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, Ruschlikon, Switzerland
[2] ETH Zurich, Zurich, Switzerland

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

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

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 [36], 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 $\|\mathbf{s}\|$ [24, 25]. While such proofs are good enough for constructing fairly efficient basic protocols (e.g. signature schemes [24, 25, 4, 15]), 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. [35, 32]) 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 [26] 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 [28]) 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 [37] 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 [23], while more interesting constructions required outputs on the order of dozens of Megabytes (e.g. [22]). A noticeable improvement was achieved in [9] 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 [38, 11, 17] had proof sizes that were on the order of several hundred kilobytes. These schemes were greatly improved in [3, 16], 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 [31] 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 [6] 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) [16], and then prove that $(\mathbf{m} - \mathbf{1}) \cdot \mathbf{m} \cdot (\mathbf{m} + 1) = 0$ [3].

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. [1, 34]), 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 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.

[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. [19, 31, 18]) and proving various relationships between them.

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 [2] and BDLOP [6] 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 [26] 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

[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.

they can utilize a specific automorphism in this ring. Towards the end of this section and in the full version of the paper [27], 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 [2], one commits to a message $\mathbf{s}_1$ using randomness $\mathbf{s}_2$, where $\|\mathbf{s}_i\|$ are small, as

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

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 [6], 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 [6]) 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 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 [33].

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 $R_1\vec{s}_1 + R_m\vec{m} = \vec{u} \bmod q$.

[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.

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 [16], 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 $\mathcal{R}_q$.

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

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 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 $\widetilde{h} = 0 \bmod q$. Hence, the fact that this proof leaks no information and that the soundness error is again $1/q_1$ is virtually identical as for (7).

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 [3]. 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 [3] 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 [3] and then see how they can be applied to the ABDLOP commitment. Suppose, for example, that we wanted to prove that

[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 $\langle \vec{b}, (1, 2, 2^2, ...0, \ldots, 0) \rangle$ 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 $\vec{b}$.

$s_1 s_2 - s_3 = 0$, 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 [3] (and some that preceded it [11, 17, 38]) 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

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

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 5 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 $\mathbf{m}$,

[9] [3] showed that the y_i were already implicitly committed to by the first part of the protocol.

[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 $\sigma(X^i - X^{d-i}) = -X^{d-i} + X^i$.

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 7.

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 [7, 8, 29, 20] 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, and the general protocol proving these approximate range proofs in combination with other quadratic functions are given in the full version of the paper [27].

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 $\mathbf{As} = \mathbf{t} \bmod q$, one proves that

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

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 $q^{-1}(\mathbf{As}-\mathbf{t})$ mod 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 7.

Using Other Rings. In proving that the norm of a polynomial s was small, we exploited the fact that in the ring $\mathcal{R}$, $\widetilde{s(X^{-1})} \cdot s = \|s\|^2$ and that $s(X^{-1})$ was an automorphism. In the full version of the paper, 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 $R = \mathbb{Z}[X]/(X^d + f_{d-1}X^{d-1} + \ldots + f_1 X \pm 1)$, 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 the full version of the paper, 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 [13, 28]. 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 [1, 34].

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 quadratic functions and the full description of the group signature is given in the full version of the paper.

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 [1, 34], 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 [13, 28] 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 [26], 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
[28]	96KB	203KB	×
This Work	48KB	92KB	✓

Table 1: Our group signature and that of [28].

We compare the instantiation of the group signature from this paper to the previously most efficient one from [28] in Table 1. We mention that there are also tree-based group signatures (e.g. [18, 10]) 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).

[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).

	Proof Size
[30]	33KB
This Work	14KB

	Ciphertext Size	Proof Size	Decryption Time Independent of Forgery Time
[26]	9KB	9KB	×
[30][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 [30] 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 [26] and [30].

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 [26], 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 [30] that followed the framework of [3] and [16]. 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 the full version of the paper.

Acknowledgements. We would like to thank Ward Beullens for generalising Lemma 3 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 $\mathsf{Aut}(\mathcal{R})$ 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 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 [5, Lemma 1.5(i)].

Lemma 1. *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. *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 Module-SIS and Module-LWE Problems

Security of the [6] commitment scheme used in our protocols relies on the well-known computational lattice problems, namely Module-LWE (MLWE) and Module-SIS (MSIS) [21, 15]. Both problems are defined over $\mathcal{R}_q$.

Definition 3 ($\mathsf{MSIS}_{\kappa,m,B}$)**.** *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 4 ($\mathsf{MLWE}_{m,\lambda,\chi}$)**.** *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*

$$\begin{aligned}\Big|\Pr\left[b = 1 \,\middle|\, \mathbf{A} \leftarrow \mathcal{R}_q^{m \times \lambda};\, \boldsymbol{s} \leftarrow \chi^\lambda;\, \boldsymbol{e} \leftarrow \chi^m;\, b \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{A}\boldsymbol{s} + \boldsymbol{e})\right] \\ - \Pr\left[b = 1 \,\middle|\, \mathbf{A} \leftarrow \mathcal{R}_q^{m \times \lambda};\, \boldsymbol{b} \leftarrow \mathcal{R}_q^m;\, b \leftarrow \mathcal{A}(\mathbf{A}, \boldsymbol{b})\right]\Big| \geqslant \epsilon.\end{aligned} \tag{16}$$

We also recall the (simplified) Extended Module-LWE problem [30].

Definition 5 (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. $(\mathbf{B}, \mathbf{Br}, c, \mathbf{z}, \mathsf{sign}\left(\langle \mathbf{z}, c\mathbf{r}\rangle\right))$ *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. $(\mathbf{B}, \mathbf{u}, c, \mathbf{z}, \mathsf{sign}\left(\langle \mathbf{z}, c\mathbf{r}\rangle\right))$ *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*

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

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

2.4 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* [25].

Lemma 2 (Rejection Sampling [25, 14, 30]). *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 $\mathbf{y} \leftarrow D^\ell_{\mathfrak{s}}$, 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 $\rho \times D^\ell_{\mathfrak{s}}$.*
2. *Let $M = \exp(1/(2\gamma^2))$. Now, sample $\mathbf{v} \leftarrow \rho$ and $\mathbf{y} \leftarrow D^\ell_{\mathfrak{s}}$, 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 $\mathbf{v} \leftarrow \rho$, $\mathbf{z} \leftarrow D^{ld}_{\mathfrak{s}}$ 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 $\mathbf{y} \leftarrow D^\ell_{\mathfrak{s}}$, 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 $\rho \times D^\ell_{\mathfrak{s}}$.*

$\underline{\mathrm{Rej}_1(\vec{z}, \vec{v}, \mathfrak{s})}$
01 $u \leftarrow [0,1)$
02 If $u > \frac{1}{M} \cdot \exp\left(\frac{-2\langle \vec{z}, \vec{v}\rangle + \|\vec{v}\|^2}{2\mathfrak{s}^2}\right)$
03 return 1 (i.e. *reject*)
04 Else
05 return 0 (i.e. *accept*)

$\underline{\mathrm{Rej}_2(\vec{z}, \vec{v}, \mathfrak{s})}$
01 If $\langle \vec{z}, \vec{v}\rangle < 0$
02 return 1 (i.e. *reject*)
03 $u \leftarrow [0,1)$
04 If $u > \frac{1}{M} \cdot \exp\left(\frac{-2\langle \vec{z}, \vec{v}\rangle + \|\vec{v}\|^2}{2\mathfrak{s}^2}\right)$
05 return 1 (i.e. *reject*)
06 Else
07 return 0 (i.e. *accept*)

Fig. 1: Two rejection sampling algorithms: the one used generally in previous works [25] (left) and the one proposed recently in [30] (right).

We recall how parameters $\mathfrak{s}$ and M in the first statement Lemma 2 are selected. Concretely, the repetition rate M is chosen to be an upper-bound on:

$$\frac{D^\ell_{\mathfrak{s}}(\mathbf{z})}{D^\ell_{\mathbf{v},\mathfrak{s}}(\mathbf{z})} = \exp\left(\frac{-2\langle \mathbf{z}, \mathbf{v}\rangle + \|\mathbf{v}\|^2}{2\mathfrak{s}^2}\right) \leqslant \exp\left(\frac{28\mathfrak{s}\|\mathbf{v}\| + \|\mathbf{v}\|^2}{2\mathfrak{s}^2}\right) = M. \tag{17}$$

For the inequality we used the which says that with probability at least $1 - 2^{128}$ we have $|\langle \mathbf{z}, \mathbf{v}\rangle| < 14\mathfrak{s}\|\mathbf{v}\|$ for $\mathbf{z} \leftarrow D^\ell_{\mathfrak{s}}$ [5, 25]. Hence, by setting $\mathfrak{s} = 13\|\mathbf{v}\|$ we obtain $M \approx 3$.

Recently, Lyubashevsky et al. [30] 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:

$$\exp\left(\frac{-2\langle \mathbf{z}, \mathbf{v}\rangle + \|\mathbf{v}\|^2}{2\mathfrak{s}^2}\right) \leqslant \exp\left(\frac{\|\mathbf{v}\|^2}{2\mathfrak{s}^2}\right) = M. \tag{18}$$

Hence, for $M \approx 3$ one would select $\mathfrak{s} = 0.675 \cdot \|\mathbf{v}\|$. Note that the probability for $\mathbf{z} \leftarrow D^{\ell}_{\mathfrak{s}}$ 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 (17), then $\mathfrak{s} = 8 \cdot \|\mathbf{v}\|$. Thus, [30] 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 [14], since even though the value M is calculated exactly as in (18), the expected number of rejections is at most M and not $2M$. We summarise the results from [14, 30] in the latter two statements of Lemma 2.

$\mathrm{Rej}_0(\vec{z}, \vec{v}, \mathfrak{s})$
01 $u \leftarrow [0, 1)$
02 If $u > \frac{1}{M \exp\left(-\frac{\|\vec{v}\|^2}{2\mathfrak{s}^2}\right) \cosh\left(\frac{\langle \vec{z}, \vec{v} \rangle}{\sigma^2}\right)}$
03 return 1 (i.e. *reject*)
04 Else
05 return 0 (i.e. *accept*)

Fig. 2: Bimodal rejection sampling [14].

Finally, we highlight that the procedure in the second statement of Lemma 2 reveals the sign of $\langle \mathbf{z}, \mathbf{v} \rangle$. This is still fine when working with "one-time commitments" [30] 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 [30] where the adversary is given the additional one bit of information about the secret. We describe this problem in Section 2.3.

2.5 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 $\mathbf{r} \in \mathcal{R}^{\ell}_q$ 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 3. *Let $\mathbf{r} \in \mathcal{R}^{\ell}_q$ and $c \in \mathcal{R}_q$. Then, for any power-of-two k, we have $\|c\mathbf{r}\| \leqslant \sqrt[2k]{\|\sigma_{-1}(c^k)\, c^k\|_1} \cdot \|\mathbf{r}\|$.*

We provide the proof in the full version of the paper. In order to apply this lemma, we fix a power-of-two k and set the challenge space $\mathcal{C}$ as:

$$\mathcal{C} := \{c \in S^{\sigma}_{\kappa} : \sqrt[2k]{\|\sigma_{-1}(c^k)\, c^k\|_1} \leqslant \eta\} \tag{19}$$

where

$$S_\kappa^\sigma := \{c \in S_\kappa : \sigma(c) = c\}\,. \tag{20}$$

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 3 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 [33]. However, if we set $\sigma := \sigma_{-1}$ then we can apply our new result in the full version of the paper 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 3, 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). In the full version of the paper, we also show how one can use this commitment scheme and proof of knowledge to prove knowledge of linear relations over $\mathbb{Z}_q$. This latter proof is best modelled 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 (19). The standard deviations $\mathfrak{s}_1$ and $\mathfrak{s}_2$ are set as in Section 2.4 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 [30], 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 [25, 30] 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 [5].

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 4 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 [3], 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 $\Pi^{(1)}_{\mathsf{many}}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \ldots, f_N))$, 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 4. *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* $\mathbf{z}_i - c\bar{\boldsymbol{s}}_i = \mathbf{z}'_i - c'\bar{\boldsymbol{s}}_i$.

We present the proof of Lemma 4 in the full version of the paper.

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 1. Similarly as for [3], 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, however, we will only consider quadratic relations.

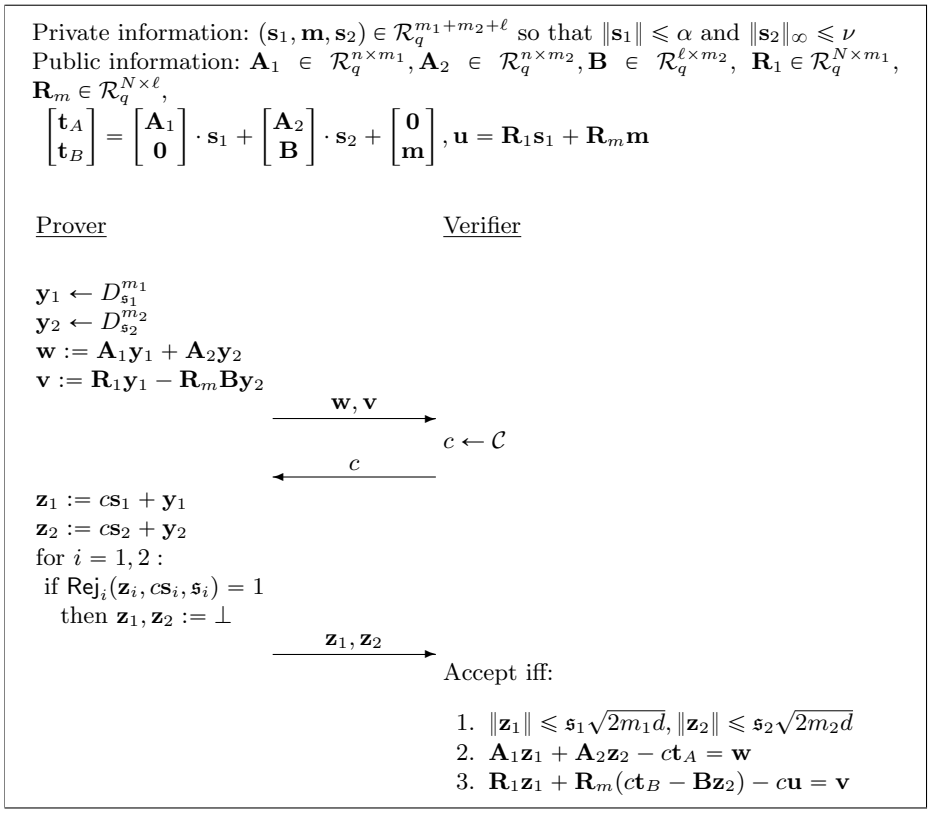

Fig. 4: Proof of knowledge $\Pi_{\mathsf{many}}^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \ldots, f_N))$ of $(\mathbf{s}_1, \mathbf{s}_2, \bar{c}) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \bar{\mathcal{C}}$ satisfying (i) $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A$, $\mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t}_B$ (ii) $\|\mathbf{s}_i\bar{c}\| \leqslant 2\mathfrak{s}_i\sqrt{2m_i d}$ for $i = 1, 2$ and (iii) $f_i(\mathbf{s}_1, \mathbf{m}) = 0$ for $i \in [N]$ 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_i 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.

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 (19) 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 $\mathbf{z}$:

$$\mathbf{y} := \begin{bmatrix} (\sigma^i(\mathbf{y}_1))_{i\in[k]} \\ -(\sigma^i(\mathbf{B}\mathbf{y}_2))_{i\in[k]} \end{bmatrix} \in \mathcal{R}_q^{k(m_1+\ell)} \tag{21}$$

and

$$\mathbf{z} := \begin{bmatrix} (\sigma^i(\mathbf{z}_1))_{i\in[k]} \\ (\sigma^i(\mathbf{z}_m))_{i\in[k]} \end{bmatrix} = c\begin{bmatrix} (\sigma^i(\mathbf{s}_1))_{i\in[k]} \\ (\sigma^i(\mathbf{m}))_{i\in[k]} \end{bmatrix} + \begin{bmatrix} (\sigma^i(\mathbf{y}_1))_{i\in[k]} \\ -(\sigma^i(\mathbf{B}\mathbf{y}_2))_{i\in[k]} \end{bmatrix} = c\mathbf{s} + \mathbf{y}. \tag{22}$$

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

$$\mathbf{z}^T\mathbf{R}_2\mathbf{z} + c\mathbf{r}_1^T\mathbf{z} + c^2 r_0 = c^2\left(\mathbf{s}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\mathbf{s} + r_0\right) + cg_1 + g_0 \tag{23}$$

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 (23) when $\mathbf{s}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\mathbf{s} + r_0 = 0$.

We present the full protocol in Fig. 5 which follows the commit-and-prove paradigm [12, 30]. Namely, we assume the prover has already sent the commitments $\mathbf{t}_A\mathbf{t}_B$ to the verifier using fresh randomness $\mathbf{s}_2 \leftarrow \chi^{m_2}$. Prover starts by sampling masking vectors $\mathbf{y}_1 \leftarrow D_{\mathbf{s}_1}^{m_1}, \mathbf{y}_2 \leftarrow D_{\mathfrak{s}}^{m_2}$ 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 (21), 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 (22) and f is defined as $f = ct - \mathbf{b}^T\mathbf{z}_2$.

We summarise security properties of the protocol in Fig. 5 in the full version of the paper.

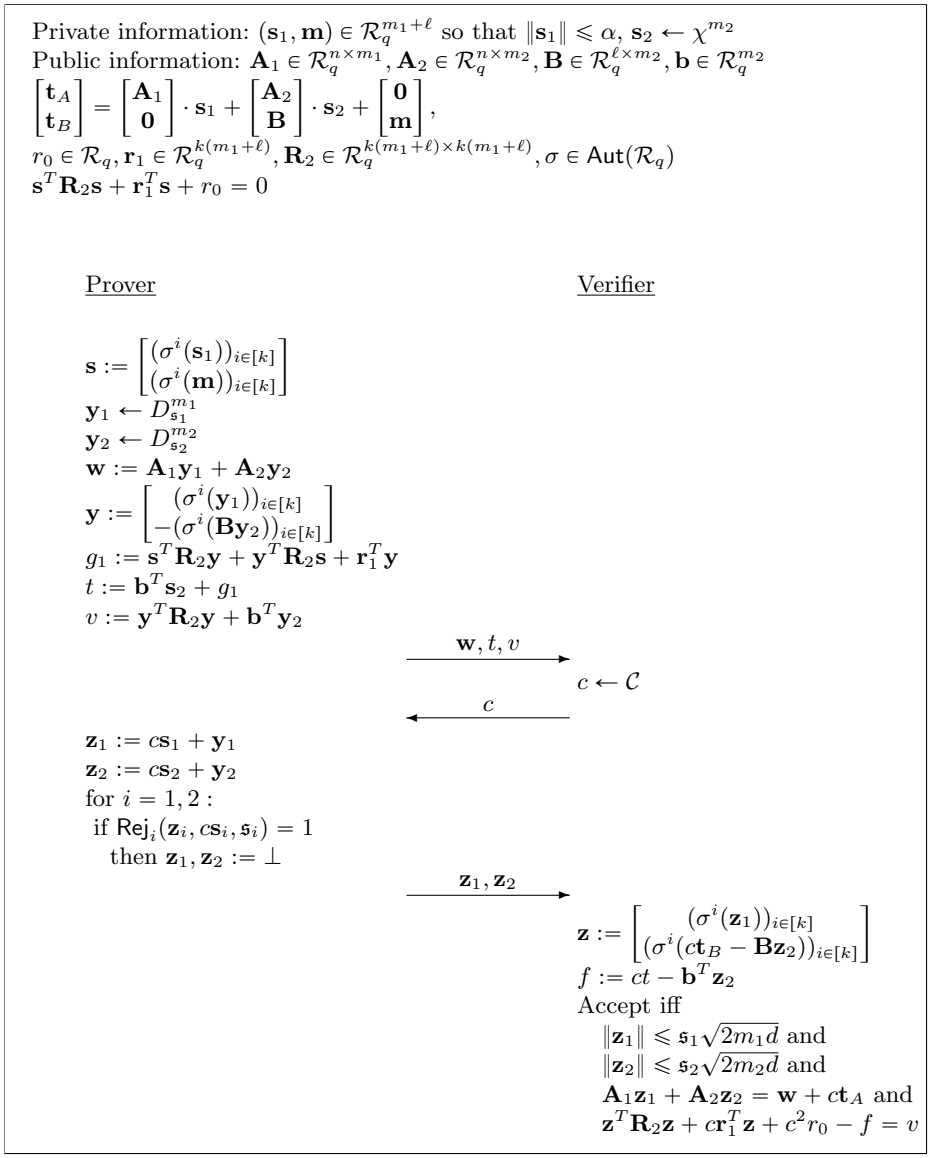

Fig. 5: Commit-and-prove protocol $\Pi^{(2)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), \sigma, f)$ for messages $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$, randomness $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$ and $\bar{c} \in \bar{\mathcal{C}}$ which satisfy: $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A, \mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t}_B$ (ii) $\|\mathbf{s}_i \bar{c}\| \leqslant 2\mathfrak{s}_i \sqrt{2m_i d}$ 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, $\mathbf{s}_2 \leftarrow \chi^{m_2}$.

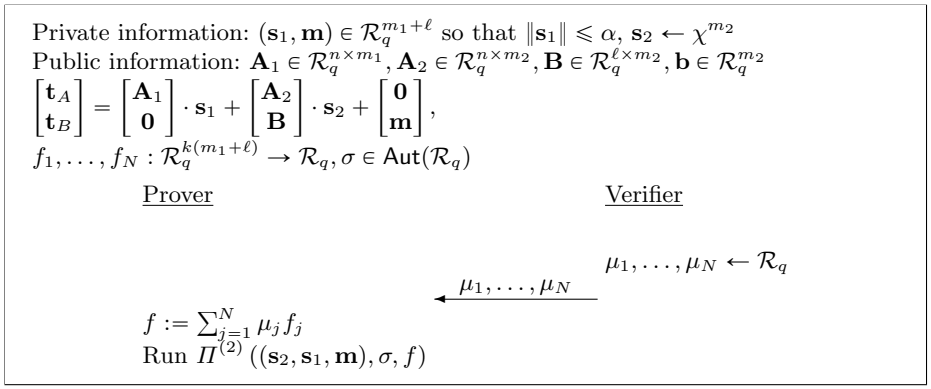

Fig. 6: Commit-and-prove protocol $\Pi^{(2)}_{\mathsf{many}}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), \sigma, (f_1, f_2, \ldots, f_N))$ for messages $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$, randomness $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$ and $\bar{c} \in \bar{\mathcal{C}}$ which 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) $\|\mathfrak{s}_i \bar{c}\| \leqslant 2\mathfrak{s}_i\sqrt{2m_i d}$ for $i = 1, 2$ (where $\mathfrak{s}_i$ are used in Fig. 5) 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 $\Pi^{(2)}$.

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. 5. 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

$$f_j\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) = 0 \text{ for } i \in [N]. \tag{24}$$

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

$$f\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) = 0 \tag{25}$$

using the protocol from Fig. 5. Now, we observe that if one of the conditions in (24) does not hold, then Equation 25 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. 6. We skip the full security analysis since it will be implicitly included in the more general case in the next subsection.

4.3 Polynomial Evaluations with Vanishing Constant Coefficients

Suppose we want to prove simultaneously N quadratic relations (i.e. (24)) 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 [16] 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

$$h := g + \sum_{j=1}^{M}\gamma_j F_j\left((\sigma^i(\mathbf{s}_1))_{i\in[k]},(\sigma^i(\mathbf{m}))_{i\in[k]}\right) \tag{26}$$

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. (26) 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,

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

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,(26) 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 (24). 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

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

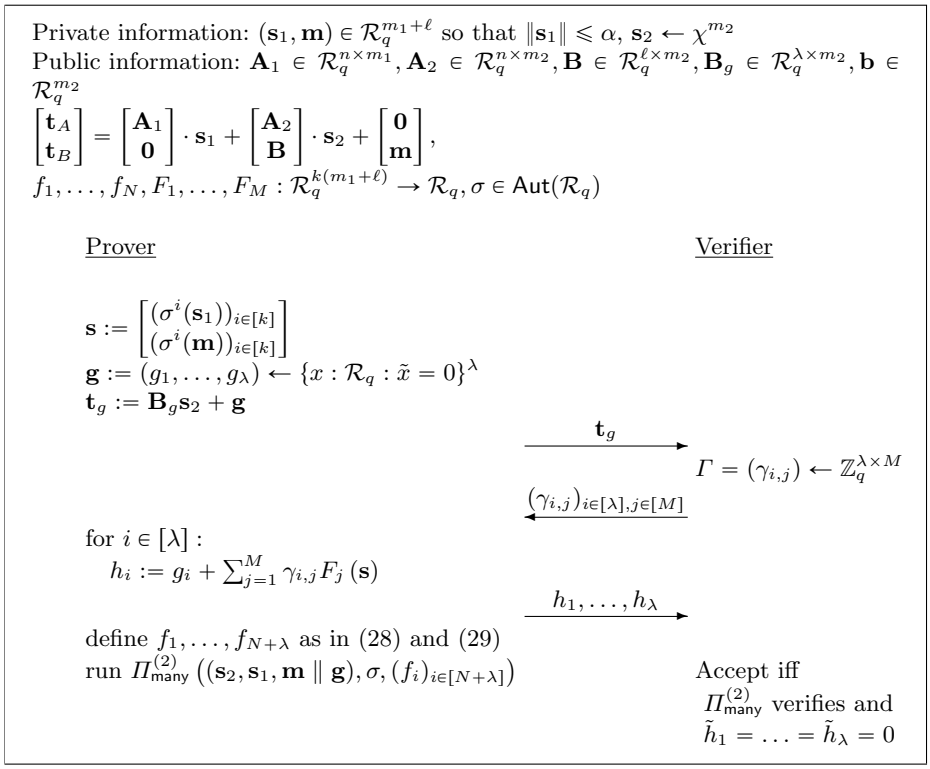

Fig. 7: Commit-and-prove protocol $\Pi^{(2)}_{\mathsf{eval}}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), \sigma, (f_1, \ldots, f_N), (F_1, \ldots, F_M))$ for messages $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$, randomness $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$ and $\bar{c} \in \bar{\mathcal{C}}$ 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 $i = 1, 2$, (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. 5) 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 $\Pi^{(2)}_{\mathsf{many}}$.

in Fig. 6.

We provide intuition for the soundness argument. Assume that the verifier is convinced that h is of the correct form (26) 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:

$$\begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_\lambda \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_\lambda \end{bmatrix} + \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,M} \\ \vdots & \vdots & \cdots & \vdots \\ \gamma_{\lambda,1} & \gamma_{\lambda,2} & \cdots & \gamma_{\lambda,M} \end{bmatrix} \begin{bmatrix} F_1\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) \\ F_2\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) \\ \vdots \\ F_M\left((\sigma^i(\mathbf{s}_1))_{i\in[k]}, (\sigma^i(\mathbf{m}))_{i\in[k]}\right) \end{bmatrix} \tag{27}$$

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,

$$f_{N+i}\left(\mathbf{x}_1, \mathbf{x}_2\right) := x_{2,1,i}^{(g)} + \sum_{j=1}^{M} \gamma_{i,j} F_j\left(\mathbf{x}_1, \mathbf{x}_2^{(m)}\right) - h_i \text{ for } i \in [\lambda]. \tag{28}$$

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 (27) 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 (24), for convenience we define polynomials $f_1, \ldots, f_N : \mathcal{R}_q^{k(m_1+\ell+\lambda)} \to \mathcal{R}_q$ as:

$$f_j(\mathbf{x}_1, \mathbf{x}_2) := f_j\left(\mathbf{x}_1, \mathbf{x}_2^{(m)}\right). \tag{29}$$

Finally, we simply run $\Pi_{\mathsf{quad-many}}\left((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}, \mathbf{g}), \sigma, (f_j)_{j\in[N+\lambda]}\right)$ from Fig. 6. We summarise the protocol in Fig. 7 and provide commitment and proof size analysis in the full version of the paper.

Note that with this approach we need to commit to additional λ garbage polynomials. In the full version of the paper we describe an optimisation which reduces the number of garbage polynomials by a factor of two in a scenario for $\sigma := \sigma_{-1}$. As discussed in the introduction, this will indeed be the automorphism that is going to be used throughout the paper.

References

1. S. Agrawal, D. Boneh, and X. Boyen. Efficient lattice (H)IBE in the standard model. In *EUROCRYPT*, pages 553–572, 2010.
2. M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *STOC*, pages 99–108, 1996.
3. T. Attema, V. Lyubashevsky, and G. Seiler. Practical product proofs for lattice commitments. In *CRYPTO (2)*, volume 12171 of *Lecture Notes in Computer Science*, pages 470–499. Springer, 2020.
4. S. Bai and S. D. Galbraith. An improved compression technique for signatures based on learning with errors. In *CT-RSA*, pages 28–47, 2014.
5. W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. *Mathematische Annalen*, 296(1):625–635, Dec 1993.
6. C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More efficient commitments from structured lattice assumptions. In *SCN*, pages 368–385, 2018.
7. C. Baum and V. Lyubashevsky. Simple amortized proofs of shortness for linear relations over polynomial rings. *IACR Cryptology ePrint Archive*, 2017:759, 2017.
8. C. Baum and A. 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.
9. W. 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.
10. W. Beullens, S. Dobson, S. Katsumata, Y. Lai, and F. Pintore. Group signatures and more from isogenies and lattices: Generic, simple, and efficient. *IACR Cryptol. ePrint Arch.*, page 1366, 2021.
11. J. Bootle, V. Lyubashevsky, and G. 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.
12. R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai. Universally composable two-party and multi-party secure computation. In *STOC*, pages 494–503. ACM, 2002.
13. R. del Pino, V. Lyubashevsky, and G. 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.
14. L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal gaussians. In *CRYPTO (1)*, pages 40–56, 2013.
15. L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):238–268, 2018.
16. M. F. Esgin, N. K. Nguyen, and G. Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In *ASIACRYPT (2)*, pages 259–288, 2020.
17. M. F. Esgin, R. Steinfeld, J. K. Liu, and D. Liu. Lattice-based zero-knowledge proofs: New techniques for shorter and faster constructions and applications. In *CRYPTO (1)*, pages 115–146. Springer, 2019.
18. M. F. Esgin, R. Steinfeld, and R. K. Zhao. Matrict+: More efficient post-quantum private blockchain payments. *IACR Cryptol. ePrint Arch.*, page 545, 2021.
19. M. F. Esgin, R. K. Zhao, R. Steinfeld, J. K. Liu, and D. Liu. Matrict: Efficient, scalable and post-quantum blockchain confidential transactions protocol. In *CCS*, pages 567–584. ACM, 2019.

20. C. Gentry, S. Halevi, and V. Lyubashevsky. Practical non-interactive publicly verifiable secret sharing with thousands of parties. *IACR Cryptol. ePrint Arch.*, page 1397, 2021.
21. A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *Designs, Codes and Cryptography*, 75(3):565–599, 2015.
22. B. Libert, S. Ling, K. Nguyen, and H. Wang. Zero-knowledge arguments for lattice-based accumulators: Logarithmic-size ring signatures and group signatures without trapdoors. In *EUROCRYPT (2)*, pages 1–31. Springer, 2016.
23. S. Ling, K. Nguyen, D. Stehlé, and H. Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In *PKC*, pages 107–124, 2013.
24. V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT*, pages 598–616, 2009.
25. V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT*, pages 738–755, 2012.
26. V. Lyubashevsky and G. Neven. One-shot verifiable encryption from lattices. In *EUROCRYPT*, 2017.
27. V. Lyubashevsky, N. K. Nguyen, and M. Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. *IACR Cryptol. ePrint Arch.*, page 284, 2022.
28. V. Lyubashevsky, N. K. Nguyen, M. Plançon, and G. Seiler. Shorter lattice-based group signatures via "almost free" encryption and other optimizations. In *ASIACRYPT (4)*, pages 218–248. Springer, 2021.
29. V. Lyubashevsky, N. K. Nguyen, and G. Seiler. Practical lattice-based zero-knowledge proofs for integer relations. In *CCS*, pages 1051–1070. ACM, 2020.
30. V. Lyubashevsky, N. K. Nguyen, and G. Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. In *Public Key Cryptography (1)*, pages 215–241. Springer, 2021.
31. V. Lyubashevsky, N. K. Nguyen, and G. 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.
32. V. Lyubashevsky, C. Peikert, and O. Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23, 2010.
33. V. Lyubashevsky and G. 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.
34. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT*, pages 700–718, 2012.
35. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), 2009.
36. C. Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO*, pages 239–252, 1989.
37. J. Stern. A new identification scheme based on syndrome decoding. In *CRYPTO*, pages 13–21, 1993.
38. R. Yang, M. H. Au, Z. Zhang, Q. Xu, Z. Yu, and W. Whyte. Efficient lattice-based zero-knowledge arguments with standard soundness: Construction and applications. In *CRYPTO (1)*, pages 147–175. Springer, 2019.