commitments $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_3$ and $\boldsymbol{t}_1'$. We know from Theorem 4.4 that, unless $\mathcal{E}$ finds a $\mathsf{MSIS}_{\mu,8\kappa\beta}$ solution, it computes vectors $\vec{\boldsymbol{y}}^*$ and $\vec{\boldsymbol{r}}^*$ such that for every accepting transcript with fixed first message $\vec{\boldsymbol{w}}_i$,

$$\boldsymbol{z}_i = \vec{\boldsymbol{y}}_i^* + \sigma^i(\boldsymbol{c})\vec{\boldsymbol{r}}^*.$$

Then let $\boldsymbol{m}_1^*, \ldots, \boldsymbol{m}_3^* \in \mathcal{R}_q$ and $\boldsymbol{m}_1^{*\prime} \in \mathcal{R}_q$ be the corresponding extracted messages which are defined by

$$\begin{aligned}
\vec{\boldsymbol{t}}_1 &= \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{r}}^* \rangle + \boldsymbol{m}_1^*, \\
\vec{\boldsymbol{t}}_2 &= \langle \vec{\boldsymbol{b}}_2, \vec{\boldsymbol{r}}^* \rangle + \boldsymbol{m}_2^*, \\
\vec{\boldsymbol{t}}_3 &= \langle \vec{\boldsymbol{b}}_3, \vec{\boldsymbol{r}}^* \rangle + \boldsymbol{m}_3^*, \\
\vec{\boldsymbol{t}}_1' &= \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{y}}^* \rangle + \boldsymbol{m}_1^{*\prime}.
\end{aligned}$$

From the above decompositions of $\vec{\boldsymbol{z}}_i$ and the expression for $\boldsymbol{t}_1$ we obtain the expression

$$\boldsymbol{f}_1^{(i)} = \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{y}}_i^* \rangle - \sigma^i(\boldsymbol{c})\boldsymbol{m}_1^*$$

and similarly for $\boldsymbol{f}_2^{(i)}$, $\boldsymbol{f}_3^{(i)}$ and $\boldsymbol{f}_1'$. Substituting these into the last verification equation gives

$$\begin{aligned}
&\left(\sum_{i=0}^{k-1} \boldsymbol{\alpha}_i \sigma^{-i}\left(\langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{y}}_i^* \rangle \langle \vec{\boldsymbol{b}}_2, \vec{\boldsymbol{y}}_i^* \rangle\right) + \boldsymbol{m}_1^{*\prime}\right) \\
&+ \boldsymbol{c}\left(\sum_{i=0}^{k-1} \boldsymbol{\alpha}_i \sigma^{-i}\left(\boldsymbol{m}_1 \langle \vec{\boldsymbol{b}}_2, \vec{\boldsymbol{y}}_i^* \rangle + \boldsymbol{m}_2 \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{y}}_i^* \rangle - \langle \vec{\boldsymbol{b}}_3, \vec{\boldsymbol{y}}_i^* \rangle\right) - \langle \vec{\boldsymbol{b}}_1, \vec{\boldsymbol{r}}^* \rangle - \boldsymbol{v}\right) \\
&+ \boldsymbol{c}^2\left(\sum_{i=0}^{k-1} \boldsymbol{\alpha}_i \sigma^{-i}\left(\boldsymbol{m}_1^* \boldsymbol{m}_2^* - \boldsymbol{m}_3^*\right)\right) \qquad (24) \\
&= \mathbf{0}.
\end{aligned}$$

Now, it is crucial that the last equation is a polynomial in $\boldsymbol{c}$ with coefficients that are independent from $\boldsymbol{c}$. More precisely, this equation holds for every $\boldsymbol{c}$ in every accepting transcript with commitments $\vec{\boldsymbol{w}}_i, \boldsymbol{t}_1', \boldsymbol{v}$. With this preparation we now bound the success probability of the prover assuming that $\boldsymbol{m}_1^* \boldsymbol{m}_2^*$ is not equal to $\boldsymbol{m}_3^*$. In this case $\boldsymbol{m}_1^* \boldsymbol{m}_2^* - \boldsymbol{m}_3^*$ is non-zero modulo at least one of the prime ideals,

$$\boldsymbol{m}_1^* \boldsymbol{m}_2^* - \boldsymbol{m}_3^* \not\equiv 0 \pmod{\sigma^i(X^{\frac{d}{l}} - \zeta^j)}$$

for some $(i, j) \in I$. But then

$$\boldsymbol{p} = \sum_{i=0}^{k-1} \boldsymbol{\alpha_i} \sigma^{-i}\left(\boldsymbol{m}_1^* \boldsymbol{m}_2^* - \boldsymbol{m}_3^*\right) \bmod \left(X^{\frac{kd}{l}} - \zeta^{kj}\right)$$

is a uniformly random polynomial for uniformly random $\boldsymbol{\alpha}_i$. So with probability $(1 - 1/q^{d/l})^k$ it is non-zero in all k prime ideals dividing $(X^{kd/l} - \zeta^k)$. In this case, modulo each prime ideal, there can be at most two points that make the evaluation of the quadratic polynomial in Equation (24) zero. So, they combine to 2^k elements modulo $X^{kd/l} - \zeta^j$. Hence, even when we assume $\boldsymbol{c} \bmod (X^{kd/l} - \zeta^{kj}) = \boldsymbol{x}$ has probability $p^{kd/l}$ for all elements $\boldsymbol{x}$ modulo $X^{kd/l} - \zeta^j$, the success probability of the prover is clearly bounded by $2^k p^{kd/l}$. Next, if $\boldsymbol{p}$ is zero in one of the k prime ideals, which happens with probability $k/q^{d/l}(1 - 1/q^{d/l})^{k-1}$, we find that there are at most $2^{k-1} q^{d/l}$ possible values for $\boldsymbol{c} \bmod (X^{kd/l} - \zeta^{kj})$ and the success probability is bounded by $2^{k-1} q^{d/l} p^{kd/l}$. Continuing in this way we see it must be that

$$\begin{aligned}
\varepsilon &\leq \sum_{i=0}^{k} \binom{k}{i} \left(\frac{1}{q^{d/l}}\right)^i \left(1 - \frac{1}{q^{d/l}}\right)^{k-i} 2^{k-i} q^{id/l} p^{kd/l} \\
&< \left(3p^{\frac{d}{l}}\right)^k.
\end{aligned}$$

This is a contradiction to the bound in the theorem and hence $\boldsymbol{m}_1^* \boldsymbol{m}_2^* = \boldsymbol{m}_3^*$.

5.2 Amortized Protocol

The protocol from the last section can be extended into a protocol for the case where the prover wants to prove multiplicative relations between many messages. In this extension there will still only be one garbage commitment necessary for proving all of the relations. So the cost for the garbage commitment is amortized over all relations. Suppose we want to prove n product relations

$$\boldsymbol{m}_1^{(j)} \boldsymbol{m}_2^{(j)} = \boldsymbol{m}_3^{(j)}$$

for $j = 1, \ldots, n$. Then virtually in the same way in which we linear combine the automorphic images of a single relation with uniform challenges, we can use even more challenges and linear combine all the automorphic

images of all the relations. Concretely, we want to prove

$$\sum_{i=0}^{k-1}\sum_{j=1}^{n}\boldsymbol{\alpha}_{in+j}\sigma^i\left(\boldsymbol{m}_1^{(j)}\boldsymbol{m}_2^{(j)}-\boldsymbol{m}_3^{(j)}\right)=0$$

with $\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_{nk}\xleftarrow{\$}\mathcal{R}_q$. Now a nice feature of the Schwartz-Zippel lemma is that this does not decrease the soundness. Intuitively, as soon as one of the relations is false, then the linear combination of all of the relations will be uniformly random, and this will be detected with overwhelming probability.

5.3 Non-Interactive Protocol and Proof Sizes

In this section we compute the size of a non-interactive proof. The protocol in Figure 4 is made non-interactive with the help of the standard Fiat-Shamir technique. This means that the challenges are computed by the prover by hashing all previous messages and public information, and the hash function is modeled as a random oracle. To shorten the length of the proof, a standard technique is to not send the input to the hash function, but rather send its output (i.e. the challenge) and let the verifier recompute the input from the later transmitted terms using the verification equation and then test that the hash of these computed input terms is indeed the challenge. Concretely, in the non-interactive version of the product proof, the $k\mu+1$ full-size polynomials $\vec{\boldsymbol{w}}_i$ and $\boldsymbol{v}$ do not have to be transmitted and only $\boldsymbol{t}_1'$ remains as a non-short polynomial. The polynomials in the vectors $\vec{\boldsymbol{z}}_i$ are short discrete Gaussian vectors with standard deviation $\mathfrak{s}$. Every coefficient is smaller than $6\mathfrak{s}$ in absolute value with probability $1-2^{-24}$ [Lyu12, Lemma 4.4]. So we can assume this is the case for all coefficients – the non-interactive prover can just restart otherwise. Eventually, we obtain that one non-interactive proof needs

$$d\lceil\log(q)\rceil+k(\lambda+\mu+3)d\lceil\log(12\mathfrak{s})\rceil+256$$

bits.

Example I. Suppose we are given a commitment to 8 polynomials in the ring $\mathcal{R}_q$ of rank $d=128$ with a prime $q\approx 2^{32}$ that splits completely. So there are 1024 secret coefficients. For this ring the maximum probability over $\mathbb{Z}_q$ of the coefficients of $\boldsymbol{c}$ mod $(X^4-\zeta^4)$ for $\boldsymbol{c}\xleftarrow{\$}C$ when a coefficient is zero with probability $1/2$ is $p=2^{-31.44}$ according to the formula in Lemma 3.3. So $k=4$ permutations of a challenge under the automorphism $\sigma=\sigma_{64}$ are sufficient to reach negligible soundness error. Further, suppose the commitment scheme uses MLWE rank $\lambda=10$ and MSIS rank $\mu=9$. We find $\|\boldsymbol{c}\vec{\boldsymbol{r}}\|_1\leq 77$ with probability bigger than $1-2^{-100}$. Then, if we set the standard deviation of the discrete Gaussian to $\mathfrak{s}=11\cdot 77\cdot\sqrt{(\lambda+\mu+8)d}=49793.23$ we find that we need $\mathsf{MSIS}_{\mu,8d\beta}$ to be secure for $\beta=\mathfrak{s}\sqrt{2(\lambda+\mu+8)d}$. We found the root Hermite factor to be approximately 1.0043. Similarly, MLWE_λ with ternary noise has hermite Factor 1.0043. Finally, the size of our product proof for these parameters is 30.32 KB.

Example II. For a fair comparison to [BDL+18, Parameter set I of Table 2], where the polynomial X^d+1 does not necessarily split into linear factors, we modify the previous example and switch to using a prime q that splits into prime ideals of degree 4 (and so there are 32 NTT slots). Then we have negligible soundness error already with $k=1$ and we don't need parallel repetitions and automorphisms. The protocol is given in Figure 3 and the proof size goes down to 8 KB.

Example III. In the above comparison to [BDL+18], we created a commitment to 1024 values (or 256 NTT coefficients each being a polynomial of degree 3). For the 32-bit range proof example stated in the introduction, we only need 128 values (i.e. we need 32 NTT coefficients each being a polynomial of degree 3). The size of such a product proof is approximately 5.6KB.

References

Ban93. Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296:625–635, 1993.

BBB+18. Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Gregory Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *IEEE Symposium on Security and Privacy*, pages 315–334, 2018.

BBHR18. Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Fast reed-solomon interactive oracle proofs of proximity. In *ICALP*, volume 107 of *LIPIcs*, pages 14:1–14:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

BCR+19. Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In *EUROCRYPT (1)*, volume 11476 of *Lecture Notes in Computer Science*, pages 103–128. Springer, 2019.

BDL+18. Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. More efficient commitments from structured lattice assumptions. In *SCN*, pages 368–385, 2018.

BKLP15. Fabrice Benhamouda, Stephan Krenn, Vadim Lyubashevsky, and Krzysztof Pietrzak. Efficient zero-knowledge proofs for commitments from learning with errors over rings. In *ESORICS*, 2015.

BLS19. Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In *CRYPTO (1)*, pages 176–202. Springer, 2019.

BV19. Emmanuel Breuillard and Péter P. Varjú. Cut-off phenomenon for the ax+b Markov chain over a finite field. 2019.

CDG87. F. R. K. Chung, Persi Diaconis, and R. L. Graham. Random walks arising in random number generation. *Ann. Probab.*, 15(3):1148–1165, 07 1987.

CLS16. Hao Chen, Kristin E. Lauter, and Katherine E. Stange. Security considerations for galois non-dual RLWE families. In *SAC*, pages 443–462, 2016.

Dia88. Persi Diaconis. Group representations in probability and statistics. *Lecture Notes-Monograph Series*, 11:i–192, 1988.

dPLS18. Rafaël del Pino, Vadim Lyubashevsky, and Gregor Seiler. Lattice-based group signatures and zero-knowledge proofs of automorphism stability. In *ACM CCS*, pages 574–591. ACM, 2018.

ENS20. Muhammed Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings., 2020. `https://eprint.iacr.org/2020/?`

ESLL19. Muhammed F. Esgin, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. In *CRYPTO (1)*, volume 11692 of *Lecture Notes in Computer Science*, pages 115–146. Springer, 2019.

Hil90. Martin Victor Hildebrand. *Rates of convergence of some random processes on finite groups.* PhD thesis, Harvard University, 1990.

Hil06. Martin Hildebrand. On the Chung-Diaconis-Graham random process. *Electronic Communications in Probability*, 11:347–356, 2006.

Kil92. Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *STOC*, pages 723–732. ACM, 1992.

KKW18. Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In *ACM Conference on Computer and Communications Security*, pages 525–537. ACM, 2018.

LLNW18. Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Lattice-based zero-knowledge arguments for integer relations. In *CRYPTO (2)*, volume 10992 of *Lecture Notes in Computer Science*, pages 700–732. Springer, 2018.

LM06. Vadim Lyubashevsky and Daniele Micciancio. Generalized compact knapsacks are collision resistant. In *ICALP (2)*, pages 144–155, 2006.

LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23, 2010.

LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptography*, 75(3):565–599, 2015.

LS18. Vadim Lyubashevsky and Gregor Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In *EUROCRYPT (1)*, pages 204–224. Springer, 2018.

Lyu09. Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, pages 598–616, 2009.

Lyu12. Vadim Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, pages 738–755, 2012.

Mic00. Silvio Micali. Computationally sound proofs. *SIAM J. Comput.*, 30(4):1253–1298, 2000.

PR06. Chris Peikert and Alon Rosen. Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In *TCC*, pages 145–166, 2006.

YAZ+19. Rupeng Yang, Man Ho Au, Zhenfei Zhang, Qiuliang Xu, Zuoxia Yu, and William Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In *CRYPTO (1)*, volume 11692 of *Lecture Notes in Computer Science*, pages 147–175. Springer, 2019.