BLOOM: Bimodal Lattice One-Out-of-Many Proofs and Applications

Vadim Lyubashevsky[1] and Ngoc Khanh Nguyen[2]

[1] IBM Research Europe, Zurich
[2] EPFL, Lausanne

Abstract. We give a construction of an efficient one-out-of-many proof system, in which a prover shows that he knows the pre-image for one element in a set, based on the hardness of lattice problems. The construction employs the recent zero-knowledge framework of Lyubashevsky et al. (Crypto 2022) together with an improved, over prior lattice-based one-out-of-many proofs, recursive procedure, and a novel rejection sampling proof that allows to use the efficient bimodal rejection sampling throughout the protocol.

Using these new primitives and techniques, we give instantiations of the most compact lattice-based ring and group signatures schemes. The improvement in signature sizes over prior works ranges between 25% and 2X. Perhaps of even more significance, the size of the user public keys, which need to be stored somewhere publicly accessible in order for ring signatures to be meaningful, is reduced by factors ranging from 7X to 15X. In what could be of independent interest, we also provide noticeably improved proofs for integer relations which, together with one-out-of-many proofs are key components of confidential payment systems.

Keywords: lattices, zero-knowledge, one-out-of-many proofs, ring signatures

1 Introduction

Zero-knowledge proofs are the cornerstone of privacy-enabling cryptography and the ones based on lattice assumptions appear to currently be the most practical potentially quantum-resistant variants. The fundamental hard problem upon which lattice cryptography is based on is finding a vector $\vec{s}$ with a small norm satisfying $A\vec{s} = \vec{t} \bmod p$. Rapid recent progress in the area resulted in the proof size for proving pre-image knowledge in this basic equation to be reduced from being on the order of megabytes [LNSW13] to as short as a dozen kilobytes [YAZ+19, ESLL19, BLS19, ALS20, ENS20, LNS21a, LNP22].

A very useful extension of proving knowledge of a pre-image is proving knowledge of a pre-image for one element in a set. That is, given a set $\{\vec{t}_1, \ldots, \vec{t}_m\}$, one would like to prove knowledge of a short vector $\vec{s}$ such that $A\vec{s} = t_i \bmod p$ without leaking any information about the $\vec{s}$ or the i. This type of a proof is related to concepts such as set membership proofs and one-out-of-many proofs [GK15].[3] These proofs have applications to ring signatures, group signatures, confidential transactions, anonymous credentials, and various other privacy-enhancing cryptographic primitives.

In this work, we improve upon existing lattice-based one-out-of-many proofs and, based on this new building block, construct the most efficient quantum-safe ring and group signatures. Our improvement uses the high-level idea from the recursive algorithm of [LNS21b], but using a different and simpler recursive step which is made possible in part by being able to prove the base case using the recent framework for zero-knowledge proofs [LNP22]. We also give a general improvement to the rejection sampling step present in most lattice-based zero-knowledge proofs. Specifically, we show that when using the zero-knowledge framework from [LNP22], which gives the most efficient linear-size proofs for quadratic relations, one can use the more efficient bimodal rejection sampling procedure [DDLL13] everywhere. In some cases, this requires an

[3] The original formal definition of a one-out-of-many proof from [GK15] is more restricted in that $A\vec{s}$ is a commitment to 0 rather than just an evaluation of a one-way function on $\vec{s}$. But we do not need to restrict to this definition in this work.

extra short commitment, but in the case of the one-out-of-many proof given in the current work, using the more efficient rejection sampling step comes completely for free and noticeably reduces the proof size. We additionally show how to apply this bimodal rejection technique, together with the framework from [LNP22], to create more efficient proofs for integer relations such as addition and multiplication. Like one-out-of-many proofs, proving integer relations is a component of confidential transaction systems, and we believe that our improved tools can be used to make such systems (e.g. [LNS21b, ESZ21]) more efficient.

1.1 Results and Techniques

One-out-of-many proofs. The general equation for a lattice-based one-out-of-m proof can be written as

$$T\vec{v} = A\vec{s} \bmod p \tag{1}$$

where $\vec{v}$ is an m-dimensional unit vector (i.e. a vector consisting of one 1 and the rest zeroes) and $\vec{s}$ is a pre-image to the column in T chosen by the unit vector. For our end application, we would like to prove that $\vec{s}$ has a small norm, but we will give a proof system for slightly more general statements, as the generality is useful in the recursive nature of the protocol. Given a commitment to a vector $\vec{v}$ and $\vec{s}$, we would like to be able to prove that $\vec{v}$ and $\vec{s}$ satisfy (1), $\vec{v}$ is a unit vector, and $\vec{s}$ additionally satisfies some arbitrary quadratic relations $f_i(\vec{s}) = 0$.[4] In applications to group and ring signatures, the dimension of $\vec{s}$ is quite small, and so it is enough to only strive for proofs that are linear in the input size. Note that to keep the size of the proof linear, we still need it to be logarithmic in m since a unit vector of dimension m only has entropy $\log m$.

The proof of knowledge of (1) follows the commit-and-prove paradigm. That is, we commit to the secrets $\vec{v}$ and $\vec{s}$ and then prove that the committed messages satisfy the requisite relations. Keeping the proof linear in $\log m$ is the main technical challenge in building a one-out-of-many proof since one can't simply commit and open the naive m-dimensional representation of $\vec{v}$. Instead, one can write the unit vector $\vec{v}$ as a tensor product of a logarithmic number of smaller dimensional unit vectors, commit to these unit vectors, and then proceed to recursively prove the relation.

We begin with the base case – proving the knowledge of a unit vector $\vec{v} \in \{0,1\}^d$ (we can think of d being a constant with respect to m) and a vector $\vec{s}$ satisfying $T\vec{v} = A\vec{s}$ where $f_i(\vec{s}) = 0$ for arbitrary quadratic functions f_i. The most efficient known proof for this statement directly follows from the recent framework of [LNP22] where one commits to the vector $\vec{v}$ and $\vec{s}$ using the "ABDLOP" commitment scheme defined in that work (it is a combination of the Ajtai [Ajt96] and BDLOP [BDL+18] commitments – see (11) and Section 2.6) and then proves that the committed values satisfy

$$T\vec{v} = A\vec{s} \bmod p, \tag{2}$$

and $f_i(\vec{s}) = 0$, and $\|\vec{v}\| = 1$.[5]

Now suppose, by the inductive hypothesis, that having a commitment to $\vec{s}'$ and (some representation of) $\vec{v}' \in \{0,1\}^m$, we are able to prove that they satisfy the linear relation

$$T'\vec{v}' = A'\vec{s}' \bmod p \tag{3}$$

as well as the quadratic relations $f_i(\vec{s}') = 0$ and that $\vec{v}'$ is a unit vector. We will now show how to prove the relation in (3) for an arbitrary unit vector in $\{0,1\}^{\mathfrak{d}\cdot m}$. First, observe that any unit vector in $\{0,1\}^{\mathfrak{d}\cdot m}$ can be written as $\vec{v}\otimes\vec{v}' \in \{0,1\}^{\mathfrak{d}\cdot m}$, where $\vec{v}$ and $\vec{v}'$ are unit vectors in $\{0,1\}^{\mathfrak{d}}$ and $\{0,1\}^m$, respectively. So by writing an arbitrary $\mathfrak{d}\cdot m$-dimensional unit vector as $\vec{v}\otimes\vec{v}'$, we would like to prove (when having a commitment to $\vec{v}, \vec{v}', \vec{s}$) that

$$T(\vec{v}\otimes\vec{v}') = A\vec{s} \bmod p, \tag{4}$$

[4] Being able to prove quadratic relations, of course also allows one to prove that the ℓ_2-norm of $\vec{s}$ is smaller than some bound.

[5] This only proves that $\vec{v}$ is either a unit vector or a negative unit vector. But this is fine because proving knowledge of $\vec{v}, \vec{s}$ satisfying $\pm T\vec{v} = A\vec{s}$ is equivalent to (1).

that $f_i(\vec{s}) = 0$, and that $\vec{v}, \vec{v}'$ are unit vectors. To prove (4), we will prove that

$$\langle \vec{\varphi}_i, T(\vec{v} \otimes \vec{v}') - A\vec{s} \rangle = \vec{0} \bmod p \tag{5}$$

where $\vec{\varphi}_i$ for $i = 1, \ldots, \mathfrak{l}$ are randomly-chosen challenge vectors in $\mathbb{Z}_p^n$. Proving one such equation for a randomly-chosen $\vec{\varphi}_i$ would result in the proof having soundness error p^{-1}, and so we would like to prove $\mathfrak{l}$ such equations in order to achieve the desired soundness error of $p^{-\mathfrak{l}}$.

To prove (5) for one of $\mathfrak{l}$ different $\vec{\varphi}_i$, we decompose the matrix T as $T = [T_1 \ \ldots \ T_{\mathfrak{d}}]$ where $T_i \in \mathbb{Z}_p^{n \times m}$, and observe that by algebraic manipulation, we can rewrite

$$\langle \vec{\varphi}_i, T(\vec{v} \otimes \vec{v}') - A\vec{s} \rangle = \langle \vec{v}, \vec{w}_i \rangle - \vec{\varphi}_i^T A\vec{s} \bmod p, \tag{6}$$

where

$$\vec{w}_i = \begin{bmatrix} \vec{\varphi}_i^T T_1 \\ \cdots \\ \vec{\varphi}_i^T T_{\mathfrak{d}} \end{bmatrix} \cdot \vec{v}' \in \mathbb{Z}_p^{\mathfrak{d}}. \tag{7}$$

Each of the $\mathfrak{l}$ different $\vec{\varphi}_i$ leads to an equation of the above form and the prover thus commits to $\vec{w}_1, \ldots, \vec{w}_{\mathfrak{l}}$ and then using the inductive hypothesis from (3), he can show that

$$\begin{bmatrix} \vec{w}_1 \\ \cdots \\ \vec{w}_{\mathfrak{l}} \end{bmatrix} = T' \cdot \vec{v}' \bmod p \tag{8}$$

where the matrix T' is defined as

$$T' = \begin{bmatrix} \vec{\varphi}_1^T T_1 \\ \cdots \\ \vec{\varphi}_1^T T_{\mathfrak{d}} \\ \cdots \\ \vec{\varphi}_{\mathfrak{l}}^T T_1 \\ \cdots \\ \vec{\varphi}_{\mathfrak{l}}^T T_{\mathfrak{d}} \end{bmatrix} \in \mathbb{Z}_p^{\mathfrak{l} \cdot \mathfrak{d} \times m}. \tag{9}$$

The inductive hypothesis also allows him to prove that $f_i(\vec{s}) = 0$ and additionally that (the quadratic functions) $\langle \vec{v}, \vec{w}_i \rangle - \vec{\varphi}_i^T A\vec{s} = 0$. To see that the inductive hypothesis is applicable to proving these statements, we define the vector $\vec{s}'$ in (3) as

$$\vec{s}' = \begin{bmatrix} \vec{w} \\ \vec{s} \\ \vec{v} \end{bmatrix}, \text{ where } \vec{w} = \begin{bmatrix} \vec{w}_1 \\ \cdots \\ \vec{w}_{\mathfrak{l}} \end{bmatrix} \tag{10}$$

and the matrix A' as $[\ I \mid 0 \mid 0\ ]$, and thus $T'\vec{v}' = A'\vec{s}' = \vec{w}$. Since we assumed to have commitments to $\vec{v}'$ and we created commitments to all parts of $\vec{s}'$, we can prove (8) and the aforementioned quadratic relations involving $\vec{s}, \vec{w}$, and $\vec{v}$. The main point is that by additionally committing to $\vec{w}$ and $\vec{v}$, we are able to use the inductive hypothesis to prove relations where the unit vector is $\mathfrak{d}$ times longer.

The commitment scheme used in [LNP22] allows to naturally commit to polynomials in the ring $\mathcal{R}_p = \mathbb{Z}_p[X]/(X^d + 1)$ where, for optimal efficiency, $d = 64$ or 128. For efficiency of the protocol, we would then like to pack the $\mathfrak{l}$ commitments to $\vec{w}_i \in \mathbb{Z}_p^{\mathfrak{d}}$ into just one vector $\mathbb{Z}_p^d$, which can then be represented by one polynomial in $\mathcal{R}_p$. We can do this in the trivial way as long as $\mathfrak{d} \cdot \mathfrak{l} \leqslant d$. Then to compute inner products $\langle \vec{v}, \vec{w}_i \rangle$, we simply put the vector $\vec{v} \in \{0,1\}^{\mathfrak{d}}$ into a vector $\mathbb{Z}_p^d$ that contains $\vec{v}$ at the top and has the rest of its coefficients set to 0. Then $\langle \vec{v}, \vec{w}_i \rangle$ is the inner product of an appropriate shift of the vector containing the $\vec{v}$ and the vector containing the $\vec{w}_i$. A downward shift of a committed vector in $\mathbb{Z}_p^d$ whose bottom coefficients are all 0 is simply a multiplication of the commitment by the polynomial X in $\mathcal{R}_p$, which can be performed by the verifier.

Thus proving $T \cdot (\vec{v} \otimes \vec{v}') = A\vec{s}$, where $\vec{v} \otimes \vec{v}'$ is a $\mathfrak{d} \cdot m$ dimensional unit vector, requires the commitments needed for proving (3), an additional commitment to $\vec{v}$, and one more commitment to the $\vec{w}_i$. Since the base

case requires one commitment to $\vec{v}$ (and a commitment to $\vec{s}$), the total number of commitments to elements in $\mathcal{R}_p$ for proving (1) when $\vec{v}$ is an $m = \mathfrak{d}^k \cdot d$-dimensional unit vector is $2k + 1$ (and a commitments to $\vec{s}$), which is logarithmic in m and linear in the dimension of $\vec{s}$. In particular, we write the unit vector $\vec{v}$ in (1) as $\vec{v}_1 \otimes \ldots \otimes \vec{v}_k \otimes \vec{v}_{k+1}$ where $\vec{v}_1, \ldots, \vec{v}_k \in \{0,1\}^{\mathfrak{d}}$ and $\vec{v}_{k+1} \in \{0,1\}^d$. Then proving (1), that $\vec{v}_i$ are unit vectors, and $f_i(\vec{s}) = 0$ requires creating an ABDLOP commitment to the vectors $\vec{v}_i, \vec{s}$, and then also creating a commitment to the above-described vector $\vec{w}_i$ at each step of the proof.

Bimodal Gaussian Rejection Sampling Everywhere. The framework of [LNP22] uses the newly-defined ABDLOP commitment scheme to commit to a low-norm polynomial vector $\mathbf{s}_1$ and an arbitrary-norm polynomial vector $\mathbf{m}$. To do this, one generates a low-norm randomness $\mathbf{s}_2$ and outputs the commitment

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} := \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{s}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} \mathbf{s}_2 + \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix} \bmod q.^6 \tag{11}$$

Proving knowledge of $\mathbf{s}_1$ and $\mathbf{m}$ is done using the "Fiat-Shamir with Aborts" technique [Lyu09, Lyu12] where, upon generating low-norm masking vectors $\mathbf{y}_1$ and $\mathbf{y}_2$, computing $\mathbf{w} = \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2$, and receiving a challenge polynomial c, the prover creates the responses $\mathbf{z}_1 = \mathbf{y}_1 + c\mathbf{s}_1$ and $\mathbf{z}_2 = \mathbf{y}_2 + c\mathbf{s}_2$ which satisfy

$$\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 = c\mathbf{t}_A + \mathbf{w} \bmod q. \tag{12}$$

He now needs to perform rejection sampling on the $\mathbf{z}_i$ in order to not leak information about the $\mathbf{s}_i$. The generic setup from [Lyu12] that results in the smallest-norm $\mathbf{z}_i$ being sent involves $\mathbf{y}_i$ being sampled from a discrete Gaussian distribution while standard deviation is approximately a factor of 12 larger than $\|c\mathbf{s}_i\|$. With the appropriate rejection sampling step, this results in the polynomial vectors $\mathbf{z}_i$ also being distributed as discrete Gaussians with standard deviation $12 \cdot \|c\mathbf{s}_i\|$.

It was shown in [DDLL13] that if one first chooses a secret bit $b \in \{-1, 1\}$ and creates $\mathbf{z}_i = \mathbf{y}_i + bc\mathbf{s}_i$ (which is a bimodal distribution with two peaks at $\pm c\mathbf{s}_i$), then one can choose the $\mathbf{y}_i$ from a discrete Gaussian distribution with standard deviation $\|c\mathbf{s}_i\|/\sqrt{2}$ and via appropriate rejection sampling, the standard deviation of $\mathbf{z}_i$ ends up being $\|c\mathbf{s}_i\|/\sqrt{2}$ as well, which is around 17X smaller than that in the previous paragraph. The outputs $\mathbf{z}_i$ having a smaller standard deviation means that the proof size will be noticeably smaller, the modulus q can be set smaller as well, which in turn results in a smaller commitment size. The main technical difficulty with using the bimodal distribution is that the bit b needs to remain secret and the $\mathbf{z}_i$ need to satisfy the verification equation (12) irrespective of the b, which implies that we need to have $\mathbf{A}_1 b\mathbf{s}_1 + \mathbf{A}_2 b\mathbf{s}_2 = c\mathbf{t}_A \bmod q$. This set-up exists in the special case of the BLISS signature scheme [DDLL13] where the modulus is set to be even, and one can also force it by modifying the equation being proved as in [TWZ20], but these techniques would extol a high extra cost on the output size of the zero-knowledge proofs.

In our work, we show how one can use the bimodal rejection sampling technique for masking the secret vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ in (11) either completely for free, or at a small increase in the commitment size. We note that using a rejection sampling procedure that had similar properties as bimodal rejection sampling was already employed in [LNS21a] on the *randomness vector* $\mathbf{s}_2$. The rejection step there allowed for a smaller standard deviation at the cost of leaking one bit of $\mathbf{s}_2$. This leakage is not a problem because the commitment scheme from (11) (and the related commitment in [LNS21a]) is used inside a commit-and-prove approach to constructing zero-knowledge proofs. In particular, when trying to prove some relation (e.g. (1)), the prover commits to the secret values ($\vec{s}$ and $\vec{v}$ in the case of (1)) using the commitment in (11) and proves relations about $\mathbf{s}_1$ and $\mathbf{m}$, which in turn implies the initial relation he set out to prove. The important part is that the commitment scheme is only used once – if the prover is to perform another proof, he will create another commitment with different randomness $\mathbf{s}_2$. Thus leaking a small part of the randomness $\mathbf{s}_2$ is not a problem

[6] We use modulus q here instead of p in the previous section to signify that the commitment scheme modulus need not be (and is usually not, though they could be related) the same as the modulus that one wants to prove relations over.

as long as the Module-LWE problem upon which the hiding of the commitment scheme is based on remains hard.

Our work improves on [LNS21a] in two ways. First, we show that directly using the bimodal rejection sampling on $\mathbf{z}_2$ (despite $\mathbf{z}_2$ not being distributed according to a bimodal distribution) only leaks a few more (i.e. $\log q$) bits of $\mathbf{s}_2$ but ends up saving a factor of 2 in the rejection sampling probability. Leaking $\log q$ bits still keeps the entropy of $\mathbf{s}_2$ very high. This problem has been investigated both theoretically and in practice [AP12, ASA16, DDGR20, BJRW21], and it does not seem that the LWE problem is weakened if a few bits of the randomness are leaked. The more interesting improvement is in also being able to use the bimodal rejection sampling on $\mathbf{z}_1$. Here one cannot leak anything because $\mathbf{s}_1$ is where the actual secret message is stored.[7] The new idea is to create a commitment to $b\mathbf{s}_1$, for a random $b \in \{-1, 1\}$ instead of to $\mathbf{s}_1$. We can show that creating such a commitment and then outputting $\mathbf{z}_1 = \mathbf{y}_1 + cb\mathbf{s}_1$ after applying the bimodal rejection sampling does not leak anything about $\mathbf{s}_1$ or b as long as the commitment is only used once.

There is, however, an obvious problem with committing to $b\mathbf{s}_1$ – the relation that one may want to prove about the message committed to in $\mathbf{s}_1$ may not hold true when the messages are negated. There are two cases here – the simple case is that what we would like to prove still holds true for the negated messages. In our one-out-of-many proof, we commit to unit vectors $\vec{v}_1, \ldots, \vec{v}_{k+1}$ and a low-norm vector $\vec{s}$ such that $\vec{v}_1 \otimes \ldots \otimes \vec{v}_{k+1} = \vec{v}$ such that (1) is satisfied. Note that if we're tensoring an even number of $\vec{v}_i$, then $\bigotimes_i \vec{v}_i = \bigotimes_i -\vec{v}_i$, and also $\|\vec{v}_i\| = 1 = \| - \vec{v}_i\|$ and $\|\vec{s}\| = \beta = \| - \vec{s}\|$.[8] Therefore proving (1) as well as $\|\vec{v}_i\| = 1$ and $\|\vec{s}\| = \beta$ can be done regardless of whether we committed to the positive or negative of these values.

In the case that we would like to prove some relations on some secret vector $\vec{s}$ which are not sign-independent, we commit to a bit $b \in \{-1, 1\}$ in the $\mathbf{m}$ part of the ABDLOP commitment and to $\vec{s}' = b\vec{s}$, and then prove knowledge of a vector $\vec{s}'$ and a bit $b \in \{-1, 1\}$ satisfying $f_i(b\vec{s}') = 0$. Very importantly, note that $f_i(b\vec{s}')$ is still a quadratic equation because all the quadratic terms in f_i remain the same (since multiplication by $b \in \{-1, 1\}$ does not change them), and it is only the linear terms that get multiplied by b, thus becoming quadratic. We then prove that if the bit b is chosen randomly in $\{-1, 1\}$, then one can use the bimodal rejection sampling on $\mathbf{z}_1 = \mathbf{y}_1 + cb\mathbf{s}_1$ without leaking anything about the secret $\mathbf{s}_1$.

Applications to Ring and Group Signatures. Being able to prove (1) immediately gives us a construction of a ring signature scheme. In particular, every user has a secret/public key pair $\vec{s}_i, \vec{t}_i$, where $\|\vec{s}_i\|$ is small and $A\vec{s}_i = \vec{t}_i \bmod p$. Given a matrix T whose columns are the public keys of a group of users, the signature of user i of a message μ is a zero-knowledge proof of knowledge (with μ being used as an input into the random oracle of the Fiat-Shamir transform) of a unit vector $\vec{v}$ and a short vector $\vec{s}$ satisfying (1). The full details are given in Appendix C. In Figure 1, we compare an instantiation of our ring signature with other known potentially quantum-safe ones. Once the group size within which one wishes to hide is larger than a few hundred members, the size of our signature is the smallest, even including the isogeny-based construction of [BKP20]. Additionally, the size of the public key of our ring signatures can be as small as 128 bytes per user, which is a significant reduction over all prior lattice-based ring signatures.[9] Having small public keys is important because the public keys of *all* users need to be stored somewhere accessible by everyone who wishes to use the ring signature.

The reason for the significant reduction in the public key size over the previous lattice schemes is that we were able to adapt the new framework from [LNP22] as the base case of our recursive one-out-of-many

[7] One might be tempted to put the secret message into the $\mathbf{m}$ part of the commitment, which does not leak even if there is leakage in $\mathbf{s}_2$, but this results in a much less efficient commitment scheme because the dimension of the commitment grows linearly with the dimension of $\mathbf{m}$, whereas $\mathbf{s}_1$ has no effect on the size of the commitment. See Section 2.6.

[8] If we want to prove that $\|\vec{s}\| \leqslant \beta$, then we could create another commitment to a vector $\vec{s}' \in \mathcal{R}_q$ such that $\|\vec{s}\|^2 + \|\vec{s}'\|^2 = \beta^2$ – the existence of such an $\vec{s}'$ is guaranteed by the four squares theorem.

[9] It is of course possible to reduce the public key size of any scheme by hashing it as $\mathsf{pk}' = H(\mathsf{pk})$ for some cryptographic hash function H with the resulting pk' being as small as 32 bytes. This technique is fine for regular signatures, where one can reveal pk as part of the signature; but ring signatures will require a zero-knowledge proof that $\mathsf{pk}' = H(\mathsf{pk})$, which will make the signatures orders of magnitude larger and slower.

	sig. sizes for N: 2^6	2^{12}	2^{21}	asymptotic sig. size	hardness assumption	(user) public key size
Raptor [LAZ19]	81	5161	–	$O(N)$	NTRU	0.9
DualRing-LB [YEL+21]	6	106	–	$O(N)$	MSIS, MLWE	$[2.8, 3.4]$
Falafl [BKP20]	32	35	39	$O(\log N)$	MSIS, MLWE	1.9
MatRiCT [EZS+19]	31	59	148	$O(\log^{1.7} N)$	MSIS, MLWE	$[3.4, 22.7]$
MatRiCT+ [ESZ21]	11	18	40(?)	$O(\log^{1.7} N)$	MSIS, MLWE	3
SMILE [LNS21b]	18	19	22	$O(\log N)$	MSIS, E-MLWE	2
Calamari [BKP20]	8	14	23	$O(\log N)$	CSIDH-512	0.06
This Work	**13**	**14**	**16**	$O(\log N)$	**MSIS, E-MLWE**	**0.13**

Fig. 1: Comparison of the different post-quantum ring signature schemes with approximately 128 bits of security. Sizes from previous constructions are either taken from the corresponding prior work or from the recent survey by Buser et al. [BDE+22]. All the values are given in KB. Here, N is the size of the ring. The signatures sizes for [ESZ21, LNS21b] only approximately correspond to the ring sizes (e.g. 18KB signature size is for the ring of 2^{10} users and not 2^{12}). For DualRing-LB and MatRiCT (and MatRiCT+) the public key size grows in the number of users. For MatRiCT+, the public key size for the ring of size 1024 is provided. Further, we extrapolate the signature size for MatRiCT+ with 2^{21} users from the smaller examples and from MatRiCT. In our construction, we rely on the Extended-MLWE problem introduced in Definition 3.2. Note that this is a different version of the E-MLWE problem compared to the one in [LNS21a] which is used in SMILE [LNS21b] (see Section 3.1 for more details).

proof. In prior ring signatures (e.g. [EZS+19, ESZ21, LNS21b]) the signer had knowledge of $\vec{v}$ and $\vec{s}$ satisfying (1), but for efficiency would only prove knowledge of an $\vec{s}'$ and an additional low-norm polynomial c with $\|\vec{s}'\| \gg \|\vec{s}\|$ satisfying $T\vec{v} = cA\vec{s}' \bmod p$ (where the right-hand side operations are over a polynomial ring $\mathcal{R}_p$.) Being able to prove knowledge of a vector that has the exact norm of $\vec{s}$ and not have an additional multiplication by c allows us to use a much smaller modulus p, which in turn also allows to reduce the number of rows in A. The public key size can, in fact, be essentially as small as the outputs in the hash function SWIFFT [LMPR08].[10]

One can construct group signatures in a somewhat similar manner as ring signatures. A technique employed in [EZS+19, ESZ21] has the public key of each member stored in the matrix T, as in the ring signature, and the secret key of user i are the $\vec{v}$ and $\vec{s}$ from (1). The keys are generated by choosing a small-norm random $\vec{s}_i$ and then putting $A\vec{s}_i = \vec{t}_i$ into the public matrix T. To sign, the group member does the same thing as in the ring signature (with an additional encryption and proof required by the group signature). Our group signature works in the same fashion except that the secret key / public key pairs are generated by first generating $\vec{t}_i = H(i)$, for some cryptographic hash function H, and then using a trapdoor for the matrix A to generate a short $\vec{s}_i$ such that $A\vec{s}_i = \vec{t}_i$. The main advantage of generating the keys in this manner is that the public key size no longer needs to be linear in the number of group members, and can just consist of the matrix A (since everyone can now generate T themselves). The disadvantage is that using a trap-door sampler to generate $\vec{s}_i$ results in $\|\vec{s}_i\|$ being larger. But because our one-out-of many proof system can prove exact norms, the proof size does not increase by too much. Using GPV-type trapdoor sampling [GPV08, DP16] along with an optimized NTRU trapdoor [HHGP+03, DLP14] and the parameters used in the Falcon signature scheme [FHK+20], we give an instantiation (in Appendix E) of a lattice-based group signature scheme with the smallest public key and signature sizes (see Figure 2). The only exception when one would want to use a different scheme is in the case that the group sizes are very large – in that case [LNPS21, LNP22] has an advantage over all others in the table due to the fact that the running time for signature generation and verification are independent of the group size, rather than linear.

[10] If we make p too small, then the signature size will increase because a smaller p requires more "garbage terms" in the zero-knowledge proof to increase soundness. In our parameter settings, we chose a particular compromise between the public key size and signature size, but one could make the public key size even smaller at the expense of a few extra kilobytes in the signature size.

	signature sizes for N: 2^6	2^{10}	2^{21}	asymptotic sig. size	anonymity	master public key size
[LNPS21, LNP22]	90	90	90	$O(1)$	CPA	48
[BDK+21][Lattice]	89	91	96	$O(\log N)$	CCA	$2.9 \cdot (N+1)$
MatRiCT [EZS+19]	34	60	148	$O(\log^{1.7} N)$	CPA	$[3.8, 24] \cdot N$
MatRiCT+ [ESZ21]	12	19	45(?)	$O(\log^{1.7} N)$	CPA	$3 \cdot N$
[BDK+21][Isogeny]	6.6	9.0	15.5	$O(\log N)$	CCA	$0.06 \cdot (N+1)$
This Work	**17**	**18**	**20**	$O(\log N)$	CPA	**3.2**

Fig. 2: Comparison of different post-quantum group signature with approximately 128 bits of security. All the values are given in KB. Here, N denotes the size of the group. We note that the schemes from [LNPS21, LNP22] not only do offer constant size signatures but also enjoy signing and verification complexity independent of the group size N which is not the case for all the other works (including ours). Further, the constructions in [BDK+21, EZS+19, ESZ21] are dynamic which results in the linear public key size in the number of users N. For MatRiCT+, the public key size for the group of size 1024 is provided. Since the signature size for MatRiCT+ was not explicitly provided for group size 2^{21}, we set the value to be three times smaller than for MatRiCT which seems to be the case for smaller examples. Finally, we remark that our scheme can achieve CCA anonymity by following the Naor-Yung paradigm [NY90], i.e. encrypting the same message under two different public keys and adding a NIZK proof that both ciphertext encrypt the same message. We estimate that with this modification our group signature sizes will be around 30KB.

Other Applications. In addition to ring and group signatures, lattice-based one-out-of-many proofs have recently found applications in the constructions of confidential transaction protocols [EZS+19, LNS21b, ESZ21]. These constructions also used other primitives, notably proofs of addition that were used to make sure that the amounts in the transactions match up. As a side contribution, in Appendix D, we also show how to use the new framework of [LNP22] in conjunction with the bimodal rejection sampling technique to construct more efficient proofs of integer addition and multiplication, which improve upon the constructions from [LNS20, ESZ21] that are used in the aforementioned instantiations. We believe that the improved one-out-of-many proof and proof of addition from this paper should noticeably shorten the confidential transaction proof sizes. We leave the integration of these tools as well as the full implementation of the confidential transaction system to future work.

Acknowledgements. We would like to thank the anonymous reviewers for 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 $q = q_1 \cdot \ldots \cdot q_n$ 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. 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. Further, denote $[n] := \{1, \ldots, n\}$.

2.2 Cyclotomic Rings

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.

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

$$X^d + 1 \equiv \left(X^{\frac{d}{2}} - \zeta_i\right)\left(X^{\frac{d}{2}} - \zeta_i^3\right) \pmod{q_i}. \tag{13}$$

In this paper we will be working with polynomials in $\mathcal{R}_p$ which are stable under the σ_{-1} automorphism. We recall the result by Lyubashevsky et al. [LNP22] which says that for specific primes p, if $c \in \mathcal{R}_p$ satisfies $\sigma_{-1}(c) = c$ and c is non-zero then c is invertible over $\mathcal{R}_p$.

Lemma 2.1 ([LNP22]). *Let $p \equiv 5 \pmod 8$ be a prime. Then all non-zero $c \in \mathcal{R}_p$ satisfying $\sigma_{-1}(c) = c$ are invertible.*

In this paper, we will only be interested in the $\sigma := \sigma_{-1}$ automorphism. The main reason is the following observation.

Lemma 2.2 ([LNP22]). *Let $\mathbf{u}, \mathbf{v} \in \mathcal{R}_q^k$. Then, the constant coefficient of $\sigma(\mathbf{u})^T \mathbf{v}$ is equal to $\langle \mathbf{u}, \mathbf{v} \rangle$.*

Thus, one reduces inner product arguments $\langle \mathbf{u}, \mathbf{v} \rangle = a$ to proving that $\sigma(\mathbf{u})^T \mathbf{v} - a \in \mathcal{R}_q$ has a vanishing constant coefficient.

We introduce the following notation:

$$\langle x \rangle_\sigma := (x, \sigma(x)) \in \mathcal{R}_q^2 \text{ for } x \in \mathcal{R}_q.$$

Similarly, for a vector $\mathbf{x} = (x_1, \ldots, x_n)$, define $\langle \mathbf{x} \rangle_\sigma = (\langle x_1 \rangle_\sigma, \ldots, \langle x_n \rangle_\sigma) \in \mathcal{R}_q^{2n}$. We will use the following simple properties.

Lemma 2.3. *For any $\mathbf{x}, \mathbf{y} \in \mathcal{R}_q^n$ and any $c \in \mathcal{R}_q$ such that $\sigma(c) = c$:*

$$\langle \mathbf{x} \parallel \mathbf{y} \rangle_\sigma = \langle \mathbf{x} \rangle_\sigma \parallel \langle \mathbf{y} \rangle_\sigma \quad \text{and} \quad \langle \mathbf{x} + c\mathbf{y} \rangle_\sigma = \langle \mathbf{x} \rangle_\sigma + c \langle \mathbf{y} \rangle_\sigma.$$

Next, we recall the definition of the discrete Gaussian distribution over $\mathcal{R}$.

Definition 2.4. *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*

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

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

We will use the standard tail bound result from [Ban93, Lemma 1.5(i)].

Lemma 2.5. *Let $\mathbf{z} \leftarrow D_\mathfrak{s}^m$. Then* $\Pr\left[\|\mathbf{z}\| > t \cdot \mathfrak{s}\sqrt{md}\right] < \left(te^{\frac{1-t^2}{2}}\right)^{md}$.

2.3 Module-SIS and Module-LWE Problems

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

Definition 2.6 ($\mathsf{MSIS}_{\kappa,m,B}$). *Given* $\mathbf{A} \leftarrow \mathcal{R}_q^{\kappa \times m}$, *the* Module-SIS *problem with parameters* $\kappa, m > 0$ *and* $0 < B < q$ *asks to find* $\mathbf{z} \in \mathcal{R}_q^m$ *such that* $\mathbf{Az} = \mathbf{0}$ *over* $\mathcal{R}_q$ *and* $0 < \|\mathbf{z}\| \leqslant B$. *An algorithm* $\mathcal{A}$ *is said to have advantage* ϵ *in solving* $\mathsf{MSIS}_{\kappa,m,B}$ *if*

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

Definition 2.7 ($\mathsf{MLWE}_{m,\lambda,\chi}$). *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*

$$\begin{aligned}\Big|\Pr\left[b = 1 \,\middle|\, \mathbf{A} \leftarrow \mathcal{R}_q^{m \times \lambda};\ \boldsymbol{s} \leftarrow \chi^\lambda;\ \boldsymbol{e} \leftarrow \chi^m;\ b \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{A}\boldsymbol{s} + \boldsymbol{e})\right] \\ - \Pr\left[b = 1 \,\middle|\, \mathbf{A} \leftarrow \mathcal{R}_q^{m \times \lambda};\ \boldsymbol{b} \leftarrow \mathcal{R}_q^m;\ b \leftarrow \mathcal{A}(\mathbf{A}, \boldsymbol{b})\right]\Big| \geqslant \epsilon.\end{aligned} \tag{14}$$

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*. We summarise the two most common techniques for rejection sampling described in [Lyu12, DDLL13].

Lemma 2.8 (Rejection Sampling [Lyu12, DDLL13]). *Let* $V \subseteq \mathcal{R}^\ell$ *be a set of polynomials with norm at most* T *and* $\rho\colon V \to [0,1]$ *be a probability distribution. Fix the standard deviation* $\mathfrak{s} = \gamma T$. *Then, the following statements hold.*

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

$\underline{\mathrm{Rej}_1(\vec{z}, \vec{v}, \mathfrak{s})}$	$\underline{\mathrm{Rej}_2(\vec{z}, \vec{v}, \mathfrak{s})}$
01 $u \leftarrow [0,1)$	01 $u \leftarrow [0,1)$
02 If $u > \frac{1}{M} \cdot \exp\left(\frac{-2\langle \vec{z}, \vec{v}\rangle + \|\vec{v}\|^2}{2\mathfrak{s}^2}\right)$	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*)	03 return 1 (i.e. *reject*)
04 Else	04 Else
05 return 0 (i.e. *accept*)	05 return 0 (i.e. *accept*)

Fig. 3: Two rejection sampling algorithms: the one used generally in previous works [Lyu12] (left) and the bimodal Gaussian one [DDLL13] (right).

2.5 Challenge Space

We recall the specific challenge space used in [LNP22]. Namely, we fix $\eta > 0$ and a power-of-two k and set the challenge space $\mathcal{C}$ as:

$$\mathcal{C} := \left\{ c \in S_\kappa : \sigma_{-1}(c) = c \wedge \sqrt[2k]{\|c^{2k}\|_1} \leqslant \eta \right\}. \tag{15}$$

Roughly speaking, the first condition, i.e. $\sigma_{-1}(c) = c$, is needed to prove quadratic equations in the committed messages which might additionally involve automorphisms, e.g. $m_1 m_2 = \sigma_{-1}(m_3)$ where m_1, m_2, m_3 are the secret messages. On the other hand, the second condition allows us to use [LNP22, Lemma 2.15] and deduce that if $\|\mathbf{r}\| \leqslant \alpha$ and $c \in \mathcal{C}$ then $\|c\mathbf{r}\| \leqslant \eta\alpha$.

Further, 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 $\mathcal{C}$. We will choose the constant η such that (experimentally) the probability for $c \leftarrow S_\kappa$ to satisfy $\sqrt[2k]{\|c^{2k}\|_1} \leqslant \eta$ is at least 99%.

For security of our protocols, we need the invertibility property of the challenge space $\mathcal{C}$, i.e. the difference of any two distinct elements of $\mathcal{C}$ is invertible over $\mathcal{R}_q$. To this end, we apply Lemma 2.1 and thus we only need the condition $\kappa < q_1/2$. Secondly, to achieve negligible soundness error, we will need $|\mathcal{C}|$ to be exponentially large. In Table 4 we propose example parameters to instantiate the challenge space $\mathcal{C}$.

d	κ	η	$\lvert\mathcal{C}\rvert$
64	8	140	2^{129}
128	2	59	2^{147}

Fig. 4: Example parameters to instantiate the challenge space $\mathcal{C} := \{c \in S_\kappa : \sigma_{-1}(c) = c \wedge \sqrt[2k]{\|c^{2k}\|_1} \leqslant \eta\}$ for a modulus q such that its smallest prime divisor q_1 is greater than 16. In our examples we picked $k = 32$.

2.6 ABDLOP Commitment

We recall the ABDLOP commitment scheme defined in [LNP22], which is a generalisation of the Ajtai [Ajt96] and BDLOP [BDL+18] constructions. Concretely, to commit to a message vector $\mathbf{s}_1 \in \mathcal{R}_q^{m_1}$ with small coefficients as well as a "full-fledged" polynomial vector $\mathbf{m} \in \mathcal{R}_q^{\ell}$, we sample a randomness vector $\mathbf{s}_2 \leftarrow \chi^{m_2}$, where χ is a probability distribution over $\mathcal{R}_q$, and compute:

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

where $\mathbf{A}_1 \leftarrow \mathcal{R}_q^{n \times m_1}, \mathbf{A}_2 \leftarrow \mathcal{R}_q^{n \times m_2}, \mathbf{B} \leftarrow \mathcal{R}_q^{\ell \times m_2}$. We observe that when $\ell = 0$ (resp. $m_1 = 0$) then this construction ends up being the Ajtai (resp. BDLOP) commitment scheme. In particular, the commitment size does not depend on the length m_1 of $\mathbf{s}_1$ (but it does on ℓ). Hence, our strategy is to commit to long vectors with small coefficients in the "Ajtai" part $\mathbf{s}_1$ and commit to a few *garbage* polynomials used for the proofs in the "BDLOP" part $\mathbf{m}$.

An opening of the commitment is a triple $(\mathbf{s}_1, \mathbf{m}, \mathbf{s}_2)$[11]. As usual in lattice-based cryptography, we also consider relaxed openings of a commitment which are defined as follows.

Definition 2.9. *A relaxed opening of the* ABDLOP *commitment* $(\mathbf{t}_A, \mathbf{t}_B)$ *is a tuple* $(\mathbf{s}_1, \mathbf{m}, \mathbf{s}_2, c)$ *which satisfies:*

$$\begin{aligned} &\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A \\ &\mathbf{A}_2\mathbf{s}_2 + \mathbf{m} = \mathbf{t}_B \\ &c \in \bar{\mathcal{C}} \text{ as defined in Section 2.5} \\ &\|c\mathbf{s}_1\| \leqslant B_1 \quad \text{and} \quad \|c\mathbf{s}_2\| \leqslant B_2. \end{aligned}$$

[11] Message $\mathbf{m}$ does not need to be included in the opening since it can be deterministically computed from $\mathbf{t}_B$ and $\mathbf{s}_2$.

As shown in [LNP22, Lemma 3.1], the ABDLOP commitment is binding with respect to relaxed openings under the Module-SIS assumption.

Lemma 2.10 ([LNP22]). *The* ABDLOP *commitment is computationally binding with respect to relaxed openings under the* $\mathsf{MSIS}_{n,m_1+m_2,B}$ *assumption where* $B := 4\eta\sqrt{B_1^2 + B_2^2}$.

The hiding property of the ABDLOP commitment scheme follows from the fact that under the Module-LWE assumption $\begin{bmatrix}\mathbf{A}_2\\ \mathbf{B}\end{bmatrix}\mathbf{s}_2$ looks pseudorandom.

Lemma 2.11. *The* ABDLOP *commitment is computationally hiding under the* $\mathsf{MLWE}_{n+\ell,m_2-n-\ell,\chi}$ *assumption.*

2.7 Framework for Proving Lattice Statements

The recently proposed framework by Lyubashevsky et al. [LNP22] can be used to prove various relations in the committed messages. Concretely, one can prove knowledge of the secret messages $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$ which satisfy all the following conditions:

1. *Quadratic relations over* $\mathcal{R}_q$ *with automorphisms.* For $i \in [N]$ and public triples $(\mathbf{R}_{i,2}, \mathbf{r}_{i,1}, r_{i,0}) \in \mathcal{R}_q^{2(m_1+\ell)\times 2(m_1+\ell)} \times \mathcal{R}_q^{2(m_1+\ell)} \times \mathcal{R}_q$, we have:
$$\langle \mathbf{s}_1 \parallel \mathbf{m}\rangle_\sigma^T \mathbf{R}_{i,2} \langle \mathbf{s}_1 \parallel \mathbf{m}\rangle_\sigma + \mathbf{r}_{i,1}^T \langle \mathbf{s}_1 \parallel \mathbf{m}\rangle_\sigma + r_{i,0} = 0. \tag{16}$$
2. *Quadratic relations over* $\mathbb{Z}_q$ *with automorphisms.* For $i \in [M]$ and public triples $(\mathbf{R}'_{i,2}, \mathbf{r}'_{i,1}, r'_{i,0}) \in \mathcal{R}_q^{2(m_1+\ell)\times 2(m_1+\ell)} \times \mathcal{R}_q^{2(m_1+\ell)} \times \mathcal{R}_q$:
$$\text{const. coeff. of } \langle \mathbf{s}_1 \parallel \mathbf{m}\rangle_\sigma^T \mathbf{R}'_{i,2} \langle \mathbf{s}_1 \parallel \mathbf{m}\rangle_\sigma + \mathbf{r}'^T_{i,1} \langle \mathbf{s}_1 \parallel \mathbf{m}\rangle_\sigma + r'_{i,0} \text{ equals } 0. \tag{17}$$
3. *Shortness in the infinity norm.* For public $\mathbf{P}_s \in \mathcal{R}_q^{n_{\mathsf{bin}}\times m_1}, \mathbf{P}_m \in \mathcal{R}_q^{n_{\mathsf{bin}}\times \ell}$ and $\mathbf{f} \in \mathcal{R}_q^{n_{\mathsf{bin}}}$, the following polynomial vector has binary coefficients
$$\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f} \in \{0,1\}^{n_{\mathsf{bin}}\cdot d}. \tag{18}$$
4. *Shortness in the Euclidean norm.* For $i \in [Z]$, public bound $\mathcal{B}_i < \sqrt{q}$ and $\mathbf{E}_s^{(i)} \in \mathcal{R}_q^{n_i\times m_1}, \mathbf{E}_m^{(i)} \in \mathcal{R}_q^{n_i\times \ell}$ and $\mathbf{v}^{(i)} \in \mathcal{R}_q^{n_i}$, we have:
$$\|\mathbf{E}_s^{(i)}\mathbf{s}_1 + \mathbf{E}_m^{(i)}\mathbf{m} + \mathbf{v}^{(i)}\| \leqslant \mathcal{B}_i.$$
This is equivalent to additionally proving knowledge of the binary polynomial $\vartheta_i \in \mathcal{R}$ such that
$$\langle \mathsf{pow}(\mathcal{B}_i^2), \vartheta_i\rangle = \mathcal{B}_i^2 - \left\|\mathbf{E}_s^{(i)}\mathbf{s}_1 + \mathbf{E}_m^{(i)}\mathbf{m} + \mathbf{v}^{(i)}\right\|^2 \text{ over } \mathbb{Z} \tag{19}$$
where $\mathsf{pow}(n) := \sum_{i=0}^{\lfloor \log n\rfloor}(2X)^i \in \mathcal{R}$ for $n \leqslant 2^{d-1}$.
5. *Approximate Shortness.* For a public bound $\mathcal{B}'$ and $\mathbf{D}_s \in \mathcal{R}_q^{n'\times m_1}, \mathbf{D}_m \in \mathcal{R}_q^{n'\times \ell}$ and $\mathbf{u} \in \mathcal{R}_q^{n'}$, we have:
$$\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\| \leqslant \mathcal{B}'. \tag{20}$$
However, we are fine with convincing the verifier that
$$\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\|_\infty \leqslant \psi \cdot \mathcal{B}' \tag{21}$$
where $\psi > 1$ is a public approximation factor.

The main strength of the aforementioned framework is that, unlike the prior works [ALS20, ENS20, LNS21b], it does not rely on the NTT-packing technique. This comes with two immediate benefits: (i) one can apply a compressing commitment scheme ABDLOP to commit to long messages of small norm (i.e. $\mathbf{s}_1$), and (ii) the protocol is one-shot and thus no expensive part of the protocol needs to be repeated for soundness amplification. In practice, [LNP22] achieves more than a factor of two improvement in proof size compared to the NTT-packing protocols for basic statements. We refer to [LNP22] for more details on the protocol.

In this paper, we will use the methodology from [LNP22, Section 6.1] to calculate the exact proof sizes.

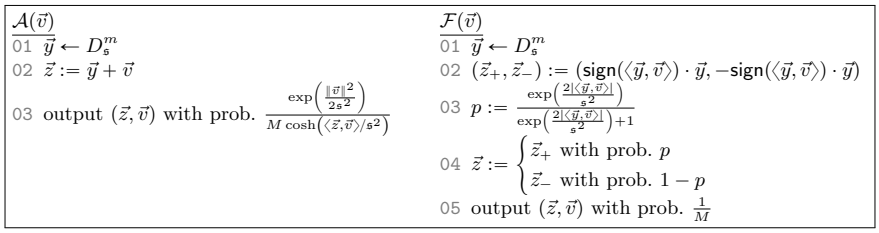

Fig. 5: Algorithms $\mathcal{A}$ and $\mathcal{F}$ for Lemma 3.1. We define $\mathsf{sign}(x) = 1$ if $x \geqslant 0$ and -1 otherwise.

3 Shorter Proofs via Bimodal Gaussians

In order to provide zero-knowledge (or more precisely, simulatability) for proving relations in the ABDLOP committed messages $(\mathbf{s}_1, \mathbf{m})$ under the randomness $\mathbf{s}_2$, one applies the rejection sampling technique. In the original protocols presented in [LNP22], the standard rejection sampling [Lyu12] is used for $\mathbf{s}_1$ and the more recent one [LNS21a] for $\mathbf{s}_2$. In this section we describe how one can apply bimodal Gaussian rejection sampling [DDLL13] on both the message and randomness which significantly reduces the standard deviations, and consequently the proof size, compared to [LNP22].

3.1 Bimodal Gaussian Rejection Sampling on the Randomness

In our constructions, we apply a rejection sampling procedure to mask a secret vector $\vec{v}$ by first sampling $\vec{y}$ from a discrete Gaussian with standard deviation $\mathfrak{s}$, and then computing $\vec{z} := \vec{v} + \vec{y}$. By Lemma 2.8, if we additionally run $\mathsf{Rej}_1(\vec{z}, \vec{v}, \mathfrak{s})$, then the distribution of $\vec{z}$ is indistinguishable to the one where we simply sample $\vec{z}$ from a discrete Gaussian and output $\vec{z}$ with certain (known) probability. Here, it is important that one could generate $\vec{z}$ without having any information on $\vec{v}$.

Now, suppose that instead of Rej_1, we run Rej_2 which is used for bimodal Gaussian rejection sampling [DDLL13]. It is now a natural question to ask whether there is a way to simulate the $\vec{z}$ by having as little information on $\vec{v}$ as possible. We answer this question positively and show that this distribution is simulatable given only the inner product $\langle \vec{z}, \vec{v}\rangle$ of $\vec{z}$ and $\vec{v}$. We summarise our observation with the following lemma.

Lemma 3.1. *Let $\vec{v} \in \mathbb{Z}^m$ be a vector of norm T. Fix $\mathfrak{s} \geqslant \gamma T$ and $M = \exp\left(\frac{1}{2\gamma^2}\right)$. Then the distributions of the outputs of $\mathcal{A}(\vec{v})$ and $\mathcal{F}(\vec{v})$ defined in Figure 5 are identical. Moreover, the probability that $\mathcal{A}$ outputs something is exactly $1/M$.*

Proof. Fix $\vec{v} \in V$ and $\vec{z} \in \mathbb{Z}^m$ and let

$$p := \frac{\exp\left(\frac{2\langle \vec{z}, \vec{v}\rangle}{\mathfrak{s}^2}\right)}{\exp\left(\frac{2\langle \vec{z}, \vec{v}\rangle}{\mathfrak{s}^2}\right)+1}.$$

By definition of $\mathcal{A}$, $\mathcal{A}(\vec{v}, \vec{z})$ is equal to

$$D_{\mathfrak{s}}^m(\vec{z}-\vec{v})\cdot\frac{\exp\left(\frac{\|\vec{v}\|^2}{2\mathfrak{s}^2}\right)}{M\cosh\left(\frac{\langle \vec{z}, \vec{v}\rangle}{\mathfrak{s}^2}\right)} = D_{\mathfrak{s}}^m(\vec{z})\cdot\frac{2\exp\left(\frac{2\langle \vec{z}, \vec{v}\rangle}{\mathfrak{s}^2}\right)}{M\left(\exp\left(\frac{2\langle \vec{z}, \vec{v}\rangle}{\mathfrak{s}^2}\right)+1\right)} = D_{\mathfrak{s}}^m(\vec{z})\cdot\frac{2p}{M}$$

Now, we focus on $\mathcal{F}(\vec{v})$. We see that by construction, $\langle \vec{z}_+, \vec{v}\rangle \geqslant 0$ and $\langle \vec{z}_-, \vec{v}\rangle \leqslant 0$. Let us consider three separate cases. First, suppose $\vec{z}$ satisfies $\langle \vec{z}, \vec{v}\rangle > 0$. Informally, we want to compute the probability that $\vec{y} = \pm\vec{z}$ and $\mathcal{F}$ picks $\vec{z}_+$. Then,

$$\mathcal{F}(\vec{v}, \vec{z}) = 2D_{\mathfrak{s}}^m(\vec{z})\cdot\frac{\exp\left(\frac{2\langle \vec{z}, \vec{v}\rangle}{\mathfrak{s}^2}\right)}{\exp\left(\frac{2\langle \vec{z}, \vec{v}\rangle}{\mathfrak{s}^2}\right)+1}\cdot\frac{1}{M} = D_{\mathfrak{s}}^m(\vec{z})\cdot\frac{2p}{M}.$$

Further, suppose $\langle \vec{z}, \vec{v} \rangle < 0$. Informally, we compute the probability that $\vec{y} = \pm\vec{z}$ and $\mathcal{F}$ picks $\vec{z}_-$. Then,

$$\mathcal{F}(\vec{v}, \vec{z}) = 2D^m_{\mathfrak{s}}(\vec{z}) \cdot \frac{1}{\exp\left(\frac{-2\langle \vec{z}, \vec{v} \rangle}{\mathfrak{s}^2}\right) + 1} \cdot \frac{1}{M} = D^m_{\mathfrak{s}}(\vec{z}) \cdot \frac{2p}{M}.$$

Finally, assume $\langle \vec{z}, \vec{v} \rangle = 0$ and thus $p = 1/2$. Then, $\mathcal{F}(\vec{v}, \vec{z})$ is simply the probability that ($\vec{y} = \vec{z} \wedge$ $\mathcal{F}$ outputs $\vec{z}_+$) or ($\vec{y} = -\vec{z} \wedge \mathcal{F}$ outputs $\vec{z}_-$). Hence,

$$\mathcal{F}(\vec{v}, \vec{z}) = D^m_{\mathfrak{s}}(\vec{z}) \cdot \frac{1}{2M} + D^m_{\mathfrak{s}}(-\vec{z}) \cdot \frac{1}{2M} = D^m_{\mathfrak{s}}(\vec{z}) \cdot \frac{1}{M} = D^m_{\mathfrak{s}}(\vec{z}) \cdot \frac{2p}{M}.$$

Therefore, we proved that for every $\vec{z}$, $\mathcal{A}(\vec{v}, \vec{z}) = \mathcal{F}(\vec{v}, \vec{z})$.

Finally, the second part of the statement follows from a simple observation that $\mathcal{F}$ outputs something with probability exactly $1/M$. □

Extended-MLWE Revisited. We observe that the only information about $\vec{v}$ needed in order to run the simulator $\mathcal{F}$ in the security proof is the value of $\langle \vec{y}, \vec{v} \rangle$. Hence, we reduce the simulatability property of our protocols to the hardness of the so-called Extended-MLWE. Here, as usual, an adversary needs to distinguish between the tuples $(\mathbf{B}, \mathbf{Bs})$ and $(\mathbf{B}, \mathbf{u})$, where $\mathbf{u}$ is a uniformly random vector but this time it is also given a "hint" of the form $(c, \mathbf{y}, \langle c\mathbf{s}, \mathbf{y} \rangle)$ where c and $\mathbf{y}$ are sampled from some known distributions. For simplicity, we will describe the problem in a "knapsack" form.

Definition 3.2 (Extended-MLWE). *The* Extended-MLWE *problem with parameters n, m and distribution χ, ξ_c, ξ_y over $\mathcal{R}$ asks the adversary $\mathcal{A}$ to distinguish between the two cases: 1) $(\mathbf{B}, \mathbf{Bs}, c, \mathbf{y}, \langle c\mathbf{s}, \mathbf{y} \rangle)$ and 2) $(\mathbf{B}, \mathbf{u}, c, \mathbf{y}, \langle c\mathbf{s}, \mathbf{y} \rangle)$ for $\mathbf{B} \leftarrow \mathcal{R}_q^{m \times (n+m)}$, a secret vector $\mathbf{s} \leftarrow \chi^{n+m}$, uniformly random vector $\mathbf{u} \in \mathcal{R}_q^m$ and $(c, \mathbf{y}) \leftarrow \xi_c \times \xi_y^{n+m}$. Then, $\mathcal{A}$ is said to have advantage ϵ in solving Extended-*$\mathsf{MLWE}_{n,m,\chi,\xi_c,\xi_y}$ *if*

$$\left| \Pr\left[b = 1 \,\middle|\, \mathbf{B} \leftarrow \mathcal{R}_q^{m \times (n+m)};\ \mathbf{s} \leftarrow \chi^{n+m};\ (c, \mathbf{y}) \leftarrow \xi_c \times \xi_y^{n+m};\ b \leftarrow \mathcal{A}(\mathbf{B}, \mathbf{Bs}, c, \mathbf{y}, \langle c\mathbf{s}, \mathbf{y} \rangle) \right] \right.$$
$$\left. - \Pr\left[b = 1 \,\middle|\, \begin{array}{c} \mathbf{B} \leftarrow \mathcal{R}_q^{m \times (n+m)};\ \mathbf{s} \leftarrow \chi^{n+m};\ (c, \mathbf{y}) \leftarrow \xi_c \times \xi_y^{n+m};\ \mathbf{u} \leftarrow \mathcal{R}_q^m; \\ b \leftarrow \mathcal{A}(\mathbf{B}, \mathbf{u}, c, \mathbf{y}, \langle c\mathbf{s}, \mathbf{y} \rangle) \end{array} \right] \right| \geqslant \epsilon.$$

We say that Extended-$\mathsf{MLWE}_{n,m,\chi,\xi_c,\xi_y}$ is hard if for all PPT adversaries $\mathcal{A}$, the advantage in solving Extended-$\mathsf{MLWE}_{n,m,\chi,\xi_c,\xi_y}$ is negligible.

We note that the (Module-)LWE problem with various side information has already been discussed in prior work e.g. [DGK+10, AP12, DDGR20]. As far as we are aware, this new variant of MLWE is the closest to the Extended Module-LWE problems defined by Lyubashevsky et al. [LNS21a], Alperin-Sheriff and Apon [ASA16], Alperin-Sheriff and Peikert [AP12] and Boudgoust et al. [BJRW21].

We observe that [ASA16] describes a similar problem with the two differences: (i) there is no c involved (assume that $c = 1$) and (ii) the hint is an arbitrary $\mathbb{Q}$-linear function on the "error" part $\mathbf{e}$ of the secret $\mathbf{s}$ (in particular it could be $\langle \mathbf{e}, \mathbf{y} \rangle \in \mathbb{Z}$ where $\mathbf{y} \leftarrow \xi_y^m$). Alperin-Sheriff and Apon show that their Extended-MLWE problem can be reduced to plain MLWE if the errors come from a discrete Gaussian with a large enough standard deviation. The proof strategy was later extended by Boudgoust et al. [BJRW21] who define another Extended-MLWE problem. This time, however, the hint becomes a whole polynomial $\langle \mathbf{e}, \mathbf{y} \rangle \in \mathcal{R}$. Finally, the only difference between our problem and the one in [LNS21a] is that the adversary is given the whole inner product $\langle c\mathbf{s}, \mathbf{y} \rangle$ instead of its sign.

If we consider our Extended-MLWE without any polynomial ring structure, then the problem becomes almost identical to the one introduced by Alperin-Sheriff and Peikert [AP12] (if we again assume $c = 1$). The authors additionally show that it is possible to reduce such a problem to plain LWE with the reduction loss $O(|\langle \vec{s}, \vec{y} \rangle|)$.

Applications. As an example, we show how to use the new rejection sampling strategy in the protocol for proving linear equations in the committed messages [LNP22][Section 3], however this approach can also be applied in all the protocols from [LNP22]. Let $(\mathbf{t}_A, \mathbf{t}_B)$ be the ABDLOP commitment to the message pair $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{\ell}$ under randomness $\mathbf{s}_2$, i.e.

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \cdot \mathbf{s}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} \cdot \mathbf{s}_2 + \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix}. \tag{22}$$

Suppose the prover wants to prove knowledge of the message $(\mathbf{s}_1, \mathbf{m})$ such that

$$\mathbf{R}_1\mathbf{s}_1 + \mathbf{R}_m\mathbf{m} = \mathbf{u}$$

where $\mathbf{R}_1 \in \mathcal{R}_q^{N \times m_1}, \mathbf{R}_m \in \mathcal{R}_q^{N \times \ell}$ and $\mathbf{u} \in \mathcal{R}_q^N$.

We present the commit-and-prove protocol in Figure 6 for proving linear relations. The only difference between this protocol and [LNP22, Fig. 4] is that for $\mathbf{z}_2$ we apply the new rejection sampling algorithm described above.

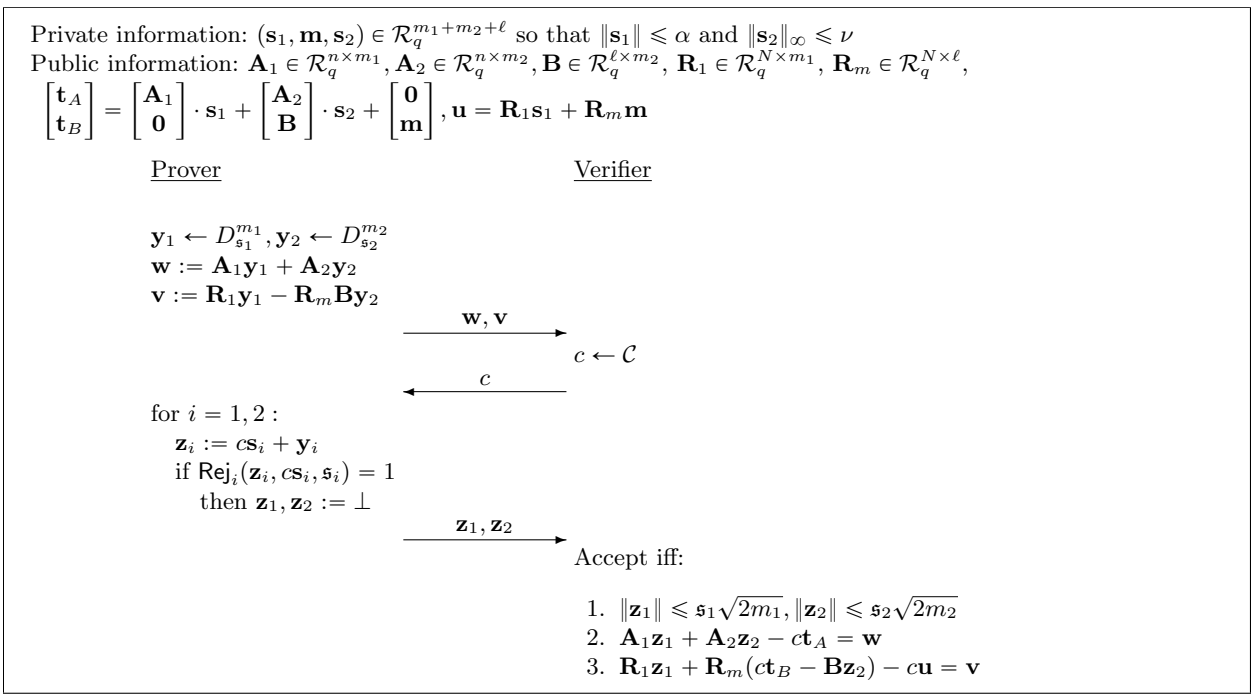

Fig. 6: Proof of knowledge $\Pi^{(1)}((\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}), (f_1, f_2, \ldots, f_N))$ of $(\mathbf{s}_1, \mathbf{s}_2, \bar{c}) \in \mathcal{R}_q^{m_1} \times \mathcal{R}_q^{m_2} \times \bar{\mathcal{C}}$ satisfying (i) $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A$, $\mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t}_B$ (ii) $\|\mathbf{s}_i\bar{c}\| \leqslant 2\mathfrak{s}_i\sqrt{2m_i d}$ for $i = 1, 2$ and (iii) $\mathbf{R}_1\mathbf{s}_1 + \mathbf{R}_m\mathbf{m} = \mathbf{u}$. Functions Rej_i are defined in Fig. 3.

3.2 Bimodal Gaussian Rejection Sampling on the Message

This subsection focuses on applying bimodal Gaussian rejection sampling on the message vector $\mathbf{s}_1$. First of all, we cannot apply Lemma 3.1 since it would potentially leak certain information about the message $\mathbf{s}_1$ which, unlike $\mathbf{s}_2$, is not freshly sampled every time a new proof is generated. Instead, we follow the original methodology from [DDLL13].

Concretely, let us focus on the protocol in Fig. 6. If one were to naively apply bimodal rejection sampling on $c\mathbf{s}_1$ then the masked opening of $c\mathbf{s}_1$ would become:

$$\mathbf{z}_1 := \mathbf{y}_1 + bc\mathbf{s}_1 \text{ where } b \leftarrow \{-1, 1\}.$$

As before, we set $\mathbf{z}_2 := \mathbf{y}_2 + c\mathbf{s}_2$. Hence, if we keep $\mathbf{w} := \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2$ then by construction:

$$\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 = \mathbf{w} + c\left(\mathbf{A}_1 b\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2\right).$$

Note that $\mathbf{A}_1 b\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2$ is a top part of the ABDLOP commitment to $(b\mathbf{s}_1, \mathbf{m})$ under randomness $\mathbf{s}_2$. Thus, it is a natural approach to simply commit to $(b\mathbf{s}_1, \mathbf{m})$ and prove the quadratic equation. However, this comes with a big obstacle, i.e. we still need to prove the underlying relation in $\mathbf{s}_1, \mathbf{m}$ even though we committed to $b\mathbf{s}_1$ and $\mathbf{m}$. It might cause a problem even in the simple case of linear relations. Indeed, initially we want to prove that $\mathbf{R}_1\mathbf{s}_1 + \mathbf{R}_m\mathbf{m} = \mathbf{u}$. Since we committed to $b\mathbf{s}_1$ and not $\mathbf{s}_1$, it makes sense to try and prove the equivalent statement:

$$\mathbf{R}_1(b\mathbf{s}_1) + \mathbf{R}_m(b\mathbf{m}) = b\mathbf{u}. \tag{23}$$

This suggests that we should also commit to $b\mathbf{m}$ and not $\mathbf{m}$. However, it does not solve the issue completely since vector $\mathbf{u}$ is still multiplied by a (secret) sign b. Hence, the intuitive solution would be to also commit to b in the ABDLOP commitment, prove $b \in \{-1, 1\}$ and the linear relation (23) in $b\mathbf{s}_1, b\mathbf{m}$ and b. Therefore, the cost of such an approach is at least committing to an extra polynomial.

We show that for certain types of statements we can circumvent committing to b and still apply bimodal Gaussian rejection sampling. Namely, we focus on *sign-invariant* relations.

Definition 3.3. *Let $R \subseteq \{0,1\}^* \times \mathcal{R}^{m_1+\ell}$ be a binary relation. We say that R is sign-invariant if for every pair $(u, \mathbf{w})$ we have: $R(u, \mathbf{w}) = 1 \iff R(u, -\mathbf{w}) = 1$.*

Suppose we want to prove knowledge of $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$ such that $(u, (\mathbf{s}_1, \mathbf{m})) \in R$ where R is a sign-invariant relation. Then, we can sample a fresh sign $b \leftarrow \{-1, 1\}$ and commit to $(b\mathbf{s}_1, b\mathbf{m})$ using the ABDLOP commitment. Further, we simply prove that $R(u, (b\mathbf{s}_1, b\mathbf{m})) = 1$ which implies that $R(u, (\mathbf{s}_1, \mathbf{m})) = 1$.

Concrete instantiation. We demonstrate our intuition with the following example. Namely, we want to prove knowledge of $(\mathbf{s}_1, \mathbf{m})$ which satisfies:

$$\sigma(\mathbf{s}_1)^T\mathbf{s}_1 + \sigma(\mathbf{m})^T\mathbf{m} = 0.$$

Clearly, $(b\mathbf{s}_1, b\mathbf{m})$ satisfies the relation above for $b \in \{-1, 1\}$. As described before, we first sample a sign $b \leftarrow \{-1, 1\}$, randomness vector $\mathbf{s}_2 \leftarrow \chi$ and compute

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \cdot b\mathbf{s}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} \cdot \mathbf{s}_2 + \begin{bmatrix} \mathbf{0} \\ b\mathbf{m} \end{bmatrix}. \tag{24}$$

Then, we simply follow the multiplicative proof from [LNP22, Section 4] to prove that

$$\sigma(b\mathbf{s}_1)^T(b\mathbf{s}_1) + \sigma(b\mathbf{m})^T(b\mathbf{m}) = 0.$$

Concretely, consider the *masked opening* $\mathbf{z}_1 := \mathbf{y}_1 + bc\mathbf{s}_1$ of $\mathbf{s}_1$. Note that

$$\sigma(\mathbf{z}_1)^T\mathbf{z}_1 = c^2\sigma(b\mathbf{s}_1)^T(b\mathbf{s}_1) + c\left(\sigma(\mathbf{y}_1)^T(b\mathbf{s}) + \mathbf{y}_1^T\sigma(b\mathbf{s})\right) + \sigma(\mathbf{y}_1)^T\mathbf{y}_1$$

and hence the coefficient corresponding to the quadratic term c^2 is what we are interested in. Here, we used the property of the challenge space $\mathcal{C}$ that $c = \sigma(c)$ for $c \in \mathcal{C}$. We cannot do the same argument with $b\mathbf{m}$ since no masked opening of $b\mathbf{m}$ was sent. However, we observe that the verifier can compute $\mathbf{t}_B - \mathbf{B}\mathbf{z}_2 = -\mathbf{B}\mathbf{y}_2 + c(b\mathbf{m})$ which is of the similar form as the masked opening of $b\mathbf{s}_1$. Then

$$\begin{aligned}\sigma(\mathbf{t}_B - \mathbf{B}\mathbf{z}_2)^T&(\mathbf{t}_B - \mathbf{B}\mathbf{z}_2) \\ &= c^2(b\mathbf{m})^T(b\mathbf{m}) - c\left(\sigma(\mathbf{B}\mathbf{y}_2)^T(b\mathbf{m}) + (\mathbf{B}\mathbf{y}_2)^T\sigma(b\mathbf{m})\right) + \sigma(\mathbf{B}\mathbf{y}_2)^T\mathbf{B}\mathbf{y}_2.\end{aligned}$$

Therefore, we want to prove that the term in front of c^2 in the following expression disappears, i.e

$$\sigma(\mathbf{z}_1)^T\mathbf{z}_1 + \sigma(\mathbf{t}_B - \mathbf{B}\mathbf{z}_2)^T(\mathbf{t}_B - \mathbf{B}\mathbf{z}_2) = cg_1 + g_0$$

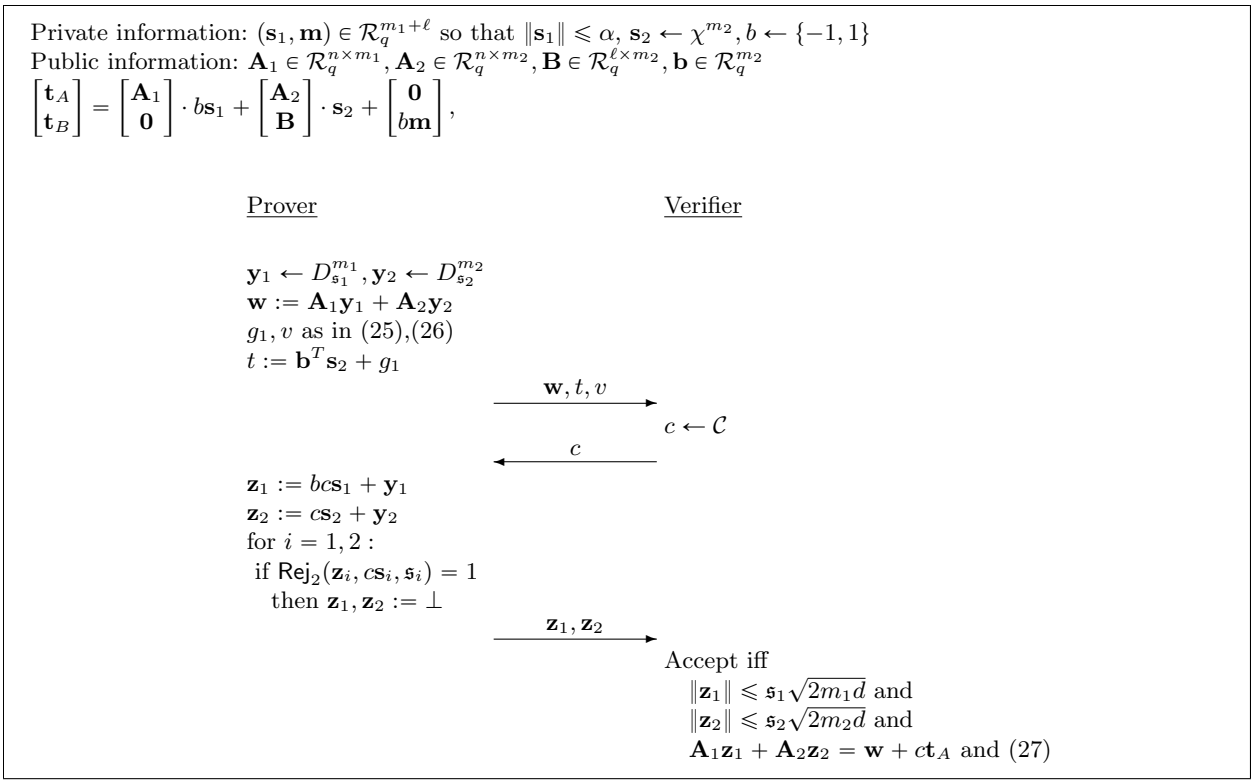

Fig. 7: Commit-and-prove protocol $\Pi_{\mathsf{quad}}(\mathbf{s}_2, \mathbf{s}_1, \mathbf{m})$ for messages $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$, randomness $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$ and $\bar{c} \in \bar{\mathcal{C}}$ which satisfy: $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A$, $\mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t}_B$ (ii) $\|\mathbf{s}_i\bar{c}\| \leqslant 2\mathfrak{s}_i\sqrt{2m_i d}$ for $i = 1, 2$ and (iii) $\sigma(\mathbf{s}_1)^T\mathbf{s}_1 + \sigma(\mathbf{m})^T\mathbf{m} = 0$.

where

$$\begin{aligned} g_1 &:= \sigma(\mathbf{y}_1)^T(b\mathbf{s}) + \mathbf{y}_1^T\sigma(b\mathbf{s}) - \sigma(\mathbf{B}\mathbf{y}_2)^T(b\mathbf{m}) - (\mathbf{B}\mathbf{y}_2)^T\sigma(b\mathbf{m}) \\ g_0 &:= \sigma(\mathbf{y}_1)^T\mathbf{y}_1 + \sigma(\mathbf{B}\mathbf{y}_2)^T\mathbf{B}\mathbf{y}_2. \end{aligned} \tag{25}$$

The idea is then to additionally send a commitment $t = \mathbf{b}^T\mathbf{s}_2 + g_1$ to g_1 and send

$$v := g_0 + \mathbf{b}^T\mathbf{y}_2 = \sigma(\mathbf{y}_1)^T\mathbf{y}_1 + \sigma(\mathbf{B}\mathbf{y}_2)^T\mathbf{B}\mathbf{y}_2 + \mathbf{b}^T\mathbf{y}_2 \tag{26}$$

in the clear. Then, the verifier can check that:

$$v \stackrel{?}{=} \sigma(\mathbf{z}_1)^T\mathbf{z}_1 + \sigma(\mathbf{t}_B - \mathbf{B}\mathbf{z}_2)^T(\mathbf{t}_B - \mathbf{B}\mathbf{z}_2) + (\mathbf{b}^T\mathbf{z}_2 - ct). \tag{27}$$

We present the protocol for proving this relation in Fig. 7 and summarise its security properties in Appendix A.

Dealing with relations which are not sign-invariant. Typically, relations do not have the property that they are sign-invariant. In this case, to apply bimodal Gaussian rejection sampling on the message $\mathbf{s}_1$ one needs to be more careful. As hinted in the discussion above, one solution would be to commit to the sign b (in the BDLOP part of the ABDLOP commitment) and prove that $b \in \{-1, 1\}$[12]. Then, for example, to prove an arbitrary quadratic equation with automorphisms (16), we commit to $(b\mathbf{s}_1, b\mathbf{m} \parallel b)$ and equivalently prove:

$$\langle b\mathbf{s}_1 \parallel b\mathbf{m}\rangle_\sigma^T\mathbf{R}_{i,2}\langle b\mathbf{s}_1 \parallel b\mathbf{m}\rangle_\sigma + b\mathbf{r}_{i,1}^T\langle b\mathbf{s}_1 \parallel b\mathbf{m}\rangle_\sigma + r_{i,0} = 0$$

[12] Proving that b is a sign has already been covered in [LNP22, Section 5.1].

which is a quadratic equation in $b\mathbf{s}_1, b\mathbf{m}$ and b. Then, in the soundness argument of [LNP22] we would extract $\bar{\mathbf{s}}_1, \bar{\mathbf{m}}$ and $\bar{b} \in \{-1, 1\}$ which satisfy

$$\langle \bar{\mathbf{s}}_1 \parallel \bar{\mathbf{m}} \rangle_\sigma^T \mathbf{R}_{i,2} \langle \bar{\mathbf{s}}_1 \parallel \bar{\mathbf{m}} \rangle_\sigma + \bar{b} \mathbf{r}_{i,1}^T \langle \bar{\mathbf{s}}_1 \parallel \bar{\mathbf{m}} \rangle_\sigma + r_{i,0} = 0.$$

Finally, since we proved that $\bar{b}$ is a sign, we define $(\mathbf{s}_1^*, \mathbf{m}^*) := (\bar{b}\bar{\mathbf{s}}_1, \bar{b}\bar{\mathbf{m}})$ and deduce that

$$\langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma^T \mathbf{R}_{i,2} \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma + \mathbf{r}_{i,1}^T \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma + r_{i,0} = 0$$

which is what we wanted to extract at the very beginning.

Similarly, to prove (18), we want to prove instead

$$\mathbf{P}_s(b\mathbf{s}_1) + \mathbf{P}_m(b\mathbf{m}) + b\mathbf{f} \in \{0, b\}^{n_{\mathsf{bin}} d}.$$

We observe that $x \in \{0, b\} = \{\frac{b-1}{2}, \frac{b+1}{2}\}$ if and only if $x - \frac{b-1}{2} \in \{0, 1\}$. Hence, the statement above is equivalent to:

$$\mathbf{P}_s(b\mathbf{s}_1) + \mathbf{P}_m(b\mathbf{m}) + b(\mathbf{f} - 2^{-1} \cdot \mathbf{1}) + 2^{-1} \cdot \mathbf{1} \in \{0, 1\}^{n_{\mathsf{bin}} d}$$

where $\mathbf{1} \in \mathcal{R}_q^{n_{\mathsf{bin}}}$ is the polynomial vector with the coefficient vector $\vec{1}$. Hence, we reduced our problem to proving that a linear combination of $b\mathbf{s}_1, b\mathbf{m}$ and b has binary coefficients. We conclude that using similar techniques, one can transform all the relations in $\mathbf{s}_1, \mathbf{m}$ described in Section 2.7 to equivalent ones in $b\mathbf{s}_1, b\mathbf{m}$ and $b \in \{-1, 1\}$.

4 Efficient One-out-of-Many Proofs

In this section we construct an efficient logarithmic-size one-out-of-many proof [GK15] with applications to lattice-based ring and group signatures using techniques from [LNP22] as the building block. In Appendix C we show how to further reduce the proof size using the techniques developed in Section 3, and eventually describe our ring signature construction.

The one-out-of-many proof considers the following problem. Informally, we want to prove knowledge of an opening to some commitment contained in a public set S without revealing any information about the commitment itself. In the lattice setting, we we would like to prove knowledge of a short vector such that $\mathbf{As} \in S$, where S is a public set $S = \{\mathbf{t}_1, \ldots, \mathbf{t}_N\} \subseteq \mathcal{R}_q^{\mathfrak{n}}$ of size $N = d \cdot \mathfrak{d}^k$. In this section we assume that $\mathbf{s} \in \{0, 1\}^{\mathfrak{m} d}$ has binary coefficients and $d = \mathfrak{l} \cdot \mathfrak{d}$ for $\mathfrak{l} \in \mathbb{N}$. For simplicity, we can already instantiate some of these parameters as $(d, \mathfrak{d}, \mathfrak{l}) = (64, 8, 16)$.

We now use the observation from [ESS+19, GK15, BCC+15] that $\mathbf{As} \in S$ if and only if there exists a binary vector $\vec{v} \in \{0, 1\}^N$ with exactly one 1, i.e. a unit vector, such that

$$\left[\vec{t}_1 \ \vec{t}_2 \cdots \vec{t}_N\right] \vec{v} = A\vec{s} \tag{28}$$

where $A = \mathsf{rot}(\mathbf{A}) \in \mathbb{Z}_q^{\mathfrak{n} d \times \mathfrak{m} d}$ is the the rotation matrix of $\mathbf{A}$. One could then directly prove knowledge of $\vec{s}$ and $\vec{v}$ which satisfy conditions above using the protocol from Section 2.7. However, the proof size grows linearly in N since we would commit to the whole vector $\vec{v}$.

In order to circumvent this limitation, [GK15, BCC+15] observe that vector $\vec{v}$ can be uniquely decomposed into unit vectors $\vec{v}_1, \ldots, \vec{v}_k \in \{0, 1\}^{\mathfrak{d}}$ and $\vec{v}_{k+1} \in \{0, 1\}^d$ such that

$$\vec{v} = \vec{v}_1 \otimes \vec{v}_2 \otimes \cdots \otimes \vec{v}_{k+1} := \vec{v}_1 \otimes (\vec{v}_2 \otimes (\cdots \otimes (\vec{v}_k \otimes \vec{v}_{k+1}))). \tag{29}$$

For notational convenience, let us define the set of polynomials $\mathcal{X}$ in $\mathcal{R}_q$ with their coefficient vectors being a unit vector. Concretely, $\mathcal{X}$ is defined as follows:

$$\mathcal{X} := \{1, X, X^2, \ldots, X^{d-1}\}.$$

In the end, we want to commit to $\mathbf{s}$ and polynomials $u_1, \ldots, u_k, v_{k+1} \in \mathcal{X}$ such that $\vec{u}_i = \vec{v}_i \parallel 0^{d-\mathfrak{d}} \in \mathbb{Z}_q^{d}$[13] for $i \in [k]$ and prove

$$T(\vec{v}_1 \otimes \cdots \otimes \vec{v}_{k+1}) = A\vec{s} \tag{30}$$

where $T \in \mathbb{Z}_q^{\mathsf{n}d \times N}$ is the matrix on the left-hand side of (28). We formally define the corresponding relation:

$$R_{\mathsf{oom}} := \left\{ \begin{array}{c} ((T, A), (\mathbf{s}, u_1, \ldots, u_k, v_{k+1})) : \mathbf{s} \in \{0,1\}^{\mathsf{m}d} \wedge T(\vec{v}_1 \otimes \cdots \otimes \vec{v}_{k+1}) = A\vec{s} \\ \wedge u_1, \ldots, u_k, v_{k+1} \in \mathcal{X} \text{ where } \vec{u}_i := \vec{v}_i \parallel 0^{d-\mathfrak{d}} \end{array} \right\}.$$

We now describe a commit-and-prove system for relation R_{oom} using the ABDLOP commitment. Suppose that $k \geqslant 1$, otherwise one can prove this relation directly using the framework from [LNP22].

First, note that proving $u_1, \ldots, u_k, v_{k+1} \in \mathcal{X}$ and $\mathbf{s} \in \{0,1\}^{\mathsf{m}d}$ can be done directly using the techniques from Section 2.7 hence we focus first on (30). Our strategy to prove this equation with $k-1$ tensor products would be somehow to reduce it to proving an equation of the same form with only $k-2$ tensor products. Then, by recursion, we will end up with a system of linear equations with no tensor products involved and thus we can apply the methods presented in Section 2.7.

The key idea to reduce the number of tensor products is to ask the verifier for $\mathfrak{l}$ challenges $\vec{\varphi}_1, \ldots, \vec{\varphi}_{\mathfrak{l}} \in \mathbb{Z}_q^{\mathsf{n}d}$ and then prove that:

$$\langle T(\vec{v}_1 \otimes \cdots \otimes \vec{v}_{k+1}) - A\vec{s}, \vec{\varphi}_i \rangle = 0 \quad \text{for } i = 1, 2, \ldots, \mathfrak{l}.$$

Note that if (30) was not true, then these $\mathfrak{l}$ equations above would hold with probability at most $q_1^{-\mathfrak{l}}$. Now, if we write

$$T := \begin{bmatrix} T_{0,1} & T_{0,2} & \cdots & T_{0,\mathfrak{d}} \end{bmatrix} \quad \text{where each } T_{0,i} \in \mathbb{Z}_q^{\mathsf{n}d \times d\mathfrak{d}^{k-1}}$$

then by simple algebraic manipulation we obtain

$$\begin{aligned} \langle T(\vec{v}_1 \otimes \cdots \otimes \vec{v}_{k+1}) - A\vec{s}, \vec{\varphi}_i \rangle &= \langle \vec{v}_1 \otimes \cdots \otimes \vec{v}_{k+1}, \vec{\varphi}_i^T T \rangle - \langle \vec{s}, A^T \vec{\varphi}_i \rangle \\ &= \langle \vec{v}_1, T_{1,i}(\vec{v}_2 \otimes \cdots \otimes \vec{v}_{k+1}) \rangle - \langle \vec{s}, A^T \vec{\varphi}_i \rangle \end{aligned}$$

where

$$T_{1,i} := \begin{bmatrix} \vec{\varphi}_i^T T_{0,1} \\ \vdots \\ \vec{\varphi}_i^T T_{0,\mathfrak{d}} \end{bmatrix} \in \mathbb{Z}_q^{\mathfrak{d} \times d\mathfrak{d}^{k-1}}.$$

Now, let us define $\vec{w}_i := T_{1,i}(\vec{v}_2 \otimes \cdots \otimes \vec{v}_{k+1})$ and $w \in \mathcal{R}_q$ such that

$$\vec{w} = \vec{w}_1 \parallel \cdots \parallel \vec{w}_{\mathfrak{l}} \in \mathbb{Z}_q^d.$$

Next, we commit to w and show that for all i,

$$\langle \vec{v}_1, \vec{w}_i \rangle - \langle \vec{s}, A^T \vec{\varphi}_i \rangle = 0 \quad \text{and} \quad \vec{w}_i := T_{1,i}(\vec{v}_2 \otimes \cdots \otimes \vec{v}_{k+1}).$$

We observe that the first statement is equivalent to proving that the constant coefficient of

$$X^{(i-1)\mathfrak{d}} u_1 \sigma(w) - \sigma(\mathbf{a}_i)^T \mathbf{s}$$

is equal to zero where the coefficient vector of $\mathbf{a}_i \in \mathcal{R}_q^{\mathsf{m}}$ is exactly $\vec{a}_i := A^T \vec{\varphi}_i$.

Lemma 4.1. *Let $i \in [\mathfrak{l}]$. Then, the constant coefficient of $X^{(i-1)\mathfrak{d}} u_1 \sigma(w) \in \mathcal{R}_q$ is equal to $\langle \vec{v}_1, \vec{w}_i \rangle$.*

Proof. First, we note that $\langle \vec{v}_1, \vec{w}_i \rangle = \langle X^{(i-1)\mathfrak{d}} u_1, w \rangle$. Here, we used the fact that the coefficient vector of u_1 is of the form $\vec{v}_1 \parallel 0^{d-\mathfrak{d}}$. Then, by Lemma 2.2, $\langle X^{(i-1)\mathfrak{d}} u_1, w \rangle$ is the constant coefficient of $X^{(i-1)\mathfrak{d}} u_1 \sigma(w)$.

[13] Alternatively, $u_i \in \{1, X, X^2, \ldots, X^{\mathfrak{d}-1}\}$.

On the other hand, the second statement can be combined for all i and written as:

$$\vec{w} = \begin{bmatrix} T_{1,1} \\ \vdots \\ T_{1,\mathfrak{l}} \end{bmatrix} (\vec{v}_2 \otimes \cdots \otimes \vec{v}_{k+1}). \tag{31}$$

Thus, we reduce the one-out-of-many problem to proving knowledge of a tuple $(\mathbf{s}, u_1, \ldots, u_k, v_{k+1}, w)$ which satisfies the following conditions: (i) $\mathbf{s} \in \{0,1\}^{\mathsf{m}d}$, (ii) $T_1(\vec{v}_2 \otimes \cdots \otimes \vec{v}_{k+1}) = \vec{w}$, (iii) for all $i \in [\mathfrak{l}]$, the constant coefficient of $X^{(i-1)\mathfrak{d}} u_1 \sigma(w) - \sigma(\mathbf{a}_i)^T \mathbf{s}$ is zero, and (iv) $u_1, \ldots, u_k, v_{k+1} \in \mathcal{X}$ where

$$\vec{u}_i := \vec{v}_i \parallel 0^{d-\mathfrak{d}} \text{ for } i \in [k] \quad \text{and} \quad T_1 := \begin{bmatrix} T_{1,1} \\ \vdots \\ T_{1,\mathfrak{l}} \end{bmatrix} \in \mathbb{Z}_q^{d \times d\mathfrak{d}^{k-1}}.$$

Note that the second statement only involves $k-2$ tensor products.

We can define the correspond relation as:

$$R := \left\{ \begin{array}{c} \big((T_1, (\mathbf{a}_i)_{i\in[\mathfrak{l}]}), (\mathbf{s}, u_1, \ldots, u_k, v_{k+1}, w)\big) : \mathbf{s} \in \{0,1\}^{\mathsf{m}d} \wedge T_1(\vec{v}_2 \otimes \cdots \otimes \vec{v}_{k+1}) = \vec{w} \\ \wedge \forall i \in [\mathfrak{l}], \text{ const coeff. of } X^{(i-1)\mathfrak{d}} u_1 \sigma(w) - \sigma(\mathbf{a}_i)^T \mathbf{s} \text{ is zero} \\ \wedge u_1, \ldots, u_k, v_{k+1} \in \mathcal{X} \text{ where } \vec{u}_i := \vec{v}_i \parallel 0^{d-\mathfrak{d}} \end{array} \right\}.$$

Intermediate relations. We construct a commit-and-prove system for relation R using recursion. Namely, take $1 \leqslant j \leqslant k$ and consider the following generalised relation

$$R_j := \left\{ \begin{array}{c} \Big((T_j \in \mathbb{Z}_q^{d \times d\mathfrak{d}^{k-j}}, (\mathbf{a}_i)_{i\in[\mathfrak{l}]}, (\varphi_{\iota,i})_{\iota\in[j-1], i\in[\mathfrak{l}]}), (\mathbf{s}, u_1, \ldots, u_k, v_{k+1}, w_1, \ldots, w_j)\Big) : \\ \mathbf{s} \in \{0,1\}^{\mathsf{m}d} \wedge T_j(\vec{v}_{j+1} \otimes \cdots \otimes \vec{v}_{k+1}) = \vec{w}_j \\ \wedge \forall i \in [\mathfrak{l}], \text{ const coeff. of } X^{(i-1)\mathfrak{d}} u_1 \sigma(w_1) - \sigma(\mathbf{a}_i)^T \mathbf{s} \text{ is zero} \\ \wedge \forall \iota \in [j-1], i \in [\mathfrak{l}], \text{ const coeff. of } X^{(i-1)\mathfrak{d}} u_{\iota+1} \sigma(w_{\iota+1}) - \sigma(\varphi_{\iota,i}) w_\iota \text{ is zero} \\ \wedge u_1, \ldots, u_k, v_{k+1} \in \mathcal{X} \text{ where } \vec{u}_i := \vec{v}_i \parallel 0^{d-\mathfrak{d}} \end{array} \right\}. \tag{32}$$

We highlight that in R_j elements $\varphi_{\iota,i}$ are polynomials in $\mathcal{R}_q$. Also, it is easy to see that $R_1 = R$.

Base case. We first show how to prove R_k only using the methods described in Section 2.7. In the following, we say that a statement is of Type-n if it corresponds to the Statement n in Section 2.7.

To begin with, using the ABDLOP commitment we commit to

$$\mathbf{s}_1 := \mathbf{s} \parallel u_1 \parallel \cdots \parallel u_k \parallel v_{k+1} \in \mathcal{R}_q^{\mathsf{m}+k+1}, \quad \mathbf{m} := (w_1, \ldots, w_k) \in \mathcal{R}_q^k.$$

Then, proving $\mathbf{s} \in \{0,1\}^{\mathsf{m}d}$ and $u_1, \ldots, u_k, v_{k+1} \in \{0,1\}^d$ is of Type-3. Next, by Lemma 2.2, proving $T_k \vec{v}_{k+1} = \vec{w}_k$ and $\langle \vec{1}, \vec{v}_i \rangle = 1$ for $i \in [k+1]$ is of Type-2. Further, it is easy to see that proving the constant coefficients of $X^{(i-1)\mathfrak{d}} u_1 \sigma(w_1) - \sigma(\mathbf{a}_i)^T \mathbf{s}$ and $X^{(i-1)\mathfrak{d}} u_{\iota+1} \sigma(w_{\iota+1}) - \sigma(\varphi_{\iota,i}) w_\iota$ vanish is of Type-2. Finally, proving that $\vec{u}_i = \vec{v}_i \parallel 0^{d-\mathfrak{d}}$ for $i \in [k]$ is equivalent to proving that the constant coefficient of $X^{-j} u_i$ is zero for $\mathfrak{d} \leqslant j \leqslant d$, which is of Type-2.

From now on, we will call the commit-and-prove protocol for relation R_k described above as Π_k.

Recursive step. Let us assume we have a commit-and-prove system Π_{j+1} for relation R_{j+1} where $2 \leqslant j+1 \leqslant k$. Now we want to use it to prove relation R_j. We observe that the only statement which is included in R_j but not in R_{j+1} is

$$T_j(\vec{v}_{j+1} \otimes \cdots \otimes \vec{v}_{k+1}) = \vec{w}_j. \tag{33}$$

We prove this equation as before. Namely, we ask the verifier for $\mathfrak{l}$ challenges $\vec{\varphi}_{j,1}, \ldots, \vec{\varphi}_\mathfrak{l} \in \mathbb{Z}_q^d$ and then prove that:

$$\langle T_j(\vec{v}_{j+1} \otimes \cdots \otimes \vec{v}_{k+1}) - \vec{w}_j, \vec{\varphi}_{j,i} \rangle = 0 \quad \text{for } i = 1, 2, \ldots, \mathfrak{l}.$$

Note that if (30) was not true, then these $\mathfrak{l}$ equations above would hold with probability at most $q_1^{-\mathfrak{l}}$. Now, if we write

$$T_j := \begin{bmatrix} T_{j,1} & T_{j,2} & \cdots & T_{j,\mathfrak{d}} \end{bmatrix} \quad \text{where each } T_{j,i} \in \mathbb{Z}_q^{d \times d\mathfrak{d}^{k-j-1}}$$

then we have

$$\begin{aligned} \langle T_j(\vec{v}_{j+1} \otimes \cdots \otimes \vec{v}_{k+1}) - \vec{w}_j, \vec{\varphi}_{j,i} \rangle &= \langle \vec{v}_{j+1} \otimes \cdots \otimes \vec{v}_{k+1}, \vec{\varphi}_{j,i}^T T_j \rangle - \langle \vec{w}_j, \vec{\varphi}_{j,i} \rangle \\ &= \langle \vec{v}_{j+1} \otimes \cdots \otimes \vec{v}_{k+1}, \vec{\varphi}_{j,i}^T T_j \rangle - \langle \vec{w}_j, \vec{\varphi}_{j,i} \rangle \\ &= \langle \vec{v}_{j+1}, T_{j+1,i}(\vec{v}_{j+2} \otimes \cdots \otimes \vec{v}_{k+1}) \rangle - \langle \vec{w}_j, \vec{\varphi}_{j,i} \rangle \end{aligned}$$

where

$$T_{j+1,i} := \begin{bmatrix} \vec{\varphi}_{j,i}^T T_{j,1} \\ \vdots \\ \vec{\varphi}_{j,i}^T T_{j,\mathfrak{d}} \end{bmatrix} \in \mathbb{Z}_q^{\mathfrak{d} \times \mathfrak{d}^{k-j-1}}.$$

Now, let us define $\vec{w}_{j+1,i} := T_{j+1,i}(\vec{v}_{j+2} \otimes \cdots \otimes \vec{v}_{k+1})$ and $w_{j+1} \in \mathcal{R}_q$ so that

$$\vec{w}_{j+1} = \vec{w}_{j+1,1} \parallel \cdots \parallel \vec{w}_{j+1,\mathfrak{l}} \in \mathbb{Z}_q^d.$$

Then, we need to show that for all i,

$$\langle \vec{v}_{j+1}, \vec{w}_{j+1,i} \rangle - \langle \vec{w}_j, \vec{\varphi}_{j,i} \rangle = 0 \quad \text{and} \quad \vec{w}_{j+1,i} = T_{j+1,i}(\vec{v}_{j+2} \otimes \cdots \otimes \vec{v}_{k+1}).$$

The first statement is equivalent to proving that the constant coefficient of

$$X^{(i-1)\mathfrak{d}} u_{j+1} \sigma(w_{j+1}) - \sigma(\varphi_{j,i}) w_j$$

is equal to zero. The second statement, however, can be combined for all i and written as:

$$\vec{w}_{j+1} = T_{j+1}(\vec{v}_{j+2} \otimes \cdots \otimes \vec{v}_{k+1}) \quad \text{where} \quad T_{j+1} := \begin{bmatrix} T_{j+1,1} \\ \vdots \\ T_{j+1,\mathfrak{l}} \end{bmatrix} \in \mathbb{Z}_q^{d \times \mathfrak{d}^{k-j-1}}. \tag{34}$$

Therefore, we reduced proving (33) to proving that

- $X^{(i-1)\mathfrak{d}} u_{j+1} \sigma(w_{j+1}) - \sigma(\varphi_{j,i}) w_j$ is equal to zero
- $\vec{w}_{j+1} = T_{j+1}(\vec{v}_{j+2} \otimes \cdots \otimes \vec{v}_{k+1})$

which in combination with other relations in R_j, it directly reduces to proving relations in R_{j+1}.

In Fig. 8 we give a commit-and-prove protocol for relation R_j which uses Π_j as a black-box. One observes by discussion above that the correctness error for Π_j is the same as for Π_k (which can be calculated directly from [LNP22]). Simulatability follows from the fact that running Π_j, and thus Π_{j+1} up to Π_k as subroutines, involves only sending intermediate commitments t_i to w_i which can be simulated by the Extended-MLWE assumption. Finally, one can prove by induction that the knowledge soundness error for the protocol Π_j is $(k-j) \cdot q_1^{-\mathfrak{l}} + \varepsilon_k$ where ε_k is the knowledge soundness error for Π_k and is computed as in [LNP22]. The expected runtime of the extractor, that has black-box access to a (potentially malicious) prover which runs in time T, is $2^{k-j} \cdot \mathsf{poly}(T)$. Consequently, we can only consider values k which are logarithmic in the security parameter. Due to space constraints, we refer to Appendix B for more details.

Back to R_{oom}. At the very beginning of this section we showed how to reduce proving relation R_{oom} to proving R_1. Later on, we described a commit-and-prove protocol for R_1. Hence, we combine these two results to obtain a commit-and-prove protocol Π_{oom} in Fig. 9 for the one-out-of-many relation R_{oom}. Arguing similarly as above, the correctness error for Π_{oom} is the same as for Π_k and the soundness error is at most $kq_1^{-\mathfrak{l}} + \varepsilon_k$.

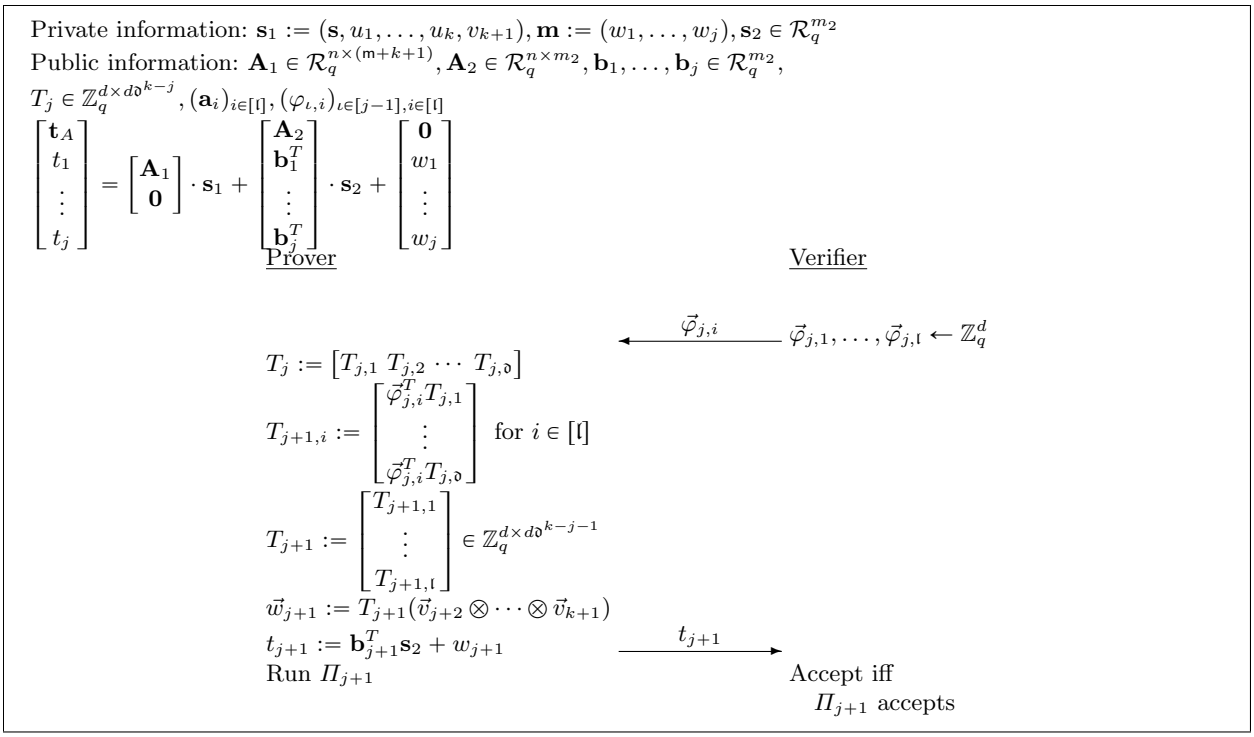

Fig. 8: Commit-and-prove protocol Π_j for the relation R_j where $j < k$.

References

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

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

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

AP12. Jacob Alperin-Sheriff and Chris Peikert. Circular and KDM security for identity-based encryption. In *Public Key Cryptography*, volume 7293 of *Lecture Notes in Computer Science*, pages 334–352. Springer, 2012.

ASA16. Jacob Alperin-Sheriff and Daniel Apon. Dimension-preserving reductions from lwe to lwr. Cryptology ePrint Archive, Report 2016/589, 2016. `https://ia.cr/2016/589`.

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

BCC+15. Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. Short accountable ring signatures based on DDH. In *ESORICS (1)*, volume 9326 of *Lecture Notes in Computer Science*, pages 243–265. Springer, 2015.

BDE+22. Maxime Buser, Rafael Dowsley, Muhammed F. Esgin, Clémentine Gritti, Shabnam Kasra Kermanshahi, Veronika Kuchta, Jason T. LeGrow, Joseph K. Liu, Raphael C.-W. Phan, Amin Sakzad, Ron Steinfeld, and Jiangshan Yu. A survey on exotic signatures for post-quantum blockchain: Challenges research directions. Cryptology ePrint Archive, Paper 2022/1151, 2022. `https://eprint.iacr.org/2022/1151`.

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

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

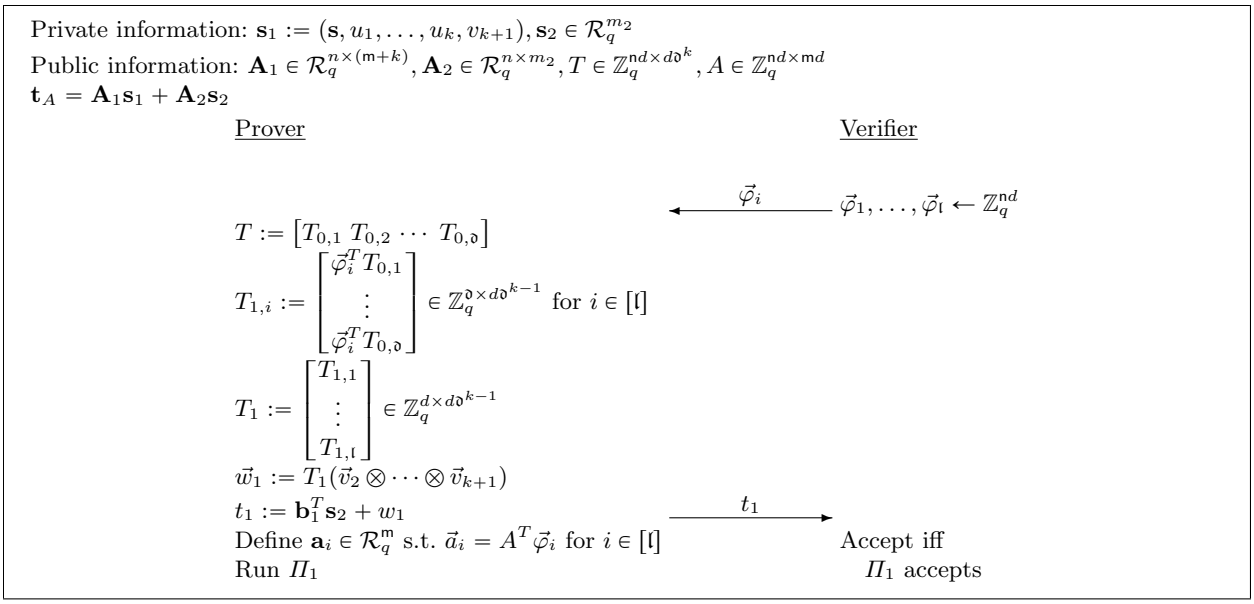

Fig. 9: Commit-and-prove protocol Π_{oom} for the relation R_{oom}.

BJRW21. Katharina Boudgoust, Corentin Jeudy, Adeline Roux-Langlois, and Weiqiang Wen. On the hardness of module-lwe with binary secret. In *CT-RSA*, volume 12704 of *Lecture Notes in Computer Science*, pages 503–526. Springer, 2021.

BKP20. Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In *ASIACRYPT (2)*, volume 12492 of *Lecture Notes in Computer Science*, pages 464–492. Springer, 2020.

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

DDGR20. Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. LWE with side information: Attacks and concrete security estimation. In *CRYPTO (2)*, volume 12171 of *Lecture Notes in Computer Science*, pages 329–358. Springer, 2020.

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

DGK+10. Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In Daniele Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, 2010.

DLP14. Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. Efficient identity-based encryption over NTRU lattices. In *ASIACRYPT*, pages 22–41, 2014.

DP16. Léo Ducas and Thomas Prest. Fast fourier orthogonalization. In *ISSAC*, pages 191–198, 2016.

ENS20. Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In *ASIACRYPT (2)*, pages 259–288, 2020.

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

ESS+19. Muhammed F. Esgin, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, and Dongxi Liu. Short lattice-based one-out-of-many proofs and applications to ring signatures. In *ACNS*, volume 11464 of *Lecture Notes in Computer Science*, pages 67–88. Springer, 2019.

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

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

FHK+20. Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Prest, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, , and Zhenfei Zhang. Falcon: Fast-fourier lattice-based compact signatures over ntru. Technical report, 2020. https:/https://falcon-sign.info/falcon.pdf.

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

GK15. Jens Groth and Markulf Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In *EUROCRYPT*, pages 253–280, 2015.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206, 2008.

HHGP+03. Jeffrey Hoffstein, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman, and William Whyte. Ntrusign: Digital signatures using the ntru lattice. In *CT-RSA*, pages 122–140, 2003.

LAZ19. Xingye Lu, Man Ho Au, and Zhenfei Zhang. Raptor: A practical lattice-based (linkable) ring signature. In *ACNS*, volume 11464 of *Lecture Notes in Computer Science*, pages 110–130. Springer, 2019.

LMPR08. Vadim Lyubashevsky, Daniele Micciancio, Chris Peikert, and Alon Rosen. SWIFFT: A modest proposal for FFT hashing. In *FSE*, pages 54–72, 2008.

LNP22. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plancon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. Cryptology ePrint Archive, Paper 2022/284, 2022. `https://eprint.iacr.org/2022/284`. To appear at CRYPTO 2022.

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

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

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

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

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

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

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

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

NY90. Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *STOC*, pages 427–437, 1990.

TWZ20. Yang Tao, Xi Wang, and Rui Zhang. Short zero-knowledge proof of knowledge for lattice-based commitment. In *PQCrypto*, volume 12100 of *Lecture Notes in Computer Science*, pages 268–283. Springer, 2020.

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

YEL+21. Tsz Hon Yuen, Muhammed F. Esgin, Joseph K. Liu, Man Ho Au, and Zhimin Ding. Dualring: Generic construction of ring signatures with efficient instantiations. In *CRYPTO (1)*, volume 12825 of *Lecture Notes in Computer Science*, pages 251–281. Springer, 2021.

A Security Analysis of the Protocol in Fig. 7

Lemma A.1. *Consider the protocol in Fig. 7 and let χ be the uniform distribution over S_ν for $\nu \in \mathbb{N}$. Suppose $\mathfrak{s}_1 = \gamma_1 \alpha\eta$ and $\mathfrak{s}_2 = \gamma_2 \nu\eta\sqrt{m_2 d}$ for some $\gamma_1, \gamma_2 > 0$ where η is chosen as in Section 2.5.*

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

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

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

For soundness, let $B_i := 2\mathfrak{s}_i\sqrt{2m_id}$ for $i = 1, 2$. Then, there is an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a probabilistic prover $\mathcal{P}^*$, which convinces $\mathcal{V}$ with probability $\varepsilon \geqslant 2/|\mathcal{C}|$, extractor $\mathcal{E}$ with probability at least $\varepsilon - 2/|\mathcal{C}|$ either outputs $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2) \in \mathcal{R}_q^{m_1+m_2+\ell}$ and $\bar{c} \in \bar{\mathcal{C}}$ such that $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c})$ is a relaxed opening of $(\mathbf{t}_A, \mathbf{t}_B)$ and $\sigma(\bar{\mathbf{s}}_1)^T\bar{\mathbf{s}}_1 + \sigma(\bar{\mathbf{m}})^T\bar{\mathbf{m}} = 0$, or a $\mathsf{MSIS}_{n,m_1+m_2,4\eta\sqrt{B_1^2+B_2^2}}$ solution for $\begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix}$ in expected time at most $3T$ where running $\mathcal{P}^*$ once is assumed to take at most T time.

Proof. The knowledge soundness follows identically as in [LNP22, Theorem 4.2] thus we only focus on completeness and simulatability.

Completeness. The correctness error comes directly from Lemmas 2.8 and 3.1. Here, we use the fact that the commitment is generated by the prover and thus the sign b is freshly sampled and not given. Next, $\|\mathbf{z}_i\| \leqslant \mathfrak{s}\sqrt{2m_id}$ with an overwhelming probability by Lemma 2.5. Finally, the other verification equations follow from the discussion above.

Simulatability. We prove the statement using a hybrid argument as before. First, we describe an efficient simulator $\mathcal{S}_1$ which still knows $\mathbf{s}_1, \mathbf{m}$ and simulates both the commitment and the transcript in the following way. It executes the prover's algorithm but instead of constructing $\mathbf{z}_2$ honestly as in the protocol, $\mathcal{S}_2$ samples $\mathbf{y}_2 \leftarrow D_{\mathfrak{s}_2}^{m_2d}$ and defines $\mathbf{z}_+ := \mathsf{sign}(\langle c\mathbf{s}_2, \mathbf{y}_2\rangle) \cdot \mathbf{y}_2$ and $\mathbf{z}_- := -\mathbf{z}_+$. Then, it sets $\mathbf{z}_2 := \mathbf{z}_+$ with probability p and $\mathbf{z}_2 := \mathbf{z}_-$ with probability $1 - p$ where p is defined as

$$p := \frac{\exp\left(\frac{|\langle c\mathbf{s}_2, \mathbf{y}_2\rangle|}{M}\right)}{\exp\left(\frac{|\langle c\mathbf{s}_2, \mathbf{y}_2\rangle|}{M}\right) + 1}. \tag{35}$$

It then continues with probability $1/M_2$ where $M_i := \exp(1/(2\gamma_i^2))$ for $i = 1, 2$. By Lemma 3.1, the non-aborted simulated commitment and transcript by $\mathcal{S}_1$ and $\mathcal{S}_2$ are identical.

Further, we define an efficient simulator $\mathcal{S}_2$, which still knows $\mathbf{s}_1, \mathbf{m}$ and simulates both the commitment and the transcript as follows. Namely, it executes the $\mathcal{S}_2$ algorithm but instead of generating $(\mathbf{t}_A, \mathbf{t}_B, t_{\mathsf{sign}}, \mathbf{t}_g, t)$ honestly, it samples $\mathbf{u} \leftarrow \mathcal{R}_q^{n+\ell+1}$ and computes:

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

Now, under the Extended-$\mathsf{MLWE}_{n+\ell+1,m_2-n-\ell-1,\chi,\mathcal{C},D_{\mathfrak{s}_2}}$ assumption, the non-aborted output distribution of $\mathcal{S}_1$ is computationally indistinguishable from the non-aborted output distribution of $\mathcal{S}_2$.

Moreover, we describe $\mathcal{S}_3$, which follows the algorithm for $\mathcal{S}_3$ but it directly samples $(\mathbf{t}_A, \mathbf{t}_B, t) \leftarrow \mathcal{R}_q^{n+\ell+1}$. Clearly, the outputs of $\mathcal{S}_2$ and $\mathcal{S}_3$ are identical. Recall that $\mathcal{S}_3$ still computes $\mathbf{z}_1 := \mathbf{y}_1 + bc\mathbf{s}_1$ for $b \leftarrow \{-1, 1\}$ and $\mathbf{y}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1}$ identically as the honest prover $\mathcal{P}$.

Finally, we define our simulator $\mathcal{S}$, which has no access to private information anymore, as follows. Concretely, it executes the $\mathcal{S}_3$ algorithm but instead of generating $\mathbf{z}_1$ honestly, it simply samples $\mathbf{z}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1}$. Now, since $b \leftarrow \{-1, 1\}$, the output distribution of $\mathcal{S}$ is identical to the non-aborted output of $\mathcal{S}_3$ by Lemma 2.8. Hence, the statement holds by the hybrid argument. □

B Security of the One-out-of-Many Proof

We start by analysing the commit-and-prove protocol Π_k defined in Section 4. It is a simple application of the general framework from [LNP22] and thus we can apply the security analysis from [LNP22, Theorem 5.3].

Lemma B.1. *Let $\gamma_1, \gamma_2, \gamma_3 > 0$ and $\lambda, \nu \in \mathbb{N}$[14]. Define χ to be the uniformly distribution on S_ν. Fix*

$$\mathfrak{s}_1 = \gamma_1 \eta \sqrt{\mathsf{m}d + k + 1}, \quad \mathfrak{s}_2 = \gamma_2 \nu \eta \sqrt{m_2 d}, \quad \mathfrak{s}_3 = \sqrt{337(\mathsf{m}d + k + 1)}$$

where η is defined in Section 2.5. Then, there is a 7-round commit-and-prove protocol Π_k for relation R_k with the following properties.

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

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

For commit-and-prove simulatability, there exists a simulator $\mathcal{S}$ that, without access to private information $\mathbf{s}_1, \mathbf{m}$, outputs a simulation of the commitment

$$\begin{bmatrix} \mathbf{t}_A \\ t_1 \\ \vdots \\ t_k \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{s} \\ u_1 \\ \vdots \\ u_k \\ v_{k+1} \end{bmatrix} + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{b}_1^T \\ \vdots \\ \mathbf{b}_k^T \end{bmatrix} \cdot \mathbf{s}_2 + \begin{bmatrix} \mathbf{0} \\ w_1 \\ \vdots \\ w_k \end{bmatrix}$$

along with a non-aborting transcript of the protocol between prover $\mathcal{P}$ and verifier $\mathcal{V}$ such that for every algorithm $\mathcal{A}$ that has advantage ε in distinguishing the simulated commitment and transcript from the real commitment and transcript, whenever the prover does not abort, there is an algorithm $\mathcal{A}'$ with the same running time that has advantage ε in distinguishing the Extended-$\mathsf{MLWE}_{n+k+\lambda+256/d+2, m_2-n-k-\lambda-256/d-2, \chi, \mathcal{C}, D_{\mathfrak{s}_2}}$.

For soundness, let $B_i := 2\mathfrak{s}_i\sqrt{2m_i d}$ for $i = 1, 2$ where $m_1 := \mathsf{m} + k + 1$ and $\mathcal{B} := 2\sqrt{\frac{256}{26}}\varrho\mathfrak{s}_3$. If

$$\begin{aligned} &q \geqslant 41 \cdot (\mathsf{m} + k + 1)\, d \cdot \mathcal{B}, && \text{to use [LNP22, Lemma 2.9]} \\ &q > \mathcal{B}_{\mathsf{arp}}^2 + \mathcal{B}\sqrt{(\mathsf{m} + k + 1)d}, && \text{to prove } \mathbf{s}, u_1, \ldots, u_k, v_{k+1} \text{ have binary coeff.} \end{aligned}$$

then there is an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a probabilistic prover $\mathcal{P}^*$, which convinces $\mathcal{V}$ with probability $\varepsilon \geqslant 2/|\mathcal{C}| + q_1^{-d/2} + q_1^{-\lambda} + 2^{-128}$, extractor $\mathcal{E}$ with probability at least $\varepsilon - 2/|\mathcal{C}| - q_1^{-d/2} - q_1^{-\lambda} - 2^{-128}$ either outputs $\mathbf{x} := ((\bar{\mathbf{s}}, \bar{u}_1, \ldots, \bar{u}_k, \bar{v}_{k+1}), (\bar{w}_1, \ldots, \bar{w}_k), \bar{\mathbf{s}}_2) \in \mathcal{R}_q^{m_1+m_2+k}$ and $\bar{c} \in \bar{\mathcal{C}}$ such that $(\mathbf{x}, \bar{c})$ is a relaxed opening of $(\mathbf{t}_A, (t_1, \ldots, t_k))$ and

$$\left(T_k \in \mathbb{Z}_q^{d \times d}, (\mathbf{a}_i)_{i \in [\mathsf{l}]}, (\varphi_{\iota,i})_{\iota \in [k-1], i \in [\mathsf{l}]}), (\bar{\mathbf{s}}, \bar{u}_1, \ldots, \bar{u}_k, \bar{v}_{k+1}, \bar{w}_1, \ldots, \bar{w}_k)\right) \in R_k$$

or a $\mathsf{MSIS}_{n, m_1+m_2, 4\eta\sqrt{B_1^2+B_2^2}}$ solution for $\begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \end{bmatrix}$ in expected time at most $24T$ where running $\mathcal{P}^*$ once is assumed to take at most T time.

Now, we can prove by induction the following security properties of the protocol Π_j for relation R_j defined in Fig. 8.

[14] Informally, parameters $\gamma_1, \gamma_2, \gamma_3$ are related to rejection sampling whereas λ is a parameter used for soundness amplification. Finally, the ABDLOP randomness vector $\mathbf{s}_2$ will have coefficients between $-\nu$ and ν.

Lemma B.2. *Let $j \in [k]$ and consider the commit-and-prove protocol in Fig. 8. Let $\gamma_1, \gamma_2, \gamma_3 > 0$ and $\lambda, \nu \in \mathbb{N}$. Define χ to be the uniformly distribution on S_ν. Fix*

$$\mathfrak{s}_1 = \gamma_1 \eta \sqrt{\mathsf{m} d + k + 1}, \quad \mathfrak{s}_2 = \gamma_2 \nu \eta \sqrt{m_2 d}, \quad \mathfrak{s}_3 = \sqrt{337(\mathsf{m} d + k + 1)}$$

where η is defined in Section 2.5.

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

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

For commit-and-prove simulatability, there exists a simulator $\mathcal{S}$ that, without access to private information $\mathbf{s}_1, \mathbf{m}$, outputs a simulation of the commitment

$$\begin{bmatrix} \mathbf{t}_A \\ t_1 \\ \vdots \\ t_j \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{s} \\ u_1 \\ \vdots \\ u_k \\ v_{k+1} \end{bmatrix} + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{b}_1^T \\ \vdots \\ \mathbf{b}_j^T \end{bmatrix} \cdot \mathbf{s}_2 + \begin{bmatrix} \mathbf{0} \\ w_1 \\ \vdots \\ w_j \end{bmatrix}$$

along with a non-aborting transcript of the protocol between prover $\mathcal{P}$ and verifier $\mathcal{V}$ such that for every algorithm $\mathcal{A}$ that has advantage ε in distinguishing the simulated commitment and transcript from the real commitment and transcript, whenever the prover does not abort, there is an algorithm $\mathcal{A}'$ with the same running time that has advantage ε in distinguishing the Extended-$\mathsf{MLWE}_{n+k+\lambda+256/d+2, m_2-n-k-\lambda-256/d-2, \chi, \mathcal{C}, D_{\mathfrak{s}_2}}$.

For soundness, let $B_i := 2\mathfrak{s}_i\sqrt{2m_i d}$ for $i = 1, 2$ where $m_1 := \mathsf{m} + k + 1$ and $\mathcal{B} := 2\sqrt{\frac{256}{26}}\varrho\mathfrak{s}_3$. If

$$q \geqslant 41 \cdot (\mathsf{m} + k + 1)\, d \cdot \mathcal{B}, \qquad \text{to use [LNP22, Lemma 2.9]}$$
$$q > \mathcal{B}_{\mathsf{arp}}^2 + \mathcal{B}\sqrt{(\mathsf{m} + k + 1)d}, \qquad \text{to prove } \mathbf{s}, u_1, \ldots, u_k, v_{k+1} \text{ have binary coeff.}$$

then there is an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a probabilistic prover $\mathcal{P}^*$, which convinces $\mathcal{V}$ with probability $\varepsilon \geqslant 2/|\mathcal{C}| + (k-j)q_1^{-\mathfrak{l}} + q_1^{-d/2} + q_1^{-\lambda} + 2^{-128}$, extractor $\mathcal{E}$ with probability at least $\varepsilon - 2/|\mathcal{C}| - (k-j)q_1^{-\mathfrak{l}} - q_1^{-d/2} - q_1^{-\lambda} - 2^{-128}$ either outputs $\mathbf{x} := ((\bar{\mathbf{s}}, \bar{u}_1, \ldots, \bar{u}_k, \bar{v}_{k+1}), (\bar{w}_1, \ldots, \bar{w}_j), \bar{\mathbf{s}}_2) \in \mathcal{R}_q^{m_1+m_2+j}$ and $\bar{c} \in \bar{\mathcal{C}}$ such that $(\mathbf{x}, \bar{c})$ is a relaxed opening of $(\mathbf{t}_A, (t_1, \ldots, t_k))$ and

$$\left(T_j \in \mathbb{Z}_q^{d \times d\mathfrak{d}^{k-j}}, (\mathbf{a}_i)_{i\in[\mathfrak{l}]}, (\varphi_{\iota,i})_{\iota\in[j-1], i\in[\mathfrak{l}]}), (\bar{\mathbf{s}}, \bar{u}_1, \ldots, \bar{u}_k, \bar{v}_{k+1}, \bar{w}_1, \ldots, \bar{w}_j)\right) \in R_j$$

or a $\mathsf{MSIS}_{n, m_1+m_2, 4\eta\sqrt{B_1^2+B_2^2}}$ solution for $\begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix}$ in expected time at most $2^{k-j} \cdot 24T$ where running $\mathcal{P}^*$ once is assumed to take at most T time.

Proof. First, correctness follows directly from Lemma B.1 and the argument presented in Section 4. Then, for simulatability, we observe that before running Π_k, the prover only sends the "bottom part" commitments to w_i and these (as a part of the whole ABDLOP commitment to $((\mathbf{s}, u_1, \ldots, u_k, v_{k+1}), (w_1, \ldots, w_k))$) can be simulated as in the proof of Lemma A.1.

From now on, we only focus on the knowledge soundness property. We prove the statement by induction. First, consider $j = k$. Then, the statement follows directly from Lemma B.1 and the corresponding extractor runs in expected $24T$ time.

Now, assume Π_{j+1} is knowledge sound with knowledge error $(k-j-1)q_1^{-\mathfrak{l}} + 2|\mathcal{C}|^{-1} + q_1^{-d/2} + q_1^{-\lambda} + 2^{-128}$ for some $j+1 \leqslant k$. Let $\mathcal{P}^*$ be a probabilistic prover which runs in time at most T and convinces the verifier with probability $\epsilon > (k-j)q_1^{-\mathfrak{l}} + 2|\mathcal{C}|^{-1} + q_1^{-d/2} + q_1^{-\lambda} + 2^{-128}$. Define a deterministic algorithm $\mathcal{A}(\rho_P, \rho_E, (\vec{\varphi}_{j,i}))$ which given randomness $\rho = (\rho_P, \rho_E) \in \mathfrak{R}_P \times \mathfrak{R}_E$ and challenge $\vec{\varphi}_{j,1}, \ldots, \vec{\varphi}_{j,\mathfrak{l}} \in \mathbb{Z}_q^d$ does the following. It first runs $\mathcal{P}^*(\rho_P)$ on randomness ρ_P with challenges $(\vec{\varphi}_{j,i})$ and stops after $\mathcal{P}^*$ sends t_{j+1}. Then, it runs the extractor $\mathcal{E}^*(\rho_E)$ for Π_{j+1} with randomness ρ_E (which runs $\mathcal{P}^*(\rho_P, (\vec{\varphi}_{j,i}))$ in a black-box way).

We say that $\mathcal{A}$ succeeds if $\mathcal{A}$ outputs $((\varphi_{j,i}), t_{j+1}, \bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{w}_{j+1}, \bar{\mathbf{s}}_2, \bar{c})$ such that $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{w}_{j+1}, \bar{\mathbf{s}}_2, \bar{c}))$ is a relaxed opening of $(\mathbf{t}_A, (t_1, \ldots, t_j))$ and

$$\left((T_{j+1} \in \mathbb{Z}_q^{d \times \mathfrak{d}^{k-j-1}}, (\mathbf{a}_i)_{i \in [\mathfrak{l}]}, (\varphi_{\iota,i})_{\iota \in [j], i \in [\mathfrak{l}]}), (\bar{\mathbf{s}}, \bar{u}_1, \ldots, \bar{u}_k, \bar{w}_1, \ldots, \bar{w}_{j+1})\right) \in R_{j+1}$$

where $\bar{\mathbf{s}}_1 = \bar{\mathbf{s}} \parallel \bar{u}_1 \parallel \cdots \parallel \bar{u}_k \parallel \bar{v}_{k+1}$ and $\bar{\mathbf{m}} := (\bar{w}_1, \ldots, \bar{w}_j)$. We assume that $\mathcal{E}^*$ does not solve the MSIS problem since if it did, then so does $\mathcal{A}$ (and later on $\mathcal{E}$). Clearly, by induction hypothesis, the probability that $\mathcal{A}$ succeeds for random ρ and $(\vec{\varphi}_{j,i})$ is at least

$$\epsilon - (k-j-1)q_1^{-\mathfrak{l}} - 2|\mathcal{C}|^{-1} - q_1^{-d/2} - q_1^{-\lambda} - 2^{-128}.$$

Moreover, the expected runtime $\mathcal{A}(\rho_P, \rho_E, (\vec{\varphi}_{j,i}))$ for any fixed $\rho_P, (\vec{\varphi}_{j,i})$ and $\rho_E \leftarrow \mathfrak{R}_E$ is $2^{k-j-1} \cdot 24T$.

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

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

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

- $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2) \neq (\bar{\mathbf{s}}'_1, \bar{\mathbf{s}}'_2)$ and both x, x' are relaxed openings of $(\mathbf{t}_A, (t_1, \ldots, t_j))$
- x is relaxed opening of $(\mathbf{t}_A, (t_1, \ldots, t_j))$ and

$$\left((T_j \in \mathbb{Z}_q^{d \times \mathfrak{d}^{k-j}}, (\mathbf{a}_i)_{i \in [\mathfrak{l}]}, (\varphi_{\iota,i})_{\iota \in [j-1], i \in [\mathfrak{l}]}), (\bar{\mathbf{s}}, \bar{u}_1, \ldots, \bar{u}_k, \bar{w}_1, \ldots, \bar{w}_j)\right) \in R_j.$$

In the first case we break the binding property of the commitment scheme and obtain a MSIS solution by Lemma 2.10. On the other hand, we extract the witness in the second case. Then, we have the following claims about $\mathcal{E}$.

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

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

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

□

This claim implies that the expected runtime of $\mathcal{E}$ is at most $2^{k-j} \cdot 24T$.

Claim. Probability that $\mathcal{E}$ succeeds is at least $\epsilon - (k-j)q_1^{-\mathfrak{l}} - 2|\mathcal{C}|^{-1} - q_1^{-d/2} - q_1^{-\lambda} - 2^{-128}$.

One proves the statement similarly as e.g. [LNP22, Theorem 4.5]. The key idea here is that if

$$T_j(\vec{v}_{j+1} \otimes \cdots \otimes \vec{v}_{k+1}) \neq \vec{w}_j$$

then only with probability at most $q_1^{-\mathfrak{l}}$ we have

$$\langle T_j(\vec{v}_{j+1} \otimes \cdots \otimes \vec{v}_k) - \vec{w}_j, \vec{\varphi}'_{j,i} \rangle = 0 \quad \text{for } i = 1, 2, \ldots, \mathfrak{l}$$

for random challenges $\vec{\varphi}'_{j,i} \leftarrow \mathbb{Z}_q^d$. Further, we know these $\mathfrak{l}$ equations hold by construction of the matrix T_{j+1} and the definition of the relation R_{j+1}. Hence, $\mathcal{E}$ succeeds with probability at most the difference of $\mathcal{A}$ succeeding and $q_1^{-\mathfrak{l}}$ which is exactly the term in the claim.

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

By applying the same approach to the protocol Π_{oom} we obtain the following result.

Theorem B.3. *Consider the commit-and-prove protocol in Fig. 9. Let $\gamma_1, \gamma_2, \gamma_3 > 0$ and $\lambda, \nu \in \mathbb{N}$. Define χ to be the uniformly distribution on S_ν. Fix*

$$\mathfrak{s}_1 = \gamma_1 \eta \sqrt{\mathsf{m}d + k + 1}, \quad \mathfrak{s}_2 = \gamma_2 \nu \eta \sqrt{m_2 d}, \quad \mathfrak{s}_3 = \sqrt{337(\mathsf{m}d + k + 1)}$$

where η is defined in Section 2.5.

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

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

For commit-and-prove simulatability, there exists a simulator $\mathcal{S}$ that, without access to private information $\mathbf{s}_1, \mathbf{m}$, outputs a simulation of the commitment

$$\mathbf{t}_A = \mathbf{A}_1 \begin{bmatrix} \mathbf{s} \\ u_1 \\ \vdots \\ u_k \\ v_{k+1} \end{bmatrix} + \mathbf{A}_2 \mathbf{s}_2$$

along with a non-aborting transcript of the protocol between prover $\mathcal{P}$ and verifier $\mathcal{V}$ such that for every algorithm $\mathcal{A}$ that has advantage ε in distinguishing the simulated commitment and transcript from the real commitment and transcript, whenever the prover does not abort, there is an algorithm $\mathcal{A}'$ with the same running time that has advantage ε in distinguishing the Extended-$\mathsf{MLWE}_{n+k+\lambda+256/d+2, m_2-n-k-\lambda-256/d-2, \chi, \mathcal{C}, D_{\mathfrak{s}_2}}$.

For soundness, let $B_i := 2\mathfrak{s}_i\sqrt{2m_i d}$ for $i = 1, 2$ where $m_1 := \mathsf{m} + k + 1$ and $\mathcal{B} := 2\sqrt{\frac{256}{26}}\varrho\mathfrak{s}_3$. If

$$\begin{aligned} &q \geqslant 41 \cdot (\mathsf{m} + k + 1)\, d \cdot \mathcal{B}, && \text{to use [LNP22, Lemma 2.9]} \\ &q > \mathcal{B}_{\mathsf{arp}}^2 + \mathcal{B}\sqrt{(\mathsf{m} + k + 1)d}, && \text{to prove } \mathbf{s}, u_1, \ldots, u_k, v_{k+1} \text{ have binary coeff.} \end{aligned}$$

then there is an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a probabilistic prover $\mathcal{P}^*$, which convinces $\mathcal{V}$ with probability $\varepsilon \geqslant 2/|\mathcal{C}| + kq_1^{-\mathfrak{l}} + q_1^{-d/2} + q_1^{-\lambda} + 2^{-128}$, extractor $\mathcal{E}$ with probability at least $\varepsilon - 2/|\mathcal{C}| - kq_1^{-\mathfrak{l}} - q_1^{-d/2} - q_1^{-\lambda} - 2^{-128}$ either outputs $\mathbf{x} := ((\bar{\mathbf{s}}, \bar{u}_1, \ldots, \bar{u}_k, \bar{v}_{k+1}), \bar{\mathbf{s}}_2) \in \mathcal{R}_q^{m_1+m_2+j}$ and $\bar{c} \in \bar{\mathcal{C}}$ such that $(\mathbf{x}, \bar{c})$ is a relaxed opening of $\mathbf{t}_A$ and

$$((T, A), (\bar{\mathbf{s}}, \bar{u}_1, \ldots, \bar{u}_k, \bar{v}_{k+1})) \in R_{\mathsf{oom}}$$

or a $\mathsf{MSIS}_{n, m_1+m_2, 4\eta\sqrt{B_1^2+B_2^2}}$ solution for $\begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix}$ in expected time at most $2^k \cdot 24T$ where running $\mathcal{P}^*$ once is assumed to take at most T time.

Security in the Random Oracle Model. As in prior works, we make the protocol non-interactive using the well-known Fiat-Shamir transformation [FS86]. However, as noted by e.g. Attema et al. [AFK21], applying a naive analysis in our case might incur a security loss in the order of Q^k where Q is the number of random oracle queries made by an adversary. Recently, Lyubashevsky et al. [LNP22] showed that the LNP framework admits a security loss of at most $Q + 1$ in the random oracle model. We note that the techniques from [LNP22, Appendix B] (especially the *probabilistic argument* which was implicitly used in the proof of Lemma B.2) can be directly adapted to our setting and thus our protocol also enjoys the linear security loss in the number of random oracle queries Q.

Proof size. Using the analysis from [LNP22, Section 6.1], the commitment and proof size of Π_{oom} (without doing any Dilithium compression) is around

$$\begin{aligned}&(n+k+\lambda+256/d+2)\,d\lceil\log q\rceil+(\mathsf{m}+k+1)d\cdot(2.57+\lceil\log \mathfrak{s}_1\rceil)\\&+m_2 d\ \cdot(2.57+\lceil\log \mathfrak{s}_2\rceil)+256\cdot(2.57+\lceil\log \mathfrak{s}_3\rceil)+\lceil\log(2\kappa+1)\rceil\cdot d \text{ bits.}\end{aligned}$$

In our ring signature size calculation, however, we also apply the commitment compression technique as described in [LNP22, Appendix A].

C Ring Signatures from One-Out-of-Many Proofs

C.1 Sign-invariant Relations and Bimodal Gaussians

In order to reduce the proof size of the one-out-of-many proof presented in Section 4, we make use of the techniques developed in Section 3. To this end, we need to modify the relation R_{oom} to be sign-invariant. In the current form, it is not sign-invariant since e.g. $\mathbf{s}\in\{0,1\}^{\mathsf{m}d}$ does not imply that $-\mathbf{s}\in\{0,1\}^{\mathsf{m}d}$. We make the following changes to R_{oom}. First, define a set of signed monomials:

$$\mathcal{B}:=\{X^i : i\in[2d]\}.$$

Then, we can define the modified relation R'_{oom} as follows:

$$R'_{\mathsf{oom}}:=\left\{\begin{array}{c}((T,A)(\mathbf{s},u_1,\ldots,u_k,v_{k+1})) : \mathbf{s}\in\{-1,1\}^{\mathsf{m}d}\wedge T(\vec{v}_1\otimes\cdots\otimes\vec{v}_{k+1})=A\vec{s}\\ \wedge u_1,\ldots,u_k,v_{k+1}\in\mathcal{B}\text{ where }\vec{u}_i:=\vec{v}_i\parallel 0^{d-\mathfrak{d}}\end{array}\right\}.$$

We claim that for certain values of k, R'_{oom} is sign-invariant.

Lemma C.1. *Let $k\geqslant 0$ be an even number. Then, relation R'_{oom} defined above is sign-invariant.*

Proof. Suppose $((T,A),(\mathbf{s},u_1,\ldots,u_k,v_{k+1}))\in R'_{\mathsf{oom}}$. First, we observe $-\mathbf{s}\in\{-1,1\}^{\mathsf{m}d}$ and $-u_1,\ldots,-u_k,-v_{k+1}\in\mathcal{B}$. Also, $-\vec{u}_i=-\vec{v}_i\parallel 0^{d-\mathfrak{d}}$ for $i\in[k]$. Finally, since k is even, we have

$$A(-\vec{s})=-T(\vec{v}_1\otimes\cdots\otimes\vec{v}_{k+1})=T\left((-\vec{v}_1)\otimes\cdots\otimes(-\vec{v}_{k+1})\right).$$

Hence, $((T,A)(-\mathbf{s},-u_1,\ldots,-u_k,-v_{k+1}))\in R'_{\mathsf{oom}}$. □

Constructing a commit-and-prove protocol Π'_{oom} for R'_{oom} can be done similarly as in the case for R_{oom} with two small changes. First, we prove that $\mathbf{s}\in\{-1,1\}^{\mathsf{m}d}$ which is equivalent to proving that vector $2^{-1}(\mathbf{s}+\mathbf{1})$ has binary coefficients. Secondly, to prove that $u_1,\ldots,u_k,v_{k+1}\in\mathcal{B}$ we use the observation that $a\in\mathcal{B}$ if and only if $\|a\|=1$ and this statement can be proven as in [LNP22].

Finally, we note that finding $\mathbf{s}_1:=(\mathbf{s},u_1,\ldots,u_k,v_{k+1})$ so that $((T,A),\mathbf{s}_1)\in R'_{\mathsf{oom}}$ implies knowing an index $i\in[N]$ of a column of T and a short vector $\vec{s}$ such that $A\vec{s}\in\{-\vec{t}_i,\vec{t}_i\}$ and thus a pre-image of $\vec{t}_i$ under A.

C.2 Ring Signature Construction

We sketch out the folklore approach to transform an one-out-of-many proof into a ring signature [ESS+19, GK15, BCC+15, LNS21b]. Suppose we have a ring of N users. Each user $i\in[N]$ has their associated private-public key $(\mathsf{sk}_i,\mathsf{pk}_i)$ such that $\mathsf{sk}_i:=\mathbf{s}^{(i)}\leftarrow\{-1,1\}^{\mathsf{m}d}$ and $\mathsf{pk}_i:=\mathbf{A}\mathbf{s}^{(i)}\bmod p$ where $\mathbf{A}\leftarrow\mathcal{R}_p^{\mathsf{n}\times\mathsf{m}}$ and p is a modulus for the ring signature.

Now, user i signs a message by producing a non-interactive one-out-of-many proof, i.e. proof of knowledge of a vector $\mathbf{s}^{(i)}$ such that $\mathbf{s}^{(i)}\in\{-1,1\}^{\mathsf{m}d}$ and

$$\mathbf{A}\mathbf{s}^{(i)}\in\{\mathsf{pk}_1,\ldots,\mathsf{pk}_N\}.$$

N	k	proof size
2^6	0	13.1 KB
2^{12}	2	14.1 KB
2^{18}	4	15.1 KB
2^{24}	6	16.2 KB

Fig. 10: Ring signature sizes for $N = d \cdot \mathfrak{d}^k$ users. For all parameter sets, we choose $(p, \mathsf{n}, \mathsf{m}, d, \mathfrak{d}) = (65437, 1, 12, 64, 8)$ and $q = 65437 \cdot 65629 \approx 2^{32}$.

We observe that if p divides q then this problem can be solved using the (non-interactive) commit-and-prove system Π'_{oom} for relation R'_{oom}.

Anonymity property of the ring signature follows directly from simulatability of Π'_{oom}. In order to argue unforgeability with respect to insider collusion, we proceed as in [ESS+19, Theorem 3]. Namely, the reduction picks a uniformly random user j and sets a uniformly random public key $\mathsf{pk}_j \leftarrow \mathcal{R}_p^{\mathsf{n}}$ (under the $\mathsf{MLWE}_{\mathsf{n},\mathsf{m}-\mathsf{n},\mathfrak{D}}$ assumption where $\mathfrak{D}$ is the distribution over $\mathcal{R}_p$ so that each coefficient is sampled uniformly at random from $\{-1, 1\}$). If there is any signing query to j, then the reduction simulates the one-out-of-many proof. Finally, the reduction will hope that: (i) the adversary does not make a corruption query to j and (ii) it forges a signature exactly for the public key pk_j. In this case, one can extract a secret key $\mathbf{s}^* \in \{-1, 1\}^{\mathsf{m}d}$ such that $\mathbf{A}\mathbf{s}^* = \mathsf{pk}_j$. Thus, $(\mathbf{s}^*, 1)$ is a non-zero vector of norm $\sqrt{\mathsf{m}d + 1}$ which is a Module-SIS solution for the matrix $[\mathbf{A} \mid \mathsf{pk}_j] \in \mathcal{R}_p^{\mathsf{n} \times (\mathsf{m}+1)}$ and thus the reduction solves $\mathsf{MSIS}_{\mathsf{n},\mathsf{m}+1,\sqrt{\mathsf{m}d+1}}$.

In Figure 10, we present ring signature sizes for various rings of size between 2^6 and 2^{24}. We set $(p, \mathsf{n}, \mathsf{m}) = (65437, 1, 12)$ so that both the $\mathsf{MLWE}_{\mathsf{n},\mathsf{m}-\mathsf{n},\mathfrak{D}}$ and the $\mathsf{MSIS}_{\mathsf{n},\mathsf{m}+1,\sqrt{\mathsf{m}d+1}}$ problems are hard. Namely, since there is a reduction loss of $1/N$, we pick the root Hermite factor $\delta \approx 1.0039$ for $\mathsf{MSIS}_{\mathsf{n},\mathsf{m}+1,\sqrt{\mathsf{m}d+1}}$ which should be enough for rings of size at most 2^{24}. In regard to $\mathsf{MLWE}_{\mathsf{n},\mathsf{m}-\mathsf{n},\mathfrak{D}}$, we aim for the root Hermite factor $\delta \approx 1.0044$ as in prior works. For such parameters, the user public key (resp. secret key) has size 128B (resp. 96B) which is more than one order of magnitude smaller than the public key in [LNS21b]. As described earlier, we pick $(d, \mathfrak{d}, \mathfrak{l}) = (64, 8, 16)$. Since we only consider even values for k in order to use bimodal Gaussians, our ring signature supports rings of size $64^{k/2+1} = 2^{3k+6}$. In all instantiations we picked $q_1 := p = 65437$ and $q_2 := 65629$ such that the proof system modulus $q = q_1 q_2 \approx 2^{32}$ and the repetition rate is ≈ 3 as in [LNS21b].

D Proving Integer Relations

This section focuses on proving integer relations using the framework described in Section 2.7. We start by proving integer addition in Section D.1 and then move to proving multiplication in Section D.2. We highlight that the relations we are interested in hold over integers, i.e. no wrap-around modulo q occurs. We will use the standard bit decomposition representation. Namely, an integer $w \in [0, 2^{\mathsf{n}} - 1]$ is represented as a vector of n bits $(w_0, \ldots, w_{\mathsf{n}-1}) \in \{0, 1\}^{\mathsf{n}}$ such that

$$w = \sum_{i=0}^{\mathsf{n}-1} w_i 2^i.$$

Our techniques can be easily adapted for the two's complement representation which includes negative integers.

In this section we will reduce the problem of proving n-bit[15] integer relations to proving various equations over a polynomial ring $\mathcal{R}' := \mathbb{Z}[X]/(X^{\mathsf{n}} + 1)$ (or $\mathbb{Z}[X]/(X^{2\mathsf{n}} + 1)$ in the case of multiplication). Since Section 2.7 only considers relations over the ring $\mathcal{R}$ which might be potentially smaller than $\mathcal{R}'$ if $d < \mathsf{n}$, it is not immediately clear if the framework from [LNP22] supports proving relations over a larger ring $\mathcal{R}'$. However,

[15] Suppose that n is a power-of-two.

it was shown by Lyubashevsky et al. [LNPS21] that one can write an equation over $\mathcal{R}'$ equivalently as a system of equations over $\mathcal{R}$.

Let $\mathsf{n} = kd$ for $k \geqslant 1$. First, we observe that $\mathcal{R}$ is isomorphic to the subring $S := \mathbb{Z}[X^k]/(X^{kd}+1)$ of $\mathcal{R}'$. Let us define the commutative ring $S^k = (S^k, +, \star)$ where $+$ is a component-wise addition and $\star$ is defined as:

$$(a_0, \ldots, a_{k-1}) \star (b_0, \ldots, b_{k-1}) = (c_0, \ldots, c_{k-1})$$

where for all $0 \leqslant \ell < k$

$$c_\ell := \sum_{\substack{0 \leqslant i,j < k \\ i+j \equiv \ell \bmod k}} a_i b_j X^{\lfloor \frac{i+j}{k} \rfloor k} \in S.$$

Thus, $(0, \ldots, 0)$ and $(1, 0, \ldots, 0)$ are the additive and multiplicative identities respectively. Then, [LNPS21] prove the following lemma.

Lemma D.1. *Let $k \geqslant 1$ be a power-of-two. Then, $\mathcal{R}' := \mathbb{Z}[X]/(X^{kd}+1) \cong S^k$.*

We explicitly write the ring isomorphism ϕ. First of all, we can write any polynomial $a \in \mathcal{R}'$ uniquely as $a = \sum_{i=0}^{k-1} a_i X^i$ where each $a_i \in S$. We define the map $\phi : \mathcal{R}' \to S^k$ as

$$\phi(a) := (a_0, \ldots, a_{k-1}) \in S^k.$$

We refer to [LNPS21, Section 2.8] for more details.

D.1 Integer Addition

In this subsection we provide an efficient commit-and-prove system for addition on the committed integers. Specifically, given commitments to integers a, b, c (depending on the application, some of these values can be given out in the clear), we want to prove that $a + b = c$. In order to consider both positive and negative values, we use the two's complement representation. Namely, let n be a power of two and suppose $\mathsf{n} = kd$ for $k \geqslant 1$. Suppose $a, b, c \in [0, 2^{\mathsf{n}} - 1]$ and we want to prove $a + b = c$. Then, a, b, c can be represented as vectors $\vec{a}, \vec{b}, \vec{c} \in \{0,1\}^{\mathsf{n}}$ which satisfy

$$a = \sum_{i=0}^{\mathsf{n}-1} a_i 2^i, \quad b = \sum_{i=0}^{\mathsf{n}-1} b_i 2^i, \quad c = \sum_{i=0}^{\mathsf{n}-1} c_i 2^i.$$

Let us define polynomials $\hat{a}, \hat{b}, \hat{c} \in \mathbb{Z}[X]$ as follows:

$$\hat{a} = \sum_{i=0}^{\mathsf{n}-1} a_i X^i, \quad \hat{b} = \sum_{i=0}^{\mathsf{n}-1} b_i X^i, \quad \hat{c} = \sum_{i=0}^{\mathsf{n}-1} c_i X^i.$$

Then, clearly we have $a + b = c$ if and only if $\hat{a}(2) + \hat{b}(2) = \hat{c}(2)$. The latter can be written equivalently as

$$\hat{a}(X) + \hat{b}(X) = \hat{c}(X) + (2 - X)\hat{f}(X) \tag{36}$$

for some $\hat{f} \in \mathbb{Z}[X]$ of degree at most $\mathsf{n} - 2$. We will call $\hat{f}$ the carry polynomial. We now show that $\hat{f}$ has binary coefficients.

Lemma D.2. *The polynomial $\hat{f} \in \mathbb{Z}[X]$ defined above has coefficients in $\{0, 1\}$.*

Proof. We prove the statement by induction and start with the constant coefficient f_0. Note that

$$2f_0 = a_0 + b_0 - c_0$$

and thus

$$-\frac{1}{2} \leqslant f_0 = \frac{a_0 + b_0 - c_0}{2} \leqslant 1.$$

Hence, $f_0 \in \{0,1\}$. Next, consider $0 < i \leqslant \mathsf{n}-2$ and suppose $f_{i-1} \in \{0,1\}$. Then

$$2f_i - f_{i-1} = a_i + b_i - c_i$$

and therefore

$$-\frac{1}{2} \leqslant f_i = \frac{a_i + b_i - c_i + f_{i-1}}{2} \leqslant \frac{3}{2}.$$

We conclude that $f_i \in \{0,1\}$. □

Our strategy will be to prove (36). We do it by first proving the equation over $\mathcal{R}'_q := \mathbb{Z}_q[X]/(X^{\mathsf{n}}+1) = \mathbb{Z}_q[X]/(X^{kd}+1)$ and then showing that no modulo q and $X^{\mathsf{n}}+1$ wrap-around occurs. Let $\hat{x} \in \mathcal{R}'_q$ be an inverse of $2-X$. Such inverse exists if $2^{kd}+1$ is not divisible by q which will be the case in our instantiations. Consider the $\phi : \mathcal{R}'_q \to \mathcal{R}_q^k$ map described above, i.e.

$$\phi(u) = (u_0, \ldots, u_{k-1}) \text{ where } u = \sum_{i=0}^{k-1} u_i(X^k)X^i \in \mathcal{R}'_q.$$

Using Lemma D.1, (36) is equivalent to

$$\phi(\hat{x}) \star (\phi(\hat{a}) + \phi(\hat{b}) - \phi(\hat{c})) = \phi(\hat{f}).$$

For simplicity denote

$$\phi(\hat{a}) := (\hat{a}_0, \ldots, \hat{a}_{k-1})$$

and similarly for $\hat{b}, \hat{c}, \hat{x}, \hat{f}$. Then this equation is equivalent to

$$\forall \iota \in \mathbb{Z}_k, \sum_{\substack{0 \leqslant i,j < k \\ i+j \equiv \iota \bmod k}} \hat{x}_i \left(\hat{a}_j + \hat{b}_j - \hat{c}_j\right) X^{\lfloor \frac{i+j}{k} \rfloor} = \hat{f}_\iota$$

over $\mathcal{R}_q$. Hence, we will commit to $\phi(\hat{a}), \phi(\hat{b}), \phi(\hat{c}) \in \mathcal{R}_q^k$ and prove the following statements:

Polynomials $\hat{a}, \hat{b}$ and $\hat{c}$ are well-formed. We need to show that all the coefficients of $\hat{a}, \hat{b}, \hat{c}$ are binary. Note that this is equivalent to proving that $\hat{a}_0, \ldots, \hat{a}_{k-1} \in \mathcal{R}_q$ all have binary coefficients and similarly for $\hat{b}, \hat{c}$. This is thus a statement of Type-3 in Section 2.7.

Polynomial $\hat{f}$ is well-formed. We prove that $\hat{f}$ has binary coefficients. This is done by proving that for all $\iota \in \mathbb{Z}_k$,

$$\sum_{\substack{0 \leqslant i,j < k \\ i+j \equiv \iota \bmod k}} \hat{x}_i \left(\hat{a}_j + \hat{b}_j - \hat{c}_j\right) X^{\lfloor \frac{i+j}{k} \rfloor} \in \mathcal{R}_q$$

has binary coefficients. Note that the expression is a linear combination in the committed messages and thus this is a statement of Type-3 in Section 2.7.

No overflow modulo q and $X^{\mathsf{n}}+1$. Recall that we prove Equation 36 over $\mathcal{R}'_q$. In order to conclude that the equation holds over integers, we prove that there is no overflow modulo q and $X^{\mathsf{n}}+1$. The first statements above make sure no wrap-around modulo q occurs when $q \geqslant 7$. For the latter issue, note that it is enough to prove that the highest degree coefficient of $\hat{f}$ is equal to zero. This is done by proving that the constant coefficient of

$$X^{-d+1} \cdot \hat{f}_{k-1} = X^{-d+1} \cdot \sum_{\substack{0 \leqslant i,j < k \\ i+j \equiv k-1 \bmod k}} \hat{x}_i \left(\hat{a}_j + \hat{b}_j - \hat{c}_j\right) X^{\lfloor \frac{i+j}{k} \rfloor}$$

is equal to zero. Clearly, this is a statement of Type-2 in Section 2.7.

We present the proof sizes for proving integer addition in Figure 11. For computing the sizes we already apply the bimodal Gaussian optimisation described in Section 3. In particular, we additionally commit to the sign $b \in \{-1, 1\}$ in order to perform bimodal Gaussian rejection sampling on the message. For each instance, we choose $(q, d) = (\approx 2^{32}, 64)$ and set the standard deviations so that the overall repetition rate is at most 3.

n	k	proof size
64	1	11.0KB
128	2	11.4KB
512	8	13.6KB

Fig. 11: Proof size comparison for proving integer addition $a + b = c$ for $a, b, c \in [0, 2^{\mathsf{n}} - 1]$.

D.2 Integer Multiplication

We show how to prove knowledge of integers a, b, c such that $ab = c$. We first present a non-optimal solution which can be done by directly applying the framework in Section 2.7. Then, we describe a way to reduce the proof size at the cost of slightly extending the framework from [LNP22].

Concretely, let us write $a, b \in [0, 2^{\mathsf{n}} - 1]$ and $c \in [0, 2^{2\mathsf{n}} - 1]$ in binary representation, i.e.

$$a = \sum_{i=0}^{\mathsf{n}-1} a_i 2^i, \quad b = \sum_{i=0}^{\mathsf{n}-1} b_i 2^i, \quad c = \sum_{i=0}^{2\mathsf{n}-1} c_i 2^i.$$

We assume that n is a power of two and $2\mathsf{n} = kd$ for $k \geqslant 2$. Now, define

$$\hat{a}(X) = a_0 + a_1 X + \cdots + a_{\mathsf{n}-1} X^{\mathsf{n}-1} \in \mathbb{Z}[X]$$

and similarly for $\hat{b}, \hat{c} \in \mathbb{Z}[X]$. Now, observe that $\hat{a}(2)\hat{b}(2) - \hat{c}(2) = 0$. Hence, there exists a "carry" polynomial $\hat{f}$ of degree at most $2(\mathsf{n} - 1)$ which satisfies:

$$\hat{a}(X)\hat{b}(X) - \hat{c}(X) = (2 - X)\hat{f}(X). \tag{37}$$

The next lemma states that coefficients of f are between $-(\mathsf{n} + 1)$ and $\mathsf{n} + 1$.

Lemma D.3. *Let $\hat{f}$ be the polynomial of degree at most $2\mathsf{n} - 2$ defined above. Then, for each coefficient f_k of $\hat{f}$ corresponding to X^k, $0 \leqslant f_k \leqslant \mathsf{n}$.*

Proof. We first show $f_0 = 0$. Consider Equation 37 for $X = 0$. Then, we have $a_0 b_0 - c_0 = 2f_0$. Since $-1 \leqslant a_0 b_0 - c_0 \leqslant 1$, we get $f_0 = 0$.

Now suppose by induction that $0 \leqslant f_{k-1} \leqslant \mathsf{n}$ for $k > 0$. In general, by considering the k-th coefficient of $\hat{a}\hat{b} - \hat{c}$ and $(2 - X)\hat{f}$, we have the following equality:

$$2f_k - f_{k-1} = \sum_{0 \leqslant i,j < \mathsf{n} \text{ s.t. } i+j=k} a_i b_j - c_k \leqslant \mathsf{n}.$$

and similarly $-1 \leqslant 2f_k - f_{k-1}$. Hence, we obtain

$$-\frac{1}{2} \leqslant \frac{f_{k-1} - 1}{2} \leqslant f_k \leqslant \frac{\mathsf{n} + f_{k-1}}{2} \leqslant \mathsf{n}.$$

Thus, the statement holds.

Unlike in Lemma D.2, the coefficients of $\hat{f}$ are much bigger than $\{0,1\}$ but still small compared to q (if q is much larger than $\mathsf{n}+1$ which will be the case). However, in order to show that no modulo q overflow occurs, we just need to prove shortness of $\hat{f}$ approximately.

Similarly as in the integer addition proof, we want to prove Equation 37 over $\mathbb{Z}[X]$. In order to do so, we consider this equation over $\mathcal{R}'_q := \mathbb{Z}_q[X]/(X^{2\mathsf{n}}+1) = \mathbb{Z}_q[X]/(X^{kd}+1)$. Namely, consider the $\phi : \mathcal{R}'_q \to \mathcal{R}_q^k$ map described earlier, i.e.

$$\phi(u) = (u_0, \ldots, u_{k-1}) \text{ where } u = \sum_{i=0}^{k-1} u_i(X^k)X^i.$$

As shown in Lemma D.1, (37) over $\mathcal{R}'_q$ is equivalent to

$$\phi(\hat{a}) \star \phi(\hat{b}) - \phi(\hat{c}) = \phi(2-X) \star \phi(\hat{f}).$$

For simplicity denote

$$\phi(\hat{a}) := (\hat{a}_0, \ldots, \hat{a}_{k-1}) \in \mathcal{R}_q^k$$

and similarly for $\hat{b}, \hat{c}, \hat{f}$. Also denote $\phi(2-X) := (\hat{x}_0, \ldots, \hat{x}_{k-1})$. Then this equation is equivalent to

$$\forall \iota \in \mathbb{Z}_k, \quad \sum_{\substack{0 \leqslant i,j < k \\ i+j \equiv \iota \bmod k}} \hat{a}_i \hat{b}_j X^{\lfloor \frac{i+j}{k} \rfloor} - \hat{c}_\iota = \sum_{\substack{0 \leqslant i,j < k \\ i+j \equiv \iota \bmod k}} \hat{x}_i \hat{f}_j X^{\lfloor \frac{i+j}{k} \rfloor}. \tag{38}$$

Now, in order to conclude that (37) holds over $\mathbb{Z}[X]$, we need to show that no wrap-around modulo q and $X^{2\mathsf{n}}+1$ occurs. For the first issue, we show that coefficients of $\hat{a}, \hat{b}$ and $\hat{c}$ are binary (by definition of the binary decomposition). As for $\hat{f}$, we conduct an approximate shortness proof to show that $\hat{f}$ has sufficiently small coefficients so that no modulo q overflow happens. Next, in order to make sure there is no wrap-around modulo $X^{2\mathsf{n}}+1$, we prove that the degree of $\hat{a}$ and $\hat{b}$ are at most $\mathsf{n}-1$ and the degree of $\hat{f}$ is at most $2\mathsf{n}-2$.

Hence, we will commit to $\phi(\hat{a}), \phi(\hat{b}), \phi(\hat{c}), \phi(\hat{f}) \in \mathcal{R}_q^k$ and prove the following statements:

Polynomials $\hat{a}, \hat{b}$ are well-formed. We need to show that all the coefficients of $\hat{a}, \hat{b}$ are binary and that the n-th,$\ldots,(2\mathsf{n}-1)$-th coefficients of $\hat{a}, \hat{b}$ are equal to zero. These statements are to make sure no wrap-around modulo q and $X^{2\mathsf{n}}+1$ occur respectively. Note that the first one is equivalent to proving that $\hat{a}_0, \ldots, \hat{a}_{k-1}$ all have binary coefficients and similarly for $\hat{b}$ (Statement of Type-3). The latter one, on the other hand, is equivalent to proving that the $d/2$-th,$\ldots,(d-1)$-th coefficients of $\hat{a}_0, \ldots, \hat{a}_{k-1}, \hat{b}_0, \ldots, \hat{b}_{k-1}$ are all zeroes, i.e. the constant coefficients of

$$X^{-i-d/2} \cdot \hat{a}_j \quad \text{and} \quad X^{-i-d/2} \cdot \hat{b}_j$$

are zeroes for $i \in \mathbb{Z}_{d/2}$ and $j \in \mathbb{Z}_k$ (Statement of Type-2).

Polynomial $\hat{c}$ is well-formed. In case of $\hat{c}$, we need to prove that $\hat{c}$ has binary coefficients. This boils down to proving that $\hat{c}_0, \ldots, \hat{c}_{k-2}, \hat{c}_{k-1} + X^{d-1}$ all have binary coefficients (Statement of Type-3).

Equation 37 holds over $\mathcal{R}'_q$. We simply prove k quadratic equations (38) (Statement of Type-1).

No overflow modulo q. We prove approximately that $\hat{f}$ has small coefficients. By Lemma D.3, $\|\phi(\hat{f})\| \leqslant \mathcal{B}' := \mathsf{n}\sqrt{2\mathsf{n}} = kd\sqrt{kd}/2$ (Statement of Type-5). We can convince the verifier that $\|\hat{f}\|_\infty = \|\phi(\hat{f})\|_\infty \leqslant \psi \cdot \mathcal{B}'$ for some approximation factor. If

$$q > \mathsf{n} + 3\psi \cdot \mathcal{B}' \tag{39}$$

and we proved that that $\hat{a}, \hat{b}, \hat{c}$ all have binary coefficients and the degree of $\hat{a}, \hat{b}$ are at most $\mathsf{n}-1$, then (37) holds over $\mathbb{Z}$ and no wrap-around modulo q occurs.

No overflow modulo $X^{2\mathsf{n}}+1$. Recall that the first statement above makes sure no wrap-around modulo $X^{2\mathsf{n}}+1$ occurs when multiplying $\hat{a}\hat{b}$. Now, to prove no such wrap-around happens when multiplying $(2-X)\hat{f}$, it is enough to prove that the highest degree coefficient of $\hat{f}$ is equal to zero. This is done by proving that the constant coefficient of $X^{-d+1} \cdot \hat{f}_{k-1}$ is equal to zero (Statement of Type-3).

It is now clear that all the statements can be directly proven using the framework in Section 2.7. Namely, we define $\mathbf{s}_1 := \phi(\hat{a}) \parallel \phi(\hat{b}) \parallel \phi(\hat{c})$ and $\mathbf{m} = \phi(\hat{f})$ and apply the general protocol from [LNP22]. The reason to set $\mathbf{m}$ this way is because the coefficients of $\hat{f}$ are much larger than the coefficients of $\hat{a}, \hat{b}, \hat{c}$.

Is Committing to the Carry Polynomials Necessary? A natural question one might ask is why we have to commit to the "carry polynomials" $\phi(\hat{f})$ in the integer multiplication case but not when doing integer addition as in the previous subsection. What is similar in both cases is that if we write $\mathbf{s}_1 := \phi(\hat{a}) \parallel \phi(\hat{b}) \parallel \phi(\hat{c}), \mathbf{m} := \varnothing$ then there are known polynomial functions $F_1, \ldots, F_k : \mathcal{R}_q^{3k} \to \mathcal{R}_q$ such that:

$$\phi(\hat{f}) = \begin{bmatrix} F_1(\mathbf{s}_1, \mathbf{m}) \\ \vdots \\ F_k(\mathbf{s}_1, \mathbf{m}) \end{bmatrix}.$$

Now, note that the framework from [LNP22] natively *only* supports proving shortness in the L_∞/L_2 norm of *linear* functions in $\mathbf{s}_1, \mathbf{m}$ (see Statements of Type-4 and Type-5 in Section 2.7). The reason is that when applying approximate range proofs, they additionally commit to signs $b^{(d)}, b^{(e)}$ in order to use bimodal Gaussian rejection sampling. Having these additional secret polynomials, e.g. $b^{(d)}$, turns a linear equation into a quadratic one. Observe that for integer addition $F_1, \ldots, F_k$ were indeed linear. However, for integer multiplication $F_1, \ldots, F_k$ become quadratic and thus the framework from [LNP22] cannot be used directly since the equation becomes cubic.

We circumvent this problem and still not commit to $\phi(\hat{f})$ by simply removing the bimodal Gaussian rejection sampling when proving approximate shortness of $\phi(\hat{f})$. Concretely, we do not commit to $b^{(d)}$ from [LNP22, Fig. 10] and thus we can prove shortness of a quadratic expression in $\mathbf{s}_1, \mathbf{m}$. The drawback is a potentially larger standard deviation $\mathfrak{s}^{(d)}$ used for the Statement of Type-5 due to the standard rejection sampling, i.e. Lemma 2.8, and consequently larger approximation factor ψ. However, this is fine in our setting as long as (39) still holds.

We include the optimised proof sizes in Figure 12. As in the case for integer addition, we apply the bimodal rejection sampling optimisation from Section 3, including committing to the additional sign b. For each instantiation, we set $(q, d) = (\approx 2^{32}, 64)$ and the parameters for rejection sampling are chosen so that the repetition rate is less than 5 (the increase comes from the fact that we perform one standard rejection sampling as described above).

n	k	proof size
64	2	13.6KB
128	4	14.4KB
512	16	19.2KB

Fig. 12: Proof size comparison for proving integer multiplication $ab = c$ for $a, b \in [0, 2^{\mathsf{n}} - 1]$ and $c \in [0, 2^{2\mathsf{n}} - 1]$.

To conclude this section, we present a comparison of our proof sizes with [LNS20] for integer addition and multiplication in Fig. 13.

N	128	512
[LNS20]	25KB	45KB
This Work	12KB	14KB

N	128	512
[LNS20]	40KB	100KB
This Work	15KB	19KB

Fig. 13: Proof size comparison for proving integer addition (on the left) and multiplication (on the right). Here, N is the bit-length of the integers. It is worth mentioning that [EZS+19, ESZ21] also construct efficient proofs of integer addition, alternatively called *balance proofs*, which use similar CRT-packing techniques as [LNS20].

E Group Signatures Based on Falcon

Another application of one-out-of-many proofs are group signatures. Indeed, they can be (naively) constructed by simply making the group public key to be the concatenation of all the user public keys, e.g. as in [EZS+19, ESZ21]. Consequently, the group public key becomes linear in the number of users. As an example, by taking the parameters from C.2, the group public key for 2^{20} users would become 128MB which is infeasible in practice.

In this section we construct a Falcon-based [FHK+20] group signature which offers significantly smaller group public key and still makes use of the nice properties of our one-out-of-many proof. We highlight that the signature generation and verification is still linear in the number of users which might become a serious limitation for large groups.

The high-level idea is as follows. Let us define $N = d \cdot \mathfrak{d}^k < 2^d$ to be size of the group and $\mathcal{R}'_p := \mathbb{Z}_p[X]/(X^{d'}+1)$ to be the underlying ring. The group public key is a polynomial $h \in \mathcal{R}'_p$ and the master secret key is a quadruple of short polynomials $(g, f, G, F) \in \mathcal{R}'^4_p$ such that: (i) $h = g \cdot f^{-1} \bmod (p, X^{d'}+1)$ and (ii) $fG - gF = p \bmod (X^{d'}+1)$. Then, the secret key of the user $i \in [N]$ is the Falcon signature of the message i, i.e. a pair of short polynomials $\mathsf{sk}_i := \left(s_1^{(i)}, s_2^{(i)}\right) \in \mathcal{R}'^2_p$ which satisfies

$$s_1^{(i)} + h s_2^{(i)} = H(i) \quad \text{and} \quad \left\| \left(s_1^{(i)}, s_2^{(i)}\right) \right\| \leqslant \beta$$

where $H : \{0,1\}^* \to \mathcal{R}'_p$ is a hash function and β is a suitably chosen bound on the secret key. Further, user $i \in [N]$ signs a message m by giving a proof of knowledge of a pair $(s_1, s_2) \in \mathcal{R}'_p$ such that $\|(s_1, s_2)\| \leqslant \beta$ and

$$s_1 + h s_2 \in \{H(j) : j \in [N]\}\,.$$

This is where we apply the one-out-of-many proof developed in Section 4. Concretely, denote $\vec{s} := \vec{s}_1 \parallel \vec{s}_2 \in \mathbb{Z}_p^{2n}$ and $A := [I_n \ \mathsf{rot}(h)]$ where $\mathsf{rot}(h) \in \mathbb{Z}_p^{n \times n}$ is the rotational matrix of h. Then, we can write the equation above equivalently as:

$$A\vec{s} = T(\vec{v}_1 \otimes \cdots \otimes \vec{v}_k \otimes \vec{v}_{k+1}) \pmod p \tag{40}$$

where T is the matrix with columns being the coefficients of $H(1), \ldots, H(N)$, and $\vec{v}_1, \ldots, \vec{v}_k \in \{0,1\}^{\mathfrak{d}}$ and $\vec{v}_k \in \{0,1\}^d$ are the unit vectors.

We have the following two remarks. First, if we pick a proof system modulus q to be divisible by the prime p then (40) can be written equivalently as an equation over $\mathbb{Z}_q$ by simply multiplying A and T by q/p. However, to fully apply the framework from [LNP22] we need all prime divisors of q to be congruent to 5 modulo 8 and thus we cannot use the original prime $p = 12289$ from Falcon. Hence, we set $p = 12301$ which is a prime congruent to 5 modulo 8 and is sufficiently close to the original prime so that other parameters from Falcon do not change [16]. Secondly, recall that in Section 4 base $\mathfrak{d}$ is defined to be a divisor of d. Then, $q_1^{-\mathfrak{l}}$ becomes one of the knowledge error terms where q_1 is the smallest prime divisor of q and $\mathfrak{l} = d/\mathfrak{d}$. When

[16] It is worth pointing out that having a different prime might not allow for fast Fourier sampling used in Falcon. If one is interested in sticking with the original Falcon prime then one could solve this issue using the modulo switching technique as in [LNP22, Section 6.3]. This approach becomes more expensive since one would need much larger proof system modulus q to make sure no wrap-around occurs.

$q_1 = p \approx 2^{13}$ and $d = 64$ as before, then picking $\mathfrak{d} = 8$ does not provide sufficient soundness, while with $\mathfrak{d} = 4$ one would need to commit to more polynomials u_i and w_i from Section 4 which increases the proof size. We point out that by simple bookkeeping, one can follow (almost) the same argument from Section 4 while not having $\mathfrak{d}$ divide d as long as $\mathfrak{d} \cdot \mathfrak{l} \leqslant d$. In our example, we will pick $\mathfrak{d} = 7$ and $\mathfrak{l} = 9$ so that $q_1^{-\mathfrak{l}} = p^{-\mathfrak{l}} \approx 2^{-122}$.

In order to be able to open the group signature, we will add a verifiable encryption to the signature exactly as in [LNP22, Section 6.3]. Namely, we want the signer to encrypt an identity of the user $i \in [N]$, using a public key associated to a decryption key that the group manager possesses, and prove that this encryption is indeed of their identity. A naive solution would be to encrypt all vectors $\vec{v}_1, \ldots, \vec{v}_{k+1}$ (or more concretely, polynomials $u_1, \ldots, u_k, v_{k+1}$) which is rather costly. What we do instead is encrypting a single polynomial $m^{(i)} \in \mathcal{R}$ for which its coefficient vector is a binary decomposition of the following number:

$$0 \leqslant \sum_{j=0}^{k-1} \mathfrak{d}^j \cdot \vec{d}^T \vec{u}_{j+1} + \mathfrak{d}^k \cdot \vec{d}^T \vec{v}_{k+1} \leqslant \mathfrak{d}^k d - 1$$

where $\vec{u}_j = \vec{v}_j \parallel 0^{d-\mathfrak{d}}$ for $j \in [k]$ and $\vec{d} = (0, 1, 2, \ldots, d-2, d-1) \in \mathbb{Z}^d$. It is easy to see that for two different users $i \neq j$ we have $m^{(i)} \neq m^{(j)}$. Now, to encrypt $m := m^{(i)}$ we sample a randomness vector $\mathbf{r} \leftarrow \xi^K$, where ξ is a distribution over $\mathcal{R}$, and compute

$$\begin{bmatrix} \boldsymbol{t}_0 \\ t_1 \end{bmatrix} := \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ \lfloor \frac{p}{2} \rceil m \end{bmatrix} \tag{41}$$

over $\mathcal{R}_p$ where $(\mathbf{A}, \mathbf{b}) \in \mathcal{R}_p^{M \times K} \times \mathcal{R}_p^M$ is the public key. Let B be an upper-bound on $\mathbf{r}$ such that the probability that $\|\mathbf{r}\| > B$ for $\mathbf{r} \leftarrow \xi^K$ is negligible. Then, in the verifiable encryption scenario, we want to prove knowledge of $\mathbf{r} \in \mathcal{R}_p^K$ and $m \in \mathcal{R}_p$ such that (i) Equation 41 is satisfied over $\mathcal{R}_p$, (ii) $\|\mathbf{r}\| \leqslant B$ and (iii) $m \in \{0, 1\}^d$. Finally, to show the relationship between m and polynomials $u_1, \ldots, u_k, v_{k+1}$, we simply prove the linear equation over $\mathbb{Z}_p$:

$$\sum_{j=0}^{k-1} \mathfrak{d}^j \cdot \vec{d}^T \vec{u}_{j+1} + \mathfrak{d}^k \cdot \vec{d}^T \vec{v}_{k+1} = \left[1 \; 2 \; \cdots \; 2^{\lfloor \log M \rfloor}\right] \vec{m}.$$

Since we assumed p is divisible by q, all these relations can be easily proven using the framework described in Section 2.7[17].

In terms of security analysis, anonymity follows directly from the simulatability of Π_{oom} and the framework from [LNP22]. Further, one can reduce the traceability property directly to the unforgeability of the Falcon signature scheme. Concretely, an adversary $\mathcal{A}$ against the traceability game first asks two types of queries: (i) the signature queries and (ii) the corruption queries where the user secret keys are revealed. For (i) the reduction will simply simulate the proofs. Then, for (ii), if $\mathcal{A}$ wants to obtain a secret key corresponding to the user i, the reduction makes a query to the Falcon signing oracle on message i and passes the valid secret key sk_i. As a standard complexity leveraging argument, suppose that at the beginning the challenger picks a uniformly random index i^* for which it aborts when $\mathcal{A}$ makes a corruption query on i^*. Finally, if $\mathcal{A}$ can forge a valid group signature on an uncorrupted user j *and* $j = i^*$, then the reduction can extract short $\bar{s}_1, \bar{s}_2$ such that $\bar{s}_1 + h\bar{s}_2 = H(j) = H(i^*)$ which is a valid Falcon signature forgery on the message i^* that was *not* queried to the Falcon signing oracle.

Concrete instantiation. We present the group signature sizes in Fig. 14. As described above, we pick the prime modulus $p := 12301$ so that we can apply the framework from [LNP22]. We select other parameters exactly as in Falcon [FHK+20], namely $d' = 512$ and $\beta = \sqrt{34034726}$. For verifiable encryption we set $M = 10$ and $K = 21$ and ξ samples each coefficient according to the binomial distribution $\mathcal{B}$ with standard deviation 1: $\mathcal{B}(-2) = \xi(2) = 1/16, \mathcal{B}(-1) = \mathcal{B}(1) = 1/4$ and $\mathcal{B}(0) = 3/8$.

[17] In particular, we do not have to commit to the whole vector $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2) \in \mathcal{R}_p^{K-M} \times \mathcal{R}_p^M$ but only to the $\mathbf{r}_1 \in \mathcal{R}_p^{K-M}$ part as described in [LNP22, Fig.14].

N	k	proof size
2^6	0	16.4 KB
$\approx 2^{11}$	2	17.4 KB
$\approx 2^{22}$	6	19.4 KB

Fig. 14: Group signature sizes for $N = d \cdot \mathfrak{d}^k$ users. For all parameter sets, we choose $(p, d, \mathfrak{d}, \mathfrak{l}) = (12301, 64, 7, 9)$ and $q = 12301 \cdot 349133 \approx 2^{32}$.

In all instantiations we picked $q_1 := p = 12301$ and $q_2 := 349133$ such that the proof system modulus $q = q_1 q_2 \approx 2^{32}$ and the repetition rate is ≈ 3. Using the computation from Falcon, the public key (resp. secret key) size is $(897 + Kd\lceil\log(12301)\rceil)/2^{10} \approx 3.2$KB (resp. $(666 + 3Md)/2^{10} \approx 2.5$KB).