Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General[⋆]

Vadim Lyubashevsky[1], Ngoc Khanh Nguyen[1,2], and Maxime Plançon[1,2]

[1] IBM Research Europe, Zurich
[2] ETH Zurich, Zurich

Abstract. We present a much-improved practical protocol, based on the hardness of Module-SIS and Module-LWE problems, for proving knowledge of a short vector $\vec{s}$ satisfying $A\vec{s} = \vec{t} \bmod q$. The currently most-efficient technique for constructing such a proof works by showing that the ℓ_∞ norm of $\vec{s}$ is small. It creates a commitment to a polynomial vector $\mathbf{m}$ whose CRT coefficients are the coefficients of $\vec{s}$ and then shows that (1) $A \cdot \mathsf{CRT}(\mathbf{m}) = \vec{t} \bmod q$ and (2) in the case that we want to prove that the ℓ_∞ norm is at most 1, the polynomial product $(\mathbf{m} - \mathbf{1}) \cdot \mathbf{m} \cdot (\mathbf{m} + \mathbf{1})$ equals to 0. While these schemes are already quite practical, the requirement of using the CRT embedding and only being naturally adapted to proving the ℓ_∞-norm, somewhat hinders the efficiency of this approach.

In this work, we show that there is a more direct and more efficient way to prove that the coefficients of $\vec{s}$ have a small ℓ_2 norm, which does not require an equivocation with the ℓ_∞ norm nor any conversion to the CRT representation. We observe that the inner product between two vectors $\vec{r}$ and $\vec{s}$ can be made to appear as a coefficient of a product (or sum of products) between polynomials which are functions of $\vec{r}$ and $\vec{s}$. Thus, by using a polynomial product proof system and hiding all but one coefficient, we are able to prove knowledge of the inner product of two vectors (or of a vector with itself) modulo q. Using a cheap "approximate range proof", one can then lift the proof to be over $\mathbb{Z}$ instead of $\mathbb{Z}_q$. Our protocols for proving short norms work over all (interesting) polynomial rings, but are particularly efficient for rings like $\mathbb{Z}[X]/(X^n + 1)$ in which the function relating the inner product of vectors and polynomial products happens to be a "nice" automorphism.

The new proof system can be plugged into constructions of various lattice-based privacy primitives in a black-box manner. As examples, we instantiate a verifiable encryption scheme and a group signature scheme which are more than twice as compact as the previously best solutions.

1 Introduction

The fundamental hardness assumption upon which lattice-based cryptography rests is that it is computationally difficult to find a low-norm vector $\mathbf{s}$ satisfying

$$\mathbf{A}\mathbf{s} = \mathbf{t} \bmod q. \tag{1}$$

It is then natural that for creating privacy-preserving protocols based on the hardness of lattice problems, one is usually required to prove the knowledge of an $\mathbf{s}$ satisfying the above, or a related, equality. Unlike in the analogous case of discrete logarithms, where proving knowledge of a secret s satisfying $g^s = t$ turns out to have a very simple and efficient solution [Sch89], the added requirement of showing that $\|\mathbf{s}\|$ is small turns out to be a major complication for *practical* lattice cryptography.

Over polynomial rings (i.e. rings of the form $\mathbb{Z}[X]/(f(X))$, where $f(X)$ is a monic, irreducible polynomial), one can give a fairly-efficient zero-knowledge proof of knowledge of a vector $\bar{\mathbf{s}}$ and a polynomial c with small coefficients satisfying

$$\mathbf{A}\bar{\mathbf{s}} = c\mathbf{t} \bmod q, \tag{2}$$

where $\|\bar{\mathbf{s}}\|$ is some factor (depending on the dimension of $\mathbf{s}$) larger than $\|\mathbf{s}\|$ [Lyu09, Lyu12]. While such proofs are good enough for constructing fairly efficient basic protocols (e.g. signature schemes [Lyu09, Lyu12,

[⋆] This is the full version of the paper presented at CRYPTO 2022.

BG14, DKL$^+$18]), the fact that the norm of the extracted $\bar{\mathbf{s}}$ is noticeably larger than that of $\mathbf{s}$, along with the presence of the extra multiplicand c, makes these proofs awkward to use in many other situations. This very often results in the protocols employing these proofs being less efficient than necessary, or in not giving the resulting scheme the desired functionality.

As simple examples of inefficiencies that may creep up when only being able to prove (2), consider Regev-style lattice-based encryption schemes (e.g. [Reg09, LPS10, LPR10]) where $\mathbf{s}$ is the randomness (including the message) and $\mathbf{t}$ is the ciphertext. In order to decrypt, it is necessary for $\mathbf{t}$ to have a short pre-image, and so being able to only prove knowledge of (2) is not enough to guarantee that the ciphertext $\mathbf{t}$ can be decrypted because it is $c\mathbf{t}$ that has a short pre-image, not $\mathbf{t}$ (and c is not known to the decryptor). A consequence of this is that the currently most-efficient lattice-based verifiable encryption scheme [LN17] has the undesirable property that the expected decryption time is equal to the adversary's running time because the decryptor needs to essentially guess c. Employing this scheme in the real world would thus require setting up a scenario where the adversary cannot use too much time to construct the proof. Other lattice-based constructions (e.g. group signature schemes [LNPS21]) were required to select much larger parameters than needed in order to accommodate the presence of the multiplicand c and the "slack" between the length of the known solution $\mathbf{s}$ and the solution $\bar{\mathbf{s}}$ that one can prove.

1.1 Prior Art for Proofs of (1)

Early protocols for exactly proving (1) used the combinatorial algorithm of Stern [Ste93] to prove that the ℓ_∞ norm of $\mathbf{s}$ is bounded by revealing a random permutation of $\mathbf{s}$. The main problem with these protocols was that their soundness error was 2/3, and so they had to be repeated around 200 times to achieve an acceptably small (i.e. 2^{-128}) soundness error. This resulted in proofs for even basic statements[3] being more than 1MB in size [LNSW13], while more interesting constructions required outputs on the order of dozens of Megabytes (e.g. [LLNW16]). A noticeable improvement was achieved in [Beu20] by generically combining Stern's protocol with a "cut-and-choose" technique to noticeably decrease the soundness error of each protocol run (at the expense of higher running times). This allowed proofs for basic statements to be around 200KB in size.

A very different, more algebraic, approach for proving (1) utilized lattice-based commitments and zero-knowledge proofs about committed values to prove relations between the coefficients of $\mathbf{s}$ and also prove a bound on its ℓ_∞ norm. The first such protocols [YAZ$^+$19, BLS19, ESLL19] had proof sizes that were on the order of several hundred kilobytes. These schemes were greatly improved in [ALS20, ENS20], where it was shown how to very efficiently prove products of polynomial products over a ring and then linear relations over the CRT coefficients of committed values. Optimizations of these techniques [LNS21b] decreased the proof size for the basic example to around 33KB.

The high level idea for these proofs, when $\mathbf{s}$ has coefficients in the set $\{-1, 0, 1\}$, is to create a BDLOP commitment [BDL$^+$18] to a polynomial $\mathbf{m}$ whose CRT coefficients are the coefficients of $\mathbf{s}$, prove this (linear) relationship as well as the one in (1) [ENS20], and then prove that $(\mathbf{m} - \mathbf{1}) \cdot \mathbf{m} \cdot (\mathbf{m} + 1) = 0$ [ALS20].

There are a few intrinsic elements of this approach which hinder its efficiency, especially in certain situations. The first is that $\mathbf{m}$ consists of large polynomial coefficients, and so committing to it requires using a more expensive commitment scheme, which is especially costly when $\mathbf{s}$ is long[4] (we discuss this in more detail when talking about various commitments in Section 1.3). Another downside is that for vectors $\mathbf{s}$ with somewhat-large coefficients, such as ones that are obtained from trapdoor sampling (e.g. [ABB10, MP12]), proving the smallness of the ℓ_∞-norm becomes significantly costlier because the degree of the polynomial product increases. There is also an incompatibility between the requirement that the underlying ring has a lot of CRT slots and negligible soundness error of the protocol – thus a part of the protocol needs to be

[3] A standard example that has been used for comparison-purposes in several works is 1024×2048 integer matrix $\mathbf{A}$, a 32-bit modulus q, and $\mathbf{s}$ having coefficients in $\{-1, 0, 1\}$ (or $\|\mathbf{s}\| \leqslant \sqrt{2048}$).

[4] The aforementioned framework was most appropriate for committing to small-dimensional messages (e.g. in protocols related to anonymous transactions (e.g. [EZS$^+$19, LNS21b, ESZ21]) and proving various relationships between them.

repeated for soundness amplification. And finally, proving the ℓ_2 norm, rather than the ℓ_∞ one, is very often what one would like to do when constructing proofs for lattice-based primitives. This is because efficient trapdoor-sampling used in many lattice primitives produces vectors of (tightly) bounded ℓ_2 norm, and noise also generation generally results in tight ℓ_2-norm bounds.

1.2 Our Results

We propose a simpler, more efficient, and more direct approach for proving a tight bound on the ℓ_2 norm of $\mathbf{s}$ satisfying (1). Unlike in the previous approach, we do not need to recommit to $\mathbf{s}$ in CRT form, and therefore don't have a ring algebra requirement which had a negative effect on the protocol soundness. Furthermore, not needing to create a BDLOP commitment to $\mathbf{s}$ noticeably shrinks the proof size. In particular, we define a commitment scheme which combines the Ajtai [Ajt96] and BDLOP [BDL$^+$18] commitments into one, and then put the long commitment to $\mathbf{s}$ into the "Ajtai" part of the commitment scheme, which does not increase the commitment size.[5]

We then observe that the inner product of two vectors over $\mathbb{Z}$ can be made to appear as the constant coefficient of a polynomial product, or as a coefficient in a sum of polynomial products. Our protocol for proving the ℓ_2-norm of $\mathbf{s}$ is then a specific application of a more general protocol that can prove knowledge of constant coefficients of quadratic relations over polynomial rings for messages that are committed in the "Ajtai" and "BDLOP" parts of our new commitment. Our protocols are built up in a black-box manner from basic building blocks, and can then also be used in a black box manner for implementing the zero-knowledge proof parts of various lattice-based primitives. As examples, the ZK proof of the basic relation from (1) is $\approx 2.5X$ shorter than in previous works, a verifiable encryption scheme can be as short as the one from [LN17] without the constraint that the decryption time is proportional to the adversary's attack time, and we give a group signature scheme whose signatures are more than $2X$ smaller than the currently most compact one.

Our proof system for the basic equality from (1) is around 14KB, and approximately 8KB of that consists of just the "minimum" commitment (i.e. a commitment to just one element in $\mathcal{R}_q$ that doesn't include $\mathbf{s}$) and its opening proof. This shows that our construction is quite close to being optimal for any approach that requires creating a commitment to $\mathbf{s}$ using known lattice-based commitment schemes. Since all zero-knowledge proofs that we're aware of for showing that a secret s satisfies $f(s)$ work by first committing to s, it appears that any significant improvement to this proof system (e.g. another factor of 2) would require noticeable improvements in fundamental lattice primitives, basing security on stronger assumptions, or a noticeable departure from the current approach.

We now give a detailed overview of the techniques and results in this work, and then sketch how our framework can be used to construct lattice-based privacy protocols.

1.3 Techniques Overview

Throughout most of the introduction and paper, we will concentrate on the ring $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d+1)$, as our constructions are most efficient here because they can utilize a specific automorphism in this ring. Towards the end of this section and in Section 7, we describe how to adapt our construction, and most applications, to other rings that do not have this algebraic structure. All our constructions will be based on the hardness of the Module-SIS and Module-LWE problems and one should think of the degree of the underlying ring d to be something small like 64 or 128 (we use 128 for all our instantiations).

Commitment Schemes. In the original Ajtai commitment scheme, implicit in [Ajt96], one commits to a message $\mathbf{s}_1$ using randomness $\mathbf{s}_2$, where $\|\mathbf{s}_i\|$ are small, as

$$\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t} \bmod q. \tag{3}$$

[5] The BDLOP part of the commitment scheme is then used for low-dimensional auxiliary elements that will need to be committed to later in the protocol.

It's easy to see that creating a second valid opening $(\mathbf{s}_1', \mathbf{s}_2')$ for the same commitment value $\mathbf{t}$ is equivalent to solving the SIS problem over $\mathcal{R}_q$, and the hiding aspect of the commitment scheme is based on the indistinguishability of $(\mathbf{A}_2, \mathbf{A}_2\mathbf{s}_2)$ from uniform. A useful feature of the above commitment scheme is that the dimension of the message $\mathbf{s}_1$ does not increase the commitment size. And since the hardness of SIS does not really depend on the dimension of the solution, increasing the dimension of $\mathbf{s}_1$ does not negatively impact the security either. On the other hand, one does need the coefficients of $\mathbf{s}_1$ to be small.

A different commitment scheme, called the BDLOP scheme [BDL+18], commits to a message $\mathbf{m}$ using randomness $\mathbf{s}$ as

$$\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix} \cdot \mathbf{s} + \begin{bmatrix}\mathbf{0}\\ \mathbf{m}\end{bmatrix} = \begin{bmatrix}\mathbf{t}_A\\ \mathbf{t}_B\end{bmatrix} \bmod q, \tag{4}$$

where only the randomness $\mathbf{s}$ needs to have a small norm. An opening of this commitment is just $\mathbf{s}$ since it uniquely determines $\mathbf{m}$, and so it is again easy to see that two different openings lead to a solution to SIS for the matrix $\mathbf{A}$. The hiding property of this commitment is based on the indistinguishability from uniform of $\left(\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix}, \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\end{bmatrix} \cdot \mathbf{s}\right)$.

This scheme has two advantages and one disadvantage over the one in (3). The disadvantage is that both the commitment size and the opening size grow linearly with the dimension of the message vector $\mathbf{m}$. An advantage is that the coefficients of $\mathbf{m}$ can be arbitrarily large modulo q. The other advantage is that if one plans ahead and sets the dimension of $\mathbf{s}$ large enough, one can very cheaply append commitments of new elements in $\mathcal{R}_q$. For example, if we have already created a commitment to $\mathbf{m}$ as in (4) and would like to commit to another polynomial vector $\mathbf{m}'$, we can compute $\mathbf{B}'\mathbf{s} + \mathbf{m}' = \mathbf{t}_B' \bmod q$, where $\mathbf{B}'$ is some public randomness. If $\left(\begin{bmatrix}\mathbf{A}\\ \mathbf{B}\\ \mathbf{B}'\end{bmatrix}, \begin{bmatrix}\mathbf{A}\\ \mathbf{B}\\ \mathbf{B}'\end{bmatrix} \cdot \mathbf{s}\right)$ is indistinguishable from uniform, then $(\mathbf{t}_A, \mathbf{t}_B, \mathbf{t}_B')$ is a commitment to $\mathbf{m}, \mathbf{m}'$. Note that committing to k extra $\mathcal{R}_q$ elements requires growing the commitment size by only k $\mathcal{R}_q$ elements, something that cannot be done using the scheme from (3).

For optimality, our construction will require features from both of these schemes, and it actually turns out to be possible to combine the two of them into one. So to commit to a message $\mathbf{s}_1$ with a small norm, and a message $\mathbf{m}$ with unrestricted coefficients (modulo q), one can create a commitment

$$\begin{bmatrix}\mathbf{A}_1\\ \mathbf{0}\end{bmatrix} \cdot \mathbf{s}_1 + \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix} \cdot \mathbf{s}_2 + \begin{bmatrix}\mathbf{0}\\ \mathbf{m}\end{bmatrix} = \begin{bmatrix}\mathbf{t}_A\\ \mathbf{t}_B\end{bmatrix} \bmod q, \tag{5}$$

where the randomness is $\mathbf{s}_2$. We will call this combination of the Ajtai and BDLOP commitment scheme, the ABDLOP commitment. The savings over creating two separate commitments is that instead of needing the $\mathbf{t}$ term from (3) and the $\mathbf{t}_A$ term from (4), we only have the $\mathbf{t}_A$ term. So we get an Ajtai commitment to $\mathbf{s}_1$ for free! And similarly, the opening does not require both $\mathbf{s}_2$ from (3) and $\mathbf{s}$ from (4).

One can show that (5) is indeed a commitment scheme and has an efficient zero-knowledge opening proof.[6] Furthermore, there is also an efficient zero-knowledge proof (much like in [BDL+18]) which allows one to efficiently show that the committed values $\mathbf{s}_1, \mathbf{m}$ satisfy a relation over $\mathcal{R}_q$

$$\mathbf{R}_1\mathbf{s}_1 + \mathbf{R}_m\mathbf{m} = \mathbf{u} \bmod q, \tag{6}$$

where the matrices $\mathbf{R}_1, \mathbf{R}_m$, and the vector $\mathbf{u}$ are public. This proof system is given in Figure 4, and we just mention that the proof size is not affected by the sizes of $\mathbf{R}_1$ and $\mathbf{R}_m$. In other words, the proof size for proving linear relations over $\mathcal{R}_q$ is the same as the proof size of just proving knowledge of the committed values. The only way in which this proof puts a restriction on the underlying ring is that the modulus q must be large enough so that the extracted SIS solution is hard, and that the challenge set $\mathcal{C}$ is such that the difference of challenges is (with high probability) invertible. This can be done by choosing the modulus

[6] As for the Ajtai and BDLOP commitments, the opening needs to be carefully defined because the ZK proof only proves approximate relations as in (2). The details are in Section 3.1.

q in a way that $X^d + 1$ splits into very few irreducible factors of the form $X^k - r_i$ modulo q (or the prime factors of q), which in turn implies that all elements of $\mathcal{R}_q$ with small coefficients are invertible [LS18].

The way this commitment scheme will be used in our protocols is that we will put high-dimensional messages with small coefficients into $\mathbf{s}_1$, while putting small-dimensional values with large coefficients – generally auxiliary "garbage terms" that we will need to commit to during the protocol which aid in proving relations among the elements in $\mathbf{s}_1$ – into $\mathbf{m}$.

Inner Products over $\mathbb{Z}_q$. Suppose that instead of just wanting to prove linear relations over $\mathcal{R}_q$, as above, we wanted to prove linear relations over $\mathbb{Z}_q$. That is, if we let R_1, R_m be integer matrices, and we write $\vec{s}_1$ and $\vec{m}$ to be integer vectors whose coefficients are the integer coefficients of the polynomial vectors $\mathbf{s}_1$ and $\mathbf{m}$, then we would like to prove that $R_1\vec{s}_1 + R_m\vec{m} = \vec{u} \bmod q$.

An important observation is the following: if $\vec{r} = (r_0, r_1, \ldots, r_{d-1}), \vec{s} = (s_0, s_1, \ldots, s_{d-1}) \in \mathbb{Z}_q^d$ are vectors and $r(X) = \sum_i r_i X^i, s(X) = \sum_i s_i X^i \in \mathcal{R}_q$ are the corresponding polynomials, then $\langle \vec{r}, \vec{s} \rangle \bmod q$ is equal to the constant coefficient of the polynomial product $r(X^{-1}) \cdot s(X)$ over $\mathcal{R}_q$.[7] Similarly, for $\vec{r}, \vec{s} \in \mathbb{Z}_q^{kd}$, one can define the corresponding polynomial vectors $\mathbf{r} = (r_1, \ldots, r_k), \mathbf{s} = (s_1, \ldots, s_k) \in \mathcal{R}_q^k$ to have the same coefficients as $\vec{r}, \vec{s}$ in the straightforward manner, then $\langle \vec{r}, \vec{s} \rangle \bmod q$ is equal to the constant coefficient of $\sum_i r_i(X^{-1}) \cdot s_i(X)$, where the multiplication is performed over $\mathcal{R}_q$.

For a polynomial $h = h_0 + h_1 X + \ldots + h_{d-1}X^{d-1} \in \mathcal{R}_q$, we will write $\widetilde{h}$ to mean the constant coefficient h_0. The procedure to prove that $\langle \vec{r}, \vec{s} \rangle \bmod q = \alpha$ is then to create polynomial vectors $\mathbf{r}, \mathbf{s}$ such that $\widetilde{\langle \mathbf{r}, \mathbf{s} \rangle}$ (where the inner product is over $\mathcal{R}_q$) is equal to $\langle \vec{r}, \vec{s} \rangle$. One can hope to use the protocol from Figure 4 to prove the linear relation over $\mathcal{R}_q$, which would imply the linear relation over $\mathbb{Z}_q$. The problem is that naively proving the relation over $\mathcal{R}_q$ would necessarily require the prover to reveal all the coefficients of $\langle \mathbf{r}, \mathbf{s} \rangle$ instead of just the constant one, which implies giving out extra information about the committed vector $\vec{s}$, and so is clearly not zero-knowledge.

We now outline the solution to this problem for general linear functions. For a linear function $f : \mathcal{R}_q^k \to \mathcal{R}_q$, we would like to prove that the committed values $\mathbf{s}_1, \mathbf{m}$ in the ABDLOP commitment satisfy $\widetilde{f}(\mathbf{s}_1, \mathbf{m}) = 0$ (for aesthetics, we will write $\widetilde{f}(x)$ to mean $\widetilde{f(x)}$). In order to mask all but the constant coefficient, we use a masking technique from [ENS20], where the prover first creates a commitment to a polynomial $g \in \mathcal{R}_q$ such that $\widetilde{g} = 0$ and all of its other coefficients are chosen uniformly at random. In our proof system, he commits to this polynomial in the "BDLOP part" of (5) by outputting $t_g = \langle \mathbf{b}, \mathbf{s}_2 \rangle + g$, where $\mathbf{b}$ is some random public polynomial vector. The verifier then sends a random challenge $\gamma \in \mathbb{Z}_q$, and the prover computes

$$h = \gamma \cdot f(\mathbf{s}_1, \mathbf{m}) + g. \tag{7}$$

The prover then creates a proof, as in Figure 4, that the committed values $\mathbf{s}_1, \mathbf{m}$, and g satisfy this linear relation, and sends h along with this proof to the verifier. The verifier simply checks the validity of the linear proof, and also that $\widetilde{h} = 0 \bmod q$.

The proof leaks no information about all but the constant coefficient of $f(\mathbf{s}_1, \mathbf{m})$ because they are masked by the completely random coefficients of g. To see that this proof is sound, note that for all g, if $\widetilde{f}(\mathbf{s}_1, \mathbf{m}) \neq 0$, then $\Pr_\gamma[\gamma \cdot \widetilde{f}(\mathbf{s}_1, \mathbf{m}) + \widetilde{g} = 0] \leqslant 1/q_1$, where q_1 is the smallest prime factor of q. In order to reduce the soundness error down to ϵ, the prover would need to create a commitment to λ different g_i, where $(1/q_1)^\lambda = \epsilon$ and then reply to λ different challenges γ_i by creating λ different h_i as in (7). Since the g_i are just one polynomial in $\mathcal{R}_q$, the h_i are also just one polynomial each, and so amplifying the proof requires sending just 2λ extra elements in $\mathcal{R}_q$.

The above shows that proving one relation $\widetilde{f}(\mathbf{s}_1, \mathbf{m}) = 0$ requires a small number λ of extra polynomials g and h. Usually, we will want to prove many such linear equations, and so it would be quite inefficient if our proof size grew linearly in their number. But, just like in the basic protocol in Figure 4, we can show that the number of equations that we need to prove does not affect the size of the proof. If we would like to prove

[7] For a polynomial $r(X) = \sum_{i=0}^{d-1} r_i X^i \in \mathcal{R}_q$, $r(X^{-1}) = r_0 - \sum_{i=1}^{d-1} r_i X^{d-i}$.

k equations $\widetilde{f}_i(\mathbf{s}_1, \mathbf{m}) = 0$, the prover still sends the term g in the first round (let's ignore the amplification for now), but this time instead of sending just one random challenge $\gamma \in \mathbb{Z}_q$, the verifier sends k random challenges γ_i. The prover then creates the equation

$$h = \sum_i \gamma_i \cdot f_i(\mathbf{s}_1, \mathbf{m}) + g, \tag{8}$$

and sends h along with a proof that the $\mathbf{s}_1, \mathbf{m}$, and g satisfy the above. The verifier checks the proof and that $\widetilde{h} = 0 \bmod q$. The fact that this proof leaks no information and that the soundness error is again $1/q_1$ is virtually identical as for (7), and we give a full description of this protocol in Figure 5.

Quadratic Relations and Norms. In the above, we saw an overview of how one can prove knowledge of inner products over $\mathcal{R}_q$ and $\mathbb{Z}_q$ when one of the values is committed to and the other is public. We now show how to do the same thing when both values are in the commitment – in other words, how to prove quadratic relations over committed values.

The most efficient protocol for proving quadratic relations between committed polynomials in $\mathcal{R}_q$ is given in [ALS20]. That protocol assumes that the elements were committed using the BDLOP commitment scheme, and one can show that a similar approach works for the ABDLOP scheme as well. And so one can prove arbitrary quadratic relations over $\mathcal{R}_q$ between the committed polynomials in the polynomial vector $\mathbf{s}_1$ and $\mathbf{m}$ in (5). We will now explain how to use this proof system, together with the ideas presented above, to construct a proof that the $\mathbf{s}$ satisfying (1) has small ℓ_2-norm. For simplicity of this description, let's just suppose that we would like to prove that $\|\mathbf{s}\| = \beta$ instead of $\|\mathbf{s}\| \leqslant \beta$.[8] The idea is to first commit to $\mathbf{s}$ as part of the $\mathbf{s}_1$ part of (5) (i.e. in the "Ajtai part" of the ABDLOP scheme). Then we use the observation from the previous section that notes that if $\mathbf{s}_1 = (s_1, \ldots, s_k) \in \mathcal{R}_q^k$, then $\|\mathbf{s}\|^2$ is the constant coefficient of $\sum_i s_i(X^{-1}) \cdot s_i(X)$. We cannot directly use the proof system for linear proofs because that one assumed that one of the multiplicands was public. We thus need to extend the protocol from [ALS20] to prove knowledge of $\sum_i s_i(X^{-1}) \cdot s_i(X)$ when having a commitment to $\mathbf{s}$.

Let us recall the main ideas from [ALS20] and then see how they can be applied to the ABDLOP commitment. Suppose, for example, that we wanted to prove that $s_1 s_2 - s_3 = 0$, and we had commitments to s_i in the Ajtai part of the ABDLOP commitment (i.e. the s_i are part of the $\mathbf{s}_1$ in (5)). If one looks at the protocol in Figure 4 for proving knowledge of committed values in the ABDLOP protocol, then we note that the prover sends the vector $\mathbf{z}_1 = c\mathbf{s}_1 + \mathbf{y}_1$. This $\mathbf{z}_1$ consists of terms $z_i = s_i c + y_i$, where c is a polynomial challenge (with small coefficients) and y_i is a masking polynomial whose job is to hide s_i.

The high level idea in which the protocol from [ALS20] (and some that preceded it [BLS19, ESLL19, YAZ$^+$19]) proves quadratic relations is by having the verifier create a quadratic equation (in c) out of the linear equations $z_i = cs_i + y_i$. That is, the verifier computes

$$z_1 z_2 - c z_3 = (s_1 s_2 - s_3)c^2 + g_1 c + g_0, \tag{9}$$

where g_1 and g_0 are some terms which depend on y_i and s_i and are committed to by the prover prior to receiving the challenge c.[9] The above is a quadratic equation in the variable c (since all the other terms are already committed to), and so if the prover shows that $z_1 z_2 - c z_3 = g_1 c + g_0$ (i.e. it's actually a linear equation) it will imply that with high probability the quadratic coefficient, $s_1 s_2 - s_3$ is equal to 0.

To prove that the constant coefficient of $s(X^{-1}) \cdot s(X)$ is some value β, one can try to do something similar. Here, it becomes important that the function mapping s to $s(X^{-1})$ is an automorphism (call it σ) for $\mathcal{R}_q$. Given the term $z = sc + y$, the verifier is able to compute

$$\sigma(z) \cdot z - \sigma(c) \cdot c \cdot \beta^2 = (\sigma(s) \cdot s - \beta^2) \cdot \sigma(c) \cdot c + \sigma(s) \cdot y \cdot \sigma(c) + s \cdot \sigma(y) \cdot c + \sigma(y) \cdot y, \tag{10}$$

[8] To prove the latter, one would commit to a vector $\vec{b}$ which is the binary representation of the integer $\beta^2 - \|\mathbf{s}\|^2$ and then prove that it is indeed binary and that $\langle \vec{b}, (1, 2, 2^2, \ldots 0, \ldots, 0) \rangle$ is $\beta^2 - \|\mathbf{s}\|^2$; which implies that the latter is positive. Note that it is still a quadratic relation in the committed values $\mathbf{s}$ and $\vec{b}$.

[9] [ALS20] showed that the y_i were already implicitly committed to by the first part of the protocol.

and, if the above is equal to $g_2 \cdot \sigma(c) + g_1 \cdot c + g_0$, would like to conclude that the coefficients in front of $\sigma(c) \cdot c$ is 0. Unfortunately, we can't conclude this because the c and $\sigma(c)$ are not independent. What we instead do is choose the challenges c from a set that is fixed under this automorphism – that is, $\sigma(c) = c$. Then (10) becomes

$$\sigma(z) \cdot z - c^2\beta^2 = (\sigma(s) \cdot s - \beta^2) \cdot c^2 + (\sigma(s) \cdot y + s \cdot \sigma(y)) \cdot c + \sigma(y) \cdot y, \tag{11}$$

and we again have a quadratic equation in c. Luckily, the requirement that $\sigma(c) = c$ does not restrict the challenge set too much. In particular, if we choose $c \in \mathcal{R}_q$ to be of the form $c = c_0 + \sum_{i=1}^{d/2-1} c_i \cdot (X^i - X^{d-i})$, where $c_i \in \mathbb{Z}_q$, then $c = \sigma(c)$.[10] So we are free to set $d/2$ coefficients of the challenge polynomial instead of the usual d. So obtaining the same soundness requires the coefficients to be a little larger, but this has a rather small effect on the proof size.

The protocol in Figure 6 is a very general protocol for proving that a quadratic function in the coefficients of $\mathbf{s}_1$ and $\mathbf{m}$, and the automorphisms of $\mathbf{s}_1$ and $\mathbf{m}$, is satisfied as long as the challenge set is fixed under the particular automorphism. If we only want to prove the ℓ_2 norm, then we do not want to prove a quadratic function over $\mathcal{R}_q$, but rather we just want to prove something about the *constant coefficient* of a quadratic relation over $\mathcal{R}_q$. To do this, we employ the same masking technique as in (7) that we used for our linear proofs over $\mathbb{Z}_q$. Furthermore, just like in the linear proofs setting, if we need to prove multiple quadratic relations, we can first combine them into one equation, and then the proof size does not increase. Also note that we can clearly combine linear and quadratic equations together into one quadratic equation. The full protocol is presented in Figure 8.

We are almost done, except for the fact that all of our proofs are modulo q. That is, the protocol only proves that $\|\mathbf{s}\|^2 = \beta^2 \bmod q$, which is not the same as proving $\|\mathbf{s}\|^2 = \beta^2$. In order to prove that there is no "wraparound" modulo q, we employ a version of the "approximate range proof" technique to show that the coefficients of $\mathbf{s}$ are all small-enough. We do not need a sharp bound on these coefficients, but just need to show that they are small-enough that no wraparound occurs. For this, we use the technique [BL17, BN20, LNS20, GHL21] of committing to a masking vector $\vec{y}$ (in the BDLOP part of (5)), receiving a $-1/0/1$ challenge matrix R, and outputting $\vec{z} = R\vec{s} + \vec{y}$ (and doing a rejection sampling to hide $\vec{s}$). It can be shown that if $\|\vec{z}\|$ is small, then $\|\vec{s}\|$ is also small. The dimension of $\vec{y}$ and $\vec{z}$ is small (between 128 and 256), and so the extra commitment to $\vec{y}$ and the revealing of $\vec{z}$ is inexpensive. The protocol for the approximate range proof is given in Figure 9, and the general protocol proving these approximate range proofs in combination with other quadratic functions is given in Figure 10.

Putting Everything Together. The structure for proving (1) involves creating an ABDLOP commitment as in (5) with $\mathbf{s}_1 = \mathbf{s}$ and making the randomness $\mathbf{s}_2$ long enough to accommodate future commitments to a few intermediate terms necessary in the proof. One then uses the aforementioned proofs to show that $\|\mathbf{s}_1\|$ is small, and that the linear equation in (1) is satisfied. Notice that we don't really need any ring structure on the equation in (1); if it is over $\mathbb{Z}_q$, we can simply prove it using the linear proofs over $\mathbb{Z}_q$. This is computationally more expensive than if the equation were over $\mathcal{R}_q$, because for every multiplication over $\mathbb{Z}_q$, we have to compute one multiplication over $\mathcal{R}_q$, but the proof size will be the same.

We also note that the modulus in (1) does not have to be the same as in the commitment scheme. In fact, it will often be necessary to use a larger modulus in the commitment scheme because it has to be larger than $\|\mathbf{s}\|^2$. For example, we can set the commitment scheme modulus to $p \cdot q$ and then simply lift the equation in (1) to this modulus by multiplying both sides of it by p. As long as the challenge differences are invertible in the ring $\mathcal{R}_q$ and $\mathcal{R}_p$, all the protocols go through unchanged.

Another possibility is, instead of proving $\mathbf{As} = \mathbf{t} \mod q$, one proves that

$$\mathbf{As} - \mathbf{t} = \mathbf{r} \cdot q \tag{12}$$

[10] This is easy to see because $\sigma(X^i - X^{d-i}) = X^{-i} - X^{i-d}$, and multiplying by $-X^d = 1$, we obtain $\sigma(X^i - X^{d-i}) = -X^{d-i} + X^i$.

over the integers. If each row of $\mathbf{A}$ consists of m integer coefficients, then each coefficient of $\mathbf{r}$ has magnitude at most mq. One can then do the proof system using a larger modulus p, and also prove that each coefficient of $q^{-1}(\mathbf{As} - \mathbf{t}) \bmod p$ is small using the approximate range proof. The advantage of this method over using pq as the modulus for the commitment scheme, as above, is that it allows the commitment scheme modulus p to be a prime, and so one needs fewer terms for coefficient masking (see the discussion after (7)), which could save a few kilobytes in the complete proof. A disadvantage is that there is now the extra secret $\mathbf{r}$ term that needs to be dealt with.

Useful Extensions. While we concentrated on proving the smallness of the ℓ_2-norm of a vector $\vec{s}$ (or more generally the knowledge of the inner product between two vectors), it is also possible to use our techniques to prove many other inter-vector relations. In particular, a useful relation (e.g. if dealing with general functions/circuits) is proving the knowledge of the component-wise product $\vec{r} \circ \vec{s}$. This can be generally accomplished by proving a polynomial product over a ring $\mathcal{R}_p$ of two vectors $\mathbf{r}$ and $\mathbf{s}$ whose CRT coefficients are $\vec{r}$ and $\vec{s}$. The important thing is to choose a prime p such that the polynomial $X^d + 1$ factors into linear factors modulo p. As mentioned above, by simply subtracting off the remainder as in (12), one can use different moduli for the commitment scheme for the relations that we would like to prove. Thus one can choose a "CRT-friendly" modulus for the underlying relation, while using a modulus that allows the polynomial differences to be invertible (so not a CRT-friendly one) for the commitment scheme.

We also point out that proving inner products can be directly used to prove another very natural function – showing that all the coefficients of a vector are from the set $\{0, 1\}$. For this, one uses the observation that $\vec{s}$ has coefficients in $\{0, 1\}$ if and only if $\langle \vec{s}, \vec{1} - \vec{s} \rangle = 0$. And since given a commitment for $\vec{s}$, one can maul it into a commitment to $\vec{1} - \vec{s}$, one can generically apply the aforementioned protocol in Figure 8.

Using Other Rings. In proving that the norm of a polynomial s was small, we exploited the fact that in the ring $\mathcal{R}$, $s\widetilde{(X^{-1})} \cdot s = \|s\|^2$ and that $s(X^{-1})$ was an automorphism. In Section 7, we show that the same high level ideas can also be made to work for rings that don't have this algebraic structure. Specifically, for all rings $R = \mathbb{Z}[X]/(X^d + f_{d-1}X^{d-1} + \ldots + f_1X \pm 1)$, there exists a linear function $g : R \to R$ such that $\widetilde{g(r)} \cdot s$ is equal to $\langle \vec{r}, \vec{s} \rangle$. If g is not an automorphism, then proving knowledge of $\|s\|^2 = \widetilde{g(s)} \cdot s$ would require the prover to commit to both s and $g(s)$, and then also prove the linear relationship between the commitments of s and $g(s)$. Opening two commitments instead of one will increase the proof size, but this is slightly mitigated by the fact that the challenges no longer need to be restricted to be fixed under any automorphism.

Sample Constructions. In Section 6, we present various instantiations of lattice-based primitives that can be constructed using our zero-knowledge proof system. We now give a very high-level description of a group signature scheme. In a group signature scheme, the Setup Authority uses a master secret keys to distribute member secret keys to the members of the group. The members can then use their secret keys to sign messages on behalf of the group. An entity known as the Opener (or group manager) also has a special secret key that allows him to obtain the identity of the signer of any message. The privacy criterion states that it should be impossible, for everyone but the Opener, to trace back a signature to the particular user, nor link that two signatures were signed by the same user. Conversely, the traceability requirement states that every message signed by a user with identity μ will get traced back to him by the Opener. Group signatures are an interesting primitive in their own right, but are particularly useful in determining the practicality of zero-knowledge proofs as they contain some ingredients which are prevalent throughout privacy-based cryptography.

We show how we can use our improved ZK proof to construct a lattice-based group signature following the framework of [dPLS18, LNPS21]. The master public key is $[\mathbf{A} \mid \mathbf{B}], \mathbf{u}$, and the secret key of a group member with identity μ is a short vector $\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix}$ such that

$$[\mathbf{A} \mid \mathbf{B} + \mu\mathbf{G}] \cdot \begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix} = \mathbf{u} \bmod q. \tag{13}$$

The setup authority with a trapdoor for the lattice $\mathcal{L} = \{\mathbf{x} \,:\, [\mathbf{A} \mid \mathbf{B}] \cdot \mathbf{x} = \mathbf{0} \bmod q\}$ can create such short vectors which are distributed according to a discrete Gaussian distribution [ABB10, MP12].

The group member's signature of a message consists of a Module-LWE encryption of his identity μ as

$$\begin{bmatrix}\mathbf{A}'\\ \mathbf{b}\end{bmatrix} \cdot \mathbf{r} + \begin{bmatrix}0\\ \lceil p/2 \rfloor \mu\end{bmatrix} = \mathbf{t} \bmod p, \tag{14}$$

where $\mathbf{A}', \mathbf{b}$ is the public key (of the Opener) and $\mathbf{r}$ is the randomness, together with a a ZKPoK that he knows $\mu, \mathbf{r}$, and $\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\end{bmatrix}$ satisfying (13) and (14). The message that the user is signing is, as usual, put into the input of the hash function used in the Fiat-Shamir transform of the ZKPoK.

To create this signature, the user commits to $\mathbf{s}_1, \mathbf{s}_2, \mathbf{r}, \mu$ in the "Ajtai" part of the ABDLOP commitment (5). He then proves that the norms of $\mathbf{s}_1, \mathbf{s}_2, \mathbf{r}$ are small, that μ has 0/1 coefficients, and that (14) and (13) hold. Notice that (14) is just a linear equation and proving (13) is proving the quadratic relation $\mathbf{A}\mathbf{s}_1 + \mathbf{B}\mathbf{s}_2 + \mathbf{G}\mu\mathbf{s}_2 = \mathbf{u} \bmod q$. All of these proofs fit into the appropriate functions in the protocol in Figure 10 and the full description of the group signature is given in Section 6.4.

The security of the scheme rests on the fact that creating a valid proof on a μ that is not the user's identity implies having a solution to (13) on a new identity, which is directly equivalent to breaking the ABB signature scheme [ABB10, MP12], which in turn implies breaking the Module-SIS problem. Prior to this work, proving tight bounds on the ℓ_2 norm of polynomial vectors with somewhat large coefficients was not very efficient, and so constructions of group signature schemes using this approach [dPLS18, LNPS21] did not prove (13), but rather proved an approximate version of it as in (2) – i.e. they proved knowledge of $\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, c$ satisfying

$$[\mathbf{A} \mid \mathbf{B} + \mu\mathbf{G}] \cdot \begin{bmatrix}\bar{\mathbf{s}}_1\\ \bar{\mathbf{s}}_2\end{bmatrix} = c\mathbf{u} \bmod q, \tag{15}$$

where $\|\bar{\mathbf{s}}_i\| \gg \|\mathbf{s}_i\|$.

A consequence of being only able to prove the above is a vicious cycle of the larger norms and the presence of c resulting in a larger extracted solution to the Module-SIS problem, which in turn requires having a larger modulus for SIS security, which then also requires a larger lattice dimension for LWE security. Furthermore, because these schemes relied on the verifiable encryption scheme of [LN17], they also did not prove (14), but rather an approximate version of it as in (2). The implication is that in order to decrypt, the Opener needed to guess the unknown c, which in expectation requires the same number of guesses as the adversary's number of calls to the random oracle during the proof. Thus special care would be needed to instantiate the scheme in an environment that would not allow the adversary to be able to have too much time to try and forge a signature. We believe that efficiently eliminating this requirement in all lattice-based schemes requiring a verifiable encryption scheme is a notable improvement on the state of affairs.

	Public Key Size	Signature Size	Opening Time Independent of Adversary's Forgery Time
[LNPS21]	96KB	203KB	×
This Work	48KB	92KB	✓

Table 1: Our group signature and that of [LNPS21].

We compare the instantiation of the group signature from this paper to the previously most efficient one from [LNPS21] in Table 1. We mention that there are also tree-based group signatures (e.g. [ESZ21, BDK$^+$21]) which have shorter outputs for small group sizes, but have the disadvantage that the signing time, verification time, and public key size are linear in the group size. The signature length of these schemes also grows slightly with the group size, and for groups having more than $\approx 2^{21}$ members, our scheme has a comparable signature size (in addition to a much smaller public key and signing/verification times).

	Proof Size
[LNS21a]	33KB
This Work	14KB

	Ciphertext Size	Proof Size	Decryption Time Independent of Forgery Time
[LN17]	9KB	9KB	×
[LNS21a][11]	4KB	33 - 44KB	✓
This Work	1KB	19KB	✓

Table 2: The table on the left compares the difference in proof size of proving knowledge of short $\vec{s}, \vec{e}$ satisfying $A\vec{s} + \vec{e} = \vec{t} \bmod q$, where $A \in \mathbb{Z}_q^{1024 \times 1024}$ and $q \approx 2^{32}$, and $\|(\vec{s}, \vec{e})\| \leqslant \sqrt{2048}$. The protocol from [LNS21a] needs to make the additional restriction that all the coefficients in $\vec{s}, \vec{e}$ are from $\{-1, 0, 1\}$. The table on the right compares our instantiation of a verifiable encryption scheme from this paper with [LN17] and [LNS21a].

Part of the group signature includes a verifiable encryption scheme, in which the encryptor proves that the encryption is valid. When looked at separately, this scheme has a similar size to the one from [LN17], but with the noticeable advantage of not having a dependency between the decryption time and the adversary's forgery time. We also give a comparison of the proof size for the basic system in (1) between our proof system and the prior best one from [LNS21a] that followed the framework of [ALS20] and [ENS20]. The comparisons for the verifiable encryption scheme and the basic proof system are in table 2 and detailed descriptions of the proofs can be found in Sections 6.2 and 6.3.

Acknowledgements. We would like to thank Ward Beullens for generalising Lemma 2.15 for all powers-of-two k (initially, the lemma only covered $k = 1$) and also Damien Stehlé and Elena Kirshanova for their very useful feedback. This work is supported by the EU H2020 ERC Project 101002845 PLAZA.

2 Preliminaries

2.1 Notation

Denote $\mathbb{Z}_p$ to be the ring of integers modulo p. Let $q = q_1, \ldots, q_n$ be a product of n odd primes where $q_1 < q_2 < \ldots < q_n$. Usually, we pick $n = 1$ or $n = 2$. We write $\vec{v} \in \mathbb{Z}_q^m$ to denote vectors over a ring $\mathbb{Z}_q$. Matrices over $\mathbb{Z}_q$ will be written as regular capital letters R. By default, all vectors are column vectors. We write $\vec{v}||\vec{w}$ for a usual concatenation of $\vec{v}$ and $\vec{w}$ (which is still a column vector). For $\vec{v}, \vec{w} \in \mathbb{Z}_q^k$, $\vec{v} \circ \vec{w}$ is the usual component-wise multiplication. For simplicity, we denote $\vec{u}^2 = \vec{u} \circ \vec{u}$. We write $x \leftarrow S$ when $x \in S$ is sampled uniformly at random from the finite set S and similarly $x \leftarrow D$ when x is sampled according to the distribution D. Let $[n] := \{1, \ldots, n\}$.

For a power of two d and a positive integer p, denote $\mathcal{R}$ and $\mathcal{R}_p$ respectively to be the rings $\mathbb{Z}[X]/(X^d + 1)$ and $\mathbb{Z}_p[X]/(X^d + 1)$. Lower-case letters denote elements in $\mathcal{R}$ or $\mathcal{R}_p$ and bold lower-case (resp. upper-case) letters represent column vectors (resp. matrices) with coefficients in $\mathcal{R}$ or $\mathcal{R}_p$. For a polynomial $f \in \mathcal{R}_p$, denote $\vec{f} \in \mathbb{Z}_q^d$ to be the coefficient vector of f. By default, we write its i-th coefficient as its corresponding regular font letter subscript i, e.g. $f_{d/2} \in \mathbb{Z}_p$ is the coefficient corresponding to $X^{d/2}$ of $f \in \mathcal{R}_p$. For the constant coefficient, however, we will denote $\tilde{f} := f_0 \in \mathbb{Z}_p$. The ring $\mathcal{R}$ has a group of automorphisms $\mathsf{Aut}(\mathcal{R})$

[11] This paper presents a verifiable *decryption* scheme, but the proof size for a verifiable encryption scheme constructed in the same manner would be similar. At the very least, it needs to be as large as the proof of the basic equation in (1).

that is isomorphic to $\mathbb{Z}_{2d}^{\times}$. Let $\sigma_i \in \mathsf{Aut}(\mathcal{R}_q)$ be defined by $\sigma_i(X) = X^i$. For readability, we denote for an arbitrary vector $\mathbf{m} \in \mathcal{R}^k$:

$$\sigma_i(\mathbf{m}) := (\sigma_i(m_1), \ldots, \sigma_i(m_k))$$

and similarly $\sigma_i(\mathbf{R})$ for any matrix $\mathbf{R}$. When we write $\langle \mathbf{u}, \mathbf{v} \rangle \in \mathbb{Z}$ for $\mathbf{u}, \mathbf{v} \in \mathcal{R}^k$, we mean the inner product of their corresponding coefficient vectors.

For an element $w \in \mathbb{Z}_q$, we write $\|w\|_\infty$ to mean $|w \bmod^{\pm} q|$. Define the ℓ_∞ and ℓ_p norms for $w = w_0 + w_1 X + \ldots + w_{d-1}X^{d-1} \in \mathcal{R}$ as follows:

$$\|w\|_\infty = \max_j \|w_j\|_\infty, \quad \|w\|_p = \sqrt[p]{\|w_0\|_\infty^p + \ldots + \|w_{d-1}\|_\infty^p}.$$

If $\mathbf{w} = (w_1, \ldots, w_m) \in \mathcal{R}^k$, then

$$\|\mathbf{w}\|_\infty = \max_j \|w_j\|_\infty, \quad \|\mathbf{w}\|_p = \sqrt[p]{\|w_1\|^p + \ldots + \|w_k\|^p}.$$

By default, $\|\mathbf{w}\| := \|\mathbf{w}\|_2$. Similarly, we define the norms for vectors over $\mathbb{Z}_q$. Denote $S_\gamma = \{x \in \mathcal{R}_q : \|x\|_\infty \leqslant \gamma\}$.

2.2 Probability Distributions

We first define the discrete Gaussian distribution used for the rejection sampling.

Definition 2.1. *The* discrete Gaussian distribution *on $\mathcal{R}^\ell$ centered around $\mathbf{v} \in \mathcal{R}^\ell$ with standard deviation $\mathfrak{s} > 0$ is given by*

$$D^\ell_{\mathbf{v},\mathfrak{s}}(\mathbf{z}) = \frac{e^{-\|\mathbf{z}-\mathbf{v}\|^2/2\mathfrak{s}^2}}{\sum_{\mathbf{z}' \in \mathcal{R}^\ell} e^{-\|\mathbf{z}'\|^2/2\mathfrak{s}^2}}.$$

When it is centered around $\mathbf{0} \in \mathcal{R}^\ell$ we write $D^\ell_{\mathfrak{s}} = D^\ell_{\mathbf{0},\mathfrak{s}}$.

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

Lemma 2.2. *Let $\mathbf{z} \leftarrow D^m_{\mathfrak{s}}$. Then* $\Pr\left[\|\mathbf{z}\| > t \cdot \mathfrak{s}\sqrt{md}\right] < \left(te^{\frac{1-t^2}{2}}\right)^{md}$.

Next, we recall the binomial distribution.

Definition 2.3. *The binomial distribution with a positive integer parameter κ, written as Bin_κ is the distribution $\sum_{i=1}^{\kappa}(a_i - b_i)$, where $a_i, b_i \leftarrow \{0,1\}$. The variance of this distribution is $\kappa/2$ and it holds that* $\mathsf{Bin}_{\kappa_1} \pm \mathsf{Bin}_{\kappa_2} = \mathsf{Bin}_{\kappa_1+\kappa_2}$.

2.3 Cyclotomic Rings

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

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

where σ_i is defined by $\sigma_i(X) = X^i$. Consider $\sigma_{-1} \in \mathsf{Aut}(\mathcal{R}_q)$. We define the following map $\mathsf{T} : \mathbb{Z}^{kd} \times \mathbb{Z}^{kd} \to \mathcal{R}$ which given vectors $\vec{a} = (a_0, \ldots, a_{kd-1})$ and $\vec{b} = (b_0, \ldots, b_{kd-1})$, it outputs:

$$\mathsf{T}(\vec{a}, \vec{b}) := \sum_{i=0}^{k-1} \sigma_{-1}\left(\sum_{j=0}^{d-1} a_{id+j}X^j\right) \cdot \left(\sum_{j=0}^{d-1} b_{id+j}X^j\right) \in \mathcal{R}. \tag{16}$$

As briefly described in the introduction and in more detail in Section 5, we will make use of the following simple property of T.

Lemma 2.4. *Let $\vec{a}, \vec{b} \in \mathbb{Z}^{kd}$ for $k \geqslant 1$. Then, the constant coefficient of $\mathsf{T}\left(\vec{a}, \vec{b}\right)$ is equal to $\langle \vec{a}, \vec{b} \rangle$.*

In Section 7 we show how to construct functions T with the same property for different underlying rings than $\mathbb{Z}[X]/(X^d + 1)$.

Suppose each (q_i) splits into 2 prime ideals of degree $d/2$ in $\mathcal{R}$. This means $X^d + 1 \equiv \varphi_0 \varphi_1 \pmod{q_i}$ with irreducible polynomials φ_j of degree $d/2$ modulo q_i. We assume that $\mathbb{Z}_q$ contains a primitive 4-th root of unity $\zeta_i \in \mathbb{Z}_q$ but no elements whose order is a higher power of two, i.e. $q_i - 1 \equiv 4 \pmod 8$. Therefore, we have

$$X^d + 1 \equiv \left(X^{\frac{d}{2}} - \zeta_i\right)\left(X^{\frac{d}{2}} - \zeta_i^3\right) \pmod{q_i}. \tag{17}$$

We recall the main result by Lyubashevsky and Seiler [LS18] which says that small polynomials over $\mathcal{R}_{q_i}$ are invertible.

Lemma 2.5 (DBLP:conf/eurocrypt/LyubashevskyN17,[LS18]). *Let $p \equiv 5 \pmod 8$ be a prime. Then, any $f \in \mathcal{R}_p$ which satisfies either $0 < \|f\|_\infty < \frac{1}{\sqrt{2}} p^{1/2}$ or $0 < \|f\| < p^{1/2}$ has an inverse in $\mathcal{R}_p$.*

In this paper we will be working with polynomials in $\mathcal{R}_p$ which are stable under the σ_{-1} automorphism. The following result says that for specific primes p, if $c \in \mathcal{R}_p$ satisfies $\sigma_{-1}(c) = c$ and c is non-zero then c is invertible over $\mathcal{R}_p$.

Lemma 2.6. *Let $p \equiv 5 \pmod 8$ be a prime. Take any $c \in \mathcal{R}_p$ such that $\sigma_{-1}(c) = c$. Then, c is invertible over $\mathcal{R}_p$ if and only if $c \neq 0$.*

Proof. Since p is congruent to 5 modulo 8, we can factor the polynomial $X^d + 1$ modulo p as

$$X^d + 1 \equiv (X^{d/2} - r)(X^{d/2} + r) \pmod p$$

for some $r \in \mathbb{Z}_p$ where polynomials $X^{d/2} \pm r$ are irreducible modulo p. Since $\sigma_{-1}(c) = c$, we can write c as

$$c = c_0 + c_1 X + \ldots + c_{d/2-1} X^{d/2-1} - c_{d/2-1} X^{d/2+1} - \ldots - c_1 X^{d-1}.$$

Now, we observe that

$$c \bmod (p, X^{d/2} \pm r) = c_0 + \sum_{i=1}^{d/2-1} (c_i \pm r c_{d/2-i}) X^i.$$

Suppose $c \neq 0$. Then, one of the coefficients $c_0, \ldots, c_{d/2-1} \in \mathbb{Z}_p$ is non-zero, say c_i. Note that if $i = d/4$ then $c_i \pm r c_{d/2-i}$ is not zero since $r \neq \pm 1$. Now, consider the case $i \neq d/4$. We claim that for any sign $b \in \{-1, 1\}$, either $c_i - brc_{d/2-i}$ or $c_{d/2-i} - brc_i$ is not zero. Indeed, assume both of them were equal to zero, concretely $c_i = brc_{d/2-i}$ and $c_{d/2-i} = brc_i$ for $b \in \{-1, 1\}$. Then we would obtain

$$c_i = brc_{d/2-i} = b^2 r^2 c_i = r^2 c_i = -c_i$$

which is a contradiction since $c_i \neq 0$. Hence, we deduce that $c \bmod (p, X^{d/2} - r)$ and $c \bmod (p, X^{d/2} + r)$ are non-zero. Therefore, by the Chinese Remainder Theorem, we conclude that c has an inverse in $\mathcal{R}_p$. □

Denote $\mathcal{R}_q^\times$ to be the set of invertible polynomials in $\mathcal{R}_q$. Recall that a polynomial f is invertible in $\mathcal{R}_q$ if and only if for each $i \in [n]$, $f \bmod q_i$ is invertible in $\mathcal{R}_{q_i}$. Hence, Lemma 2.6 says that if $f \in \mathcal{R}_q$ satisfies $0 < \|f\|_\infty < q_1$ and $\sigma_{-1}(f) = f$ then $f \in \mathcal{R}_q^\times$.

2.4 Approximate Range Proofs

In some cases, we will not need to prove a tight bound on the norm of a vector, but it will be enough for us to prove that its coefficients are small. The application of this proof is in showing that the inner product of a vector is small enough that it is the same modulo q and over the integers. The intuition for obtaining such proofs is the observation that the inner product (modulo q) of a random vector $\vec{r} \leftarrow \mathsf{Bin}_1^m$ with an arbitrary vector $\vec{w} \in \mathbb{Z}_q^m$ is less than $\frac{1}{2}\|\vec{w}\|$ with probability at most $\frac{1}{2}$ [BL17]. The slightly more general lemma from [LNS21a] that we will be using is

Lemma 2.7. *Let $\vec{w} \in \mathbb{Z}_q^m$ and $\vec{y} \in \mathbb{Z}_q^k$. Then*

$$\Pr_{R \leftarrow \mathsf{Bin}_1^{k \times m}} \left[\|R\vec{w} + \vec{y}\|_\infty < \frac{1}{2}\|\vec{w}\|_\infty \right] \leqslant 2^{-k}.$$

For a large m, the gap between the upper bound ($m \cdot \|\vec{w}\|_\infty$) and the lower bound ($\frac{1}{2} \cdot \|\vec{w}\|_\infty$) is a factor of m. One can probabilistically lower it to $O(\sqrt{m})$, but there is a way to get a constant-size gap by considering the ℓ_2-norm. A well-known result of Johnson and Lindenstrauss says that any set of k points in m-dimensional Euclidean space can be embedded into a much smaller ℓ-dimensional Euclidean space, where $\ell = O(\log m)$ and independent of k, so that all pairwise distances are preserved within an arbitrarily small factor. In practical scenarios, such embeddings are simply random projections.

Recently, Gentry et al. [GHL21] applied this result in the context of proving shortness of a committed vector $\vec{w} \in \mathbb{Z}^m$. Concretely, the idea is to choose a random rectangular matrix $R \leftarrow \mathsf{Bin}_2^{256 \times m}$ and prove that the projection $\vec{v} = R\vec{w}$ with respect to R has small norm. When m not too small, substituting the continuous normal distribution by a binomial one (with the same variance) should heuristically result in very similar tail bounds. In [GHL21], arguments regarding the moments of Bin_1 and experimental results were used to support this heuristic. Using the fact that the distribution $\|R \cdot 1^d\|$, where entries of R are chosen from the normal distribution with mean 0 and variance $\kappa/2$, is the scaled χ^2 distribution with 256 degrees of freedom, i.e. $\frac{\kappa}{2} m \cdot \chi^2[256]$, we obtain the following (heuristic) generalization of [GHL21][Corollary 3.2] (we only use this lemma for the case of $\kappa = 1, 2$).

Lemma 2.8. *Under the heuristic substitution of Bin_κ with the normal distribution of variance $\kappa/2$, for any $\vec{w} \in \mathbb{Z}^m$,*

1. $\Pr_{R \leftarrow \mathsf{Bin}_\kappa^{256 \times m}} \left[\|R\vec{w}\|^2 < \|\vec{w}\|^2 \cdot 13 \cdot \kappa \right] \approx \Pr_{y \leftarrow \chi^2[256]} [y < 26] \leqslant 2^{-256}$
2. $\Pr_{R \leftarrow \mathsf{Bin}_\kappa^{256 \times m}} \left[\|R\vec{w}\|^2 > \|\vec{w}\|^2 \cdot 337 \cdot \kappa \right] \approx \Pr_{y \leftarrow \chi^2[256]} [y > 674] \leqslant 2^{-128}.$

Gentry et al. construct a proof for the shortness of a long vector $\vec{w} \in \mathbb{Z}_q^m$ as follows. They first commit to the random projection $\vec{v} := R\vec{w} \in \mathbb{Z}_q^{256}$, where $R \leftarrow \mathsf{Bin}_1^{256 \times m}$, and prove that the norm of $\vec{v}$ is small and that $\vec{v}$ is a projection of $\vec{w}$. Then, [GHL21][Corollary 3.3] says that if $\|\vec{v}\| < b\sqrt{30}$, where $b \leqslant q/(45m)$, then we must have $\|\vec{w}\| \leqslant b$ (with an overwhelming probability). In our protocols, we will need a modified version of this result which says that for every vector $\vec{y} \in \mathbb{Z}_q^{256}$, if $\|R\vec{w} + \vec{y}\|$ is small then we must have that $\|\vec{w}\|$ is small. Even though we believe this generalisation is true for the constants described in [GHL21][Corollary 3.3] (and a generalization for the analogous result in the ℓ_∞ norm is true [LNS21b]), we don't know how to extend the proof to this setting. We thus provide a modified proof which results in slightly worse bounds.

Lemma 2.9. *Fix $m, P \in \mathbb{N}$ and a bound $b \leqslant P/41m$, and let $\vec{w} \in [\pm P/2]^m$ with $\|\vec{w}\| \geqslant b$, and let $\vec{y}$ be an arbitrary vector in $[\pm P/2]^m$. Then*

$$\Pr_{R \leftarrow \mathsf{Bin}_1^{256 \times m}} \left[\|R\vec{w} + \vec{y} \bmod P\| < \frac{1}{2} b\sqrt{26} \right] < 2^{-128}.$$

Proof. We first prove an analogous result to [GHL21][Corollary 3.3] with error 2^{-256} rather than 2^{-128}.

Lemma 2.10. *Fix $m, P \in \mathbb{N}$ and a bound $b \leqslant P/41m$, and let $\vec{w} \in [\pm P/2]^m$ with $\|\vec{w}\| \geqslant b$. Then*

$$\Pr_{R \leftarrow \mathsf{Bin}_2^{256 \times m}} [\|R\vec{w} \bmod P\| < b\sqrt{26}] < 2^{-256}.$$

Proof. We have two cases:

– The first case is when $\|\vec{w}\|_\infty \geqslant P/4m$. Let i be an index of an entry in $\vec{w}$ with magnitude at least $P/4m$, and consider any row of R (denoted $\vec{r}$): After choosing all but the i'th entry in $\vec{r}$, at most one of the three values $\{0, \pm 1\}$ yields $|\langle \vec{w}, \vec{r} \rangle \bmod P| < P/8m$. Since the total probability of any two of those is at least 1/2 (i.e. $\Pr[0] = 3/8$ and $\Pr[\pm 1] = 1/4$), we have that the probability that all the rows of R yield entries smaller than $P/8m$ is at most $(1/2)^{256}$. Since $b \leqslant P/41m$ then $P/8m > b\sqrt{26}$ and therefore

$$\Pr_{R \leftarrow \mathsf{Bin}_2^{256\times m}}[\|R\vec{w} \bmod P\| < b\sqrt{26}] \leqslant \Pr_{R}[\|R\vec{w} \bmod P\| < P/8m] \ \leqslant\ 2^{-256}.$$

– The second case is when $\|\vec{w}\|_\infty < P/4m$. Here with probability one we have $R\vec{w} \in [\pm P/2]^{256}$, so mod-$P$ reduction has no effect and the assertion follows directly from Lemma 2.8.

□

We now use the above Lemma to prove Lemma 2.9. Suppose for contradiction that for some $\vec{w}, \vec{y}$,

$$\Pr_{R \leftarrow \mathsf{Bin}_1^{256\times m}}\left[\|R\vec{w} + \vec{y} \bmod P\| < \frac{1}{2} b\sqrt{26}\right] \geqslant 2^{-128}.$$

This implies that

$$\Pr_{R_1, R_2 \leftarrow \mathsf{Bin}_1^{256\times m}}\left[\|R_1\vec{w} + \vec{y} \bmod P\| < \frac{1}{2} b\sqrt{26} \ \wedge\ \|R_2\vec{w} + \vec{y} \bmod P\| < \frac{1}{2} b\sqrt{26}\right] \geqslant 2^{-256}.$$

By the triangle inequality (which holds even modulo P), we have

$$\Pr_{R_1, R_2 \leftarrow \mathsf{Bin}_1^{256\times m}}\left[\|(R_1 - R_2)\vec{w} \bmod P\| < b\sqrt{26}\right] \geqslant 2^{-256}.$$

Since the distribution of $R_1 - R_2$ is exactly $\mathsf{Bin}_2^{256\times m}$, the above implies that

$$\Pr_{R \leftarrow \mathsf{Bin}_2^{256\times m}}\left[\|R\vec{w} \bmod P\| < b\sqrt{26}\right] \geqslant 2^{-256},$$

which is a contradiction with the statement of Lemma 2.10. □

2.5 Module-SIS and Module-LWE Problems

Security of the [BDL+18] commitment scheme used in our protocols relies on the well-known computational lattice problems, namely Module-LWE (MLWE) and Module-SIS (MSIS) [LS15, DKL+18]. Both problems are defined over $\mathcal{R}_q$.

Definition 2.11 ($\mathsf{MSIS}_{\kappa,m,B}$)**.** *Given $\mathbf{A} \leftarrow \mathcal{R}_q^{\kappa\times m}$, the* Module-SIS *problem with parameters $\kappa, m > 0$ and $0 < B < q$ asks to find $\mathbf{z} \in \mathcal{R}_q^m$ such that $\mathbf{Az} = \mathbf{0}$ over $\mathcal{R}_q$ and $0 < \|\mathbf{z}\| \leqslant B$. An algorithm $\mathcal{A}$ is said to have advantage ϵ in solving $\mathsf{MSIS}_{\kappa,m,B}$ if*

$$\Pr\left[0 < \|\mathbf{z}\|_\infty \leqslant B \ \wedge\ \mathbf{Az} = \mathbf{0} \,\middle|\, \mathbf{A} \leftarrow \mathcal{R}_q^{\kappa\times m};\ \mathbf{z} \leftarrow \mathcal{A}(\mathbf{A})\right] \geqslant \epsilon.$$

Definition 2.12 ($\mathsf{MLWE}_{m,\lambda,\chi}$)**.** *The* Module-LWE *problem with parameters $m, \lambda > 0$ and an error distribution χ over $\mathcal{R}$ asks the adversary $\mathcal{A}$ to distinguish between the following two cases: 1) $(\mathbf{A}, \mathbf{A}\boldsymbol{s} + \boldsymbol{e})$ for $\mathbf{A} \leftarrow \mathcal{R}_q^{m\times\lambda}$, a secret vector $\boldsymbol{s} \leftarrow \chi^\lambda$ and error vector $\boldsymbol{e} \leftarrow \chi^m$, and 2) $(\mathbf{A}, \boldsymbol{b}) \leftarrow \mathcal{R}_q^{m\times\lambda} \times \mathcal{R}_q^m$. Then, $\mathcal{A}$ is said to have advantage ϵ in solving $\mathsf{MLWE}_{m,\lambda,\chi}$ if*

$$\begin{aligned}\big|&\Pr\left[b = 1 \,\middle|\, \mathbf{A} \leftarrow \mathcal{R}_q^{m\times\lambda};\ \boldsymbol{s} \leftarrow \chi^\lambda;\ \boldsymbol{e} \leftarrow \chi^m;\ b \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{A}\boldsymbol{s} + \boldsymbol{e})\right] \\ &- \Pr\left[b = 1 \,\middle|\, \mathbf{A} \leftarrow \mathcal{R}_q^{m\times\lambda};\ \boldsymbol{b} \leftarrow \mathcal{R}_q^m;\ b \leftarrow \mathcal{A}(\mathbf{A}, \boldsymbol{b})\right]\big| \geqslant \epsilon.\end{aligned} \tag{18}$$

We also recall the (simplified) Extended Module-LWE problem [LNS21a].

Definition 2.13 (Extended-$\mathsf{MLWE}_{m,\lambda,\chi,\mathcal{C},\mathfrak{s}}$). *The* Extended Module-LWE *problem with parameters $m, \lambda > 0$, probability distribution χ over $\mathcal{R}_q$, challenge space $\mathcal{C} \subseteq \mathcal{R}_q$ and the standard deviation $\mathfrak{s}$ asks the adversary $\mathcal{A}$ to distinguish between the following two cases:*

1. $(\mathbf{B}, \mathbf{Br}, c, \mathbf{z}, \mathsf{sign}(\langle \mathbf{z}, c\mathbf{r}\rangle))$ *for* $\mathbf{B} \leftarrow \mathcal{R}_q^{m\times(m+\lambda)}$*, a secret vector* $\mathbf{r} \leftarrow \chi^{m+\lambda}$ *and* $\mathbf{z} \leftarrow D_{\mathfrak{s}}^{(m+\lambda)}, c \leftarrow \mathcal{C}$
2. $(\mathbf{B}, \mathbf{u}, c, \mathbf{z}, \mathsf{sign}(\langle \mathbf{z}, c\mathbf{r}\rangle))$ *for* $\mathbf{B} \leftarrow \mathcal{R}_q^{m\times(m+\lambda)}, \mathbf{u} \leftarrow \mathcal{R}_q^m,\ \mathbf{z} \leftarrow D_{\mathfrak{s}}^{(m+\lambda)}, c \leftarrow \mathcal{C},$

where $\mathsf{sign}(a) = 1$ *if* $a \geqslant 0$ *and* 0 *otherwise. Then,* $\mathcal{A}$ *is said to have advantage* ϵ *in solving Extended-*$\mathsf{MLWE}_{m,\lambda,\chi,\mathcal{C},\mathfrak{s}}$ *if*

$$\Big|\Pr\Big[b = 1 \,\Big|\, \mathbf{B} \leftarrow \mathcal{R}_q^{m\times(m+\lambda)};\ \mathbf{r} \leftarrow \chi^{m+\lambda};\ \mathbf{z} \leftarrow D_{\mathfrak{s}}^{(m+\lambda)};\ c \leftarrow \mathcal{C};\ b \leftarrow \mathcal{A}(\mathbf{B}, \mathbf{Br}, \mathbf{z}, c, s)\Big]$$
$$- \Pr\Big[b = 1 \,\Big|\, \mathbf{B} \leftarrow \mathcal{R}_q^{m\times\lambda};\ \mathbf{u} \leftarrow \mathcal{R}_q^m;\ \mathbf{z} \leftarrow D_{\mathfrak{s}}^{(m+\lambda)}; c \leftarrow \mathcal{C}\,;\ b \leftarrow \mathcal{A}(\mathbf{B}, \mathbf{u}, \mathbf{z}, c, s)\Big]\Big| \geqslant \epsilon.$$

where $s = \mathsf{sign}(\langle \mathbf{z}, c\mathbf{r}\rangle)$.

2.6 Rejection Sampling

In lattice-based zero-knowledge proofs, the prover will want to output a vector $\mathbf{z}$ whose distribution should be independent of a secret message/randomness vector $\mathbf{r}$, so that $\mathbf{z}$ cannot be used to gain any information on the prover's secret. During the protocol, the prover computes $\mathbf{z} = \mathbf{y} + c\mathbf{r}$ where $\mathbf{r}$ is either a secret vector or randomness used to commit to the prover's secret, $c \leftarrow \mathcal{C}$ is a challenge polynomial, and $\mathbf{y}$ is a "masking" vector. In order to remove the dependency of $\mathbf{z}$ on $\mathbf{r}$, one applies *rejection sampling* [Lyu12].

Lemma 2.14 (Rejection Sampling [Lyu12, DDLL13, LNS21a]). *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. Fix the standard deviation* $\mathfrak{s} = \gamma T$*. Then, the following statements hold.*

1. *Let* $M = \exp(14/\gamma + 1/(2\gamma^2))$*. Now, sample* $\mathbf{v} \leftarrow \rho$ *and* $\mathbf{y} \leftarrow D_{\mathfrak{s}}^\ell$*, set* $\mathbf{z} = \mathbf{y} + \mathbf{v}$*, and run* $b \leftarrow \mathsf{Rej}_1(\mathbf{z}, \mathbf{v}, \mathfrak{s})$ *as defined in Fig. 1. Then, the probability that* $b = 0$ *is at least* $(1 - 2^{-128})/M$ *and the distribution of* $(\mathbf{v}, \mathbf{z})$*, conditioned on* $b = 0$*, is within statistical distance of* 2^{-128} *of the product distribution* $\rho \times D_{\mathfrak{s}}^\ell$*.*
2. *Let* $M = \exp(1/(2\gamma^2))$*. Now, sample* $\mathbf{v} \leftarrow \rho$ *and* $\mathbf{y} \leftarrow D_{\mathfrak{s}}^\ell$*, set* $\mathbf{z} = \mathbf{y} + \mathbf{v}$*, and run* $b \leftarrow \mathsf{Rej}_2(\mathbf{z}, \mathbf{v}, \mathfrak{s})$ *as defined in Fig. 1. Then, the probability that* $b = 0$ *is at least* $1/(2M)$ *and the distribution of* $(\mathbf{v}, \mathbf{z})$*, conditioned on* $b = 0$*, is identical to the distribution* $\mathcal{F}$ *where* $\mathcal{F}$ *is defined as follows: sample* $\mathbf{v} \leftarrow \rho$*,* $\mathbf{z} \leftarrow D_{\mathfrak{s}}^{ld}$ *conditioned on* $\langle \mathbf{v}, \mathbf{z}\rangle \geqslant 0$ *and output* $(\mathbf{v}, \mathbf{z})$*.*
3. *Let* $M = \exp(1/(2\gamma^2))$*. Now, sample* $\mathbf{v} \leftarrow \rho, \beta \leftarrow \{0, 1\}$ *and* $\mathbf{y} \leftarrow D_{\mathfrak{s}}^\ell$*, set* $\mathbf{z} = \mathbf{y} + (-1)^\beta \mathbf{v}$*, and run* $b \leftarrow \mathsf{Rej}_0(\mathbf{z}, \mathbf{v}, \mathfrak{s})$ *as defined in Fig. 2. Then, the probability that* $b = 0$ *is at least* $1/M$ *and the distribution of* $(\mathbf{v}, \mathbf{z})$*, conditioned on* $b = 0$*, is identical to the product distribution* $\rho \times D_{\mathfrak{s}}^\ell$*.*

```
$\mathrm{Rej}_1(\vec{z}, \vec{v}, \mathfrak{s})$
$u \leftarrow [0, 1)$
If $u > \frac{1}{M} \cdot \exp\left(\frac{-2\langle \vec{z}, \vec{v}\rangle + \|\vec{v}\|^2}{2\mathfrak{s}^2}\right)$
    return 1 (i.e. reject)
Else
    return 0 (i.e. accept)
```

```
$\mathrm{Rej}_2(\vec{z}, \vec{v}, \mathfrak{s})$
If $\langle \vec{z}, \vec{v}\rangle < 0$
    return 1 (i.e. reject)
$u \leftarrow [0, 1)$
If $u > \frac{1}{M} \cdot \exp\left(\frac{-2\langle \vec{z}, \vec{v}\rangle + \|\vec{v}\|^2}{2\mathfrak{s}^2}\right)$
    return 1 (i.e. reject)
Else
    return 0 (i.e. accept)
```

Fig. 1: Two rejection sampling algorithms: the one used generally in previous works [Lyu12] (left) and the one proposed recently in [LNS21a] (right).

We recall how parameters $\mathfrak{s}$ and M in the first statement Lemma 2.14 are selected. Concretely, the repetition rate M is chosen to be an upper-bound on:

$$\frac{D_{\mathfrak{s}}^{\ell}(\mathbf{z})}{D_{\mathbf{v},\mathfrak{s}}^{\ell}(\mathbf{z})} = \exp\left(\frac{-2\langle \mathbf{z}, \mathbf{v}\rangle + \|\mathbf{v}\|^2}{2\mathfrak{s}^2}\right) \leqslant \exp\left(\frac{28\mathfrak{s}\|\mathbf{v}\| + \|\mathbf{v}\|^2}{2\mathfrak{s}^2}\right) = M. \tag{19}$$

Here, we used the fact which says that with probability at least $1 - 2^{128}$ we have $|\langle \mathbf{z}, \mathbf{v}\rangle| < 14\mathfrak{s}\|\mathbf{v}\|$ for $\mathbf{z} \leftarrow D_{\mathfrak{s}}^{\ell}$ [Ban93, Lyu12]. Hence, by setting $\mathfrak{s} = 13\|\mathbf{v}\|$ we obtain $M \approx 3$.

Recently, Lyubashevsky et al. [LNS21a] proposed a modified rejection sampling algorithm (see $\mathsf{Rej}_2(\boldsymbol{z}, \boldsymbol{v}, \mathfrak{s})$ in Fig. 1) where it forces $\mathbf{z}$ to satisfy $\langle \mathbf{z}, \mathbf{v}\rangle \geqslant 0$, otherwise it aborts. With this additional assumption, we can set M in the following way:

$$\exp\left(\frac{-2\langle \mathbf{z}, \mathbf{v}\rangle + \|\mathbf{v}\|^2}{2\mathfrak{s}^2}\right) \leqslant \exp\left(\frac{\|\mathbf{v}\|^2}{2\mathfrak{s}^2}\right) = M. \tag{20}$$

Hence, for $M \approx 3$ one would select $\mathfrak{s} = 0.675 \cdot \|\mathbf{v}\|$. Note that the probability for $\mathbf{z} \leftarrow D_{\mathfrak{s}}^{\ell}$ that $\langle \mathbf{z}, \mathbf{v}\rangle \geqslant 0$ is at least 1/2. Hence, the expected number of rejections would be at most $2M = 6$. On the other hand, if one aims for $M = 6$ repetitions using (19), then $\mathfrak{s} = 8 \cdot \|\mathbf{v}\|$. Thus, [LNS21a] manages to reduce the standard deviation by more than a factor of 10. Further, we remark that this method is still not as efficient as using bimodal Gaussians [DDLL13], since even though the value M is calculated exactly as in (20), the expected number of rejections is at most M and not $2M$. We summarise the results from [DDLL13, LNS21a] in the latter two statements of Lemma 2.14.

```
$\mathrm{Rej}_0(\vec{z}, \vec{v}, \mathfrak{s})$
$u \leftarrow [0, 1)$
If $u > \frac{1}{M \exp\left(-\frac{\|\vec{v}\|^2}{2\mathfrak{s}^2}\right) \cosh\left(\frac{\langle \vec{z}, \vec{v}\rangle}{\sigma^2}\right)}$
    return 1 (i.e. *reject*)
Else
    return 0 (i.e. *accept*)
```

Fig. 2: Bimodal rejection sampling [DDLL13].

Finally, we highlight that the procedure in the second statement of Lemma 2.14 reveals the sign of $\langle \mathbf{z}, \mathbf{v}\rangle$. This is still fine when working with "one-time commitments" [LNS21a] since we only leak one bit of information if $\mathbf{v}$ is a randomness vector which is generated every execution. However, secure signature schemes cannot be produced using this method because each generation of a signature reveals some information about the secret key.

By using this technique, zero-knowledge property (or rather commit-and-prove simulatability as described in later sections) of our protocols relies on the (simplified) Extended-MLWE problem [LNS21a] where the adversary is given the additional one bit of information about the secret. We describe this problem in Section 2.5.

2.7 Challenge Space

In our applications, the set $V \subseteq \mathcal{R}^{\ell}$ will consist of vectors of the form $c\mathbf{r}$ where $c \in \mathcal{R}_q$ is sampled from a challenge space $\mathcal{C}$ and $\mathbf{r} \in \mathcal{R}_q^{\ell}$ comes from a set of secret (either randomness or message) vectors. In order to set the standard deviation for rejection sampling, we need to bound the norm of such vectors. Here, we present a new way to bound $\|c\mathbf{r}\|$.

Lemma 2.15. *Let $\mathbf{r} \in \mathcal{R}^{\ell}$ and $c \in \mathcal{R}$. Then, for any power-of-two k, we have $\|c\mathbf{r}\| \leqslant \sqrt[2k]{\|\sigma_{-1}(c^k)\, c^k\|_1} \cdot \|\mathbf{r}\|$.*

Proof. Let $C = \mathsf{rot}(c) \in \mathbb{Z}^{d\times d}$. We simply want to upper-bound the operator norm $\|C\|_2$ of the matrix C. We will use the following two facts from linear algebra. Namely, we have that $\|C\|_2 = \sqrt{\|C^TC\|_2}$ and for every power-of-two k, $\|C^TC\|_2^k = \|(C^TC)^k\|_2$ since C^TC is symmetric. Also, note that for any $u, v \in \mathcal{R}_q$, $\|uv\| \leqslant \|u\|_1 \cdot \|v\|$, and thus $\|\mathsf{rot}(u)\|_2 \leqslant \|u\|_1$. Therefore, using the observation that $C^T = \mathsf{rot}(\sigma_{-1}(c))$, we deduce

$$\|C\|_2^{2k} = \|C^TC\|_2^k = \|(C^TC)^k\|_2 = \|\mathsf{rot}(\sigma_{-1}(c^k)c^k)\|_2 \leqslant \|\sigma_{-1}(c^k)c^k\|_1.$$

Hence, $\|C\|_2 \leqslant \sqrt[2k]{\|\sigma_{-1}(c^k)\, c^k\|_1}$ and thus the statement holds.

□

In order to apply this lemma, we fix a power-of-two k and set the challenge space $\mathcal{C}$ as:

$$\mathcal{C} := \{c \in S_\kappa^\sigma : \sqrt[2k]{\|\sigma_{-1}(c^k)\, c^k\|_1} \leqslant \eta\} \tag{21}$$

where

$$S_\kappa^\sigma := \{c \in S_\kappa : \sigma(c) = c\}. \tag{22}$$

and the $\sigma \in \mathsf{Aut}(\mathcal{R}_q)$ will be specified in our protocols. Also, we denote $\bar{\mathcal{C}} := \{c - c' : c, c' \in \mathcal{C} \text{ and } c \neq c'\}$ to be the set of differences of any two distinct elements in $\mathcal{C}$. In practice, $\sigma \in \{\sigma_1, \sigma_{-1}\}$. We will choose the constants η such that (experimentally) the probability for $c \leftarrow S_\kappa^\sigma$ to satisfy $\sqrt[2k]{\|\sigma_{-1}(c^k)\, c^k\|_1} \leqslant \eta$ is at least 99%. In our experiments, we observe that the bounds in Lemma 2.15 are about $4-6$X larger than the actual norms $\|c\mathbf{r}\|$.

For security of our protocols, we need $\kappa < \frac{1}{2\sqrt{2}}q_1^{1/2}$ to ensure the invertibility property of the challenge space $\mathcal{C}$, i.e. the difference of any two distinct elements of $\mathcal{C}$ is invertible over $\mathcal{R}_q$. Indeed, this property follows from Lemma 2.5. However, if we set $\sigma := \sigma_{-1}$ then we can apply Lemma 2.6 and thus we only need $\kappa < q_1/2$. Secondly, to achieve negligible soundness error under the MSIS assumption, we will need $|\mathcal{C}|$ to be exponentially large. In Table 3 we propose example parameters to instantiate the challenge space $\mathcal{C}$ for different automorphisms σ. Finally, for implementation purposes, in order to sample from $\mathcal{C}$, we simply generate $c \leftarrow S_\kappa^\sigma$ and check whether $\sqrt[2k]{\|\sigma_{-1}(c^k)\, c^k\|_1} \leqslant \eta$. Hence, we cannot choose k to be too large.

σ	d	κ	η	$\lvert S_\kappa^\sigma\rvert$	$\lvert\mathcal{C}\rvert$
σ_1	128	1	27	2^{202}	2^{201}
σ_{-1}	128	2	59	2^{148}	2^{147}

Fig. 3: Example parameters to instantiate the challenge space $\mathcal{C} := \{c \in S_\kappa : \sigma(c) = c \wedge \sqrt[2k]{\|\sigma_{-1}(c^k)\, c^k\|_1} \leqslant \eta\}$ for a modulus q such that its smallest prime divisor q_1 is greater than 8. In our examples we picked $k = 32$.

Setting the Standard Deviation. By definition of the challenge space $\mathcal{C}$ and Lemma 2.15, if we know that $\|\mathbf{r}\| \leqslant \alpha$, then we can set the standard deviation $\mathfrak{s} := \gamma\eta\alpha$ where $\gamma > 0$ defines the repetition rate M. On the other hand, if $\|\mathbf{r}\|_\infty \leqslant \nu$, e.g. because $\mathbf{r} \leftarrow S_\nu^\ell$, then we can set $\mathfrak{s} := \gamma\nu\eta\sqrt{\ell n}$.

3 The ABDLOP Commitment Scheme and Proofs of Linear Relations

In this section we formally present the ABDLOP commitment scheme together with ZKPoK of the committed messages. In the same protocol, we also include a proof of knowledge that the committed messages satisfy some arbitrary linear relations over $\mathcal{R}_q$ (Figure 4). We then show how one can use this commitment scheme and proof of knowledge to prove knowledge of linear relations over $\mathbb{Z}_q$ (Figure 5). This latter proof is best modeled as a commit-and-prove protocol because it will be creating some intermediate commitments under the same randomness, which cannot be simulated. In particular, what we prove is that the view, for all possible committed messages, is computationally indistinguishable from commitments to 0.

3.1 The ABDLOP Commitment Scheme

Figure 4 presents the ABDLOP commitment scheme, which commits to messages $\mathbf{s}_1$ and $\mathbf{m}$, using randomness $\mathbf{s}_2$, and then proves knowledge of these messages and that they satisfy the relation $\mathbf{R}_1\mathbf{s}_1 + \mathbf{R}_m\mathbf{m} = \mathbf{u}$. The challenge space $\mathcal{C}$ is as in (21). The standard deviations $\mathfrak{s}_1$ and $\mathfrak{s}_2$ are set as in Section 2.6 so as to provide a balance between the running time of the algorithm (the lower the values, the higher the probability that the protocol will need to be repeated) and the security of the commitment scheme based on the hardness of the MSIS problem (the higher the values, the easier the problem becomes). Because the most common way in which our commitment scheme will be used involves committing to some values, proving that they satisfy some relations, and then never using the commitment again, we use a more efficient rejection sampling (Rej_2 in Figure 1) from [LNS21a], which ends up leaking one bit of the secret, on the *randomness* part of the commitment (i.e. $\mathbf{s}_2$). If one will not be throwing out this commitment, then one should use Rej_1 for everything.

The hiding property of the commitment scheme follows from the MLWE problem when $\mathbf{s}_2$ is chosen from some distribution such that $\left(\begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix}, \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix}\cdot \mathbf{s}_2\right)$ is indistinguishable from uniform. The zero-knowledge property of the protocol follows from the standard argument from [Lyu12, LNS21a] showing that $\mathbf{z}_1, \mathbf{z}_2$ are distributed according to $D_{\mathfrak{s}_1}^{m_1}$ and $D_{\mathfrak{s}_2}^{m_2}$ (possibly with 1 bit of leakage for the latter) independent of $\mathbf{s}_1$ and $\mathbf{s}_2$. The correctness of the protocol then follows due to the fact that $m_i d$-dimensional integer vectors sampled from a discrete Gaussian with standard deviation $\mathfrak{s}_i$ has norm at most $\mathfrak{s}_i\sqrt{2m_i d}$ with overwhelming probability [Ban93].

The commitment opening needs to be defined to be whatever one can extract from the protocol. Since the protocol is an approximate proof of knowledge, it does not prove knowledge of $\mathbf{s}_1, \mathbf{s}_2$ satisfying $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A$, but instead an approximate proof as in (2). Lemma 3.1 states that under the assumption that the Module-SIS problem is hard, the extracted values $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2)$ are unique and they satisfy the desired linear equation $\mathbf{R}_1\bar{\boldsymbol{s}}_1 + \mathbf{R}_m(\mathbf{t}_B - \mathbf{B}\bar{\boldsymbol{s}}_2) = \mathbf{u}$, where $\mathbf{m}$ is implicitly defined as $\mathbf{t}_B - \mathbf{B}\bar{\boldsymbol{s}}_2$. The last statement proved in the Lemma shows, as in [ALS20], that not only are the extracted commitments $\mathbf{s}_i$, unique but also $\mathbf{z}_i - c\bar{\mathbf{s}}_i$ is uniquely determined by the first two moves of the protocol. This is crucial to efficiently proving knowledge of polynomial products later in the paper.

As far as the communication complexity of the protocol, it is important to note that in the real protocol, one would not actually send $\mathbf{w}$ and $\mathbf{v}$, but instead send their hash. Then one would verify the hash of the equalities. Therefore proving linear relations over $\mathcal{R}_q$ is not any more costly, communication-wise, than just proving knowledge of the committed values. We don't write the hashes in our protocols because when they eventually get converted to non-interactive ones using the Fiat-Shamir transform, the hashes will naturally enter the picture.

We will refer to the protocol in Figure 4 as $\Pi_{\mathsf{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \ldots, f_N))$, where the f_i are linear functions mapping $(\mathbf{s}_1, \mathbf{m})$ to $\mathcal{R}_q$ such that $f_i(\mathbf{s}_1, \mathbf{m}) = 0$, represented by the rows of $\mathbf{R}_1, \mathbf{R}_m$, and $\mathbf{u}$.

Lemma 3.1. *The protocol in Figure 4 is a proof of knowledge of* $(\bar{\boldsymbol{s}}_1, \bar{\boldsymbol{s}}_2, \bar{c}) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \bar{\mathcal{C}}$ *satisfying*

1. $\mathbf{A}_1\bar{\boldsymbol{s}}_1 + \mathbf{A}_2\bar{\boldsymbol{s}}_2 = \mathbf{t}_A$
2. $\|\bar{\mathbf{s}}_i\bar{c}\| \leqslant 2\mathfrak{s}_i\sqrt{2m_i d}$ *for* $i = 1, 2$
3. $\mathbf{R}_1\bar{\boldsymbol{s}}_1 + \mathbf{R}_m(\mathbf{t}_B - \mathbf{B}\bar{\boldsymbol{s}}_2) = \mathbf{u}$

Furthermore, under the assumption that $\mathsf{MSIS}_{n,m_1+m_2,B}$ *is hard for* $B = 8\eta\sqrt{(\mathfrak{s}_1\sqrt{2m_1 d})^2 + (\mathfrak{s}_2\sqrt{2m_2 d})^2}$,

4. *This* $(\bar{\boldsymbol{s}}_1, \bar{\boldsymbol{s}}_2)$ *is unique*
5. *For any two valid transcripts* $(\mathbf{w}, \mathbf{v}, c, \mathbf{z}_1, \mathbf{z}_2)$ *and* $(\mathbf{w}, \mathbf{v}, c', \mathbf{z}_1', \mathbf{z}_2')$, *it holds that* $\mathbf{z}_i - c\bar{\boldsymbol{s}}_i = \mathbf{z}_i' - c'\bar{\boldsymbol{s}}_i$.

Proof. Let $(\mathbf{w}, \mathbf{v}, c, \mathbf{z}_1, \mathbf{z}_2)$ and $(\mathbf{w}, \mathbf{v}, c', \mathbf{z}_1', \mathbf{z}_2')$ be two accepting transcripts which are obtained via rewinding the prover who sends $\mathbf{w}, \mathbf{v}$ in the first step. Because the transcripts are accepting, they satisfy the second verification equation, and by subtracting the two equalities, we obtain

$$\mathbf{A}_1\bar{\boldsymbol{z}}_1 + \mathbf{A}_2\bar{\boldsymbol{z}}_2 - \bar{c}\mathbf{t}_A = 0, \tag{23}$$

Private information: $(\mathbf{s}_1, \mathbf{m}, \mathbf{s}_2) \in \mathcal{R}_q^{m_1+m_2+\ell}$ so that $\|\mathbf{s}_1\| \leqslant \alpha$ and $\|\mathbf{s}_2\|_\infty \leqslant \nu$
Public information: $\mathbf{A}_1 \in \mathcal{R}_q^{n\times m_1}, \mathbf{A}_2 \in \mathcal{R}_q^{n\times m_2}, \mathbf{B} \in \mathcal{R}_q^{\ell\times m_2}$, $\mathbf{R}_1 \in \mathcal{R}_q^{N\times m_1}$, $\mathbf{R}_m \in \mathcal{R}_q^{N\times \ell}$,
$\begin{bmatrix}\mathbf{t}_A\\ \mathbf{t}_B\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{0}\end{bmatrix}\cdot\mathbf{s}_1 + \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix}\cdot\mathbf{s}_2 + \begin{bmatrix}\mathbf{0}\\ \mathbf{m}\end{bmatrix}, \mathbf{u} = \mathbf{R}_1\mathbf{s}_1 + \mathbf{R}_m\mathbf{m}$

Prover | Verifier

$\mathbf{y}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1}$
$\mathbf{y}_2 \leftarrow D_{\mathfrak{s}_2}^{m_2}$
$\mathbf{w} := \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2$
$\mathbf{v} := \mathbf{R}_1\mathbf{y}_1 - \mathbf{R}_m\mathbf{B}\mathbf{y}_2$

$\xrightarrow{\mathbf{w}, \mathbf{v}}$

$c \leftarrow \mathcal{C}$

$\xleftarrow{c}$

$\mathbf{z}_1 := c\mathbf{s}_1 + \mathbf{y}_1$
$\mathbf{z}_2 := c\mathbf{s}_2 + \mathbf{y}_2$
for $i = 1, 2$:
 if $\mathsf{Rej}_i(\mathbf{z}_i, c\mathbf{s}_i, \mathfrak{s}_i) = 1$
  then $\mathbf{z}_1, \mathbf{z}_2 := \perp$

$\xrightarrow{\mathbf{z}_1, \mathbf{z}_2}$

Accept iff:

1. $\|\mathbf{z}_1\| \leqslant \mathfrak{s}_1\sqrt{2m_1d}, \|\mathbf{z}_2\| \leqslant \mathfrak{s}_2\sqrt{2m_2d}$
2. $\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 - c\mathbf{t}_A = \mathbf{w}$
3. $\mathbf{R}_1\mathbf{z}_1 + \mathbf{R}_m(c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2) - c\mathbf{u} = \mathbf{v}$

Fig. 4: Proof of knowledge $\Pi^{(1)}_{\mathsf{many}}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \ldots, f_N))$ of $(\mathbf{s}_1, \mathbf{s}_2, \bar{c}) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \bar{\mathcal{C}}$ satisfying (i) $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A$, $\mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t}_B$ (ii) $\|\mathbf{s}_i\bar{c}\| \leqslant 2\mathfrak{s}_i\sqrt{2m_id}$ for $i = 1, 2$ and (iii) $f_j(\mathbf{s}_1, \mathbf{m}) = 0$ for $j \in [N]$ where each $f_1, \ldots, f_N : \mathcal{R}_q^{m_1+\ell} \to \mathcal{R}_q$ is a linear function. The linear functions f_j are represented by the corresponding rows of matrices $\mathbf{u}, \mathbf{R}_1, \mathbf{R}_m$ and prove $\mathbf{u} = \mathbf{R}_1\mathbf{s}_1 + \mathbf{R}_m\mathbf{m}$ where $\mathbf{R}_1^{N\times m_1}, \mathbf{R}_m^{N\times\ell}, \mathbf{u} \in \mathcal{R}_q^N$ are public.

where $\bar{\boldsymbol{z}}_i = \mathbf{z}_i - \mathbf{z}'_i$ and $\bar{c} = c - c'$. Dividing the above equation by $\bar{c}$, we obtain Lemma statement 1 where $\bar{\boldsymbol{s}}_i = \bar{\boldsymbol{z}}_i/\bar{c}$. Because the first verification checks that $\|\mathbf{z}_i\| \leqslant \mathfrak{s}_i\sqrt{2m_id}$, we know that $\|\bar{\boldsymbol{z}}_i\| \leqslant \mathfrak{s}_i\sqrt{2m_id}$, and so Lemma statement 2 is satisfied. By subtracting the two equalities satisfying the third verification equation, we obtain

$$\mathbf{R}_1\bar{\boldsymbol{z}}_1 + \mathbf{R}_m(\bar{c}\mathbf{t}_B - \mathbf{B}\bar{\boldsymbol{z}}_2) - \bar{c}\mathbf{u} = 0. \tag{24}$$

Dividing by $\bar{c}$ and plugging in $\bar{\boldsymbol{s}}_i = \bar{\boldsymbol{z}}_i/\bar{c}$, we get Lemma statement 3.

Now suppose that the extractor extracts another triplet $(\bar{\boldsymbol{s}}'_1, \bar{\boldsymbol{s}}'_2, \bar{c}')$ with $(\bar{\boldsymbol{s}}_1, \bar{\boldsymbol{s}}_2) \neq (\bar{\boldsymbol{s}}'_1, \bar{\boldsymbol{s}}'_2)$, which, as we already proved, must satisfy the first two statements of the lemma. Then we have

$$\mathbf{A}_1\bar{\boldsymbol{s}}_1 + \mathbf{A}_2\bar{\boldsymbol{s}}_2 = \mathbf{A}_1\bar{\boldsymbol{s}}'_1 + \mathbf{A}_2\bar{\boldsymbol{s}}'_2, \tag{25}$$

and multiplying the above by $\bar{c}\bar{c}'$ yields

$$\mathbf{A}_1(\bar{\boldsymbol{s}}_1 - \bar{\boldsymbol{s}}'_1)\bar{c}\bar{c}' + \mathbf{A}_2(\bar{\boldsymbol{s}}_2 - \bar{\boldsymbol{s}}'_2)\bar{c}\bar{c}' = \mathbf{0}. \tag{26}$$

‘By Lemma condition 2, we know that $\bar{\boldsymbol{s}}_i\bar{c}, \bar{\boldsymbol{s}}'_i\bar{c}' \leqslant 2\mathfrak{s}_i\sqrt{2m_id}$, and so the above can be rewritten as

$$\mathbf{A}_1(\bar{\boldsymbol{z}}_1\bar{c}' - \bar{\boldsymbol{z}}'_1\bar{c}) + \mathbf{A}_2(\bar{\boldsymbol{z}}_2\bar{c}' - \bar{\boldsymbol{z}}'_2\bar{c}) = \mathbf{0}, \tag{27}$$

where $\|\bar{\boldsymbol{z}}_i\|, \|\bar{\boldsymbol{z}}'_i\| \leqslant 2\mathfrak{s}_i\sqrt{2m_id}$. By Lemma 2.15, multiplication by $c \in \mathcal{C}$ increases the ℓ_2 norm by a factor of η, where η is defined in Figure 3. Thus multiplication by $\bar{c} \in \bar{\mathcal{C}}$ increases the norm by a factor of 2η, and thus $\|\bar{\boldsymbol{z}}_i\bar{c}' - \bar{\boldsymbol{z}}'_i\bar{c}\| \leqslant 8\eta\mathfrak{s}_i\sqrt{2m_id}$. If $\mathsf{MSIS}_{n,m_1+m_2,B}$ is hard for $B = 8\eta\sqrt{(\mathfrak{s}_1\sqrt{2m_1d})^2 + (\mathfrak{s}_2\sqrt{2m_2d})^2}$, it implies that $\bar{\boldsymbol{z}}_i\bar{c}' - \bar{\boldsymbol{z}}'_i\bar{c} = \mathbf{0}$, which means that $\bar{\boldsymbol{s}}_i = \bar{\boldsymbol{z}}_i/\bar{c} = \bar{\boldsymbol{z}}'_i/\bar{c}' = \bar{\boldsymbol{s}}'_i$, and this proves Lemma statement 4.

To prove Lemma statement 5, suppose that $\mathbf{z}_i - c\bar{\boldsymbol{s}}_i = \mathbf{z}'_i - c'\bar{\boldsymbol{s}}_i + \mathbf{r}$ for some $\mathbf{r}$. Then, we can rewrite this as $\bar{\boldsymbol{z}}_i/\bar{c} = \bar{\boldsymbol{s}}_i + \mathbf{r}/\bar{c}$. Since we already proved that $\bar{\boldsymbol{z}}_i/\bar{c} = \bar{\boldsymbol{s}}_i$, and the $\bar{\boldsymbol{s}}_i$ are unique, it means that $\mathbf{r} = \mathbf{0}$. □

Private information: $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$ so that $\|\mathbf{s}_1\| \leqslant \alpha$, $\mathbf{s}_2 \leftarrow \chi^{m_2}$
Public information: $\mathbf{A}_1 \in \mathcal{R}_q^{n\times m_1}, \mathbf{A}_2 \in \mathcal{R}_q^{n\times m_2}, \mathbf{B} \in \mathcal{R}_q^{\ell\times m_2}, \mathbf{B}_g \in \mathcal{R}_q^{\lambda\times m_2}$,
$\begin{bmatrix}\mathbf{t}_A\\ \mathbf{t}_B\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{0}\end{bmatrix}\cdot \mathbf{s}_1 + \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix}\cdot \mathbf{s}_2 + \begin{bmatrix}\mathbf{0}\\ \mathbf{m}\end{bmatrix}$, linear functions $F_1, \ldots, F_M : \mathcal{R}_q^{m_1+\ell} \to \mathcal{R}_q$

Prover		Verifier
$\mathbf{g} := (g_1, \ldots, g_\lambda) \leftarrow \{x \in \mathcal{R}_q : x_0 = 0\}^\lambda$ $\mathbf{t}_g := \mathbf{B}_g\mathbf{s}_2 + \mathbf{g}$		
	$\xrightarrow{\mathbf{t}_g}$	
		$(\gamma_{j,u}) \leftarrow \mathbb{Z}_q^{\lambda\times M}$
	$\xleftarrow{(\gamma_{j,u})_{j\in[\lambda],u\in[M]}}$	
for $j \in [\lambda]$: $h_j := g_j + \sum_{u=1}^{M} \gamma_{j,u} F_u(\mathbf{s}_1, \mathbf{m})$ define function f_j as in (28)		
	$\xrightarrow{h_1, \ldots, h_\lambda}$	
run $\Pi := \Pi^{(1)}_{\mathsf{many}}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}\|\|\mathbf{g}), (f_1, \ldots, f_\lambda))$		Accept iff : Π verifies and $\forall j \in [\lambda], \tilde{h}_j = 0$

Fig. 5: Commit-and-prove protocol $\Pi^{(1)}_{\mathsf{eval}}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (F_1, F_2, \ldots, F_M))$ for messages $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$, randomness $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$ and $\bar{c} \in \bar{\mathcal{C}}$ which satisfy: $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A$, $\mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t}_B$ (ii) $\|\mathbf{s}_i\bar{c}\| \leqslant 2\mathfrak{s}_i\sqrt{2m_i d}$ for $i = 1, 2$ ($\mathfrak{s}_i$ are from Figure 4) and (iii) linear functions $F_1, \ldots, F_M : \mathcal{R}_q^{m_1+\ell} \to \mathcal{R}_q$ for which all the evaluations $\widetilde{F}_u(\mathbf{s}_1, \mathbf{m}) = 0$. Here, we assume that the commitment $(\mathbf{t}_A, \mathbf{t}_B)$ was generated honestly and already sent by the prover. In particular, $\mathbf{s}_2 \leftarrow \chi^{m_2}$.

3.2 Linear Proofs over $\mathbb{Z}_q$

In this section we show how to transform the protocol from Figure 4 which proves that committed values satisfy a linear relation over $\mathcal{R}_q$ into one that proves knowledge of the *constant coefficient* of a linear relation over $\mathcal{R}_q$ (Figure 5). As shown in the introduction and Section 7, the inner product between two integer vectors appears in the constant coefficient of the polynomial product of two polynomials derived from these vectors. Thus proving knowledge that the constant coefficient of some linear function over $\mathcal{R}_q$ is 0 is equivalent to proving knowledge that the output of a linear function over $\mathbb{Z}_q$ is 0.

While it may see like proving knowledge of just the constant coefficient of a linear function over $\mathcal{R}_q$ should not be much different than proving knowledge of the entire linear function as in Figure 4, the protocols do have some important differences. The main difference is that due to the need to mask all but the constant coefficient, we will need to create additional commitments during the proof. The most efficient way to do this is to append these commitments to the BDLOP part of the commitment scheme using the public randomness $\mathbf{B}_g$ in Figure 5. The implication of needing to append committed values is that one can no longer reuse the commitment $\mathbf{t}_A, \mathbf{t}_B$ since every run of the protocol essentially reveals more information about the randomness $\mathbf{s}_2$. Thus, instead of proving that the protocol is zero-knowledge, we show that the protocol is of a "commit-and-prove" type, where the security requirement is that the view of the commitment and the protocol output is computationally indistinguishable for all committed messages. All the other protocols in this paper also have this characteristic. This does not pose any problems for applications because the way we use a commitment scheme is in an auxiliary way to aid in proving that the value we care about satisfies some relations. Thus the commitment never needs to be reused.

The protocol begins by picking masking values $g_i \in \mathcal{R}_q$, which are uniformly random everywhere except in the constant coefficient, in which they are 0. These values are then appended to the commitment of $\mathbf{m}$ as $\mathbf{t}_g = \mathbf{B}_g \mathbf{s}_2 + \mathbf{g}$ and then sent to the verifier. The verifier picks λ random challenges for each of the M linear functions and the prover computes $h_j = g_j + \sum_{u=1}^{M} \gamma_{j,u} F_u\left(\mathbf{s}_1, \mathbf{m}\right)$ for each of the λ different j. Notice that the preceding are now linear functions

$$f_j(\mathbf{s}_1, \mathbf{m}||\mathbf{g}) := g_j + \sum_{u=1}^{M} \gamma_{j,u} F_u\left(\mathbf{s}_1, \mathbf{m}\right) - h_j \tag{28}$$

over committed inputs $\mathbf{s}_1, \mathbf{m}||\mathbf{g}$. The prover completes the proof by sending the h_j, which completes the description of the functions, and begins the protocol in Figure 4 for proving that $f_j(\mathbf{s}_1, \mathbf{m}||\mathbf{g}) = 0$. The verifier accepts if the constant coefficient of h_j is 0 and the proof from Figure 4 is valid.

We now sketch the security and soundness properties of the protocol. This protocol is a warm-up for the full one in Figure 8 which proves knowledge of the constant coefficient of quadratic (rather than linear) functions over $\mathcal{R}_q$, and so we do not give a complete proof for it. To see that the view of the protocol is computationally indistinguishable for all messages $\mathbf{s}_1, \mathbf{m}$, we first observe that the full commitment that includes $\mathbf{g}$ is indistinguishable from uniform based on (Extended)-Module-LWE as long as $\left(\begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\\ \mathbf{B}_g\end{bmatrix}, \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\\ \mathbf{B}_g\end{bmatrix} \cdot \mathbf{s}_2\right)$ is indistinguishable from uniform when $\mathbf{s}_2 \leftarrow \chi^{m_2}$. To simulate the protocol, the simulator can simply pick $\mathbf{t}_g$ uniformly at random and also choose $h_1, \ldots, h_\lambda$ at random (but having the first coefficient being 0). He can then simulate the protocol from figure 4 on the commitment $(\mathbf{t}_A, \mathbf{t}_B, \mathbf{t}_g)$ and functions f_j. Thus the distribution is computationally indistinguishable from the correct one and is independent of the messages $\mathbf{s}_1, \mathbf{m}$.

To show that this protocol indeed proves that $\widetilde{F_u}(\mathbf{s}_1, \mathbf{m}) = 0$, notice that the probability over the challenges $\gamma_{j,u}$ that the equation $h_j = g_j + \sum_{u=1}^{M} \gamma_{j,u} F_u\left(\mathbf{s}_1, \mathbf{m}\right)$ is satisfied when $\widetilde{h_j} = 0$ and yet some $\widetilde{F_u}\left(\mathbf{s}_1, \mathbf{m}\right) \neq 0$ is at most $1/q_1$, where q_1 is the smallest prime factor of q. The above holds because the values $\mathbf{s}_1, \mathbf{m}$, and $\mathbf{g}$ were committed to prior to the verifier sending the challenges. The latter, as well as the fact that the linear equations f_j are satisfied, is proved by the protocol $\Pi^{(1)}_{\mathsf{many}}\left((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}||\mathbf{g}), (f_1, \ldots, f_\lambda)\right)$. The soundness error of the protocol is therefore $q_1^{-\lambda}$.

4 Proofs of Quadratic Relations

In this section we show how to prove various quadratic equations between committed messages using the ABDLOP commitment. More concretely, suppose we have message vectors $\mathbf{s}_1 \in \mathcal{R}_q^{m_1}$ and $\mathbf{m} \in \mathcal{R}_q^{\ell}$ such that $\|\mathbf{s}_1\| \leqslant \alpha$. Let $\sigma \in \mathsf{Aut}(\mathcal{R}_q)$ be a public automorphism over $\mathcal{R}$ of degree k and for presentation purposes define:

$$(\sigma^i(\mathbf{x}))_{i \in [k]} := (\mathbf{x}, \sigma(\mathbf{x}), \ldots, \sigma^{k-1}(\mathbf{x})) \in \mathcal{R}_q^{ka}$$

for arbitrary vector $\mathbf{x} \in \mathcal{R}_q^a$. Then, we consider the following statements:

- *Single quadratic equation with automorphisms.* For a public $k(m_1 + \ell)$-variate quadratic function f over $\mathcal{R}_q$,
$$f\left((\sigma^i(\mathbf{s}_1))_{i \in [k]}, (\sigma^i(\mathbf{m}))_{i \in [k]}\right) = 0.$$
- *Many quadratic equations with automorphisms.* For N public $k(m_1 + \ell)$-variate quadratic functions $f_1, \ldots, f_N$ over $\mathcal{R}_q$,
$$f_j\left((\sigma^i(\mathbf{s}_1))_{i \in [k]}, (\sigma^i(\mathbf{m}))_{i \in [k]}\right) = 0 \text{ for } j \in [N].$$
- *Many quadratic equations with automorphisms and a proof that polynomial evaluations have no constant coefficients.* For $N + M$ public $k(m_1 + \ell)$-variate quadratic functions $f_1, \ldots, f_N$ and $F_1, \ldots, F_M$ over $\mathcal{R}_q$, the following hold:

- $f_j\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) = 0$ for $j \in [N]$,
- let $x_j := F_j\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) \in \mathcal{R}_q$ for $j \in [M]$. Then $\tilde{x}_1 = \ldots = \tilde{x}_M = 0$.

Remark 4.1. Similarly as for [ALS20], our techniques can be easily generalized to prove higher degree relations. Concretely, if we want to prove degree k equations, we end up committing to $k-1$ additional garbage terms. Throughout this paper (apart from Section 6.5), however, we will only consider quadratic relations.

4.1 Single Quadratic Equation with Automorphisms

Let $(\mathbf{t}_A, \mathbf{t}_B)$ be the commitment to the message pair $(\mathbf{s}_1, \mathbf{m})$ under randomness $\mathbf{s}_2$, i.e.

$$\begin{bmatrix}\mathbf{t}_A\\ \mathbf{t}_B\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{0}\end{bmatrix}\cdot\mathbf{s}_1 + \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix}\cdot\mathbf{s}_2 + \begin{bmatrix}\mathbf{0}\\ \mathbf{m}\end{bmatrix}.$$

Suppose the prover wants to prove knowledge of the message

$$\mathbf{s} = \begin{bmatrix}(\sigma^i(\mathbf{s}_1))_{i\in[k]}\\ (\sigma^i(\mathbf{m}))_{i\in[k]}\end{bmatrix} \in \mathcal{R}_q^{k(m_1+\ell)}$$

such that $f(\mathbf{s}) = 0$ where f is a $k(m_1+\ell)$-variate quadratic function over $\mathcal{R}_q$. Note that each function f can be written explicitly as:

$$f(\mathbf{s}) = \mathbf{s}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\mathbf{s} + r_0$$

where $r_0 \in \mathcal{R}_q, \mathbf{r}_1 \in \mathcal{R}_q^{k(m_1+\ell)}$ and $\mathbf{R}_2 \in \mathcal{R}_q^{k(m_1+\ell)\times k(m_1+\ell)}$.

In order to prove this relation, let us consider the protocol for proving linear equations over $\mathcal{R}_q$ in Fig. 4. In the last round, the honest prover sends the *masked openings* $\mathbf{z}_i = c\mathbf{s}_i + \mathbf{y}_i$ of $\mathbf{s}_i$ for $i = 1, 2$ where the challenge space $\mathcal{C}$ is defined as in (21) with the σ automorphism. Even though this is not the case for $\mathbf{m}$, we can define the masked opening of $\mathbf{m}$ as

$$\mathbf{z}_m := c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2 = c\mathbf{m} - \mathbf{B}\mathbf{y}_2.$$

By construction, $\mathbf{z}_m$ can be computed by the verifier.

Define the following vectors $\mathbf{y}$ and $\mathbf{z}$:

$$\mathbf{y} := \begin{bmatrix}(\sigma^i(\mathbf{y}_1))_{i\in[k]}\\ -(\sigma^i(\mathbf{B}\mathbf{y}_2))_{i\in[k]}\end{bmatrix} \in \mathcal{R}_q^{k(m_1+\ell)} \tag{29}$$

and

$$\mathbf{z} := \begin{bmatrix}(\sigma^i(\mathbf{z}_1))_{i\in[k]}\\ (\sigma^i(\mathbf{z}_m))_{i\in[k]}\end{bmatrix} = c\begin{bmatrix}(\sigma^i(\mathbf{s}_1))_{i\in[k]}\\ (\sigma^i(\mathbf{m}))_{i\in[k]}\end{bmatrix} + \begin{bmatrix}(\sigma^i(\mathbf{y}_1))_{i\in[k]}\\ -(\sigma^i(\mathbf{B}\mathbf{y}_2))_{i\in[k]}\end{bmatrix} = c\mathbf{s} + \mathbf{y}. \tag{30}$$

Here we used the fact that for $c \in \mathcal{C}, \sigma(c) = c$. Then, we have

$$\mathbf{z}^T\mathbf{R}_2\mathbf{z} + c\mathbf{r}_1^T\mathbf{z} + c^2 r_0 = c^2\left(\mathbf{s}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\mathbf{s} + r_0\right) + cg_1 + g_0 \tag{31}$$

where polynomials g_1 and g_0 are defined as:

$$g_1 = \mathbf{s}^T\mathbf{R}_2\mathbf{y} + \mathbf{y}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\mathbf{y}, \quad g_0 = \mathbf{y}^T\mathbf{R}_2\mathbf{y}.$$

Hence, we want to prove that the quadratic term in the expression $\mathbf{z}^T\mathbf{R}_2\mathbf{z} + c\mathbf{r}_1^T\mathbf{z} + c^2 r_0$ vanishes. This is done by first sending a commitment t to the polynomial g_1, i.e. $t = \mathbf{b}^T\mathbf{s}_2 + g_1$ as well as $v := g_0 + \mathbf{b}^T\mathbf{y}_2$ in the clear. Then, given t and the masked opening $\mathbf{z}_2$ of $\mathbf{s}_2$, the verifier can compute $f = ct - \mathbf{b}^T\mathbf{z}_2 = cg_1 - \mathbf{b}^T\mathbf{y}_2$. Finally, it checks whether

$$\mathbf{z}^T\mathbf{R}_2\mathbf{z} + c\mathbf{r}_1^T\mathbf{z} + c^2 r_0 - f \stackrel{?}{=} v$$

which is a simple transformation of (31) when $\mathbf{s}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\mathbf{s} + r_0 = 0$.

Private information: $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$ so that $\|\mathbf{s}_1\| \leqslant \alpha$, $\mathbf{s}_2 \leftarrow \chi^{m_2}$

Public information: $\mathbf{A}_1 \in \mathcal{R}_q^{n \times m_1}, \mathbf{A}_2 \in \mathcal{R}_q^{n \times m_2}, \mathbf{B} \in \mathcal{R}_q^{\ell \times m_2}, \mathbf{b} \in \mathcal{R}_q^{m_2}$

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \cdot \mathbf{s}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} \cdot \mathbf{s}_2 + \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix},$$

$r_0 \in \mathcal{R}_q, \mathbf{r}_1 \in \mathcal{R}_q^{k(m_1+\ell)}, \mathbf{R}_2 \in \mathcal{R}_q^{k(m_1+\ell) \times k(m_1+\ell)}, \sigma \in \mathsf{Aut}(\mathcal{R}_q)$

$\mathbf{s}^T \mathbf{R}_2 \mathbf{s} + \mathbf{r}_1^T \mathbf{s} + r_0 = 0$

Prover		Verifier
$\mathbf{s} := \begin{bmatrix} (\sigma^i(\mathbf{s}_1))_{i\in[k]} \\ (\sigma^i(\mathbf{m}))_{i\in[k]} \end{bmatrix}$ $\mathbf{y}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1}$ $\mathbf{y}_2 \leftarrow D_{\mathfrak{s}_2}^{m_2}$ $\mathbf{w} := \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2$ $\mathbf{y} := \begin{bmatrix} (\sigma^i(\mathbf{y}_1))_{i\in[k]} \\ -(\sigma^i(\mathbf{B}\mathbf{y}_2))_{i\in[k]} \end{bmatrix}$ $g_1 := \mathbf{s}^T\mathbf{R}_2\mathbf{y} + \mathbf{y}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\mathbf{y}$ $t := \mathbf{b}^T\mathbf{s}_2 + g_1$ $v := \mathbf{y}^T\mathbf{R}_2\mathbf{y} + \mathbf{b}^T\mathbf{y}_2$		
	$\xrightarrow{\mathbf{w}, t, v}$	
		$c \leftarrow \mathcal{C}$
	$\xleftarrow{c}$	
$\mathbf{z}_1 := c\mathbf{s}_1 + \mathbf{y}_1$ $\mathbf{z}_2 := c\mathbf{s}_2 + \mathbf{y}_2$ for $i = 1, 2$: if $\mathsf{Rej}_i(\mathbf{z}_i, c\mathbf{s}_i, \mathfrak{s}_i) = 1$ then $\mathbf{z}_1, \mathbf{z}_2 := \perp$		
	$\xrightarrow{\mathbf{z}_1, \mathbf{z}_2}$	
		$\mathbf{z} := \begin{bmatrix} (\sigma^i(\mathbf{z}_1))_{i\in[k]} \\ (\sigma^i(c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2))_{i\in[k]} \end{bmatrix}$ $f := ct - \mathbf{b}^T\mathbf{z}_2$ Accept iff $\|\mathbf{z}_1\| \leqslant \mathfrak{s}_1\sqrt{2m_1d}$ and $\|\mathbf{z}_2\| \leqslant \mathfrak{s}_2\sqrt{2m_2d}$ and $\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 = \mathbf{w} + c\mathbf{t}_A$ and $\mathbf{z}^T\mathbf{R}_2\mathbf{z} + c\mathbf{r}_1^T\mathbf{z} + c^2r_0 - f = v$

Fig. 6: Commit-and-prove protocol $\Pi^{(2)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), \sigma, f)$ for messages $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$, randomness $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$ and $\bar{c} \in \bar{\mathcal{C}}$ which satisfy: $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A$, $\mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t}_B$ (ii) $\|\mathbf{s}_i\bar{c}\| \leqslant 2\mathfrak{s}_i\sqrt{2m_id}$ for $i = 1, 2$ and (iii) $f\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) = 0$ where function $f : \mathcal{R}_q^{k(m_1+\ell)} \to \mathcal{R}_q$ is defined as $f(\mathbf{x}) := \mathbf{x}^T\mathbf{R}_2\mathbf{x} + \mathbf{r}_1^T\mathbf{x} + r_0$. Here, we assume that the commitment $(\mathbf{t}_A, \mathbf{t}_B)$ was generated honestly and already sent by the prover. In particular, $\mathbf{s}_2 \leftarrow \chi^{m_2}$.

We present the full protocol in Fig. 6 which follows the commit-and-prove paradigm [CLOS02, LNS21a]. Namely, we assume the prover has already sent the commitment $(\mathbf{t}_A, \mathbf{t}_B)$ to the verifier using fresh randomness $\mathbf{s}_2 \leftarrow \chi^{m_2}$. Prover starts by sampling masking vectors $\mathbf{y}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1}, \mathbf{y}_2 \leftarrow D_{\mathfrak{s}}^{m_2}$ and computing $\mathbf{w} = \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2$. Then, it calculates $g_1 = \mathbf{s}^T\mathbf{R}_2\mathbf{y} + \mathbf{y}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\mathbf{y}$, where $\mathbf{y}$ is defined in (29), and the commitment $t = \mathbf{b}^T\mathbf{s}_2 + g_1$ to g_1. Finally, the prover sets $v = \mathbf{y}^T\mathbf{R}_2\mathbf{y} + \mathbf{b}^T\mathbf{y}_2$ and sends $\mathbf{w}, t, v$ to the verifier.

Next, given a challenge $c \leftarrow C$, the prover computes $\mathbf{z}_i = c\mathbf{s}_i + \mathbf{y}_i$ for $i = 1, 2$ and applies rejection sampling. If it does not abort, the prover outputs $\mathbf{z}_1, \mathbf{z}_2$.

Eventually, the verifier checks whether $\mathbf{z}_1$ and $\mathbf{z}_2$ have small norms, $\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 = \mathbf{w} + c\mathbf{t}_A$ and $\mathbf{z}^T\mathbf{R}_2\mathbf{z} + c\mathbf{r}_1^T\mathbf{z} + c^2 r_0 - f = v$ where $\mathbf{z}$ is defined in (30) and f is defined as $f = ct - \mathbf{b}^T\mathbf{z}_2$.

Security Analysis. We summarise security properties of the protocol in Fig. 6 below.

Theorem 4.2. *Consider the protocol in Fig. 6 and let $\chi = S_\nu$. Suppose $\mathfrak{s}_1 = \gamma_1\alpha\eta$ and $\mathfrak{s}_2 = \gamma_2\nu\eta\sqrt{m_2 d}$ for some $\gamma_1, \gamma_2 > 0$ where η is chosen as in Section 2.7.*

For completeness, if $m_1, m_2 \geqslant 640/d$ then the honest prover $\mathcal{P}$ convinces the honest verifier $\mathcal{V}$ with probability

$$\approx \frac{1}{2\exp\left(\frac{14}{\gamma_1} + \frac{1}{2\gamma_1^2} + \frac{1}{2\gamma_2^2}\right)}.$$

For commit-and-prove simulatability, there exists a simulator $\mathcal{S}$ that, without access to private information $\mathbf{s}_1, \mathbf{m}$, outputs a simulation of a commitment $(\mathbf{t}_A, \mathbf{t}_B)$ along with a non-aborting transcript of the protocol between prover $\mathcal{P}$ and verifier $\mathcal{V}$ such that for every algorithm $\mathcal{A}$ that has advantage ε in distinguishing the simulated commitment and transcript from the real commitment and transcript, whenever the prover does not abort, there is an algorithm $\mathcal{A}'$ with the same running time that has advantage $\varepsilon/2 - 2^{-128}$ in distinguishing the Extended-$\mathsf{MLWE}_{n+\ell+1, m_2-n-\ell-1, \chi, \mathcal{C}, \mathfrak{s}_2}$.

For soundness, there is an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a probabilistic prover $\mathcal{P}^*$, which convinces $\mathcal{V}$ with probability $\varepsilon \geqslant 2/|\mathcal{C}|$, extractor $\mathcal{E}$ with probability at least $\varepsilon - 2/|\mathcal{C}|$ either outputs $(\bar{\mathbf{s}}_2, \bar{\mathbf{s}}_1, \bar{\mathbf{m}}) \in \mathcal{R}_q^{m_1+m_2+\ell}$ and $\bar{c} \in \mathcal{R}_q^\times$ such that

- $\begin{bmatrix}\mathbf{t}_A \\ \mathbf{t}_B\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1 \\ \mathbf{0}\end{bmatrix} \cdot \bar{\mathbf{s}}_1 + \begin{bmatrix}\mathbf{A}_2 \\ \mathbf{B}\end{bmatrix} \cdot \bar{\mathbf{s}}_2 + \begin{bmatrix}\mathbf{0} \\ \bar{\mathbf{m}}\end{bmatrix}$
- $\|\bar{c}\|_\infty \leqslant 2\kappa$
- $\|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ *and* $\|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$
- $f\left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) = 0$

or a $\mathsf{MSIS}_{n, m_1+m_2, B}$ solution for $\begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix}$ in expected time at most $3T$ where running $\mathcal{P}^*$ once is assumed to take at most T time and $B = 8\eta\sqrt{(\mathfrak{s}_1\sqrt{2m_1 d})^2 + (\mathfrak{s}_2\sqrt{2m_2 d})^2}$.

Proof. We first focus on completeness. To begin with, we bound the norm of $c\mathbf{s}_1$ and $c\mathbf{s}_2$. Note that by Lemma 2.15 and the definition of $\mathcal{C}$ in (21): $\|c\mathbf{s}_1\| \leqslant \eta\alpha$ and $\|c\mathbf{s}_2\| \leqslant \eta\nu\sqrt{m_2 d}$. Then, by Lemma 2.14, the probability that Rej_1 and Rej_2 do not abort is at least

$$\frac{1}{2 \cdot \exp\left(\frac{14}{\gamma_1} + \frac{1}{2\gamma_1^2}\right) \cdot \exp\left(\frac{1}{2\gamma_2^2}\right)}.$$

Furthermore, by Lemma 2.2 for $t = \sqrt{2}$ and our assumption that $m_1, m_2 \geqslant 640/d$, the probability that $\|\mathbf{z}_1\| \leqslant \mathfrak{s}_1\sqrt{2m_1 d}$ and $\|\mathbf{z}_2\| \leqslant \mathfrak{s}_2\sqrt{2m_2 d}$ is overwhelming. The other verification equations hold based on the discussion above.

Commit-and-prove simulatability. We can simulate the commitment and a non-aborting transcript between the honest prover and the honest verifier in the following way.

First, we define a hybrid simulator $\mathcal{S}_0$ which still knows secret information $\mathbf{s}_1, \mathbf{m}$. Given a challenge $c \leftarrow \mathcal{C}$, it honestly generates the commitment $(\mathbf{t}_A, \mathbf{t}_B, t)$ under randomness $\mathbf{s}_2 \leftarrow \chi^{m_2}$. Further, it samples fresh masked opening $\mathbf{z}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1 d}$ and $\mathbf{z}_2 \leftarrow D_{\mathfrak{s}_2}^{d}$ conditioned on $\langle \mathbf{s}_2, \mathbf{z}_2 \rangle \geqslant 0$. Finally, it sets $\mathbf{w} := \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 - c\mathbf{t}_A$ and $v := \mathbf{z}^T\mathbf{R}_2\mathbf{z} + c\mathbf{r}_1^T\mathbf{z} + c^2 r_0 - ct + \mathbf{b}^T\mathbf{z}_2$. Then, by Lemma 2.14, the distribution of the commitment and a transcript output by $\mathcal{S}_0$ is statistically close to the one in the actual non-aborting protocol.

Next, we define the simulator $\mathcal{S}_1$, which still knows secret information $\mathbf{s}_1, \mathbf{m}$, as follows. It runs identically as $\mathcal{S}_0$ but instead of generating the commitment $(\mathbf{t}_A, \mathbf{t}_B, t)$ honestly, it samples $\mathbf{u} \leftarrow \mathcal{R}_q^{n+\ell+1}$ and sets

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \\ t \end{bmatrix} = \mathbf{u} + \begin{bmatrix} \mathbf{A}_1\mathbf{s}_1 \\ \mathbf{m} \\ g_1 \end{bmatrix}. \tag{32}$$

We claim that if there is a PPT adversary $\mathcal{A}$ distinguishes between the outputs of $\mathcal{S}_0$ and $\mathcal{S}_1$ with probability ε, then there exists a PPT adversary $\mathcal{B}$ which solves the Extended-$\mathsf{MLWE}_{n+\ell+1, m_2-n-\ell-1, \chi, \mathcal{C}, \mathfrak{s}_2}$ with probability at least $\varepsilon/2$. Indeed, we can define $\mathcal{B}$ as follows. Given an Extended-MLWE tuple $(\mathbf{C}, \mathbf{u}, \mathbf{z}_2, b)$, where

$$\mathbf{C} := \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \\ \mathbf{b}^T \end{bmatrix},$$

$\mathcal{B}$ sets $(\mathbf{t}_A, \mathbf{t}_B, t)$ as in (32) and simulates the rest of the transcripts identically as $\mathcal{S}_0$ and $\mathcal{S}_1$. Then, it outputs the commitment and the transcript to $\mathcal{A}$. Let us assume that $b = 1$. Note that if $\mathbf{u} = \mathbf{C}\mathbf{s}_2$ then the output of $\mathcal{B}$ comes from the distribution of $\mathcal{S}_0$. Similarly, if $\mathbf{u}$ was uniformly random, then the output of $\mathcal{B}$ comes from the distribution of $\mathcal{S}_1$. Hence, conditioned on $b = 1$, $\mathcal{B}$ solves the Extended-MLWE problem with probability at least ε. Since the probability of $b = 1$ is at least $1/2$, the statement follows.

Finally, we can simply set $\mathcal{S}$ (which does not use any secret information) to proceed identically as $\mathcal{S}_1$ but instead of defining $(\mathbf{t}_A, \mathbf{t}_B, t)$ as in (32), it directly samples $(\mathbf{t}_A, \mathbf{t}_B, t) \leftarrow \mathcal{R}_q^{n+\ell+1}$. Then, the output distributions of $\mathcal{S}$ and $\mathcal{S}_1$ are identical. Hence, the statement holds by the hybrid argument.

Soundness. We apply the strategy by Attema et al. [ACK21]. Namely, let $H \in \{0,1\}^{R \times N}$ be a binary matrix where the R rows correspond to the prover's randomness and N columns correspond to verifier's randomness, i.e. different choices for the challenge c. For simplicity, we denote $H(r, c)$ to be the entry corresponding to randomness r and challenge $c \in \mathcal{C}$. Clearly, an extractor can check values of each entry in H in time at most T.

We define the following extractor $\mathcal{E}$:

1. $\mathcal{E}$ first samples fresh randomness r and challenge $c^{(0)} \leftarrow \mathcal{C}$. Then, it checks if $H(r, c^{(0)}) = 1$. If not, $\mathcal{E}$ aborts.
2. Otherwise, $\mathcal{E}$ samples along row r *without replacement* until it finds two $c^{(1)}, c^{(2)}$ such that $H(r, c^{(0)}) = H(r, c^{(1)}) = H(r, c^{(2)}) = 1$[12].

By [ACK21, Remark 2], the expected time of $\mathcal{E}$ is at most $3T$ and $\mathcal{E}$ extracts three valid transcripts

$$\mathsf{tr}^{(i)} = (\mathbf{w}, t, v, c^{(i)}, \mathbf{z}_1^{(i)}, \mathbf{z}_2^{(i)}) \text{ for } i = 0, 1, 2$$

with probability at least $\varepsilon - 2/|\mathcal{C}|$.

First we focus on $\mathsf{tr}^{(0)}$ and $\mathsf{tr}^{(1)}$. Define

$$\bar{c} := c^{(1)} - c^{(0)} \text{ and } \bar{\mathbf{s}}_i = \frac{\mathbf{z}_i^{(1)} - \mathbf{z}_i^{(0)}}{c^{(1)} - c^{(0)}} \text{ for } i = 1, 2.$$

[12] By construction, $c^{(0)}, c^{(1)}, c^{(2)}$ are pairwise distinct.

By construction, we $\|\bar{c}\|_\infty \leqslant 2\kappa$, $\|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ and $\|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$. Moreover, we have $\mathbf{A}_1\bar{\mathbf{s}}_1 + \mathbf{A}_2\bar{\mathbf{s}}_2 = \mathbf{t}_A$. Further, we define the extracted message vector $\bar{\mathbf{m}} := \mathbf{t}_B - \mathbf{B}\bar{\mathbf{s}}_2$ and $\bar{g}_1 := t - \mathbf{b}^T\bar{\mathbf{s}}_2$. Then, we have

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \\ t \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \\ 0 \end{bmatrix} \cdot \bar{\mathbf{s}}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \\ \mathbf{b}^T \end{bmatrix} \cdot \bar{\mathbf{s}}_2 + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{m}} \\ \bar{g}_1 \end{bmatrix}.$$

Next, let $\bar{\mathbf{y}}_i := \mathbf{z}_i^{(1)} - c^{(1)}\bar{\mathbf{s}}_i = \mathbf{z}_i^{(0)} - c^{(0)}\bar{\mathbf{s}}_i$ for $i = 1, 2$. Moreover, consider the third transcript $\mathsf{tr}^{(2)}$ and define $\mathbf{y}_i^{(2)} := \mathbf{z}_i^{(2)} - c^{(2)}\bar{\mathbf{s}}_i$ for $i = 1, 2$. Using the identical argument as in the proof of Lemma 3.1, either $(\bar{\mathbf{y}}_1, \bar{\mathbf{y}}_2) = (\mathbf{y}_1^{(2)}, \mathbf{y}_2^{(2)})$ or $\mathcal{E}$ has found a $\mathsf{MSIS}_{n, m_1+m_2, B}$ solution for the matrix $\begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix}$. From now on, we assume the former case.

Finally, let us define the following vectors:

$$\bar{\mathbf{s}} := \begin{bmatrix} (\sigma^i(\bar{\mathbf{s}}_1))_{i \in [k]} \\ (\sigma^i(\bar{\mathbf{m}}))_{i \in [k]} \end{bmatrix} \text{ and } \bar{\mathbf{y}} := \begin{bmatrix} (\sigma^i(\bar{\mathbf{y}}_1))_{i \in [k]} \\ -(\sigma^i(\mathbf{B}\bar{\mathbf{y}}_2))_{i \in [k]} \end{bmatrix}.$$

Then, from the verification equations we have

$$\mathbf{z}^{(i)^T}\mathbf{R}_2\mathbf{z}^{(i)} + c^{(i)}\mathbf{r}_1^T\mathbf{z}^{(i)} + c^{(i)^2}r_0 - \left(c^{(i)}t - \mathbf{b}^T\mathbf{z}_2^{(i)}\right) = v \text{ for } i = 0, 1, 2 \tag{33}$$

where

$$\mathbf{z}^{(i)} := \begin{bmatrix} (\sigma^i(\mathbf{z}_1^{(i)}))_{i \in [k]} \\ (\sigma^i(c^{(i)}\mathbf{t}_B - \mathbf{B}\mathbf{z}_2^{(i)}))_{i \in [k]} \end{bmatrix} = c^{(i)}\bar{\mathbf{s}} + \bar{\mathbf{y}}.$$

By expanding Equation 33, we obtain

$$c^{(i)^2}\left(\bar{\mathbf{s}}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\bar{\mathbf{s}} + r_0\right) + c^{(i)}g_1' + g_0' = 0 \text{ for } i = 0, 1, 2$$

where

$$\begin{aligned} g_1' &= \bar{\mathbf{s}}^T\mathbf{R}_2\bar{\mathbf{y}} + \bar{\mathbf{y}}^T\mathbf{R}_2\bar{\mathbf{s}} + \mathbf{r}_1^T\bar{\mathbf{y}} - \bar{g}_1 \\ g_0' &= \bar{\mathbf{y}}^T\mathbf{R}_2\bar{\mathbf{y}} + \mathbf{b}^T\bar{\mathbf{y}}_2 - v. \end{aligned}$$

Alternatively, we can write these three equations as follows:

$$\begin{bmatrix} 1 & c^{(0)} & c^{(0)^2} \\ 1 & c^{(1)} & c^{(1)^2} \\ 1 & c^{(2)} & c^{(2)^2} \end{bmatrix} \begin{bmatrix} g_0' \\ g_1' \\ \bar{\mathbf{s}}^T\mathbf{R}_2\bar{\mathbf{s}} + \mathbf{r}_1^T\bar{\mathbf{s}} + r_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Since the difference of each two challenges in $\{c^{(0)}, c^{(1)}, c^{(2)}\}$ is invertible over $\mathcal{R}_q$, we must have that $\bar{\mathbf{s}}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\bar{\mathbf{s}} + r_0 = 0$. Hence, the statement holds. □

4.2 Many Quadratic Equations with Automorphisms

We consider a scenario when the prover wants to simultaneously prove N quadratic relations. Clearly, if one were to prove them separately using the approach from Section 4.1, one would end up committing to N garbage polynomials g. Here, we circumvent this issue by linear-combining the N equations into one quadratic equation and prove it using the protocol in Fig. 6. This results in committing to only one garbage polynomials at the cost of reducing the soundness error by a negligible additive factor.

More precisely, suppose that we want to prove for N public $k(m_1 + \ell)$-variate quadratic functions $f_1, \ldots, f_N$ over $\mathcal{R}_q$ that

$$f_j\left((\sigma^i(\mathbf{s}_1))_{i \in [k]}, (\sigma^i(\mathbf{m}))_{i \in [k]}\right) = 0 \text{ for } i \in [N]. \tag{34}$$

Private information: $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$ so that $\|\mathbf{s}_1\| \leqslant \alpha$, $\mathbf{s}_2 \leftarrow \chi^{m_2}$
Public information: $\mathbf{A}_1 \in \mathcal{R}_q^{n\times m_1}, \mathbf{A}_2 \in \mathcal{R}_q^{n\times m_2}, \mathbf{B} \in \mathcal{R}_q^{\ell\times m_2}, \mathbf{b} \in \mathcal{R}_q^{m_2}$
$\begin{bmatrix}\mathbf{t}_A \\ \mathbf{t}_B\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1 \\ \mathbf{0}\end{bmatrix} \cdot \mathbf{s}_1 + \begin{bmatrix}\mathbf{A}_2 \\ \mathbf{B}\end{bmatrix} \cdot \mathbf{s}_2 + \begin{bmatrix}\mathbf{0} \\ \mathbf{m}\end{bmatrix},$
$f_1, \ldots, f_N : \mathcal{R}_q^{k(m_1+\ell)} \to \mathcal{R}_q, \sigma \in \mathsf{Aut}(\mathcal{R}_q)$

Prover		Verifier
		$\mu_1, \ldots, \mu_N \leftarrow \mathcal{R}_q$
	$\xleftarrow{\mu_1, \ldots, \mu_N}$	
$f := \sum_{j=1}^{N} \mu_j f_j$ Run $\mathit{\Pi}^{(2)}\left((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), \sigma, f\right)$		

Fig. 7: Commit-and-prove protocol $\mathit{\Pi}^{(2)}_{\mathsf{many}}\left((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), \sigma, (f_1, f_2, \ldots, f_N)\right)$ for messages $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$, randomness $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$ and $\bar{c} \in \bar{\mathcal{C}}$ which satisfy: $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A$, $\mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t}_B$ (ii) $\|\mathbf{s}_i\bar{c}\| \leqslant 2\mathfrak{s}_i\sqrt{2m_i d}$ for $i = 1, 2$ (where $\mathfrak{s}_i$ are used in Fig. 6) and (iii) $f_j\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) = 0$ for $j \in [N]$. Vector $\mathbf{b}$ is used in the sub-protocol $\mathit{\Pi}^{(2)}$.

We let the verifier begin by sending challenges $\mu_1, \ldots, \mu_N \leftarrow \mathcal{R}_q$. Then, we define a single quadratic function

$$f := \sum_{i=j}^{N} \mu_j f_j$$

and prove that

$$f\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) = 0 \tag{35}$$

using the protocol from Fig. 6. Now, we observe that if one of the conditions in (34) does not hold, then Equation 35 is satisfied with probability at most $q_1^{-d/2}$ (recall that $X^d + 1$ splits into two irreducible factors modulo each q_i).

The protocol is provided in Fig. 7. We skip the full security analysis since it will be implicitly included in the more general case in Theorem 4.5 but we only consider knowledge soundness.

Lemma 4.3. *Consider the protocol in Fig. 7. Then, there is an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a probabilistic prover $\mathcal{P}^*$, which convinces $\mathcal{V}$ with probability $\varepsilon \geqslant 2/|\mathcal{C}| + q_1^{-d/2}$, extractor $\mathcal{E}$ with probability at least $\varepsilon - 2/|\mathcal{C}| - q_1^{-d/2}$ either outputs $(\bar{\mathbf{s}}_2, \bar{\mathbf{s}}_1, \bar{\mathbf{m}}) \in \mathcal{R}_q^{m_1+m_2+\ell}$ and $\bar{c} \in \mathcal{R}_q^{\times}$ such that*

- $\begin{bmatrix}\mathbf{t}_A \\ \mathbf{t}_B\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1 \\ \mathbf{0}\end{bmatrix} \cdot \bar{\mathbf{s}}_1 + \begin{bmatrix}\mathbf{A}_2 \\ \mathbf{B}\end{bmatrix} \cdot \bar{\mathbf{s}}_2 + \begin{bmatrix}\mathbf{0} \\ \bar{\mathbf{m}}\end{bmatrix}$
- *for all $j \in [N]$, $f_j\left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) = 0$*
- $\|\bar{c}\|_\infty \leqslant 2\kappa$
- $\|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ *and* $\|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$

or a $\mathsf{MSIS}_{n,m_1+m_2,B}$ solution for $\begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix}$ in expected time at most $6T$ where running $\mathcal{P}^*$ once is assumed to take at most T time and $B = 8\eta\sqrt{(\mathfrak{s}_1\sqrt{2m_1 d})^2 + (\mathfrak{s}_2\sqrt{2m_2 d})^2}$.

Proof. Let $\mathcal{P}^*$ be a probabilistic prover which convinces the verifier with probability $\varepsilon > 2|\mathcal{C}|^{-1} + q_1^{-d/2}$ and runs in time at most T. We define a deterministic algorithm $\mathcal{A}(\rho, \boldsymbol{\mu})$ which given randomness $\rho \in \mathfrak{R}$ and a challenge $\boldsymbol{\mu} \in \mathcal{R}_q^N$, it does the following. It simply runs the extractor $\mathcal{E}^*(\rho)$ from the proof of Theorem 4.2 with randomness ρ which then calls $\mathcal{P}^*(\boldsymbol{\mu})$ in a black-box way. We say that $\mathcal{A}$ succeeds if $\mathcal{A}$ outputs $(\boldsymbol{\mu}, \bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c})$ such that

- $\begin{bmatrix}\mathbf{t}_A \\ \mathbf{t}_B\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1 \\ \mathbf{0}\end{bmatrix} \cdot \bar{\mathbf{s}}_1 + \begin{bmatrix}\mathbf{A}_2 \\ \mathbf{B}\end{bmatrix} \cdot \bar{\mathbf{s}}_2 + \begin{bmatrix}\mathbf{0} \\ \bar{\mathbf{m}}\end{bmatrix}$
- $\|\bar{c}\|_\infty \leqslant 2\kappa$

- $\|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ and $\|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$
- $\sum_{j=1}^{N} \mu_j f_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) = 0.$

Note that $\mathcal{A}$ (and later on $\mathcal{E}$) could also extract a valid MSIS solution. For presentation, we will assume this never occurs. From Theorem 4.2 we know that the expected runtime of $\mathcal{A}$ for any $\boldsymbol{\mu}$ and $\rho \leftarrow \mathfrak{R}$ is at most $3T$ and the probability that $\mathcal{A}$ succeeds for random ρ and $\boldsymbol{\mu}$ is at least $\epsilon - 2/|\mathcal{C}|$.

We introduce the following notation. Let $H \subseteq \mathfrak{R}\times\mathcal{R}_q^N$ be the set of pairs $(\rho,\boldsymbol{\mu})$ such that $\mathcal{A}(\rho,\boldsymbol{\mu})$ succeeds. Also, define $H(\rho)$ to be the set of all $\boldsymbol{\mu}$ for which $(\rho,\boldsymbol{\mu}) \in H$. For fixed $(\rho,\boldsymbol{\mu}) \in H$, denote $\bar{\mathbf{s}}_1^{(\rho,\boldsymbol{\mu})}$ to be the $\bar{\mathbf{s}}_1$ part of the output of $\mathcal{A}(\rho,\boldsymbol{\mu})$ (and similarly for other variables) and denote

$$\bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})} := \begin{bmatrix} (\sigma^i(\bar{\mathbf{s}}_1^{(\rho,\boldsymbol{\mu})}))_{i\in[k]} \\ (\sigma^i(\bar{\mathbf{m}}^{(\rho,\boldsymbol{\mu})}))_{i\in[k]} \end{bmatrix} \in \mathcal{R}_q^{k(m_1+\ell)}.$$

Finally, we define

$$H' := \left\{(\rho,\boldsymbol{\mu}) \in H : \exists j \in [N], f_j\left(\bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})}\right) \neq 0\right\}.$$

Then, we have the following claim.

Lemma 4.4. *If $(\rho,\boldsymbol{\mu}) \in H$ then $\Pr_{\boldsymbol{\mu}'\leftarrow\mathcal{R}_q^N}[(\rho,\boldsymbol{\mu}') \in H] > 0$. Moreover, if $(\rho,\boldsymbol{\mu}) \in H'$ then*

$$\Pr_{\boldsymbol{\mu}'\leftarrow\mathcal{R}_q^N}\left[\sum_{j=1}^{N} \mu'_j f_j\left(\bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})}\right) = 0\right] \leqslant q_1^{-d/2}.$$

Proof. First, we observe that if $(\rho,\boldsymbol{\mu}) \in H$ then

$$\Pr_{\boldsymbol{\mu}'\leftarrow\mathcal{R}_q^N}[(\rho,\boldsymbol{\mu}') \in H] \geqslant \Pr_{\boldsymbol{\mu}'\leftarrow\mathcal{R}_q^N}[\boldsymbol{\mu}' = \boldsymbol{\mu}] > 0.$$

Now, if $f_\iota\left(\bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})}\right) \neq 0$ for some ι, then for any fixed $a \in \mathcal{R}_q$, the probability over $\mu'_\iota \leftarrow \mathcal{R}_q$ that $\mu'_\iota \cdot f_\iota\left(\bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})}\right) = a$ is at most $q_1^{-d/2}$. Hence, the claim follows. □

Now, we can define our extractor $\mathcal{E}$.

1. Sample $\rho \leftarrow \mathfrak{R}$ and $\boldsymbol{\mu} \in \mathcal{R}_q^N$ and run $\mathcal{A}(\rho,\boldsymbol{\mu})$. If $\mathcal{A}(\rho,\boldsymbol{\mu})$ does not succeed, abort.
2. If $\mathcal{A}(\rho,\boldsymbol{\mu})$ succeeds, run $\mathcal{A}(\rho',\boldsymbol{\mu}')$ with fresh $\rho' \leftarrow \mathfrak{R}$ and $\boldsymbol{\mu}' \leftarrow \mathcal{R}_q^N$ until $\mathcal{A}$ succeeds.

We say that $\mathcal{E}$ succeeds if it extracts two tuples $x = (\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c})$ and $x' = (\bar{\mathbf{s}}'_1, \bar{\mathbf{m}}', \bar{\mathbf{s}}'_2, \bar{c}')$ such that one of the conditions below holds:

- $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2) \neq (\bar{\mathbf{s}}'_1, \bar{\mathbf{s}}'_2)$, $\max(\|\bar{c}\|_\infty, \|\bar{c}'\|_\infty) \leqslant 2\kappa$ and $\max(\|\bar{c}\bar{\mathbf{s}}_i\|, \|\bar{c}'\bar{\mathbf{s}}'_i\|) \leqslant 2\mathfrak{s}_i\sqrt{2m_i d}$ for $i = 1, 2$, and

$$\begin{bmatrix}\mathbf{A}_1\\ \mathbf{0}\end{bmatrix}\cdot\bar{\mathbf{s}}_1 + \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix}\cdot\bar{\mathbf{s}}_2 + \begin{bmatrix}\mathbf{0}\\ \bar{\mathbf{m}}\end{bmatrix} = \begin{bmatrix}\mathbf{t}_A\\ \mathbf{t}_B\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{0}\end{bmatrix}\cdot\bar{\mathbf{s}}'_1 + \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix}\cdot\bar{\mathbf{s}}'_2 + \begin{bmatrix}\mathbf{0}\\ \bar{\mathbf{m}}'\end{bmatrix}$$

- for all $j \in [N], f_j\left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) = 0$ and $\|\bar{c}\|_\infty \leqslant 2\kappa$ and $\|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ and $\|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$ and

$$\begin{bmatrix}\mathbf{t}_A\\ \mathbf{t}_B\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{0}\end{bmatrix}\cdot\bar{\mathbf{s}}_1 + \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix}\cdot\bar{\mathbf{s}}_2 + \begin{bmatrix}\mathbf{0}\\ \bar{\mathbf{m}}\end{bmatrix}.$$

In the first case we break the binding property of the commitment scheme and thus find the relevant MSIS solution. On the other hand, we extract the witness in the second case. Then, we have the following claims about $\mathcal{E}$.

Claim. The expected number of calls to $\mathcal{A}$ is at most 2.

Proof. Let X be the expected number of calling $\mathcal{A}$ and let ε be the probability that $\mathcal{A}(\rho, \boldsymbol{\mu})$ succeeds for random ρ and $\boldsymbol{\mu}$. Define E to be the event that $\mathcal{A}$ succeeds in the first step. Then,

$$\mathbb{E}[X] = \mathbb{E}[X|E] \cdot \varepsilon + \mathbb{E}[X|E] \cdot (1-\varepsilon) = \left(1 + \frac{1}{\varepsilon}\right) \cdot \varepsilon + 1 \cdot (1-\varepsilon) = 2.$$

□

We conclude from the claim above that the expected runtime of $\mathcal{E}$ is at most $6T$.

Claim. Probability that $\mathcal{E}$ succeeds is at least $\epsilon - 2/|\mathcal{C}| - q_1^{-d/2}$.

Proof. First, we observe that $\mathcal{E}$ terminates (without aborting) with probability at least $\epsilon - 2/|\mathcal{C}|$. Suppose $\mathcal{E}$ indeed terminates and let us write $(\boldsymbol{\mu}, \bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c})$ and $(\boldsymbol{\mu}', \bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{s}}_2', \bar{c}')$ to be the respective outputs of $\mathcal{A}$ in the first and second step of $\mathcal{E}$. We have the three disjoint cases as described below:

Case 1:

- $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2) \neq (\bar{\mathbf{s}}_1', \bar{\mathbf{m}}, \bar{\mathbf{s}}_2')$
- $\sum_{j=1}^{N} \mu_j f_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) = 0$ and $\sum_{j=1}^{N} \mu_j' f_j \left((\sigma^i(\bar{\mathbf{s}}_1'))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}'))_{i\in[k]}\right) = 0$
- $\max(\|\bar{c}\|_\infty, \|\bar{c}'\|_\infty) \leqslant 2\kappa$ and $\max(\|\bar{c}\bar{\mathbf{s}}_i\|, \|\bar{c}'\bar{\mathbf{s}}_i'\|) \leqslant 2\mathfrak{s}_i\sqrt{2m_i d}$ for $i = 1, 2$
- $\begin{bmatrix}\mathbf{A}_1\\ \mathbf{0}\end{bmatrix} \cdot \bar{\mathbf{s}}_1 + \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix} \cdot \bar{\mathbf{s}}_2 + \begin{bmatrix}\mathbf{0}\\ \bar{\mathbf{m}}\end{bmatrix} = \begin{bmatrix}\mathbf{t}_A\\ \mathbf{t}_B\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{0}\end{bmatrix} \cdot \bar{\mathbf{s}}_1' + \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix} \cdot \bar{\mathbf{s}}_2' + \begin{bmatrix}\mathbf{0}\\ \bar{\mathbf{m}}'\end{bmatrix}$

Case 2:

- $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2) = (\bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{s}}_2')$
- $\sum_{j=1}^{N} \mu_j f_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) = 0$ and $\sum_{j=1}^{N} \mu_j' f_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) = 0$
- $\|\bar{c}\|_\infty \leqslant 2\kappa$ and $\|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ and $\|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$
- $\begin{bmatrix}\mathbf{A}_1\\ \mathbf{0}\end{bmatrix} \cdot \bar{\mathbf{s}}_1 + \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix} \cdot \bar{\mathbf{s}}_2 + \begin{bmatrix}\mathbf{0}\\ \bar{\mathbf{m}}\end{bmatrix} = \begin{bmatrix}\mathbf{t}_A\\ \mathbf{t}_B\end{bmatrix}$
- for all $j \in [N]$, $f_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) = 0$.

Case 3:

- $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2) = (\bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{s}}_2')$
- $\sum_{j=1}^{N} \mu_j f_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) = 0$ and $\sum_{j=1}^{N} \mu_j' f_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) = 0$
- $\|\bar{c}\|_\infty \leqslant 2\kappa$ and $\|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ and $\|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$
- $\begin{bmatrix}\mathbf{A}_1\\ \mathbf{0}\end{bmatrix} \cdot \bar{\mathbf{s}}_1 + \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix} \cdot \bar{\mathbf{s}}_2 + \begin{bmatrix}\mathbf{0}\\ \bar{\mathbf{m}}\end{bmatrix} = \begin{bmatrix}\mathbf{t}_A\\ \mathbf{t}_B\end{bmatrix}$
- there exists $j \in [N]$ so that $f_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) \neq 0$.

Define E_i to be the event that $\mathcal{E}$ terminates and Case i occurs. Then, we have

$$\epsilon - 2/|\mathcal{C}| \leqslant \Pr[\mathcal{E} \text{ terminates}] = \Pr[E_1 \vee E_2 \vee E_3]$$

and

$$\Pr[\mathcal{E} \text{ succeeds}] \geqslant \Pr[E_1 \vee E_2].$$

Hence, we only need to upper-bound the probability $\Pr[E_3]$. We apply Lemma 4.4 as follows:

$$\begin{aligned}
\Pr[E_3] &\leqslant \Pr\left[\begin{array}{c}(\mathcal{A}(\rho,\boldsymbol{\mu}) \text{ succeeds}) \wedge \left(\sum_{j=1}^{N} \mu'_j f_j\left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) = 0\right) \\ \wedge \left(\exists j \in [N] : f_j\left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) \neq 0\right)\end{array}\right] \\
&\leqslant \frac{1}{|\mathfrak{R}| \cdot q^{Nd}} \sum_{(\rho,\boldsymbol{\mu})\in H'} \Pr_{\boldsymbol{\mu}'}\left[\sum_{j=1}^{N} \mu'_j f_j\left(\bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})}\right) = 0\right] \\
&\leqslant \frac{1}{|\mathfrak{R}| \cdot q^{Nd}} \sum_{(\rho,\boldsymbol{\mu})\in H'} q_1^{-d/2} \\
&\leqslant \frac{1}{|\mathfrak{R}| \cdot q^{Nd}} \sum_{(\rho,\boldsymbol{\mu})\in \mathfrak{R}\times\mathcal{R}_q^N} q_1^{-d/2} \\
&\leqslant q_1^{-d/2}.
\end{aligned}$$

□

The statement thus follows by combining the two previous claims. □

4.3 Polynomial Evaluations with Vanishing Constant Coefficients

Suppose we want to prove simultaneously N quadratic relations (i.e. (34)) and *additionally* prove that for quadratic $k(m_1+\ell)$-variate polynomials $F_1,\ldots,F_M$, evaluations $F_j\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right)$ have the constant coefficient equal to zero. Concretely, if we denote

$$x_j := F_j\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) \in \mathcal{R}_q$$

then $\widetilde{x}_j = 0$ for $j \in [M]$.

For simplicity we first present an approach with soundness error $1/q_1$. We apply the strategy from [ENS20] and first commit to a random masking polynomial $g \leftarrow \{x \in \mathcal{R}_q : \widetilde{x} = 0\}$. Then, given random challenges $\gamma_1,\ldots,\gamma_M \leftarrow \mathbb{Z}_q$, we send

$$h := g + \sum_{j=1}^{M} \gamma_j F_j\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) \tag{36}$$

to the verifier. Then, it simply checks whether the constant coefficient of h is indeed equal to zero. What is left to prove is that h is well-formed, i.e. (36) holds. This is done by defining the quadratic function $f_{N+1} : \mathcal{R}_q^{k(m_1+\ell+1)} \to \mathcal{R}_q$ as follows.

Let $\mathbf{x}_1 \in \mathcal{R}_q^{km_1}$, $\mathbf{x}_2 = (\mathbf{x}_{2,1},\ldots,\mathbf{x}_{2,k}) \in \mathcal{R}_q^{k(\ell+1)}$ and denote

$$\mathbf{x}_{2,j} := \mathbf{x}_{2,j}^{(m)} \parallel x_{2,j}^{(g)} \in \mathcal{R}_q^{\ell+1} \text{ for } j \in [k], \quad \mathbf{x}_2^{(m)} := \left(\mathbf{x}_{2,1}^{(m)},\ldots,\mathbf{x}_{2,k}^{(m)}\right).$$

Then,

$$f_{N+1}(\mathbf{x}_1,\mathbf{x}_2) := x_{2,1}^{(g)} + \sum_{j=1}^{M} \gamma_j F_j\left(\mathbf{x}_1, \mathbf{x}_2^{(m)}\right) - h.$$

By construction, if $(\mathbf{x}_1,\mathbf{x}_2) = (\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m} \parallel g))_{i\in[k]}$ then

$$\mathbf{x}_1 = \sigma^i(\mathbf{s}_1))_{i\in[k]}, \quad \mathbf{x}_2^{(m)} = (\sigma^i(\mathbf{m}))_{i\in[k]} \quad \text{and} \quad x_{2,1}^{(g)} = g.$$

Moreover,(36) holds if and only if

$$f_{N+1}\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m} \parallel g))_{i\in[k]}\right) = 0.$$

Recall that we also want to prove (34). We can define analogous polynomials $f_1, \ldots, f_N : \mathcal{R}_q^{k(m_1+\ell+1)} \to \mathcal{R}_q$ as:

$$f_j(\mathbf{x}_1, \mathbf{x}_2) := f_j\left(\mathbf{x}_1, \mathbf{x}_2^{(m)}\right).$$

Hence, we simply want to prove that for every $j = 1, 2, \ldots, N+1$:

$$f_j\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m} \parallel g))_{i\in[k]}\right) = 0.$$

Finally, this can then be directly done using the protocol

$$\Pi_{\mathsf{many}}^{(2)}\left((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}, g), \sigma, (f_1, f_2, \ldots, f_{N+1})\right)$$

in Fig. 7.

Private information: $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$ so that $\|\mathbf{s}_1\| \leqslant \alpha$, $\mathbf{s}_2 \leftarrow \chi^{m_2}$
Public information: $\mathbf{A}_1 \in \mathcal{R}_q^{n\times m_1}, \mathbf{A}_2 \in \mathcal{R}_q^{n\times m_2}, \mathbf{B} \in \mathcal{R}_q^{\ell\times m_2}, \mathbf{B}_g \in \mathcal{R}_q^{\lambda\times m_2}, \mathbf{b} \in \mathcal{R}_q^{m_2}$
$\begin{bmatrix}\mathbf{t}_A \\ \mathbf{t}_B\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1 \\ \mathbf{0}\end{bmatrix} \cdot \mathbf{s}_1 + \begin{bmatrix}\mathbf{A}_2 \\ \mathbf{B}\end{bmatrix} \cdot \mathbf{s}_2 + \begin{bmatrix}\mathbf{0} \\ \mathbf{m}\end{bmatrix},$
$f_1, \ldots, f_N, F_1, \ldots, F_M : \mathcal{R}_q^{k(m_1+\ell)} \to \mathcal{R}_q, \sigma \in \mathsf{Aut}(\mathcal{R}_q)$

Prover | Verifier

$\mathbf{s} := \begin{bmatrix}(\sigma^i(\mathbf{s}_1))_{i\in[k]} \\ (\sigma^i(\mathbf{m}))_{i\in[k]}\end{bmatrix}$
$\mathbf{g} := (g_1, \ldots, g_\lambda) \leftarrow \{x : \mathcal{R}_q : \tilde{x} = 0\}^\lambda$
$\mathbf{t}_g := \mathbf{B}_g\mathbf{s}_2 + \mathbf{g}$

$\xrightarrow{\mathbf{t}_g}$

$\Gamma = (\gamma_{i,j}) \leftarrow \mathbb{Z}_q^{\lambda\times M}$

$\xleftarrow{(\gamma_{i,j})_{i\in[\lambda], j\in[M]}}$

for $i \in [\lambda]$:
  $h_i := g_i + \sum_{j=1}^{M} \gamma_{i,j} F_j(\mathbf{s})$

$\xrightarrow{h_1, \ldots, h_\lambda}$

define $f_1, \ldots, f_{N+\lambda}$ as in (38) and (39)
run $\Pi_{\mathsf{many}}^{(2)}\left((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m} \parallel \mathbf{g}), \sigma, (f_i)_{i\in[N+\lambda]}\right)$

Accept iff
$\Pi_{\mathsf{many}}^{(2)}$ verifies and
$\tilde{h}_1 = \ldots = \tilde{h}_\lambda = 0$

Fig. 8: Commit-and-prove protocol $\Pi_{\mathsf{eval}}^{(2)}\left((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), \sigma, (f_1, \ldots, f_N), (F_1, \ldots, F_M)\right)$ for messages $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$, randomness $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$ and $\bar{c} \in \bar{\mathcal{C}}$ which satisfy: $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A$, $\mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t}_B$ (ii) $\|\mathbf{s}_i\bar{c}\| \leqslant 2\mathfrak{s}_i\sqrt{2m_i d}$ for $i = 1, 2$, (iii) $f_j\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) = 0$ for $j \in [N]$ (where $\mathfrak{s}_i$ are used in Fig. 6) and (iv) all the evaluations $F_j\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right)$, where $j \in [M]$, have constant coefficients equal to zero. Vector $\mathbf{b}$ is used in the sub-protocol $\Pi_{\mathsf{many}}^{(2)}$.

We provide intuition for the soundness argument. Assume that the verifier is convinced that h is of the correct form (36) and $\widetilde{h} = 0$. Also, note that a cheating prover committed to g before seeing the challenges $\gamma_1, \ldots, \gamma_M$. Hence, if for some $j \in [M]$, the constant coefficient of $F_j\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right)$ is non-zero, then the cheating prover has probability at most $1/q_1$ of guessing the constant coefficient of $\sum_{j=1}^{M} \gamma_j F_j\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right)$.

Boosting Soundness. We exponentially decrease the soundness error by parallel repetition. Namely, in order to obtain $q_1^{-\lambda}$ soundness error, we commit to λ random masking polynomials $\mathbf{g} = (g_1, \ldots, g_\lambda) \leftarrow \{x : \mathcal{R}_q : \widetilde{x} = 0\}^\lambda$ as follows:

$$\mathbf{t}_g := \mathbf{B}_g\mathbf{s}_2 + \mathbf{g}.$$

Then, we send $\mathbf{t}_g$ to the verifier which in return outputs the challenge matrix $(\gamma_{i,j})_{i\in[\lambda],j\in[M]} \leftarrow \mathbb{Z}_q^{\lambda\times M}$. Then, we compute the vector $\mathbf{h} = (h_1, \ldots, h_\lambda)$ as follows:

$$\begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_\lambda \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_\lambda \end{bmatrix} + \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,M} \\ \vdots & \vdots & \cdots & \vdots \\ \gamma_{\lambda,1} & \gamma_{\lambda,2} & \cdots & \gamma_{\lambda,M} \end{bmatrix} \begin{bmatrix} F_1\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) \\ F_2\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) \\ \vdots \\ F_M\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) \end{bmatrix} \tag{37}$$

and send it to the verifier. It directly checks if all polynomials $h_1, \ldots, h_\lambda \in \mathcal{R}_q$ have constant coefficients equal to zero.

As before, we still need to prove that vector $\mathbf{h}$ was constructed correctly. We reduce this problem to proving quadratic relations. Namely, we define polynomials $f_{N+1}, \ldots, f_{N+\lambda} : \mathcal{R}_q^{k(m_1+\ell+\lambda)} \to \mathcal{R}_q$ as follows.

Let $\mathbf{x}_1 \in \mathcal{R}_q^{km_1}$, $\mathbf{x}_2 = (\mathbf{x}_{2,1}, \ldots, \mathbf{x}_{2,k}) \in \mathcal{R}_q^{k(\ell+\lambda)}$ and denote

$$\begin{aligned} \mathbf{x}_{2,j} &:= \left(\mathbf{x}_{2,j}^{(m)}, \mathbf{x}_{2,j}^{(g)}\right) \in \mathcal{R}_q^{\ell+\lambda} \text{ for } j \in [k], \\ \mathbf{x}_2^{(m)} &:= \left(\mathbf{x}_{2,1}^{(m)}, \ldots, \mathbf{x}_{2,k}^{(m)}\right), \quad \mathbf{x}_{2,1}^{(g)} := \left(x_{2,1,1}^{(g)}, \ldots, x_{2,1,\lambda}^{(g)}\right). \end{aligned}$$

Then,

$$f_{N+i}(\mathbf{x}_1, \mathbf{x}_2) := x_{2,1,i}^{(g)} + \sum_{j=1}^{M} \gamma_{i,j} F_j\left(\mathbf{x}_1, \mathbf{x}_2^{(m)}\right) - h_i \text{ for } i \in [\lambda]. \tag{38}$$

By construction, if $(\mathbf{x}_1, \mathbf{x}_2) = (\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m} \parallel \mathbf{g}))_{i\in[k]}$ then

$$\mathbf{x}_1 = (\sigma^i(\mathbf{s}_1))_{i\in[k]}, \quad \mathbf{x}_2^{(m)} = (\sigma^i(\mathbf{m}))_{i\in[k]} \quad \text{and} \quad x_{2,1,i}^{(g)} = g_i.$$

Furthermore, Equation (37) is true if and only if for all $j \in [\lambda]$ we have:

$$f_{N+j}\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m} \parallel \mathbf{g}))_{i\in[k]}\right) = 0.$$

Since we also need to prove (34), for convenience we define polynomials $f_1, \ldots, f_N : \mathcal{R}_q^{k(m_1+\ell+\lambda)} \to \mathcal{R}_q$ as:

$$f_j(\mathbf{x}_1, \mathbf{x}_2) := f_j\left(\mathbf{x}_1, \mathbf{x}_2^{(m)}\right). \tag{39}$$

Finally, we simply run $\mathit{\Pi}_{\mathsf{quad-many}}\left((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}, \mathbf{g}), \sigma, (f_j)_{j\in[N+\lambda]}\right)$ from Fig. 7. We summarise the protocol in Fig. 8 and provide commitment and proof size analysis in Section 6.1.

Note that with this approach we need to commit to additional λ garbage polynomials.

Security Analysis. We present the security properties of the protocol in Fig. 8 below.

Theorem 4.5. *Consider the protocol in Fig. 8 and let $\chi = S_\nu$. Suppose $\mathfrak{s}_1 = \gamma_1 \alpha \eta$ and $\mathfrak{s}_2 = \gamma_2 \nu \eta \sqrt{m_2 d}$ for some $\gamma_1, \gamma_2 > 0$ where η is chosen as in Section 2.7.*

For completeness, let $m_1, m_2 \geqslant 640/d$. Then, the honest prover $\mathcal{P}$ convinces the honest verifier $\mathcal{V}$ with probability

$$\approx \frac{1}{2\exp\left(\frac{14}{\gamma_1} + \frac{1}{2\gamma_1^2} + \frac{1}{2\gamma_2^2}\right)}.$$

For commit-and-prove simulatability, there exists a simulator $\mathcal{S}$ that, without access to private information $\mathbf{s}_1, \mathbf{m}$, outputs a simulation of a commitment $(\mathbf{t}_A, \mathbf{t}_B)$ along with a non-aborting transcript of the protocol between prover $\mathcal{P}$ and verifier $\mathcal{V}$ such that for every algorithm $\mathcal{A}$ that has advantage ε in distinguishing the simulated commitment and transcript from the actual commitment and transcript, whenever the prover *does not abort, there is an algorithm $\mathcal{A}'$ with the same running time that has advantage $\varepsilon/2 - 2^{-128}$ in distinguishing Extended-$\mathsf{MLWE}_{n+\ell+\lambda+1,m_2-n-\ell-\lambda-1,\chi,\mathcal{C},\mathfrak{s}_2}$.*

For soundness, there is an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a probabilistic prover $\mathcal{P}^*$, which convinces $\mathcal{V}$ with probability $\varepsilon \geqslant 2/|\mathcal{C}| + q_1^{-d/2} + q_1^{-\lambda}$, extractor $\mathcal{E}$ with probability at least $\varepsilon - 2/|\mathcal{C}| - q_1^{-d/2} - q_1^{-\lambda}$ either outputs $(\bar{\mathbf{s}}_2, \bar{\mathbf{s}}_1, \bar{\mathbf{m}}) \in \mathcal{R}_q^{m_1+m_2+\ell}$ and $\bar{c} \in \mathcal{R}_q^\times$ such that

- $\begin{bmatrix}\mathbf{t}_A\\ \mathbf{t}_B\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{0}\end{bmatrix}\cdot\bar{\mathbf{s}}_1 + \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix}\cdot\bar{\mathbf{s}}_2 + \begin{bmatrix}\mathbf{0}\\ \bar{\mathbf{m}}\end{bmatrix}$
- $f_j\left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) = 0$ *for* $j \in [N]$
- *each* $F_j\left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) \in \mathcal{R}_q$, *where* $j \in [M]$, *has constant coefficient equal to zero*
- $\|\bar{c}\|_\infty \leqslant 2\kappa$
- $\|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1d}$ *and* $\|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2\mathfrak{s}_2\sqrt{2m_2d}$

or a $\mathsf{MSIS}_{n,m_1+m_2,B}$ solution for $\begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix}$ in expected time at most $12T$ where running $\mathcal{P}^*$ once is assumed to take at most T time and $B = 8\eta\sqrt{(\mathfrak{s}_1\sqrt{2m_1d})^2 + (\mathfrak{s}_2\sqrt{2m_2d})^2}$.

Proof. Completeness follows directly from the proof of Theorem 4.2 and the discussion in Section 4.3. As for commit-and-prove simulatability, we simulate the commitment and the transcript identically as in the proof of Theorem 4.2 with two additional steps: (i) we simulate the commitment $\mathbf{t}_g$ to $\mathbf{g}$ by setting $\mathbf{t}_g \leftarrow \mathcal{R}_q^\lambda$ to be a uniformly random vector and (ii) we simulate the polynomials $h_1, \ldots, h_\lambda$ by choosing them uniformly at random from $X := \{x \in \mathcal{R}_q : \tilde{x} = 0\}$. Note that we perfectly simulate each h_i since in the real execution, i.e. (37), g_i's are also sampled uniformly from X and $\sum_{j=1}^{M}\gamma_{i,j}F_j\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) \in X$.

Knowledge Soundness. Let $\mathcal{P}^*$ be a probabilistic prover which runs in time at most T and convinces the verifier with probability $\epsilon > 2|\mathcal{C}|^{-1} + q_1^{-d/2} + q_1^{-\lambda}$. Define a deterministic algorithm $\mathcal{A}(\rho_P, \rho_E, \varGamma)$ which given randomness $\rho = (\rho_P, \rho_E) \in \mathfrak{R}_P \times \mathfrak{R}_E$ and challenge $\varGamma \in \mathbb{Z}_q^{\lambda\times M}$ does the following. It first runs $\mathcal{P}^*(\rho_P)$ on randomness ρ_P with challenge $\varGamma$ and stops after the third round. Let $\mathbf{t}_g$ and $\mathbf{h}$ be the output of $\mathcal{P}^*$ in the first and third round respectively. Then, it runs the extractor $\mathcal{E}^*(\rho_E)$ defined in the proof of Lemma 4.3 with randomness ρ_E (which runs $\mathcal{P}^*(\rho_P, \varGamma)$ in a black-box way). We say that $\mathcal{A}$ succeeds if $\mathcal{A}$ outputs $(\mathbf{t}_g, \varGamma, \mathbf{h}, \bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{g}}, \bar{\mathbf{s}}_2, \bar{c})$ such that

- $\begin{bmatrix}\mathbf{t}_A\\ \mathbf{t}_B\\ \mathbf{t}_g\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix}\cdot\bar{\mathbf{s}}_1 + \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\\ \mathbf{B}_g\end{bmatrix}\cdot\bar{\mathbf{s}}_2 + \begin{bmatrix}\mathbf{0}\\ \bar{\mathbf{m}}\\ \bar{\mathbf{g}}\end{bmatrix}$
- $\tilde{h}_1 = \ldots = \tilde{h}_\lambda = 0$
- $f_j\left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) = 0$ for $j \in [N]$
- for all $i \in [\lambda]$, $h_i = \bar{g}_i + \sum_{j=1}^{M}\gamma_{i,j}F_j\left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right)$
- $\|\bar{c}\|_\infty \leqslant 2\kappa$
- $\|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1d}$ and $\|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2\mathfrak{s}_2\sqrt{2m_2d}$

As before, we assume that $\mathcal{E}^*$ does not solve MSIS since if it did, then so does $\mathcal{A}$ (and later on $\mathcal{E}$). Clearly, by Theorem 4.3, the probability that $\mathcal{A}$ succeeds for random ρ and $\varGamma$ is at least $\epsilon - 2/|\mathcal{C}| - q_1^{-d/2}$. Moreover, the expected runtime $\mathcal{A}(\rho_P, \rho_E, \varGamma)$ for any fixed $\rho_P, \varGamma$ and $\rho_E \leftarrow \mathfrak{R}_E$ is at most $6T$.

We introduce the following notation. Let $H \subseteq \mathfrak{R}_P \times \mathfrak{R}_E \times \mathbb{Z}_q^{\lambda\times M}$ be the set of triples $(\rho, \varGamma)$ such that $\mathcal{A}(\rho, \varGamma)$ succeeds. Also, define $H(\rho_P)$ to be the set of all $(\rho_E, \varGamma)$ for which $(\rho_P, \rho_E, \varGamma) \in H$. For fixed $(\rho, \varGamma) \in H$, denote $\bar{\mathbf{s}}_1^{(\rho,\varGamma)}$ to be the $\bar{\mathbf{s}}_1$ part of the output of $\mathcal{A}(\rho, \varGamma)$ (and similarly for other variables) and denote

$$\bar{\mathbf{s}}^{(\rho,\varGamma)} := \begin{bmatrix}(\sigma^i(\bar{\mathbf{s}}_1^{(\rho,\varGamma)}))_{i\in[k]}\\ (\sigma^i(\bar{\mathbf{m}}^{(\rho,\varGamma)}))_{i\in[k]}\end{bmatrix} \in \mathcal{R}_q^{k(m_1+\ell)}.$$

Finally, we define

$$H' := \left\{(\rho, \varGamma) \in H : \exists j \in [M],\ \text{const. coeff. of } F_j\left(\bar{\mathbf{s}}^{(\rho,\varGamma)}\right)\ \text{is non-zero}\right\}.$$

Then, we have the following claim, almost identical to Lemma 4.4.

Lemma 4.6. *If* $(\rho_P, \rho_E, \Gamma) \in H$ *then* $\Pr_{(\rho'_E, \Gamma') \leftarrow \mathfrak{R}_E \times \mathbb{Z}_q^{\lambda \times M}}[(\rho_P, \rho'_E, \Gamma') \in H] > 0$. *Moreover, if* $(\rho_P, \rho_E, \Gamma) \in H'$ *then*

$$\Pr_{\Gamma' \leftarrow \mathbb{Z}_q^{\lambda \times M}} \left[\forall i \in [\lambda], \tilde{x}_i = 0 \middle| x_i := \bar{g}_i^{(\rho,\Gamma)} + \sum_{j=1}^{M} \gamma'_{i,j} F_j \left(\bar{\mathbf{s}}^{(\rho,\Gamma)} \right) \right] \leqslant q_1^{-\lambda}.$$

Now, we define our extractor $\mathcal{E}$.

1. Sample $\rho = (\rho_P, \rho_E) \leftarrow \mathfrak{R}_P \times \mathfrak{R}_E$ and $\Gamma \in \mathbb{Z}_q^{\lambda \times M}$ and run $\mathcal{A}(\rho, \Gamma)$. If $\mathcal{A}(\rho, \Gamma)$ does not succeed, abort.
2. If $\mathcal{A}(\rho, \Gamma)$ succeeds, run $\mathcal{A}(\rho_P, \rho'_E, \Gamma')$ for the same prover randomness ρ_P but fresh $\rho'_E \leftarrow \mathfrak{R}_E$ and $\Gamma' \leftarrow \mathbb{Z}_q^{\lambda \times M}$ until $\mathcal{A}$ succeeds.

We say that $\mathcal{E}$ succeeds if it extracts two tuples $x = (\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c})$ and $x' = (\bar{\mathbf{s}}'_1, \bar{\mathbf{m}}', \bar{\mathbf{s}}'_2, \bar{c}')$ such that one of the conditions below holds:

– $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2) \neq (\bar{\mathbf{s}}'_1, \bar{\mathbf{s}}'_2)$, $\max(\|\bar{c}\|_\infty, \|\bar{c}'\|_\infty) \leqslant 2\kappa$ and $\max(\|\bar{c}\bar{\mathbf{s}}_i\|, \|\bar{c}'\bar{\mathbf{s}}'_i\|) \leqslant 2\mathfrak{s}_i\sqrt{2m_i d}$ for $i = 1, 2$, and

$$\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \cdot \bar{\mathbf{s}}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} \cdot \bar{\mathbf{s}}_2 + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{m}} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \cdot \bar{\mathbf{s}}'_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} \cdot \bar{\mathbf{s}}'_2 + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{m}}' \end{bmatrix}$$

– for all $j \in [N]$, $f_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i \in [k]}, (\sigma^i(\bar{\mathbf{m}}))_{i \in [k]} \right) = 0$ and for all $j \in [M]$, the constant coefficient of $F_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i \in [k]}, (\sigma^i(\bar{\mathbf{m}}))_{i \in [k]} \right)$ equals zero, and $\|\bar{c}\|_\infty \leqslant 2\kappa$ and $\|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ and $\|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$ and

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \cdot \bar{\mathbf{s}}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} \cdot \bar{\mathbf{s}}_2 + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{m}} \end{bmatrix}.$$

In the first case we break the binding property of the commitment scheme and thus find a relevant MSIS solution. On the other hand, we extract the witness in the second case. Then, we have the following claims about $\mathcal{E}$.

Claim. The expected number of calls to $\mathcal{A}$ is at most 2.

The proof follows identically as in the proof of Lemma 4.3. We conclude that the expected runtime of $\mathcal{E}$ is at most $12T$.

Claim. Probability that $\mathcal{E}$ succeeds is at least $\epsilon - 2/|\mathcal{C}| - q_1^{-d/2} - q_1^{-\lambda}$.

Proof. First, we observe that $\mathcal{E}$ terminates (without aborting) with probability at least $\epsilon - 2/|\mathcal{C}| - q_1^{-d/l}$. Suppose $\mathcal{E}$ indeed terminates and let us write $(\mathbf{t}_g, \Gamma, \mathbf{h}, \bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{g}}, \bar{\mathbf{s}}_2, \bar{c})$ and $(\mathbf{t}_g, \Gamma', \mathbf{h}', \bar{\mathbf{s}}'_1, \bar{\mathbf{m}}', \bar{\mathbf{g}}', \bar{\mathbf{s}}'_2, \bar{c}')$ to be the respective outputs of $\mathcal{A}$ in the first and second step of $\mathcal{E}$. We have the following three disjoint cases.

Case 1:

- $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{g}}, \bar{\mathbf{s}}_2) \neq (\bar{\mathbf{s}}'_1, \bar{\mathbf{m}}', \bar{\mathbf{g}}', \bar{\mathbf{s}}'_2)$
- for $i \in [\lambda]$, $\tilde{h}_i = \tilde{h}'_i = 0$
- for $i \in [\lambda]$, $h_i = \bar{g}_i + \sum_{j=1}^{M} \gamma_{i,j} F_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i \in [k]}, (\sigma^i(\bar{\mathbf{m}}))_{i \in [k]} \right)$
- for $i \in [\lambda]$, $h'_i = \bar{g}'_i + \sum_{j=1}^{M} \gamma'_{i,j} F_j \left((\sigma^i(\bar{\mathbf{s}}'_1))_{i \in [k]}, (\sigma^i(\bar{\mathbf{m}}'))_{i \in [k]} \right)$
- for $j \in [N]$, $f_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i \in [k]}, (\sigma^i(\bar{\mathbf{m}}))_{i \in [k]} \right) = 0$ and $f_j \left((\sigma^i(\bar{\mathbf{s}}'_1))_{i \in [k]}, (\sigma^i(\bar{\mathbf{m}}'))_{i \in [k]} \right) = 0$
- $\max(\|\bar{c}\|_\infty, \|\bar{c}'\|_\infty) \leqslant 2\kappa$ and $\max(\|\bar{c}\bar{\mathbf{s}}_i\|, \|\bar{c}'\bar{\mathbf{s}}'_i\|) \leqslant 2\mathfrak{s}_i\sqrt{2m_i d}$ for $i = 1, 2$
- $\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \cdot \bar{\mathbf{s}}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \\ \mathbf{B}_g \end{bmatrix} \cdot \bar{\mathbf{s}}_2 + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{m}} \\ \bar{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \\ \mathbf{t}_g \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \cdot \bar{\mathbf{s}}'_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \\ \mathbf{B}_g \end{bmatrix} \cdot \bar{\mathbf{s}}'_2 + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{m}}' \\ \bar{\mathbf{g}}' \end{bmatrix}.$

Case 2:

- $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{g}}, \bar{\mathbf{s}}_2) = (\bar{\mathbf{s}}'_1, \bar{\mathbf{m}}', \bar{\mathbf{g}}', \bar{\mathbf{s}}'_2)$
- for $i \in [\lambda]$, $h_i = h'_i = 0$
- for $i \in [\lambda]$, $h_i = \bar{g}_i + \sum_{j=1}^{M} \gamma_{i,j} F_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right)$
- for $i \in [\lambda]$, $h'_i = \bar{g}_i + \sum_{j=1}^{M} \gamma'_{i,j} F_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right)$
- for $j \in [N]$, $f_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) = 0$
- $\|\bar{c}\|_\infty \leqslant 2\kappa$ and $\|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ and $\|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$
- $\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \cdot \bar{\mathbf{s}}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \\ \mathbf{B}_g \end{bmatrix} \cdot \bar{\mathbf{s}}_2 + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{m}} \\ \bar{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \\ \mathbf{t}_g \end{bmatrix}$
- for $j \in [M]$, the constant coefficient of $F_j \left((\sigma^i(\bar{\mathbf{s}}'_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}'))_{i\in[k]}\right)$ is zero.

Case 3:

- $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{g}}, \bar{\mathbf{s}}_2) = (\bar{\mathbf{s}}'_1, \bar{\mathbf{m}}', \bar{\mathbf{g}}', \bar{\mathbf{s}}'_2)$
- for $i \in [\lambda]$, $h_i = h'_i = 0$
- for $i \in [\lambda]$, $h_i = \bar{g}_i + \sum_{j=1}^{M} \gamma_{i,j} F_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right)$
- for $i \in [\lambda]$, $h'_i = \bar{g}_i + \sum_{j=1}^{M} \gamma'_{i,j} F_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right)$
- for $j \in [N]$, $f_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) = 0$
- $\|\bar{c}\|_\infty \leqslant 2\kappa$ and $\|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ and $\|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$
- $\begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \cdot \bar{\mathbf{s}}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \\ \mathbf{B}_g \end{bmatrix} \cdot \bar{\mathbf{s}}_2 + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{m}} \\ \bar{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \\ \mathbf{t}_g \end{bmatrix}$
- there exists $j \in [M]$, so that the constant coefficient of $F_j \left((\sigma^i(\bar{\mathbf{s}}'_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}'))_{i\in[k]}\right)$ is non-zero.

Define E_i to be the event that $\mathcal{E}$ terminates and Case i occurs. Then, we have

$$\epsilon - 2/|\mathcal{C}| - q_1^{-d/2} \leqslant \Pr[\mathcal{E} \text{ terminates}] = \Pr[E_1 \vee E_2 \vee E_3]$$

and

$$\Pr[\mathcal{E} \text{ succeeds}] \geqslant \Pr[E_1 \vee E_2].$$

Hence, we only need to upper-bound the probability $\Pr[E_3]$. Now, by Lemma 4.6 we obtain:

$$\begin{aligned}
\Pr[E_3] &\leqslant \Pr\left[\begin{array}{c} (\mathcal{A}(\rho, \mathit{\Gamma}) \text{ succeeds}) \wedge \left(\exists j \in [M] : \text{ const. coeff. of } F_j \left((\sigma^i(\bar{\mathbf{s}}'_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}'))_{i\in[k]}\right) \text{ is non-zero}\right) \\ \wedge \left(\forall i \in [\lambda], \text{const coeff. of } \bar{g}_i + \sum_{j=1}^{M} \gamma'_{i,j} F_j \left((\sigma^i(\bar{\mathbf{s}}_1))_{i\in[k]}, (\sigma^i(\bar{\mathbf{m}}))_{i\in[k]}\right) \text{ is zero}\right) \end{array}\right] \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho,\mathit{\Gamma})\in H'} \Pr_{(\rho'_E, \mathit{\Gamma}') \leftarrow H(\rho_P)} \left[\forall i \in [\lambda], \text{const coeff. of } \bar{g}_i^{(\rho,\mathit{\Gamma})} + \sum_{j=1}^{M} \gamma'_{i,j} F_j \left(\bar{\mathbf{s}}^{(\rho,\mathit{\Gamma})}\right) \text{ is zero}\right] \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho,\mathit{\Gamma})\in H'} \frac{\Pr_{\mathit{\Gamma}' \leftarrow \mathbb{Z}_q^{\lambda \times M}} \left[\forall i \in [\lambda], \text{const coeff. of } \bar{g}_i^{(\rho,\mathit{\Gamma})} + \sum_{j=1}^{M} \gamma'_{i,j} F_j \left(\bar{\mathbf{s}}^{(\rho,\mathit{\Gamma})}\right) \text{ is zero}\right]}{\Pr_{(\rho'_E, \mathit{\Gamma}') \leftarrow \mathfrak{R}_E \times \mathbb{Z}_q^{\lambda \times M}} [(\rho'_E, \mathit{\Gamma}') \in H(\rho_P)]} \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho,\mathit{\Gamma})\in H'} \frac{q_1^{-\lambda} \cdot q^{\lambda M} \cdot |\mathfrak{R}_E|}{|H(\rho_P)|} \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho,\mathit{\Gamma})\in H} \frac{q_1^{-\lambda} \cdot q^{\lambda M} \cdot |\mathfrak{R}_E|}{|H(\rho_P)|} \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{\rho_P \in \mathfrak{R}_P} \sum_{(\rho_E,\mathit{\Gamma})\in H(\rho_P)} \frac{q_1^{-\lambda} \cdot q^{\lambda M} \cdot |\mathfrak{R}_E|}{|H(\rho_P)|} \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{\rho_P \in \mathfrak{R}_P} |H(\rho_P)| \cdot \frac{q_1^{-\lambda} \cdot q^{\lambda M} \cdot |\mathfrak{R}_E|}{|H(\rho_P)|} \\
&\leqslant q_1^{-\lambda}.
\end{aligned}$$

□

Finally, the statement follows by combining the two claims about the extractor $\mathcal{E}$. □

4.4 Reducing the Number of Garbage Commitments

The approach in Section 4.3 requires us to commit to λ additional polynomials g_i. Here, we consider a special case when $\sigma := \sigma_{-1}$[13] and show how to reduce this number by a factor of two for free. In particular, will use the following property of σ_{-1}.

Lemma 4.7. *Define the σ_{-1}-trace map $\mathsf{Tr} : \mathcal{R}_q \mapsto \mathcal{R}_q$ as*

$$\mathsf{Tr}(x) = 2^{-1}\left(x + \sigma_{-1}(x)\right).$$

Then for any $a, b \in \mathcal{R}_q$, the polynomial $y = \mathsf{Tr}\left(a\right) + X^{d/2}\mathsf{Tr}\left(b\right)$ satisfies:

$$y_0 = a_0 \text{ and } y_{d/2} = b_0.$$

Proof. We first observe that for any $c \in \mathcal{R}_q$ such that $\sigma_{-1}(c) = c$ we have $c_{d/2} = 0$. Indeed, if we compare the $d/2$-th coefficient of c and $\sigma_{-1}(c)$, we get $c_{d/2} = -c_{d/2}$ and thus $c_{d/2} = 0$.

Let $a' = \mathsf{Tr}(a)$ and $b' = \mathsf{Tr}(b)$. Clearly, a', b' are stable under the σ_{-1} automorphism and hence we have $a'_{d/2} = b'_{d/2} = 0$. Also, by construction $a'_0 = a_0$ and $b'_0 = b_0$. Therefore, $y_0 = a'_0 - b'_{d/2} = a'_0 = a_0$. Similarly, $y_{d/2} = a'_{d/2} + b'_0 = b_0$. □

For simplicity, suppose that λ is even. The strategy here is to consider each pair $(a^{(j)}, b^{(j)})_{j \in [\lambda/2]}$ defined as

$$a^{(j)} := \sum_{u=1}^{M} \gamma_{2j-1,u} \tilde{f}_u\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right)$$
$$b^{(j)} := \sum_{u=1}^{M} \gamma_{2j,u} \tilde{f}_u\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right)$$

and apply Lemma 4.7 to simultaneously prove that the constant coefficient of both elements in $\mathcal{R}_q$ is equal to zero. Concretely, we prove that the constant *and* middle coefficient of each

$$\mathsf{Tr}\left(a^{(j)}\right) + X^{d/2}\mathsf{Tr}\left(b^{(j)}\right) \in \mathcal{R}_q$$

is equal to zero.

Similarly as before, we first generate $\lambda/2$ random masking polynomials $\mathbf{g} = (g_1, \ldots, g_{\lambda/2}) \leftarrow \{x \in \mathcal{R}_q : x_0 = x_{d/2} = 0\}^{\lambda/2}$. Then, given a challenge matrix $\mathit{\Gamma} = (\gamma_{i,j}) \leftarrow \mathbb{Z}_q^{\lambda \times M}$, we construct $a^{(j)}$ and $b^{(j)}$ as above and send $h_1, \ldots, h_{\lambda/2}$ defined as follows:

$$h_j = g_j + \mathsf{Tr}\left(a^{(j)}\right) + X^{d/2}\mathsf{Tr}\left(b^{(j)}\right) \text{ for } j \in [\lambda/2]. \tag{40}$$

The verifier then checks whether the constant and middle coefficient of each h_j is equal to zero.

Finally, we need to prove that all $h_1, \ldots, h_{\lambda/2}$ are well-formed. As before, our goal will be to define $\lambda/2$ $2(m_1 + \ell + \lambda/2)$-variate quadratic functions $f_{N+1}, \ldots, f_{n+\lambda/2} : \mathcal{R}_q^{2(m_1+\ell+\lambda/2)} \to \mathcal{R}_q$ such that (40) holds if and only if

$$f_{N+j}\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m} \parallel \mathbf{g}))_{i\in[k]}\right) = 0 \text{ for } j \in [\lambda/2].$$

[13] Thus its degree k is equal to 2.

First, we observe that:

$$\sigma(\mathbf{s}) = \sigma\left(\begin{bmatrix}\mathbf{s}_1\\ \sigma(\mathbf{s}_1)\\ \mathbf{m}\\ \sigma(\mathbf{m})\end{bmatrix}\right) = \begin{bmatrix}\sigma(\mathbf{s}_1)\\ \mathbf{s}_1\\ \sigma(\mathbf{m})\\ \mathbf{m}\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \mathbf{I}_{km_1} & \mathbf{0} & \mathbf{0}\\ \mathbf{I}_{km_1} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}_{k\ell}\\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{k\ell} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{s}_1\\ \sigma(\mathbf{s}_1)\\ \mathbf{m}\\ \sigma(\mathbf{m})\end{bmatrix} = \mathbf{U}\mathbf{s}$$

where $\mathbf{U} \in \mathcal{R}_q^{2(m_1+\ell)\times 2(m_1+\ell)}$ is the matrix defined above. Hence, we have the following lemma.

Lemma 4.8. *Let $\mathbf{s}_1 \in \mathcal{R}_q^{m_1}, \mathbf{m} \in \mathcal{R}_q^{\ell}$ and set $\mathbf{s} := (\mathbf{s}_1, \sigma(\mathbf{s}_1), \mathbf{m}, \sigma(\mathbf{m}))$. For any $2(m_1 + \ell)$-variate quadratic function $f : \mathcal{R}_q^{2(m_1+\ell)} \to \mathcal{R}_q$ of the form $f(\mathbf{x}) = \mathbf{x}^T\mathbf{R}_2\mathbf{x} + \mathbf{r}_1^T\mathbf{x} + r_0$, define $\mathsf{Tr}(f)$ to be the quadratic function*

$$\mathsf{Tr}(f)(\mathbf{x}) := \mathbf{x}^T\left(\frac{\mathbf{R}_2 + \mathbf{U}^T\sigma(\mathbf{R}_2)\mathbf{U}}{2}\right)\mathbf{x} + \left(\frac{\mathbf{r}_1^T + \sigma(\mathbf{r}_1^T)\mathbf{U}}{2}\right)\mathbf{x} + \left(\frac{r_0 + \sigma(r_0)}{2}\right).$$

Then, we have $\mathsf{Tr}(f)(\mathbf{s}) = \mathsf{Tr}\left(f(\mathbf{s})\right)$.

Proof. We compute $\mathsf{Tr}\left(f(\mathbf{s})\right)$ from the definition of trace in Lemma 4.7:

$$\begin{aligned}\mathsf{Tr}\left(f(\mathbf{s})\right) &= \frac{f(\mathbf{s}) + \sigma(f(\mathbf{s}))}{2}\\ &= \frac{\mathbf{s}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\mathbf{s} + r_0}{2} + \frac{\sigma(\mathbf{s}^T)\sigma(\mathbf{R}_2)\sigma(\mathbf{s}) + \sigma(\mathbf{r}_1^T)\sigma(\mathbf{s}) + \sigma(r_0)}{2}\\ &= \frac{\mathbf{s}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\mathbf{s} + r_0}{2} + \frac{\mathbf{s}^T\mathbf{U}\sigma(\mathbf{R}_2)\mathbf{U}\mathbf{s} + \sigma(\mathbf{r}_1^T)\mathbf{U}\mathbf{s} + \sigma(r_0)}{2}\\ &= \mathsf{Tr}(f)(\mathbf{s}).\end{aligned}$$

Here, we used the observation that $\sigma(\mathbf{s}) = \mathbf{U}\mathbf{s}$. □

Let $\mathbf{x}_1 \in \mathcal{R}_q^{km_1}$, $\mathbf{x}_2 = (\mathbf{x}_{2,1}, \mathbf{x}_{2,2}) \in \mathcal{R}_q^{2(\ell+\lambda)}$. Denote

$$\mathbf{x}_{2,1} = \left(\mathbf{x}_{2,1}^{(m)}, x_{1,1}^{(g)}, \ldots, x_{1,\lambda/2}^{(g)}\right) \text{ and } \mathbf{x}_{2,2} = \left(\mathbf{x}_{2,2}^{(m)}, x_{2,1}^{(g)}, \ldots, x_{2,\lambda/2}^{(g)}\right)$$

and set $\mathbf{x}_2^{(m)} := \left(\mathbf{x}_{2,1}^{(m)}, \mathbf{x}_{2,2}^{(m)}\right)$. Then, define

$$\begin{aligned}f_{N+j}(\mathbf{x}_1, \mathbf{x}_2) := x_{1,j}^{(g)} &+ \mathsf{Tr}\left(\sum_{u=1}^{M}\gamma_{2j-1,u}\tilde{f}_u\right)\left(\mathbf{x}_1, \mathbf{x}_2^{(m)}\right)\\ &+ X^{d/2}\mathsf{Tr}\left(\sum_{u=1}^{M}\gamma_{2j}\tilde{f}_u\right)\left(\mathbf{x}_1, \mathbf{x}_2^{(m)}\right) - h_j.\end{aligned} \tag{41}$$

Then, by Lemma 4.8 we have

$$f_{N+j}\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m} \parallel \mathbf{g}))_{i\in[k]}\right) = 0 \text{ for } j \in [\lambda/2]$$

if and only if Equation 40 holds.

As before, in order to prove (34), we define quadratic functions $f_1, \ldots, f_N : \mathcal{R}_q^{2(m_1+\ell+\lambda/2)} \to \mathcal{R}_q$ as:

$$f_j(\mathbf{x}_1, \mathbf{x}_2) := f_j\left(\mathbf{x}_1, \mathbf{x}_2^{(m)}\right).$$

Finally, we run $\mathit{\Pi}_{\mathsf{many}}^{(2)}\left((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}, \mathbf{g}), \sigma, (f_j)_{j\in[N+\lambda/2]}\right)$ from Fig. 7.

5 Applications to Proving Norm Bounds

In this section we provide examples of compound zero-knowledge proofs for various statements based on the protocol in Figure 8. This protocol defined in the previous section proves simultaneously quadratic relations and that the constant coefficient of evaluations of some quadratic functions are 0. We only commit (via ABDLOP) to the message $(\mathbf{s}_1, \mathbf{m})$, but notice that the proven relations may also take as input some automorphisms of the message. We focus on one specific automorphism to instantiate the general framework of Section 4, that is $\sigma := \sigma_{-1}$ using notations from Lemma 2.5. With this choice of automorphism, Lemma 2.4 claims that T allows us to prove inner products modulo q via Figure 8.

In the first subsection, we describe a protocol that proves approximate shortness of the commitment in the Euclidean norm. In the next subsection, we describe a protocol that encompasses a variety of useful statements in lattice-based cryptography that is tailored for applications to cryptography via a single instantiation. In particular, this protocol allows to prove an upper bound on the ℓ_2-norm of a commitment with no tightness loss. In the third subsection, we detail an optimization related to the general protocol described in the second subsection. Finally in the last subsection, we describe the changes in a particular instantiation of the general protocol.

5.1 Approximate Range Proof

We first describe at a high-level a protocol to prove that a vector $\mathbf{s} = (\mathbf{s}_1, \mathbf{m})$ committed to via ABDLOP is such that $\|\mathbf{s}\| \leqslant B$ for some bound B. The bound we can prove with this method is looser than the actual bound on the norm of $\mathbf{s}$, but the counterpart is that the proof is fairly cheap. We will use this protocol to show that when $\mathbf{s}$ satisfies some relation over $\mathbb{Z}_q$ and $\|\mathbf{s}\| \leqslant B$ for small enough B, then this relation holds over $\mathbb{Z}$. The technique is inspired by [GHL21], itself reusing a technique from the ℓ_∞ approximate range proof of [LNS21a] adapted to the Euclidean norm. For the sake of simplicity, we assume that the prover wants to give a proof that his commitment $\mathbf{s} = (\mathbf{s}_1, \mathbf{m})$ satisfies $\|\mathbf{s}\| \leqslant B$. The more general statement $\|\mathbf{D}\mathbf{s} - \mathbf{e}\| \leqslant B$ for some matrix $\mathbf{D}$ and vector $\mathbf{e}$ can also be proven using the same strategy as detailed in the next subsection in Figure 10.

Description of the strategy. The foundation for this protocol is Lemma 2.9. In a nutshell, this Lemma says that for some distribution on the matrix R, the random projection $R\vec{s}$ of $\mathbf{s}$ has approximately the same norm as $\mathbf{s}$. This way, we have the opportunity to shrink a potentially very long vector $\mathbf{s}$ to a much shorter one (e.g length 256) with approximately the same norm. This projection is a $\mathbb{Z}_q$-linear map with respect to $\mathbf{s}$, which the prover can mask (which entails a slack in the bound we can prove with this method), then send and prove well-formedness of the mask to the verifier.

The matrix R is a challenge sent by the verifier, and the prover shall prove that $R\vec{s}$ has small norm so the verifier concludes that so does $\vec{s}$. The problem with this method is that for zero-knowledge, the prover cannot reveal the full vector $R\vec{s}$. Instead of revealing this vector, the prover commits to a Gaussian mask $\mathbf{y}$ of standard deviation $\mathfrak{s}_3$ for the projection before receiving R. He then applies rejection sampling on the masked projection $\vec{z} := \vec{y} + R\vec{s}$, and computes a zero-knowledge proof of the well-formedness of $\vec{z}$. The statement to be proven is captured by Figure 7, and thanks to the rejection sampling step, the $\vec{z}$ can be revealed to the verifier without leaking information on $\vec{s}$. If the well-formedness proof of $\vec{z}$ checks and $\|\vec{z}\|$ is small, then it is a matter of parameters for Lemma 2.9 to convince the verifier that $\vec{s}$ has small norm.

Bimodal rejection optimization. This mask $\vec{z}$ of $R\vec{s}$ is suited to the use of the bimodal trick to reduce the standard deviation $\mathfrak{s}_3$ of $\vec{y}$ (therefore also reduce the standard deviation of $\vec{z}$, hence the length of the proof). Explicitly, the prover choses a random sign $b \in \{-1, 1\}$, computes $\vec{z} := bR\vec{s} + \vec{y}$, and runs the rejection sampling algorithm $\mathsf{Rej}_0(\vec{z}, bR\vec{s}, \mathfrak{s}_3)$. The new distribution of $\vec{z}$ reaches the same number of repetitions as the usual rejection sampling for a lower standard deviation $\mathfrak{s}_3$, which shrinks the bit length of $\vec{z}$. The extra cost is 1) a commitment to the polynomial b and 2) a proof that $b \in \{-1, 1\}$. The commitment 1) is added to the BDLOP part, and is fairly cheap since b is a single polynomial. The zero-knowledge proof that b is a sign 2) comes almost for free as it is a $\mathbb{Z}_q$-linear proof amortized with the well-formedness proof of $\vec{z}$.

Proving that a polynomial is a sign. To perform the bimodal rejection sampling, we need to give a zero-knowledge proof that $b \in \{-1, 1\}$. We do this in two steps:

1. We prove that b is an integer
2. We prove that $(b-1)(b+1) = 0$.

As $\mathbb{Z}_q$ is a field, it follows directly from $(b-1)(b+1) = 0$ that b indeed is a sign.

We prove that b is an integer by proving that for each non-constant mononomial of degree i: $\delta_i := X^i \in \mathcal{R}_q, 1 \leqslant i \leqslant d-1$, the inner product $\langle \delta_i, b \rangle = 0$. This inner product maps b to its i-th coefficient, and shall therefore be 0 for all positions i except for the constant coefficient. Second, $(b-1)(b+1) = 0$ is a quadratic function, which we can prove using $\Pi^{(2)}_{\mathsf{eval}}$ as well. The instantation of $\Pi^{(2)}_{\mathsf{eval}}$ is detailed in the next paragraph.

Instantiation of $\Pi^{(2)}_{\mathsf{eval}}$. After 2 rounds, the proof reduces to one amortized zero-knowledge proof for quadratic functions and evaluations. First, the well formedness of the mask $\mathbf{z}$ of the projection $R\vec{s}$, then the proof that b is a sign. For each of the 256 rows of $\mathbf{z}$, we define a function F_i, and for each of the $d-1$ vectors δ_j, we define a function G_j.

$$\forall 1 \leqslant i \leqslant 256,\ F_i(\mathbf{s}, \mathbf{y}, b) = z_i - \mathsf{T}(b\vec{r_i}, \vec{s}) - y_i$$
$$\forall 1 \leqslant j \leqslant d-1,\ G_j(b) = \mathsf{T}(\vec{\delta_j}, \vec{b}),$$

where $\vec{r_i} \in \mathbb{Z}_q^{d(m_1+\ell)}$ is the i-th row of R. Finally, to prove that $b \in \{-1, 1\}$, we use the functions G_j's defined above and the quadratic function $f(b) = (b-1)(b+1)$. For clarity, we define

$$\Psi = (F_1, \ldots, F_{256}, G_1, \ldots, G_{d-1}). \tag{42}$$

Proposition 5.1. *Consider an ABDLOP commitment of messages $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$ with randomness $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$ satisfying $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A,\ \mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t}_B$. Assume that*

$$\mathfrak{s}_3 \geqslant \gamma\sqrt{337}\beta,\ t \geqslant 1.64,\ q \geqslant 41(m_1+\ell)2\sqrt{256/26}t\gamma\sqrt{337}\beta.$$

Then the protocol described on Figure 9 is a zero-knowledge proof for the statement $\|\mathbf{s}\| \leqslant 2\sqrt{256/26}t\gamma\sqrt{337}\beta (\leqslant 189\gamma\beta)$. More precisely, let P_{eval} be the success probability of a honest prover in Π, $P_{\mathcal{E}'}, T_{\mathcal{E}'}$ be respectively the success probability and the run time of the extractor $\mathcal{E}'$ from Theorem 4.5 running on Π.

For correctness, if the prover and the verifier follow the protocol honestly, then the verifier shall accept with probability $\approx P_{\mathsf{eval}} \exp(-\frac{1}{2\gamma^2})$.

For soundness, let $\mathcal{P}$ be a probabilistic prover with success probability $\epsilon \geqslant \frac{2}{|\mathcal{C}|} - q^{-d/2} - q^{-\lambda} - 2^{-128}$. There exists an extractor $\mathcal{E}'$ that with rewindable black-box access to $\mathcal{P}$ either breaks the binding of the ABDLOP commitment, or finds a valid opening $(\bar{\mathbf{s}_2}, \bar{\mathbf{s}_1}, \bar{\mathbf{y}}, \bar{\mathbf{m}}, \bar{b}, \bar{c})$ to the commitment $(\mathbf{t}, t_b, \mathbf{t}_y)$ with $\|(\bar{\mathbf{s}}_1, \bar{\mathbf{m}})\| \leqslant 2\sqrt{\frac{256}{26}}t\gamma\sqrt{337}\beta$, with probability $P_{\mathcal{E}'}(1 - 2^{-128})$ in expected time $2T_{\mathcal{E}'}$.

For commit-and-prove simulatability, there exists a simulator $\mathcal{S}$ that, without access to private information $\mathbf{s}_1, \mathbf{m}$, outputs a simulation of a commitment $(\mathbf{t}, \mathbf{t}_y, t_b)$ along with a non-aborting transcript of the protocol between prover $\mathcal{P}$ and verifier $\mathcal{V}$ such that for every algorithm $\mathcal{A}$ that has advantage ε in distinguishing the simulated commitment and transcript from the actual commitment and transcript, whenever the prover does not abort, there is an algorithm $\mathcal{A}'$ with the same running time that has advantage $\varepsilon/2 - 2^{-100}$ in distinguishing Extended-$\mathsf{MLWE}_{n+\ell+\lambda+256/d+2, m_2-n-\ell-\lambda-256/d-2, \chi, \mathcal{C}, \mathfrak{s}_2}$.

Proof. We only detail correctness and soundness, commit-and-prove simulatability follows directly from the same property from Figure 8, the rejection sampling and the hiding property of ABDLOP.
Correctness. Let $i \in [256]$. If the prover and verifier follow the protocol honestly, we have:

$$F_i(\mathbf{s}, \mathbf{y}, b) = z_i - \mathsf{T}(b\vec{r_i}, \vec{s}) - y_i \tag{43}$$
$$\widetilde{F}_i(\mathbf{s}, \mathbf{y}, b) = z_i - b\langle \vec{r_i}, \vec{s} \rangle - \vec{y_i} \tag{44}$$
$$= b\langle \vec{r_i}, \vec{s} \rangle - b\langle \vec{r_i}, \vec{s} \rangle = 0. \tag{45}$$

Public information:
Commitment $\mathbf{t} = (\mathbf{t}_A, \mathbf{t}_B) \in \mathcal{R}_q^{n+\ell}, \mathbf{A}_1 \in \mathcal{R}_q^{n\times(m_1+v_e)}, \mathbf{A}_2 \in \mathcal{R}_q^{n\times m_2}, \mathbf{B} \in \mathcal{R}_q^{\ell\times m_2}$ such that
$\mathbf{t} = \begin{bmatrix}\mathbf{t}_A \\ \mathbf{t}_B\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1 \\ \mathbf{0}\end{bmatrix} \mathbf{s}_1 + \begin{bmatrix}\mathbf{A}_2 \\ \mathbf{B}\end{bmatrix} \mathbf{s}_2 + \begin{bmatrix}\mathbf{0} \\ \mathbf{m}\end{bmatrix}$. $\mathbf{B}_2 \in \mathcal{R}_q^{256/d\times m_2},\ \mathbf{b}_1 \in \mathcal{R}_q^{m_2}$.
Gaussian mask standard deviation $\mathfrak{s}_3 := \gamma\sqrt{337}\beta$, acceptance coefficient $t = 1.64$.
Private information: $\mathbf{s} = (\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$ such that $\|\mathbf{s}\| \leqslant \beta$, randomness $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$.

Prover		Verifier
Sample $b \leftarrow \{-1, 1\} \subset \mathcal{R}_q$ Sample $\mathbf{y} \leftarrow D_{\mathfrak{s}_3}^{256/d}$ $t_b = \mathbf{b}_1^T \mathbf{s}_2 + b$ $\mathbf{t}_y = \mathbf{B}_2\mathbf{s}_2 + \mathbf{y}$		
	$\xrightarrow{t_b, \mathbf{t}_y}$	
		$R \leftarrow \mathsf{Bin}_1^{256\times(m_1+\ell)}$
	$\xleftarrow{R}$	
$\vec{z} := bR\vec{s} + \vec{y}$ If $\mathsf{Rej}_0(\vec{z}, bR\vec{s}, \mathfrak{s}_3)$ Then continue, Else abort		
	$\xrightarrow{\vec{z}}$	
Run $\mathit{\Pi} := \mathit{\Pi}_{\mathsf{eval}}^{(2)}((\mathbf{s}_2, \mathbf{s}_1, (\mathbf{m}, b)), \sigma, f, \mathit{\Psi})$		
		Accept iff $\mathit{\Pi}$ verifies and $\|\vec{z}\| \leqslant t\sqrt{256}\mathfrak{s}_3$

Fig. 9: Commit-and-prove protocol for the messages $\mathbf{s} = (\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$, randomness $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$ and $\bar{c} \in \bar{\mathcal{C}}$ which satisfy: $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A$, $\mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t}_B$ (ii) $\|\mathbf{s}_i\bar{c}\| \leqslant 2\mathfrak{s}_i\sqrt{2m_i d}$ for $i = 1, 2$ (where $\mathfrak{s}_i$ are used in Fig. 6) and (iii) $\|\mathbf{s}\| \leqslant 2\sqrt{\frac{256}{26}}t\gamma\sqrt{337}\beta$.

From Equation (43) to Equation (44) comes from Lemma 2.4. Equation (44) to Equation (45) is true because the prover formed $\vec{z} = R\vec{s} + \vec{y}$ correctly. Obviously, since $b \in \{-1, 1\}$, $f(b) = 0$. Again using Lemma 2.4 on the G_j's, each functions maps to a non-constant coefficient, which is 0 since in particular $b \in \mathbb{Z}_q$. We proved that the inputs of $\mathit{\Pi}$ are correct, hence with probability P_{eval}, the verifier accepts $\mathit{\Pi}$. The probability that the prover passes the rejection sampling step is given by $\exp(-\frac{1}{(2\gamma)^2})$ according to Lemma 2.14. Finally, using the tail bounds from Lemma 2.2 on $\vec{z}$, we have that $\mathbb{P}(\|\mathbf{z}\| \geqslant t\sqrt{256}\mathfrak{s}_3) \leqslant (te^{\frac{1-t^2}{2}})^{256}$, so the verifier also checks $\|\mathbf{z}\| \leqslant t\sqrt{256}\mathfrak{s}_3$ with probability at least $1 - (te^{\frac{1-t^2}{2}})^{256}$. For $t \geqslant 1.64$, we have $(te^{\frac{1-t^2}{2}})^{256} \leqslant 2^{-128}$. The success probability of the prover is at least the probability that 1) $\mathsf{Rej}_0(\vec{z}, bR\vec{s}, \mathfrak{s}_3)$ does not abort, and 2) $\mathit{\Pi}$ does not abort and the verifier accepts the proof, and 3) the norm verification passes. Therefore the verifier accepts with probability at least $P_{\mathsf{eval}} \exp(-\frac{1}{2\gamma^2})(1 - 2^{-128})$.

Soundness. We let $\mathcal{E}$ be the extractor for this zero-knowledge proof and $\mathcal{E}'$ be the extractor from $\mathit{\Pi}_{\mathsf{eval}}^{(2)}$. The extractor $\mathcal{E}$ proceeds as follows:

1. Run the prover until the third round on a honestly generated challenge R then run $\mathcal{E}'$. If $\mathcal{E}$ does not obtain a valid opening $\bar{\mathbf{s}}, \bar{\mathbf{y}}, \bar{b}$ satisfying the relations given by Equation (42), then abort otherwise continue.
2. Rewind the prover until the third round, send a honestly and freshly generated challenge R' and then run $\mathcal{E}'$ until $\mathcal{E}$ successfully obtains a valid opening $\bar{\mathbf{s}}', \bar{\mathbf{y}}', \bar{b}'$ satisfying the relations given by Equation (42).

With this extraction strategy, the expected run time of $\mathcal{E}$ is 2 times the expected run time of $\mathcal{E}'$ and $\mathcal{E}$ has the same success probability as $\mathcal{E}'$, see Lemma 4.3 for justifications.

Notice that since $\epsilon \geqslant \frac{2}{|\mathcal{C}|} + q^{-d/2} + q^{-\lambda}$, in particular the success probability of the prover at producing a valid Π in the last step is also at least $\frac{2}{|\mathcal{C}|} + q^{-d/2} + q^{-\lambda}$ and therefore $P_{\mathcal{E}'} \geqslant \epsilon - \frac{2}{|\mathcal{C}|} - q^{-d/2} - q^{-\lambda} \geqslant 2^{-128}$.

If $\mathcal{E}$ found two valid openings to different messages $(\bar{\mathbf{s}}, \bar{\mathbf{y}}) \neq (\bar{\mathbf{s}}', \bar{\mathbf{y}}')$, then $\mathcal{E}$ breaks the binding property of the commitment scheme. We now consider the second possible outcome, that is the extractor $\mathcal{E}$ finds $(\bar{\mathbf{s}}_2, \bar{\mathbf{s}}_1, \bar{\mathbf{y}}, \bar{\mathbf{m}}, \bar{b}) \in \mathcal{R}_q^{m_2+m_1+256/d+\ell+1}$ and $\bar{c} \in \mathcal{R}_q^{\times}$ such that $(\bar{\mathbf{s}}_1, \bar{\mathbf{y}}, \bar{\mathbf{m}}, \bar{b})$ are valid ABDLOP messages for the randomness $\bar{\mathbf{s}}_2$ and

- $f(\bar{b}) = 0$
- For $1 \leqslant i \leqslant 256$, $\widetilde{F}_i(\bar{\mathbf{s}}, \bar{\mathbf{y}}, \bar{b}) = 0$
- For $1 \leqslant j \leqslant d$, $\widetilde{G}_j(\bar{b}) = 0$, and similarly for the second transcript.

We define $\bar{\mathbf{s}} = (\bar{\mathbf{s}}_1, \bar{\mathbf{m}})$. Plugging together the fact that all the $\widetilde{F}_i(\bar{\mathbf{s}}, \bar{\mathbf{y}}, \bar{b})$ are 0 and Equation (44), we have that $\vec{z}$ is of the correct form, that is $\vec{z} = R\vec{\bar{s}} + \vec{\bar{y}}$. The latter also holds for $\vec{z} = R'\vec{\bar{s}}' + \vec{\bar{y}}$, but in this case R' and $(\bar{\mathbf{s}}, \bar{\mathbf{y}}) = (\bar{\mathbf{s}}', \bar{\mathbf{y}}')$ are independent. Under Lemma 2.4, $\widetilde{G}_j(\bar{b}) = 0$ yields that every non-constant coefficient of $\bar{b}$ is 0, hence $b \in \mathbb{Z}_q$. Since $\mathbb{Z}_q$ is an integral domain, $f(\bar{b}) = (\bar{b} - 1)(\bar{b} + 1) = 0$ ensures that $\bar{b}$ is a sign.

From the norm verification, we have that

$$\|\mathbf{z}\| \leqslant t\sqrt{256}\mathfrak{s}_3 \tag{46}$$

$$\|R'\vec{\bar{s}} + \vec{\bar{y}} \mod q\| \leqslant t\sqrt{256}\mathfrak{s}_3 \tag{47}$$

$$\|R'\vec{\bar{s}} + \vec{\bar{y}} \mod q\| \leqslant t\sqrt{256}\gamma\sqrt{337}\beta \tag{48}$$

$$\|R'\vec{\bar{s}} + \vec{\bar{y}} \mod q\| \leqslant \frac{1}{2}\sqrt{26}\left(2\sqrt{\frac{256}{26}}t\gamma\sqrt{337}\beta\right), \tag{49}$$

where Equation (46) to Equation (47) follows from the proven well-formedness of $\mathbf{z}$, Equation (47) to Equation (48) follows from the assumption on $\mathfrak{s}_3$ and Equation (48) to Equation (49) is simply reformulating the upper bound so it fits Lemma 2.9. We now apply Lemma 2.9, which is possible since $(\bar{\mathbf{s}}, \bar{\mathbf{y}}) = (\bar{\mathbf{s}}', \bar{\mathbf{y}}')$ and R' are independent. Under the condition that $2\sqrt{\frac{256}{26}}t\gamma\sqrt{337}\beta \leqslant \frac{q}{41d(m_1+\ell)}$, we have that if $\|\bar{\mathbf{s}}\| \geqslant 2\sqrt{\frac{256}{26}}t\gamma\sqrt{337}\beta$, then the probability over the randomness of the challenge R that Equation (46) is less than 2^{-128}. By contraposition, with overwhelming probability $1 - 2^{-128}$, we have $\|\bar{\mathbf{s}}\| \leqslant \sqrt{\frac{256}{26}}t\gamma\sqrt{337}\beta$, which completes the soundness proof. □

5.2 General Protocol With Exact ℓ_2-Norm Proof

In this subsection, we describe a general protocol to prove various quadratic relations on $\mathbf{s} = (\mathbf{s}_1, \mathbf{m}, \sigma(\mathbf{s}_1), \sigma(\mathbf{m}))$, where $(\mathbf{s}_1, \mathbf{m})$ is the message of an ABDLOP commitment. The statements proven in this protocol are such that the applications to cryptographic primitives detailed in Section 6 result in a single instantiation of this protocol. We highlight that among the relations this protocol proves is an exact norm proof $\|\mathbf{s}\| \leqslant \beta$, where β is tight.

In a nutshell, we prove simultaneously quadratic relations over $\mathcal{R}_q$, quadratic relations over $\mathbb{Z}_q$, approximate bound on the infinity norm, exact bound on the ℓ_2 norm and finally that a vector is binary. All the later statements are gathered in this single protocol as they rely on proving inner products, which is possible to prove efficiently using Figure 8. Explicitly, we define public parameters:

- Quadratic functions for $i \in [\rho]$ $f_i : \mathcal{R}_q^{2(m_1+\ell)} \longrightarrow \mathcal{R}_q$
- Evaluation functions for $i \in [\rho_{\mathsf{eval}}]$ $F_i : \mathcal{R}_q^{2(m_1+\ell)} \longrightarrow \mathcal{R}_q$
- For $i \in [\upsilon_d]$, $\mathbf{D}_i \in \mathcal{R}_q^{k_i \times 2(m_1+\ell)}$, $\mathbf{u}_i \in \mathcal{R}_q^{k_i}$
- For $i \in [\upsilon_e]$, $\mathbf{E}_i \in \mathcal{R}_q^{p_i \times 2(m_1+\ell)}$, $\mathbf{v}_i \in \mathcal{R}_q^{p_i}$

- Bounds $(\beta_i^{(d)})_{i\in[\upsilon_d]}, (\beta_i^{(e)})_{i\in[\upsilon_e]}$.
- Matrix $\mathbf{E}_{\mathsf{bin}} \in \mathcal{R}_q^{k_{\mathsf{bin}}\times 2(m_1+\ell)}$ and vector $\mathbf{v}_{\mathsf{bin}} \in \mathcal{R}_q^{k_{\mathsf{bin}}}$.

The general statement proven in Figure 10 includes the knowledge of a vector $\mathbf{s} = (\mathbf{s}_1, \sigma(\mathbf{s}_1), \mathbf{m}, \sigma(\mathbf{m})) \in \mathcal{R}_q^{2m_1} \times \mathcal{R}_q^{2\ell}$ such that

$$\forall 1 \leqslant i \leqslant \rho,\ f_i(\mathbf{s}) = 0 \tag{50}$$

$$\forall 1 \leqslant i \leqslant \rho_{\mathsf{eval}},\ \widetilde{F}_i(\mathbf{s}) = 0 \tag{51}$$

$$\forall 1 \leqslant i \leqslant \upsilon_d,\ \|\mathbf{D}_i\mathbf{s} - \mathbf{v}_i\|_\infty \leqslant \beta_i^{(d)}. \tag{52}$$

$$\forall 1 \leqslant i \leqslant \upsilon_e,\ \|\mathbf{E}_i\mathbf{s} - \mathbf{v}_i\| \leqslant \beta_i^{(e)}. \tag{53}$$

$$\mathbf{E}_{\mathsf{bin}}\mathbf{s} - \mathbf{v}_{\mathsf{bin}} \in \{0,1\}^{dk_{\mathsf{bin}}}. \tag{54}$$

The functions f_i are quadratic relations, and the functions F_i are also quadratic relations but for which we only prove the constant coefficient. The matrices $\mathbf{D}_i$ and vectors $\mathbf{u}_i$ are such that $\|\mathbf{D}_i\mathbf{s} - \mathbf{u}_i\|_\infty$ is small, and we prove the latter with a looser bound $\beta_i^{(d)}$ than the actual bound on $\|\mathbf{D}_i\mathbf{s} - \mathbf{u}_i\|_\infty$. The matrices $\mathbf{E}_i$ and vectors $\mathbf{v}_i$ are such that $\|\mathbf{E}_i\mathbf{s} - \mathbf{v}_i\| \leqslant \beta_i^{(e)}$, which we prove exactly in the sense that the proven bound is $\beta_i^{(e)}$. Finally, the matrix $\mathbf{E}_{\mathsf{bin}}$ and vector $\mathbf{v}_{\mathsf{bin}}$ are such that $\mathbf{E}_{\mathsf{bin}}\mathbf{s} - \mathbf{v}_{\mathsf{bin}}$ is binary, which we prove.

General strategy. Suppose we have an ABDLOP commitment to a vector $(\mathbf{s}_1, \mathbf{m})$ and we want to prove Equations (50) to (54) on $\mathbf{s} = (\mathbf{s}_1, \sigma(\mathbf{s}_1), \mathbf{m}, \sigma(\mathbf{m}))$. To prove the quadratic relations and evaluations Equations (50) and (51), we simply pass on the functions to the input of the instantiation of Figure 8 that we will need later anyway. To prove Equation (52), we use the technique from Figure 9 with the ℓ_∞-norm instead. The proof of Equation (54) is detailed in the paragraph below. We now focus on Equation (53). Remind that one can use $\Pi_{\mathsf{eval}}^{(2)}$ to give a zero-knowledge proof that the inner product of two commitments $\bmod q$ is some public constant. Therefore we can prove that $\langle \mathbf{E}_i\mathbf{s} - \mathbf{v}_i, \mathbf{E}_i\mathbf{s} - \mathbf{v}_i\rangle \bmod q$ is some constant. We use the approximate range proof from Figure 9 to prove that the computation of $\langle \mathbf{E}_i\mathbf{s} - \mathbf{v}_i, \mathbf{E}_i\mathbf{s} - \mathbf{v}_i\rangle$ does not induce a wraparound modulo q, and therefore also holds over $\mathbb{Z}$.

Remember that we do not want to give away the exact norm of $\mathbf{E}_i\mathbf{s} - \mathbf{v}_i$, but rather prove that it is lower than some bound. To circumvent this, we prove that the difference between the bound and the norm is a positive integer. Explicitly, we prove that $(\beta_i^{(e)})^2 - \langle \mathbf{E}_i\mathbf{s} - \mathbf{v}_i, \mathbf{E}_i\mathbf{s} - \mathbf{v}_i\rangle$ can be written with a binary representation $\vec{x}_i$ of length $2\log(\beta_i^{(e)}) \leqslant d$. Overall, proving exact norm reduces to the combination of proofs for the relations between $\mathbf{s}$ and $(x_i)_{i\in[\upsilon_e]}$, and a proof that each x_i is binary. Notice that both proofs are over $\mathbb{Z}$ rather than $\mathbb{Z}_q$, so we need a third proof to lift the relations we can only prove directly over $\mathbb{Z}_q$ to $\mathbb{Z}$.

Proving that a vector is binary. We detail a simple technique to prove that a vector has binary coefficients. This proof is enabled by the efficiency of proving inner product relations as it relies on the following fact.

Lemma 5.2. *Let $n \in \mathbb{N}$ and $\vec{x} \in \mathbb{Z}^n$. If $\langle \vec{x}, \vec{x} - \vec{1}_n\rangle = 0$, then $\vec{x} \in \{0,1\}^n$.*

Proof. Let $x = (x_1\ \dots\ x_n) \in \mathbb{Z}^n$. Every term in the inner product $\langle \vec{x}, \vec{x} - \vec{1}_n\rangle$ is of the form $x_i(x_i - 1)$. Moreover, the map $a \longmapsto a(a-1)$ is a positive over the integers, therefore $\langle \vec{x}, \vec{x} - \vec{1}_n\rangle \geqslant 0$ with equality if and only if every term is 0, i.e $\vec{x}$ is binary. □

In other words, it is enough to prove $\langle \vec{x}, \vec{x} - \vec{1}_n\rangle = 0$ to infer that $\vec{x}$ is a binary vector. In our protocol, we prove that the υ_e vectors $\vec{x}_i$ and the vector $\mathbf{E}_{\mathsf{bin}}\mathbf{s} - \mathbf{v}_{\mathsf{bin}}$ are binary, which we do in two steps:

1. We prove that $\langle (\vec{x}||\mathbf{E}_{\mathsf{bin}}\mathbf{s} - \mathbf{v}_{\mathsf{bin}}), (\vec{x}||\mathbf{E}_{\mathsf{bin}}\mathbf{s} - \mathbf{v}_{\mathsf{bin}}) - \vec{1}_n\rangle = 0 \bmod q$, which is a direct application of Figure 7.
2. We prove that $\|(\vec{x}||\mathbf{E}_{\mathsf{bin}}\mathbf{s} - \mathbf{v}_{\mathsf{bin}})\| \leqslant B$ for some bound B using Figure 9.

Provided that B is such that $B^2 + \sqrt{(\upsilon_e + k_{\mathsf{bin}})d}B < q$ (which is actually very easily met for reasonable parameters), $\langle (\vec{x}||\mathbf{E}_{\mathsf{bin}}\mathbf{s} - \mathbf{v}_{\mathsf{bin}}), (\vec{x}||\mathbf{E}_{\mathsf{bin}}\mathbf{s} - \mathbf{v}_{\mathsf{bin}}) - \vec{1}_{(\upsilon_e + k_{\mathsf{bin}})d} \rangle = 0 \mod q$ holds over the integers, and Lemma 5.2 yields that $\vec{x}$ and $\mathbf{E}_{\mathsf{bin}}\mathbf{s} - \mathbf{v}_{\mathsf{bin}}$ are binary.

Specifications and instantiation. To begin with, the prover appends a commitment to the binary representation vector $\mathbf{x} = (x_1 || \dots || x_{\upsilon_e})$ in the Ajtai part of the commitment to $(\mathbf{s}_1, \mathbf{m})$[14]. This vector is the concatenation of the binary decompositions $\vec{x}_i$ of $(\beta_i^{(e)})^2 - \|\mathbf{E}_i\mathbf{s} - \mathbf{v}_i\|^2$. We write $\mathbf{x}'$ the concatenation of $\mathbf{x}$ and $\mathbf{E}_{\mathsf{bin}}\mathbf{s} - \mathbf{v}_{\mathsf{bin}}$. The verifier samples two approximate range proof challenge matrices $R^{(d)}, R^{(e)}$. The first one $R^{(d)}$ is used for the approximate norm proofs Equation (52) and the second one is used for the exact norm proofs Equation (53). He sends both matrices to the prover. Finally, the prover computes a zero-knowledge proof for the following statements:

$$\forall 1 \leqslant i \leqslant \rho,\ f_i(\mathbf{s}) = 0 \tag{55}$$

$$\forall 1 \leqslant i \leqslant \rho_{\mathsf{eval}},\ \widetilde{F}_i(\mathbf{s}) = 0 \tag{56}$$

$$\forall 1 \leqslant i \leqslant \upsilon_d,\ \|\mathbf{D}_i\mathbf{s} - \mathbf{v}_i\|_\infty \leqslant \beta_i^{(d)} \tag{57}$$

$$\langle \mathbf{x}', \mathbf{x}' - \mathbf{1}_{(\upsilon_e + k_{\mathsf{bin}})d} \rangle = 0 \mod q \tag{58}$$

$$\forall 1 \leqslant i \leqslant \upsilon_e,\ \langle \mathbf{E}_i\mathbf{s} - \mathbf{v}_i, \mathbf{E}_i\mathbf{s} - \mathbf{v}_i \rangle + \begin{pmatrix} 1 & 2 & \dots & 2^{2\log(\beta_i^{(e)})} & 0 & \dots & 0 \end{pmatrix} \vec{x}_i = (\beta_i^{(e)})^2 \mod q \tag{59}$$

$$(\|\mathbf{E}_i\mathbf{s} - \mathbf{v}_i\|_\infty)_{i \in [\upsilon_e]}, \|\mathbf{x}'\| \text{ are small enough so } \textit{Equation } (58) \text{ and } \textit{Equation } (59) \text{ hold over } \mathbb{Z}. \tag{60}$$

We proceed to describe the functions in the input of $\Pi_{\mathsf{eval}}^{(2)}$. Let us first introduce some notations to make the exposition more compact: we write $\vec{p}_i = (1\ 2\ \dots\ 2^{2\log(\beta_i^{(e)})}\ 0\ \dots\ 0)$, and for $i \in [d, e]$ we write $\mathbf{r}_j^{(i)}$ the j-th row of $R^{(i)}$, $y_j^{(i)}$ (respectively $z_j^{(i)}$) the j-th coordinate of $\vec{y}^{(i)}$ (respectively $\vec{z}^{(i)}$). Remember that $\forall i \in [d],\ \delta_i = X^i$ is the unitary monomial of degree i in $\mathcal{R}_q$. We remind that $\mathbf{x}'$ is defined as the concatenation of the binary decompositions $\vec{x}_i$ and $\mathbf{E}_{\mathsf{bin}}\mathbf{s} - \mathbf{v}_{\mathsf{bin}}$. Finally, we define

$$\mathbf{e}^{(d)} = \begin{bmatrix} \mathbf{D}_1\mathbf{s} - \mathbf{u}_1 \\ \vdots \\ \mathbf{D}_{\upsilon_d}\mathbf{s} - \mathbf{u}_{\upsilon_d} \end{bmatrix},\ \mathbf{e}^{(e)} = \begin{bmatrix} \mathbf{E}_1\mathbf{s} - \mathbf{v}_1 \\ \vdots \\ \mathbf{E}_{\upsilon_e}\mathbf{s} - \mathbf{v}_{\upsilon_e} \\ \mathbf{x}' \end{bmatrix}. \tag{61}$$

We define the following functions to instantiate $\Pi_{\mathsf{eval}}^{(2)}$:

$$\forall i \in \{d, e\},\ g^{(i)}(b^{(i)}) = (b^{(i)} - 1)(b^{(i)} + 1) \tag{62}$$

$$G(\mathbf{x}') = \mathsf{T}(\mathbf{x}', \mathbf{x}' - \mathbf{1}_{(\upsilon_e + k_{\mathsf{bin}})d}) \tag{63}$$

$$\forall j \in [256],\ H_j^{(d)}(\mathbf{s}, \mathbf{y}^{(d)}, b^{(d)}) = z_j^{(d)} - \mathsf{T}(b^{(d)}\mathbf{r}_j^{(d)}, \mathbf{e}^{(d)}) - y_j^{(d)} \tag{64}$$

$$\forall j \in [256],\ H_j^{(e)}(\mathbf{x}', \mathbf{s}, \mathbf{y}^{(e)}, b^{(e)}) = z_j^{(e)} - \mathsf{T}(b^{(e)}\mathbf{r}_j^{(e)}, \mathbf{e}^{(e)}) - y_j^{(e)} \tag{65}$$

$$\forall i \in [\upsilon_e],\ I_i(\mathbf{s}, \mathbf{x}) = \mathsf{T}\left(\mathbf{E}_i\mathbf{s} - \mathbf{v}_i, \mathbf{E}_i\mathbf{s} - \mathbf{v}_i\right) + \mathsf{T}(\vec{p}_i, \vec{x}_i) - (\beta_i^{(e)})^2 \tag{66}$$

$$\forall i \in \{d, e\},\ 1 \leqslant j \leqslant d - 1,\ J_j^{(i)}(b^{(i)}) = \mathsf{T}(\delta_j, b^{(i)}) \tag{67}$$

We now pack the functions that are the input of $\Pi_{\mathsf{eval}}^{(2)}$ for more clarity. We let

$$\phi = (f_1, \dots, f_\rho, g^{(d)}, g^{(e)}) \tag{68}$$

$$\Psi = \left((F_i)_{i \in \rho_{\mathsf{eval}}}, G, (H_j^{(d)})_{j \in [256]}, (H_j^{(e)})_{j \in [256]}, (I_i)_{i \in \upsilon_e}, (J_j^{(i)})_{i \in \{d,e\}, j \in [d]} \right). \tag{69}$$

[14] Note that appending a commitment in the Ajtai part can only be done at the same time as the commitment to $\mathbf{s}_1$. If for some reason it is not possible to commit ahead of time to $\vec{x}$, one has to commit to $\vec{x}$ in the BDLOP part instead.

Public information:

Commitment $\mathbf{t} \in \mathcal{R}_q^{n+\ell}, \mathbf{A}_1 \in \mathcal{R}_q^{n\times(m_1+\upsilon_e)}, \mathbf{A}_2 \in \mathcal{R}_q^{n\times m_2}, \mathbf{B} \in \mathcal{R}_q^{\ell\times m_2}$ such that
$\mathbf{t} = \begin{bmatrix}\mathbf{A}_1\\ \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{s}_1\\ \mathbf{x}\end{bmatrix} + \begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix}\mathbf{s}_2 + \begin{bmatrix}\mathbf{0}\\ \mathbf{m}\end{bmatrix}$. $\mathbf{B}^{(d)} \in \mathcal{R}_q^{(256/d)\times m_2}, \mathbf{B}^{(e)} \in \mathcal{R}_q^{(256/d)\times m_2}$, $\mathbf{b}^{(d)} \in \mathcal{R}_q^{m_2}$, $\mathbf{b}^{(e)} \in \mathcal{R}_q^{m_2}$.
For $i \in [\rho]$, quadratic functions $f_i : \mathcal{R}_q^{2(m_1+\ell)} \longrightarrow \mathcal{R}_q$
For $i \in [\rho_{\mathsf{eval}}]$, quadratic functions $F_i : \mathcal{R}_q^{2(m_1+\ell)} \longrightarrow \mathcal{R}_q$
For $i \in [\upsilon_e]$, matrix $(\mathbf{D}_i) \in \mathcal{R}_q^{k_i\times 2(m_1+\ell)}$, vector $\mathbf{u}_i \in \mathcal{R}_q^{k_i}$, bound $\beta_i^{(d)}$
For $i \in [\upsilon_e]$, matrix $(\mathbf{E}_i) \in \mathcal{R}_q^{p_i\times 2(m_1+\ell)}$, vector $\mathbf{v}_i \in \mathcal{R}_q^{p_i}$, bound $\beta_i^{(e)}$
Matrix $(\mathbf{E}_{\mathsf{bin}}) \in \mathcal{R}_q^{k_bin\times 2(m_1+\ell)}$
Bounds $\alpha^{(d)}$, $\alpha^{(e)}$ such that $\|\mathbf{e}^{(d)}\| \leqslant \alpha^{(d)}$, $\|\mathbf{e}^{(e)}\| \leqslant \alpha^{(e)}$
Standard deviations $\mathfrak{s}^{(d)} = \gamma^{(d)}\sqrt{337}\alpha^{(d)}$, $\mathfrak{s}^{(e)} = \gamma^{(e)}\sqrt{337}\alpha^{(e)}$, acceptance coefficient $t \in \mathbb{R}$
Challenge dimensions $c^{(d)} = d\sum_{i=1}^{\upsilon_e} k_i$, $c^{(e)} = d(k_{\mathsf{bin}} + \sum_{i=1}^{\upsilon_e}(p_i+1))$
Input functions of $\Pi_{\mathsf{eval}}^{(2)}$ ϕ, Ψ defined in Equations (68) and (69).

Private information:

Randomness $\mathbf{s}_2 \leftarrow \chi^{m_2}$, message $\mathbf{s} = (\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$
such that Equations (50) to (54) hold. Binary decomposition $x_i \in \mathcal{R}_q$ of $(\beta_i^{(e)})^2 - \|\mathbf{E}_i\mathbf{s} - \mathbf{v}_i\|^2$.
Vectors $\mathbf{e}^{(d)} = (\mathbf{D}_1\mathbf{s} - \mathbf{u}_1||...||\mathbf{D}_{\upsilon_d}\mathbf{s} - \mathbf{u}_{\upsilon_d})$, $\mathbf{e}^{(e)} = (\mathbf{E}_1\mathbf{s} - \mathbf{v}_1||...||\mathbf{E}_{\upsilon_e}\mathbf{s} - \mathbf{v}_{\upsilon_e}||\mathbf{x})$.

Prover		Verifier
$b^{(d)}, b^{(e)} \leftarrow \{-1,1\} \subset \mathcal{R}_q$ $\mathbf{y}^{(d)} \leftarrow D_{\mathfrak{s}^{(d)}}^{256/d}$, $\mathbf{y}^{(e)} \leftarrow D_{\mathfrak{s}^{(e)}}^{256/d}$ $\mathbf{t}^{(d)} := \mathbf{B}^{(d)}\mathbf{s}_2 + \mathbf{y}^{(d)}$ $\mathbf{t}^{(e)} := \mathbf{B}^{(e)}\mathbf{s}_2 + \mathbf{y}^{(e)}$ $t^{(d)} := (\mathbf{b}^{(d)})^T\mathbf{s}_2 + b^{(d)}$ $t^{(e)} := (\mathbf{b}^{(e)})^T\mathbf{s}_2 + b^{(e)}$		
	$\xrightarrow{\mathbf{t}^{(d)}, t^{(d)}, \mathbf{t}^{(e)}, t^{(e)}}$	
		$R^{(d)} \leftarrow \mathsf{Bin}_1^{256\times c^{(d)}}$ $R^{(e)} \leftarrow \mathsf{Bin}_1^{256\times c^{(e)}}$
	$\xleftarrow{R^{(d)}, R^{(e)}}$	
$\vec{z}^{(d)} := b^{(d)}R^{(d)}\vec{e}^{(d)} + \vec{y}^{(d)}$ $\vec{z}^{(e)} := b^{(e)}R^{(e)}\vec{e}^{(e)} + \vec{y}^{(e)}$ If $\mathsf{Rej}_0(\vec{z}^{(d)}, b^{(d)}R^{(d)}\vec{e}^{(d)}, \mathfrak{s}^{(d)})$ and $\mathsf{Rej}_0(\vec{z}^{(e)}, b^{(e)}R^{(e)}\vec{e}^{(e)}, \mathfrak{s}^{(e)})$ Then continue, Else abort $\mathbf{s}^* := (\mathbf{s}_2,\ (\mathbf{s}_1, \mathbf{x}),\ (\mathbf{m}, \mathbf{y}^{(d)}, \mathbf{y}^{(e)}, b^{(d)}, b^{(e)}))$		
	$\xrightarrow{\vec{z}^{(d)}, \vec{z}^{(e)}}$	
Run $\Pi = \Pi_{\mathsf{eval}}^{(2)}(\mathbf{s}^*, \sigma, \phi, \Psi)$		
		Accept iff : • Π verifies • $\|\vec{z}^{(d)}\|_\infty \leqslant 14\mathfrak{s}^{(d)}$ • $\|\vec{z}^{(e)}\| \leqslant t\sqrt{256}\mathfrak{s}^{(e)}$

Fig. 10: Commit-and-prove protocol for messages $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$, randomness $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$ and $\bar{c} \in \bar{\mathcal{C}}$ which satisfy: $\mathbf{A}_1\begin{bmatrix}\mathbf{s}_1\\ \mathbf{x}\end{bmatrix} + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A$, $\mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t}_B$ (ii) $\|\mathbf{s}_i\bar{c}\| \leqslant 2\mathfrak{s}_i\sqrt{2m_id}$ for $i = 1, 2$ (where $\mathfrak{s}_i$ are used in Fig. 6) and $\mathbf{s} := (\mathbf{s}_1, \sigma(\mathbf{s}_1), \mathbf{m}, \sigma(\mathbf{m}))$ verifies Equations (50) to (54)).

Theorem 5.3. *Consider an ABDLOP commitment to a message* $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$ *with randomness* $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$ *satisfying* $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A$, $\mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t}_b$. *Let* $B^{(d)} := 28\sqrt{337}\gamma^{(d)}\alpha^{(d)}$, $B^{(e)} := 2\sqrt{\frac{256}{26}t\gamma^{(e)}}\sqrt{337}\alpha^{(e)}$, *and assume that* $t \geqslant 1.64$ *and*

$$
\begin{aligned}
&B^{(e)} < \frac{q}{41c^{(e)}}\\
&(B^{(e)})^2 + \sqrt{(\upsilon_e + k_{\mathsf{bin}})d}B^{(e)} < q,\\
&2(\max_{i\in[\upsilon_e]}\beta_i^{(e)})^2 + (B^{(e)})^2 - 1 < q.
\end{aligned}
$$

Then the protocol described on Figure 10 is a commit-and-prove protocol for proving Equations (50) *to* (54). *Concretely, let* P_{eval} *be the success probability of a honest prover in* $\mathit{\Pi}$, $P_{\mathcal{E}'}, T_{\mathcal{E}'}$ *be respectively the success probability and the run time of the extractor* $\mathcal{E}'$ *from Theorem 4.5. For correctness, if the prover and the verifier follow the protocol in Figure 10 honestly and* $t = 1.64$, *then the verifier shall accept with probability*

$$\approx P_{\mathsf{eval}}\exp(-\frac{1}{2(\gamma^{(d)})^2})\exp(-\frac{1}{2(\gamma^{(e)})^2}).$$

For soundness, let $\mathcal{P}$ *be a probabilistic prover with success probability* $\epsilon \geqslant \frac{2}{|\mathcal{C}|} - q^{-d/2} - q^{-\lambda} - 2^{-127}$. *There exists an extractor that with rewindable black-box access to* $\mathcal{P}$, *either breaks the binding of the commitment or recovers a valid opening*

$$(\bar{\mathbf{s}}_2, (\bar{\mathbf{s}}_1, \bar{\mathbf{x}}), (\bar{\mathbf{m}}, \bar{\mathbf{y}}^{(d)}, \bar{\mathbf{y}}^{(e)}, \bar{b}^{(d)}, \bar{b}^{(e)}), \bar{c}) \in \mathcal{R}_q^{m_1+m_2+\ell+\upsilon_d\cdot 256/d+\upsilon_e\cdot 256/d+1+1} \times \mathcal{R}_q^{\times}$$

for the commitment $(\mathbf{t}, \mathbf{t}^{(d)}, \mathbf{t}^{(e)}, t^{(d)}, t^{(e)})$ *in expected time* $2T_{\mathsf{eval}}$, *satisfying Equations* (50) *to* (54).

For commit-and-prove simulatability, there exists a simulator $\mathcal{S}$ *that, without access to private information* $\mathbf{s}_1, \mathbf{m}$, *outputs a simulation of a commitment* $(\mathbf{t}, \mathbf{t}^{(d)}, t^{(d)}, \mathbf{t}^{(e)}, t^{(e)})$ *along with a non-aborting transcript of the protocol between prover* $\mathcal{P}$ *and verifier* $\mathcal{V}$ *such that for every algorithm* $\mathcal{A}$ *that has advantage* ε *in distinguishing the simulated commitment and transcript from the actual commitment and transcript, whenever the prover does not abort, there is an algorithm* $\mathcal{A}'$ *with the same running time that has advantage* $\varepsilon/2 - 2^{-100}$ *in distinguishing Extended-*$\mathsf{MLWE}_{n+\ell+\lambda+(256/d+1)+(256/d+1), m_2-n-\ell-\lambda-(256/d+1)+(256/d+1), \chi, \mathcal{C}, \mathfrak{s}_2}$.

Proof. **Correctness.** The success probability of a honest prover is at least the probability that 1) Both rejection sampling steps on $\vec{z}^{(d)}$ and $\vec{z}^{(e)}$ do not abort 2) The zero-knowledge proof $\mathit{\Pi}$ successfully convinces the verifier and 3) Both norm checks are verified. For the rejection sampling steps, each have an independent probability of respectively $\exp(-\frac{1}{2(\gamma^{(d)})^2})$, $\exp(-\frac{1}{2(\gamma^{(e)})^2})$ not to abort, which yields a probability of $\exp(-\frac{1}{2(\gamma^{(d)})^2})\exp(-\frac{1}{2(\gamma^{(e)})^2})$ for 1). The tail bound Lemma 2.2 indicates that with our bounds, both norm checks in 3) are verified with probability respectively $512\exp(-14^2/2)$ and $\left(te^{\frac{1-t^2}{2}}\right)^{256}$. With $t \geqslant 1.64$, both probabilities are $1 - 2^{-128}$. Remains to show 2), which we do by showing that $\mathit{\Pi}$ is a valid instantiation of Figure 8 and therefore convinces the verifier with probability P_{eval}. Since the prover is honest, we assume that $\mathbf{s}$ verifies Equations (50) to (54) and look at each input function one by one: For all $1 \leqslant i \leqslant \rho$, Equation (50) implies $f_i(\mathbf{s}) = 0$, and since both $b^{(d)}$ and $b^{(e)}$ are signs, we also have $g^{(d)}(b^{(d)}) = g^{(e)}(b^{(e)}) = 0$. From Equation (51), we have $\widetilde{F}_i(\mathbf{s}) = 0$. The vector $\mathbf{x}'$ is binary by construction and by assumption from Equation (54). Under Lemma 2.4 $G(\mathbf{x}') = \langle \mathbf{x}', 1 - \mathbf{x}' \rangle \mod q$, and under Lemma 5.2 we have $\widetilde{G}(\mathbf{x}') = 0$. Again under Lemma 2.4, if both $\vec{z}^{(d)}$ and $\vec{z}^{(e)}$ are honestly constructed, the functions $H_j^{(d)}$ and $H_j^{(e)}$ also have constant coefficients 0. For each $i \in [\upsilon_e]$, the vector $\vec{x}_i$ is constructed as the binary decomposition of $(\beta_i^{(e)})^2 - \|\mathbf{E}_i\mathbf{s} - \mathbf{v}_i\|^2$, and this vector $\vec{x}_i$ therefore has support at most $2\log\beta_i^{(e)} \leqslant d$. With $\vec{p}_i$ defined as the vector whose coefficients are the list of powers-of-two until $2\log\beta_i^{(e)}$ (and then zeros), we have

$$\langle \vec{p}_i, \vec{x}_i \rangle = (\beta_i^{(e)})^2 - \langle \mathbf{E}_i\mathbf{s} - \mathbf{v}_i, \mathbf{E}_i\mathbf{s} - \mathbf{v}_i \rangle.$$

From Lemma 2.4, the latter equation implies that $\widetilde{I}_i(\mathbf{s}, \mathbf{x}) = 0$, except the inner products are taken modulo q rather than over the integers. Since we assumed $2(\max_{i \in [\upsilon_e]} \beta_i^{(e)})^2 + (B^{(e)})^2 - 1 < q$, these computations also hold over the integers and therefore we do have $\widetilde{I}_i(\mathbf{s}, \mathbf{x}) = 0$. Finally, $\forall i \in \{d, e\},\ 1 \leqslant j \leqslant d-1,\ \widetilde{J}_j^{(i)}(b^{(i)}) = 0$ is immediate since the function maps $b^{(i)}$ to its j-th coefficient and $b^{(i)}$ is a constant, which completes the correctness proof.

Commit-and-prove simulatability follows from the commit-and-prove simulatability of Figure 8, the rejection sampling and the hiding property of ABDLOP.

Soundness. We let $\mathcal{E}$ be the extractor for this zero-knowledge proof and $\mathcal{E}'$ be the extractor from $\Pi_{\mathsf{eval}}^{(2)}$. The extractor $\mathcal{E}$ proceeds as follows:

1. Run the prover until the third round on honestly generated challenges $R^{(d)}, R^{(e)}$ then run $\mathcal{E}'$. If $\mathcal{E}$ does not obtain a valid opening $(\overline{\mathbf{s}}_2, (\overline{\mathbf{s}}_1, \overline{\mathbf{x}}), (\overline{\mathbf{m}}, \overline{\mathbf{y}}^{(d)}, \overline{\mathbf{y}}^{(e)}, \overline{b}^{(d)}, \overline{b}^{(e)}), \overline{c}) \in \mathcal{R}_q^{m_1+m_2+\ell+\upsilon_d \cdot 256/d+\upsilon_e \cdot 256/d+1+1} \times \mathcal{R}_q^{\times}$ satisfying the relations given by Equations (50) to (54), then abort otherwise continue.
2. Rewind the prover until the third round, send honestly and freshly generated challenges $R^{(d)'}, R^{(e)'}$ and run $\mathcal{E}'$ until $\mathcal{E}$ successfully obtains a valid opening $(\overline{\mathbf{s}}_2', (\overline{\mathbf{s}}_1', \overline{\mathbf{x}}'), (\overline{\mathbf{m}}', \overline{\mathbf{y}}^{(d)'}, \overline{\mathbf{y}}^{(e)'}, \overline{b}^{(d)'}, \overline{b}^{(e)'}), \overline{c}')$ satisfying the relations given by Equations (50) to (54).

With this extraction strategy, the expected run time of $\mathcal{E}$ is 2 times the expected run time of $\mathcal{E}'$ and $\mathcal{E}$ has the same success probability as $\mathcal{E}'$, see Lemma 4.3 for justifications. Notice that since $\epsilon \geqslant \frac{2}{|\mathcal{C}|} + q^{-d/2} + q^{-\lambda}$, in particular the success probability of the prover at producing a valid Π in the last step is also at least $\frac{2}{|\mathcal{C}|} + q^{-d/2} + q^{-\lambda}$ and therefore $P_{\mathcal{E}'} \geqslant \epsilon - \frac{2}{|\mathcal{C}|} - q^{-d/2} - q^{-\lambda} \geqslant 2^{-128}$.

Similarly as in the soundness proof of Proposition 5.1, either $\mathcal{E}$ breaks the binding property of the commitment scheme, or the messages in both transcripts are the same, which in turn implies that the challenge matrices $R^{(d)}$ and $R^{(e)}$ are independent of those messages. We focus on the latter case.

We have the following:

1. $\forall i \in [\rho],\ f_i(\overline{\mathbf{s}}, \sigma(\overline{\mathbf{s}}))$
2. For $i \in [\rho_{\mathsf{eval}}],\ \widetilde{F}_i(\overline{\mathbf{s}}, \sigma(\overline{\mathbf{s}})) = 0$
3. $\forall i \in \{d, e\},\ g^{(i)}(\overline{b}^{(i)}) = 0$
4. $G(\overline{\mathbf{x}}') = 0$
5. For $j \in [256],\ \widetilde{H}_j^{(d)}(\overline{\mathbf{s}}, \overline{\mathbf{y}}_d, b_d) = 0$
6. For $j \in [256],\ \widetilde{H}_j^{(i)}(\overline{\mathbf{x}}', \overline{\mathbf{s}}, \overline{\mathbf{y}}_e, b_e) = 0$
7. For $i \in [\upsilon_e],\ \widetilde{I}_i(\overline{\mathbf{s}}, \overline{\mathbf{x}}) = 0$
8. For $i \in \{d, e\},\ j \in \{1, \ldots, d-1\},\ \widetilde{J}_i(b_i) = 0.$

We use repeatedly Lemma 2.4 to infer inner product relations from the list of equations satisfied above, see the correctness paragraph for full explanation. First, 1) and 2) imply respectively that Equations (50) and (51) are satisfied.

Next, 3) implies that $\overline{b}^{(d)}$ and $\overline{b}^{(e)}$ are roots of $(X-1)(X+1)$, and 8) implies that all coefficients of $\overline{b}^{(d)}$ and $\overline{b}^{(e)}$ are 0 except for the constant one. Since $\mathbb{Z}_q$ is an integral domain, we have that $\overline{b}^{(d)}$ and $\overline{b}^{(e)}$ are signs.

5) implies the well-formedness of $\vec{z}^{(d)} = \overline{b}^{(d)} R^{(d)} \overline{\vec{e}}^{(d)} + \overline{\vec{y}}^{(d)}$, where $\overline{\vec{e}}^{(d)}$ is defined as in Equation (61) with the extracted messages. From the norm verification on $\vec{z}^{(d)}$, we have that $\|\vec{z}^{(d)}\|_\infty = \|\overline{b}^{(d)} R^{(d)} \overline{\vec{e}}^{(d)} + \overline{\vec{y}}^{(d)}\|_\infty \leqslant t256\mathfrak{s}^{(d)}$. Notice that since $\overline{b}^{(d)}$ is a sign, the distribution of $\overline{b}^{(d)} R^{(d)}$ is $\mathsf{Bin}_1^{256 \times c^{(d)}}$. As we assumed that the extractor does not break the binding of the commitment, $\overline{\vec{e}}^{(d)}$ is fixed and the hypotheses of Lemma 2.7 are satisfied. Using the latter Lemma, we have that the probability (over the randomness of the challenge $R^{(d)}$) that $\|\vec{z}^{(d)}\|_\infty \leqslant \frac{1}{2}\|\overline{\vec{e}}^{(d)}\|$ is less than 2^{-256}. Rearranging the terms, we have that with probability that $\|\overline{\vec{e}}^{(d)}\|_\infty \leqslant 24\mathfrak{s}^{(d)}$ is at least $1 - 2^{-256}$, hence Equation (52) is satisfied.

6) implies the well-formedness of $\vec{z}^{(e)} = \overline{b}^{(e)} R^{(e)} \overline{\vec{e}}^{(e)} + \overline{\vec{y}}^{(e)}$, where $\overline{\vec{e}}^{(e)}$ is defined as in Equation (61) with the extracted messages. Similarly as above, $\overline{b}^{(e)}$ is a sign hence $\overline{b}^{(e)} R^{(e)}$ follows $\mathsf{Bin}_1^{256 \times c^{(e)}}$ and is independent

of $\vec{\bar{e}}^{(e)}$. As we assumed $B^{(e)} < \frac{q}{41c^{(e)}}$, we can use Lemma 2.9 which yields that if $\|\vec{\bar{e}}^{(e)}\| \geqslant B^{(e)}$, then the probability that $\|\vec{z}^{(e)}\| \leqslant 1/2B^{(e)}\sqrt{26}$ is less than 2^{-128}. Rearranging the terms, we obtain that the probability that $\|\vec{\bar{e}}^{(e)}\| \leqslant B^{(e)}$ is at least $1 - 2^{-128}$.

4) implies that $\bar{\mathbf{x}}'$ (again defined accordingly with the extracted messages) $\langle \mathbf{x}', \mathbf{x}' - \mathbf{1} \rangle = 0 \mod q$. Moreover, as we assumed $(B^{(e)})^2 + \sqrt{(\upsilon_e + k_{\mathsf{bin}})d}B^{(e)} < q$, the latter inner product does not entail a wraparound modulo q and therefore holds over the integers. Under Lemma 5.2, this implies that $\mathbf{x}'$ is binary. In particular, Equation (54) is satisfied.

Finally 7) implies that $\langle \vec{p}_i, \vec{\bar{x}}_i \rangle = (\beta_i^{(e)})^2 - \|\vec{\bar{e}}^{(e)}\|^2 \mod q$. Moreover, as we assumed $2(\max_{i \in [\upsilon_e]} \beta_i^{(e)})^2 + (B^{(e)})^2 - 1 < q$, then the computation above holds over the integers. In particular, the latter equation implies that $(\beta_i^{(e)})^2 - \|\vec{\bar{e}}^{(e)}\|^2$ is a positive integer and therefore Equation (53) is satisfied.

To conclude, either $\mathcal{E}$ breaks the binding of the commitment or $\mathcal{E}$ finds a valid opening to messages satisfying Equations (50) to (54) with probability $P_{\mathcal{E}'}(1 - 2^{-256})(1 - 2^{-128})$ in time $2T_{\mathcal{E}'}$. □

5.3 Packing Signs

Recall that we commit to each sign $b^{(e)}$ and $b^{(d)}$ separately. We can reduce the proof size by committing to both of them in the following way. Namely, we compute

$$b := b^{(e)} + X^{d/2} b^{(d)} \in \mathcal{R}_q$$

and commit to b:

$$t_b := \mathbf{b}_b^T \mathbf{s} + b.$$

In order to prove certain properties of $b^{(e)}$ and $b^{(d)}$, we observe that:

$$b^{(e)} = 2^{-1} \cdot (b + \sigma(b)) \text{ and } b^{(d)} = 2^{-1} \cdot (X^{d/2} b + \sigma(X^{d/2} b)).$$

Then, for example, to prove that $b^{(e)}$ is a sign, we show that

$$(b^{(e)})^2 - 1 = \left(2^{-1} \cdot (b + \sigma(b))\right)^2 - 1 = 4^{-1} \cdot (b^2 + 2\sigma(b)b + \sigma(b)^2) - 1 = 0$$

and the constant coefficient of

$$X^i \cdot b^{(e)} = X^i \cdot 2^{-1} \cdot (b + \sigma(b))$$

is equal to zero for $i = 1, 2, \ldots, d-1$. Hence, these quadratic relations (with automorphisms) can be handled directly by $\Pi_{\mathsf{eval}}^{(2)}$.

5.4 Version of the ℓ_2-Norm Proof Without Approximate ℓ_∞ Proof

In this subsection, we deal with the particular instantiation of Figure 10 for $\upsilon_d = 0$. Simply setting this parameter to 0, for example still requires the prover to send the commitments $\mathbf{t}^{(d)}, t^{(d)}$ although these are not useful. We detail the savings and changes in the protocol for this particular instance.

We assume $\upsilon_d = 0$. In this case, the prover does not sample $b^{(d)}$ nor $\vec{y}^{(d)}$ and hence does not send the two commitments $\mathbf{t}^{(d)}, t^{(d)}$ in the first round.

The challenge matrix $R^{(d)}$ is $256 \times c^{(d)}$, where $c^{(d)}$ is 0 hence the verifier does not send this challenge either. The prover only computes $\vec{z}^{(e)}$, which means he only runs the rejection sampling $\mathsf{Rej}_0(\vec{z}^{(e)}, b^{(e)} R^{(e)} \vec{e}^{(e)}, \mathfrak{s}^{(e)})$ and only sends this $\vec{z}^{(e)}$ in the third round. The vector $\mathbf{s}^*$ is defined as $(\mathbf{s}_2,\ (\mathbf{s}_1, \mathbf{x}),\ (\mathbf{m}, \mathbf{y}^{(e)}, b^{(e)}))$, and the functions Φ, Ψ are defined as

$$\begin{aligned} \phi &= (f_1, \ldots, f_\rho, g^{(e)}) \\ \Psi &= \left((F_i)_{i \in \rho_{\mathsf{eval}}}, G, (H_j^{(e)})_{j \in [256]}, (I_i)_{i \in \upsilon_e}, (J_j^{(e)})_{j \in [d]}\right). \end{aligned}$$

6 Concrete Instantiations

In this section we show how to make use of our techniques for proving norms in the real-world applications, such as proving knowledge of a Module-LWE secret, verifiable encryption and group signatures. In order to show significance of our results, we compare our efficiency with relevant prior work.

6.1 General Strategy

We first provide a general strategy on instantiating the protocol in Fig. 10 with an improvement presented in Section 4.4. Firstly, we pick the challenge space $\mathcal{C}$ as described in Section 2.7. Further, we choose λ and q_1 such that terms $q_1^{-\lambda}$ and $q_1^{-d/2}$ are negligible.

There are four rejection sampling algorithms: the first two to mask $c\mathbf{s}_i$ for $i = 1, 2$ and then the latter two to mask $R^{(e)}\vec{e}^{(e)}$ and $R^{(d)}\vec{e}^{(d)}$. Let $\gamma_1, \gamma_2, \gamma^{(d)}, \gamma^{(e)} > 0$. Then, we define

$$\mathfrak{s}_1 = \gamma_1\eta\sqrt{\alpha^2 + v_e \cdot d}, \qquad \mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d}, \quad \mathfrak{s}^{(e)} = \gamma_3\sqrt{337}\alpha^{(e)}, \quad \mathfrak{s}^{(d)} = \gamma_4\sqrt{337}\alpha^{(d)}.$$

Thus, the non-aborting probability of the prover is

$$\approx \frac{1}{2\exp\left(\frac{14}{\gamma_1} + \frac{1}{2\gamma_1^2} + \frac{1}{2\gamma_2^2} + \frac{1}{2\gamma_3^2} + \frac{1}{2\gamma_4^2}\right)}.$$

Now we set n and m_2 such that Extended-MLWE and MSIS from Theorem 4.5 are hard against known attacks. We measure the hardness with the root Hermite factor δ and aim for $\delta < 1.0045$ similarly as in [BLS19, ALS20, ENS20, LNS21a]. For Module-SIS, we applied the standard methodology from [MR09, GN08]. Also, we assume that Extended-MLWE is almost as hard as plain MLWE (see [LNS21a] for more discussion) and applied the LWE-Estimator by Albrecht et al. [APS15].

Further, we look at the size of the non-interactive proof outputs via the Fiat-Shamir transform of the protocol in Fig. 10. First, note that for the non-interactive proof the messages $\mathbf{w}$ and v need not be included in the output as they are uniquely determined by the remaining components. Moreover, all the challenges apart from c can be computed as a hash of the previous components of the proof. On the other hand, sending c requires at most $\lceil \log(2\kappa + 1)\rceil \cdot d$ bits.

As "full-sized" elements of $\mathcal{R}_q$, we have $\mathbf{t}_A, \mathbf{t}_B, \mathbf{t}^{(d)}, t^{(d)}, \mathbf{t}^{(e)}, t^{(e)}, \mathbf{t}_g, t$ and h_i. Therefore, we have in total $n + \ell + 2 \cdot (256/d + 1) + 2\lambda + 1$ full-sized elements of $\mathcal{R}_q$, which altogether costs at most

$$(n + \ell + 512/d + 2\lambda + 3)\, d\lceil \log q\rceil \text{ bits.}$$

Integer	Representation	Bits
0	00	2
1	01	2
-1	10	2
$k \geqslant 2$	$110^{2k-4}1$	$2k - 1$
$k \leqslant -2$	$110^{2k-3}1$	$2k$

Table 3: Prefix-free encoding [DLL+17].

Now, the only remaining part are the vectors $\mathbf{z}_1, \mathbf{z}_2, \vec{z}^{(d)}, \vec{z}^{(e)}$. We can encode them using the Huffman coding. Concretely, suppose that $z \leftarrow D_{\mathfrak{s}}$. Then, we can write

$$z := z_1 \cdot 2^{\delta+1} + z_0$$

where $z_0 = z \bmod {}^{\pm}2^{\delta+1}$. Since the expected absolute value of z is $\mathfrak{s}$ and assuming that $2^\delta \approx \mathfrak{s}$, the value of z_0 is close to being uniformly random between -2^δ and 2^δ. Due to the discrete Gaussian tails, the tails of the distribution of z_1 decrease very fast. Hence, the idea is to send z_0 in the clear (which has $\delta + 1$ bits) and then encode z_1 using the Huffman coding. If we assume that $\mathfrak{s} = 2^\delta$ and the tails of z_1 are the same as in the normal distribution centred at zero [15], then the above compression requires on average approximately 1.57 bits to represent z_1. Thus, the total representation of z requires on average $\approx 2.57 + \delta$ bits. Applying this strategy to $\mathbf{z}_1, \mathbf{z}_2, \vec{z}^{(d)}, \vec{z}^{(e)}$, the overall commitment and proof length is around

$$(n + \ell + 512/d + 2\lambda + 3)\, d\lceil \log q\rceil + \lceil \log(2\kappa + 1)\rceil \cdot d + (m_1 + \upsilon_e)d \cdot (2.57 + \lceil \log \mathfrak{s}_1\rceil) + m_2 d \ \cdot (2.57 + \lceil \log \mathfrak{s}_2\rceil) \\ + 256 \cdot (2.57 + \lceil \log \mathfrak{s}^{(e)}\rceil) + 256 \cdot (2.57 + \lceil \log \mathfrak{s}^{(d)}\rceil) \text{ bits.}$$

Further, we can reduce the number of garbage terms g_j from λ to $\lambda/2$ using the optimisation based on the σ_{-1} automorphism described in Section 4.4. Moreover, as described in Section 5.3, we can commit to $b^{(d)}$ and $b^{(e)}$ as one polynomial. Hence, the total proof size becomes:

$$(n + \ell + 512/d + \lambda + 2)\, d\lceil \log q\rceil + \lceil \log(2\kappa + 1)\rceil \cdot d + (m_1 + \upsilon_e)d \cdot (2.57 + \lceil \log \mathfrak{s}_1\rceil) + m_2 d \ \cdot (2.57 + \lceil \log \mathfrak{s}_2\rceil) \\ + 256 \cdot (2.57 + \lceil \log \mathfrak{s}^{(e)}\rceil) + 256 \cdot (2.57 + \lceil \log \mathfrak{s}^{(d)}\rceil) \text{ bits.}$$

Dilithium compression. For fair comparison with prior works, we further reduce the commitment and proof size by applying Dilithium-G [DLL+17] compression techniques, as in [LNS21a] and [ESZ21]. We describe the optimisation in Appendix A. The only change from the previous case is the introduction of the variables D (for cutting low-order bits of the commitment $\mathbf{t}_A$) and γ (for cutting low-order bits of $\mathbf{w}$ which allows us not to send some part of the masked opening $\mathbf{z}_2$ of the commitment randomness $\mathbf{s}_2$). Then, by Theorem A.2, we choose n, m_2 and D, γ so that the $\mathsf{MSIS}_{n,m_1+m_2,B}$ is hard for $B := 4\eta \cdot \sqrt{B_1^2 + B_2^2}$ where $B_1 = 2\mathfrak{s}_1\sqrt{2m_1 d}$ and $B_2 = 2\mathfrak{s}_2\sqrt{2m_2 d} + 2^D\eta\sqrt{nd} + \gamma\sqrt{nd}$.

As a rule of thumb, we first set $D = \gamma = 0$ and pick the largest n such that $\mathsf{MSIS}_{n,m_1+m_2,B}$ is hard. Next, we find the largest γ (note that D is still zero) for which the Module-SIS problem is still hard. Finally, after fixing n and γ, we choose the largest D such that $\mathsf{MSIS}_{n,m_1+m_2,B}$ is still hard and also $2^{D-1}\kappa d < \gamma$. Note that having larger D decreases the commitment size at the cost of having larger hints and therefore, there is no advantage in picking larger D than $\log(\gamma/(\kappa d)) + 1$.

Now, we provide an asymptotic analysis of bounding the size of the hint vector $\mathbf{h}$. First, note that the coefficient vector $\mathbf{h}$ with high probability satisfies $\|\mathbf{h}\|_\infty \leqslant \|\mathsf{HighBits}_q(c\mathbf{t}_{A,2} - \mathbf{z}_{2,2})\|_\infty$ (here we assume the low-order bits $\mathbf{w}_0$ of $\mathbf{w}$ do not cause the increase in the high-order bits). Then, $\|c\mathbf{t}_{A,2} + \mathbf{z}_{2,2}\|_\infty \leqslant 2^{D-1}\kappa d + 16\mathfrak{s}_2$ with an overwhelming probability by Lemma 2.2. Hence, we conclude that (with high probability) the coefficients of $\mathbf{h}$ are between $-x$ and x where

$$x := \left\lceil \frac{2^{D-1}\kappa d + 16\mathfrak{s}_2}{\gamma} \right\rceil. \tag{70}$$

For our parameters, the standard deviation $\mathfrak{s}_2$ will be much smaller than γ and thus x will be close to $2^{D-1}\kappa d/\gamma$. Finally, by picking D such that $2^{D-1}\kappa d < \gamma$, we conclude that the coefficients of $\mathbf{h}$ are between -1 and 1 with high probability. Assuming heuristically that they follow a binomial distribution, we encode $\mathbf{h}$ using a prefix-free encoding [16] [DLL+17] as shown in Table 3. As computed in [DLL+17], encoding a coefficient of $\mathbf{h}$ requires on average ≈ 2.25 bits.

The final proof size including compression becomes:

$$nd(\lceil \log q\rceil - D) + 2.25 \cdot nd + (\ell + 512/d + \lambda + 2)\, d\lceil \log q\rceil + \lceil \log(2\kappa + 1)\rceil \cdot d + (m_1 + \upsilon_e)d \cdot (2.57 + \lceil \log \mathfrak{s}_1\rceil) \\ + m_2 d \ \cdot (2.57 + \lceil \log \mathfrak{s}_2\rceil) + 256 \cdot (2.57 + \lceil \log \mathfrak{s}^{(e)}\rceil) + 256 \cdot (2.57 + \lceil \log \mathfrak{s}^{(d)}\rceil) \text{ bits.}$$

[15] This assumption is needed so that we can compute the frequencies for the Huffman coding.

[16] One could apply the Huffman coding as before, however this requires computing the frequencies of the hint coefficients.

Skipping the non-exact norm proof. In certain applications, we will not perform any non-exact ℓ_∞ norm proofs, as described in Section 5.4 In this scenario we do not send the commitments $\mathbf{t}^{(d)}, t^{(d)}$ and the masked opening $\vec{z}^{(d)}$. Also, the packing technique from Section 5.3 becomes pointless. In conclusion, the proof size for this case becomes:

$$nd(\lceil \log q \rceil - D) + (\ell + 256/d + \lambda + 2)\, d\lceil \log q \rceil + \lceil \log(2\kappa + 1) \rceil \cdot d + (m_1 + \upsilon_e) d \cdot (2.57 + \lceil \log \mathfrak{s}_1 \rceil) \\ + m_2 d \ \cdot (2.57 + \lceil \log \mathfrak{s}_2 \rceil) + 2.25 \cdot nd + 256 \cdot (2.57 + \lceil \log \mathfrak{s}^{(e)} \rceil) \text{ bits.}$$

We additionally provide SAGE [The22] scripts which compute parameters for the examples described in this section:

`https://github.com/khalvador/LBZKP`.

6.2 Proving Knowledge of a Module-LWE Secret

As a primary benchmark for comparison with prior work [ENS20, LNS21a], we prove knowledge of a Module-LWE secret. Namely, we want to prove knowledge of $(\mathbf{s}, \mathbf{e}) \in \mathcal{R}_q^{M+N}$ such that $\|(\mathbf{s}, \mathbf{e})\| \leqslant B$ and

$$\mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{u} \pmod q \tag{71}$$

where $\mathbf{A} \in \mathcal{R}_q^{N \times M}$ and $\mathbf{u} \in \mathcal{R}_q^N$ are public.

We propose the following solution using the framework developed in Section 5. Simply, we commit to $\mathbf{s}_1 := \mathbf{s}$ and prove that

$$\left\| \begin{bmatrix} \mathbf{s} \\ \mathbf{A}\mathbf{s} - \mathbf{u} \end{bmatrix} \right\| = \left\| \begin{bmatrix} \mathbf{I}_M \\ \mathbf{A} \end{bmatrix} \mathbf{s} - \begin{bmatrix} \mathbf{0} \\ \mathbf{u} \end{bmatrix} \right\| \leqslant B.$$

In Fig. 11 we show to properly instantiate the protocol in Fig. 10 to prove knowledge of a Module-LWE secret.

variable	description	instantiation
ρ	# of equations to prove	0
ρ_{eval}	# of evaluations with const. coeff. zero	0
υ_e	# of exact norm proofs	1
υ_d	# non-exact norm proofs	0
k_{bin}	length of the binary vector to prove	0
$\mathbf{s}_1$	committed message in the Ajtai part	$\mathbf{s}$
$\mathbf{m}$	committed message in the BDLOP part	$\varnothing$ (no message)
$\mathbf{E}_1$	public matrix for proving $\|\mathbf{E}_1\mathbf{s} - \mathbf{v}_1\| \leqslant \beta_1$	$\begin{bmatrix} \mathbf{I}_M \\ \mathbf{A} \end{bmatrix}$
$\mathbf{v}_1$	public vector for proving $\|\mathbf{E}_1\mathbf{s} - \mathbf{v}_1\| \leqslant \beta_1$	$\begin{bmatrix} \mathbf{0} \\ \mathbf{u} \end{bmatrix}$
β_1	upper-bound on $\|\mathbf{E}_1\mathbf{s} - \mathbf{v}_1\| \leqslant \beta_1$	B

Fig. 11: Instantiation of the protocol in Fig. 10 for proving $\mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{u} \pmod q$ and $\|(\mathbf{s}, \mathbf{e})\| \leqslant B$. The variables in the first two columns refer to the ones defined in Section 5 and the ones in the last column refer to the parameters in this subsection.

Remark 6.1. We note that [ENS20, LNS21a] could not avoid committing to $\mathbf{e}$ without having additional commitments. Indeed, previous work efficiently prove smallness of a vector $\mathbf{s}$, e.g. $\|\mathbf{s}\|_\infty \leqslant 1$, by committing to its coefficient vector $\vec{s}_1$ using NTT slots and then proving that $\vec{s} \circ (\vec{s} - \vec{1}) \circ (\vec{s} + \vec{1}) = \vec{0}$ [ALS20]. If one were not to commit to $\mathbf{e}$, then one would need to prove an equation of the form

$$(A\vec{s} - \vec{u}) \circ \left(A\vec{s} - \vec{u} - \vec{1}\right) \circ (A\vec{s} - \vec{u} + \vec{1}) = \vec{0}.$$

parameters	description	value
q	prime modulus	$2^{32}-99$
d	ring dimension for of $\mathcal{R}$	128
l	# factors X^d+1 splits into mod q	2
N	height of the $\mathbf{A}$ matrix	8
M	width of the $\mathbf{A}$ matrix	8
γ_1	rejection sampling constant for $c\mathbf{s}_1$	19
γ_2	rejection sampling constant for $c\mathbf{s}_2$	1
$\gamma^{(e)}$	rejection sampling constant for the ARP	6
κ	maximum coefficient of a challenge in $\mathcal{C}$	2
n	height of matrices $\mathbf{A}_1, \mathbf{A}_2$ in ABDLOP	9
m_1	length of the message $\mathbf{s}_1$ in the "Ajtai" part	8
ℓ	length of the message $\mathbf{m}$ in the "BDLOP" part	0
λ	$2 \cdot (\#$ of garbages $g_j \in \mathcal{R}_q$ for boosting soundness)	4
m_2	length of the randomness $\mathbf{s}_2$ in ABDLOP	25
ν	randomness $\mathbf{s}_2$ is sampled from $S_\nu^{m_2}$	1
γ	parameter to cut low-order bits of $\mathbf{w}$	131052
D	number of low-order bits cut from $\mathbf{t}_A$	9
	repetition rate	7
	commitment + proof size	14.4KB

Fig. 12: Parameter selection for proving $\mathbf{As}+\mathbf{e}=\mathbf{u} \pmod q$ and $\|(\mathbf{s},\mathbf{e})\| \leqslant \sqrt{2048}$ using the protocol in Fig. 10

However, this relation, which is a mix of linear and product relations, cannot be proven using current methods included in [ENS20, LNS21a] without making intermediate commitments.

Parameters. We instantiate our protocol for the case when $q \approx 2^{32}$ and $N = M = 1024/d$ similarly as in [BLS19, ENS20, LNS21a] using the methodology in Section 6.1. We provide a summary of our parameter selection in Table 12.

Let us pick prime $q := 2^{32} - 99$ (i.e. $q = q_1$) and set $d = 128, l = 2$ and $(\alpha, B) = (\sqrt{1024}, \sqrt{2048})$[17]. Then we define the randomness distribution as uniform over S_1. For the challenge space, we set $\kappa = 2$, $\eta = 59$ as in Fig. 3. Also, for $q \approx 2^{32}$, we choose $\lambda = 4$. Then, $q^{-d/2} < q^{-\lambda} \approx 2^{-128}$ and $\kappa < q_1/2$.

There are three rejection sampling algorithms: one to mask $c\mathbf{s}_1$, another one to mask $c\mathbf{s}_2$ and the last one to mask $\|R\vec{e}^{(e)}\|$. Denote $\mathfrak{s}_i = \gamma_i T_i$ where T_1, T_2, T_3 are the upper-bounds on $\|c\mathbf{s}_1\|, \|c\mathbf{s}_2\|$ and $\|R\vec{e}^{(e)}\|$ respectively. The repetition rate in our case is at least

$$2\exp\left(\frac{14}{\gamma_1} + \frac{1}{2\gamma_1^2} + \frac{1}{2\gamma_2^2} + \frac{1}{2{\gamma^{(e)}}^2}\right).$$

The rate in [LNS21a] is around 7 hence we set $\gamma_1 = 19, \gamma_2 = 1$ and $\gamma^{(e)} = 6$. All in all, with our parameters we obtain proofs of size 14.4KB.

6.3 Verifiable Encryption

For presentation, we will consider a standard Regev public-key encryption scheme [Reg09] but similar analysis can be applied for more complex construction, such as Kyber [BDK+18], Saber [DKRV18] and NTRU [HPS98] (see [LNS21a][Section 4] for more details). Namely, let p be a prime modulus of the encryption scheme. In order to encrypt a binary message $m \in \{0,1\}^d$, a user samples a randomness vector $\mathbf{r} \leftarrow \xi^k$, where ξ is a distribution over $\mathcal{R}$, and compute

$$\begin{bmatrix} \boldsymbol{t}_0 \\ t_1 \end{bmatrix} := \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ \lfloor \frac{p}{2} \rceil m \end{bmatrix} \tag{72}$$

[17] It is the case when $\mathbf{s}_1, \mathbf{e}$ only consist of ternary coefficients as assumed in the prior work.

over $\mathcal{R}_p := \mathbb{Z}_p[X]/(X^d+1)$ where $(\mathbf{A}, \mathbf{b}) \in \mathcal{R}_p^{N \times K} \times \mathcal{R}_p^N$ is the public key [18]. Let B be an upper-bound on $\mathbf{r}$ such that the probability that $\|\mathbf{r}\| > B$ for $\mathbf{r} \leftarrow \xi^K$ is negligible. Then, in the verifiable encryption scenario, we want to prove knowledge of $\mathbf{r} \in \mathcal{R}^K$ and $m \in \mathcal{R}$ such that (i) Equation 72 is satisfied over $\mathcal{R}_p$, (ii) $\|\mathbf{r}\| \leqslant B$ and (iii) $m \in \{0,1\}^d$.

The high-level idea is to commit to $\mathbf{s}_1 := (\mathbf{r}, m)$ using the ABDLOP commitment modulo q and prove these three statements. Note that the latter two have already been covered in Section 5. Hence, from now on we focus on proving the first statement.

We first observe that if q is divisible by p then (72) can be transformed into a linear equation modulo q and can be proven as described in Section 4. However, in practical instantiations p will be significantly small relative to q (e.g. $p = 3329$ in Kyber). Consequently, if q has a small prime divisor p then by Theorem 4.5, we would need to commit to more garbage polynomials g_i in order to keep the soundness error negligible. Moreover, for implementation purposes one might want p to be a prime such that $X^d + 1$ splits into many factors modulo p (e.g. $p = 3329$). In this case, if p divides q, then the challenge space $\mathcal{C}$ does not have the invertibility property which is necessary for the soundness proof. In Fig. 14 we propose an example instantiation for the case when q is divisible by p (see parameter set II).

Now, suppose that p is co-prime to q. Then, (72) is true if and only if there exists a vector $\mathbf{v} \in \mathcal{R}^{N+1}$ such that

$$\begin{bmatrix} \boldsymbol{t}_0 \\ t_1 \end{bmatrix} := \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ \lfloor \frac{p}{2} \rceil m \end{bmatrix} + p\mathbf{v} \tag{73}$$

over $\mathcal{R}$. From a simple calculation, $\|\mathbf{v}\|_\infty \leqslant B\sqrt{Kd}/2 + 1$. We can avoid committing to $\mathbf{v}$, similarly as in Section 6.5, by proving directly that vector

$$\mathbf{v} := p^{-1} \cdot \left(\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{b}^T & \lfloor \frac{p}{2} \rceil \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ m \end{bmatrix} - \begin{bmatrix} \boldsymbol{t}_0 \\ t_1 \end{bmatrix} \right) \in \mathcal{R}_q^n \tag{74}$$

has norm at most $B_v := (B\sqrt{Kd}/2 + 1)\sqrt{(N+1)d}$. Since this expression is linear in the committed messages $\mathbf{r}$ and m, we can apply the protocol in Fig. 10 to prove its norm. As we will show below, it is enough to prove an approximate bound, i.e. $\|\mathbf{v}\|_\infty \leqslant B_v \cdot \psi$, where $\psi := 2 \cdot 14 \cdot \gamma^{(d)} \cdot \sqrt{337}$, as described in Section 5. Indeed, in the soundness argument we would extract a pair $(\mathbf{r}^*, m^*)$ which satisfies

$$\begin{cases} m^* \in \{0,1\}^d, \\ \|\mathbf{r}^*\| \leqslant B, \\ \left\| p^{-1} \cdot \left(\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{b}^T & \lfloor \frac{p}{2} \rceil \end{bmatrix} \begin{bmatrix} \mathbf{r}^* \\ m^* \end{bmatrix} - \begin{bmatrix} \boldsymbol{t}_0 \\ t_1 \end{bmatrix} \right) \right\|_\infty \leqslant B_v \psi. \end{cases}$$

Denote the third expression as $\mathbf{v}^* \in \mathcal{R}^{N+1}$. Then, we have

$$\begin{bmatrix} \boldsymbol{t}_0 \\ t_1 \end{bmatrix} \equiv \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^T \end{bmatrix} \mathbf{r}^* + \begin{bmatrix} \mathbf{0} \\ \lfloor \frac{p}{2} \rceil m^* \end{bmatrix} + p\mathbf{v}^* \pmod{q}. \tag{75}$$

Thus,

$$\left\| \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^T \end{bmatrix} \mathbf{r}^* + \begin{bmatrix} \mathbf{0} \\ \lfloor \frac{p}{2} \rceil m^* \end{bmatrix} + p\mathbf{v}^* - \begin{bmatrix} \boldsymbol{t}_0 \\ t_1 \end{bmatrix} \right\|_\infty \leqslant p \left(B\sqrt{Kd}/2 + 1 + B_v \psi \right).$$

Hence, if q is bigger than the right-hand side of this inequality, then we conclude that Equation (75) holds over integers. In particular $(\mathbf{t}_0, t_1)$ is a valid encryption of m under randomness $\mathbf{r}$ over $\mathcal{R}_p$.

In Fig. 13 we instantiate the protocol from Fig. 10 for verifiable encryption as described above.

Remark 6.2. Note that the current state-of-the-art lattice based verifiable encryption [LN17], which is used in e.g. [dPLS18, LNPS21], only provide *relaxed* verifiable encryption. Namely, the soundness argument only

[18] Recall that all coefficients of the terms involved in (72) are between $-p/2$ and $p/2$.

variable	description	instantiation
ρ	# of equations to prove	0
ρ_{eval}	# of evaluations with const. coeff. zero	0
v_e	# of exact norm proofs	1
v_d	# non-exact norm proofs	1
k_{bin}	length of the binary vector to prove	1
$\mathbf{s}_1$	committed message in the Ajtai part	$(\mathbf{r}, m)$
$\mathbf{m}$	committed message in the BDLOP part	$\varnothing$ (no message)
$\mathbf{E}_1$	public matrix for proving $\\|\mathbf{E}_1\mathbf{s} - \mathbf{v}_1\\| \leqslant \beta_1^{(e)}$	$\begin{bmatrix}\mathbf{I}_K & \mathbf{0}\end{bmatrix}$
$\mathbf{v}_1$	public vector for proving $\\|\mathbf{E}_1\mathbf{s} - \mathbf{v}_1\\| \leqslant \beta_1^{(e)}$	$\mathbf{0}$
$\beta_1^{(e)}$	upper-bound on $\\|\mathbf{E}_1\mathbf{s} - \mathbf{v}_1\\| \leqslant \beta_1^{(e)}$	B
$\mathbf{D}_1$	public matrix for proving $\\|\mathbf{D}_1\mathbf{s} - \mathbf{u}_1\\| \leqslant \beta_1^{(d)}$	$p^{-1} \cdot \begin{bmatrix}\mathbf{A} & \mathbf{0} \\ \mathbf{b}^T & \lfloor \frac{p}{2} \rceil\end{bmatrix}$
$\mathbf{u}_1$	public vector for proving $\\|\mathbf{D}_1\mathbf{s} - \mathbf{u}_1\\| \leqslant \beta_1^{(d)}$	$p^{-1} \cdot \begin{bmatrix}\boldsymbol{t}_0 \\ t_1\end{bmatrix}$
$\beta_1^{(d)}$	upper-bound on $\\|\mathbf{D}_1\mathbf{s} - \mathbf{u}_1\\| \leqslant \beta_1^{(d)}$	$(B\sqrt{Kd}/2 + 1)\sqrt{(N+1)d}$
$\mathbf{E}_{\mathsf{bin}}$	matrix for proving binary	$\begin{bmatrix}\mathbf{0} & 1\end{bmatrix}$
$\mathbf{v}_{\mathsf{bin}}$	matrix for proving binary	0

Fig. 13: Instantiation of the protocol in Fig. 10 for verifiable encryption. The variables in the first two columns refer to the ones defined in Section 5 and the ones in the last column refer to the parameters in this subsection. Triple $(\mathbf{E}_1, \mathbf{v}_1, \beta_1^{(e)})$ corresponds to proving exactly that $\|\mathbf{r}\| \leqslant B$. The next triple $(\mathbf{D}_1, \mathbf{u}_1, \beta_1^{(d)})$ corresponds to proving approximately that $\|\mathbf{v}\| \leqslant (B\sqrt{Kd}/2+1)\sqrt{(N+1)d}$ where $\mathbf{v}$ is defined in (74). Finally, $(\mathbf{E}_{\mathsf{bin}}, \mathbf{v}_{\mathsf{bin}})$ is defined to prove that m has binary coefficients.

guarantees knowledge of a message and randomness corresponding to the ciphertext $(\bar{c}\mathbf{t}_0, \bar{c}t_1)$, where $\bar{c} \in \mathcal{R}_p$ is called a relaxation factor. More importantly, $\bar{c}$ is not known to the decryptor and thus it guesses a $\bar{c}$ and attempts to recover the ciphertext $(\bar{c}\mathbf{t}_0, \bar{c}t_1)$. Consequently, the prior works had to equate the decryption time with the adversary's running time. Here, since we commit to $\mathbf{r}$ and m using a separate ABDLOP commitment, we circumvent the relaxation factor by proving exact norms on $\mathbf{r}$ and $m \in \{0,1\}^d$.

Parameters. We provide our parameters choices[19] in Fig. 14. For the ciphertext modulus and dimensions, we follow the Kyber instantiation. In particular, $N = 4, K = 9$ and $\mathbf{b} = \mathbf{A}^T\mathbf{s} + \mathbf{e}$ where the secret key $\mathbf{s}$ and error $\mathbf{e}$ come from Bin_2^{Nd} and Bin_2^{Kd} respectively. For the randomness distribution $\xi := \mathsf{Bin}_2^d$. Hence, we can set the upper-bound B on the norm of $\mathbf{r} \leftarrow \xi^K$ as $B = 2\sqrt{Kd}$ and thus $B_v = (Kd+1)\sqrt{(N+1)d}$.

To compute the decryption error probability, we want to calculate the probability that for $\mathbf{r}, \mathbf{e} \leftarrow \mathsf{Bin}_2^{Kd}$, $\|\langle \mathbf{r}, \mathbf{e} \rangle\|_\infty < q/4$. First, we compute that for any $\vec{r}, \vec{e} \leftarrow \mathsf{Bin}_2^{Kd}$, the probability that $\|\langle \vec{r}, \vec{e} \rangle\|_\infty > 800$ is less than 2^{-360}. Then, by the union-bound, the probability that $\|\langle \mathbf{r}, \mathbf{e} \rangle\|_\infty > 800$ is still at most 2^{-300}. Hence, in our parameter selection, we will pick a prime p larger than 3200.

The rest of the parameters are chosen similarly as in Sections 6.1 and 6.2. Finally, we need to check that

$$q \approx 2^{36} > p \cdot \left(B\sqrt{Kd}/2 + 1 + (B\sqrt{Kd}/2 + 1)\sqrt{(N+1)d}\psi\right).$$

The term on the right-hand side is much less than 2^{36} thus the inequality holds.

[19] One can also instantiate the encryption scheme over a larger ring, e.g. $\mathcal{R}' := \mathbb{Z}[X]/(X^{256}+1)$. Then, in order to apply our proof system over a smaller ring $\mathcal{R}$, one would first map the equations to work over $\mathcal{R}$ rather than $\mathcal{R}'$ as described in [LNPS21][Section 2.8].

parameter set	description	I	II
p	encryption modulus	3329	3253
N	height of $\mathbf{A}$	4	4
K	width of $\mathbf{A}$	9	9
ξ	ξ^K is the randomness distribution of $\mathbf{r}$	Bin_2^d	Bin_2^d
q	proof system modulus	$2^{36} - 579$	$2^{31} - 305215$
d	dimension of $\mathcal{R}$	128	128
l	# factors $X^d + 1$ splits into mod q	2	2
γ_1	rej. samp. constant for $c\mathbf{s}_1$	41	16
γ_2	rej. samp. constant for $c\mathbf{s}_2$	1.1	1.5
$\gamma^{(e)}$	rej. samp. constant for exact ARP	16	1.8
$\gamma^{(d)}$	rej. samp. constant for non-exact ARP	1	–
κ	maximum coefficient of a challenge in $\mathcal{C}$	2	2
n	height of $\mathbf{A}_1, \mathbf{A}_2$ in ABDLOP	9	9
m_1	length of the “Ajtai” message $\mathbf{s}_1$	10	6
ℓ	length of the “BDLOP” message $\mathbf{m}$	0	0
λ	$2 \cdot$ (# of garbage g_j for soundness)	4	12
m_2	length of randomness $\mathbf{s}_2$	29	29
ν	randomness $\mathbf{s}_2$ is sampled from $S_\nu^{m_2}$	1	1
γ	parameter to cut low-order bits cut from $\mathbf{w}$	503742	54096
D	number of low-order bits cut from $\mathbf{t}_A$	11	8
	repetition rate	7	7
	ciphertext size	1KB	1KB
	commitment + proof size	19.0KB	18.6KB

Fig. 14: Parameter selection, ciphertext and proof sizes for verifiable encryption. For the second parameter set we choose $q := 660061 \cdot 3253$. Since p divides q, we do not need to do an approximate range proof of $\mathbf{v}$ as for I. Consequently, we can pick smaller modulus q and apply a similar strategy as in Section 6.2. In particular, we do not commit to the whole vector $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_0) \in \mathcal{R}_q^{K-N} \times \mathcal{R}_q^N$, but only a part of it, i.e. $\mathbf{r}_1$.

6.4 Group Signature

We apply our proof system to the recent group signature construction by Lyubashevsky et al. [LNPS21]. Our construction inherits a big advantage from [dPLS18, LNPS21], namely signature generation and verification time do not depend on the size of the group. We first sketch the scheme and refer to [LNPS21] for more details. In this subsection, we work over the larger ring $\mathcal{R}_{kd} := \mathbb{Z}[X]/(X^{kd}+1)$ where $k \geqslant 1$ is a power-of-two. Then, define $\mathcal{R}_{kd,p} := \mathcal{R}_{kd}/(p)$ for an integer p. The benefit of having a larger ring than $\mathcal{R}$ is small public key size of our group signature. Operations in the construction will be over $\mathcal{R}_{kd,p}$ where p is prime.

Overview. Let $G \subseteq \mathcal{R}_{kd,p}$ be the identity space. The group manager first samples $\mathbf{A} \leftarrow \mathcal{R}_{kd,p}^{N\times(N+M)}, \mathbf{B}' \leftarrow \mathcal{R}_{kd,p}^{N\times\tau N}$, randomness matrix $\mathbf{R} \leftarrow S_{kd,1}^{(N+M)\times\tau N}$, where

$$S_{kd,1} := \{x \in \mathcal{R}_{kd} : \|x\|_\infty \leqslant 1\}$$

and sets $\mathbf{B} := \mathbf{AR}$. Further, it samples $\mathbf{u} \leftarrow \mathcal{R}_{kd,p}^N$. Then, the public key is a tuple

$$gpk := (\mathbf{A}, \mathbf{B}, \mathbf{B}', \mathbf{u}).$$

Now, for each user with identity $i \in G$, the group manager samples the secret key

$$sk_i := (\mathbf{s}_1^{(i)}, \mathbf{s}_2^{(i)}, \mathbf{s}_3^{(i)}) \leftarrow D_{\mathfrak{s}}^{((2\tau+1)N+M)kd}$$

such that

$$\begin{bmatrix}\mathbf{A}|\mathbf{B}+i\mathbf{G}|\mathbf{B}'\end{bmatrix}\begin{bmatrix}\mathbf{s}_1^{(i)}\\ \mathbf{s}_2^{(i)}\\ \mathbf{s}_3^{(i)}\end{bmatrix}=\mathbf{u}$$

using the [MP12] trapdoor sampling with standard deviation $\mathfrak{s}$ where $\mathbf{G} := \mathbf{I}_N \otimes [1\ g\ \cdots\ g^{\tau-1}]$ is a gadget matrix and $g := \lceil p^{1/\tau} \rceil$.

The high level idea for signing is for the user with identity $i \in G$ to prove knowledge of i and their secret key $sk_i := (\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3) \in \mathcal{R}_{kd,p}^{(2\tau+1)N+M}$ which satisfy:

$$\begin{bmatrix}\mathbf{A}|\mathbf{B}+i\mathbf{G}|\mathbf{B}'\end{bmatrix}\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\\ \mathbf{s}_3\end{bmatrix}=\mathbf{u}, \quad \left\|\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\\ \mathbf{s}_3\end{bmatrix}\right\| \leqslant B := \mathfrak{s}\sqrt{2((2\tau+1)N+M)kd}, \quad i \in G. \tag{76}$$

For the bound B we used Lemma 2.2 for $t = \sqrt{2}$.

In order to be able to open the group signature scheme, we will add a verifiable encryption to the signature. Namely, we want the signer to encrypt their identity i, using a public key associated to a decryption key that the group manager possesses, and prove that this encryption is indeed of their identity. We do this exactly as described in Section 6.3 with a prime $p_{\mathsf{enc}} := 3329$. Similarly, all the dimensions and bounds included in that Section will be written with subscript enc.

Efficient Proof of (76)**.** To begin with, note that relations over $\mathcal{R}_{kd,p}$ such as the first one in Equation (76) can be written equivalently over our usual subring $\mathcal{R}_p$. Indeed, Lyubashevsky et al. showed that there is an efficiently computable ring isomorphism between $\mathcal{R}_{kd}$ and $\mathcal{R}^k$, for an appropriately defined vector multiplication in $\mathcal{R}$, which preserves norms (see [LNPS21][Section 2.8] for more details). Hence, arbitrary relations we need to prove over $\mathcal{R}_{kd,p}$ can be proven by showing that some corresponding relations over $\mathcal{R}_p$ hold true.

Secondly, we observe that if we choose a proof system modulus q to be divisible by p and commit to $(i, \mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3)$ in the "Ajtai" part of the ABDLOP commitment then the first statement in (76) is simply a system of quadratic equations in the committed messages as in Section 4. Indeed, we pick $q = q_1 p$ where $q_1 < p$ and then prove an equivalent quadratic relation over $\mathcal{R}_q$, namely:

$$q_1\begin{bmatrix}\mathbf{A}|\mathbf{B}+i\mathbf{G}|\mathbf{B}'\end{bmatrix}\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\\ \mathbf{s}_3\end{bmatrix} = q_1\begin{bmatrix}\mathbf{A}|\mathbf{B}|\mathbf{G}|\mathbf{B}'\end{bmatrix}\begin{bmatrix}\mathbf{s}_1\\ \mathbf{s}_2\\ i\mathbf{s}_2\\ \mathbf{s}_3\end{bmatrix} = q_1\mathbf{u}. \tag{77}$$

Further, the second statement is about norms which is covered in Section 5.

Moreover, we define the identity space G. It should be designed so that we can efficiently prove that $i \in G$ (third statement). Let $\mathcal{B}$ be the set of non-zero binary polynomials in $\mathcal{R}_p$. Then, we define the identity space[20] as

$$G := \{i(X^k) \in \mathcal{R}_{kd,p} : i \in \mathcal{B} \text{ and } \|i\|_1 = \omega\}.$$

We choose ω so that the set G has size $\approx 2^{23}$ for comparison with related work [BDK+21, EZS+19]. Note that for an appropriate p, a difference of two distinct elements from G is still invertible over $\mathcal{R}_{kd,p}$ which is crucial for trapdoor sampling.

Note that the space G is constructed in such a way that when we map equations over $\mathcal{R}_{kd,p}$ to $\mathcal{R}_p^k$, then we only need to commit to one polynomial $i \in \mathcal{R}_p$ using our ABDLOP commitment instead of k polynomials, i.e. $i(X^k) \in \mathcal{R}_{kd,p}$. Similarly, we only need to send an encryption of i over $\mathcal{R}_p$ instead of $i(X^k)$. Hence, for

[20] Previous works [dPLS18, LNPS21] define the identity space G to be a set of integers $\mathbb{Z}_p$ since it was easier to prove set membership $i \in G$ with their proof system. Here, we make a small modification and set the identity space to be a subset of binary polynomials with fixed norm.

variable	description	instantiation
ρ	# of equations to prove	N
ρ_{eval}	# of evaluations with const. coeff. zero	1
υ_e	# of exact norm proofs	2
υ_d	# non-exact norm proofs	1
k_{bin}	length of the binary vector to prove	1
$\mathbf{s}_1$	committed message in the Ajtai part	$(\mathbf{s}_1^{(i)}, \mathbf{s}_2^{(i)}, \mathbf{s}_3^{(i)}, \mathbf{r}_{\mathsf{enc}}, i)$
$\mathbf{m}$	committed message in the BDLOP part	$\varnothing$ (no message)
$f_1, \ldots, f_N$	equations to prove	Equation 77
F_1	evaluation to prove const coeff. zero	$\sigma_{-1}(\sum_{i=0}^{d-1} X^i) \cdot i - \omega$
$\mathbf{E}_1$	public matrix for proving $\Vert\mathbf{E}_1\mathbf{s} - \mathbf{v}_1\Vert \leqslant \beta_1^{(e)}$	$\left[\mathbf{I}_{k(N+M+2\tau N)}\ \mathbf{0}\right]$
$\mathbf{v}_1$	public vector for proving $\Vert\mathbf{E}_1\mathbf{s} - \mathbf{v}_1\Vert \leqslant \beta_1^{(e)}$	$\mathbf{0}$
$\beta_1^{(e)}$	upper-bound on $\Vert\mathbf{E}_1\mathbf{s} - \mathbf{v}_1\Vert \leqslant \beta_1^{(e)}$	$\mathfrak{s}\sqrt{2((2\tau+1)N+M)kd}$
$\mathbf{E}_2$	public matrix for proving $\Vert\mathbf{E}_2\mathbf{s} - \mathbf{v}_2\Vert \leqslant \beta_2^{(e)}$	$\left[\mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{I}_{K_{\mathsf{enc}}}\mathbf{0}\right]$
$\mathbf{v}_2$	public vector for proving $\Vert\mathbf{E}_2\mathbf{s} - \mathbf{v}_2\Vert \leqslant \beta_2^{(e)}$	$\mathbf{0}$
$\beta_2^{(e)}$	upper-bound on $\Vert\mathbf{E}_2\mathbf{s} - \mathbf{v}_2\Vert \leqslant \beta_2^{(e)}$	$\sqrt{B_{\mathsf{enc}}}$
$\mathbf{D}_1$	public matrix for proving $\Vert\mathbf{D}_1\mathbf{s} - \mathbf{u}_1\Vert \leqslant \beta_1^{(d)}$	$p_{\mathsf{enc}}^{-1} \cdot \begin{bmatrix} \mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{A}_{\mathsf{enc}} & \mathbf{0} \\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{b}_{\mathsf{enc}}^T & \lfloor \frac{p_{\mathsf{enc}}}{2} \rceil \end{bmatrix}$
$\mathbf{u}_1$	public vector for proving $\Vert\mathbf{D}_1\mathbf{s} - \mathbf{u}_1\Vert \leqslant \beta_1^{(d)}$	$p_{\mathsf{enc}}^{-1} \cdot \begin{bmatrix} \boldsymbol{t}_0 \\ t_1 \end{bmatrix}$
$\beta_1^{(d)}$	upper-bound on $\Vert\mathbf{D}_1\mathbf{s} - \mathbf{u}_1\Vert \leqslant \beta_1^{(d)}$	$B_{v,\mathsf{enc}}$
$\mathbf{E}_{\mathsf{bin}}$	matrix for proving binary	$\left[\mathbf{0}\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}\ 1\right]$
$\mathbf{v}_{\mathsf{bin}}$	matrix for proving binary	0

Fig. 15: Instantiation of the protocol in Fig. 10 for the group signature. The variables in the first two columns refer to the ones defined in Section 5 and the ones in the last column refer to the parameters in this subsection. Variables with subscript enc are defined for the verifiable encryption in Section 6.3. Functions F_1 is used to prove that identity i has exactly ω ones. Triples $(\mathbf{E}_1, \mathbf{v}_1, \beta_1^{(e)})$ and $(\mathbf{E}_2, \mathbf{v}_2, \beta_2^{(e)})$ correspond to proving exactly $\Vert(\mathbf{s}_1^{(i)}, \mathbf{s}_2^{(i)}, \mathbf{s}_3^{(i)})\Vert \leqslant B$ and $\Vert\mathbf{r}_{\mathsf{enc}}\Vert \leqslant B_{\mathsf{enc}}$ respectively. The last triple $(\mathbf{D}_1, \mathbf{u}_1, \beta_1^{(d)})$ corresponds to proving approximately that $\Vert\mathbf{v}_{\mathsf{enc}}\Vert \leqslant B_{v,\mathsf{enc}} := (B_{\mathsf{enc}}\sqrt{K_{\mathsf{enc}}d}/2+1)\sqrt{(N_{\mathsf{enc}}+1)d}$ where $\mathbf{v}_{\mathsf{enc}}$ is defined in (74).Finally, $(\mathbf{E}_{\mathsf{bin}}, \mathbf{v}_{\mathsf{bin}})$ is defined to prove that i has binary coefficients.

such a set G, proving $i(X^k) \in G$ is equivalent to proving that i has binary coefficients and the sum of coefficients of i is equal to ω which is covered in Section 5.

Last but not least, we observe that including a verifiable encryption from Section 6.3 does not have a significant impact on the signature size. Indeed, identity i is already committed using the ABDLOP scheme and additionally committing to the randomness $\mathbf{r}$ (in the "Ajtai part") does not increase the commitment size. Hence, the only extra cost consists of: (i) a ciphertext, (ii) masked opening of the randomness $\mathbf{r}$, (iii) commitments and masked openings to polynomials involved in the approximate range proof for $\mathbf{v}$ in (74). As described in Fig. 16, for our instantiation the verifiable encryption costs around 6.5KB compared to 19.0KB shown in Fig. 14.

In summary, we show in Fig. 15 how to instantiate the protocol in Fig. 10 to construct a group signature.

Parameters. We present our parameter selection in Fig. 16 for a group signature instantiation which achieves security level 111. We start by setting $p = 2^{38} - 107$ and $q = (2^{26} - 371) \cdot p \approx 2^{64}$. Then, we choose $d = 128, k = 4$ and $l = 2$, thus $\mathcal{R}_{kd,p} = \mathbb{Z}[X]/(X^{512}+1)$. Next, let $N = 2, M = 3$ and $\tau = 5$, hence $g = \lceil p^{1/5} \rceil$. Further, we pick large enough standard deviation $\mathfrak{s}$ used for trapdoor sampling. We know from [MP12] that $\mathfrak{s} \geqslant 2(s_1(\mathbf{R})+1)\sqrt{g^2+1}$ where s_1 is the operator norm. Note that if $\mathbf{R}$ did not have a polynomial structure, i.e $R \leftarrow \{-1,0,1\}^{(N+M)kd \times \tau Nkd}$, we could use upper-bounds for norms of random subgaussian matrices, e.g.

[MP12][Lemma 2.9]. Namely, we would obtain the following bound

$$s_1(R) \leqslant \sqrt{(N+M)kd} + \sqrt{\tau N kd} + 6 \approx 128$$

with probability at least $1 - 2^{163}$. We found experimentally that for our structured matrix $\mathbf{R}$ a similar bound holds with at least 99% probability

$$s_1(\mathbf{R}) \leqslant \psi := 113$$

and thus we set

$$\mathfrak{s} := 2(\psi + 1)\sqrt{p^{2/\tau} + 1}.$$

Further, we describe how we choose N and M, i.e. the height and the width of the matrix $\mathbf{A}$. Concretely, in the traceability proof, the challenger sets $\mathbf{B} := \mathbf{AR} - i^*\mathbf{G}$ and $\mathbf{B}' = \mathbf{AR}'$ where $\mathbf{R}, \mathbf{R}' \leftarrow S_{kd,1}^{(N+M)\times\tau N}$ and $i^* \leftarrow G$. Additionally, it samples $sk^{\mathsf{gm}} := (\mathbf{s}_1^{\mathsf{gm}}, \mathbf{s}_2^{\mathsf{gm}}, \mathbf{s}_3^{\mathsf{gm}}) \leftarrow D_{\mathfrak{s}}^{((2\tau+1)N+M)kd}$ and computes $\mathbf{u} := [\mathbf{A}|\mathbf{AR}|\mathbf{AR}']\, sk^{\mathsf{gm}}$. It will hope that an adversary forges a signature for the identity i^{*}[21] In that case, we can extract from the forged signature the secret vector $sk_{i^*} = (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{s}}_3)$ such that

$$\begin{bmatrix}\mathbf{A}|\mathbf{AR}|\mathbf{AR}'\end{bmatrix}\begin{bmatrix}\bar{\mathbf{s}}_1\\ \bar{\mathbf{s}}_2\\ \bar{\mathbf{s}}_3\end{bmatrix} = \mathbf{u} = \begin{bmatrix}\mathbf{A}|\mathbf{AR}|\mathbf{AR}'\end{bmatrix}\begin{bmatrix}\mathbf{s}_1^{\mathsf{gm}}\\ \mathbf{s}_2^{\mathsf{gm}}\\ \mathbf{s}_3^{\mathsf{gm}}\end{bmatrix}$$

and thus

$$\mathbf{s} := \bar{\mathbf{s}}_1 - \mathbf{s}_1^{\mathsf{gm}} + \mathbf{R}(\bar{\mathbf{s}}_2 - \mathbf{s}_2^{\mathsf{gm}}) + \mathbf{R}'(\bar{\mathbf{s}}_3 - \mathbf{s}_3^{\mathsf{gm}})$$

is a MSIS solution for the matrix $\mathbf{A}$ [22]. Also, with high probability we have $\mathbf{s} \neq 0$ since sk^{gm} was chosen independently by the challenger. Now, we need to bound the norm of $\mathbf{s}$. In order to do so, we will use the property that for any $\mathbf{x} \in \mathcal{R}_p^{\tau N}$, $\|\mathbf{Rx}\| \leqslant s_1(\mathbf{R})\|\mathbf{x}\| \leqslant \psi\|\mathbf{x}\|$. Thus, we can bound the norm of $\mathbf{s}$ defined above using the Cauchy-Schwarz inequality as follows:

$$\begin{aligned}\|\mathbf{s}\| &\leqslant \|\bar{\mathbf{s}}_1 - \mathbf{s}_1^{\mathsf{gm}}\| + \psi\|\bar{\mathbf{s}}_2 - \mathbf{s}_2^{\mathsf{gm}}\| + \psi\|\bar{\mathbf{s}}_3 - \mathbf{s}_3^{\mathsf{gm}}\| \\ &\leqslant \sqrt{1 + \psi^2 + \psi^2} \cdot \sqrt{\|\bar{\mathbf{s}}_1 - \mathbf{s}_1^{\mathsf{gm}}\|^2 + \|\bar{\mathbf{s}}_2 - \mathbf{s}_2^{\mathsf{gm}}\|^2 + \|\bar{\mathbf{s}}_3 - \mathbf{s}_3^{\mathsf{gm}}\|^2}.\end{aligned}$$

Finally, we observe that we can bound the second term as:

$$\left\|\begin{bmatrix}\bar{\mathbf{s}}_1 - \mathbf{s}_1^{\mathsf{gm}}\\ \bar{\mathbf{s}}_2 - \mathbf{s}_2^{\mathsf{gm}}\\ \bar{\mathbf{s}}_3 - \mathbf{s}_3^{\mathsf{gm}}\end{bmatrix}\right\|^2 \leqslant 2 \cdot \left(\left\|\begin{bmatrix}\bar{\mathbf{s}}_1\\ \bar{\mathbf{s}}_2\\ \bar{\mathbf{s}}_3\end{bmatrix}\right\|^2 + \left\|\begin{bmatrix}\mathbf{s}_1^{\mathsf{gm}}\\ \mathbf{s}_2^{\mathsf{gm}}\\ \mathbf{s}_3^{\mathsf{gm}}\end{bmatrix}\right\|^2\right) \leqslant 4B^2 = (2B)^2.$$

Hence

$$\|\mathbf{s}\| \leqslant B_{\mathsf{MSIS}} := 2\mathfrak{s} \cdot \sqrt{1 + 2\psi^2} \cdot \sqrt{2((2\tau+1)N + M)kd}.$$

Thus we have to choose N such that $\mathsf{MSIS}_{N,N+M,B_{\mathsf{MSIS}}}$ is hard over $\mathcal{R}_{kd,p}$ and take into account the $1/|G|$ security loss. Not to mention the fact that we want $\mathbf{AR}$ to be computationally indistinguishable from a random matrix $\mathbf{B}$, i.e. the $\mathsf{MLWE}_{N,M,S_{kd,1}}$ problem over $\mathcal{R}_{kd,p}$ to be hard.

Parameters for the ABDLOP commitment are chosen similarly as in the previous examples. In particular, the proof system modulus q has to be large enough to prove exactly that the norm of a user secret key is at most $B = \mathfrak{s}\sqrt{2((2\tau+1)N + M)kd}$. Also, we aim for repetition rate 7 as in the previous examples.

[21] Hence, there is a $1/|G|$ security loss.

[22] Since we prove the norm of sk_{i^*} exactly, there is no relaxation factor c in front of the vector $\mathbf{u}$ as in previous works.

parameters	description	value
p	modulus for the group signature	$2^{38} - 107$
d	ring dimension for of $\mathcal{R}$	128
k	kd is the ring dimension of $\mathcal{R}_{kd}$	4
N	height of the $\mathbf{A}$ matrix	2
M	$N + M$ is the width of the $\mathbf{A}$ matrix	3
τ	τN is the width of the gadget matrix $\mathbf{G}$	5
ω	#1's in the identity $i \in G$	4
$\|G\|$	size of the identity space	$\approx 2^{23}$
p_{enc}	encryption modulus	3329
N_{enc}	height of $\mathbf{A}_{\mathsf{enc}}$	4
K_{enc}	width of $\mathbf{A}_{\mathsf{enc}}$	9
ξ_{enc}	ξ_{enc}^K is the randomness distribution of $\mathbf{r}_{\mathsf{enc}}$	Bin_2^d
q	modulus for the proof system	$\approx 2^{64}$
l	# factors $X^d + 1$ splits into mod q	2
γ_1	rejection sampling constant for $c\mathbf{s}_1$	17
γ_2	rejection sampling constant for $c\mathbf{s}_2$	1.2
$\gamma^{(e)}$	rejection sampling constant exact ARP	2.5
$\gamma^{(d)}$	rejection sampling constant for non-exact ARP	12
κ	maximum coefficient of a challenge in $\mathcal{C}$	2
n	height of matrices $\mathbf{A}_1, \mathbf{A}_2$ in ABDLOP	12
m_1	length of the message $\mathbf{s}_1$ in the "Ajtai" part	110
ℓ	length of the message $\mathbf{m}$ in the "BDLOP" part	0
λ	$2 \cdot$ (# of garbage g_j for soundness)	6
m_2	length of the randomness $\mathbf{s}_2$ in ABDLOP	41
ν	randomness $\mathbf{s}_2$ is sampled from $S_\nu^{m_2}$	1
γ	parameter to cut low-order bits from $\mathbf{w}$	$\approx 2^{37}$
D	number of low-order bits cut from $\mathbf{t}_A$	29
	repetition rate	7
	extra cost of adding verifiable encryption	6.5KB
	signature size	92KB
	public key size	47.5KB
	secret key size	6.3KB

Fig. 16: Parameter selection and concrete sizes for the group signature scheme.

6.5 Product Proofs over $\mathcal{R}_p$ for a co-prime p

Another application of our techniques is a product proof over $\mathcal{R}_p$ where $p < q$ is co-prime to our proof system modulus q. Namely, suppose we want to prove n equations of the form:

$$a_i b_i = c_i \text{ for } i = 1, 2, \ldots, n \tag{78}$$

where all $a_i, b_i, c_i \in \mathcal{R}_p$.

Note that if p was a divisor of q, i.e. $q = kp$ for some integer k, then we would simply apply the methodology from Section 4.2 to prove $ka_ib_i = kc_i$ over $\mathcal{R}_q$. This immediately implies (78).

There are two fundamental reasons why we would consider proving such statements. Firstly, this allows us to efficiently prove quadratic relations when p is small. Indeed, suppose that we choose p which is divisible by q. Recall that the soundness error of the protocols in Section 4 mainly depends on the smallest prime divisor of q, i.e. $q_1 \leqslant p$. Hence, if we wish to have small p, we would need to decrease the number of subfields l that $\mathcal{R}_q$ splits into (so that $p^{-d/l}$ is negligible). Moreover, if we additionally want to execute the protocol in Fig. 8, e.g. in order to prove binary or L_2 norms, we would need to increase the number of garbage terms $g_1, \ldots, g_\lambda$ so that $p^{-\lambda}$ is negligible. This, unfortunately, has a negative impact on the overall communication size.

The second reason is that, for suitable primes p, we could prove point-wise product relations $\vec{a} \circ \vec{b} = \vec{c}$ over $\mathbb{Z}_p$ which is a fundamental component in proving general circuit satisfiability and R1CS statements [BCR$^+$19]. Indeed, if we choose p such that $X^d + 1$ splits into linear factors modulo p, then using Number Theory Transform identically as done in [ALS20, ENS20], we reduce the problem to proving products over $\mathcal{R}_p$.

We first provide a naive strategy for proving (78). Namely, we commit to a_i, b_i, c_i using the ABDLOP commitment [23] and prove that the L_2 norms of each polynomial is at most $p\sqrt{d}/2$. Then, we commit to each

$$k_i := \frac{a_i b_i - c_i}{p}$$

and prove that $\|k_i\| \leqslant (pd + 2)\sqrt{d}/4$. Finally, we prove quadratic equations

$$a_i b_i - c_i = p k_i \tag{79}$$

over $\mathcal{R}_q$.

The intuition for soundness is that if we proved that a_i, b_i, c_i and k_i have small coefficients with respect to q, and that (79) holds over $\mathcal{R}_q$, then this implies that $a_i b_i - c_i = p k_i$ is true over integers since no modulo wrap-around occurs[24]. Consequently, we get $a_i b_i = c_i$ over $\mathcal{R}_p$.

Unfortunately, the cost of this method is committing to additional k_i for each out of n equations. We circumvent this issue by not committing to k_i but instead proving that $p^{-1}(a_i b_i - c_i) \in \mathcal{R}_q$ has small coefficients. As described before, we do that by committing to the masking polynomials $(y_1, \ldots, y_{256/d}) \in \mathcal{R}_q^{256/d}$ and computing $(z_1, \ldots, z_{256/d}) \in \mathcal{R}_q^{256/d}$ such that

$$\begin{bmatrix} \vec{z}_1 \\ \vdots \\ \vec{z}_{256/d} \end{bmatrix} := R \begin{bmatrix} p^{-1}(\overrightarrow{a_1 b_1} - \vec{c_1}) \\ \vdots \\ p^{-1}(\overrightarrow{a_n b_n} - \vec{c_n}) \end{bmatrix} + \begin{bmatrix} \vec{y}_1 \\ \vdots \\ \vec{y}_{256/d} \end{bmatrix}$$

where R is a challenge matrix and $\overrightarrow{a_i b_i}$ is a coefficient vector of $a_i b_i \in \mathcal{R}_q$ [25]. Then, we need to prove that polynomials z_i were well-formed.

Let us focus on the constant coefficient $\tilde{z}_1 \in \mathbb{Z}_q$ of z_1 since proving all the other ones follows identically. Then, if we denote the first row of R by $(r_1, \ldots, r_n) \in \mathcal{R}_q^n$, we have:

$$\tilde{z}_1 = p^{-1} \sum_{i=1}^{n} \vec{r}_i^T (\overrightarrow{a_i b_i} - \vec{c_i}) + \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix} \vec{y}_1.$$

Hence, we simply need to prove that the constant coefficient of

$$p^{-1} \sum_{i=1}^{n} \sigma_{-1}(r_i)(a_i b_i - c_i) + y_1 - z_1$$

is equal to zero. Note that all a_i, b_i, c_i and y_1 are committed. Hence, this is a quadratic relation with an automorphism and thus we can apply the protocol in Fig. 8 to prove this property.

7 Working Over General Rings

Throughout the paper, we have focused on working over the polynomial ring $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$, and in particular used the fact that $s(X^{-1})$ is an automorphism in this ring. In this section, we explain how our

[23] Hence, each coefficient of a_i, b_i, c_i is between $-p/2$ and $p/2$.

[24] This strategy was already used to prove integer multiplication in [LNS20].

[25] For simplicity, we omit bimodal rejection sampling which would end up having to prove cubic rather than quadratic equations.

main results can be generalized to virtually any other ring that one could be interested in. In particular, let us define $R = \mathbb{Z}[X]/(X^d + f_{d-1}X^{d-1} + f_{d-2}X^{d-2} + \ldots + f_1X \pm 1)$, where $f_i \in \mathbb{Z}$.

The first thing to note is that all our protocols for proving linear a quadratic relations over $\mathcal{R}_q$ did not use any special properties of the ring except that the challenge differences need to be invertible. For purposes of security, one should also be mindful of the "expansion factor" of the ring, which controls the growth of polynomial products in the ring – if it is too big, then the reductions involving (Ring/Module)-SIS become meaningless [LM06].

For our proofs over the ring R_q to be meaningfully applied to proving knowledge of inner products over $\mathbb{Z}$, one needs a correspondence between the inner product and the constant coefficient of a polynomial multiplication. Below, we show how one can achieve such a correspondence for any R. The multiplication of $a \cdot b$ in the ring R can be written as a matrix-vector product $A\vec{b}$, where $\vec{b}$ consists of the coefficients of b and the i^{th} column of A (if we number them starting from 0) consists of vectors whose elements are the coefficients of the polynomial $a \cdot X^i \in R$. It's not hard to see that the first row of A is the vector $\vec{a}^T \cdot M$, where

$$M = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & \pm 1 \\ 0 & 0 & 0 & \ldots & \pm 1 & c_{2,d-1} \\ \ldots\ldots & & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \pm 1 & \ldots & c_{d-2,d-2} & c_{d-2,d-1} \\ 0 & \pm 1 & c_{d-1,2} & \ldots & c_{d-1,d-2} & c_{d-1,d-1} \end{bmatrix}, \tag{80}$$

for some integers $c_{i,j}$ which are of no particular importance to this section. Therefore the inner product $\langle \vec{r}, \vec{s} \rangle$ is equal to $\widetilde{a \cdot s}$ where $\vec{a}^T \cdot M = \vec{r}^T$. Since the determinant of M is ± 1, M^{-1} is also an integer matrix, and thus $\vec{a}^T = \vec{r}^T \cdot M^{-1}$ is an integer vector and so $a \in R$.

The protocol for proving a bound on $\|s\|^2$ over the ring $\mathcal{R}_q$, uses the fact that the matrix M^{-1} actually corresponds to an automorphism over $\mathcal{R}_q$, and so the prover does not need to create a commitment to both s and $\vec{s}^T \cdot M^{-1}$ – the verifier can essentially derive the latter by himself. In rings where $\vec{s}^T \cdot M^{-1}$ is not an automorphism, the prover would additionally need to commit to the polynomial corresponding to $\vec{r} = \vec{s}^T \cdot M^{-1}$, and then give a linear proof showing that this relationship is indeed satisfied, along with the proof on the bound of $\|s\|^2 = \widetilde{r \cdot s}$. The modification for proving that s contains only 0/1 coefficients would proceed in the same manner. Proving component-wise products over general rings R can also be done, but ends up again doubling the committed vector. Recall that the idea when working over the ring $\mathcal{R}_q$ was to pick a prime $p \ll q$ such that $X^d + 1$ fully splits modulo p and then embed the coefficients into the CRT slots. If, for a particular ring R, there is no such p, then one would need to use a different ring than the one used for the commitment scheme which does have such a p, and make sure that multiplication of committed values over this ring corresponds to the one used in the commitment scheme. One way to do this is to only commit to polynomials of less than half the degree of the ring, so that multiplications in both rings is the same as over $\mathbb{Z}[X]$.

References

ABB10. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, pages 553–572, 2010.

ACK21. Thomas Attema, Ronald Cramer, and Lisa Kohl. A compressed \$\varsigma \$-protocol theory for lattices. In *CRYPTO (2)*, volume 12826 of *Lecture Notes in Computer Science*, pages 549–579. Springer, 2021.

AFK21. Thomas Attema, Serge Fehr, and Michael Klooß. Fiat-shamir transformation of multi-round interactive proofs. Cryptology ePrint Archive, Paper 2021/1377, 2021. `https://eprint.iacr.org/2021/1377`.

Ajt96. Miklós Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108, 1996.

ALS20. Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler. Practical product proofs for lattice commitments. In *CRYPTO (2)*, volume 12171 of *Lecture Notes in Computer Science*, pages 470–499. Springer, 2020.

APS15. Martin R Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

Ban93. Wojciech Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, Dec 1993.

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.

BCS16. Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. Interactive oracle proofs. In *TCC (B2)*, volume 9986 of *Lecture Notes in Computer Science*, pages 31–60, 2016.

BDK$^+$18. Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - kyber: A cca-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy, EuroS&P*, pages 353–367, 2018.

BDK$^+$21. Ward Beullens, Samuel Dobson, Shuichi Katsumata, Yi-Fu Lai, and Federico Pintore. Group signatures and more from isogenies and lattices: Generic, simple, and efficient. *IACR Cryptol. ePrint Arch.*, page 1366, 2021.

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.

Beu20. Ward Beullens. Sigma protocols for mq, PKP and sis, and fishy signature schemes. In *EUROCRYPT (3)*, volume 12107 of *Lecture Notes in Computer Science*, pages 183–211. Springer, 2020.

BG14. Shi Bai and Steven D. Galbraith. An improved compression technique for signatures based on learning with errors. In *CT-RSA*, pages 28–47, 2014.

BL17. Carsten Baum and Vadim Lyubashevsky. Simple amortized proofs of shortness for linear relations over polynomial rings. *IACR Cryptology ePrint Archive*, 2017:759, 2017.

BLS19. Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. Algebraic techniques for short(er) exact lattice-based zero-knowledge proofs. In *CRYPTO (1)*, volume 11692 of *Lecture Notes in Computer Science*, pages 176–202. Springer, 2019.

BN20. Carsten Baum and Ariel Nof. Concretely-efficient zero-knowledge arguments for arithmetic circuits and their application to lattice-based cryptography. In *Public Key Cryptography (1)*, pages 495–526. Springer, 2020.

BR93. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS*, pages 62–73. ACM, 1993.

CLOS02. Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *STOC*, pages 494–503. ACM, 2002.

DDLL13. Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In *CRYPTO (1)*, pages 40–56, 2013.

DKL$^+$18. Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):238–268, 2018.

DKRV18. Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. Saber: Module-lwr based key exchange, cpa-secure encryption and cca-secure KEM. In *AFRICACRYPT*, pages 282–305, 2018.

DLL$^+$17. Leo Ducas, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehle. Crystals – dilithium: Digital signatures from module lattices. Cryptology ePrint Archive, Report 2017/633, 2017. `https://ia.cr/2017/633`. "The Dilithium-G" scheme can be found in the June 2017 version of this report. The direct url is `https://eprint.iacr.org/eprint-bin/getfile.pl?entry=2017/633&version=20170627:201152&file=633.pdf`.

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

ENS20. Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In *ASIACRYPT (2)*, pages 259–288, 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)*, pages 115–146. Springer, 2019.

ESLR22. Muhammed F. Esgin, Ron Steinfeld, Dongxi Liu, and Sushmita Ruj. Efficient hybrid exact/relaxed lattice proofs and applications to rounding and vrfs. *IACR Cryptol. ePrint Arch.*, page 141, 2022.

ESZ21. Muhammed F. Esgin, Ron Steinfeld, and Raymond K. Zhao. Matrict+: More efficient post-quantum private blockchain payments. *IACR Cryptol. ePrint Arch.*, page 545, 2021.

EZS$^+$19. Muhammed F. Esgin, Raymond K. Zhao, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. Matrict: Efficient, scalable and post-quantum blockchain confidential transactions protocol. In *CCS*, pages 567–584. ACM, 2019.

FS86. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO*, pages 186–194, 1986.

GHL21. Craig Gentry, Shai Halevi, and Vadim Lyubashevsky. Practical non-interactive publicly verifiable secret sharing with thousands of parties. *IACR Cryptol. ePrint Arch.*, page 1397, 2021.

GN08. Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In *EUROCRYPT*, pages 31–51, 2008.

GT21. Ashrujit Ghoshal and Stefano Tessaro. Tight state-restoration soundness in the algebraic group model. In *CRYPTO (3)*, volume 12827 of *Lecture Notes in Computer Science*, pages 64–93. Springer, 2021.

HPS98. Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS*, pages 267–288, 1998.

LLNW16. Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In *EUROCRYPT (2)*, pages 1–31. Springer, 2016.

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

LN17. Vadim Lyubashevsky and Gregory Neven. One-shot verifiable encryption from lattices. In *EUROCRYPT*, 2017.

LNPS21. Vadim Lyubashevsky, Ngoc Khanh Nguyen, Maxime Plançon, and Gregor Seiler. Shorter lattice-based group signatures via ”almost free” encryption and other optimizations. In *ASIACRYPT (4)*, pages 218–248. Springer, 2021.

LNS20. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Practical lattice-based zero-knowledge proofs for integer relations. In *CCS*, pages 1051–1070. ACM, 2020.

LNS21a. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. In *Public Key Cryptography (1)*, pages 215–241. Springer, 2021.

LNS21b. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. SMILE: set membership from ideal lattices with applications to ring signatures and confidential transactions. In *CRYPTO (2)*, volume 12826 of *Lecture Notes in Computer Science*, pages 611–640. Springer, 2021.

LNSW13. San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *PKC*, pages 107–124, 2013.

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

LPS10. Vadim Lyubashevsky, Adriana Palacio, and Gil Segev. Public-key cryptographic primitives provably as secure as subset sum. In *TCC*, pages 382–400, 2010.

LS15. Adeline Langlois and Damien Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and 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.

MP12. Daniele Micciancio and Chris Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, pages 700–718, 2012.

MR09. Daniele Micciancio and Oded Regev. Lattice-based cryptography. In *Post-quantum cryptography*, pages 147–191. Springer, 2009.

Reg09. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.

Sch89. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO*, pages 239–252, 1989.

Ste93. Jacques Stern. A new identification scheme based on syndrome decoding. In *CRYPTO*, pages 13–21, 1993.

The22. The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.3)*, 2022. `https://www.sagemath.org`.

Wik21. Douglas Wikström. Special soundness in the random oracle model. Cryptology ePrint Archive, Paper 2021/1265, 2021. https://eprint.iacr.org/2021/1265.

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)*, pages 147–175. Springer, 2019.

A Dilithium Compression

In this section, we reduce the commitment and communication size by applying compression techniques from Dilithium-G [DLL+17].

A.1 Low/High Order Bits

In order to reduce the size of the commitment, we need some algorithms that extract "higher-order" and "lower-order" bits of elements in $\mathbb{Z}_q$. The goal is that when given an arbitrary element $r \in \mathbb{Z}_q$ and another small element $z \in \mathbb{Z}_q$, we would like to be able to recover the higher order bits of $r + z$ without needing to store z. The algorithms are exactly as in [DLL+17], and we repeat them for completeness in Figure 17. They are described as working on integers modulo q, but one can extend it to (vectors of) polynomials in $\mathcal{R}_q$ by simply being applied individually to each coefficient.

```
$\mathsf{Power2Round}_q(r, D)$
$r := r \bmod^+ q$
$r_0 := r \bmod^\pm 2^D$
return $(r - r_0)/2^D$

$\mathsf{UseGHint}_q(y, r, \gamma)$
$m := (q-1)/\gamma$
$r_1 := \mathsf{HighBits}_q(r, \gamma)$
return $(r_1 + y) \bmod {}^\pm m$

$\mathsf{MakeGHint}_q(z, r, \gamma)$
$m = (q-1)/\gamma$
$r_1 := \mathsf{HighBits}_q(r, \gamma)$
$v_1 := \mathsf{HighBits}_q(r + z, \gamma)$
return $(v_1 - r_1) \bmod {}^\pm m$

$\mathsf{Decompose}_q(r, \gamma)$
$r := r \bmod^+ q$
$r_0 := r \bmod^\pm \gamma$
if $r - r_0 = q - 1$
    then $r_1 := 0$; $r_0 := r_0 - 1$
else $r_1 := (r - r_0)/\gamma$
return $(r_1, r_0)$

$\mathsf{HighBits}_q(r, \gamma)$
$(r_1, r_0) := \mathsf{Decompose}_q(r, \gamma)$
return $r_1$

$\mathsf{LowBits}_q(r, \gamma)$
$(r_1, r_0) := \mathsf{Decompose}_q(r, \gamma)$
return $r_0$
```

Fig. 17: Supporting algorithms for commitment compression.

Lemma A.1. *Suppose that q and γ are positive integers satisfying $q \equiv 1 \pmod{\gamma}$. Fix $m := (q-1)/\gamma$. Let $\mathbf{r}$ and $\mathbf{z}$ be vectors of elements in R_q where $\|\mathbf{z}\|_\infty \leqslant \gamma/2$, and let $\mathbf{y}, \mathbf{y}'$ be integral vectors of elements in $(-m/2, m/2]$. Then the $\mathsf{HighBits}_q$, $\mathsf{MakeGHint}_q$, and $\mathsf{UseGHint}_q$ algorithms satisfy the following properties:*

1. $\mathsf{UseGHint}_q(\mathsf{MakeGHint}_q(\mathbf{z}, \mathbf{r}, \gamma), \mathbf{r}, \gamma) = \mathsf{HighBits}_q(\mathbf{r} + \mathbf{z}, \gamma)$.
2. *If* $\mathsf{UseGHint}_q(\mathbf{y}, \mathbf{r}, \gamma) = \mathsf{UseGHint}_q(\mathbf{y}', \mathbf{r}, \gamma)$*, then* $\mathbf{y} = \mathbf{y}'$*.*

Private information: $(\mathbf{s}_1, \mathbf{m}, \mathbf{s}_2) \in \mathcal{R}_q^{m_1+m_2+\ell}$ so that $\|\mathbf{s}_1\| \leqslant \alpha$ and $\|\mathbf{s}_2\|_\infty \leqslant \nu$

Public information: $\mathbf{A}_1 \in \mathcal{R}_q^{n \times m_1}, \mathbf{A}_2' \in \mathcal{R}_q^{n\times(m_2-n)}, \mathbf{B}' \in \mathcal{R}_q^{\ell\times(m_2-n)}$,

$\begin{bmatrix}\mathbf{t}_A \\ \mathbf{t}_B\end{bmatrix} = \begin{bmatrix}\mathbf{A}_1 \\ \mathbf{0}\end{bmatrix} \cdot \mathbf{s}_1 + \begin{bmatrix}\mathbf{A}_2' \\ \mathbf{B}'\end{bmatrix} \cdot \mathbf{s}_{2,1} + \begin{bmatrix}\mathbf{s}_{2,2} \\ \mathbf{m}\end{bmatrix}$

<u>Prover</u> <u>Verifier</u>

```
$\mathbf{y}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1}$
$\mathbf{y}_{2,1} \leftarrow D_{\mathfrak{s}_2}^{m_2-n}$
$\mathbf{y}_{2,2} \leftarrow D_{\mathfrak{s}_2}^{n}$
$\mathbf{w} := \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2'\mathbf{y}_{2,1} + \mathbf{y}_{2,2}$
$(\mathbf{w}_1, \mathbf{w}_0) := \mathsf{Decompose}_q(\mathbf{w}, \gamma)$
                                   $\xrightarrow{\mathbf{w}_1}$
                                   $\xleftarrow{c}$          $c \leftarrow \mathcal{C}$
$\mathbf{z}_1 := \mathbf{y}_1 + c\mathbf{s}_1$
$\mathbf{z}_2 = \begin{bmatrix}\mathbf{z}_{2,1} \\ \mathbf{z}_{2,2}\end{bmatrix} := \begin{bmatrix}\mathbf{y}_{2,1} \\ \mathbf{y}_{2,2}\end{bmatrix} + c\begin{bmatrix}\mathbf{s}_{2,1} \\ \mathbf{s}_{2,2}\end{bmatrix}$
for $i = 1, 2$ :
  if $\mathsf{Rej}_i(\mathbf{z}_i, c\mathbf{s}_i, \mathfrak{s}_i) = 1$
    then $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{h}) := (\perp, \perp, \perp)$
$\mathbf{z}_{2,2} := \mathbf{z}_{2,2} - c\mathbf{t}_{A,2} - \mathbf{w}_0$
if $\|(\mathbf{z}_{2,1}, \mathbf{z}_{2,2})\| > B$
  then $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{h}) := (\perp, \perp, \perp)$
$\mathbf{h} := \mathsf{MakeGHint}_q(\mathbf{z}_{2,2}, \gamma\mathbf{w}_1 - \mathbf{z}_{2,2}, \gamma)$
                                   $\xrightarrow{\mathbf{z}_1, \mathbf{z}_{2,1}, \mathbf{h}}$
                                             $\mathbf{w}_1 \overset{?}{=} \mathsf{UseGHint}_q\left(\mathbf{h}, \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2'\mathbf{z}_{2,1} - c \cdot 2^D\boldsymbol{t}_{A,1}, \gamma\right)$
                                             $\|\mathbf{z}_1\| \overset{?}{\leqslant} \mathfrak{s}_1\sqrt{2m_1d}$
                                             $\|(\mathbf{z}_{2,1}, \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2'\mathbf{z}_{2,1} - c \cdot 2^D\boldsymbol{t}_{A,1} - \gamma\mathbf{w}_1)\| \overset{?}{\leqslant} B$
                                             $\|\mathbf{h}\|_\infty \overset{?}{\leqslant} \frac{q-1}{2\gamma}$
```

Fig. 18: Proof of knowledge of $(\mathbf{s}_1, \mathbf{s}_2 := (\mathbf{s}_{2,1}, \mathbf{s}_{2,2}), \mathbf{m}, \bar{c}) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \mathcal{R}_q^{\ell} \times \bar{\mathcal{C}}$ satisfying (i) $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2'\mathbf{s}_{2,1} + \mathbf{s}_{2,2} = 2^D \cdot \mathbf{t}_{A,1}$, $\mathbf{B}'\mathbf{s}_{2,1} + \mathbf{m} = \mathbf{t}_B$ (ii) $\|\mathbf{s}_1\bar{c}\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1d}$ and (iii) $\|\mathbf{s}_2\bar{c}\| \leqslant 2B$.

A.2 ABDLOP Commitment Compression

We apply the aforementioned compression techniques in the opening proof presented above. First, we reduce the size of the ABDLOP commitment by not sending the low-order bits of $\boldsymbol{t}_A$. Namely, for a suitable $D \in \mathbb{N}$ we write

$$\boldsymbol{t}_A = \boldsymbol{t}_{A,1} \cdot 2^D + \boldsymbol{t}_{A,2} \text{ where } \|\boldsymbol{t}_{A,2}\|_\infty \leqslant 2^{D-1}$$

and only send $\boldsymbol{t}_{A,1}$. Thus, we reduce the commitment size by Dnd bits.

Further, instead of sampling uniformly random matrices $\mathbf{A}_2$ and $\boldsymbol{B}$, we can choose them in the more structured way as originally in [BDL+18]

$$\begin{bmatrix}\mathbf{A}_2 \\ \boldsymbol{B}\end{bmatrix} := \begin{bmatrix}\mathbf{A}_2' & \boldsymbol{I}_n \\ \boldsymbol{B}' & \mathbf{0}_{\ell\times n}\end{bmatrix}\mathcal{R}_q^{(n+\ell)\times m_2}. \tag{81}$$

We present the ABDLOP opening proof in Fig. 18. Prover $\mathcal{P}$ starts by sampling vectors $\mathbf{y}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1}, \mathbf{y}_{2,1} \leftarrow D_{\mathfrak{s}_2}^{m_2-n}$ and $\mathbf{y}_{2,2} \leftarrow D_{\mathfrak{s}_2}^{n}$ from discrete Gaussians and computing $\mathbf{w} = \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2'\mathbf{y}_{2,1} + \mathbf{y}_{2,2}$. Additionally, $\mathcal{P}$ calculates $(\mathbf{w}_1, \mathbf{w}_0) = \mathsf{Decompose}_q(\mathbf{w}, 2\gamma)$ and sends $\mathbf{w}_1$ to the verifier where $q-1$ is divisible by γ.

After receiving a challenge polynomial $c \leftarrow \mathcal{C}$ from $\mathcal{V}$, the prover computes

$$\mathbf{z}_1 = \mathbf{y}_1 + c\mathbf{s}_1 \text{ and } \mathbf{z}_2 = \begin{bmatrix}\mathbf{z}_{2,1} \\ \mathbf{z}_{2,2}\end{bmatrix} := \begin{bmatrix}\mathbf{y}_{2,1} \\ \mathbf{y}_{2,2}\end{bmatrix} + c\begin{bmatrix}\mathbf{s}_{2,1} \\ \mathbf{s}_{2,2}\end{bmatrix}$$

and applies rejection sampling for $\mathbf{z}_1$ and $\mathbf{z}_2$. If it accepts, $\mathcal{P}$ modifies $\mathbf{z}_{2,2} := \mathbf{z}_{2,2} - c\mathbf{t}_{A,2} - \mathbf{w}_0$ and calculates the hint vector $\mathbf{h} = \mathsf{MakeGHint}_q\left(\mathbf{z}_{2,2}, \gamma\mathbf{w}_1 - \mathbf{z}_{2,2}, \gamma\right)$. Finally, the prover sends $(\mathbf{z}_1, \mathbf{z}_{2,1}, \mathbf{h})$. In the last stage, verifier $\mathcal{V}$ checks whether vectors $\mathbf{z}_1$ and $(\mathbf{z}_{2,1}, \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}'_2\mathbf{z}_{2,1} - c \cdot 2^D\boldsymbol{t}_{A,1} - \gamma\mathbf{w}_1)$ has small norm and the coefficients of $\mathbf{h}$ are between $-\frac{q-1}{2\gamma}$ and $\frac{q-1}{2\gamma}$ and

$$\mathbf{w}_1 \stackrel{?}{=} \mathsf{UseHint}_q\left(\mathbf{h}, \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}'_2\mathbf{z}_{2,1} - c \cdot 2^D\boldsymbol{t}_{A,1}, \gamma\right).$$

As opposed to the standard opening proof, the prover does not send any masked opening of $\boldsymbol{s}_{2,2}$. Instead, $\mathcal{P}$ sends a vector of hints $\mathbf{h}$ which has much smaller impact on the communication size as opposed to $\mathbf{z}_{2,2}$.

Theorem A.2. *Let $\mathfrak{s}_1 = \gamma_1\eta\alpha$ and $\mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d}$. Then, the protocol in Fig. 18 is a zero-knowledge proof of knowledge.*

For completeness, let $m_1, m_2 \geqslant 640/d$, γ be an even divisor of $q-1$ and B be defined as

$$B = \mathfrak{s}_2\sqrt{2m_2 d} + \eta 2^{D-1}\sqrt{nd} + \frac{\gamma\sqrt{nd}}{2}.$$

Then, the honest prover $\mathcal{P}$ convinces the honest verifier $\mathcal{V}$ with probability

$$\frac{1}{2\exp\left(\frac{14}{\gamma_1} + \frac{1}{2\gamma_1^2} + \frac{1}{2\gamma_2^2}\right)}.$$

For soundness, there is an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a probabilistic prover $\mathcal{P}^*$, which convinces $\mathcal{V}$ with probability $\varepsilon \geqslant 1/|\mathcal{C}|$, extractor $\mathcal{E}$ with probability at least $\varepsilon - 1/|\mathcal{C}|$ outputs $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2 := (\bar{\mathbf{s}}_{2,1} \parallel \bar{\mathbf{s}}_{2,2}), \bar{\mathbf{m}}, \bar{c}) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \mathcal{R}_q^{\ell} \times \bar{\mathcal{C}}$ satisfying

$$\mathbf{A}_1\bar{\mathbf{s}}_1 + \mathbf{A}'_2\bar{\mathbf{s}}_{2,1} + \bar{\mathbf{s}}_{2,2} = 2^D \cdot \mathbf{t}_{A,1}, \quad \mathbf{B}'\bar{\mathbf{s}}_{2,1} + \bar{\mathbf{m}} = \mathbf{t}_B, \quad \|\bar{\mathbf{s}}_1\bar{c}\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1 d}, \quad \|\bar{\mathbf{s}}_2\bar{c}\| \leqslant 2B.$$

Proof. Since zero-knowledge follows from the standard techniques, we only focus on correctness and knowledge soundness.

Correctness. First, if the rejection sampling steps pass, the distributions of $\mathbf{z}_1, \mathbf{z}_2$ are discrete Gaussians centered at 0 with standard deviations $\mathfrak{s}_1$ and $\mathfrak{s}_2$ respectively (though the latter one is conditioned on $\langle \mathbf{z}_2, c\mathbf{s}_2 \rangle \geqslant 0$). Since $m_1 d, m_2 d \geqslant 640$, we have that

$$\Pr_{\mathbf{z}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1}}\left[\|\mathbf{z}_1\| \leqslant \mathfrak{s}_1\sqrt{2m_1 d}\right] \geqslant 1 - 2^{-141}$$

and

$$\Pr_{\mathbf{z}_2 \leftarrow D_{\mathfrak{s}_2}^{m_2}}\left[\|\mathbf{z}_2\| \leqslant \mathfrak{s}_2\sqrt{2m_2 d} \mid \langle \mathbf{z}_2, c\mathbf{s}_2 \rangle \geqslant 0\right] \geqslant 1 - \frac{\Pr_{\mathbf{z}_2 \leftarrow D_{\mathfrak{s}_2}^{m_2}}\left[\|\mathbf{z}_2\| > \mathfrak{s}_2\sqrt{2m_2 d}\right]}{\Pr_{\mathbf{z}_2 \leftarrow D_{\mathfrak{s}_2}^{m_2}}\left[\langle \mathbf{z}_2, c\mathbf{s}_2 \rangle \geqslant 0\right]} \geqslant 1 - 2^{-140}$$

by Lemma 2.2 for $t = \sqrt{2}$. Now, since we perturb the vector $\mathbf{z}_{2,2}$, the bound on $\|\mathbf{z}_2\|$ increases slightly. Using the inequalities $\|c\mathbf{t}_{A,2}\| \leqslant \eta\|\mathbf{t}_{A,2}\| = \eta 2^{D-1}\sqrt{nd}$ and $\|\mathbf{w}_0\| \leqslant \gamma\sqrt{nd}/2$, we get

$$\left\|\begin{bmatrix}\mathbf{z}_{2,1} \\ \mathbf{z}_{2,2} - c\mathbf{t}_{A,2} - \mathbf{w}_0\end{bmatrix}\right\| \leqslant \left\|\begin{bmatrix}\mathbf{z}_{2,1} \\ \mathbf{z}_{2,2}\end{bmatrix}\right\| + \left\|\begin{bmatrix}\mathbf{0} \\ c\mathbf{t}_{A,2}\end{bmatrix}\right\| + \left\|\begin{bmatrix}\mathbf{0} \\ \mathbf{w}_0\end{bmatrix}\right\| \leqslant \mathfrak{s}_2\sqrt{2m_2 d} + \eta 2^{D-1}\sqrt{nd} + \frac{\gamma\sqrt{nd}}{2} = B.$$

The verification equation on $\mathbf{h}$ follows by definition of $\mathsf{MakeGHint}$. Finally, note that

$$\begin{aligned}\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}'_2\mathbf{z}_{2,1} + \mathbf{z}_{2,2} &= c2^D\mathbf{t}_{A,1} + \mathbf{w} - \mathbf{w}_0 \\ &= c2^D\mathbf{t}_{A,1} + \gamma\mathbf{w}_1\end{aligned}$$

and thus

$$\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2'\mathbf{z}_{2,1} - c2^D\mathbf{t}_{A,1} = \gamma\mathbf{w}_1 - \mathbf{z}_{2,2}.$$

Consequently, by Lemma A.1:

$$\begin{aligned}\mathsf{UseGHint}_q\left(\mathbf{h}, \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2'\mathbf{z}_{2,1} - c\cdot 2^D\boldsymbol{t}_{A,1}, \gamma\right) &= \mathsf{UseGHint}_q\left(\mathsf{MakeGHint}_q\left(\mathbf{z}_{2,2}, \gamma\mathbf{w}_1 - \mathbf{z}_{2,2}, \gamma\right), \gamma\mathbf{w}_1 - \mathbf{z}_{2,2}, \gamma\right)\\ &= \mathsf{HighBits}_q(\gamma\mathbf{w}_1, \gamma)\\ &= \mathbf{w}_1.\end{aligned}$$

Knowledge Soundness. Let $\mathcal{P}^*$ be a probabilistic prover which convinces the verifier with probability $\varepsilon > |\mathcal{C}|^{-1}$. Then, by [ACK21][Lemma 4] there is an algorithm $\mathcal{E}$ which extracts two accepting transcripts with the same first message $\mathbf{w}_1$ and distinct challenges with probability at least $\epsilon - 1/|\mathcal{C}|$:

$$\mathsf{tr}_i = \left(\mathbf{w}_1, c^{(i)}, \mathbf{z}_1^{(i)}, \mathbf{z}_{2,1}^{(i)}, \mathbf{h}^{(i)}\right) \text{ for } i = 0, 1.$$

Let us define $\bar{c} := c^{(1)} - c^{(0)} \in \bar{\mathcal{C}}$. Note that by definition of the challenge space, $\bar{c}$ is invertible over $\mathcal{R}_q$ and $\|\bar{c}\|_\infty \leqslant 2\kappa$. Let us define

$$\mathbf{u}^{(i)} := \gamma\mathbf{w}_1 + c^{(i)}\cdot 2^D\boldsymbol{t}_{A,1} - \mathbf{A}_1\mathbf{z}_1^{(i)} - \mathbf{A}_2'\mathbf{z}_{2,1}^{(i)}.$$

Then, we have $\|(\mathbf{z}_{2,1}^{(i)}, \mathbf{u}^{(i)})\| \leqslant B$ for $i = 0, 1$. Then, by combining the two equations on $\mathbf{u}^{(i)}$ we get

$$\mathbf{A}_1(\mathbf{z}_1^{(1)} - \mathbf{z}_1^{(0)}) + \mathbf{A}_2'(\mathbf{z}_2^{(1)} - \mathbf{z}_2^{(0)}) + (\mathbf{u}^{(1)} - \mathbf{u}^{(0)}) = \bar{c}\cdot 2^D\boldsymbol{t}_{A,1}.$$

Next, we set

$$\bar{\mathbf{s}}_1 := \frac{\mathbf{z}_1^{(1)} - \mathbf{z}_1^{(0)}}{\bar{c}}, \quad \bar{\mathbf{s}}_2 = \begin{bmatrix}\bar{\mathbf{s}}_{2,1}\\ \bar{\mathbf{s}}_{2,2}\end{bmatrix} := \frac{1}{\bar{c}}\cdot\begin{bmatrix}\mathbf{z}_{2,1}^{(1)} - \mathbf{z}_{2,1}^{(0)}\\ \mathbf{u}^{(1)} - \mathbf{u}^{(0)}\end{bmatrix}, \quad \bar{\mathbf{m}} := \mathbf{t}_B - \mathbf{B}\bar{\mathbf{s}}.$$

Then, by construction $\|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ and $\|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2B$. □

B Security in the Random Oracle Model

All prior efficient lattice-based *interactive* proofs, e.g. [ALS20, ENS20, LNS21a], can be made non-interactive using the Fiat-Shamir transformation [FS86] and proven secure in the random oracle model [BR93]. Indeed, even though the original transformation was applied to Σ-protocols, i.e. public-coin three-round protocols, they can also be applied to multi-round protocols. However, the main bottleneck in the $(2\mu + 1)$-round[26] case, is that, in general, the security loss obtained by applying the Fiat-Shamir transformation can be in the order of $O(Q^\mu)$ where Q is the number of random oracle queries made by an adversary. Note that $\mu > 1$ for all aforementioned exact proof systems since they are either five-round [BLS19, YAZ$^+$19, ENS20, LNS21a], seven-round [ESLR22] or even nine-round as in Section 5. Hence, even the case $\mu > 1$ might have a significant impact on the proof size, since one would need to aim for a much larger security level to accommodate for the loss. However, the schemes assume that in practice the security loss is much milder than the general bound $O(Q^\mu)$, and hence ignore it when setting up the parameters.

There have been several works on translating security properties of interactive protocols under the Fiat-Shamir transformation in the random oracle model, e.g. [BCS16, GT21, AFK21, Wik21]. The most recent results, which are the most relevant to lattice-based zero-knowledge proofs, were proposed independently by Attema et al. [AFK21] and Wikström [Wik21] who showed that the Fiat-Shamir transformation of any $(k_1, \ldots, k_\mu)$-special-sound interactive proofs has a security loss of at most $Q + 1$. Even though this result can be directly applied to some exact proofs systems, e.g. [BLS19, YAZ$^+$19], more efficient protocols, such as [ALS20, ENS20] or our protocol in Section 5, do not appear to be special-sound. Hence, it is still an open

[26] We assume that the first and last messages are sent from the prover.

problem whether these interactive proofs also incur linear loss in the number of random oracle queries when applying the Fiat-Shamir transformation.

In this section, we show that the Fiat-Shamir transformation of our interactive proof Π described in Section 5 admits a security loss of at most $Q+1$ in the random oracle model where Q is the number of random oracle queries made by an adversary. Namely, let κ be the knowledge error of Π, i.e. the maximal probability that a cheating prover can convince the verifier without having a valid witness for the claimed statement. Then, the Fiat-Shamir transformed protocol $\mathsf{FS}[\Pi]$ has knowledge error at most $(Q+1)\cdot\kappa$. To describe the core part of our techniques, we formalize the extraction strategy called "proof using a probabilistic argument" which, as far as we are aware, was first used in [ALS20]. Our methods will heavily rely on the *abstract sampling game* framework introduced by Attema et al. [AFK21].

We note that this section only focuses on the knowledge soundness aspect. Indeed, correctness holds directly by construction of the Fiat-Shamir transform, while non-interactive zero-knowledge follows from the fact that the interactive protocol itself is (non-abort) honest-verifier zero-knowledge [Lyu12].

B.1 Negative Multinomial Distribution

In our proofs, we will use certain properties of the following simplified negative multinomial distribution. Namely, consider a bucket containing a green, b blue and c red balls. In this experiment, ball are drawn uniformly at random with replacement until the first green ball has been found. Let Y be the number of blue balls drawn in this experiment which is denoted by $Y \sim \mathsf{NMN}(a,b,c)$. Then, we have the following result.

Lemma B.1. *Let $a, b \geqslant 1$ and $c \geqslant 0$ and $Y \sim \mathsf{NMN}(a,b,c)$. Then ,$\mathbb{E}[Y] = \frac{b}{a}$.*

Proof. By a counting argument, for $i \geqslant 0$ we have

$$\Pr[Y=i] = \sum_{k=0}^{\infty} \frac{\binom{i+k}{i} b^i c^k a}{(a+b+c)^{k+i+1}}.$$

We claim that

$$\sum_{k=0}^{\infty} \frac{\binom{i+k}{i} c^k (a+b)^{i+1}}{(a+b+c)^{k+i+1}} = 1$$

which directly implies that $\Pr[Y=i] = \frac{b^i a}{(a+b)^{i+1}}$. To this end, let $p := \frac{c}{a+b+c}$. Then

$$\begin{aligned}\sum_{k=0}^{\infty} \frac{\binom{i+k}{i} c^k (a+b)^{i+1}}{(a+b+c)^{k+i+1}} &= \sum_{k=0}^{\infty} \binom{i+k}{i} \cdot \left(\frac{c}{a+b+c}\right)^k \cdot \left(\frac{a+b}{a+b+c}\right)^{i+1} \\ &= \sum_{k=0}^{\infty} \binom{i+k}{i} \cdot p^k \cdot (1-p)^{i+1} \\ &= \sum_{k=0}^{\infty} f(k, i+1, 1-p) = 1\end{aligned}$$

where f is the probability mass function of the negative binomial distribution. Hence, we obtain that

$$\Pr[Y=i] = \frac{b^i a}{(a+b)^{i+1}} = \binom{i}{0} \cdot \frac{a}{a+b} \cdot \left(\frac{b}{a+b}\right)^i = f\left(i, 1, \frac{a}{a+b}\right).$$

Hence, using the formula for the expected value of the negative binomial distribution we get:

$$\mathbb{E}[Y] = \frac{\frac{b}{a+b}}{1-\frac{b}{a+b}} = \frac{b}{a}$$

which concludes the proof. □

B.2 Abstract Sampling Game Revisited

The key building block in analysing a knowledge extractor for our Fiat-Shamir transformed proof will be the abstract sampling game defined below. Given the access to a prover $\mathcal{P}^*$ attacking the non-interactive proof $\mathsf{FS}(\mathit{\Pi})$ in the random oracle model, our extractor will essentially play this game (in the recursive manner). Since the game itself is similar to [AFK21], we also adapt the notation from the aforementioned work.

Notation. Let h and $U_1, \ldots, U_h, R, N_1, \ldots, N_h$ be positive integers. Denote $\mathcal{N} := [N_1]^{U_1} \times \cdots \times [N_h]^{U_h}$. Further, let $\mathcal{X} \subseteq \{0,1\}^* \times \{0,1\}^*$ be a set. Looking ahead, h denotes the number of random oracles that a prover (or a subextractor) has access to and U_i (resp. N_i) is the cardinality of the domain (resp. co-domain) of the i-th random oracle[27]. Similarly, R is the size of the randomness space of a prover/sub-extractor. Next, $\mathcal{X}$ consists of *valid* witnesses along with auxiliary information.

The game considers a fixed $(U_1 + \ldots + U_h + 1)$-dimensional array M, where for all $r \in [R]$ and $\vec{j}_i \in [N_i]^{U_i}$, the entry $M(r, \vec{j}_1, \ldots, \vec{j}_h)$ is of the form $(\mathbb{v}, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z})$ where $\mathbb{v} \in \{0,1\}$ (indicates correctness), $\mathbb{h} \in [h]$ (specifies for which random oracle we want to fork), $\mathbb{i} \in [U_{\mathbb{h}}]$ (says for which query we will fork) and $\mathbb{s}, \mathbb{z} \in \{0,1\}^*$ (outputs a witness *candidate* along with auxiliary information). Looking ahead, we will treat M as an extractor of the sub-protocol which outputs the candidate witness. We say an entry of M is valid if $\mathbb{v} = 1$.

We will rely on the following property of the set $\mathcal{X}$ with respect to the array M. It basically says that if the extractor M outputs a witness $\mathbb{s}$ which is not valid, then by running the extractor again by forking the random oracle in an appropriate place, the probability that M again outputs $\mathbb{s}$ is at most κ.

Definition B.2. *We say that set $\mathcal{X}$ is κ-probabilistically-testable w.r.t. the array M if for any $(r, \vec{j}_1, \ldots, \vec{j}_h) \in [R] \times [N_1]^{U_1} \times \cdots \times [N_h]^{U_h}$ such that $M(r, \vec{j}_1, \ldots, \vec{j}_h) = (1, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z})$ and $(\mathbb{s}, \mathbb{z}) \notin \mathcal{X}$, we have*

$$\Pr_{j^* \leftarrow [N_{\mathbb{h}}]}\left[M(r, \vec{j}_1, \ldots, \vec{j}_{\mathbb{h}-1}, \vec{j}^*, \vec{j}_{\mathbb{h}+1}, \ldots, \vec{j}_h) = (1, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z}')\right] \leqslant \kappa$$

where $\vec{j}^* := (j_{\mathbb{h},1}, \ldots, j_{\mathbb{h},\mathbb{i}-1}, j^*, j_{\mathbb{h},\mathbb{i}+1}, \ldots, j_{\mathbb{h},U_{\mathbb{h}}}) \in [N_{\mathbb{h}}]^{U_{\mathbb{h}}}$.

Finally, for $\mathbb{h} \in [h]$ and $\mathbb{i} \in [U_{\mathbb{h}}]$, we define the function

$$a_{\mathbb{h},\mathbb{i}} : [R] \times \mathcal{N} \to \mathbb{N}_{\geqslant 0}, \quad (r, \vec{j}_1, \ldots, \vec{j}_h) \mapsto |\{j^* : M(r, \vec{j}_1, \ldots, \vec{j}_{\mathbb{h}-1}, \vec{j}^*, \vec{j}_{\mathbb{h}+1}, \ldots, \vec{j}_h) = (1, \mathbb{h}, \mathbb{i}, \cdot, \cdot)\}|$$

where $\vec{j}^* := (j_{\mathbb{h},1}, \ldots, j_{\mathbb{h},\mathbb{i}-1}, j^*, j_{\mathbb{h},\mathbb{i}+1}, \ldots, j_{\mathbb{h},U_{\mathbb{h}}})$.

Abstract Game. We formally define the game in Fig. 19 and summarise its key properties in Lemma B.3. We are interested in the following two cases when the abstract game does not abort. First case is when

Parameters: $h, U_1, \ldots, U_h, R, N_1, \ldots, N_h \in \mathbb{N}$ and the $(U_1 + \ldots + U_h + 1)$-dimensional array M.

1. Sample $(\mathsf{R}, \vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) \leftarrow [R] \times [N_1]^{U_1} \times \cdots \times [N_h]^{U_h}$. Let $M(\mathsf{R}, \vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) = (\mathsf{V}, \mathsf{H}, \mathsf{I}, \mathsf{S}, \mathsf{Z})$.
2. If $\mathsf{V} = 0$, return $\perp$.
3. Sample $\mathsf{J}^* \leftarrow [N_{\mathsf{H}}]$ and set $\vec{\mathsf{J}}^* := (\mathsf{J}_{\mathsf{H},1}, \ldots, \mathsf{J}_{\mathsf{H},\mathsf{I}-1}, \mathsf{J}^*, \mathsf{J}_{\mathsf{H},\mathsf{I}+1}, \ldots, \mathsf{J}_{\mathsf{H},U_{\mathsf{H}}})$.
4. $M(\mathsf{R}, \vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_{\mathsf{H}-1}, \vec{\mathsf{J}}^*, \vec{\mathsf{J}}_{\mathsf{H}+1}, \ldots, \vec{\mathsf{J}}_h) = (\mathsf{V}', \mathsf{H}', \mathsf{I}', \mathsf{S}', \mathsf{Z}')$.
5. If $\mathsf{V}' = 0$ or $(\mathsf{H}, \mathsf{I}) \neq (\mathsf{H}', \mathsf{I}')$, go to Step 3.
6. Return $(\mathsf{S}, \mathsf{Z}, \mathsf{S}', \mathsf{Z}')$.

Fig. 19: Abstract sampling game.

[27] To avoid technical difficulties, we restrict the domain of a random oracle from $\{0,1\}^*$ to $\{0,1\}^{\leqslant u}$ similarly as in [AFK21].

$\mathsf{S} \neq \mathsf{S}'$. In our applications, this will imply that we break the binding property of the underlying commitment scheme. The second one is when $\mathsf{S} = \mathsf{S}' \wedge (\mathsf{S}, \mathsf{Z}) \in \mathcal{X}$ which directly says that we obtained a valid witness. Finally, using the fact that $\mathcal{X}$ is κ-probabilistically-testable w.r.t. M, we can bound the probability of the last case occurring, i.e. $\mathsf{S} = \mathsf{S}' \wedge (\mathsf{S}, \mathsf{Z}) \notin \mathcal{X}$.

Lemma B.3. *Let $\mathcal{X} \subseteq \{0,1\}^* \times \{0,1\}^*$ be a κ-probabilistically-testable set w.r.t. the array M and consider the algorithm $\mathcal{A}$ described in Fig. 19. For $\mathbb{h} \in [h]$ and $\mathbb{i} \in [U_{\mathbb{h}}]$, denote the random variable $A_{\mathbb{h},\mathbb{i}} := a_{\mathbb{h},\mathbb{i}}(\mathsf{R}, \vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h)$ and let $P := \sum_{\mathbb{h}=1}^{h} \sum_{\mathbb{i}=1}^{U_{\mathbb{h}}} \Pr[A_{\mathbb{h},\mathbb{i}} > 0]$. Then, the expected number of entries sampled by $\mathcal{A}$ is $P + 1$. Moreover, if the fraction of valid entries in M is ε, then*

$$\Pr\left[(\mathsf{S}, \mathsf{Z}, \mathsf{S}', \mathsf{Z}') \leftarrow \mathcal{A} \wedge \left(\mathsf{S} \neq \mathsf{S}' \vee \left(\mathsf{S} = \mathsf{S}' \wedge (\mathsf{S}, \mathsf{Z}) \in \mathcal{X}\right)\right)\right] \geqslant \varepsilon - P\kappa.$$

Proof. We first focus on the expected runtime. Let T be the number of samples generated by $\mathcal{A}$ excluding the first one. Then, $\mathbb{E}[\mathsf{T}|\mathsf{V} = 0] \cdot \Pr[\mathsf{V} = 0] = 0$. On the other hand, we observe that for fixed $\mathbb{h} \in [h], \mathbb{i} \in [U_{\mathbb{h}}]$ and $a \in [N_{\mathbb{h}}]$:

$$\mathbb{E}[\mathsf{T}|\mathsf{V} = 1 \wedge \mathsf{H} = t \wedge \mathsf{I} = \mathbb{i} \wedge A_{\mathbb{h},\mathbb{i}} = a] = \frac{N_{\mathbb{h}}}{a} \quad \text{and} \quad \Pr[\mathsf{V} = 1 \wedge \mathsf{H} = \mathbb{h} \wedge \mathsf{I} = \mathbb{i} | A_{\mathbb{h},\mathbb{i}} = a] = \frac{a}{N_{\mathbb{h}}}.$$

Hence,

$$\begin{aligned}
\mathbb{E}[\mathsf{T}|\mathsf{V} = 1] \cdot \Pr[\mathsf{V} = 1] &= \sum_{\mathbb{h}=1}^{h} \sum_{\mathbb{i}=1}^{U_{\mathbb{h}}} \sum_{a=1}^{N_{\mathbb{h}}} \mathbb{E}[\mathsf{T}|\mathsf{V} = 1 \wedge \mathsf{H} = \mathbb{h} \wedge \mathsf{I} = \mathbb{i} \wedge A_{\mathbb{h},\mathbb{i}} = a] \cdot \Pr[\mathsf{V} = 1 \wedge \mathsf{H} = \mathbb{h} \wedge \mathsf{I} = \mathbb{i} \wedge A_{\mathbb{h},\mathbb{i}} = a] \\
&= \sum_{\mathbb{h}=1}^{h} \sum_{\mathbb{i}=1}^{U_{\mathbb{h}}} \sum_{a=1}^{N_{\mathbb{h}}} \frac{N_{\mathbb{h}}}{a} \cdot \frac{a}{N_{\mathbb{h}}} \cdot \Pr[A_{\mathbb{h},\mathbb{i}} = a] \\
&= \sum_{\mathbb{h}=1}^{h} \sum_{\mathbb{i}=1}^{U_{\mathbb{h}}} \Pr[A_{\mathbb{h},\mathbb{i}} > 0] = P
\end{aligned}$$

which concludes the first part of the proof.

Next, we prove the lower bound on the success probability. We first introduce the following sets:

$$\begin{aligned}
&\mathsf{Good}(\mathbb{h}, \mathbb{i}, a) := \{(r, \vec{j}_1, \ldots, \vec{j}_h) : M(r, \vec{j}_1, \ldots, \vec{j}_h) = (1, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z}) \wedge a_{\mathbb{h},\mathbb{i}}(r, \vec{j}_1, \ldots, \vec{j}_h) = a\} \\
&\mathsf{InvWit}(\mathbb{h}, \mathbb{i}, a) := \{(r, \vec{j}_1, \ldots, \vec{j}_h) : M(r, \vec{j}_1, \ldots, \vec{j}_h) = (1, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z}) \wedge (\mathbb{s}, \mathbb{z}) \notin \mathcal{X}\} \\
&\mathsf{GIW}(\mathbb{h}, \mathbb{i}, a) := \mathsf{Good}(\mathbb{h}, \mathbb{i}, a) \cap \mathsf{InvWit}(\mathbb{h}, \mathbb{i}, a).
\end{aligned}$$

Then, by definition

$$\frac{a}{N_{\mathbb{h}}} \cdot \Pr[A_{\mathbb{h},\mathbb{i}} = a] = \Pr[\mathsf{V} = 1 \wedge \mathsf{H} = \mathbb{h} \wedge \mathsf{I} = \mathbb{i} \wedge A_{\mathbb{h},\mathbb{i}} = a] = \frac{|\mathsf{Good}(\mathbb{h}, \mathbb{i}, a)|}{R \cdot N_1^{U_1} \cdot \ldots \cdot N_h^{U_h}}. \tag{82}$$

Now, consider the probability that $\mathcal{A}$ does not output $\perp$. Note that the probability that $\mathcal{A}$ does not abort in the second step is exactly ε. Assuming $\mathcal{A}$ goes through the second step, one observes that the probability that it terminates, i.e. outputs $(\mathsf{S}, \mathsf{Z}, \mathsf{S}', \mathsf{Z}')$, is equal to one. Hence,

$$\begin{aligned}
\varepsilon = \Pr[(\mathsf{S}, \mathsf{Z}, \mathsf{S}', \mathsf{Z}') \leftarrow \mathcal{A}] = \Pr[(\mathsf{S}, \mathsf{Z}, \mathsf{S}', \mathsf{Z}') &\leftarrow \mathcal{A} \wedge \mathsf{S} \neq \mathsf{S}'] \\
&+ \Pr[(\mathsf{S}, \mathsf{Z}, \mathsf{S}', \mathsf{Z}') \leftarrow \mathcal{A} \wedge \mathsf{S} = \mathsf{S}' \wedge (\mathsf{S}, \mathsf{Z}) \in \mathcal{X}] \\
&+ \Pr[(\mathsf{S}, \mathsf{Z}, \mathsf{S}', \mathsf{Z}') \leftarrow \mathcal{A} \wedge \mathsf{S} = \mathsf{S}' \wedge (\mathsf{S}, \mathsf{Z}) \notin \mathcal{X}].
\end{aligned}$$

To prove the statement, we just need to show that $\Pr[(\mathsf{S},\mathsf{Z},\mathsf{S}',\mathsf{Z}') \leftarrow \mathcal{A} \wedge \mathsf{S} = \mathsf{S}' \wedge (\mathsf{S},\mathsf{Z}) \notin \mathcal{X}] \leqslant P\kappa$. To this end, consider any $(r, \vec{j}_1, \ldots, \vec{j}_h) \in \mathsf{GIW}(\mathbb{h}, \mathbb{i}, a)$ for $a > 0$ and define $M(r, \vec{j}_1, \ldots, \vec{j}_h) = (1, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z})$. Then

$$\begin{aligned}
\Pr[(\mathsf{S},\mathsf{Z},\mathsf{S}',\mathsf{Z}') \leftarrow \mathcal{A} \wedge (\mathsf{S},\mathsf{Z}) \notin \mathcal{X} \mid (\mathsf{R}, \vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) = (r, \vec{j}_1, \ldots, \vec{j}_h)]& \\
&= \Pr_{\mathsf{J}^* \leftarrow [N_{\mathbb{h}}]}\left[\mathsf{S}' = \mathbb{s} \middle| M(r, \vec{j}_1, \ldots, \vec{j}_{\mathbb{h}-1}, \vec{j}^*, \vec{j}_{\mathbb{h}+1}, \ldots, \vec{j}_h) = (1, \mathbb{h}, \mathbb{i}, \mathsf{S}', \mathsf{Z}')\right] \\
&= \frac{\Pr_{\mathsf{J}^* \leftarrow [N_{\mathbb{h}}]}\left[M(r, \vec{j}_1, \ldots, \vec{j}_{\mathbb{h}-1}, \vec{j}^*, \vec{j}_{\mathbb{h}+1}, \ldots, \vec{j}_h) = (1, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathsf{Z}')\right]}{\Pr_{\mathsf{J}^* \leftarrow [N_{\mathbb{h}}]}\left[M(r, \vec{j}_1, \ldots, \vec{j}_{\mathbb{h}-1}, \vec{j}^*, \vec{j}_{\mathbb{h}+1}, \ldots, \vec{j}_h) = (1, \mathbb{h}, \mathbb{i}, \mathsf{S}', \mathsf{Z}')\right]} \\
&\leqslant \kappa \cdot \frac{N_{\mathbb{h}}}{a}
\end{aligned}$$

where $\vec{j}^* := (\vec{j}_{\mathbb{h},1}, \ldots, \vec{j}_{\mathbb{h},\mathbb{i}-1}, \mathsf{J}^*, \vec{j}_{\mathbb{h},\mathbb{i}+1}, \ldots, \vec{j}_{\mathbb{h},U_{\mathbb{h}}})$. For the inequality, we used the fact that $\mathcal{X}$ is κ-probabilistically-testable w.r.t. M and that the denominator is equal to $a/N_{\mathbb{h}}$ since $(r, \vec{j}_1, \ldots, \vec{j}_h) \in \mathsf{Good}(\mathbb{h}, \mathbb{i}, a)$. Using this inequality, we obtain

$$\begin{aligned}
\Pr[(\mathsf{S},\mathsf{Z},\mathsf{S}',\mathsf{Z}') \leftarrow \mathcal{A} \wedge \mathsf{S} = \mathsf{S}' \wedge (\mathsf{S},\mathsf{Z}) \notin \mathcal{X}] &\leqslant \sum_{\mathbb{h}=1}^{h} \sum_{\mathbb{i}=1}^{U_{\mathbb{h}}} \sum_{a=1}^{N_{\mathbb{h}}} \sum_{(r, \vec{j}_1, \ldots, \vec{j}_h) \in \mathsf{GIW}(\mathbb{h}, \mathbb{i}, a)} \kappa \cdot \frac{N_{\mathbb{h}}}{a} \cdot \frac{1}{R \cdot N_1^{U_1} \cdot \ldots \cdot N_h^{U_h}} \\
&\leqslant \sum_{\mathbb{h}=1}^{h} \sum_{\mathbb{i}=1}^{U_{\mathbb{h}}} \sum_{a=1}^{N_{\mathbb{h}}} \sum_{(r, \vec{j}_1, \ldots, \vec{j}_h) \in \mathsf{Good}(\mathbb{h}, \mathbb{i}, a)} \kappa \cdot \frac{N_{\mathbb{h}}}{a} \cdot \frac{1}{R \cdot N_1^{U_1} \cdot \ldots \cdot N_h^{U_h}} \\
&\leqslant \sum_{\mathbb{h}=1}^{h} \sum_{\mathbb{i}=1}^{U_{\mathbb{h}}} \sum_{a=1}^{N_{\mathbb{h}}} \kappa \cdot \frac{N_{\mathbb{h}}}{a} \cdot \frac{|\mathsf{Good}(\mathbb{h}, \mathbb{i}, a)|}{R \cdot N_1^{U_1} \cdot \ldots \cdot N_h^{U_h}} \\
&\leqslant \sum_{\mathbb{h}=1}^{h} \sum_{\mathbb{i}=1}^{U_{\mathbb{h}}} \sum_{a=1}^{N_{\mathbb{h}}} \kappa \cdot \Pr[A_{\mathbb{h},\mathbb{i}} = a] \leqslant P\kappa
\end{aligned}$$

where for the second-to-last inequality we used (82). This concludes the proof. □

The next lemma describes a way to bound the value P. In our applications, P will be bounded by the number of random oracle queries the cheating prover makes. It is a simple generalisation of [AFK21, Lemma 3], but we provide the proof for completeness.

Lemma B.4 ([AFK21]). *Consider the game in Fig. 19 and define functions $\mathfrak{v}, \mathfrak{h}, \mathfrak{i}, \mathfrak{s}, \mathfrak{z}$ such that*

$$M(r, \vec{j}_1, \ldots, \vec{j}_h) = (\mathfrak{v}(r, \vec{j}_1, \ldots, \vec{j}_h), \mathfrak{h}(r, \vec{j}_1, \ldots, \vec{j}_h), \mathfrak{i}(r, \vec{j}_1, \ldots, \vec{j}_h), \mathfrak{s}(r, \vec{j}_1, \ldots, \vec{j}_h), \mathfrak{z}(r, \vec{j}_1, \ldots, \vec{j}_h)).$$

As before, for $\mathbb{h} \in [h]$ and $\mathbb{i} \in [U_{\mathbb{h}}]$, denote the random variable $A_{\mathbb{h},\mathbb{i}} := a_{\mathbb{h},\mathbb{i}}(\mathsf{R}, \vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h)$. Additionally, let us assume that for all $(r, \vec{j}_1, \ldots, \vec{j}_h)$, there exists a subset

$$\mathcal{S}(r, \vec{j}_1, \ldots, \vec{j}_h) \subseteq \{(\mathbb{h}, \mathbb{i}) : \mathbb{h} \in [h] \wedge \mathbb{i} \in [U_{\mathbb{h}}]\}$$

of cardinality at most Q, such that $\mathfrak{h}(r, \vec{j}_1, \ldots, \vec{j}_h) = \mathfrak{h}(r, \vec{j}'_1, \ldots, \vec{j}'_h)$ and $\mathfrak{i}(r, \vec{j}_1, \ldots, \vec{j}_h) = \mathfrak{i}(r, \vec{j}'_1, \ldots, \vec{j}'_h)$ for all $(\vec{j}'_1, \ldots, \vec{j}'_h)$ which satisfy: $\forall (t, i) \in \mathcal{S}(r, \vec{j}_1, \ldots, \vec{j}_h), j_{t,i} = j'_{t,i}$. Then

$$P := \sum_{\mathbb{h}=1}^{h} \sum_{\mathbb{i}=1}^{U_{\mathbb{h}}} \Pr[A_{\mathbb{h},\mathbb{i}} > 0] \leqslant Q + 1.$$

Proof. By the law of total probability, we have

$$\begin{aligned}
P &= \sum_{\mathbb{h}=1}^{h} \sum_{\mathbb{i}=1}^{U_{\mathbb{h}}} \Pr[A_{\mathbb{h},\mathbb{i}} > 0] \\
&= \sum_{r,\vec{j}} \Pr[\mathsf{R} = r \wedge (\vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) = \vec{j}] \cdot \sum_{\mathbb{h}=1}^{h} \sum_{\mathbb{i}=1}^{U_{\mathbb{h}}} \Pr[A_{\mathbb{h},\mathbb{i}} > 0 | \mathsf{R} = r \wedge (\vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) = \vec{j}] \\
&\leqslant \sum_{r,\vec{j}} \Pr[\mathsf{R} = r \wedge (\vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) = \vec{j}] \cdot \left(\sum_{(\mathbb{h},\mathbb{i}) \in \mathcal{S}(r,\vec{j})} 1 + \sum_{(\mathbb{h},\mathbb{i}) \notin \mathcal{S}(r,\vec{j})} \Pr[A_{\mathbb{h},\mathbb{i}} > 0 | \mathsf{R} = r \wedge (\vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) = \vec{j}] \right) \\
&\leqslant Q + \sum_{r,\vec{j}} \Pr[\mathsf{R} = r \wedge (\vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) = \vec{j}] \cdot \sum_{(\mathbb{h},\mathbb{i}) \notin \mathcal{S}(r,\vec{j})} \Pr[A_{\mathbb{h},\mathbb{i}} > 0 | \mathsf{R} = r \wedge (\vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) = \vec{j}].
\end{aligned}$$

Now, by definition of the set $\mathcal{S}(r, \vec{j}_1, \ldots, \vec{j}_h)$, if $(\mathbb{h}, \mathbb{i}) \notin \mathcal{S}(r, \vec{j}_1, \ldots, \vec{j}_h)$ then for any $j^*_{\mathbb{h},\mathbb{i}} \in [N_{\mathbb{h}}]$:

$$\mathfrak{h}(r, \vec{j}_1, \vec{j}_{\mathbb{h}-1}, \vec{j}^*, \vec{j}_{\mathbb{h}+1}, \ldots, \vec{j}_h) = \mathfrak{h}(r, \vec{j}_1, \ldots, \vec{j}_h) \quad \text{and} \quad \mathfrak{i}(r, \vec{j}_1, \vec{j}_{\mathbb{h}-1}, \vec{j}^*, \vec{j}_{\mathbb{h}+1}, \ldots, \vec{j}_h) = \mathfrak{i}(r, \vec{j}_1, \ldots, \vec{j}_h)$$

where $\vec{j}^* := (j_{\mathbb{h},1}, \ldots, j_{\mathbb{h},\mathbb{i}-1}, j^*_{\mathbb{h},\mathbb{i}}, j_{\mathbb{h},\mathbb{i}+1}, \ldots, j_{\mathbb{h},U_{\mathbb{h}}})$.

Hence, for all $(\mathbb{h}, \mathbb{i}) \notin \left\{ \left(\mathfrak{h}(r, \vec{j}_1, \ldots, \vec{j}_h), \mathfrak{i}(r, \vec{j}_1, \ldots, \vec{j}_h) \right) \right\} \cup \mathcal{S}(r, \vec{j}_1, \ldots, \vec{j}_h)$ we have

$$\Pr[A_{\mathbb{h},\mathbb{i}} > 0 | \mathsf{R} = r \wedge (\vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) = \vec{j}] = 0$$

which implies that

$$\begin{aligned}
\sum_{(\mathbb{h},\mathbb{i}) \notin \mathcal{S}(r,\vec{j})} \Pr\left[A_{\mathbb{h},\mathbb{i}} > 0 | \ \mathsf{R} = r \wedge (\vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) = \vec{j} \right] &\leqslant \Pr\left[A_{\mathfrak{h}(r,\vec{j}_1,\ldots,\vec{j}_h),\mathfrak{i}(r,\vec{j}_1,\ldots,\vec{j}_h)} > 0 | \mathsf{R} = r \wedge (\vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) = \vec{j} \right] \\
&\leqslant 1.
\end{aligned}$$

Therefore $P \leqslant Q + \sum_{r,\vec{j}} \Pr[\mathsf{R} = r \wedge (\vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) = \vec{j}] \leqslant Q + 1$. □

Weighted Version of the Abstract Sampling Game. Similarly as in [AFK21], in order to obtain tight bounds on the runtime of the extractor for the multi-round protocol, we need to consider a refined analysis of the cost of playing the abstract sampling game. Informally, for multi-round protocols, every entry will correspond to running the extractor of the underlying sub-protocol. As noted by Attema et al. [AFK21], some invocations of the extractor are expensive, while others are cheap. To this end, we introduce a cost function Γ and a constant γ which correspond to the expensive and cheap computation respectively.

The following lemma gives an upper-bound on the total cost of playing the abstract sampling game in terms of the cost functions mentioned above. We simply adapt the proof from [AFK21, Lemma 5] to our setting.

Lemma B.5. *Consider the abstract sampling game in Fig. 19. Let $\Gamma : [R] \times [N_1]^{U_1} \times \cdots \times [N_h]^{U_h} \to \mathbb{R}_{\geqslant 0}$ be a cost function and $\gamma \in \mathbb{R}_{\geqslant 0}$ be a constant cost. As before, for $\mathbb{h} \in [h]$ and $\mathbb{i} \in [U_{\mathbb{h}}]$, denote the random variable $A_{\mathbb{h},\mathbb{i}} := a_{\mathbb{h},\mathbb{i}}(\mathsf{R}, \vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h)$. We define the cost of sampling an entry $M(r, \vec{j}_1, \ldots, \vec{j}_h) = (\mathbb{v}, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z})$ with indices $(\mathbb{h}, \mathbb{i}) = (\mathsf{H}, \mathsf{I})$ to be $\Gamma(r, \vec{j}_1, \ldots, \vec{j}_h)$ and the cost of sampling an entry $M(r, \vec{j}_1, \ldots, \vec{j}_h) = (\mathbb{v}, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z})$ with indices $(\mathbb{h}, \mathbb{i}) \neq (\mathsf{H}, \mathsf{I})$ to be γ. Let Δ be the total cost of playing this game. Then*

$$\mathbb{E}[\Delta] \leqslant 2 \cdot \mathbb{E}[\Gamma(\mathsf{R}, \vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h)] + T \cdot \gamma$$

where $T := \sum_{\mathbb{h}=1}^{h} \sum_{\mathbb{i}=1}^{U_{\mathbb{h}}} \Pr[(\mathsf{H}, \mathsf{I}) \neq (\mathbb{h}, \mathbb{i}) \wedge A_{\mathbb{h},\mathbb{i}} > 0]$.

Proof. We first split the total cost Δ into the three cost measures $\Delta_1, \Delta_2, \Delta_3$ defined as follows.

- Δ_1 denotes the total cost of the elements $M(r, \vec{j}_1, \ldots, \vec{j}_h) = (1, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z})$ with $(\mathbb{h}, \mathbb{i}) = (\mathsf{H}, \mathsf{I})$ sampled in the game, while X denotes the number of entries of the form $(1, \mathbb{h}, \mathbb{i}, \cdot, \cdot)$, such that $(\mathbb{h}, \mathbb{i}) = (\mathsf{H}, \mathsf{I})$, sampled (including the first one if $\mathsf{V} = 1$).
- Δ_2 is the total cost of the elements $M(r, \vec{j}_1, \ldots, \vec{j}_h) = (0, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z})$ with $(\mathbb{h}, \mathbb{i}) = (\mathsf{H}, \mathsf{I})$ sampled, while Y denotes the number of entries of the form $(0, \mathbb{h}, \mathbb{i}, \cdot, \cdot)$, such that $(\mathbb{h}, \mathbb{i}) = (\mathsf{H}, \mathsf{I})$, sampled (including the first one if $\mathsf{V} = 0$).
- Δ_3 denotes the total cost of the elements $M(r, \vec{j}_1, \ldots, \vec{j}_h) = (\mathbb{v}, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z})$ with $(\mathbb{h}, \mathbb{i}) \neq (\mathsf{H}, \mathsf{I})$ sampled in the game, while Z denotes the number of entries of the form $(\mathbb{v}, \mathbb{h}, \mathbb{i}, \cdot, \cdot)$, such that $(\mathbb{h}, \mathbb{i}) \neq (\mathsf{H}, \mathsf{I})$, sampled.

By definition, we have $\Delta = \Delta_1 + \Delta_2 + \Delta_3$. Also, since the cost γ is constant, we have $\mathbb{E}[\Delta_3] = \gamma \cdot \mathbb{E}[Z]$. The first goal is to relate $\mathbb{E}[\Delta_1]$ and $\mathbb{E}[\Delta_2]$ to $\mathbb{E}[X]$ and $\mathbb{E}[Z]$ respectively. Now, for any $\mathbb{h} \in [h]$ and $\mathbb{i} \in [U_{\mathbb{h}}]$, denote

$$\vec{\mathsf{J}}^*_{\mathbb{h},\mathbb{i}} := (\vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_{\mathbb{h}-1}, \vec{\mathsf{J}}', \vec{\mathsf{J}}_{\mathbb{h}+1}, \ldots, \vec{\mathsf{J}}_h) \text{ where } \vec{\mathsf{J}}' := (\mathsf{J}_{\mathbb{h},1}, \ldots, \mathsf{J}_{\mathbb{h},\mathbb{i}-1}, \mathsf{J}_{\mathbb{h},\mathbb{i}+1}, \ldots, \mathsf{J}_{\mathbb{h},U_{\mathbb{h}}})$$
$$\vec{\mathsf{J}}^*_{\mathbb{h},\mathbb{i}}(\mathsf{X}) := (\vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_{\mathbb{h}-1}, \vec{\mathsf{J}}_{\mathsf{X}}, \vec{\mathsf{J}}_{\mathbb{h}+1}, \ldots, \vec{\mathsf{J}}_h) \text{ where } \vec{\mathsf{J}}_{\mathsf{X}} := (\mathsf{J}_{\mathbb{h},1}, \ldots, \mathsf{J}_{\mathbb{h},\mathbb{i}-1}, \mathsf{X}, \mathsf{J}_{\mathbb{h},\mathbb{i}+1}, \ldots, \mathsf{J}_{\mathbb{h},U_{\mathbb{h}}}).$$

Moreover, for any $(r^*, \vec{j}^*) \in [R] \times [N_1]^{U_1} \times \cdots \times [N_{\mathbb{h}-1}]^{U_{\mathbb{h}-1}} \times [N_{\mathbb{h}}]^{U_{\mathbb{h}}-1} \times [N_{\mathbb{h}+1}]^{U_{\mathbb{h}+1}} \times \cdots \times [N_h]^{U_h}$, let $\Lambda^*(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*)$ denote the event

$$\Lambda^*(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*) := \Big[(\mathsf{H}, \mathsf{I}) = (\mathbb{h}, \mathbb{i}) \wedge (\mathsf{R}, \vec{\mathsf{J}}^*_{\mathbb{h},\mathbb{i}}) = (r^*, \vec{j}^*)\Big].$$

We first bound $\mathbb{E}[\Delta_1 | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*)]$. Note that for any $\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*$ such that $\Pr[\Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*)] > 0$, we have

$$\mathbb{E}[\Delta_1 | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*)] = \sum_{\ell=0}^{\infty} \Pr[X = \ell | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*)] \cdot \mathbb{E}[\Delta_1 | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*) \wedge X = \ell].$$

First, fix $\ell > 0$. Suppose that $\Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*) \wedge X = \ell$ hold and let $\mathsf{K}_1, \ldots, \mathsf{K}_\ell$ be the random variables for which $M(\mathsf{R}, \vec{\mathsf{J}}_{\mathbb{h},\mathbb{i}}(\mathsf{K}_i)) = (1, \mathbb{h}, \mathbb{i}, \cdot, \cdot)$. Note that they are pairwise independent and their distribution is identical, as seen in Step 3 of Fig. 19. Then, by definition of Δ_1 we get

$$\begin{aligned}\mathbb{E}[\Delta_1 | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*) \wedge X = \ell] &= \sum_{i=1}^{\ell} \mathbb{E}[\Gamma(\mathsf{R}, \vec{\mathsf{J}}_{\mathbb{h},\mathbb{i}}(\mathsf{K}_i)) | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*) \wedge \mathsf{V} = 1] \\ &= \ell \cdot \mathbb{E}[\Gamma(\mathsf{R}, \vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*) \wedge \mathsf{V} = 1].\end{aligned}$$

One can easily check that this equality also holds when $\ell = 0$. Therefore, we get

$$\begin{aligned}\mathbb{E}[\Delta_1 | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*)] &= \mathbb{E}[\Gamma(\mathsf{R}, \vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*) \wedge \mathsf{V} = 1] \cdot \sum_{\ell=0}^{\infty} \Pr[X = \ell | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*)] \cdot \ell \\ &= \mathbb{E}[\Gamma(\mathsf{R}, \vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*) \wedge \mathsf{V} = 1] \cdot \mathbb{E}[X | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*)].\end{aligned}$$

Similarly, one proves that

$$\mathbb{E}[\Delta_2 | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*)] = \mathbb{E}[\Gamma(\mathsf{R}, \vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*) \wedge \mathsf{V} = 0] \cdot \mathbb{E}[Y | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*)].$$

Next, we bound the expected values of X and Y conditioned on $\Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*)$. First, using the observation that $\mathsf{V} = 0$ implies $X = 0$ and $\mathsf{V} = 1$ implies $X = 2$, we have

$$\mathbb{E}[X | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*)] = 2 \cdot \Pr[\mathsf{V} = 1 | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*)]$$

and thus

$$\mathbb{E}[\Delta_1 | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*)] = 2 \cdot \Pr[\mathsf{V} = 1 | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*)] \cdot \mathbb{E}[\Gamma(\mathsf{R}, \vec{\mathsf{J}}_1, \ldots, \vec{\mathsf{J}}_h) | \Lambda(\mathbb{h}, \mathbb{i}, r^*, \vec{j}^*) \wedge \mathsf{V} = 1].$$

Next, we focus on $\mathbb{E}[Y|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)]$. For fixed $\mathbb{h},\mathbb{i},r^*,\vec{j}^*$, denote

$$a := a_{\mathbb{h},\mathbb{i}}(r^*,\vec{j}^*) = |\{j : M(r,\vec{j}_1^*,\ldots,\vec{j}_{\mathbb{h}-1}^*,\vec{j},\vec{j}_{\mathbb{h}+1}^*,\ldots,\vec{j}_h^*) = (1,\mathbb{h},\mathbb{i},\cdot,\cdot)\}|$$

where $\vec{j} := (j^*_{\mathbb{h},1},\ldots,j^*_{\mathbb{h},\mathbb{i}-1},j,j^*_{\mathbb{h},\mathbb{i}+1},\ldots,j^*_{\mathbb{h},U_{\mathbb{h}}})$. Similarly, we define

$$b := b_{\mathbb{h},\mathbb{i}}(r^*,\vec{j}^*) = |\{j : M(r,\vec{j}_1^*,\ldots,\vec{j}_{\mathbb{h}-1}^*,\vec{j},\vec{j}_{\mathbb{h}+1}^*,\ldots,\vec{j}_h^*) = (0,\mathbb{h},\mathbb{i},\cdot,\cdot)\}|.$$

We observe that

$$\Pr[\mathsf{V} = 1|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] = \frac{a}{a+b} \quad \text{and} \quad \Pr[\mathsf{V} = 0|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] = \frac{a}{a+b}$$

assuming that $\Pr[\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] > 0$. Hence, if we condition the event $\mathsf{V} = 1 \wedge \varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)$, we implicitly assume that $a > 0$. Now, to bound $\mathbb{E}[Y|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)]$, we observe that conditioned on $\mathsf{V} = 1 \wedge \varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)$, random variable Y follows the simplified negative multinomial distribution with parameters $(a, b, N_{\mathbb{h}} - a - b)$ described in Section B.1. Hence, by Lemma B.1

$$\mathbb{E}[Y|\mathsf{V} = 1 \wedge \varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] = \frac{b}{a}.$$

Thus

$$\begin{aligned}
\mathbb{E}[Y|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] &= \Pr[\mathsf{V} = 0|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] + \Pr[\mathsf{V} = 1|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] \cdot \frac{b}{a} \\
&= \frac{b}{a+b} + \frac{a}{a+b} \cdot \frac{b}{a} \\
&= 2 \cdot \frac{b}{a+b} \\
&= 2 \cdot \Pr[\mathsf{V} = 0|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)]
\end{aligned}$$

where in the first equality we used the fact that $\mathbb{E}[Y|\mathsf{V} = 0 \wedge \varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] = 1$. Hence,

$$\mathbb{E}[\varDelta_2|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] = 2 \cdot \Pr[\mathsf{V} = 0|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] \cdot \mathbb{E}[\varGamma(\mathsf{R},\vec{\mathsf{J}}_1,\ldots,\vec{\mathsf{J}}_h)|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*) \wedge \mathsf{V} = 0],$$

and by combining the previous results

$$\mathbb{E}[\varDelta_1 + \varDelta_2|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] = 2 \cdot \mathbb{E}[\varGamma(\mathsf{R},\vec{\mathsf{J}}_1,\ldots,\vec{\mathsf{J}}_h)|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*) \wedge \mathsf{V} = 0].$$

Since the equality holds for all $\mathbb{h},\mathbb{i},r^*,\vec{j}^*$ such that $\Pr[\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] > 0$, we deduce that

$$\mathbb{E}[\varDelta_1 + \varDelta_2] = 2 \cdot \mathbb{E}[\varGamma(\mathsf{R},\vec{\mathsf{J}}_1,\ldots,\vec{\mathsf{J}}_h)].$$

Finally, we still need to prove that $\mathbb{E}[Z] \leqslant T$ which implies that $\mathbb{E}[\varDelta_3] = \gamma\mathbb{E}[Z] \leqslant T \cdot \gamma$. We use a similar approach as for computing the expected value of Y. Namely, recall the values a, b defined above for fixed $\mathbb{h},\mathbb{i},r^*,\vec{j}^*$ such that $\Pr[\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] > 0$. Then, conditioned on $\mathsf{V} = 1 \wedge \varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)$, Z follows the simplified negative multinomial distribution with parameters $(a, N - a - b, b)$. Hence, by Lemma B.1 we have

$$\mathbb{E}[Z|\mathsf{V} = 1 \wedge \varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] = \frac{N - a - b}{a}.$$

Moreover, $\mathbb{E}[Z|\mathsf{V} = 0 \wedge \varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] = 0$ by definition. In the following, we will use the observation that

$$\Pr[(\mathsf{H},\mathsf{I}) = (\mathbb{h},\mathbb{i})|(\mathsf{R},\mathsf{J}^*_{\mathbb{h},\mathbb{i}}) = (r,\vec{j}^*)] = \frac{a+b}{N_{\mathsf{h}}}.$$

Thus

$$\begin{aligned}
\mathbb{E}[Z|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] &= \Pr[\mathsf{V}=1|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)]\cdot\mathbb{E}[Z|\mathsf{V}=1\wedge\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] \\
&= \frac{a}{a+b}\cdot\frac{N-a-b}{a} \\
&= \frac{1}{\Pr[(\mathsf{H},\mathsf{I})=(\mathbb{h},\mathbb{i})|(\mathsf{R},\mathsf{J}^*_{\mathbb{h},\mathbb{i}})=(r,\vec{j}^*)]}-1 \\
&= \frac{\Pr[(\mathsf{R},\mathsf{J}^*_{\mathbb{h},\mathbb{i}})=(r,\vec{j}^*)]-\Pr[(\mathsf{H},\mathsf{I})=(\mathbb{h},\mathbb{i})\wedge(\mathsf{R},\mathsf{J}^*_{\mathbb{h},\mathbb{i}})=(r,\vec{j}^*)]}{\Pr[\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)]} \\
&= \frac{\Pr[(\mathsf{H},\mathsf{I})\neq(\mathbb{h},\mathbb{i})\wedge(\mathsf{R},\mathsf{J}^*_{\mathbb{h},\mathbb{i}})=(r,\vec{j}^*)]}{\Pr[\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)]}.
\end{aligned}$$

Recall that we implicitly assumed that $a>0$. For $\mathbb{h},\mathbb{i},r^*,\vec{j}^*$ such that $a=0$, $\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)$ implies that $\mathsf{V}=0$. Consequently, $\mathbb{E}[Z|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)]=0$. To sum up, we have

$$\begin{aligned}
\mathbb{E}[Z] &= \sum_{\mathbb{h}=1}^{h}\sum_{\mathbb{i}=1}^{U_{\mathbb{h}}}\sum_{\substack{r^*,\vec{j}^*\text{ s.t.}\\ a_{\mathbb{h},\mathbb{i}}(r^*,\vec{j}^*)>0}}\Pr[\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)]\cdot\mathbb{E}[Z|\varLambda(\mathbb{h},\mathbb{i},r^*,\vec{j}^*)] \\
&= \sum_{\mathbb{h}=1}^{h}\sum_{\mathbb{i}=1}^{U_{\mathbb{h}}}\sum_{\substack{r^*,\vec{j}^*\text{ s.t.}\\ a_{\mathbb{h},\mathbb{i}}(r^*,\vec{j}^*)>0}}\Pr[(\mathsf{H},\mathsf{I})\neq(\mathbb{h},\mathbb{i})\wedge(\mathsf{R},\mathsf{J}^*_{\mathbb{h},\mathbb{i}})=(r,\vec{j}^*)] \\
&= \sum_{\mathbb{h}=1}^{h}\sum_{\mathbb{i}=1}^{U_{\mathbb{h}}}\Pr[(\mathsf{H},\mathsf{I})\neq(\mathbb{h},\mathbb{i})\wedge A_{\mathbb{h},\mathbb{i}}>0] \\
&= T.
\end{aligned}$$

which completes the proof. □

The next lemma shows how to upper-bound the parameter T. Since the result is almost identical to Lemma B.4 and [AFK21, Lemma 6], we omit the formal proof.

Lemma B.6 ([AFK21]). *Consider the game in Fig. 19 and define functions* $\mathfrak{v},\mathfrak{h},\mathfrak{i},\mathfrak{s},\mathfrak{z}$ *such that*

$$M(r,\vec{j}_1,\dots,\vec{j}_h)=(\mathfrak{v}(r,\vec{j}_1,\dots,\vec{j}_h),\mathfrak{h}(r,\vec{j}_1,\dots,\vec{j}_h),\mathfrak{i}(r,\vec{j}_1,\dots,\vec{j}_h),\mathfrak{s}(r,\vec{j}_1,\dots,\vec{j}_h),\mathfrak{z}(r,\vec{j}_1,\dots,\vec{j}_h)).$$

As before, for $\mathbb{h}\in[h]$ *and* $\mathbb{i}\in[U_{\mathbb{h}}]$*, denote the random variable* $A_{\mathbb{h},\mathbb{i}}:=a_{\mathbb{h},\mathbb{i}}(\mathsf{R},\vec{\mathsf{J}}_1,\dots,\vec{\mathsf{J}}_h)$*. Additionally, let us assume that for all* $(r,\vec{j}_1,\dots,\vec{j}_h)$*, there exists a subset*

$$\mathcal{S}(r,\vec{j}_1,\dots,\vec{j}_h)\subseteq\{(\mathbb{h},\mathbb{i}):\mathbb{h}\in[h]\wedge\mathbb{i}\in[U_{\mathbb{h}}]\}$$

of cardinality at most Q*, such that* $\mathfrak{h}(r,\vec{j}_1,\dots,\vec{j}_h)=\mathfrak{h}(r,\vec{j}'_1,\dots,\vec{j}'_h)$ *and* $\mathfrak{i}(r,\vec{j}_1,\dots,\vec{j}_h)=\mathfrak{i}(r,\vec{j}'_1,\dots,\vec{j}'_h)$ *for all* $(\vec{j}'_1,\dots,\vec{j}'_h)$ *which satisfy:* $\forall(t,i)\in\mathcal{S}(r,\vec{j}_1,\dots,\vec{j}_h),j_{t,i}=j'_{t,i}$*. Then*

$$T:=\sum_{\mathbb{h}=1}^{h}\sum_{\mathbb{i}=1}^{U_{\mathbb{h}}}\Pr[(\mathsf{H},\mathsf{I})\neq(\mathbb{h},\mathbb{i})\wedge A_{\mathbb{h},\mathbb{i}}>0]\leqslant Q.$$

B.3 Fiat-Shamir Transformation

We are ready to analyse the non-interactive version of our protocols using the Fiat-Shamir transformation. For concreteness and simplicity, we focus on the particular protocol for proving knowledge of a polynomial vector $\mathbf{s}_1\in\mathcal{R}_q^{m_1}$ which satisfies $\mathbf{P}\mathbf{s}_1=\mathbf{u}$ over $\mathcal{R}_q$ and $\|\mathbf{s}_1\|^2=B$ for public $\mathbf{P}\in\mathcal{R}_q^{n_1\times m_1}$ and $\mathbf{u}\in\mathcal{R}_q^{n_1}$. For convenience, we will denote $(\mathbf{p}_i)_{i\in[n_1]}$ to be the rows of $\mathbf{P}$. Hence $\mathbf{P}\mathbf{s}_1 = \mathbf{u}$ is equivalent to $\forall i \in [n_1], \mathbf{p}_i^T\mathbf{s}_1 = u_i$. We formally define the statement x and the public parameters pp (i.e. the common random string) as follows:

$$x := (\mathbf{P}, \mathbf{u}, B) \quad \text{and} \quad \mathsf{pp} := \left(\mathbf{A}_1 \in \mathcal{R}_q^{n\times m_1}, \mathbf{A}_2 \in \mathcal{R}_q^{n\times m_2}, \mathbf{B}^{(e)} \in \mathcal{R}_q^{256/d\times m_2}, \mathbf{B}_g \in \mathcal{R}_q^{\lambda\times m_2}, \mathbf{b} \in \mathcal{R}_q^{m_2}\right).$$

Non-Interactive Protocol. We construct the non-interactive proof for the statement above using the toolbox from Section 5. We highlight a few small differences. First, we send the ABDLOP commitment in the first round, i.e. it is not given as a part of the statement. Second, we skip the rejection sampling steps in the prover algorithm since we only focus on knowledge soundness. In particular, we do not commit to the sign which is used for bimodal rejection sampling. Last but not least, for presentation purposes, we still keep $\mathbf{w}, v$ (unlike in Section 6.1) as part of the proof in order to follow the traditional Fiat-Shamir transformation without any optimisations.

The prover algorithm starts by generating the ABDLOP commitment to $\mathbf{s}_1$. Namely, it samples $\mathbf{s}_2 \leftarrow S_\nu^{m_2}$ and computes $\mathbf{t}_A := \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2$. Then comes the approximate range proof part. Concretely, it samples a masking $\mathbf{y}^{(e)} \leftarrow D^{256/d}_{\mathfrak{s}^{(e)}}$ and the corresponding commitment $\mathbf{t}^{(e)} := \mathbf{B}^{(e)}\mathbf{s}_2 + \mathbf{y}^{(e)}$. Hence, the first message of the prover and the corresponding challenge are

$$a_1 := \left(\mathbf{t}_A, \mathbf{t}^{(e)}\right) \quad \text{and} \quad \left(R_0^{(e)}, R_1^{(e)}\right) := \mathsf{RO}_1\left(\mathsf{pp}, x, a_1\right)$$

where $\mathsf{RO}_1 : \{0,1\}^{\leqslant u_1} \to \left(\{0,1\}^{256\times m_1 d}\right)^2$ is the first random oracle. From the challenge the prover computes the matrix $R^{(e)} := R_0^{(e)} - R_1^{(e)}$ which of the same distribution as $R^{(e)}$ in Fig. 9. For convenience, we define $\mathbf{r}_i^{(e)} \in \mathcal{R}_q^{m_1}$ to be the polynomial vector for which its coefficient vector is the i-th row of $R^{(e)}$ for $i \in [256]$.

Next, the prover computes $\vec{z}^{(e)} := R^{(e)}\vec{s}_1 + \vec{y}^{(e)}$. Further, it samples $\mathbf{g} := (g_1, \ldots, g_\lambda) \leftarrow \{x : \mathcal{R}_q : \tilde{x} = 0\}^\lambda$ and creates the corresponding commitment $\mathbf{t}_g := \mathbf{B}_g\mathbf{s}_2 + \mathbf{g}$. Thus, the second message of the prover and the corresponding challenge are

$$a_2 := \left(\vec{z}^{(e)}, \mathbf{t}_g\right) \quad \text{and} \quad (\gamma_{i,j})_{i\in[\lambda], j\in[257]} := \mathsf{RO}_2\left(\mathsf{pp}, x, a_1, a_2\right)$$

where $\mathsf{RO}_2 : \{0,1\}^{\leqslant u_2} \to \mathbb{Z}_q^{\lambda\times 257}$ is the second random oracle.

Now, the prover wants to prove that $\vec{z}^{(e)} = R^{(e)}\vec{s}_1 + \vec{y}^{(e)}$ and $\langle \vec{s}_1, \vec{s}_1\rangle = B$ modulo q. One observes that this is equivalent to proving that the constant coefficients of

$$\sigma\left(\mathbf{r}_j^{(e)}\right)^T\mathbf{s}_1 + X^{(j-1) \bmod d} \cdot y^{(e)}_{\lceil j/d\rceil} - z_j^{(e)} \text{ for } j \in [256], \quad \text{and} \quad \sigma(\mathbf{s}_1)^T\mathbf{s}_1 - B$$

are equal to zero. To this end, the prover computes

$$h_i := g_i + \sum_{j=1}^{256} \gamma_{i,j} \cdot \left(\sigma(\mathbf{r}_j)^T\mathbf{s}_1 + X^{(j-1) \bmod d} \cdot y^{(e)}_{\lceil j/d\rceil} - z_j^{(e)}\right) + \gamma_{i,257} \cdot \left(\sigma(\mathbf{s}_1)^T\mathbf{s}_1 - B\right) \text{ for } i = 1, 2, \ldots, \lambda. \quad (83)$$

Hence, the prover's next message and the corresponding challenge are

$$a_3 := (h_1, \ldots, h_\lambda) \quad \text{and} \quad (\mu_1, \ldots, \mu_{\lambda+n_1}) := \mathsf{RO}_3\left(\mathsf{pp}, x, a_1, a_2, a_3\right)$$

where $\mathsf{RO}_3 : \{0,1\}^{\leqslant u_3} \to \mathcal{R}_q^{\lambda+n_1}$ is the third random oracle.

Up to this point, the prover wants to prove $\lambda + n_1$ quadratic equations (with automorphisms) over $\mathcal{R}_q$, i.e. Equation 83 and $\mathbf{p}_i^T\mathbf{s}_1 = u_i$ for $i = 1, 2, \ldots, n_1$. By amortising, we only prove a single quadratic equation:

$$\begin{aligned} 0 = \sum_{i=1}^{\lambda} \mu_i \cdot &\left(g_i + \sum_{j=1}^{256} \gamma_{i,j} \cdot \left(\sigma(\mathbf{r}_j)^T\mathbf{s}_1 + X^{(j-1) \bmod d} \cdot y^{(e)}_{\lceil j/d\rceil} - z_j^{(e)}\right) + \gamma_{i,257} \cdot \left(\sigma(\mathbf{s}_1)^T\mathbf{s}_1 - B\right) - h_i\right) \\ &+ \sum_{i=1}^{n_1} \mu_{\lambda+i} \cdot \left(\mathbf{p}_i^T\mathbf{s}_1 - u_i\right). \end{aligned}$$

Let us define

$$\mathbf{B} := \begin{bmatrix} \mathbf{B}^{(e)} \\ \mathbf{B}_g \end{bmatrix}, \quad \mathbf{t}_B := \begin{bmatrix} \mathbf{t}^{(e)} \\ \mathbf{t}_g \end{bmatrix}, \quad \mathbf{m} := \begin{bmatrix} \mathbf{y}^{(e)} \\ \mathbf{g} \end{bmatrix} \in \mathcal{R}_q^{256/d+\lambda} \quad \text{and} \quad \mathbf{s} := \begin{bmatrix} \mathbf{s}_1 \\ \sigma(\mathbf{s}_1) \\ \mathbf{m} \\ \sigma(\mathbf{m}) \end{bmatrix}.$$

Then, the quadratic equation above can be written equivalently as

$$\mathbf{s}^T \mathbf{R}_2 \mathbf{s} + \mathbf{r}_1^T \mathbf{s} + r_0 = 0$$

where

$$\mathbf{R}_2 := \begin{bmatrix} \mathbf{0} & \mathbf{0}\,\mathbf{0}\,\mathbf{0} \\ \sum_{i=1}^{\lambda} \mu_i \cdot \gamma_{i,257} & \mathbf{0}\,\mathbf{0}\,\mathbf{0} \\ \mathbf{0} & \mathbf{0}\,\mathbf{0}\,\mathbf{0} \\ \mathbf{0} & \mathbf{0}\,\mathbf{0}\,\mathbf{0} \end{bmatrix} \in \mathcal{R}_q^{2(m_1+256/d+\lambda)\times 2(m_1+256/d+\lambda)}$$

$$\mathbf{r}_1 := \begin{bmatrix} \sum_{i=1}^{\lambda} \sum_{j=1}^{256} \mu_i \gamma_{i,j} \sigma(\mathbf{r}_j) + \sum_{i=1}^{n_1} \mu_{\lambda+i} \mathbf{p}_i \\ \mathbf{0} \\ \sum_{i=1}^{\lambda} \sum_{j=1}^{d} \mu_i \gamma_{i,j} X^{j-1} \\ \vdots \\ \sum_{i=1}^{\lambda} \sum_{j=1}^{d} \mu_i \gamma_{i,256/d-d+j} X^{j-1} \\ \mu_1 \\ \vdots \\ \mu_\lambda \\ \mathbf{0} \end{bmatrix} \in \mathcal{R}_q^{2(m_1+256/d+\lambda)}$$

$$r_0 = -\sum_{i=1}^{\lambda} (\mu_i \gamma_{i,257} B + h_i) - \sum_{i=1}^{n_1} \mu_{\lambda+i} u_i \in \mathcal{R}_q.$$

Finally, we run the sub-protocol for proving a single quadratic equation with automorphism. Namely, the prover samples masking vectors $\mathbf{y}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1}, \mathbf{y}_2 \leftarrow D_{\mathfrak{s}}^{m_2}$ and computes $\mathbf{w} = \mathbf{A}_1 \mathbf{y}_1 + \mathbf{A}_2 \mathbf{y}_2$. Then, it calculates $g_1 = \mathbf{s}^T \mathbf{R}_2 \mathbf{y} + \mathbf{y}^T \mathbf{R}_2 \mathbf{s} + \mathbf{r}_1^T \mathbf{y}$, where $\mathbf{y}$ is defined as

$$\mathbf{y} := \begin{bmatrix} \mathbf{y}_1 \\ \sigma(\mathbf{y}_1) \\ -\mathbf{B}\mathbf{y}_2 \\ -\sigma(\mathbf{B}\mathbf{y}_2) \end{bmatrix} \in \mathcal{R}_q^{2(m_1+256/d+\lambda)}, \tag{84}$$

and the commitment $t = \mathbf{b}^T \mathbf{s}_2 + g_1$ to g_1. Then, the prover sets $v = \mathbf{y}^T \mathbf{R}_2 \mathbf{y} + \mathbf{b}^T \mathbf{y}_2$. Hence, the prover's fourth message and the corresponding challenge are

$$a_4 := (t, \mathbf{w}, v) \quad \text{and} \quad c := \mathsf{RO}_4\left(\mathsf{pp}, x, a_1, a_2, a_3, a_4\right)$$

where $\mathsf{RO}_4 : \{0,1\}^{\leqslant u_4} \to \mathcal{C}$ is the fourth random oracle.

Further, given a challenge c, the prover computes $\mathbf{z}_i = c\mathbf{s}_i + \mathbf{y}_i$ for $i = 1, 2$. So, the prover's last message is

$$a_5 := (\mathbf{z}_1, \mathbf{z}_2)$$

and thus the proof consists of

$$\pi := (a_1, a_2, a_3, a_4, a_5).$$

The verifier is given a proof π and recomputes the corresponding challenges as well as $\mathbf{R}_2, \mathbf{r}_1, r_0$. Define

$$\mathbf{z} := \begin{bmatrix} \mathbf{z}_1 \\ \sigma(\mathbf{z}_1) \\ c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2 \\ \sigma(c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2) \end{bmatrix} \quad \text{and} \quad f := ct - \mathbf{b}^T\mathbf{z}_2. \tag{85}$$

Then, the verifier checks whether the following relations hold:

$$\begin{aligned}
&\|\vec{z}^{(e)}\| \overset{?}{\leqslant} B_e \\
&\tilde{h}_i \overset{?}{=} 0 \text{ for } i = 1, 2, \ldots, \lambda \\
&\|\mathbf{z}_1\| \overset{?}{\leqslant} B_1 \quad \text{and} \quad \|\mathbf{z}_2\| \overset{?}{\leqslant} B_2 \\
&\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 \overset{?}{=} \mathbf{w} + c\mathbf{t}_A \\
&\mathbf{z}^T\mathbf{R}_2\mathbf{z} + c\mathbf{r}_1^T\mathbf{z} + c^2 r_0 - f \overset{?}{=} v.
\end{aligned}$$

Knowledge Soundness. We are ready to prove the main result of this section.

Theorem B.7 (Knowledge Soundness). *Let $q > \max\left(B, 41 \cdot \frac{2}{\sqrt{26}} \cdot m_1 d B_e, \frac{2}{13}B_e^2 - B_e\right)$. Then, there exists an algorithm $\mathcal{E}$, called a knowledge extractor, with the following properties. Given public parameters* pp *and a statement x, as well as oracle access to any (potentially dishonest) Q-query random oracle prover $\mathcal{P}^*$, which outputs a valid proof with probability ε, the extractor runs in expected* $\mathsf{poly}(|x|, Q)$ *number of steps and with probability at least*

$$\varepsilon - \left(\frac{2}{|\mathcal{C}|} + q_1^{-d/2} + q_1^{-\lambda} + 2^{-128}\right)$$

it either outputs $\bar{\mathbf{s}}_1 \in \mathcal{R}_q^{m_1}$ such that $\mathbf{P}\bar{\mathbf{s}}_1 = \mathbf{u}$ and $\|\mathbf{s}_1\|^2 = B$, or a $\mathsf{MSIS}_{n,m_1+m_2,\bar{B}}$ solution for the matrix $[\mathbf{A}_1 \mid \mathbf{A}_2]$ where $\bar{B} := 4\eta\sqrt{B_1^2 + B_2^2}$.

Proof. Consider a dishonest Q-query random oracle prover $\mathcal{P}^*$ which provides a valid proof with probability ε. After making at most Q queries to the random oracles, $\mathcal{P}^*$ outputs a proof $\pi = (a_1, a_2, a_3, a_4, a_5)$. We reformat the output and define:

$$I_1 := (\mathsf{pp}, x, a_1), I_2 := (\mathsf{pp}, x, a_1, a_2), \ldots, I_5 := (\mathsf{pp}, x, a_1, a_2, a_3, a_4, a_5)$$

as $\mathcal{P}^*$'s output. As in [AFK21], we extend $\mathcal{P}^*$ to a random oracle algorithm $\mathcal{A}$ which also checks the validity of the proof π. Namely, by relaying all the random oracle queries $\mathcal{P}^*$ is making, $\mathcal{A}$ runs $\mathcal{P}^*$ to obtain $\mathbf{I} = (I_1, I_2, I_3, I_4, I_5)$ and π. Then, it obtains the challenges by querying

$$\left(R_0^{(e)}, R_1^{(e)}\right) := \mathsf{RO}_1(I_1), \quad (\gamma_{i,j})_{i \in [\lambda], j \in [257]} := \mathsf{RO}_2(I_2), \quad (\mu_1, \ldots, \mu_{\lambda+n_1}) := \mathsf{RO}_3(I_3), \quad c := \mathsf{RO}_4(I_4)$$

and then outputs

$$\mathbf{I}, \quad y := \left(\mathsf{pp}, x, a_1, (R_0^{(e)}, R_1^{(e)}), a_2, (\gamma_{i,j}), a_3, (\mu_i), a_4, c, a_5\right), \quad \text{and} \quad v := V(\mathsf{pp}, x, y)$$

where $V(\mathsf{pp}, x, y) = 1$ if y is the accepting transcript for statement x with public parameters pp, and 0 otherwise. It is easy to see that $\mathcal{A}$ makes at most $Q + 4$ random oralce queries and the probability that $\mathcal{A}$ outputs $v = 1$ is exactly ε.

The goal now is to define an extractor $\mathcal{E}$, which given black-box access to $\mathcal{A}$, either outputs $\bar{s}_1 \in \mathcal{R}_q^{m_1}$ such that $\mathbf{P}\bar{\mathbf{s}}_1 = \mathbf{u}$ and $\|\bar{\mathbf{s}}_1\| = B$, or a $\mathsf{MSIS}_{n,m_1+m_2,\bar{B}}$ solution for the matrix $[\mathbf{A}_1 \mid \mathbf{A}_2]$. To this end, we

```
Black-box access to: $\mathcal{A}$

1. Run $\mathcal{A}$ with randomness $\rho \leftarrow R$ as follows: relay the $Q + \mu$ random oracle queries to the random oracles
   and record all query-response pairs. Set $i = I_4$ and let $c_i$ be the response to query $i$. Obtain $(\mathbf{I}, y_0, v)$.
2. If $v = 0$, abort and return $v = 0$.
3. Otherwise, repeat:
   – sample $c'_i \in \mathcal{C}\backslash\{c_i\}$ without replacement,
   – run $\mathcal{A}$ with randomness $\rho$ as follows to obtain $(\mathbf{I}', y', v')$: answer the query to $i$ with $c'_i$ while answering
     all the other queries consistently if the query was performed by $\mathcal{A}$ in the previous run, and with fresh
     random value otherwise,
   until either 2 additional challenges $c'_i$ with $v' = 1$ and $I'_4 = I_4$ have been found or until all challenges $c'_i \in \mathcal{C}$
   have been tried.
4. If the latter case occurs, output $v = 0$.
5. For $i = 0, 1, 2$, let $y_i := \left(\mathsf{pp}, x, a_1, (R_0^{(e)}, R_1^{(e)}), a_2, (\gamma_{i,j}), a_3, (\mu_i), a_4, c^{(i)}, a_5^{(i)}\right)$ be the extracted transcripts.
6. Parse $a_5^{(i)} := (\mathbf{z}_1^{(i)}, \mathbf{z}_2^{(i)})$ and $a_1, a_2, a_3, a_4$ as in the protocol description above.
7. Define $\bar{c} := c^{(1)} - c^{(0)}, \bar{\mathbf{s}}_i = \frac{\mathbf{z}_i^{(1)} - \mathbf{z}_i^{(0)}}{c^{(1)} - c^{(0)}}$ for $i = 1, 2$ and $\bar{\mathbf{m}} = (\bar{\mathbf{y}}^{(e)}, \bar{\mathbf{g}}) := \mathbf{t}_B - \mathbf{B}\bar{\mathbf{s}}_2$.
8. If $\mathbf{z}_i^{(1)} - c^{(1)}\bar{\mathbf{s}}_i \neq \mathbf{z}_i^{(2)} - c^{(2)}\bar{\mathbf{s}}_i$ for some $i = 1, 2$:
   – return $\mathbf{I}$, $\begin{bmatrix} \bar{c} \cdot \left(\left(\mathbf{z}_1^{(2)} - \mathbf{z}_1^{(1)}\right) - \left(c^{(2)} - c^{(1)}\right)\bar{\mathbf{s}}_1\right) \\ \bar{c} \cdot \left(\left(\mathbf{z}_2^{(2)} - \mathbf{z}_2^{(1)}\right) - \left(c^{(2)} - c^{(1)}\right)\bar{\mathbf{s}}_2\right) \end{bmatrix}$ as the $\mathsf{MSIS}$ solution to $[\mathbf{A}_1 \mid \mathbf{A}_2]$ and $v = 1$.
9. Otherwise, return $\mathbf{I}$, $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}}, \bar{c})$ and $v = 1$.
```

Fig. 20: Subextractor $\mathcal{E}_4$ as a $(Q + 4)$-query random oracle algorithm [AFK21].

introduce a sequence of subextractors $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$ which are $(Q+4)$-query random oracle algorithms. Then, we will define $\mathcal{E}$ which runs $\mathcal{E}_1$ and answers its queries using lazy sampling.

We start by defining the extractor $\mathcal{E}_4$, which is a $(Q+4)$-query random oracle algorithm in Fig. 20. Informally, $\mathcal{E}_4$ focuses on finding the witness for a single quadratic equation with automorphism. We summarise the security properties of $\mathcal{E}_4$ below.

Lemma B.8. *The extractor $\mathcal{E}_4$ makes an expected number of at most $3 + 2Q$ queries to $\mathcal{A}$. Furthemore, $\mathcal{E}_4$ outputs $v = 1$ with probability at least*

$$\varepsilon - (Q + 1) \cdot \frac{2}{|\mathcal{C}|}.$$

Next, let $\mathbf{I} = (I_1, I_2, I_3, I_4, I_5)$ be the index vector obtained by $\mathcal{E}_4$ in the first step and denote

$$\left(R_0^{(e)}, R_1^{(e)}\right) := \mathsf{RO}_1\left(I_1\right), \quad (\gamma_{i,j})_{i \in [\lambda], j \in [257]} := \mathsf{RO}_2\left(I_2\right), \quad (\mu_1, \ldots, \mu_{\lambda+n_1}) := \mathsf{RO}_3\left(I_3\right).$$

as the corresponding (recorded) random oracle responses. Further, parse

$$I_4 := (\mathsf{pp}, x, \mathbf{t}_A, \mathbf{t}^{(e)}, \vec{z}^{(e)}, \mathbf{t}_g, h_1, \ldots, h_\lambda, t, \mathbf{w}, v).$$

Then, conditioned on $v = 1$, the extractor either returns $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}}, \bar{c}) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \mathcal{R}_q^{256/d+\lambda} \times \bar{\mathcal{C}}$ which satisfies the following relations:

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}^{(e)} \\ \mathbf{t}_g \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \bar{\mathbf{s}}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B}^{(e)} \\ \mathbf{B}_g \end{bmatrix} \bar{\mathbf{s}}_2 + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{y}}^{(e)} \\ \bar{\mathbf{g}} \end{bmatrix}, \quad \|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2B_1, \quad \|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2B_2$$

and

$$0 = \sum_{i=1}^{\lambda} \mu_i \cdot \left(\bar{g}_i + \sum_{j=1}^{256} \gamma_{i,j} \cdot \left(\sigma(\mathbf{r}_j)^T \bar{\mathbf{s}}_1 + X^{(j-1) \bmod d} \cdot \bar{y}^{(e)}_{\lceil j/d \rceil} - z^{(e)}_j \right) + \gamma_{i,257} \cdot \left(\sigma(\bar{\mathbf{s}}_1)^T \bar{\mathbf{s}}_1 - B \right) - h_i \right) + \sum_{i=1}^{n_1} \mu_{\lambda+i} \cdot \left(\mathbf{p}_i^T \bar{\mathbf{s}}_1 - u_i \right),$$

or a $\mathsf{MSIS}_{n,m_1+m_2,\bar{B}}$ *solution for the matrix* $[\mathbf{A}_1 \mid \mathbf{A}_2]$.

Proof. To bound the probability of outputting $v = 1$, we directly apply [AFK21, Proposition 2] for $K_m = 3$. This result[28] says that $\mathcal{E}_4$ makes at most $3 + Q \cdot (3-1)$ queries to $\mathcal{A}$ and the probability that $\mathcal{E}_4$ does returns 1 is at least $\varepsilon - (Q+1) \cdot \frac{2}{|\mathcal{C}|}$. Then, [AFK21, Proposition 1] says that by the end of Step 4 in Fig. 20, if $v \neq 0$ then $\mathcal{E}_4$ manages to extract a $(1,1,1,3)$-tree of accepting transcripts. Finally, we run the extraction strategy from the proof of Theorem 4.2 to extract the ABDLOP opening and a solution to the quadratic equation, or a MSIS solution. □

We now turn to defining the subextractor $\mathcal{E}_3$ which is informally responsible for proving knowledge of a solution of multiple quadratic equations. We describe the subextractor in Fig. 21. Here, $\mathcal{E}_3$ uses the early abort feature of $\mathcal{E}_4$, as described in [AFK21]. Namely, $\mathcal{E}_4$ computes the index vector $\mathbf{I}$ by running $\mathcal{P}^*$ as the first step. This allows the executions in the repeat loop of $\mathcal{E}_3$ to abort right after a single run of $\mathcal{P}^*$ if $I_3' \neq I_3$. This is highlighted in Step 3 of $\mathcal{E}_3$.

We now summarise the key properties of the extractor $\mathcal{E}_3$.

Lemma B.9. *The extractor* $\mathcal{E}_3$ *makes an expected number of at most* $6 + 5Q$ *queries to* $\mathcal{A}$. *Also,* $\mathcal{E}_3$ *outputs* $v = 1$ *with probability at least*

$$\varepsilon - (Q+1) \cdot \left(\frac{2}{|\mathcal{C}|} + q_1^{-d/2} \right).$$

Next, let $\mathbf{I} = (I_1, I_2, I_3, I_4, I_5)$ *be the index vector obtained by* $\mathcal{E}_3$ *in the first step and denote*

$$\left(R_0^{(e)}, R_1^{(e)} \right) := \mathsf{RO}_1\left(I_1\right), \quad (\gamma_{i,j})_{i \in [\lambda], j \in [257]} := \mathsf{RO}_2\left(I_2\right), \quad I_3 := (\mathsf{pp}, x, \mathbf{t}_A, \mathbf{t}^{(e)}, \vec{z}^{(e)}, \mathbf{t}_g, h_1, \ldots, h_\lambda).$$

Then, conditioned on $v = 1$, *the extractor either returns* $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}}, \bar{c}) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \mathcal{R}_q^{256/d+\lambda} \times \bar{\mathcal{C}}$ *which satisfies the following relations:*

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}^{(e)} \\ \mathbf{t}_g \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \bar{\mathbf{s}}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B}^{(e)} \\ \mathbf{B}_g \end{bmatrix} \bar{\mathbf{s}}_2 + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{y}}^{(e)} \\ \bar{\mathbf{g}} \end{bmatrix}, \quad \|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2B_1, \quad \|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2B_2$$

and

$$\begin{aligned} h_i &= \bar{g}_i + \sum_{j=1}^{256} \gamma_{i,j} \cdot \left(\sigma(\mathbf{r}_j)^T \bar{\mathbf{s}}_1 + X^{(j-1) \bmod d} \cdot \bar{y}^{(e)}_{\lceil j/d \rceil} - z^{(e)}_j \right) + \gamma_{i,257} \cdot \left(\sigma(\bar{\mathbf{s}}_1)^T \bar{\mathbf{s}}_1 - B \right) \text{ for } i = 1, 2, \ldots, \lambda, \\ \mathbf{p}^T \bar{\mathbf{s}}_1 &= u_i \text{ for } i = 1, 2, \ldots, n_1, \end{aligned} \tag{86}$$

or a $\mathsf{MSIS}_{n,m_1+m_2,\bar{B}}$ *solution for the matrix* $[\mathbf{A}_1 \mid \mathbf{A}_2]$.

Proof. This is the place where we apply techniques from Section B.2. First, note that while running $\mathcal{E}_3$, all the queries made by different invocations of $\mathcal{E}_4$ are answered consistently using lazy sampling, except for the query i where $\mathcal{E}_3$ provides with fresh and potentially different answers. Hence, this is indistinguishable from

[28] Even though the original result only considers determinstic provers $\mathcal{P}^*$, it can be extended to probabilistic provers by linearity of expected runtime and the success probability.

```
Black-box access to: $\mathcal{E}_4$

1. Run $\mathcal{E}_4$ with randomness $\rho \leftarrow R$ as follows: relay the $Q + \mu$ random oracle queries to the random oracles
   and record all query-response pairs. Set $i = I_3$ and let $(\mu_1, \ldots, \mu_{\lambda+n_1})$ be the response to query $i$. Obtain
   $(\mathbf{I}, y, v)$.
2. If $v = 0$, abort and return $v = 0$.
3. Otherwise, repeat:
   – sample $(\mu'_1, \ldots, \mu'_{\lambda+n_1}) \leftarrow \mathcal{R}_q^{\lambda+n_1}$,
   – run $\mathcal{E}_4$ with randomness $\rho$ as follows to obtain $(\mathbf{I}', y', v')$, aborting right after the initial run of $\mathcal{P}^*$ if
     $I'_3 \neq I_3$: answer the query to $i$ with $(\mu'_1, \ldots, \mu'_{\lambda+n_1})$ while answering all the other queries consistently if
     the query was performed by $\mathcal{E}_4$ in the previous run, and with fresh random value otherwise,
   until a challenge $(\mu'_1, \ldots, \mu'_{\lambda+n_1})$ with $v' = 1$ and $I'_3 = I_3$ has been found.
4. If $y$ (resp. $y'$) is a $\mathsf{MSIS}$ solution for the matrix $[\mathbf{A}_1 \mid \mathbf{A}_2]$, return $\mathbf{I}$, $y$ (resp. $y'$) and $v = 1$.
5. Parse $y = (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}} = (\bar{\mathbf{y}}^{(e)}, \bar{\mathbf{g}}), \bar{c})$ and $y' = (\bar{\mathbf{s}}'_1, \bar{\mathbf{s}}'_2, \bar{\mathbf{m}}' = (\bar{\mathbf{y}}^{(e)'}, \bar{\mathbf{g}}'), \bar{c}')$.
6. If $\bar{\mathbf{s}}_1 \neq \bar{\mathbf{s}}'_1$ or $\bar{\mathbf{s}}_2 \neq \bar{\mathbf{s}}'_2$, return $\mathbf{I}$, $\begin{bmatrix} \bar{c}\bar{c}'(\bar{\mathbf{s}}_1 - \bar{\mathbf{s}}'_1) \\ \bar{c}\bar{c}'(\bar{\mathbf{s}}_2 - \bar{\mathbf{s}}'_2) \end{bmatrix}$ as the $\mathsf{MSIS}$ solution to $[\mathbf{A}_1 \mid \mathbf{A}_2]$ and $v = 1$.
7. Parse $\left(R_0^{(e)}, R_1^{(e)}\right) := \mathsf{RO}_1\left(I_1\right), (\gamma_{i,j})_{i\in[\lambda],j\in[257]} := \mathsf{RO}_2\left(I_2\right)$ [a] and $I_3 := (\mathsf{pp}, x, \mathbf{t}_A, \mathbf{t}^{(e)}, \vec{z}^{(e)}, \mathbf{t}_g, h_1, \ldots, h_\lambda)$.
8. If all the following equations hold:
   – $h_i = \bar{g}_i + \sum_{j=1}^{256} \gamma_{i,j} \cdot \left(\sigma(\mathbf{r}_j)^T \bar{\mathbf{s}}_1 + X^{(j-1) \bmod d} \cdot \bar{y}^{(e)}_{\lceil j/d \rceil} - z_j^{(e)}\right) + \gamma_{i,257} \cdot \left(\sigma(\bar{\mathbf{s}}_1)^T \bar{\mathbf{s}}_1 - B\right)$ for $i = 1, \ldots, \lambda$,
   – $\mathbf{p}^T \bar{\mathbf{s}}_1 = u_i$ for $i = 1, 2, \ldots, n_1$,
   return $\mathbf{I}$, $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}}, \bar{c})$ and $v = 1$.
9. Otherwise, return $v = 0$.
```

[a] The extractor does not explicitly query the random oracles, but instead retrieves the outputs from the recorded list of query-response pairs from Step 1.

Fig. 21: Subextractor $\mathcal{E}_3$ as a $(Q + 4)$-query random oracle algorithm.

getting the responses from the full-fledged random oracles $\vec{j}_i \in C_i^{U_i}$ where $U_i = |\{0,1\}^{\leqslant u_i}|$ and C_i is the codomain of the i-th random oracle. Hence, the extractor is essentially running the abstract sampling game from Fig. 19. Indeed, we can treat the array M as the function table of the extractor $\mathcal{E}_4$. Given randomness r and random oracle tables $\vec{j}_i$, an entry of M consists of a tuple $(\mathbb{v}, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z})$, where $\mathbb{v}$ denotes the value of v, $\mathbb{h} = 3$ denotes the thirds random oracle, $\mathbb{i}$ is the value of $i = I_3$, $\mathbb{s}$ is the extracted vector (i.e. either a MSIS solution or a valid solution for the single quadratic equation) and $\mathbb{z} = \mathbf{I}$.

We now show how different parameters, such as $\mathcal{X}, P, T$, relate to our scenario. Let us define $\mathcal{X} = \mathcal{X}_0 \cup \mathcal{X}_1$ where

$$\mathcal{X}_0 := \left\{(\mathbf{x}, \mathbf{I}) \in \mathcal{R}_q^{m_1+m_2} \backslash \{\mathbf{0}\} \times \{0,1\}^* : \left[\mathbf{A}_1 \ \mathbf{A}_2\right] \mathbf{x} = \mathbf{0} \wedge \|x\| \leqslant \bar{B}\right\}$$

$$\mathcal{X}_1 := \left\{\left((\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{y}}^{(e)}, \bar{\mathbf{g}}), \mathbf{I}\right) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \mathcal{R}_q^{256/d+\lambda} \times \{0,1\}^* : \text{ equations in (86) hold}\right\}.$$

We claim that $\mathcal{X}$ is $q_1^{-d/2}$-probabilistically-testable w.r.t. M. To see that, suppose that $\mathcal{E}_4$ outputs $(\mathbb{v}, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z})$, where $\mathbb{v} = 1, \mathbb{h} = 3$, and suppose that $(\mathbb{s}, \mathbb{z}) \notin \mathcal{X}$. First, this means that the output of $\mathbb{s}$ is not a valid MSIS solution. But since $\mathbb{v} = 1$, we have $\mathbb{s} = (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{y}}^{(e)}, \bar{\mathbf{g}})$ and

$$0 = \sum_{i=1}^{\lambda} \mu_i \cdot \left(\bar{g}_i + \sum_{j=1}^{256} \gamma_{i,j} \cdot \left(\sigma(\mathbf{r}_j)^T \bar{\mathbf{s}}_1 + X^{(j-1) \bmod d} \cdot \bar{y}^{(e)}_{\lceil j/d \rceil} - z_j^{(e)}\right) + \gamma_{i,257} \cdot \left(\sigma(\bar{\mathbf{s}}_1)^T \bar{\mathbf{s}}_1 - B\right) - h_i\right)$$
$$+ \sum_{i=1}^{n_1} \mu_{\lambda+i} \cdot \left(\mathbf{p}_i^T \bar{\mathbf{s}}_1 - u_i\right).$$

If $\big((\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{y}}^{(e)}, \bar{\mathbf{g}}), \mathbf{I}\big) \notin \mathcal{X}$, then with probability at most $q_1^{-d/2}$ over the choice of $\mu'_1, \ldots, \mu'_{n_1+\lambda} \leftarrow \mathcal{R}_q$, we have

$$0 = \sum_{i=1}^{\lambda} \mu'_i \cdot \left(\bar{g}_i + \sum_{j=1}^{256} \gamma_{i,j} \cdot \left(\sigma(\mathbf{r}_j)^T \bar{\mathbf{s}}_1 + X^{(j-1) \bmod d} \cdot \bar{y}^{(e)}_{\lceil j/d \rceil} - z^{(e)}_j \right) + \gamma_{i,257} \cdot \left(\sigma(\bar{\mathbf{s}}_1)^T \bar{\mathbf{s}}_1 - B \right) - h_i \right)$$
$$+ \sum_{i=1}^{n_1} \mu'_{\lambda+i} \cdot \left(\mathbf{p}_i^T \bar{\mathbf{s}}_1 - u_i \right).$$

This means that if we run $\mathcal{E}_4$ for the same fixed random oracles apart from the i-th query to RO_3, where we pick a fresh random challenge $(\mu'_1, \ldots, \mu'_{n_1+\lambda})$, the extractor outputs exactly $(1, 3, i, (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{y}}^{(e)}, \bar{\mathbf{g}}), \mathbf{I}')$ with probability at most $q_1^{-d/2}$.

Next, to get the bounds on P and T used in Lemmas B.4 and B.6, we use the following observation from [AFK21]. First, the index vector outputted by $\mathcal{E}_4$ matches the index vector outputted by $\mathcal{P}^*$ by [AFK21, Lemma 7]. Further, for fixed randomness r, its output can only change if a random oracle is reprogrammed in one of the queried indices. Hence, for each randomness r and fixed random oracles $\vec{j}_1, \vec{j}_2, \vec{j}_3, \vec{j}_4$, there exists a set $S(r, \vec{j}_1, \vec{j}_2, \vec{j}_3, \vec{j}_4) \subseteq \{(\mathbb{h}, \mathbb{i}) : \mathbb{h} \in [4] \wedge \mathbb{i} \in \{0,1\}^{\leqslant u_1}\}$ (indicating queries made by $\mathcal{P}^*$) such that $\mathcal{P}^*$'s output will not change if reprogrammed at the $\mathbb{i}$-th index of the $\mathbb{h}$-th random oracle where $(\mathbb{h}, \mathbb{i}) \notin S(r, \vec{j}_1, \vec{j}_2, \vec{j}_3, \vec{j}_4)$. Hence, the conditions of Lemmas B.4 and B.6 are satisfied and thus $P \leqslant Q + 1$ and $T \leqslant Q$.

We can now analyse the algorithm $\mathcal{E}_3$. Concretely, by Lemmas B.3, B.4 and B.8, we get that $\mathcal{E}_3$ outputs $v = 1$ with probability at least

$$\left(\varepsilon - (Q+1) \cdot \frac{2}{|\mathcal{C}|} \right) - (Q+1) \cdot q_1^{-d/2} = \varepsilon - (Q+1) \cdot \left(\frac{2}{|\mathcal{C}|} + q_1^{-d/2} \right).$$

For the expected number of queries to $\mathcal{A}$, we apply the strategy from the proof of [AFK21, Proposition 2] and analyse the weighted version of the abstract sampling game where $\Gamma(r, \vec{j}_1, \vec{j}_2, \vec{j}_3, \vec{j}_4)$ is the (expected) cost of fully running $\mathcal{E}_4$ and $\gamma = 1$ which indicates the cost of an early abort invocation of $\mathcal{E}_4$. Then, by Lemmas B.5, B.6 and B.8 we get that the expected cost of running $\mathcal{E}_3$ is at most

$$2 \cdot (3 + 2Q) + Q = 6 + 5Q$$

which concludes the proof. □

Further, we turn to defining the subextractor $\mathcal{E}_2$ which is informally responsible for proving knowledge $\vec{s}_1$ and $\vec{y}$ such that $R^{(e)} \vec{s}_1 + \vec{y} = \vec{z}^{(e)}$. We describe the subextractor in Fig. 22. As before, $\mathcal{E}_2$ uses the early abort feature as $\mathcal{E}_3$. We summarise its security properties below.

Lemma B.10. *The extractor $\mathcal{E}_2$ makes an expected number of at most $12 + 11Q$ queries to $\mathcal{A}$. Also, $\mathcal{E}_2$ outputs $v = 1$ with probability at least*

$$\varepsilon - (Q+1) \cdot \left(\frac{2}{|\mathcal{C}|} + q_1^{-d/2} + q_1^{-\lambda} \right).$$

Next, let $\mathbf{I} = (I_1, I_2, I_3, I_4, I_5)$ be the index vector obtained by $\mathcal{E}_2$ in the first step and denote

$$\left(R_0^{(e)}, R_1^{(e)} \right) := \mathsf{RO}_1(I_1), \quad I_2 := (\mathsf{pp}, x, \mathbf{t}_A, \mathbf{t}^{(e)}, \vec{z}^{(e)}, \mathbf{t}_g).$$

Then, conditioned on $v = 1$, the extractor either returns: $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{y}}^{(e)}, \bar{\mathbf{g}}, \bar{c}) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \mathcal{R}_q^{256/d+\lambda} \times \bar{\mathcal{C}}$ which satisfies the following relations:

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}^{(e)} \\ \mathbf{t}_g \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \bar{\mathbf{s}}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B}^{(e)} \\ \mathbf{B}_g \end{bmatrix} \bar{\mathbf{s}}_2 + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{y}}^{(e)} \\ \bar{\mathbf{g}} \end{bmatrix}, \quad \|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2B_1, \quad \|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2B_2$$

```
Black-box access to: $\mathcal{E}_3$

1. Run $\mathcal{E}_3$ with randomness $\rho \leftarrow R$ as follows: relay the $Q + \mu$ random oracle queries to the random oracles and record all query-response pairs. Set $i = I_2$ and let $(\gamma_{i,j})_{i\in[\lambda],j\in[257]}$ be the response to query $i$. Obtain $(\mathbf{I}, y, v)$.
2. If $v = 0$, abort and return $v = 0$.
3. Otherwise, repeat:
   – sample $(\gamma'_{i,j})_{i\in[\lambda],j\in[257]} \leftarrow \mathbb{Z}_q^{\lambda\times 257}$,
   – run $\mathcal{E}_3$ with randomness $\rho$ as follows to obtain $(\mathbf{I}', y', v')$, aborting right after the initial run of $\mathcal{P}^*$ if $I'_2 \neq I_2$: answer the query to $i$ with $(\gamma'_{i,j})_{i\in[\lambda],j\in[257]}$ while answering all the other queries consistently if the query was performed by $\mathcal{E}_3$ in the previous run, and with fresh random value otherwise,
   until a challenge $(\gamma'_{i,j})$ with $v' = 1$ and $I'_2 = I_2$ has been found.
4. If $y$ (resp. $y'$) is a $\mathsf{MSIS}$ solution for the matrix $[\mathbf{A}_1 \mid \mathbf{A}_2]$, return $\mathbf{I}$, $y$ (resp. $y'$) and $v = 1$.
5. Parse $y = (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}} = (\bar{\mathbf{y}}^{(e)}, \bar{\mathbf{g}}), \bar{c})$ and $y' = (\bar{\mathbf{s}}'_1, \bar{\mathbf{s}}'_2, \bar{\mathbf{m}}' = (\bar{\mathbf{y}}^{(e)'}, \bar{\mathbf{g}}'), \bar{c}')$.
6. If $\bar{\mathbf{s}}_1 \neq \bar{\mathbf{s}}'_1$ or $\bar{\mathbf{s}}_2 \neq \bar{\mathbf{s}}'_2$, return $\mathbf{I}$, $\begin{bmatrix} \bar{c}\bar{c}'(\bar{\mathbf{s}}_1 - \bar{\mathbf{s}}'_1) \\ \bar{c}\bar{c}'(\bar{\mathbf{s}}_2 - \bar{\mathbf{s}}'_2) \end{bmatrix}$ as the $\mathsf{MSIS}$ solution to $[\mathbf{A}_1 \mid \mathbf{A}_2]$ and $v = 1$.
7. Parse $\left(R_0^{(e)}, R_1^{(e)}\right) := \mathsf{RO}_1(I_1)$ and $I_2 := (\mathsf{pp}, x, \mathbf{t}_A, \mathbf{t}^{(e)}, \vec{z}^{(e)}, \mathbf{t}_g)$.
8. If $\left(R_0^{(e)} - R_1^{(e)}\right)\vec{\bar{s}}_1 + \vec{\bar{y}}^{(e)} = \vec{z}^{(e)}$ and $\langle \vec{\bar{s}}_1, \vec{\bar{s}}_1 \rangle = B$, then return $\mathbf{I}$, $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}}, \bar{c})$ and $v = 1$.
9. Otherwise, return $v = 0$.
```

Fig. 22: Subextractor $\mathcal{E}_2$ as a $(Q + 4)$-query random oracle algorithm.

and

$$\left(R_0^{(e)} - R_1^{(e)}\right)\vec{\bar{s}}_1 + \vec{\bar{y}}^{(e)} = \vec{z}^{(e)}, \quad \langle \vec{\bar{s}}_1, \vec{\bar{s}}_1 \rangle = B, \quad \mathbf{p}^T\bar{\mathbf{s}}_1 = u_i \textit{ for } i = 1, 2, \ldots, n_1, \tag{87}$$

or a $\mathsf{MSIS}_{n,m_1+m_2,\bar{B}}$ *solution for the matrix* $[\mathbf{A}_1 \mid \mathbf{A}_2]$.

Proof. Almost identically as in the previous proof, we reduce the problem to the abstract sampling game from Fig. 19. Namely, let the array M be the function table of the extractor $\mathcal{E}_3$. Given randomness r and random oracle tables $\vec{j}_i$, an entry of M consists of a tuple $(\mathbb{v}, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z})$, where $\mathbb{v}$ denotes the value of v, $\mathbb{h} = 2$ denotes the second random oracle, $\mathbb{i}$ is the value of $i = I_2$, $\mathbb{s}$ is the extracted vector (i.e. either a MSIS solution or a valid solution for the multiple quadratic equations). and $\mathbb{z} = \mathbf{I}$, which in particular contains $(h_1, \ldots, h_\lambda)$ such that $\tilde{h}_1 = \ldots = \tilde{h}_\lambda = 0$ if $\mathbb{v} = 1$.

Let $\mathcal{X} = \mathcal{X}_0 \cup \mathcal{X}_1$ where

$$\mathcal{X}_0 := \left\{(\mathbf{x}, \mathbf{I}) \in \mathcal{R}_q^{m_1+m_2}\backslash\{\mathbf{0}\} \times \{0,1\}^* : \left[\mathbf{A}_1\ \mathbf{A}_2\right]\mathbf{x} = \mathbf{0} \wedge \|x\| \leqslant \bar{B}\right\}$$
$$\mathcal{X}_1 := \left\{\left((\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{y}}^{(e)}, \bar{\mathbf{g}}), \mathbf{I}\right) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \mathcal{R}_q^{256/d+\lambda} \times \{0,1\}^* : \text{ equations in (87) hold}\right\}.$$

We claim that $\mathcal{X}$ is $q_1^{-\lambda}$-probabilistically-testable w.r.t. M. To see that, suppose that $\mathcal{E}_3$ outputs $(\mathbb{v}, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z})$, where $\mathbb{v} = 1, \mathbb{h} = 2$, and suppose that $(\mathbb{s}, \mathbb{z}) \notin \mathcal{X}$. First, this means that the output of $\mathbb{s}$ is not a valid MSIS solution. Moreover, since $\mathbb{v} = 1$, we have $\mathbb{s} = (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{y}}^{(e)}, \bar{\mathbf{g}})$ and

$$h_i = \bar{g}_i + \sum_{j=1}^{256} \gamma_{i,j} \cdot \left(\sigma(\mathbf{r}_j)^T\bar{\mathbf{s}}_1 + X^{(j-1) \bmod d} \cdot \bar{y}^{(e)}_{\lceil j/d \rceil} - z_j^{(e)}\right) + \gamma_{i,257} \cdot \left(\sigma(\bar{\mathbf{s}}_1)^T\bar{\mathbf{s}}_1 - B\right) \text{ for } i = 1, 2, \ldots, \lambda$$
$$\mathbf{p}^T\bar{\mathbf{s}}_1 = u_i \text{ for } i = 1, 2, \ldots, n_1.$$

Now, because $(\mathbb{s}, \mathbb{z}) \notin \mathcal{X}$, either

$$\left(R_0^{(e)} - R_1^{(e)}\right)\vec{\bar{s}}_1 + \vec{\bar{y}}^{(e)} \neq \vec{z}^{(e)} \quad \text{or} \quad \langle \vec{\bar{s}}_1, \vec{\bar{s}}_1 \rangle \neq B.$$

Thus, for *any* polynomials $h'_1, \dots, h'_\lambda \in \mathcal{R}_q$ with constant coefficients equal to zero, with probability at most $q_1^{-\lambda}$ over the randomness of $(\gamma'_{i,j}) \leftarrow \mathbb{Z}_q^{\lambda \times 257}$, we have[29]

$$h'_i = \bar{g}_i + \sum_{j=1}^{256} \gamma'_{i,j} \cdot \left(\sigma(\mathbf{r}_j)^T \bar{\mathbf{s}}_1 + X^{(j-1) \bmod d} \cdot \bar{y}^{(e)}_{\lceil j/d \rceil} - z_j^{(e)}\right) + \gamma'_{i,257} \cdot \left(\sigma(\bar{\mathbf{s}}_1)^T \bar{\mathbf{s}}_1 - B\right) \text{ for } i = 1, 2, \dots, \lambda.$$

This means that if we run $\mathcal{E}_3$ for the same fixed random oracles apart from the i-th query to RO_2, where we pick a fresh random challenge $(\gamma'_{i,j})$, the extractor outputs exactly $(1, 2, i, (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{y}}^{(e)}, \bar{\mathbf{g}}), \mathbf{I}')$ with probability at most $q_1^{-\lambda}$. The rest of the proof follows identically as in the proof of Lemma B.9. □

Finally, we define the extractor $\mathcal{E}_1$ in Fig. 23 which is informally responsible for proving approximate shortness of $\vec{s}_1$. We then summarise the properties of $\mathcal{E}_1$ below.

```
Black-box access to: $\mathcal{E}_2$

1. Run $\mathcal{E}_2$ with randomness $\rho \leftarrow R$ as follows: relay the $Q + \mu$ random oracle queries to the random oracles and
   record all query-response pairs. Set $i = I_1$ and let $(R_0^{(e)}, R_1^{(e)})$ be the response to query $i$. Obtain $(\mathbf{I}, y, v)$.
2. If $v = 0$, abort and return $v = 0$.
3. Otherwise, repeat:
     – sample $(R_0^{(e)'}, R_1^{(e)'}) \leftarrow \left(\{0,1\}^{256 \times m_1 d}\right)^2$,
     – run $\mathcal{E}_2$ with randomness $\rho$ as follows to obtain $(\mathbf{I}', y', v')$, aborting right after the initial run of $\mathcal{P}^*$ if
       $I'_2 \neq I_2$: answer the query to $i$ with $(R_0^{(e)'}, R_1^{(e)'})$ while answering all the other queries consistently if
       the query was performed by $\mathcal{E}_2$ in the previous run, and with fresh random value otherwise,
   until a challenge $(R_0^{(e)'}, R_1^{(e)'})$ with $v' = 1$ and $I'_1 = I_1$ has been found.
4. If $y$ (resp. $y'$) is a $\mathsf{MSIS}$ solution for the matrix $[\mathbf{A}_1 \mid \mathbf{A}_2]$, return $\mathbf{I}$, $y$ (resp. $y'$) and $v = 1$.
5. Parse $y = (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{m}} = (\bar{\mathbf{y}}^{(e)}, \bar{\mathbf{g}}), \bar{c})$ and $y' = (\bar{\mathbf{s}}'_1, \bar{\mathbf{s}}'_2, \bar{\mathbf{m}}' = (\bar{\mathbf{y}}^{(e)'}, \bar{\mathbf{g}}'), \bar{c}')$.
6. If $\bar{\mathbf{s}}_1 \neq \bar{\mathbf{s}}'_1$ or $\bar{\mathbf{s}}_2 \neq \bar{\mathbf{s}}'_2$, return $\mathbf{I}$, $\begin{bmatrix} \bar{c}\bar{c}'(\bar{\mathbf{s}}_1 - \bar{\mathbf{s}}'_1) \\ \bar{c}\bar{c}'(\bar{\mathbf{s}}_2 - \bar{\mathbf{s}}'_2) \end{bmatrix}$ as the $\mathsf{MSIS}$ solution to $[\mathbf{A}_1 \mid \mathbf{A}_2]$ and $v = 1$.
7. If $\|\bar{\mathbf{s}}_1\| < \frac{2}{\sqrt{26}} B_e$, return $\mathbf{I}$, $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{y}}^{(e)}, \bar{c})$ and $v = 1$.
8. Otherwise, return $v = 0$.
```

Fig. 23: Subextractor $\mathcal{E}_3$ as a $(Q + 4)$-query random oracle algorithm.

Lemma B.11. *The extractor $\mathcal{E}_1$ makes an expected number of at most $24 + 23Q$ queries to $\mathcal{A}$. Also, $\mathcal{E}_1$ outputs $v = 1$ with probability at least*

$$\varepsilon - (Q + 1) \cdot \left(\frac{2}{|\mathcal{C}|} + q_1^{-d/2} + q_1^{-\lambda} + 2^{-128} \right).$$

Next, let $\mathbf{I} = (I_1, I_2, I_3, I_4, I_5)$ be the index vector obtained by $\mathcal{E}_1$ in the first step and denote $I_1 := (\mathsf{pp}, x, \mathbf{t}_A, \mathbf{t}^{(e)})$. Then, conditioned on $v = 1$, the extractor either returns: $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{y}}^{(e)}, \bar{c}) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \mathcal{R}_q^{256/d + \lambda} \times \bar{\mathcal{C}}$ which satisfies the following relations:

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}^{(e)} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \bar{\mathbf{s}}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B}^{(e)} \end{bmatrix} \bar{\mathbf{s}}_2 + \begin{bmatrix} \mathbf{0} \\ \bar{\mathbf{y}}^{(e)} \end{bmatrix}, \quad \|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2B_1, \quad \|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2B_2$$

and

$$\|\bar{\mathbf{s}}_1\| < \frac{2}{\sqrt{26}} B_e, \quad \langle \vec{\bar{s}}_1, \vec{\bar{s}}_1 \rangle = B, \quad \mathbf{p}^T \bar{\mathbf{s}}_1 = u_i \text{ for } i = 1, 2, \dots, n_1, \tag{88}$$

or a $\mathsf{MSIS}_{n, m_1 + m_2, \bar{B}}$ solution for the matrix $[\mathbf{A}_1 \mid \mathbf{A}_2]$

[29] This observation follows directly by analysing this equation on the level of constant coefficients.

Proof. As in the proof of Lemma B.9, we reduce the problem to the abstract sampling game. Let the array M be the function table of the extractor $\mathcal{E}_2$. Given randomness r and random oracle tables $\vec{j}_i$, an entry of M consists of a tuple $(\mathbb{v}, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z})$, where $\mathbb{v}$ denotes the value of v, $\mathbb{h} = 1$ denotes the first random oracle, $\mathbb{i}$ is the value of $i = I_1$, $\mathbb{s}$ is the extracted vector (i.e. either a MSIS solution or a valid solution for the multiple quadratic equations). and $\mathbb{z} = \mathbf{I}$, which in particular contains $\vec{z}^{(e)}$ such that $\|\vec{z}^{(e)}\| \leqslant B_e$ if $\mathbb{v} = 1$.

Let $\mathcal{X} = \mathcal{X}_0 \cup \mathcal{X}_1$ where

$$\begin{aligned}\mathcal{X}_0 &:= \left\{(\mathbf{x}, \mathbf{I}) \in \mathcal{R}_q^{m_1+m_2} \backslash \{\mathbf{0}\} \times \{0,1\}^* : \begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix} \mathbf{x} = \mathbf{0} \wedge \|x\| \leqslant \bar{B}\right\} \\ \mathcal{X}_1 &:= \left\{\left((\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{y}}^{(e)}), \mathbf{I}\right) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \mathcal{R}_q^{256/d} \times \{0,1\}^* : \text{equations in (88) hold}\right\}.\end{aligned}$$

We claim that $\mathcal{X}$ is 2^{-128}-probabilistically-testable w.r.t. M. Indeed, suppose that $\mathcal{E}_2$ outputs $(\mathbb{v}, \mathbb{h}, \mathbb{i}, \mathbb{s}, \mathbb{z})$, where $\mathbb{v} = 1, \mathbb{h} = 1$, and suppose that $(\mathbb{s}, \mathbb{z}) \notin \mathcal{X}$. First, this means that the output of $\mathbb{s}$ is not a valid MSIS solution. Because $\mathbb{v} = 1$, we have $\mathbb{s} = (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{y}}^{(e)}, \bar{\mathbf{g}})$ and

$$\left(R_0^{(e)} - R_1^{(e)}\right) \vec{\bar{s}}_1 + \vec{\bar{y}}^{(e)} = \vec{z}^{(e)}, \quad \langle \vec{\bar{s}}_1, \vec{\bar{s}}_1 \rangle = B, \quad \mathbf{p}^T \bar{\mathbf{s}}_1 = u_i \text{ for } i = 1, 2, \ldots, n_1.$$

Now, because $(\mathbb{s}, \mathbb{z}) \notin \mathcal{X}$, this means that $\|\mathbf{s}_1\| > \frac{2}{\sqrt{26}} B_e$. Then, by Lemma 2.9, for any vector $\vec{z}^{(e)'} \in \mathbb{Z}_q^{256}$ such that $\|\vec{z}^{(e)'}\| \leqslant B_e$ we have:

$$\Pr_{(R_0^{(e)'}, R_1^{(e)'}) \leftarrow \left(\{0,1\}^{256 \times m_1 d}\right)^2} \left[\left(R_0^{(e)'} - R_1^{(e)'}\right) \vec{\bar{s}}_1 + \vec{\bar{y}}^{(e)} = \vec{z}^{(e)'} \bmod q\right] \leqslant 2^{-128}.$$

This implies that if we run $\mathcal{E}_2$ for the same fixed random oracles apart from the i-th query to RO_1, where we pick a fresh random challenge $(R_0^{(e)'}, R_1^{(e)'})$, the extractor outputs exactly $(1, 1, i, (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{y}}^{(e)}), \mathbf{I}')$ with probability at most 2^{-128}. The rest of the proof follows identically as in the proof of Lemma B.9. □

Eventually, we define the extractor $\mathcal{E}$ to basically run $\mathcal{E}_1$ and answer any random oracle queries using lazy sampling. It then obtains a vector $\mathbf{s}_1 \in \mathcal{R}_q^{m_1}$ such that $\mathbf{P}\mathbf{s}_1 = \mathbf{u}$ over $\mathcal{R}_q$, as well as $\langle \vec{\bar{s}}_1, \vec{\bar{s}}_1 \rangle = B$ over $\mathbb{Z}_q$ and $\|\bar{\mathbf{s}}_1\| < \frac{2}{\sqrt{26}} B_e$. However, by assumption on q:

$$-q < -B \leqslant \|\bar{\mathbf{s}}_1\|^2 - B \leqslant \left(\frac{2}{\sqrt{26}} B_e\right)^2 - B_e = \frac{2}{13} B_e^2 - B_e < q$$

and thus $\|\bar{\mathbf{s}}_1\|^2 = B$ which concludes the proof. □