Rejection-Free Framework of Zero-Knowledge Proof Based on Hint-MLWE

Antoine Douteau and Adeline Roux-Langlois

Université Caen Normandie, ENSICAEN, CNRS, Normandie Univ, GREYC UMR6072, F-14000 Caen, France
antoine.douteau@unicaen.fr, adeline.roux-langlois@cnrs.fr

Abstract. Commit-and-prove zero-knowledge proofs are generalized versions of zero-knowledge protocols that permit proving relations over the committed elements in addition testifying the knowledge of the initial message. For example, the existing framework (LNP, Crypto22) allow a user to prove that the secret element committed satisfies quadratic relations with bounded norm (ℓ_2 or ℓ_∞). Security of these frameworks, regarding the zero knowledge property, is mainly assumed by the use of rejection sampling introduced by Lyubashevsky (Asiacrypt09). The main problems with rejection sampling are non-constant execution time and the cost of protecting this step from side-channel attacks.
Our contribution is a new framework of proof for zero-knowledge property that proves knowledge and quadratic relations over lattices without basing the security over rejection sampling. The security of our framework is based on the recent Hint-MLWE (KLSS, Crypto23) assumption. This variant of MLWE gives additional hints about the secret in addition to the original input, and is shown to be as hard as its associated MLWE instance when secrets follow discrete Gaussian distributions.

Keywords: Lattices, Zero-Knowledge Protocol, Commit-and-prove, Hint-MLWE, Rejection sampling

1 Introduction

Lattices have been a well-studied and cryptanalysed mathematical tools that have made it possible to build several optimised, efficient and resistant cryptographic constructions for current standards, as well as for future advanced primitives. Originally, fundamental lattice assumptions such as the Closest Vector Problem or the Shortest Vector Problem are NP-hard [Ajt98, GMSS99] but the mathematical aspect of lattices inspired relaxed assumptions more cryptographic-oriented, known as the Learning with Errors (LWE) problem [Reg05] and its dual Short Integer Solution (SIS) from [Ajt96]. They reduce to these problems and also benefit from being "average-case" unlike the fundamental assumptions. In a global way in cryptography, lack of structure and redundancy can have a negative impact on efficiency and on having reasonable schemes in both size and running time. This explains why lattice-based constructions are restricted to

Module Lattices, originally defined in [LS15] that work over polynomials. This special structure is used on ML-DSA and ML-KEM [DKL+18, BDK+18] as ML stands for Module Lattices, over their assumptions Module-LWE and Module-SIS, variants of LWE and SIS.

Signatures over Lattices. The first lattice-based framework of signature scheme [GPV08] exploits a private trapdoor, allowing the signer to invert a one-way function, usually called Hash-and-Sign. The second main framework relies on the Fiat-Shamir paradigm [FS87] which performs a transformation of a zero-knowledge protocol to a non-interactive instance, secured on the Random Oracle Model (ROM). The idea is to prove knowledge of a specific element, and being zero-knowledge means that the execution of the protocol does not give any additional information on the secret element. The FSp transposes this proof in a single execution without interactivity between a prover and a verifier. The transcript generated defines the signature; the final verification steps are identical.

Rejection Sampling. One of the initial construction over lattices was considered by Lyubashevsky in [Lyu09] under the FS with aborts idea: it allows the prover to send various elements strongly related to the secret committed into the transcript. To ensure the zero-knowledge property, those elements need to be carefully chosen after passing a rejection step successfully. The method is known as rejection sampling, as those temporary elements need to be resampled upon failure. The main advantage is that it allows the prover to control the distribution of the leaks. Since the proving phase aborts with a non-negligible probability, it does not run in constant time. This method is used in several constructions over lattices, isogenies and error-correcting codes [DG19, BKP20, LNP22, BDK+23, DKL+18].

Commit-and-Prove System of Proofs. In order to be useful in practice and permit more than just proving the knowledge of a secret element, the notion of zero-knowledge commit-and-prove proofs [CLOS02] extends the definition of zero-knowledge proofs and allows proving additional information and relations about a committed element. This system of proofs can be modified regarding the Fiat-Shamir (with aborts) paradigm in order to construct efficient advanced signature schemes. The recent framework [LNP22] generalizes the construction of this protocol in order to additionnaly prove polynomial relations over the secret and norm bounds of those structure. This enables, for instance, to prove knowledge of an MSIS element or an MLWE secret. The framework also comes with different applications (verifiable encryption, group signature) and is used (a posteriori) in various constructions [LN22, AKSY22, BLNS23, NS24] to take advantage of its versatility. The overall security of the commitment scheme relies on module assumptions, and the zero-knowledge property is established using different optimized rejection sampling mechanisms [Lyu09, DDLL13, LNS21].

Constraints of Rejection Sampling. In general, rejection sampling is theoretically proven secure, but several flaws have been discovered in [EFGT17, DFPS23, BBD+23], affecting most of the actual rejection-based constructions over both lattices and isogenies constructions. A major disadvantage is that the proving phase does not run in constant time, complicating both implementation and security, as in [GHJ+22], which tackles earlier proposals for error-correcting code systems HQC and BIKE [AAB+22, ABB+22]. Those attacks are called *side-channel attacks* and enable an attacker the ability to extract information by analyzing the physical device running the algorithm.

One of the main methods for protecting against side-channel attacks is to use secret-sharing also called *masking* [ISW03, PR13]. This countermeasure implies a decomposition of an element in different shares. The masking countermeasure must be applied to all functions and variables that involve secret elements, including rejection sampling. Masking rejection sampling has been studied in depth in [MGTF19, BBE+18, EFGT17, CGL+24], as the leakage of execution information from the proving phase would conceptually contradict the goal of rejection sampling. Despite it has been optimized, masking is particularly expensive: in the recent paper [CGL+24] that analyze the second level of security of ML-DSA, the action of masking an execution multiplies the running time by at least 33 (for the first level of security). In a further comparison, the masked rejection sampling step alone takes 3× longer than the total duration of a single (rejection-free) unmasked instance of ML-DSA.

Rejection sampling can also be an issue regarding some constructions with multiple executions of a protocol, such as for the Polynomial Commitment from [HSS24] where its rejection rate would increase exponentially. Then, another perspective would be to completely remove the use of rejection sampling without significantly affecting the overall size of the construction in which it is used.

1.1 Contribution

Our contribution is to build a new lattice-based commit-and-prove framework for zero-knowledge proofs, without performing rejection sampling, based on existing constructions from [LNP22, HSS24]. The security of our framework is based on the Hint-MLWE assumption, introduced in [KLSS23]. This MLWE variant provides additional hints about the secret, and is shown to be as hard as its associated MLWE instance when secret elements follow discrete Gaussian distributions.

Our key idea is to exploit this new assumption in the [LNP22] framework over the new Modified-Ajtai commitment scheme from [HSS24], as it only focused on constructing a polynomial commitment using the Hint-MLWE assumption, we generalize their idea in order to construct and implement a concrete commit-and-prove framework of proof. In a first step, we modify the framework of [LNP22] with the special Gaussianisation process from [HSS24], allowing the proof of knowledge of a secret vector of polynomials $\mathbf{s}$ without any restriction on their bounds, unlike the original [LNP22] framework. In a second step, our

framework gives to the prover the ability to prove multiple quadratic and linear relations over both $\mathcal{R}_p$ and $\mathbb{Z}_p$ exploiting the automorphism σ_{-1} without additionally committing to it. We also provide the implementation of our modified framework based on the recent LaZer Library [LSS24]. As the structure is relatively close to the original paper [LNP22], the code structure complies with LaZer standards and has been implemented in such a way that it can be used in addition to the original library. As the main contribution lies in the elimination of rejection sampling, which implies a modification of the execution times of the different functions, we add a concrete comparison between instances of [LNP22] using [LSS24] and our implementation on similar parameter sets in Table 3. We provide a static analysis of the complexity, which shows that an instance of the proving phase is at most 4 times longer than a single [LNP22], resulting in faster executions, as the rejection rate increases, but slower (less than 4 times) for other phases. It also affects the size of the transcripts, which are ≈ 3 times larger. Our construction provides an alternative of [LNP22] that runs in constant time. The code is available in the attached file  and in this repository.

Size of the transcripts (in kilobytes KB)						
Protocol	Protocol/Phase	Param. 1	Param. 2	Param. 3	Param. 4	Param. 5
[LNP22]	**Opening proof**	13.22	20.77	30.53	×	41.05
	+ 3 Quad.	13.93	21.86	31.31	×	41.68
	+ 3 Eval.	15.16	22.96	32.09	×	42.93
Ours	**Opening proof**	24.23	45.69	103.19	37.33	96.52
	+ 3 Quad.	24.83	46.76	103.95	38.08	97.12
	+ 3 Eval.	25.42	47.82	104.70	38.83	97.72
Mean times (in million cycles)						
[LNP22]	**KeyGen**	7.63	10.32	19.32	×	39.35
	Commit	7.29	20.89	31.33	×	39.42
	Prove	324.22	352.22	4289.68	×	4462.29
	Verify	18.54	39.17	1744.10	×	1629.12
Ours	**KeyGen**	19.01	47.04	66.30	29.55	62.01
	Commit	19.86	86.54	102.47	45.73	67.15
	Prove	59.45	134.26	4248.67	220.98	3961.24
	Verify	41.74	96.79	3897.43	180.31	3624.34

Table 1: The first part of the Table explains the size of the transcripts in kilobytes. In the second part, the results show our benchmarks for the mean running time of each phase. The 5 different sets of parameters are detailed in Table 2. In the comparison, the running time of the Prove part of [LNP22] using the [LSS24] implementation takes into account several executions due to rejection sampling.

1.2 Technical Overview

The initial idea of our construction is to provide a rejection-free framework of zero-knowledge proofs based on lattice assumptions. We start from [LNP22] based on the Ajtai commitment scheme on a module lattice $\mathcal{R}_p = \mathbb{Z}_p[X]/\langle \phi_d \rangle$ with its modulo p and a power of 2 degree: d. The original construction proves

knowledge of the message $\mathbf{s}_1 \in \mathcal{R}_p^{m_1}$, along with a randomness $\mathbf{s}_2 \in \mathcal{R}_p^{m_2}$ by computing the commitment $\mathbf{t_A} = \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2$ for uniformly sampled matrices $\mathbf{A}_1 \in \mathcal{R}_p^{n\times m_1}$ and $\mathbf{A}_2 \in \mathcal{R}_p^{n\times m_2}$. Considering an Hermite Normal Form of the matrix $\mathbf{A}_2$ i.e. with an identity structure such that $\mathbf{A}_2 = \left[\mathbf{I}_n \mid \mathbf{A}'_2\right]$ generates the same lattice with a faster sampling, inducing a MLWE structure. This commitment scheme is relevant only if the norm of $\mathbf{s}_i$ is bounded, as its binding security is based on the MSIS assumption. The hiding security is provided assuming a hard instance of MLWE with the second component involving the secret randomness $\mathbf{s}_2$. The opening of the Ajtai commitment, coming from [Lyu12], needs to provide to the verifier the responses of the form $c\mathbf{s}_i + \mathbf{y}_i$ for a challenge c. Those elements can leak information on $\mathbf{s}_i$ which is avoided using rejection sampling. We want to modify the commitment scheme in order to prevent the use of rejection sampling by having non-leaking responses. The recent assumption Hint-MLWE [KLSS23] is aimed at resolving this problem, saying that the responses (no matter their distribution) do not leak any information about $\mathbf{s}_i$, in the specific case where $\mathbf{s}_i$ follows a Gaussian distribution. This was used in [HSS24] where the authors construct the Modified-Ajtai commitment scheme with an opening satisfying the zero-knowledge property over Hint-MLWE. In order to prove knowledge of an element $\mathbf{s}_1 \in \mathcal{R}_p^{m_1}$, they simply redefined the Ajtai commitment for a Gaussian encoded element $\tilde{\mathbf{s}}_1 \in \mathcal{R}_p^{km_1}$ (linked to $\mathbf{s}_1$) by transmitting $\mathbf{t_A} = \tilde{\mathbf{A}}_1\tilde{\mathbf{s}}_1 + \mathbf{A}_2\mathbf{s}_2$. The overall security is also based on MSIS and MLWE, only the zero-knowledge property of the opening is modified from the original version by being shown over the hardness of Hint-MLWE instead of different instances of rejection sampling. This is done by the "Gaussianisation" process that encodes $\mathbf{s}_1$ in base b (i.e. $\mathbf{s}_1 = \sum_{i\in[k-1]} \hat{\mathbf{s}}_{1,i} b^i$). This process in done coordonate-wise (over the vector, and over the coefficients of the polynomials). Proceeding over the modulo p, the process can be made probabilistic as in [MP12, Theorem 4.1] and can be seen as $\hat{\mathbf{s}} = \mathbf{G}_{b,\sigma}^{-1}\mathbf{s} \mod p$ for a positive σ such that $\mathbf{G}_b\hat{\mathbf{s}} = \mathbf{s} \mod p$ (deterministic) by denoting the gadget vector $\mathbf{G}_b := \mathbf{I}_{m_1} \otimes \mathbf{g}_{b,k}^\top := \mathbf{I}_{m_1} \otimes \left[1\ b \cdots b^{k-1}\right]$.

In order to provide a general framework, the authors of [LNP22] showed that the Ajtai commitment can be launched with additional BDLOP commitments from [BDL+18] of the form $\mathbf{t_B} = \mathbf{B}\mathbf{s}_2 + \mathbf{m}$ for unbounded messages $\mathbf{m}$ with the same randomness $\mathbf{s}_2$. This commitment is called ABDLOP. As the commitment size is linear in the size of the message, the BDLOP part was intended to be used carefully. Another important point of the Modified-Ajtai commitment scheme is that the committed encodings are not bounded. In our case, we adapt this construction and join to it the same BDLOP construction forming the Rejection-Free ABDLOP commitment scheme. With the Gaussianisation, we commit to elements only by the Ajtai part; the (expensive) BDLOP part will only be used dynamically (used to commit elements during the execution), as both norm bounded and unbounded elements can be initially committed using the (rejection-free) Ajtai part of the commitment scheme. As in the original ABDLOP, the overall opening process is restricted to the one the Ajtai part by computing $\mathbf{m} = \mathbf{t_B} - \mathbf{B}\mathbf{s}_2$. We also adapt the original opening process for a different challenge space $\mathcal{C}$. This involves enlarging some parameters and slightly modifying the security proofs.

Proving only the knowledge of an element $\mathbf{s}$ is not optimal. Commit-and-prove protocol such as the framework [LNP22] additionally allows proving that a secret satisfies a bunch of relations. The prover can ensure quadratic relations over $\mathbf{s}$ but also over an automorphism $\sigma(\mathbf{s})$ (without additionally committing to it). In a second step, they also provide the ability to the prover to prove norm bounds of these relations. Unlike [LNP22], we do not commit the initial message but a randomized encoded version of it, therefore we must modify the framework in consequence and show new adapted proofs of completeness. In [LNP22], they used the fact that a quadratic relation regarding a message $\mathbf{s}$ can be viewed as the result of the function $f(\mathbf{s}) = \mathbf{s}^\top \mathbf{R}_2 \mathbf{s} + \mathbf{r}_1^\top \mathbf{s} + r_0$. We adapt it to a new relation $\tilde{f}$ over the randomized encoding $\tilde{\mathbf{s}}$ such that $\tilde{f}(\tilde{\mathbf{s}}) = 0$ implies that $f(\mathbf{s}) = 0$. As we have the relation $\mathbf{s} = \left[\mathbf{I}_n \otimes \mathbf{g}_{b,k}^\top\right] \cdot \tilde{\mathbf{s}}$ over $\mathcal{R}_p$, we obtain the equality $\tilde{f}(\tilde{\mathbf{s}}) = \tilde{\mathbf{s}}^\top \cdot \left(\mathbf{R}_2 \otimes \left[\mathbf{g}_{b,k}^\top \otimes \mathbf{g}_{b,k}\right]\right) \cdot \tilde{\mathbf{s}} + \left[\mathbf{r}_1 \otimes \mathbf{g}_{b,k}^\top\right] \cdot \tilde{\mathbf{s}} + r_0 = f(\mathbf{s}) \mod p$.

In [LNP22], their protocol allowed relations over $\mathbf{s}$ and any $\mathcal{R}$-automorphism of it with the only the commitment of $\mathbf{s}$. In our case we only consider the conjugate automorphism $\sigma_{-1} : X \longmapsto X^{-1}$, which has an important impact. The relevance is simple and comes from an easy tool from [LNP22, Lemma 2.4]; the constant coefficient of the trace application $\mathsf{tr} : \sum_{i\in[n]} \sigma_{-1}(\mathbf{a}_i) \cdot \mathbf{b}_i$ is equal to $\langle \mathbf{a}, \mathbf{b} \rangle \in \mathbb{Z}_p$ for $\mathbf{a}, \mathbf{b} \in \mathcal{R}^n$. Then, it permits to prove relations over $\mathbb{Z}_p$ and not just over $\mathcal{R}_p$ by using the same mechanism as in [LNP22] that allows to prove that a relation has a zero constant coefficient. In our protocol, as for the quadratic relations, this is implied from the encoding protocol as we have $\sigma_{-1}(\tilde{\mathbf{s}}) = \widetilde{\sigma_{-1}(\mathbf{s})}$. Putting everything together, we can launch a protocol of proof of knowledge with quadratic relations over $\mathcal{R}_p$ and/or $\mathbb{Z}_p$, exploiting [LNP22, Section 4].

1.3 Limitations

The original version of the protocol permits to prove relations over the secret and one of its automorphisms. In our case, we only specify the case of σ_{-1} because we exploit an inherent structure of this automorphism. A generalization regardless of the automorphism may be useful but will involve modifying the Gaussianisation method and inevitably our entire construction. Along with the first limitation, intrinsically dependent on the Gaussianisation method, the protocol imposes a proof environment $\mathcal{R}_p$ where $p = b^k + 1$ a prime modulus. Together with our protocol, our challenge space restricts the choice of k to the specific case of $k = 2$. Finding a new challenge space that does not restrict ourselves to this choice of k together with the automorphism invariant condition $c = \sigma_{-1}(c)$ is an interesting open question. As the bounds of the security assumptions necessary depend on the choices of b (and k), it restricts the choice of p to be a prime higher than $\approx 2^{50}$. The last limitations is that we are not able to prove approximate norm bounds of relations over the secrets as it was build in the [LNP22] framework. At least, it is still possible to apply the [LNP22, Section 5] strategy by recomputing the secret from the encoding and perform rejection sampling. It would be interesting to prove those relations using the pros of our protocol in order to obtain a fully rejection-free proof.

1.4 Related works

As already mentioned, [LNP22] constructs a generalized framework for proving knowledge of an element $\mathbf{s}$ such that $f(\mathbf{s}) = 0$ for every $f \in \mathcal{F}$. They prove the zero-knowledge property assuming various instances of rejection sampling, without considering potential side-channels attacks neither the cost of protecting it with masking. In our generated framework, we provide a rejection-free alternative of [LNP22] inspired by techniques from [HSS24]. In this paper, the authors construct a polynomial commitment allowing a prover to commit to a polynomial f and convincing a prover that $y = f(x)$ for public y and x without revealing f. They based their construction on a homomorphic decoding protocol. In our work, we exploit their techniques, but in order to commit to an element x and convince a prover that $f(x) = 0$ for a public relation f without revealing x.

We identify in the literature different alternatives for removing the rejection sampling [BCM21, DFPS22, DPS23]. In the first article, the authors propose an alternative to the Lyubashevsky Signature Scheme [Lyu09] by restricting the construction to deal with both secret keys and masks being distributed uniformly over a hypercube. They also conjecture that all the rejection sampling steps do not have the same impact and that it may be difficult to remove abortion steps while maintaining the correctness of the protocol. In the other papers, the authors exploits Gaussian convolution distributions: by choosing compensating covariance matrices, they manage to construct hints that do not leak information even without rejection sampling. These papers only focused on constructing a rejection-free signature scheme and not a commit-and-prove framework.

2 Preliminaries

2.1 Notations and Definitions

Notations For $n \in \mathbb{N}$, we define $[n] := \{0, \dots, n-1\}$. Lower-case v letters mean elements of $\mathcal{R}$, $\vec{v}$ are the same elements viewed as vectors (of size d) of coefficients. Bold lower-case $\mathbf{v}$ letters denote column-vectors with coefficients being elements of $\mathcal{R}$, and bold upper-case $\mathbf{A}$ letters for matrices with coefficients in $\mathcal{R}$. We write $x \xleftarrow{\$} S$ when x is uniformly sampled from a finite set S.

Number Theory. We denote $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$, $\mathcal{K} := \mathbb{Q}[X]/\langle \phi_d(X) \rangle$ where $\phi_d(X)$ denotes the d-cyclotomic polynomial (we restrict the choice of d being a power of 2), and $\mathcal{R}$ its ring of integers $\mathcal{R} := \mathbb{Z}[X]/\langle \phi_d(X) \rangle$. Also we denote $\mathcal{R}_q := \mathcal{R}/q\mathcal{R}$. We specify the notation of the automorphism $\sigma := \sigma_{-1} \in \mathsf{Aut}(\mathcal{R}) : X \mapsto X^{-1}$.

Norms. For a vector of polynomials $\mathbf{f} = (f_0, \dots, f_{n-1}) \in \mathcal{R}^n$, the norm is the norm of each coefficient: $\|\mathbf{f}\|_q := \sqrt[q]{\sum_{i \in [n]} \|f_i\|_q^q}$, same for the infinite norm: $\|\mathbf{f}\|_\infty := \max_{i \in [n]} \|f_i\|_\infty$. For $\mathcal{M} \in \mathbb{C}^{n \times m}$, $\|\mathcal{M}\| := s_{\mathsf{max}}(\mathbf{A}) = \sqrt{|\lambda_{\mathsf{max}}(\mathcal{M}^\top \mathcal{M})|}$.

Gadgets. We remind the construction of the "gadget" vector and its standard definition for an integer b: $\mathbf{g}_{b,k}^{\mathsf{T}} = \left[1, b, \cdots, b^{k-1}\right] \in \mathcal{R}^k$. We extend the definition over n elements names as the gadget matrix $\mathbf{G}_{b,k,n} := \mathbf{I}_n \otimes \mathbf{g}_{b,k}^{\mathsf{T}} \in \mathcal{R}^{n \times nk}$. We denote a probabilistic "inverse" $\mathbf{G}_{b,\sigma}^{-1}$ allowing to generate a non-unique preimage $\hat{\mathbf{t}}$ of $\mathbf{t}$ such that $\mathbf{G}\hat{\mathbf{t}} = \mathbf{t}$ i.e $\hat{\mathbf{t}} \leftarrow \mathcal{D}_{\Lambda_{\mathbf{t}}^{\top}(\mathbf{G}),\sigma}$ as explained in [MP12, Theorem 4.1].

2.2 Lattices

Lattices. A lattice, denoted by Λ, is a discrete subgroup of $\mathbb{R}^n$ with additive operations. It can be defined by a basis $\mathcal{B} = (\vec{b}_i)_{i \in [m]} \in \mathbb{R}^{n \times m}$ such that it is denoted by $\Lambda = \left\{ \sum_{i \in [m]} x_i \cdot \vec{b}_i \;\middle|\; \vec{x} \in \mathbb{Z}^m \right\}$, where m is its dimension. We defined its dual lattice $\Lambda^* := \{x \mid \langle x, z \rangle \in \mathbb{Z} \; \forall z \in \Lambda\}$. We have $\Lambda = (\Lambda^*)^*$. We remind structured lattice using cyclotomic ring in Appendix B.1.

Probability Distributions. In our case, $\mathcal{D}_{\Lambda,\vec{c},\sigma}$ describe a discrete Gaussian distribution over Λ, centered in $\vec{c} \in \mathbb{R}^m$ with standard deviation σ such that for any $\vec{x} \in \mathbb{R}^n, \rho_{\vec{c},\sigma}(\vec{x}) := \exp\left(-\frac{\|\vec{x}-\vec{c}\|^2}{2\sigma^2}\right)$, and $\mathcal{D}_{\Lambda,\vec{c},\sigma}(\vec{x}) := \frac{\rho_{\vec{c},\sigma}(\vec{x})}{\rho_{\vec{c},\sigma}(\Lambda)}$ where $\rho_{\vec{c},\sigma}(\Lambda) = \sum_{x \in \Lambda} \rho_{\vec{c},\sigma}(\vec{x})$. We also define the sampling of a vector of an Elliptic Gaussian using an associated covariance matrix defined by a positive-definite matrix, denoted by $\mathbf{\Sigma} := \sqrt{\mathbf{\Sigma}}^{\top}\sqrt{\mathbf{\Sigma}}$. The probability is now defined by: $\rho_{\sqrt{\mathbf{\Sigma}}}(\vec{x}) = \exp\big(-\pi(\vec{x}-\vec{c})^{\mathsf{T}}\mathbf{\Sigma}^{-1}(\vec{x}-\vec{c})\big)$. If Λ is a structured lattice, we sample its vector of coefficients and embbed it as a polynomial of $\mathcal{R}$. We write directly polynomials.

Smoothing Parameter. We remind the definition of the smoothing parameter from [MR07, Definition 3.1] as $\eta_\varepsilon(\Lambda)$, the smallest standard deviation σ such that $\rho_{\frac{1}{\sigma}}(\Lambda^* \backslash 0) \leq \varepsilon$. The definition has been extended for covariance matrices in [Pei10, Definition 2.3] such that if $\eta_\varepsilon(\sqrt{\mathbf{\Sigma}}^{-1}\Lambda) \leq 1$ therefore $\sqrt{\mathbf{\Sigma}} \geq \eta_\varepsilon(\Lambda)$.

Lemma 2.1 ([KLSS23, Lemma 5]). *If $\|\mathbf{\Sigma}^{-1}\| \leq \frac{1}{\eta_\varepsilon(\Lambda)^2}$ then $\sqrt{\mathbf{\Sigma}} \geq \eta_\varepsilon(\Lambda)$.*

Lemma 2.2 ([MR07, Lemma 4.4]). *Let Λ a n-dimensional lattice, an arbitrary vector $\vec{c} \in \mathbb{R}^n$, $0 < \varepsilon < 1$ and $\sigma \geq \eta_\varepsilon(\Lambda)$. Let $\vec{z} \leftarrow \mathcal{D}_{\Lambda,\vec{c},\sigma}$. Then: $\mathbb{P}\left[\|\vec{z}-\vec{c}\| > \sigma\sqrt{n}\right] \leq \frac{1+\varepsilon}{1-\varepsilon} \cdot 2^{-n}$. If $0 < \varepsilon < \frac{1}{3}$ then: $\mathbb{P}\left[\|\vec{z}-\vec{c}\| > \sigma\sqrt{n}\right] \leq 2^{-n+1}$.*

In our construction, we will use [HSS24, Lemma 11] in the case of structured lattice with polynomials instead of vector of coefficients.

Lemma 2.3 (Structured version of [HSS24, Lemma 11]). *Let $\mathcal{R}_p^n$ a nd-dimensional lattice over the ring $\mathcal{R}$ and a modulo p. Let $\ell > 0$ an integer. Define $\mathbf{\Sigma} \in \mathbb{R}^{nd \times nd}$ a covariance matrix, associated to a secret sampling. Define ℓ covariance matrices $\mathbf{\Sigma}_i \in \mathbb{R}^{nd \times nd}$, generating $c_i \in \mathcal{R}_p$ and $C_i := \mathsf{Rot}_{nd}(c_i)$.*

If we suppose $\sqrt{\left(\mathbf{\Sigma}^{-1} + \sum_{i \in [\ell]} C_i^T \mathbf{\Sigma}_i^{-1} C_i\right)^{-1}} \geq \eta_\varepsilon(\mathcal{R}^n)$, then for any center $\vec{u} \in \mathbb{R}^{nd}$, then the following two distributions over $(\mathcal{R}^n)^\ell$ are at statistical 2ε:

$$\left\{(\mathbf{z}_1,\cdots,\mathbf{z}_\ell)\;\middle|\;\mathbf{s}\leftarrow\mathcal{D}_{\mathbf{u}+\mathcal{R}^n,\sqrt{\mathbf{\Sigma}}},\mathbf{y}_i\leftarrow\mathcal{D}_{-C_i\mathbf{u}+\mathcal{R}^n,\sqrt{\mathbf{\Sigma}_i}},\mathbf{z}_i=c_i\mathbf{s}+\mathbf{y}_i\right\},$$
$$\left\{(\mathbf{z}_1,\cdots,\mathbf{z}_\ell)\;\middle|\;\mathbf{s}'\leftarrow\mathcal{D}_{\mathcal{R}^n,\sqrt{\mathbf{\Sigma}}},\mathbf{y}_i\leftarrow\mathcal{D}_{\mathcal{R}^n,\sqrt{\mathbf{\Sigma}_i}},\mathbf{z}_i=c_i\mathbf{s}'+\mathbf{y}_i\right\}.$$

We use the simplified discrete convolution lemma, initially defined in [Pei10].

Lemma 2.4 (Discrete Convolution Lemma from [HSS24, Lemma 7]). *Let $\mathbf{\Sigma}_1,\mathbf{\Sigma}_2$ be positive-definite matrices. Let two n-dimensional lattices Λ_1 and Λ_2 such that $\Lambda_2\subseteq\Lambda_1$. If $\sqrt{\mathbf{\Sigma}_1}\geq\eta_\varepsilon(\Lambda_1)$ and $\mathbf{\Sigma}_3^{-1}:=\mathbf{\Sigma}_1^{-1}+\mathbf{\Sigma}_2^{-1}$ satisfies $\sqrt{\mathbf{\Sigma}_3}\geq\eta_\varepsilon(\Lambda_2)$ for $0<\varepsilon<\frac{1}{2}$. Then, for arbitrary $\vec{c}_1,\vec{c}_2\in\mathbb{R}^n$, the distribution of the convolution defined by $\mathcal{C}:=\{\mathbf{x}_1+\mathbf{x}_2\mid\mathbf{x}_1\leftarrow\mathcal{D}_{\Lambda_1,\vec{c}_1,\sqrt{\mathbf{\Sigma}_1}},\ \mathbf{x}_2\leftarrow\mathcal{D}_{\Lambda_2,\vec{c}_2,\sqrt{\mathbf{\Sigma}_2}}\}$ is within statistical distance 8ε of $\mathcal{D}_{\Lambda_1,\vec{c}_1+\vec{c}_2,\sqrt{\mathbf{\Sigma}_1+\mathbf{\Sigma}_2}}$.*

Assumptions. We remind the definitions of M(I)SIS and (Hint-)MLWE.

Definition 2.1 (M(I)SIS [LS15]). *Given $n>0$, $m(n),B(n)>0$ and a modulo $q(n)$ such that $B<q$. Consider a public and uniformly sampled matrix $\mathbf{A}\in\mathcal{R}_q^{n\times m}$, a target $\mathbf{t}\in\mathcal{R}_q^n$, the Module (Inhomogenous) Short Integer Solution $\mathit{MISIS}_{\mathcal{R},q,n,m,\beta}$ problem asks for a solution $\mathbf{x}\in\mathcal{R}_q^m$ with $0<\|\mathbf{x}\|\leq\beta$ that solves $\mathbf{Ax}=\mathbf{t}$ over $\mathcal{R}_q$. The $\mathit{MSIS}_{\mathcal{R},q,n,m,\beta}$ assumption is the special case where $\mathbf{t}=\mathbf{0}$.*

Definition 2.2 (MLWE [LS15, Definition 4.1]). *Given $n,\ell,m>0$ and a distribution χ over $\mathcal{R}^{m+n}$. Consider a random matrix $\mathbf{A}\xleftarrow{\$}\mathcal{R}_q^{m\times n}$, a secret $\mathbf{r}\leftarrow\chi$ parsed as $\left[\mathbf{s}^\top,\mathbf{e}^\top\right]$ for the MLWE sample. The $\mathit{MLWE}_{\mathcal{R},q,n,m,\ell,\chi}$ problem asks to distinguish between an honest $(\mathbf{A},\mathbf{As}+\mathbf{e})$ and $(\mathbf{A},\mathbf{b})$ for $\mathbf{b}\xleftarrow{\$}\mathcal{R}_q^m$.*

Definition 2.3 (Hint-MLWE [KLSS23, Definition 7]). *Given $n,\ell,m>0$ and distributions $\chi,\ \xi_1,\ldots,\xi_\ell$ over $\mathcal{R}^{m+n}$ and Γ is one over $\mathcal{R}$. Consider a random matrix $\mathbf{A}\xleftarrow{\$}\mathcal{R}_q^{m\times n}$, a secret $\mathbf{r}\leftarrow\chi$ parsed as $\left[\mathbf{s}^\top,\mathbf{e}^\top\right]$ for the MLWE sample. Consider ℓ hints as $\mathbf{z}_i=c_i\mathbf{r}+\mathbf{y}_i$, with $c_i\leftarrow\Gamma$ and $\mathbf{y}_i\leftarrow\xi_i$. The $\mathit{Hint\text{-}MLWE}_{\mathcal{R},q,n,m,\ell,\chi,\xi_1,\ldots,\xi_\ell,\Gamma}$ problem asks to distinguish between an honest $(\mathbf{A},\mathbf{As}+\mathbf{e},(c_i)_{i\in[\ell]},(\mathbf{z}_i)_{i\in[\ell]})$ and $(\mathbf{A},\mathbf{b},(c_i)_{i\in[\ell]},(\mathbf{z}_i)_{i\in[\ell]})$ for $\mathbf{b}\xleftarrow{\$}\mathcal{R}_q^m$.*

As shown above, the assumption is a relaxed alternative to the classical MLWE with ℓ linear hints over the secret element $\mathbf{r}$. The assumption was shown in [KLSS23] to be as hard as a MLWE instance in the restricted case where the element $\mathbf{r}$ and the masks $\mathbf{y}_i$ follow Gaussian distributions. We recall the hardness of Hint-MLWE from [ENP24] with Elliptic Gaussians distributions.

Theorem 2.1 (Hardness of Hint-MLWE [ENP24, Theorem 2]). *Let n,m,q and ℓ be positive integers and Γ a distribution over $\mathcal{R}$. We define upper bounds β_i such that $\beta_i^2\geq\|c_i\|_1^2>0$, where each c_i is sampled from Γ. For the covariance matrices $\mathbf{\Sigma}_1,\mathbf{\Sigma}_{2,1},\ldots,\mathbf{\Sigma}_{2,\ell}$, we define σ as $\frac{1}{\sigma^2}=2\left(s_1(\mathbf{\Sigma}_1^{-1})+\sum_{i\in[\ell]}\beta_i^2 s_1(\mathbf{\Sigma}_{2,i}^{-1})\right)$. If $\sigma\geq\sqrt{2}\cdot\eta_\varepsilon(\mathcal{R}^{n+m})$, for some $0<\varepsilon\leq\frac{1}{2}$, then there is an polynomial reduction from $\mathit{MLWE}_{\mathcal{R},q,n,m,\mathcal{D}_\sigma}$ to $\mathit{Hint\text{-}MLWE}_{\mathcal{R},q,n,m,\ell,\mathcal{D}_{\sqrt{\mathbf{\Sigma}_1}},\mathcal{D}_{\sqrt{\mathbf{\Sigma}_{2,1}}},\ldots,\mathcal{D}_{\sqrt{\mathbf{\Sigma}_{2,\ell}}},\Gamma}$ that reduces the advantage by at most 2ε.*

2.3 Ajtai and ABDLOP commitment schemes

Ajtai commitment scheme We remind the original Ajtai commitment scheme inspired from [Ajt96] and describe a Lyubashevsky-like proof of opening [Lyu12].

Parameters : Let $n \in \mathbb{N}$, $(\alpha_1, m_1) \in \mathbb{N}^2$ the message parameters and the others $m_2 \in \mathbb{N}, p \in \mathbb{N}, \sigma_2 \in \mathbb{R}^+$ are parametrised by n. $\mathcal{R}_p$ is a ring of degree d (a power of 2). By defining $\mathcal{M}_{\alpha,m} := \{\|\mathbf{s}\|_2 \leq \alpha \mid \mathbf{s} \in \mathcal{R}_p^m\}$, the message space is restricted to $\mathcal{M}_{\alpha_1,m_1}$ (with chosen α_1) and randomness to $\mathcal{M}_{\alpha_2,m_2}$ for $\alpha_2 := \sigma_2\sqrt{m_2 d}$.

- **Setup**$(1^\lambda) \to ck$: Given a security parameter λ, it generates a commitment key $ck = (\mathbf{A}_1, \mathbf{A}_2)$ where $\mathbf{A}_1 \xleftarrow{\$} \mathcal{R}_p^{n\times m_1}$ and $\mathbf{A}_2 \xleftarrow{\$} \mathcal{R}_{p,\mathsf{HNF}}^{n\times m_2}$.
- **Commit**$(ck, \mathbf{s}_1, \sigma) \to (\mathbf{t_A}, \delta)$: Given a message $\mathbf{s}_1 \in \mathcal{M}_{\alpha_1,m_1}$ and ck parsed as $(\mathbf{A}_1, \mathbf{A}_2)$. First, it samples $\mathbf{s}_2 \xleftarrow{\$} \mathcal{D}_{\mathcal{R}_p,\sigma_2}^{m_2}$ and verifies that $\mathbf{s}_2 \in \mathcal{M}_{\alpha_2,m_2}$. Then, it computes $\mathbf{t_A} = \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2$ for the opening $\delta = (\mathbf{s}_1, \mathbf{s}_2)$.

An honest commitment can be viewed as an element of $\mathsf{R}_{\mathsf{ajtai}}$:

$$\mathsf{R}_{\mathsf{ajtai}} := \left\{ \frac{(\mathbf{s}_1, \mathbf{s}_2) \in \mathcal{M}_{\alpha_1,m_1} \times \mathcal{M}_{\alpha_2,m_2} :}{(\mathbf{A}_1, \mathbf{A}_2, \mathbf{t_A}) \in \mathcal{R}_p^{n\times m_1} \times \mathcal{R}_{p,\mathsf{HNF}}^{n\times m_2} \times \mathcal{R}_p^n \;\middle|\; \mathbf{A_1}\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t_A}} \right\}.$$

The binding property is based on $\mathsf{MSIS}_{\mathcal{R},p,n,m_1+m_2,2\sqrt{\alpha_1^2+\alpha_2^2}}$. For the hiding property, an hard instance of $\mathsf{MLWE}_{\mathcal{R},p,n,m_2,\mathcal{D}_{\sigma_2}}$ ensures indistinguishability of the hiding component $\mathbf{A}_2\mathbf{s}_2 = \begin{bmatrix}\mathbf{I}_n & \mathbf{A}_2'\end{bmatrix}\mathbf{s}_2$. Both rely on computational difficulty.

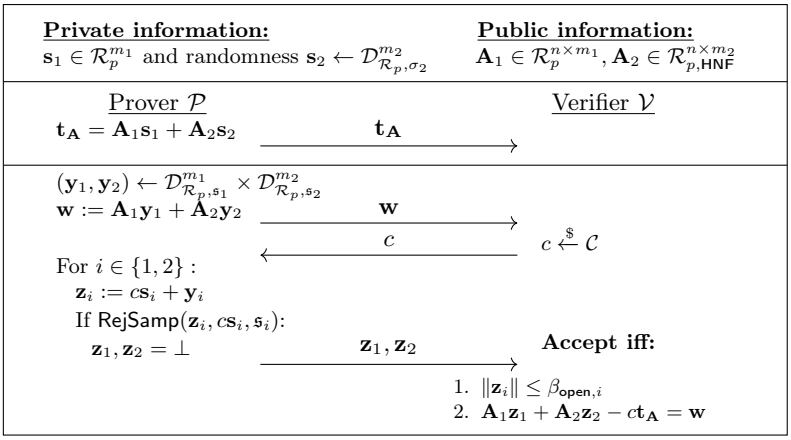

Fig. 1: Proof of Weak Opening of the Ajtai Commitment with a chosen α_i

Challenge Space. In the opening proof, we specify the same challenge space $\mathcal{C}$ than in [LNP22] using their Lemma 15 that ensures both the stability and the

smallness of element of $\mathcal{R}_p$ under a specific automorphism $\sigma_{-1} : X \mapsto X^{-1}$:

$$\mathcal{C} := \left\{ c \in \mathcal{R}_p : \|c\|_1 \leq \kappa, \sigma_{-1}(c) = c, \sqrt[2k]{\|\sigma(c^k) \cdot c^k\|_1} \leq \eta \right\}.$$

For the opening, we must have a space $\overline{\mathcal{C}} := \{c - c' : c, c' \in \mathcal{C} \text{ where } c \neq c'\} \subseteq \mathcal{R}_q^\times$.

Lemma 2.5 ([LNP22, Lemma 2.6]). *Let $p = 5 \mod 8$ being a prime. For any $c \in \mathcal{R}_p$ such that $c = \sigma(c)$. Then c is invertible if and only if $c \neq 0$.*

Original proof of Opening of Lyubashevsky [Lyu12]. The process opens to an element of $\overline{\mathsf{R}}_{\mathsf{ajtai}}$. We need to specify for a good choice of parameters that with overwhelming probability, we have $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}) \in \overline{\mathsf{R}}_{\mathsf{ajtai}} \Rightarrow (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2) \in \mathsf{R}_{\mathsf{ajtai}}$.

$$\overline{\mathsf{R}}_{\mathsf{ajtai}} := \left\{ \frac{(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}) \in \mathcal{M}_{\beta_1,m_1} \times \mathcal{M}_{\beta_2,m_2} \times \overline{\mathcal{C}} :}{(\mathbf{A}_1, \mathbf{A}_2, \mathbf{t_A}) \in \mathcal{R}_p^{n \times m_1} \times \mathcal{R}_{p,\mathsf{HNF}}^{n \times m_2} \times \mathcal{R}_p^n} \;\middle|\; \mathbf{A_1}\bar{\mathbf{s}}_1 + \mathbf{A}_2\bar{\mathbf{s}}_2 = \mathbf{t_A} \right\}.$$

Soundness and Completeness are shown for a good choice of parameters and standard deviations $\mathfrak{s}_i$ of the masks $\mathbf{y}_i$. The zero-knowledge property is ensured assuming rejection sampling methods on the responses $\mathbf{z}_i$. We remind the proof of opening of the Ajtai commitment scheme in Figure 1 with $\mathfrak{s}_i := \gamma_i \eta \alpha_i$ with γ_i the rejection rate of the i-th rejection sampling. Consider a good ring $\mathcal{R}$ such that $m_i d - 1 \geq \lambda$ (the security parameter). By taking $\sigma_2 \leq \eta_\varepsilon(\mathcal{R}_p^{m_2})$, we restrict randomness to $\alpha_2 := \sigma_2\sqrt{m_2 d}$. and set valid opening bounds $\beta_i := \mathfrak{s}_i\sqrt{2m_i d}$.

ABDLOP Commitment Scheme [LNP22]. The Ajtai commitment suffers from two distinct limitations. First, the message space is restricted to bounded polynomials (by α_1), second it does not permit transmitting and committing to additional elements dynamically (during the execution of the protocol). As the global aim of the framework is to prove various relations depending on each other, we need to extend the actual Ajtai commitment scheme to solve that point. In [LNP22], the authors join to it the BDLOP construction from [BDL+18], giving the new ABDLOP commitment scheme, exploiting the same randomness $\mathbf{s}_2$ for both components. The lower part of the commitment (the BDLOP instance) is only used for dynamic messages $\mathbf{m} \in \mathcal{R}_p^\ell$ depending on the execution of the protocol, as $\mathbf{B}$ can be extended and transmitted during the execution.

$$\mathsf{R}_{\mathsf{abdlop}} := \left\{ \frac{(\mathbf{s}_1, \mathbf{s}_2, \mathbf{m}) \in \mathcal{M}_{\alpha_1,m_1} \times \mathcal{M}_{\alpha_2,m_2} \times \mathcal{R}_p^\ell :}{\Big(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, (\mathbf{t_A}, \mathbf{t_B})\Big) \in \mathcal{R}_p^{n \times m_1} \times \mathcal{R}_{p,\mathsf{HNF}}^{n \times m_2} \times \mathcal{R}_{p,\mathsf{HNF}}^{\ell \times m_2} \times \mathcal{R}_p^{n+\ell}} \;\middle|\; \begin{matrix} \mathbf{A_1}\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t_A} \\ \mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t_B} \end{matrix} \right\}.$$

As explained in the original paper, for well chosen parameters, we have with overwhelming probability that $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}) \in \overline{\mathsf{R}}_{\mathsf{abdlop}} \Rightarrow (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \mathbf{t_B} - \mathbf{B}\bar{\mathbf{s}}_2) \in \mathsf{R}_{\mathsf{abdlop}}$. The opening relies essentially on the Ajtai opening $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}) \in \overline{\mathsf{R}}_{\mathsf{ajtai}}$.

$$\overline{\mathsf{R}}_{\mathsf{abdlop}} := \left\{ \frac{(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}) \in \mathcal{M}_{\beta_1,m_1} \times \mathcal{M}_{\beta_2,m_2} \times \overline{\mathcal{C}} :}{\Big(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, (\mathbf{t_A}, \mathbf{t_B})\Big) \in \mathcal{R}_p^{n \times m_1} \times \mathcal{R}_{p,\mathsf{HNF}}^{n \times m_2} \times \mathcal{R}_{p,\mathsf{HNF}}^{\ell \times m_2} \times \mathcal{R}_p^{n+\ell}} \;\middle|\; \begin{matrix} \mathbf{A_1}\bar{\mathbf{s}}_1 + \mathbf{A}_2\bar{\mathbf{s}}_2 = \mathbf{t_A} \\ \overline{\mathbf{m}} = \mathbf{t_B} - \mathbf{B}\bar{\mathbf{s}}_2 \end{matrix} \right\}.$$

3 Another usage of the Gaussianisation process

To build a rejection sampling free commit-and-prove protocol, we start by modifying the ABDLOP commitment scheme used in [LNP22] using the Modified-Ajtai commitment combined with the Gaussianisation process from [HSS24]. We manage to base the zero-knowledge security over a hard instance of Hint-MLWE instead of various instances of rejection sampling.

The Gaussianisation method from [HSS24, Section 3]. The authors managed to construct an randomized encoding that follows a Gaussian distribution centered in the secret encoding. First, it computes an encoding $\hat{\mathbf{s}}$ of an element $\mathbf{s} \in \mathcal{R}_p^m$ for $p = b^k + 1$, exploiting $\hat{\mathbf{s}} = \mathbf{G}_b^{-1}\mathbf{s} \in \mathcal{R}_p^{km}$, a decomposition in base b. In a second step, the encoding is randomized by sampling a non-influent error. At the end, it generates an additional error $\mathbf{e} \leftarrow \mathcal{D}_{\mathcal{R}^{km},\sigma}$ to construct the final randomized encoding $\tilde{\mathbf{t}} = \hat{\mathbf{t}} + \mathfrak{p}_m\mathbf{e}$, with $\mathfrak{p}_m := X^{md} - b$. We remind the [HSS24] Gaussianisation method in Appendix B.3.

3.1 RF-ABDLOP Commitment scheme

Here, we adapt the idea of the Modified-Ajtai commitment scheme from [HSS24] that commit to an encoded element $\tilde{\mathbf{s}}_1$ rather than $\mathbf{s}_1$ directly to ABDLOP, giving the Rejection-Free ABDLOP commitment scheme defined with:

$$\mathsf{R}_{\mathsf{RF}} := \left\{ \begin{array}{l} \underline{(\mathbf{s}_1, \mathbf{s}_2, \mathbf{m}) \in \mathcal{R}_p^{m_1} \times \mathcal{M}_{\alpha_2,m_2} \times \mathcal{R}_p^{\ell} :} \\ \Big(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, (\mathbf{t_A}, \mathbf{t_B})\Big) \in \mathcal{R}_p^{n\times km_1} \times \mathcal{R}_{p,\mathsf{HNF}}^{n\times m_2} \times \mathcal{R}_{p,\mathsf{HNF}}^{\ell\times m_2} \times \mathcal{R}_p^{n+\ell} \end{array} \middle| \begin{array}{c} \mathbf{A}_1\tilde{\mathbf{s}}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t_A} \\ \mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t_B} \\ \tilde{\mathbf{s}}_1 = \mathsf{RandomizedEncode}(\mathbf{s}_1, \sigma_1) \end{array} \right\}.$$

Theorem 3.1 (Binding). *The Rejection-Free ABDLOP commitment scheme is binding if it satisfies for $\varepsilon = \mathsf{negl}(\lambda)$, the following conditions: $\sigma_1 \geq \eta_\varepsilon(\mathcal{R}^{km_1})$, $\sigma_2 \geq \eta_\varepsilon(\mathcal{R}^{m_2})$, $km_1d - 1 \geq \lambda$, $m_2d - 1 \geq \lambda$ and the hardness of the assumption $\mathsf{MSIS}_{\mathcal{R},p,n,km_1+m_2,2\sqrt{\alpha_1^2+\alpha_2^2}}$ for $\alpha_1 = (b+1)\sigma_1\sqrt{km_1d}$ and $\alpha_2 = \sigma_2\sqrt{m_2d}$.*

Proof. Suppose we have two accepted tuples $(\mathbf{s}_1, \mathbf{m}, \mathbf{s}_2)$ and $(\mathbf{s}_1', \mathbf{m}', \mathbf{s}_2')$ for a same commitment $\mathbf{t_A}, \mathbf{t_B}$. Then, we can recover $\mathbf{A}_1\mathbf{s}_1^* + \mathbf{A}_2\mathbf{s}_2^* := \mathbf{A}_1(\tilde{\mathbf{s}}_1 - \tilde{\mathbf{s}}_1') + \mathbf{A}_2(\mathbf{s}_2 - \mathbf{s}_2') = \mathbf{0}$. If the tuples are not equal, then we find a valid MSIS solution using Lemma 2.2 to bound $\|(\mathbf{s}_1^*, \mathbf{s}_2^*)\| \leq 2\|(\tilde{\mathbf{s}}_1, \mathbf{s}_2)\| \leq 2\sqrt{\|\tilde{\mathbf{s}}_1\|^2 + \|\mathbf{s}_2\|^2} \leq 2\sqrt{\alpha_1^2 + \alpha_2^2}$ as we can bound $\|\tilde{\mathbf{s}}_1\| = \|\hat{\mathbf{s}}_1 + \mathfrak{p}_m\mathbf{e}\| \leq \|\mathfrak{p}_m\| \cdot \|\mathbf{e}\| \leq (b+1)\sigma_1\sqrt{km_1d}$. □

Theorem 3.2 (Hiding). *$\mathsf{MLWE}_{\mathcal{R},p,n+\ell,m_2,\mathcal{D}_{\sigma_2}}$ implies the hiding of RF-ABDLOP.*

Proof. Hardness of MLWE implies $\begin{bmatrix}\mathbf{A}_2^\top & \mathbf{B}^\top\end{bmatrix}^\top \mathbf{s}_2 \approx_c \mathcal{U}(\mathcal{R}_p^{n+\ell})$, hence hiding. □

Security parameters will change accordingly, as the length of the $\tilde{\mathbf{s}}_1$ is k times longer $\mathbf{s}_1$. We choose parameters in order to have with overwhelming probability that $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}) \in \overline{\mathsf{R}}_{\mathsf{RF}} \Rightarrow \big(\mathsf{Decode}(\bar{\mathbf{s}}_1), \mathbf{t_B} - \mathbf{B}\bar{\mathbf{s}}_2, \bar{\mathbf{s}}_2\big) \in \mathsf{R}_{\mathsf{RF}}$.

$$\overline{\mathsf{R}}_{\mathsf{RF}} := \left\{ \frac{(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}) \in \mathcal{R}_p^{km_1} \times \mathcal{M}_{\beta_2, m_2} \times \overline{\mathcal{C}} :}{\Big(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, (\mathbf{t_A}, \mathbf{t_B})\Big) \in \mathcal{R}_p^{n \times km_1} \times \mathcal{R}_{p,\mathsf{HNF}}^{n \times m_2} \times \mathcal{R}_{p,\mathsf{HNF}}^{\ell \times m_2} \times \mathcal{R}_p^{n+\ell} \;\middle|\; \begin{matrix} \mathbf{A}_1\bar{\mathbf{s}}_1 + \mathbf{A}_2\bar{\mathbf{s}}_2 = \mathbf{t_A} \\ \overline{\mathbf{m}} = \mathbf{t_B} - \mathbf{B}\bar{\mathbf{s}}_2 \end{matrix}} \right\}.$$

The proof of opening is ensured over well-chosen standard deviations of the mask $\mathbf{y}_i$. These masks follow discrete Gaussian distribution with standard deviations $\sqrt{2}\mathfrak{s}_i$. In our work, the slight modification only concerns the challenge described in Section 2.3, which only affects the bounds related to the lattice assumptions, resulting in the following proofs. As the transmitted $\tilde{\mathbf{s}}_i$ follows a Gaussian distribution, the zero-knowledge property is now assumed over a hard instance of Hint-MLWE, using the transmitted $\mathbf{z}_i$ as hints.

The protocol is defined as a subpart of the future framework of proof, presented in Figure 2 for $\mathcal{F} = \varnothing$. The proof of opening follows the same construction as the one from the Ajtai commitment scheme presented in Figure 1.

Theorem 3.3 (Completeness). *The proof of opening knowledge satisfies completeness if it satisfies for $\varepsilon = \mathsf{negl}(\lambda)$ the following conditions: $\sigma_1, \sqrt{2}\mathfrak{s}_1 \geq \eta_\varepsilon(\mathcal{R}^{km_1}); \sigma_2, \mathfrak{s}_2 \geq \eta_\varepsilon(\mathcal{R}^{m_2})$ for the bounds $\beta_{\mathsf{open},1} = (b+1)(\eta\sigma_1 + \sqrt{2}\mathfrak{s}_1)\sqrt{km_1 d}$ and $\beta_{\mathsf{open},2} = (\eta\sigma_2 + \mathfrak{s}_2)\sqrt{m_2 d}$.*

Proof. For the first verification step, $\|\mathbf{z}_i\| \leq \beta_{\mathsf{open},i}$ it is checked by Lemma 2.2 instantiated for the 4 conditions about the standard deviations. The norm of $\tilde{\mathbf{s}}_1$ is bounded the same way as in the proof of binding of the commitment scheme, for $\tilde{\mathbf{s}}_1$ and $\tilde{\mathbf{y}}_1$. The second verification step follows from the original construction. □

Theorem 3.4 (Knowledge Soundness). *Consider the parameters defined in Theorem 3.1. An extractor either produce a valid unique opening $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}) \in \overline{\mathsf{R}}_{RF}$ with non negligible probability or a $\mathsf{MSIS}_{\mathcal{R},p,n,km_1+m_2,8\eta\sqrt{\beta^2_{\mathsf{open},1}+\beta^2_{\mathsf{open},2}}}$ solution.*

Proof. Assuming the parameters, RF-ABDLOP is binding. Let's assume that the extractor finds two acceptable transcripts with a rewind before the challenge sampling. The two transcripts are $(\mathbf{w}, c, \tilde{\mathbf{z}}_1, \mathbf{z}_2)$ and $(\mathbf{w}, c', \tilde{\mathbf{z}}'_1, \mathbf{z}'_2)$ for a same commitment $(\mathbf{t_A}, \mathbf{t_B})$. The proof follows the one from [LNP22, Lemma 3.1]. Define $\bar{c} := c' - c, \bar{\mathbf{z}}_i := \tilde{\mathbf{z}}'_i - \tilde{\mathbf{z}}_i$ and $\bar{\mathbf{s}}_i := \frac{\bar{\mathbf{z}}_i}{\bar{c}}$. As the transcripts were accepted, we have $\mathbf{A}_1\tilde{\mathbf{s}}_1 + \mathbf{A}_2\mathbf{s}_2 - c\mathbf{t_A} = \mathbf{A}_1\tilde{\mathbf{s}}'_1 + \mathbf{A}_2\mathbf{s}'_2 - c'\mathbf{t_A}$ that implies $\mathbf{A}_1\bar{\mathbf{s}}_1 + \mathbf{A}_2\bar{\mathbf{s}}_2 = \mathbf{t_A}$ then we will have $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}) \in \overline{\mathsf{R}}_{\mathsf{RF}}$.

The unicity follows from [LNP22, Lemma 3.1], take $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}), (\bar{\mathbf{s}}'_1, \bar{\mathbf{s}}'_2, \bar{c}') \in \overline{\mathsf{R}}_{\mathsf{RF}}$ for the same public parameters $(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \mathbf{t_A}, \mathbf{t_B})$. Take also $\bar{\mathbf{z}}_i$ and $\bar{\mathbf{z}}'_i$ with the same construction as above. Then, we have $\mathbf{A_1}(\bar{\mathbf{s}}_1 - \bar{\mathbf{s}}'_1) + \mathbf{A_2}(\bar{\mathbf{s}}_2 - \bar{\mathbf{s}}'_2) = \mathbf{0}$, but here the MSIS is not bounded. Therefore, we proceed as in [LNP22, Lemma 3.1], we construct another SIS solution with $\mathbf{s}^*_i := (\bar{\mathbf{s}}_i - \bar{\mathbf{s}}'_i)\bar{c} \cdot \bar{c}' = \bar{\mathbf{z}}_i\bar{c}' - \bar{\mathbf{z}}'_i\bar{c}$ where

Private information:
Initial messages: $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_p^{m_1+\ell}$,
Sampled on instantiation: $\mathbf{s}_2 \leftarrow \mathcal{D}^{m_2}_{\mathcal{R}_p,\sigma_2}, \tilde{\mathbf{s}}_1 = \mathsf{RandomizedEncode}(\mathbf{s}_1, \sigma_1) \in \mathcal{R}_p^{km_1}$.
Public information:
$\mathbf{A}_1 \in \mathcal{R}_p^{n\times km_1}, \mathbf{A}_2 \in \mathcal{R}_p^{n\times m_2}, \mathbf{B} \in \mathcal{R}_p^{\ell\times m_2}, \underline{\mathbf{b}_1} \in \mathcal{R}_p^{m_2}, \sigma := \sigma_{-1} \in \mathsf{Aut}(\mathcal{R}_p)$,
$\begin{bmatrix}\mathbf{A}_1 \\ \mathbf{0}^{\ell\times m_1}\end{bmatrix}\tilde{\mathbf{s}}_1 + \begin{bmatrix}\mathbf{A}_2 \\ \mathbf{B}\end{bmatrix}\mathbf{s}_2 + \begin{bmatrix}\mathbf{0} \\ \mathbf{m}\end{bmatrix} = \begin{bmatrix}\mathbf{t}_\mathbf{A} \\ \mathbf{t}_\mathbf{B}\end{bmatrix} \mod p$,
$\underline{\mathcal{F}} = \{f_i\}_{i\in[N]}$ a set of quadratic relations as in eq. (1) and (2) s.t $f_i(\mathbf{s}) = 0 \mod p$ with $\mathbf{s}^\top := (\mathbf{s}_1^\top, \sigma(\mathbf{s}_1)^\top, \mathbf{m}^\top, \sigma(\mathbf{m})^\top)$.

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
	$\underline{\mu}$	$\underline{\mu} = (\mu_i)_{i\in[N]} \xleftarrow{\$} \mathcal{R}_p^N$
$\underline{f} := \sum_{i\in[N]} \mu_i f_i$		
$\underline{\tilde{f}}$ with $\mathbf{R}_2'$ and $\mathbf{r}_1'$ defined as in eq. (3).		
$\tilde{\mathbf{s}}^\top := \left[\tilde{\mathbf{s}}_1^\top, \widetilde{\sigma(\mathbf{s}_1)}^\top, \mathbf{m}^\top, \sigma(\mathbf{m})^\top\right]$		
$\mathbf{y}_{\mathsf{rand}} \xleftarrow{\$} \mathcal{R}_p^{m_1}$		
$\tilde{\mathbf{y}}_1 \leftarrow \mathsf{RandomizedEncode}(\mathbf{y}_{\mathsf{rand}}, \sqrt{2}\mathfrak{s}_1)$		
$\mathbf{y}_2 \leftarrow \mathcal{D}^{m_2}_{\mathcal{R}_p,\mathfrak{s}_2}$		
$\mathbf{w} := \mathbf{A}_1\tilde{\mathbf{y}}_1 + \mathbf{A}_2\mathbf{y}_2$		
$\underline{\mathbf{y}^\top} := \left[\tilde{\mathbf{y}}_1^\top, \sigma(\tilde{\mathbf{y}}_1^\top), (-\mathbf{B}\mathbf{y}_2)^\top, \sigma(-\mathbf{B}\mathbf{y}_2)^\top\right]$		
$\underline{\mathfrak{g}_0} := \mathbf{y}^\mathsf{T}\mathbf{R}_2'\mathbf{y} + \mathbf{b}_1^\mathsf{T}\mathbf{y}_2$		
$\underline{\mathfrak{g}_1} := \tilde{\mathbf{s}}^\mathsf{T}\mathbf{R}_2'\mathbf{y} + \mathbf{y}^\top\mathbf{R}_2'\tilde{\mathbf{s}} + \mathbf{y}^\top\mathbf{r}_1'$		
$\underline{t_1} := \mathbf{b}_1^\mathsf{T}\mathbf{s}_2 + \mathfrak{g}_1$		
	$\mathbf{w}, \underline{\mathfrak{g}_0, t_1}$	
		$c \xleftarrow{\$} \mathcal{C}$
	c	
$\tilde{\mathbf{z}}_1 := c\tilde{\mathbf{s}}_1 + \tilde{\mathbf{y}}_1$		
$\mathbf{z}_2 := c\mathbf{s}_2 + \mathbf{y}_2$		
	$\mathbf{z}_1, \mathbf{z}_2$	**Accept iff**:
		– $\|z_i\| \le \beta_{\mathsf{open},i}$
		– $\mathbf{A}_1\tilde{\mathbf{z}}_1 + \mathbf{A}_2\mathbf{z}_2 - c\mathbf{t}_\mathbf{A} = \mathbf{w}$
		– <u>By computing:</u>
		• $\mathbf{z}^\top := \begin{bmatrix}\tilde{\mathbf{z}}_1 \\ \sigma(\tilde{\mathbf{z}}_1) \\ c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2 \\ \sigma(c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2)\end{bmatrix}$
		• $\underline{v := ct_1 + \mathfrak{g}_0 - \mathbf{b}_1^\mathsf{T}\mathbf{z}_2}$
		– <u>Check if: $\mathbf{z}^\mathsf{T}\mathbf{R}_2'\mathbf{z} + c\mathbf{z}^\mathsf{T}\mathbf{r}_1' + c^2 r_0 = v$</u>

Fig. 2: Proof of knowledge of an element $(\mathbf{s}_1, \mathbf{m})$ along with the proof that for $f_i \in \mathcal{F}$ where $\mathcal{F}$ is a public set of linear and quadratic relations, we have $f_i(\psi(\mathbf{s}_1, \mathbf{m})) = 0$. The protocol can also be instantiated with $\mathcal{F} = \varnothing$ without the underlined elements of the transcript (and their construction) and the underlined verification steps.

$\mathcal{S}_{\underline{\mathsf{REL\text{-}AUT\text{-}}\mathcal{F}}}(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \underline{\mathbf{b}_1})$

1. **Targets**: $\mathbf{t_A}, \mathbf{t_B} \xleftarrow{\$} \mathcal{R}_p^{n+\ell}$, **Challenges**: $c \xleftarrow{\$} \mathcal{C}$, $\underline{\mu = (\mu_i)_{i\in N} \xleftarrow{\$} \mathcal{R}_p^N}$;
2. $\tilde{\mathbf{s}}_\mathbf{1} \leftarrow \mathsf{RandomizedEncode}(\mathbf{0}, \sigma_1), \mathbf{s}_2 \leftarrow \mathcal{D}^{m_2}_{\mathcal{R}_p, \sigma_2}$;
3. $\mathbf{y}_{\mathsf{rand}} \xleftarrow{\$} \mathcal{R}_p^{m_1}, \tilde{\mathbf{y}}_1 \leftarrow \mathsf{RandomizedEncode}(\mathbf{y}_{\mathsf{rand}}, \sqrt{2}\mathfrak{s}_1), \mathbf{y}_2 \leftarrow \mathcal{D}^{m_2}_{\mathcal{R}_p, \mathfrak{s}_2}$;
4. $\tilde{\mathbf{z}}_1 = c\tilde{\mathbf{s}}_\mathbf{1} + \tilde{\mathbf{y}}_1$, $\mathbf{z}_2 = c\mathbf{s}_2 + \mathbf{y}_2$;
5. $\mathbf{w} = \mathbf{A}_1\tilde{\mathbf{z}}_1 + \mathbf{A}_2\mathbf{z}_2 - c\mathbf{t_A}$;
6. $\underline{f := \sum_{i\in[N]} \mu_i f_i}$, constructs $\tilde{f}$ with $\mathbf{R}'_2$ and $\mathbf{r}'_1$ defined as in eq. (3);
7. $\underline{\mathbf{y} := \begin{bmatrix} \tilde{\mathbf{y}}_1 \\ \sigma(\tilde{\mathbf{y}}_1) \\ -\mathbf{B}\mathbf{y}_2 \\ \sigma(-\mathbf{B}\mathbf{y}_2) \end{bmatrix}}$, $\mathbf{z} := \begin{bmatrix} \tilde{\mathbf{z}}_1 \\ \sigma(\tilde{\mathbf{z}}_1) \\ c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2 \\ \sigma(c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2) \end{bmatrix}$;
8. $\underline{\mathfrak{g}_0 := \mathbf{y}^\mathsf{T}\mathbf{R}'_2\mathbf{y} + \mathbf{b}_1^\mathsf{T}\mathbf{y}_2}$;
9. $\underline{t_1 = c^{-1} \cdot \left(\mathbf{z}^\mathsf{T}\mathbf{R}'_2\mathbf{z} + c\mathbf{z}^\mathsf{T}\mathbf{r}'_1 + c^2 r_0 - \mathfrak{g}_0 + \mathbf{b}_1^\mathsf{T}\mathbf{z}_2\right)}$;
10. Output $(\mathbf{t_A}, \mathbf{t_B}, \mathbf{w}, c, \mathbf{z}_1, \mathbf{z}_2, \underline{\mu, \mathfrak{g}_0, t_1})$.

Fig. 3: Definition of $\mathcal{S}(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B})$ (without underlined elements) is the simulator for RF-ABDLOP in Theorem 3.5, the output is defined as the set of $\mathcal{H}_6$.
Definition of $\mathcal{S}_{\mathsf{REL\text{-}AUT\text{-}}\mathcal{F}}(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \mathbf{b}_1)$ generalizes the simulator in order to prove the simulatability in Theorem 4.5 for a set $\mathcal{F} = \{f_i\}_{i\in[N]}$.

we exploit the verification step (of the norms of the responses) to prove the size of the solution: $\|(\mathbf{s}_1^*, \mathbf{s}_2^*)\| \leq 2 \cdot 2\eta \cdot 2\|(\tilde{\mathbf{z}}_1, \mathbf{z}_2)\| \leq 8\eta\sqrt{\beta^2_{\mathsf{open},1} + \beta^2_{\mathsf{open},2}}$. □

Theorem 3.5 (Simulatability). *The proof of opening knowledge protocol satisfies simulatability if the commitment scheme is initially hiding (assuming Theorem 3.2) and if for* $\varepsilon = \mathsf{negl}(\lambda)$, $\frac{1}{\varsigma_1^2} := 2\left(\frac{1}{\sigma_1^2} + \frac{\eta^2}{\mathfrak{s}_1^2}\right)$ *and* $\frac{1}{\varsigma_2^2} := 2\left(\frac{1}{\sigma_2^2} + \frac{\eta^2}{2\mathfrak{s}_2^2}\right)$ *it satisfies the following conditions:* $\sigma_1, \sqrt{2}\mathfrak{s}_1 \geq \eta_\varepsilon(\mathcal{R}^{km_1})$; $\mathfrak{s}_1, 2\varsigma_1 \geq \frac{\sqrt{2}}{(b-1)} \cdot \eta_\varepsilon(\mathfrak{p}_{m_1}\mathcal{R}^{km_1})$;$\frac{\sigma_2}{\sqrt{2}}, \mathfrak{s}_2, \varsigma_2 \geq \sqrt{2} \cdot \eta_\varepsilon(\mathcal{R}^{m_2})$ *and* $\mathsf{MLWE}_{\mathcal{R},p,n,m_2,\varsigma_2}$ *is hard (this implies* Hint-MLWE *via Theorem 2.1), implying indistinguishability between an output of* $\mathcal{S}$ *from Figure 3 and an honest transcript* π *generated from* R_{RF}.

Proof. The proof essentially follows the original ones [KLSS23, HSS24], which involves Hint-MLWE. We construct the experiments $\mathsf{Exp}_i(\mathcal{A})$ (in 1) where a PPT adversary $\mathcal{A}$ is asked to distinguish between the two sets $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$. We define $\mathsf{Adv}_i(\mathcal{A}) = |\mathbb{P}[\mathsf{Exp}_i(\mathcal{A}) = 1] - \frac{1}{2}|$. $\mathcal{H}_0$ is the set defined by an honest generation of the protocol meanwhile the set $\mathcal{H}_6$ is the one generated by the simulator $\mathcal{S}$ from Figure 3.

$\mathsf{Exp}_i(\mathcal{A})$:

$(\mathbf{s}_1, \mathbf{s}_2, \mathbf{m}) \xleftarrow{\$} \mathcal{R}_p^{m_1} \times \mathcal{M}_{\alpha_2,m_2} \times \mathcal{R}_p^{\ell}$
$\mathsf{pp} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}) \xleftarrow{\$} \mathcal{R}_p^{n \times km_1} \times \mathcal{R}_{p,\mathsf{HNF}}^{n \times m_2} \times \mathcal{R}_{p,\mathsf{HNF}}^{\ell \times m_2}$
$\mathsf{msg} =$
$b \xleftarrow{\$} \{0,1\}$
if $i + b \in \{0,1,2,3,4\}$ **then**
  $\mathsf{msg} = \mathbf{s}_1$,
if $i + b = 0$ **then**
  $\mathsf{msg} = \mathsf{msg} \| (\mathbf{s}_2, \mathbf{m})$,
$\pi \leftarrow \mathcal{H}_{i+b}(\mathsf{pp}, \mathsf{msg})$
$\hat{b} \leftarrow \mathcal{A}(\mathsf{pp}, \pi)$
return 1 if $\hat{b} = b$ and 0 otherwise.

Claim. Assuming the commitment scheme is hiding, $\mathsf{Adv}_0(\mathcal{A})$ is negligible.

Proof. The $\mathbf{s}_2$ element is the randomness of the commitment scheme, which changes between different launches. We simply put its sampling in the set in order to show that it can be artifically sampled by the simulator. Another modification is assuming the hiding of the lower part of the commitment scheme: it is hard to distinguish between 2 chosen message $\mathbf{m}$ and $\mathbf{0}$, we will simply omit the message $\mathbf{m}$ in the following. □

Claim. Assuming the commitment scheme is hiding, if $\mathsf{Adv}_1(\mathcal{A})$ non negligible, then there exists a polynomial solver of the $\mathsf{Hint\text{-}MLWE}_{\mathcal{R},p,n,m_2,1,\sigma_2,\mathfrak{s}_2,\mathcal{C}}$ problem for the secret $\mathbf{s}_2$ succeeding with non negligible probability. Therefore, assuming the hardness of $\mathsf{Hint\text{-}MLWE}_{\mathcal{R},p,n,m_2,1,\sigma_2,\mathfrak{s}_2,\mathcal{C}}$, $\mathsf{Adv}_1(\mathcal{A})$ is negligible.

Proof. Suppose we have $\mathcal{A}$ a PPT adversary succeeding $\mathsf{Exp}_1(\mathcal{A})$ for any choice of pp with non-negligible advantage i.e. distinguishing between elements from $\mathcal{H}_1$ and $\mathcal{H}_2$. We construct an efficient polynomial probabilistic algorithm $\mathcal{B}$ solving $\mathsf{Hint\text{-}MLWE}_{\mathcal{R},p,n,m_2,1,\sigma_2,\mathfrak{s}_2,\mathcal{C}}$.
Denote the $\mathsf{Hint\text{-}MLWE}$ instance received by the challenger $(\mathbf{A}, \mathbf{u}, c, \mathbf{z})$:
We will proceed as follows:

1. $\mathbf{A}_2 := [\mathbf{I}_n \mid \mathbf{A}] \in \mathcal{R}_p^{n \times m_2}$, $\mathbf{A}_1 \xleftarrow{\$} \mathcal{R}_p^{n \times m_1}, \mathbf{B}' \xleftarrow{\$} \mathcal{R}_p^{\ell \times n}, \mathbf{B} := \mathbf{B}'\mathbf{A}_2$,
2. $\mathbf{t_A} := \mathbf{u} + \mathbf{A}_1\tilde{\mathbf{s}}, \mathbf{t_B} := \mathbf{B}'\mathbf{u}, \mathbf{s} \xleftarrow{\$} \mathcal{R}_p^{m_1}, \tilde{\mathbf{s}} \leftarrow \mathsf{RandomizedEncode}(\mathbf{s}, \sigma_1)$,
3. $\mathbf{y}_{\mathsf{rand}} \xleftarrow{\$} \mathcal{R}_p^{m_1}, \tilde{\mathbf{y}} \leftarrow \mathsf{RandomizedEncode}(\mathbf{y}_{\mathsf{rand}}, \mathfrak{s}_1)$, $\mathbf{w} = \mathbf{A}_2\mathbf{z} - c\mathbf{u} + \mathbf{A}_1\tilde{\mathbf{y}}$, $\tilde{\mathbf{z}} := c\tilde{\mathbf{s}} + \tilde{\mathbf{y}}$,
4. Output $\mathcal{B}\left((\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}_1), (\mathbf{t_A}, \mathbf{t_B}, \mathbf{w}, c, \tilde{\mathbf{z}}, \mathbf{z})\right)$.

Suppose $\mathbf{u} = [\mathbf{I}_n \mid \mathbf{A}]\mathbf{r}$ where $\mathbf{r} \leftarrow \mathcal{D}_{\mathcal{R}_p^{m_2}, \sigma_2}$. Then, $\mathbf{t_A} = \mathbf{u} + \mathbf{A}_1\tilde{\mathbf{s}} = \mathbf{A}_1\tilde{\mathbf{s}} + \mathbf{A}_2\mathbf{r}$, where $\mathbf{s}$ is known as in both distributions. Coming from the initial instance, the hint $\mathbf{z} = c\mathbf{r} + \mathbf{y}$ follows the same distributions in $\mathcal{H}_1$ and $\mathcal{H}_2$. The challenge c is also unmodified. For $\mathbf{w}$, we have $\mathbf{w} = \mathbf{A}_2\mathbf{z} - c\mathbf{u} + \mathbf{A}_1\tilde{\mathbf{y}} = c\mathbf{A}_2\mathbf{r} + \mathbf{A}_2\mathbf{y} -$

$\mathcal{H}_0\,(\mathsf{pp}, (\mathbf{s}_1, \mathbf{s}_2, \mathbf{m}))$:

$\tilde{\mathbf{s}}_\mathbf{1} \leftarrow \mathsf{RandEnc}(\mathbf{s}_1, \sigma_1)$
$\mathbf{t_A} = \mathbf{A}_1\tilde{\mathbf{s}}_\mathbf{1} + \mathbf{A}_2\mathbf{s}_2, \mathbf{t_B} = \mathbf{B}\mathbf{s}_2 + \mathbf{m}$
$c \xleftarrow{\$} \mathcal{C}, \mathbf{y}_{\mathsf{rand}} \xleftarrow{\$} \mathcal{R}_p^{m_1}, \mathbf{y}_2 \leftarrow \mathcal{D}^{m_2}_{\mathcal{R}_p,\mathfrak{s}_2}$
$\tilde{\mathbf{y}}_1 \leftarrow \mathsf{RandEnc}(\mathbf{y}_{\mathsf{rand}}, \sqrt{2}\mathfrak{s}_1)$
$\mathbf{w} = \mathbf{A}_1\tilde{\mathbf{y}}_\mathbf{1} + \mathbf{A}_2\mathbf{y}_2$
$\tilde{\mathbf{z}}_1 = c\tilde{\mathbf{s}}_\mathbf{1} + \tilde{\mathbf{y}}_1,\ \mathbf{z}_2 = c\mathbf{s}_2 + \mathbf{y}_2$
Return $(\mathbf{t_A}, \mathbf{t_B}, \mathbf{w}, c, \tilde{\mathbf{z}}_1, \mathbf{z}_2)$

$\mathcal{H}_1\,(\mathsf{pp}, \mathbf{s}_1)$:

$\tilde{\mathbf{s}}_\mathbf{1} \leftarrow \mathsf{RandEnc}(\mathbf{s}_1, \sigma_1), \mathbf{s}_2 \leftarrow \mathcal{D}^{m_2}_{\mathcal{R}_p,\sigma_2}$
$\mathbf{t_A} = \mathbf{A}_1\tilde{\mathbf{s}}_\mathbf{1} + \mathbf{A}_2\mathbf{s}_2, \mathbf{t_B} = \mathbf{B}\mathbf{s}_2 + \mathbf{0}$
$c \xleftarrow{\$} \mathcal{C}, \mathbf{y}_{\mathsf{rand}} \xleftarrow{\$} \mathcal{R}_p^{m_1}, \mathbf{y}_2 \leftarrow \mathcal{D}^{m_2}_{\mathcal{R}_p,\mathfrak{s}_2}$
$\mathbf{y}_1 \leftarrow \mathsf{RandEnc}(\mathbf{y}_{\mathsf{rand}}, \sqrt{2}\mathfrak{s}_1)$
$\mathbf{w} = \mathbf{A}_1\tilde{\mathbf{y}}_\mathbf{1} + \mathbf{A}_2\mathbf{y}_2$
$\tilde{\mathbf{z}}_1 = c\tilde{\mathbf{s}}_\mathbf{1} + \tilde{\mathbf{y}}_1,\ \mathbf{z}_2 = c\mathbf{s}_2 + \mathbf{y}_2$
Return $(\mathbf{t_A}, \mathbf{t_B}, \mathbf{w}, c, \tilde{\mathbf{z}}_1, \mathbf{z}_2)$

$\mathcal{H}_2\,(\mathsf{pp}, \mathbf{s}_1)$:

$\tilde{\mathbf{s}}_\mathbf{1} \leftarrow \mathsf{RandEnc}(\mathbf{s}_1, \sigma_1), \mathbf{s}_2 \leftarrow \mathcal{D}^{m_2}_{\mathcal{R}_p,\sigma_2}$
$\mathbf{t_A} \xleftarrow{\$} \mathcal{R}_p^n, \mathbf{t_B} \xleftarrow{\$} \mathcal{R}_p^\ell$
$c \xleftarrow{\$} \mathcal{C}, \mathbf{y}_{\mathsf{rand}} \xleftarrow{\$} \mathcal{R}_p^{m_1}, \mathbf{y}_2 \leftarrow \mathcal{D}^{m_2}_{\mathcal{R}_p,\mathfrak{s}_2}$
$\tilde{\mathbf{y}}_1 \leftarrow \mathsf{RandEnc}(\mathbf{y}_{\mathsf{rand}}, \sqrt{2}\mathfrak{s}_1)$
$\mathbf{w} \xleftarrow{\$} \mathcal{R}_p^n$
$\tilde{\mathbf{z}}_1 = c\tilde{\mathbf{s}}_\mathbf{1} + \tilde{\mathbf{y}}_1,\ \mathbf{z}_2 = c\mathbf{s}_2 + \mathbf{y}_2$
Return $(\mathbf{t_A}, \mathbf{t_B}, \mathbf{w}, c, \tilde{\mathbf{z}}_1, \mathbf{z}_2)$

$\mathcal{H}_3\,(\mathsf{pp}, \mathbf{s}_1)$:

$\tilde{\mathbf{s}}_\mathbf{1} \leftarrow \mathsf{RandEnc}(\mathbf{s}_1, \sigma_1), \mathbf{s}_2 \leftarrow \mathcal{D}^{m_2}_{\mathcal{R}_p,\sigma_2}$
$\mathbf{t_A} \xleftarrow{\$} \mathcal{R}_p^n, \mathbf{t_B} \xleftarrow{\$} \mathcal{R}_p^\ell$
$c \xleftarrow{\$} \mathcal{C}, \mathbf{y}_{\mathsf{rand}} \xleftarrow{\$} \mathcal{R}_p^{m_1}, \mathbf{y}_2 \leftarrow \mathcal{D}^{m_2}_{\mathcal{R}_p,\mathfrak{s}_2}$
$\tilde{\mathbf{y}}_{1,1} \leftarrow \mathsf{RandEnc}(-c\mathbf{s}_1, \mathfrak{s}_1)$
$\tilde{\mathbf{y}}_{1,2} \leftarrow \mathsf{RandEnc}(\mathbf{y}_{\mathsf{rand}} + c\mathbf{s}_1, \mathfrak{s}_1)$
$\tilde{\mathbf{y}}_1 = \tilde{\mathbf{y}}_{1,1} + \tilde{\mathbf{y}}_{1,2}$, $\mathbf{w} \xleftarrow{\$} \mathcal{R}_p^n$
$\tilde{\mathbf{z}}_1 = c\tilde{\mathbf{s}}_\mathbf{1} + \tilde{\mathbf{y}}_1,\ \mathbf{z}_2 = c\mathbf{s}_2 + \mathbf{y}_2$
Return $(\mathbf{t_A}, \mathbf{t_B}, \mathbf{w}, c, \tilde{\mathbf{z}}_1, \mathbf{z}_2)$

$\mathcal{H}_4\,(\mathsf{pp})$:

$\tilde{\mathbf{s}}_\mathbf{1} \leftarrow \mathsf{RandEnc}(\mathbf{0}, \sigma_1)$,
$\mathbf{s}_2 \leftarrow \mathcal{D}^{m_2}_{\mathcal{R}_p,\sigma_2}$
$\mathbf{t_A} \xleftarrow{\$} \mathcal{R}_p^n, \mathbf{t_B} \xleftarrow{\$} \mathcal{R}_p^\ell$
$c \xleftarrow{\$} \mathcal{C}, \mathbf{y}_{\mathsf{rand}} \xleftarrow{\$} \mathcal{R}_p^{m_1}, \mathbf{y}_2 \leftarrow \mathcal{D}^{m_2}_{\mathcal{R}_p,\mathfrak{s}_2}$
$\tilde{\mathbf{y}}_{1,1} \leftarrow \mathsf{RandEnc}(\mathbf{0}, \mathfrak{s}_1)$
$\tilde{\mathbf{y}}_{1,2} \leftarrow \mathsf{RandEnc}(\mathbf{y}_{\mathsf{rand}}, \mathfrak{s}_1)$
$\tilde{\mathbf{y}}_1 = \tilde{\mathbf{y}}_{1,1} + \tilde{\mathbf{y}}_{1,2}, \mathbf{w} \xleftarrow{\$} \mathcal{R}_p^n$
$\tilde{\mathbf{z}}_1 = c\tilde{\mathbf{s}}_\mathbf{1} + \tilde{\mathbf{y}}_1,\ \mathbf{z}_2 = c\mathbf{s}_2 + \mathbf{y}_2$
Return $(\mathbf{t_A}, \mathbf{t_B}, \mathbf{w}, c, \tilde{\mathbf{z}}_1, \mathbf{z}_2)$

$\mathcal{H}_5\,(\mathsf{pp})$:

$\tilde{\mathbf{s}}_\mathbf{1} \leftarrow \mathsf{RandEnc}(\mathbf{0}, \sigma_1), \mathbf{s}_2 \leftarrow \mathcal{D}^{m_2}_{\mathcal{R}_p,\sigma_2}$
$\mathbf{t_A} \xleftarrow{\$} \mathcal{R}_p^n, \mathbf{t_B} \xleftarrow{\$} \mathcal{R}_p^\ell$
$c \xleftarrow{\$} \mathcal{C}, \mathbf{y}_{\mathsf{rand}} \xleftarrow{\$} \mathcal{R}_p^{m_1}, \mathbf{y}_2 \leftarrow \mathcal{D}^{m_2}_{\mathcal{R}_p,\mathfrak{s}_2}$
$\tilde{\mathbf{y}}_1 \leftarrow \mathsf{RandEnc}(\mathbf{y}_{\mathsf{rand}}, \sqrt{2}\mathfrak{s}_1)$
$\mathbf{w} \xleftarrow{\$} \mathcal{R}_p^n$
$\tilde{\mathbf{z}}_1 = c\tilde{\mathbf{s}}_\mathbf{1} + \tilde{\mathbf{y}}_1,\ \mathbf{z}_2 = c\mathbf{s}_2 + \mathbf{y}_2$
Return $(\mathbf{t_A}, \mathbf{t_B}, \mathbf{w}, c, \tilde{\mathbf{z}}_1, \mathbf{z}_2)$

Fig. 4: Hybrid functions used in the proof of the simulatability Theorem 3.5. Highlighted elements in yellow are those that are changed between each functions, in green the one that will be changed between $\mathcal{H}_5$ and $\mathcal{H}_6$, generated by the simulator $\mathcal{S}$ from Figure 3.

$c\mathbf{A}_2\mathbf{r} + \mathbf{A}_1\tilde{\mathbf{y}} = \mathbf{A}_1\tilde{\mathbf{y}} + \mathbf{A}_2\mathbf{y}$ with well-distributed $\tilde{\mathbf{y}}$ and $\mathbf{y}$. Also, $\mathbf{t_B} = \mathbf{B}'\mathbf{u} = \mathbf{B}\mathbf{r}$. Therefore, this case explicitly generates an element from $\mathcal{H}_1((\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}), \mathbf{s})$.

Suppose $\mathbf{u}$ is uniformly sampled from $\mathcal{R}_p^n$: $\mathbf{t_A}, \mathbf{t_B}$ and $\mathbf{w}$ are also distributed uniformly over $\mathcal{R}_p$. As in the precedent case, $\mathbf{z}$ and the challenge c remain the same and keep the same distribution. So, this case generates an element of $\mathcal{H}_2((\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}), \tilde{\mathbf{s}})$.
Our algorithm $\mathcal{B}$ outputting the same bit than the one resulted by $\mathcal{A}$, we have that $\mathcal{B}$ has the same non-negligible advantage distinguishing Hint-MLWE samples. □

Claim. $\mathsf{Adv}_2(\mathcal{A})$ is negligible.

Proof. First, we remind that the encoding function is homomorphic. What we need to show here is that the distribution of $\mathbf{z}_1$ is the same in both case. Especially here, we will show that $\tilde{\mathbf{y}}_1$ in both distribution are statistically close. Without any loss of generality, we fix c. First, we remind the definition of the encoding function:

$$\tilde{\mathbf{y}}_{1,1} \leftarrow \mathcal{D}_{\mathsf{Encode}(-c\mathbf{s}_1)+\mathfrak{p}_{m_1}\mathcal{R}^{km_1}, \mathfrak{s}_1 P_{m_1}} \text{ and } \tilde{\mathbf{y}}_{1,2} \leftarrow \mathcal{D}_{\mathsf{Encode}(\mathbf{y_{rand}}+c\mathbf{s}_1)+\mathfrak{p}_{m_1}\mathcal{R}^{km_1}, \mathfrak{s}_1 P_{m_1}}.$$

By applying the Convolution Lemma 2.4 and the homomorphic property, $\tilde{\mathbf{y}}_1$ is at distance 8ε of $\mathcal{D}_{\mathsf{Encode}(\mathbf{y_{rand}})+\mathfrak{p}_{m_1}\mathcal{R}^{km_1}, \sqrt{2}\mathfrak{s}_1 P_{m_1}}$, requiring $\mathfrak{s}_1 P_{m_1} \geq \eta_\varepsilon(\mathfrak{p}_{m_1}\mathcal{R}^{km_1})$, that is true by our conditions. $\mathsf{Adv}_2(\mathcal{A}) \leq 8\varepsilon = \mathsf{negl}(1^\lambda)$. □

Claim. $\mathsf{Adv}_3(\mathcal{A})$ is negligible.

Proof. The focus will be on distinguishing $\tilde{\mathbf{z}}_1$ generated with and without knowledge of $\mathbf{s}_1$. Here we use Lemma 2.3. Without any loss of generality, we fix c, and define $C := \mathsf{Rot}_{km_1 d}(c)$. We take a look of $\mathcal{H}_3$. We view $\mathbf{s}_1$ as a sample i.e. $\mathcal{D}_{\mathsf{Encode}(\mathbf{s}_1)+\mathfrak{p}_{m_1}\mathcal{R}^{km_1}, \sigma_1 P_{m_1}}$. We clearly see $\tilde{\mathbf{y}}_{1,1}$ of as the masking part of the hint. By lemma 2.3, if the condition:

$$\sqrt{\left(\sigma_1^{-2}(P_{m_1}^\mathsf{T} P_{m_1})^{-1} + \mathfrak{s}_1^{-2} C^\mathsf{T}(P_{m_1}^\mathsf{T} P_{m_1})^{-1} C\right)^{-1}} \geq \eta_\varepsilon(\mathfrak{p}_{m_1}\mathcal{R}^{km_1})$$

is satisfied then the distributions $\{c\tilde{\mathbf{s}}_1 + \tilde{\mathbf{y}}_{1,1}\}$ of both $\mathcal{H}_3$ and $\mathcal{H}_4$ are without 2ε-statistical distance. By lemma 2.1, we bound:

$$\begin{aligned}
\left\|\sigma_1^{-2}(P_{m_1}^\mathsf{T} P_{m_1})^{-1} + \mathfrak{s}_1^{-2} C^\mathsf{T}(P_{m_1}^\mathsf{T} P_{m_1})^{-1} C\right\|_2 &\leq (\sigma_1^{-2} + \mathfrak{s}_1^{-2}\|C\|_2^2)\|P_{m_1}^{-1}\|^2 \\
&\leq (\sigma_1^{-2} + \eta^2\mathfrak{s}_1^{-2})(b-1)^{-2} \\
&= (\sqrt{2}\varsigma_1)^{-2}(b-1)^{-2} \\
&\leq \eta_\varepsilon\left(\mathfrak{p}_{m_1}\mathcal{R}^{m_1}\right)^{-2}
\end{aligned}$$

if assuming that $\varsigma_1 \geq \frac{1}{\sqrt{2}(b-1)}\eta_\varepsilon(\mathfrak{p}_{m_1}\mathcal{R}^{km_1})$ and by the fact that $\|P_{m_1}^{-1}\|_2 \leq \frac{1}{b-1}$. We also exploit the bound $\|C\|_2^2 \leq \eta^2$ in Appendix B.4.

As $\mathbf{y}_{\mathsf{rand}}$ is a uniformly sampled from of $\mathcal{R}_p^n$, it follows that the distribution $\mathbf{y}_{\mathsf{rand}} + c\mathbf{s}_1$ is uniform over the same set. Therefore $\tilde{\mathbf{y}}_{1,2}$ follows the same distribution in both $\mathcal{H}_3$ and $\mathcal{H}_4$. Finally, this implies that both $\tilde{\mathbf{z}}_1$ are also within a 2ε statistical distance. $\mathsf{Adv}_2(\mathcal{A}) \leq 2\varepsilon = \mathsf{negl}(1^\lambda)$. □

Claim. $\mathsf{Adv}_4(\mathcal{A})$ is negligible.

Proof. The proof is the same as the proof of negligibility of $\mathsf{Adv}_2(\mathcal{A})$. □

Claim. $\mathsf{Adv}_5(\mathcal{A})$ is negligible i.e. distinguishing between samples from $\mathcal{H}_5$ and $\mathcal{H}_6$ cannot be achived with non-negligible advantage (where $\mathcal{H}_6$ is defined as the output of the simulator $\mathcal{S}$ from Figure 3).

Proof. Between $\mathcal{H}_5$ and $\mathcal{H}_6$, as the elements constructing $\mathbf{w}$ are uniformly random, $\mathbf{w}$ from $\mathcal{H}_6$ follows the uniform distribution as in $\mathcal{H}_5$. □

Therefore, we have $\mathsf{Adv}_{\mathsf{sim}}(\mathcal{A}) \leq \sum_{i=0}^{5} \mathsf{Adv}_i(\mathcal{A}) = \mathsf{negl}(\lambda)$, that quantify the advantage to distinguish between the simulated output of $\mathcal{S}$ and an honest transcript for the same pp. □

4 Proving more than just the knowledge

The aim of our construction is to give additional information about a committed secret message, such as proving linear or quadratic relations over the message.

The [LNP22] framework uses the fact that the message was not modified but only hidden in a commitment scheme. As we are not sending the message unmodified but its randomized encoding version, we can no longer use the original construction. This section demonstrates how to adapt the proof of both linear and quadratic relations over the randomized encoding, to proofs of the same relations over the initial message. Our protocol manages to extend the prove of knowledge of a committed element using the RF-ABDLOP commitment scheme, therefore, we no longer specify the rings and lengths. We refer to an instance of R_{RF} in (3.1) in Section 3.1 for the public matrices $\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}$.

4.1 Transform the relations

From relations over the secret to relations over its encoding. A main tool of the [LNP22] framework is the proof of both linear and quadratic relations over $\mathcal{R}_p$ of messages $\mathbf{s}_1$ and $\mathbf{m}$, which can be viewed as a matrix product over the coefficient of the matrices. We denote by $\mathcal{F}$ the set of functions f_i to be proven equal to 0 (w.l.o.g.), i.e. such that $\mathbf{s}$ is in $\mathsf{Ker}\mathcal{F} := \cap_{f_i \in \mathcal{F}} \mathsf{Ker} f_i$.

A quadratic relation over the committed element can be seen as proving the equation $f(\mathbf{s}) = \mathbf{s}^\top \mathbf{R}_2 \mathbf{s} + \mathbf{r}_1^\top \mathbf{s} + r_0 = 0$ for a committed message $\mathbf{s}^\top := (\mathbf{s}_1^\top, \mathbf{m}^\top)$, a matrix $\mathbf{R}_2 \in \mathcal{R}_p^{(m_1+\ell)\times(m_1+\ell)}$ and elements $\mathbf{r}_1 \in \mathcal{R}_p^{m_1+\ell}$ and $r_0 \in \mathcal{R}_p$. The linear function restricts f to $\mathbf{R}_2$ being equal to $\mathbf{0}^{(m_1+\ell)\times(m_1+\ell)}$.

Remark 4.1. We can identify $\mathbf{R}_2$ as a triangular matrix (upper or lower depending on the transpose case in which we are) and, by abuse of notation, restrict the quadratic component to $\mathbf{s}^\top \mathbf{R}_2 \mathbf{s} = \mathbf{s}_1^\top \mathbf{R}_{1,1} \mathbf{s}_1 + \mathbf{s}_1^\top \mathbf{R}_{1,m} \mathbf{m} + \mathbf{m}^\top \mathbf{R}_{m,m} \mathbf{m}$ with only an upper triangular matrix $\mathbf{R}_2 := \begin{bmatrix} \mathbf{R}_{1,1} & \mathbf{R}_{1,m} \\ \mathbf{0}^{\ell \times m_1} & \mathbf{R}_{m,m} \end{bmatrix} \in \mathcal{R}_p^{(m_1+\ell)\times(m_1+\ell)}$. We extend the decomposition to the linear term $\mathbf{r}_1^\top = \left[\mathbf{r}_{1,1}^\top \parallel \mathbf{r}_{1,m}^\top\right]$.

We define the set $\mathsf{R}_{\mathcal{F}}$ of potentially committed elements satisfying the equations $f_i(\mathbf{s}) = 0$ for each $f_i \in \mathcal{F}$ for a publicly known set of functions $\mathcal{F}$.

$$\mathsf{R}_{\mathcal{F}} := \left\{ \begin{array}{l} \underline{(\mathbf{s}_1, \mathbf{m}, \mathbf{s}_2) :} \\ \\ \Big(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, (\mathbf{t_A}, \mathbf{t_B})\Big) \end{array} \middle| \begin{array}{c} \mathbf{A}_1\tilde{\mathbf{s}}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t_A} \\ \mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t_B} \\ \tilde{\mathbf{s}}_1 = \mathsf{RandomizedEncode}(\mathbf{s}_1, \sigma_1) \\ (\mathbf{s}_1, \mathbf{m}) \in \cap_{f_i \in \mathcal{F}} \mathsf{Ker} f_i \end{array} \right\}.$$

In our case, the RF-Ajtai part does not involve $\mathbf{s}_1$ but $\tilde{\mathbf{s}}_1$, therefore we can not execute the original construction in [LNP22]. Our new procedure consists of performing the decoding protocol during the execution of the function. For a function f_i in $\mathcal{F}$, we construct a new $\tilde{f}_i$ and prove it holds for $\tilde{f}_i$ instead.

Theorem 4.1. *Let $\mathcal{F}$ a set of linear and quadratic functions over $(\mathbf{s}_1, \mathbf{m})$. Then, for every function $f_i \in \mathcal{F}$, the associated $\tilde{f}_i := f_i \circ \big[\mathsf{Decode}(\cdot), \mathsf{id}(\cdot)\big]$ satisfies:*

1. *$\tilde{f}_i$ acts over $\tilde{\mathbf{s}} := (\tilde{\mathbf{s}}_1, \mathbf{m})$, is computable and preserves the f_i structure,*
2. *if $(\tilde{\mathbf{s}}_1, \mathbf{m}) \in \mathsf{Ker}\tilde{f}_i$ then $(\mathbf{s}_1, \mathbf{m}) \in \mathsf{Ker} f_i$,*
3. *if $(\tilde{\mathbf{s}}_1, \mathbf{m}, \mathbf{s_2}) \in \mathsf{R}_{\mathit{RF}\text{-}\mathcal{F}}$ then $(\mathbf{s}_1, \mathbf{m}, \mathbf{s_2}) \in \mathsf{R}_{\mathcal{F}}$.*

$$\mathsf{R}_{\mathit{RF}\text{-}\mathcal{F}} := \left\{ \begin{array}{l} \underline{(\tilde{\mathbf{s}}_1, \mathbf{m}, \mathbf{s}_2) :} \\ \\ \Big(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, (\mathbf{t_A}, \mathbf{t_B})\Big) \end{array} \middle| \begin{array}{c} \mathbf{A}_1\tilde{\mathbf{s}}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t_A} \\ \mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t_B} \\ (\tilde{\mathbf{s}}_1, \mathbf{m}) \in \cap_{f_i \in \mathcal{F}} \mathsf{Ker}\tilde{f}_i \end{array} \right\}.$$

By "preserving the structure", we mean that the function $\tilde{f}$ is still linear resp. quadratic just like f.

The following lemma proves the stability of the structure of the quadratic component performing over a (possibly randomized) encoding.

Lemma 4.1. *Let m, n, u, b, k positive integers. Let $p := b^k + 1$. Let $\mathbf{A} \in \mathbb{Z}^{m \times n}$, $\mathbf{x}_1 \in \mathbb{Z}_p^m$ and $\mathbf{x}_2 \in \mathbb{Z}_p^n$. Then, we have for both $\overline{\mathbf{x}}_1 \in \{\hat{\mathbf{x}}_1, \tilde{\mathbf{x}}_1\}$ and $\overline{\mathbf{x}}_2 \in \{\hat{\mathbf{x}}_2, \tilde{\mathbf{x}}_2\}$:*

$$\mathbf{x}_1^\top \mathbf{A}\mathbf{x}_2 = \overline{\mathbf{x}}_1^\top \Big(\mathbf{A} \otimes \mathbf{g}_{b,k}\Big)\mathbf{x}_2,\ \mathbf{x}_1^\top \mathbf{A}\mathbf{x}_2 = \mathbf{x}_1^\top \Big(\mathbf{A} \otimes \mathbf{g}_{b,k}^\top\Big)\overline{\mathbf{x}}_2$$
$$\textit{and } \mathbf{x}_1^\top \mathbf{A}\mathbf{x}_2 = \overline{\mathbf{x}}_1^\top \Big(\mathbf{A} \otimes (\mathbf{g}_{b,k}^\top \otimes \mathbf{g}_{b,k})\Big)\overline{\mathbf{x}}_2.$$

Proof. First, we have $\mathbf{G}\hat{\mathbf{t}} = \mathbf{G}\tilde{\mathbf{t}}$ because $\tilde{\mathbf{t}} = \hat{\mathbf{t}} \mod \mathfrak{p}$. The second equality comes from the linear transformation and the associativity of the matrix product. For the first one, the equality comes from $\mathbf{x}_1^\top \mathbf{A}\mathbf{x}_2 = (\mathbf{A}^\top \mathbf{x}_1)^\top \cdot \mathbf{x}_2 = \Big((\mathbf{A}^\top \otimes \mathbf{g}_{b,k}^\top) \cdot \overline{\mathbf{x}}_1\Big)^\top \cdot \mathbf{x}_2 = \overline{\mathbf{x}}_1^\top \cdot \Big(\mathbf{A} \otimes \mathbf{g}_{b,k}\Big) \cdot \mathbf{x}_2$. The generalized equality is due to the two previous equalities and the associativity of the Kronecker product. □

Proof (of Theorem 4.1). We have $\tilde{f}_i(\tilde{\mathbf{s}}_1, \mathbf{m}) = f_i(\mathbf{G}_{b,k,dm_1}\tilde{\mathbf{s}}_1, \mathbf{m}) = f_i(\mathbf{s}_1, \mathbf{m}) = 0$, implying the second point and the computability. The third point is clearly implied by the second point. For the initial point, we mean that if f is linear (resp.

quadratic), then $\tilde{f}$ is still linear (resp. quadratic). As identifying f_i as a quadratic construction showed in remark 4.1, we construct $\tilde{f}(\tilde{\mathbf{s}}) = \tilde{\mathbf{s}}^\top \mathbf{R}'_2 \tilde{\mathbf{s}} + \mathbf{r}'^\top_1 \tilde{\mathbf{s}} + r_0$ with:

$$\mathbf{r}'^\top_1 = \left[\mathbf{r}^\top_{1,1} \otimes \mathbf{g}^\top_{b,k} \parallel \mathbf{r}^\top_{1,m}\right] \text{ and } \mathbf{R}'_2 := \begin{bmatrix} \mathbf{R}_{1,1} \otimes (\mathbf{g}^\top_{b,k} \otimes \mathbf{g}_{b,k}) & \mathbf{R}_{1,m} \otimes \mathbf{g}_{b,k} \\ \mathbf{0}^{\ell \times km_1} & \mathbf{R}_{m,m} \end{bmatrix},$$

that maintains a quadratic structure over $\tilde{\mathbf{s}}$. For the linear structure, we study the quadratic component; if $\mathbf{R}_2 = \mathbf{0}^{(m_1+\ell)\times(m_1+\ell)}$, then by construction, we have $\mathbf{R}'_2 = \mathbf{0}^{(km_1+\ell)\times(km_1+\ell)}$ that maintains the linear structure over $\tilde{\mathbf{s}}$. □

4.2 Include the automorphism σ

Our goal is to prove quadratic relations over committed messages and their automorphism $\sigma : X \mapsto X^{-1}$ over $\mathcal{R}_p$. We define the full committed message as $\mathbf{s}^\top := \left(\mathbf{s}_1^\top, \sigma(\mathbf{s}_1)^\top, \mathbf{m}^\top, \sigma(\mathbf{m})^\top\right) \in \mathcal{R}_p^{2(m_1+\ell)}$. Therefore, together with the remark 4.1 and the explanation above, the relation can be seen as proving the equation $f(\mathbf{s}) = \mathbf{s}^\top \mathbf{R}_2 \mathbf{s} + \mathbf{s}^\top \mathbf{r}_1 + r_0 = 0$ with the (public) matrices:

$$\mathbf{R}_2 = \begin{bmatrix} \mathbf{R}_{1,1} & \mathbf{R}_{1,\sigma(1)} & \mathbf{R}_{1,\mathbf{m}} & \mathbf{R}_{1,\sigma(\mathbf{m})} \\ \mathbf{0}^{m_1 \times m_1} & \mathbf{R}_{\sigma(1),\sigma(1)} & \mathbf{R}_{\sigma(1),\mathbf{m}} & \mathbf{R}_{\sigma(1),\sigma(\mathbf{m})} \\ \mathbf{0}^{\ell \times m_1} & \mathbf{0}^{\ell \times m_1} & \mathbf{R}_{\mathbf{m},\mathbf{m}} & \mathbf{R}_{\mathbf{m},\sigma(\mathbf{m})} \\ \mathbf{0}^{\ell \times m_1} & \mathbf{0}^{\ell \times m_1} & \mathbf{0}^{\ell \times \ell} & \mathbf{R}_{\sigma(\mathbf{m}),\sigma(\mathbf{m})} \end{bmatrix} \in \mathcal{R}_p^{2(m_1+\ell)\times 2(m_1+\ell)}, \quad (1)$$

$$\mathbf{r}_1^\top = \begin{bmatrix} \mathbf{r}^\top_{1,1} & \mathbf{r}^\top_{1,\sigma(1)} & \mathbf{r}^\top_{1,\mathbf{m}} & \mathbf{r}^\top_{1,\sigma(\mathbf{m})} \end{bmatrix} \in \mathcal{R}_p^{2(m_1+\ell)}. \quad (2)$$

Remark 4.2 ([LNP22, Section 4]). A naive idea is to commit both the message and its automorphism. To be more efficient, the action of σ can be computed from $(\mathbf{s}_1, \mathbf{m})$ and its commitment. It is an automorphism of $\mathcal{R}_p$, so when defining σ over an element $\mathcal{R}_p^k$, we compute it coordinate-wise. We show that the automorphism of an element can be computed based on its (randomized) encoding.

Lemma 4.2. *Let $\sigma \in \mathsf{Aut}(\mathcal{R}_p)$ and $\mathbf{x} \in \mathcal{R}_p^{m_1}$ a secret, and $\overline{\mathbf{x}} \in \{\hat{\mathbf{x}}, \tilde{\mathbf{x}}\}$ seen as an element of $\mathcal{R}_p^{km_1}$ its encoding (resp. randomized encoding) of $\mathbf{x}$. Then, we have for each $\overline{\mathbf{x}} \in \{\hat{\mathbf{x}}, \tilde{\mathbf{x}}\}$: $\overline{\sigma(\mathbf{x})} = \sigma(\overline{\mathbf{x}})$ decoded in a unique $\sigma(\mathbf{x})$.*

The proof relies on the specififc Encode algorithm chosen (described in section 3). The proof is straightforward and is done in Appendix B.1.

Theorem 4.2. *Let $\mathcal{G}$ a set of linear and quadratic functions over $\mathbf{s}^\top$. Define the map $\psi : (\mathbf{x}_1, \mathbf{x}_2) \longmapsto (\mathbf{x}_1, \sigma(\mathbf{x}_1), \mathbf{x}_2, \sigma(\mathbf{x}_2))$. Then, for every function $g_i \in \mathcal{G}$, the associated function $\tilde{g}_i := g_i \circ \left[\mathsf{Decode}(\cdot), \mathsf{Decode}(\cdot), \mathsf{id}(\cdot), \mathsf{id}(\cdot)\right]$ satisfies:*

1. *$\tilde{g}_i$ acts over $\tilde{\mathbf{s}} := \psi(\tilde{\mathbf{s}}_1, \mathbf{m})$, is computable and preserves the g_i structure,*
2. *if $\psi(\tilde{\mathbf{s}}_1, \mathbf{m}) \in \mathsf{Ker} \tilde{g}_i$ then $\psi(\mathbf{s}_1, \mathbf{m}) \in \mathsf{Ker} g_i$,*
3. *if $(\tilde{\mathbf{s}}_1, \mathbf{m}, \mathbf{s_2}) \in \mathsf{R}_{\mathsf{RF\text{-}AUT\text{-}}\mathcal{G}}$ then $\left((\mathbf{s}_1^\top \| \sigma(\mathbf{s}_1)^\top), (\mathbf{m}^\top \| \sigma(\mathbf{m})^\top), \mathbf{s}_2^\top\right)^\top \in \mathsf{R}_{\mathcal{G}}$.*

$$\mathsf{R}_{\mathit{RF\text{-}AUT\text{-}\mathcal{G}}} := \left\{ \begin{array}{c} \underline{(\tilde{\mathbf{s}}_1, \mathbf{m}, \mathbf{s}_2) :} \\ \Big(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, (\mathbf{t_A}, \mathbf{t_B})\Big) \end{array} \middle| \begin{array}{c} \mathbf{A}_1\tilde{\mathbf{s}}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t_A} \\ \mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t_B} \\ \psi(\tilde{\mathbf{s}}_1, \mathbf{m}) \in \cap_{g_i \in \mathcal{G}} \mathsf{Ker}\tilde{g}_i \end{array} \right\}.$$

Proof. For 2 and 3, as $\tilde{g}_i(\psi(\tilde{\mathbf{s}}_1, \mathbf{m})) = g_i\big(\mathbf{G}_{b,k,dm_1}\tilde{\mathbf{s}}_1, \mathbf{G}_{b,k,dm_1}\sigma(\tilde{\mathbf{s}}_1), \mathbf{m}, \sigma(\mathbf{m})\big)$, we have $\tilde{g}_i(\psi(\tilde{\mathbf{s}}_1, \mathbf{m})) = g_i(\mathbf{s}_1, \sigma(\mathbf{s}_1), \mathbf{m}, \sigma(\mathbf{m})) = g_i(\psi(\mathbf{s}_1, \mathbf{m})) = 0$. We keep the same analysis done in Theorem 4.1 as the decoding and the identity are done coordinate-wise leading to a function with input $\big((\mathbf{s}_1^\top \| \sigma(\mathbf{s}_1)^\top), (\mathbf{m}^\top \| \sigma(\mathbf{m})^\top), \mathbf{s}_2^\top\big)^\top$. We can now see $\tilde{g}_i$ as $\tilde{g}_i(\tilde{\mathbf{s}}) = \tilde{\mathbf{s}}^\top \mathbf{R}'_2 \tilde{\mathbf{s}} + \mathbf{r}'^\top_1 \tilde{\mathbf{s}} + r_0$ with $\mathbf{R}'_2 \in \mathcal{R}_p^{2(km_1+\ell)\times 2(km_1+\ell)}$ such that:

$$\mathbf{R}'_2 := \begin{bmatrix} \mathbf{R}_{1,1} \otimes (\mathbf{g}_{b,k}^\top \otimes \mathbf{g}_{b,k}) & \mathbf{R}_{1,\sigma(1)} \otimes (\mathbf{g}_{b,k}^\top \otimes \mathbf{g}_{b,k}) & \mathbf{R}_{1,\mathbf{m}} \otimes \mathbf{g}_{b,k} & \mathbf{R}_{1,\sigma(\mathbf{m})} \otimes \mathbf{g}_{b,k} \\ \mathbf{0}^{km_1 \times km_1} & \mathbf{R}_{\sigma(1),\sigma(1)} \otimes (\mathbf{g}_{b,k}^\top \otimes \mathbf{g}_{b,k}) & \mathbf{R}_{\sigma(1),\mathbf{m}} \otimes \mathbf{g}_{b,k} & \mathbf{R}_{\sigma(1),\sigma(\mathbf{m})} \otimes \mathbf{g}_{b,k} \\ \mathbf{0}^{\ell \times km_1} & \mathbf{0}^{\ell \times km_1} & \mathbf{R}_{\mathbf{m},\mathbf{m}} & \mathbf{R}_{\mathbf{m},\sigma(\mathbf{m})} \\ \mathbf{0}^{\ell \times km_1} & \mathbf{0}^{\ell \times km_1} & \mathbf{0}^{\ell \times \ell} & \mathbf{R}_{\sigma(\mathbf{m}),\sigma(\mathbf{m})} \end{bmatrix},$$

$$\text{and } \mathbf{r}'^\top_1 := \begin{bmatrix} \mathbf{r}_{1,1}^\top \otimes \mathbf{g}_{b,k}^\top & \mathbf{r}_{1,\sigma(1)}^\top \otimes \mathbf{g}_{b,k}^\top & \mathbf{r}_{1,\mathbf{m}}^\top & \mathbf{r}_{1,\sigma(\mathbf{m})}^\top \end{bmatrix} \in \mathcal{R}_p^{2(km_1+\ell)}. \tag{3}$$

Regarding the computability, it exploits Lemma 4.2: the (randomized) encoding, the (unique) decoding, and the automorphism σ_{-1} are commutative. □

In order to prove the security of the opening process, we define the associated opening relation. In particular, we choose parameters such that if $(\overline{\mathbf{s}}_1, \overline{\mathbf{s}}_2, \overline{c}) \in \overline{\mathsf{R}}_{\mathsf{RF\text{-}AUT\text{-}\mathcal{G}}}$ then $(\overline{\mathbf{s}}_1, \mathbf{t_B} - \mathbf{B}\overline{\mathbf{s}}_2, \overline{\mathbf{s}}_2) \in \mathsf{R}_{\mathsf{RF\text{-}AUT\text{-}\mathcal{G}}}$ with overwhelming probability for:

$$\overline{\mathsf{R}}_{\mathsf{RF\text{-}AUT\text{-}\mathcal{G}}} := \left\{ \begin{array}{c} \underline{(\overline{\mathbf{s}}_1, \overline{\mathbf{s}}_2, \overline{c}) :} \\ \Big(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, (\mathbf{t_A}, \mathbf{t_B})\Big) \end{array} \middle| \begin{array}{c} \mathbf{A}_1\overline{\mathbf{s}}_1 + \mathbf{A}_2\overline{\mathbf{s}}_2 = \mathbf{t_A} \\ \overline{\mathbf{m}} = \mathbf{t_B} - \mathbf{B}\overline{\mathbf{s}}_2 \\ \psi(\overline{\mathbf{s}}_1, \overline{\mathbf{m}}) \in \cap_{g_i \in \mathcal{G}} \mathsf{Ker}\tilde{g}_i \end{array} \right\}.$$

4.3 Proving the relations

One relation holds. We show in Theorem 4.2 that we manage to preserve the structure of the original relations f when constructing $\tilde{f}$. Therefore, the idea behind the preservation was to keep using the same construction from [LNP22, Section 4.1]. Now that we can compute the automorphism over the encoding by Lemma 4.2, we define $\tilde{\mathbf{s}} := \psi(\tilde{\mathbf{s}}_1, \mathbf{m})$ and $\mathbf{y} := \psi(\tilde{\mathbf{y}}_1, -\mathbf{B}\mathbf{y}_2)$ both in $\mathcal{R}_p^{2(km_1+\ell)}$. We also construct the response $\mathbf{z}_m$ (computable by the verifier) as in [LNP22] by defining $\mathbf{z}_m := c\mathbf{t_B} - \mathbf{B}\mathbf{z}_2 = c \cdot \mathbf{m} - \mathbf{B}\mathbf{y}_2$. We have:

$$\mathbf{z} := \psi\big(\mathbf{z}_1, \mathbf{z}_m\big) = c \begin{bmatrix} \tilde{\mathbf{s}}_1 \\ \sigma(\tilde{\mathbf{s}}_1) \\ \mathbf{m} \\ \sigma(\mathbf{m}) \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{y}}_1 \\ \sigma(\tilde{\mathbf{y}}_1) \\ -\mathbf{B}\mathbf{y}_2 \\ \sigma(-\mathbf{B}\mathbf{y}_2) \end{bmatrix} = c \begin{bmatrix} \tilde{\mathbf{s}}_1 \\ \widetilde{\sigma(\mathbf{s}_1)} \\ \mathbf{m} \\ \sigma(\mathbf{m}) \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{y}}_1 \\ \sigma(\tilde{\mathbf{y}}_1) \\ -\mathbf{B}\mathbf{y}_2 \\ \sigma(-\mathbf{B}\mathbf{y}_2) \end{bmatrix} = c\tilde{\mathbf{s}} + \mathbf{y}.$$

The idea of [LNP22, Section 4.1] is to prove that a coefficient vanishes, reducing the problem to a linear relation. The quadratic relation $\tilde{f}$ is now modified in order to have $\mathbf{z}^\top \mathbf{R}'_2 \mathbf{z} + c\mathbf{z}^\top \mathbf{r}'_1 + c^2 r_0 - c \cdot \mathfrak{g}_1 - \mathfrak{g}_0 = c^2 \tilde{f}(\mathbf{s})$, with two garbage polynomials $\mathfrak{g}_0 := \mathbf{y}^\top \mathbf{R}'_2 \mathbf{y} \in \mathcal{R}_p$ and $\mathfrak{g}_1 := \tilde{\mathbf{s}}^\top \mathbf{R}'_2 \mathbf{y} + \mathbf{y}^\top \mathbf{R}'_2 \tilde{\mathbf{s}} + \mathbf{y}^\top \mathbf{r}'_1 \in \mathcal{R}_p$. The prover finally commits $\mathfrak{g}_1$ by computing $t_1 = \mathbf{b}_1^\top \mathbf{s}_2 + \mathfrak{g}_1$ (in the BDLOP part) and transmits directly $\mathfrak{g}_0$ as it is not linked with any secret but only the randomness $\mathbf{s}_2$ freshly sampled in each execution of the protocol. The verifier checks if $\mathbf{z}^\mathsf{T} \mathbf{R}'_2 \mathbf{z} + c\mathbf{z}^\mathsf{T} \mathbf{r}'_1 + c^2 r_0 \stackrel{?}{=} ct_1 + \mathfrak{g}_0 - \mathbf{b}_1^\mathsf{T} \mathbf{z}_2$ implying $\tilde{f}(\mathbf{s}) = 0$.

Prove many relations. We simply generalize to the proof of $N = |\mathcal{F}|$ relations using the exact same construction as in [LNP22, Section 4.2].

Lemma 4.3 (Induced by [LNP22, Lemma 4.3]). *Consider a set of relations $\mathcal{F} = \{f_i\}_{i \in [N]}$. Define $f := \sum_{i \in [N]} \mu_i f_i$ with $\mu = (\mu_i)_{i \in [N]} \xleftarrow{\$} \mathcal{R}_p^N$. Then,*

$$\mathbb{P}\left[(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}) \notin \overline{\mathsf{R}}_{(\mathsf{RF\text{-}AUT\text{-}})\mathcal{F}} \mid (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}) \in \overline{\mathsf{R}}_{(\mathsf{RF\text{-}AUT\text{-}})\{f\}}\right] \leq p^{-\frac{d}{2}}.$$

We give the protocol in Figure 2 with the following theorems ensuring security.

Theorem 4.3 (Completeness). *The protocol instantiated in Figure 2 satisfies completeness if it satisfies conditions/constructions from Theorems 3.3 and 4.2.*

Proof. The proof of completeness follows from the explanations in Section 4 as the elements transmitted allows the right construction in order to prove $f(\mathbf{s}) = 0$ for each function in $\mathcal{F}$. The last thing to check is that the commitment scheme used is relevant and ensure completeness: we simply launch parameters and conditions from Theorems 3.3 and 4.2. □

Theorem 4.4 (Knowledge Soundness).

There exists an extractor of the protocol in Figure 2 that either produces an opening $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}) \in \overline{\mathsf{R}}_{\mathsf{RF\text{-}AUT\text{-}}\mathcal{F}}$ with non-negligible probability or solves an instance of a $\mathsf{MSIS}_{\mathcal{R},p,n,km_1+m_2,8\eta\sqrt{\beta^2_{\mathsf{open},1}+\beta^2_{\mathsf{open},2}}}$.

Proof. Let's assume that the extractor finds two accepted transcripts for a same commitment $(\mathbf{t_A}, \mathbf{t_B})$ with a rewind before the sampling of the challenge $(\mu, \mathbf{w}, \mathfrak{g}_0, t_1, c, \tilde{\mathbf{z}}_1, \mathbf{z}_2)$ and $(\mu', \mathbf{w}, \mathfrak{g}_0, t_1, c', \tilde{\mathbf{z}}'_1, \mathbf{z}'_2)$. Regarding the elements involved in the proof of opening knowledge, the proof has already been shown in Theorem 3.4. The final part involves proving that the process of proving relationships cannot be compromised. Consider μ not involved as we can reduce the multiple relation to a single one by launching Lemma 4.3 by setting $p^{\frac{d}{2}}$ being negligible. The proof follows essentially the one from [LNP22, Lemma 4.2] as it has the same structure, taking into account 3 rewinds in order to prove $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}) \in \mathsf{R}_{\mathsf{RF\text{-}AUT\text{-}}\{f\}}$. □

Theorem 4.5 (Simulatability). *For the simulatability of the Figure 2, we consider the parameters from Theorems 3.3,3.4 and 3.5 and the simulator in Figure 3 ensuring indistinguishability between an honest transcript π generated from elements of $\mathsf{R}_{\mathsf{RF\text{-}AUT\text{-}}\mathcal{F}}$ and an output of the simulator $\mathcal{S}_{\mathsf{REL\text{-}AUT\text{-}}\mathcal{F}}$.*

Proof. We immediately use Theorem 3.5 which defines the simulatability of RF-ABDLOP based on Hint-MLWE for the response $\mathbf{z}_2$. The right construction of the additional $(\mu, \mathfrak{g}_0, t_1)$ follows from the correctness and the fact that $\overline{\mathcal{C}} \subseteq \mathcal{R}_q^\times$. □

5 Instantiations of the framework

We provide a library in `C` that uses and extends the actual LaZer library [LSS24], with a set of new rejection-free functions. We provide a concrete analysis of the running time of our code with the existing [LNP22] framework launched with the [LSS24] implementation. We obtain the results shown in Table 3 made with a compilation of our code on a laptop with Intel Core Ultra 7 165H, using `gcc version 11.4.0` with the options `-O3 -march=native`, `-mtune=native` and `-pthread` to ensure optimality of the results. We recall the list of conditions our framework needs to follow to ensure security and compare the running time and the size of the transcript of our framework with the original ones of [LNP22].

5.1 Overview of all the conditions

First, the framework can be instantiated for a particular prime p such that $p = 5 \mod 8$ in order to use Lemma 2.5 but since $p = b^k + 1$, this can only be done for $k = 2$. Second, the false-positive probability in Lemma 4.3 must be negligible. Third, we must ensure that our standard deviations satisfy the initial conditions mentioned in the security Theorems. By taking $\varepsilon = 2^{-\lambda}$ with the security parameter λ and as $\lambda_{kmd}(\widehat{\mathcal{R}}^m) = 1$ and $\lambda_{kmd}(\mathfrak{p}_m\widehat{\mathcal{R}}^m) = \sqrt{b^2+1} = \sqrt{p}$ for any m, we bound the smoothing parameters using [MR07, Lemma 3.3]. Overall, the parameters of the protocol depend on the size m_1, the degree d, the MSIS dimension n, and the size m_2 which is adjusted to achieve the MLWE security. As our security is based on MSIS and MLWE (through Hint-MLWE), we exploit the LWE-Estimator [APS15] to compute m_2 and n, ensuring the hardness of:

- $\mathsf{MSIS}_{\mathcal{R},p,n,km_1+m_2,8\eta\sqrt{(\eta(b+1)\sigma_1+\sqrt{2}\mathfrak{s}_1)^2(km_1d)+(\eta\sigma_2+\mathfrak{s}_2)^2(m_2d)}}$ (Know. Sound.),
- $\mathsf{MSIS}_{\mathcal{R},p,n,km_1+m_2,2\sqrt{(b+1)^2\sigma_1^2m_1kd+\sigma_2^2m_2d}}$ (Binding),
- $\mathsf{MLWE}_{\mathcal{R},p,n+\ell,m_2,\sigma_2}$ (Hiding) and $\mathsf{MLWE}_{\mathcal{R},p,n+\ell,m_2,\varsigma_2}$ (Simulatability).

5.2 Sizes and running time of the framework

Size of an instance of our framework. The optimisations are the same as in [LNP22]. We use the Dilithium-based compression from [LNP22, Appendix A.2], with variables D and γ cutting low-order bits for the transmitted elements $\mathbf{t_A}$ and $\mathbf{w}$. We use Huffman coding to optimise the transcript size regarding the distribution of the responses $\mathbf{z}_i$. As the set of relations is public and unaffecting the extractor, the challenge μ can be the output of the hash-function of the previous transmissions. Unlike the challenge c, that is transmitted and quantified

by the bounds η and κ defining $\mathcal{C}$ in Section 2.3. The rest of the size depends on the ν BDLOP commits used dynamically. Finally, a transcript π has length:

$$\begin{aligned}|\pi| := \; & nd(\lceil \log p \rceil + 2.25 - D) + \nu d \cdot \lceil \log p \rceil + d \cdot \lceil \log(2\kappa + 1) \rceil \\ & + km_1 d \cdot \left(2.57 + \left\lceil \log(\eta(b+1)\sigma_1 + \sqrt{2}\mathfrak{s}_1) \right\rceil\right) + m_2 d \cdot (2.57 + \lceil \log(\eta\sigma_2 + \mathfrak{s}_2) \rceil)\,.\end{aligned}$$

Complexity of the proving phase. We study the complexity of the commitment. The prover needs to construct the elements $\tilde{\mathbf{s}}_1$ by encoding it, samples an error $\mathbf{e}$, and adds it. This is done in $\mathcal{O}(n(km_1 + m_2))$ operations over $\mathcal{R}_p$ considering optimal methods of sampling from spherical discrete Gaussian from [HSS24, Section 3.2]. The proof is constructed with $\mathcal{O}(n(km_1+m_2)+k^2m_1^2)$ operations over $\mathcal{R}_p$ because the computations are bounded by $\mathbf{w}$ or $\mathfrak{g}_i$. If we consider that the multiplication in $\mathcal{O}(d \log d)$ is the heavier operation in $\mathcal{R}$, then the prover complexity is $\mathcal{O}\big(d \log d(n(km_1 + m_2) + k^2m_1^2)\big)$ only enhancing the running time of a [LNP22] instance by a factor at most k^2. But, we are restricted to the case $k = 2$. Then by denoting $T_{\mathcal{P},\mathsf{LNP}}$ the initial time of a single instance of the proving phase (without rejection), we can bound the proving phase of our protocol: $T_{\mathcal{P},\mathsf{new}} \leq 4T_{\mathcal{P},\mathsf{LNP}}$. Then, our framework executes in constant time and has a faster proving phase if the rejection rate is higher than 3.

Complexity of the verifying phase. With the same analysis, the verifier complexity is $\mathcal{O}\big(d \log d(n(km_1 + m_2) + k^2m_1^2)\big)$, majored by the checks regarding $\mathbf{w}$ or v. We bound the running time of the prover by $T_{\mathcal{V},\mathsf{new}} \leq 4T_{\mathcal{V},\mathsf{LNP}}$.

We manage to show a comparison of the size of the transcripts and the evolution of running time with [LNP22] in the Table 3 for 5 parameter sets defined in the Figure 2.

Acknowledgments. We thank previous reviewers of this work for their valuable feedback. This work was published as an extended abstract version ([DR25]). This work is supported by the PEPR quantique France 2030 program (ANR-22-PETQ-0008 PQ-TLS) and by the ASTRID program under the national project AMIRAL with reference ANR-21-ASTR-0016. Antoine Douteau is partially funded by the region of Normandy, France.

References

AAB+22. C. Aguilar-Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, E. Persichetti, G. Zémor, J. Bos, A. Dion, J. Lacan, J.-M. Robert, and P. Veron. HQC. Technical report, National Institute of Standards and Technology, 2022. available at `https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions`.

ABB+22. N. Aragon, P. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Gueron, T. Guneysu, C. Aguilar-Melchor, R. Misoczki, E. Persichetti, N. Sendrier, J.-P. Tillich, G. Zémor, V. Vasseur, S. Ghosh,

and J. Richter-Brokmann. BIKE. Technical report, National Institute of Standards and Technology, 2022. available at `https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions`.

Ajt96. M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *28th ACM STOC*, pages 99–108. ACM Press, May 1996.

Ajt98. M. Ajtai. The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). In *30th ACM STOC*, pages 10–19. ACM Press, May 1998.

AKSY22. S. Agrawal, E. Kirshanova, D. Stehlé, and A. Yadav. Practical, round-optimal lattice-based blind signatures. In *ACM CCS 2022*, pages 39–53. ACM Press, November 2022.

APS15. M. R. Albrecht, R. Player, and S. Scott. On the concrete hardness of learning with errors. In *Journal of Mathematical Cryptology*, pages 169–203, 10 2015.

BBD+23. M. Barbosa, G. Barthe, C. Doczkal, J. Don, S. Fehr, B. Grégoire, Y.-H. Huang, A. Hülsing, Y. Lee, and X. Wu. Fixing and mechanizing the security proof of Fiat-Shamir with aborts and Dilithium. In *CRYPTO 2023, Part V*, *LNCS* 14085, pages 358–389. Springer, Cham, August 2023.

BBE+18. G. Barthe, S. Belaïd, T. Espitau, P.-A. Fouque, B. Grégoire, M. Rossi, and M. Tibouchi. Masking the GLP lattice-based signature scheme at any order. In *EUROCRYPT 2018, Part II*, *LNCS* 10821, pages 354–384. Springer, Cham, April / May 2018.

BCM21. R. Behnia, Y. Chen, and D. Masny. On removing rejection conditions in practical lattice-based signatures. In *PQCrypto 2021*, pages 380–398. Springer, Cham, 2021.

BDK+18. J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy*, pages 353–367. IEEE Computer Society Press, April 2018.

BDK+23. W. Beullens, S. Dobson, S. Katsumata, Y.-F. Lai, and F. Pintore. Group signatures and more from isogenies and lattices: generic, simple, and efficient. *DCC*, 91(6):2141–2200, 2023.

BDL+18. C. Baum, I. Damgård, V. Lyubashevsky, S. Oechsner, and C. Peikert. More efficient commitments from structured lattice assumptions. In *SCN 18*, *LNCS* 11035, pages 368–385. Springer, Cham, September 2018.

BKP20. W. Beullens, S. Katsumata, and F. Pintore. Calamari and Falafl: Logarithmic (linkable) ring signatures from isogenies and lattices. In *ASIACRYPT 2020, Part II*, *LNCS* 12492, pages 464–492. Springer, Cham, December 2020.

BLNS23. J. Bootle, V. Lyubashevsky, N. K. Nguyen, and A. Sorniotti. A framework for practical anonymous credentials from lattices. In *CRYPTO 2023, Part II*, *LNCS* 14082, pages 384–417. Springer, Cham, August 2023.

CGL+24. J.-S. Coron, F. Gérard, T. Lepoint, M. Trannoy, and R. Zeitoun. Improved high-order masked generation of masking vector and rejection sampling in Dilithium. *IACR TCHES*, 2024(4):335–354, 2024.

CLOS02. R. Canetti, Y. Lindell, R. Ostrovsky, and A. Sahai. Universally composable two-party and multi-party secure computation. In *34th ACM STOC*, pages 494–503. ACM Press, May 2002.

DDLL13. L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky. Lattice signatures and bimodal Gaussians. In *CRYPTO 2013, Part I*, *LNCS* 8042, pages 40–56. Springer, Berlin, Heidelberg, August 2013.

DFPS22. J. Devevey, O. Fawzi, A. Passelègue, and D. Stehlé. On rejection sampling in Lyubashevsky's signature scheme. In *ASIACRYPT 2022, Part IV*, *LNCS* 13794, pages 34–64. Springer, Cham, December 2022.

DFPS23. J. Devevey, P. Fallahpour, A. Passelègue, and D. Stehlé. A detailed analysis of Fiat-Shamir with aborts. In *CRYPTO 2023, Part V*, *LNCS* 14085, pages 327–357. Springer, Cham, August 2023.

DG19. L. De Feo and S. D. Galbraith. SeaSign: Compact isogeny signatures from class group actions. In *EUROCRYPT 2019, Part III*, *LNCS* 11478, pages 759–789. Springer, Cham, May 2019.

DKL+18. L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR TCHES*, 2018(1):238–268, 2018.

DPS23. J. Devevey, A. Passelègue, and D. Stehlé. G+G: A Fiat-Shamir lattice signature based on convolved gaussians. In *ASIACRYPT 2023, Part VII*, *LNCS* 14444, pages 37–64. Springer, Singapore, December 2023.

DR25. A. Douteau and A. Roux-Langlois. Rejection-free framework of zero-knowledge proof based on hint-MLWE. In *INDOCRYPT 2025*, *LNCS* 16372, pages 119–144. Springer, Cham, December 2025.

EFGT17. T. Espitau, P.-A. Fouque, B. Gérard, and M. Tibouchi. Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongSwan and electromagnetic emanations in microcontrollers. In *ACM CCS 2017*, pages 1857–1874. ACM Press, October / November 2017.

ENP24. T. Espitau, G. Niot, and T. Prest. Flood and submerse: Distributed key generation and robust threshold signature from lattices. In *CRYPTO 2024, Part VII*, *LNCS* 14926, pages 425–458. Springer, Cham, August 2024.

FS87. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO'86*, *LNCS* 263, pages 186–194. Springer, Berlin, Heidelberg, August 1987.

GHJ+22. Q. Guo, C. Hlauschek, T. Johansson, N. Lahr, A. Nilsson, and R. L. Schröder. Don't reject this: Key-recovery timing attacks due to rejection-sampling in HQC and BIKE. *IACR TCHES*, 2022(3):223–263, 2022.

GMR85. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985.

GMSS99. O. Goldreich, D. Micciancio, S. Safra, and J.-P. Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55–61, 1999.

GPV08. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *40th ACM STOC*, pages 197–206. ACM Press, May 2008.

HSS24. I. Hwang, J. Seo, and Y. Song. Concretely efficient lattice-based polynomial commitment from standard assumptions. In *CRYPTO 2024, Part X*, *LNCS* 14929, pages 414–448. Springer, Cham, August 2024.

ISW03. Y. Ishai, A. Sahai, and D. Wagner. Private circuits: Securing hardware against probing attacks. In *CRYPTO 2003*, *LNCS* 2729, pages 463–481. Springer, Berlin, Heidelberg, August 2003.

KLSS23. D. Kim, D. Lee, J. Seo, and Y. Song. Toward practical lattice-based proof of knowledge from hint-MLWE. In *CRYPTO 2023, Part V*, *LNCS* 14085, pages 549–580. Springer, Cham, August 2023.

LN22. V. Lyubashevsky and N. K. Nguyen. BLOOM: Bimodal lattice one-out-of-many proofs and applications. In *ASIACRYPT 2022, Part IV*, *LNCS* 13794, pages 95–125. Springer, Cham, December 2022.

LNP22. V. Lyubashevsky, N. K. Nguyen, and M. Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In *CRYPTO 2022, Part II*, *LNCS* 13508, pages 71–101. Springer, Cham, August 2022.

LNS21. V. Lyubashevsky, N. K. Nguyen, and G. Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. In *PKC 2021, Part I*, *LNCS* 12710, pages 215–241. Springer, Cham, May 2021.

LS15. A. Langlois and D. Stehlé. Worst-case to average-case reductions for module lattices. *DCC*, 75(3):565–599, 2015.

LSS24. V. Lyubashevsky, G. Seiler, and P. Steuer. The LaZer library: Lattice-based zero knowledge and succinct proofs for quantum-safe privacy. In *ACM CCS 2024*, pages 3125–3137. ACM Press, October 2024.

Lyu09. V. Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *ASIACRYPT 2009*, *LNCS* 5912, pages 598–616. Springer, Berlin, Heidelberg, December 2009.

Lyu12. V. Lyubashevsky. Lattice signatures without trapdoors. In *EUROCRYPT 2012*, *LNCS* 7237, pages 738–755. Springer, Berlin, Heidelberg, April 2012.

MGTF19. V. Migliore, B. Gérard, M. Tibouchi, and P.-A. Fouque. Masking Dilithium - efficient implementation and side-channel evaluation. In *ACNS 2019*, *LNCS* 11464, pages 344–362. Springer, Cham, June 2019.

MP12. D. Micciancio and C. Peikert. Trapdoors for lattices: Simpler, tighter, faster, smaller. In *EUROCRYPT 2012*, *LNCS* 7237, pages 700–718. Springer, Berlin, Heidelberg, April 2012.

MR07. D. Micciancio and O. Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

NS24. N. K. Nguyen and G. Seiler. Greyhound: Fast polynomial commitments from lattices. In *CRYPTO 2024, Part X*, *LNCS* 14929, pages 243–275. Springer, Cham, August 2024.

Pei10. C. Peikert. An efficient and parallel Gaussian sampler for lattices. In *CRYPTO 2010*, *LNCS* 6223, pages 80–97. Springer, Berlin, Heidelberg, August 2010.

PR13. E. Prouff and M. Rivain. Masking against side-channel attacks: A formal security proof. In *EUROCRYPT 2013*, *LNCS* 7881, pages 142–159. Springer, Berlin, Heidelberg, May 2013.

Reg05. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *37th ACM STOC*, pages 84–93. ACM Press, May 2005.

A Simulatability proof

B Additionnal preliminaries

B.1 Structured lattices

We consider a structured case of lattices defined over $\mathcal{R}$. It implies an isomorphism between $\mathcal{R}$ and $\mathbb{Z}^d$. We define the invertible operation Coeffs which

transforms vectors of coefficients into polynomials. Then, the operations can be adapted to the specific case. The sum is intuitive, but the product needs to exploit the rotation matrix associated with the polynomial. For two elements $a = \sum_{0 \leq i < d} a_i X^i \in \mathcal{R}$ and $b = \sum_{0 \leq i < d} b_i X^i \in \mathcal{R}$ where $a_i, b_i \in \mathbb{Z}$, we have:

$$\mathsf{Coeffs}(a \cdot b) = \mathsf{Rot}_d(a) \cdot \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{d-1} \end{bmatrix} \text{ with } \mathsf{Rot}_d(a) := \begin{bmatrix} a_0 & -a_{d-1} & \cdots & -a_1 \\ a_1 & a_0 & \cdots & -a_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{d-1} & a_{d-2} & \cdots & a_0 \end{bmatrix} \in \mathbb{Z}^{d \times d}.$$

It can be generalized to a greater amount of such polynomials. Consider a matrix $\mathbf{A}$ with coefficients in $\mathcal{R}$; simply generate the block matrix composed with rotation matrices of each coefficient.

B.2 Protocols

Interactive proofs. A prover $\mathcal{P}$ wants to prove knowledge to a third party, called Verifier $\mathcal{V}$, without revealing the private information. The idea originally revealed in [GMR85] is to involve $\mathcal{V}$ in the process and interact until they are convinced that $\mathcal{P}$ has the knowledge. The relation proven can be seen as a couple (x,w), with x being a public statement associated to a (private) witness w. We associate with such a proof different notions of security:
Completeness. A proof is complete if the verifier $\mathcal{V}$, with the knowledge of an accepting witness w, produces an accepting transcript (with overwhelming probability) of the protocol with an honest verifier $\mathcal{V}$.
(Knowledge) Soundness. Soundness means that a verifier $\mathcal{V}$ rejects false statements (with overwhelming probability). The case of Knowledge Soundness is a stronger notion and ensures that if a prover (not necessarily honest) convinces a verifier multiple times for the same statement x, then the prover must know (with overwhelming probability) a valid witness w. We defined as an extractor a PPT algorithm exploiting a convincing prover.
Simulatability. The *simulator* $\mathcal{S}$ is a PPT algorithm that produces a transcript π associated with an interactive protocol $(\mathcal{P}, \mathcal{V})$ with the public statement x. We say that the protocol is simulatable if the output generated by $\mathcal{S}$ is indistinguishable from the transcript π of a fully honest execution of the interactive protocol.
Zero-Knowledge proofs. As explained in [KLSS23, LNP22, LNS21], we prefer to define the notion of simulatability, which is a stronger notion than the zero-knowledge property. The zero-knowledge property means that the verifier does not gain any information on the witness w used and known by the prover $\mathcal{P}$. The simulatability means that the verifier cannot distinguish between valid transcript (over public and private information) and simulated one (over only public information).

Commitment Scheme. In order to preserve the zero-knowledge property but also guarantee to $\mathcal{V}$ that $\mathcal{P}$ does not trick him, the notion of commitment scheme

ensures a potentially well-constructed protocol. We recall the definition of a commitment scheme. A commitment scheme is a tuple of three PPT algorithms:

- $\mathbf{Setup}(1^\lambda) \to \mathsf{ck}$: Given a security parameter λ, it generates a key ck,
- $\mathbf{Commit}(\mathsf{ck}, m) \to (c, \delta)$: Given a message m and the commitment key ck, it returns an associated commitment c and an opening δ,
- $\mathbf{Open}(\mathsf{ck}, c, m, \delta) \to \{\bot, \top\}$: Given a commitment key ck, a commitment c, a message m, and an opening δ, it outputs the validity of the opening.

A well-constructed commitment scheme needs to satisfy two properties to ensure its security. The first one is called *binding*, the hardness of opening a single commitment to two distinct initial messages. The second security notion is the *hiding* property. It is the "IND-CPA security" adapted to commitment schemes.

Proof of Opening Knowledge. A "proof of opening knowledge" is the interactive proof of knowledge regarding the process knowledge of the message committed using a commitment scheme.

B.3 The Gaussianisation method from [HSS24]

The [HSS24] method is used to construct an encoding following a Gaussian distribution centered in the secret encoding. First, it computes an encoding version of an element $\mathbf{s} \in \mathcal{R}_p^m$, exploiting the inverse $\mathbf{G}^{-1}$ of the gadget matrix. In a second step, the encoding is randomized by sampling a non-influent error.

Encoding protocol. The encoding protocol treats a full polynomial as a large vector of its coefficients, encoding each coefficient by the signed decomposition in base b (of length k) denoted by $\mathbf{G}_{b,k,d}^{-1}$ and detailed in Algorithm 8.

- $\mathsf{Encode}(\mathbf{t}) \to \hat{\mathbf{t}}$: For an element $\mathbf{t} \in \mathcal{R}_p^m$, we compute $\hat{\mathbf{t}} = \mathbf{G}_{b,k,dm}^{-1}\mathbf{t} \in \mathcal{R}_p^{km}$. The output satisfies $\|\hat{\mathbf{t}}\|_\infty \leq \frac{b+2}{2}$. The action of $\mathbf{G}_{b,k,d}^{-1}\mathbf{t}$ is defined in the Algorithm 8. For any m, we proceed by defining: $\mathbf{G}_{b,k,dm}^{-1} := \mathbf{I}_m \otimes \mathbf{G}_{b,k,d}^{-1}$.
- $\mathsf{Decode}(\hat{\mathbf{t}}) \to \mathbf{t}$: Output the matrix product $\mathbf{t} = \mathbf{G}_{b,k,dm}\hat{\mathbf{t}}$ over $\mathcal{R}_p$.

Randomization step. We recall the randomization process from [HSS24]. The key analysis is that encoded elements are equal modulo $\mathfrak{p}$ when considered as elements of $\widehat{\mathcal{R}}$ i.e. if $\hat{a} = \hat{a}' \mod \mathfrak{p}$, then $\mathsf{Decode}(\hat{a}) = \mathsf{Decode}(\hat{a}')$ over $\mathcal{R}_p$. This property is used to randomize a message over all the possibilities such that it satisfies the same decoding. However, it forces us to restrain the message space as $\mathcal{R}_p$ with a valid p, but it is allowed by the fact that $\|\hat{\mathbf{t}}\|_\infty \leq \frac{b+2}{2} < p$. We generalize to a length m with $P_m := \mathsf{Rot}_{kdm}(\mathfrak{p}_m) = \mathsf{Rot}_{kd}(\mathfrak{p}) \otimes \mathbf{I}_m \in \mathbb{Z}_p^{kdm}$, where we have $\|P_m\| = \|\mathfrak{p}_m\| = b + 1$.

- $\mathsf{RandomizedEncode}(\mathbf{t}, \sigma) \to \tilde{\mathbf{t}}$: Compute $\hat{\mathbf{t}} \leftarrow \mathsf{Encode}(\mathbf{t})$. Then, sample an error $\vec{e} \leftarrow \mathcal{D}_{\mathbb{Z}^{kdm}, \sigma}$. Return $\tilde{\mathbf{t}} = \hat{\mathbf{t}} + \mathsf{Coeffs}^{-1}(P_m\vec{e})$ as an element of $(\mathcal{R}_p^k)^m$,
- $\mathsf{RandomizedDecode}$ is the same as Decode.

During the rest of the paper, we distinguish the two notations related to the encoding of an element $\mathbf{t} \in \mathcal{R}_p^m$: the "classical" encoding is denoted by $\hat{\mathbf{t}}$ and the "randomized" encoding $\tilde{\mathbf{t}} = \hat{\mathbf{t}} + \mathfrak{p}_m \mathbf{e}$ that follows a Gaussian distribution. Both are elements of $\mathcal{R}_p^{km}$.

Algorithm 8: Encoding method from [HSS24].

Input: $\sum_{i=0}^{d-1} a_i X^i \in \mathcal{R}_p$
Output: $a \in \mathcal{R}^k$
for $0 \leq i < d$ **do**
  if $a_i = -1 \mod p$ **then**
    $(a_{i,0}, \ldots, a_{i,k-1}) \leftarrow (0, \ldots, 0, b)$
  else
    $(a_{i,0}, \ldots, a_{i,k-1}) \leftarrow$ the b-representation of $a_i < b^k$
for $0 \leq i < d, 0 \leq j < k$ **do**
  if $a_{i,j} > \frac{b}{2}$ **then**
    $a_{i,j} \leftarrow a_{i,j} - b$; $c_{i,j} \leftarrow 1$
  else
    $c_{i,j} \leftarrow 0$
$\hat{a} \leftarrow \sum_{0 \leq i < d} \sum_{0 \leq j < k} a_{i,j} X^{dj+i} + c_{i,j} X^{d(j+1)+i} \in \widehat{\mathcal{R}}$
Return a as an element of $\mathcal{R}^k$ with the same vector of coefficients of $\hat{a}$

B.4 Bound of the norm of the challenge matrix

We recall the two definition of norm of matrices:

$$\|C\|_1 := \|C\|_{1,1} = \max_{j \in [n]} \sum_{i \in [n]} |a_{i,j}| \text{ and } \|C\|_\infty := \|C\|_{\infty,\infty} = \max_{i \in [n]} \sum_{j \in [n]} |a_{i,j}|.$$

Take a challenge $c \in \mathcal{C}$, and its associated rotated matrix $C := \mathsf{Rot}_n(c)$. For the norm, we have $\|C\|^2 \leq \|C\|_1 \cdot \|C\|_\infty$. We have by definition of the challenge space that $|c| \leq \eta$. Exploiting the structure of the rotation matrix, shown above in Appendix B.1, we have $\|C\|_1 = \|C\|_\infty = \eta$, and therefore the bound $\|C\|_2^2 \leq \eta^2$.

B.5 Communativity between encoding and conjugate automorphism

Lemma B.1 (Lemma 4.2). *Let $\sigma \in \mathsf{Aut}(\mathcal{R}_p)$ and $\mathbf{x} \in \mathcal{R}_p^{m_1}$ a secret, and $\overline{\mathbf{x}} \in \{\hat{\mathbf{x}}, \tilde{\mathbf{x}}\}$ seen as an element of $\mathcal{R}_p^{km_1}$ its encoding (resp. randomized encoding) of $\mathbf{x}$. Then, we have for each $\overline{\mathbf{x}} \in \{\hat{\mathbf{x}}, \tilde{\mathbf{x}}\}$: $\overline{\sigma(\mathbf{x})} = \sigma(\overline{\mathbf{x}})$ decoded in a unique $\sigma(\mathbf{x})$.*

Proof. We initiate the proof by seeing the action of an automorphism over the coefficients. Considering a polynomial $f \in \mathcal{R}$, we denote by f^* its reciprocal polynomial. We have the equality $f^* = X^{d-1}\sigma(f)$ that implies $-X \cdot f^* = \sigma(f)$.

Then, we consider the automorphism as a matrix product over the coefficients seen as $\mathsf{Coeffs}\big(\sigma(f)\big) = \mathsf{Rot}_d(-X) \cdot \mathbf{J_d} \cdot \mathsf{Coeffs}(f) = \begin{bmatrix} 1 & 0 \\ 0 & -\mathbf{J_{d-1}} \end{bmatrix} \cdot \mathsf{Coeffs}(f)$.

In order to prove the Lemma, we first study the case of the $\hat{\mathbf{s}}$ (the encoding without randomization). Suppose without loss of generality (via the concatenation process) that $m_1 = 1$ i.e. $\mathbf{s}$ is composed as a single message in $\mathcal{R}_p$. Then, we use the equality between the decomposition in base b of an element a and its inverse $-a$. Especially here, the decomposition used in the encoding protocol made our coefficients live in $[-\frac{b}{2}, \frac{b}{2}]$ with a carry c:

$$a := \sum_{0 \leq j < k} a_j b^j + c_j b^{j+1} = (a_0 - c_{k-1}) + \sum_{0 \leq j < k-1} (a_j + c_{j-1}) b^j,$$

so we have similarly the same decomposition for $-a$:

$$-a = -(a_0 - c_{k-1}) + \sum_{1 \leq j < k} -(a_j + c_{j-1}) b^j.$$

As noted earlier, we have for any element $t \in \mathcal{R}_p$, the equality $\vec{\sigma}(t) = \mathbf{A}\vec{t}$ by denoting $\mathbf{A} := \mathsf{Rot}_d(-X) \cdot \mathbf{J}_d$. For an element in $\mathcal{R}_p^k$ we involve the Kronecker product to apply the action of σ over the k polynomials such as $\vec{\sigma}(\mathbf{t}) = (\mathbf{I}_k \otimes \mathbf{A})\vec{t}$. Now, we expand the total formula of $\mathsf{Encode}\big(\sigma(t)\big)$ by considering $t = \sum_{i=0}^{d-1} t_i X^i$ and the associated polynomial $\sigma(t) =: t' = \sum_{i=0}^{d-1} t'_i X^i$

$$\mathsf{Encode}(t') = \begin{pmatrix} \big(t'_{0,0} - c'_{0,k-1} & , & t'_{1,0} - c'_{1,k-1} & , \cdots , & t'_{d-1,0} - c'_{d-1,k-1}\big) \\ \big(t'_{0,1} + c'_{0,0} & , & t'_{1,1} + c'_{1,0} & , \cdots , & t'_{d-1,1} + c'_{d-1,0}\big) \\ & & \vdots & & \\ \big(t'_{0,k-1} + c'_{0,k-2} & , & t'_{1,k-1} + c'_{1,k-2} & , \cdots , & t'_{d-1,k-1} + c'_{d-1,k-2}\big) \end{pmatrix},$$

where $\begin{cases} t'_0 = t_0, \\ t'_i = -t_{d-i} \text{ for } i \in [1, d-1]. \end{cases}$

that implies $\begin{cases} \forall j \in [0, k-1], t'_{0,j} = t_{0,j} & \text{and } c'_{0,j} = c_{0,j}, \\ \forall j \in [0, k-1], t'_{i,j} = -t_{d-i,j} & \text{and } c'_{i,j} = -c_{d-i,j}. \end{cases}$

$$\mathsf{Encode}(t') = \begin{pmatrix} \big(t_{0,0} - c_{0,k-1} & , & -(t'_{d-1,0} - c'_{d-1,k-1}) & , \cdots , & -(t'_{1,0} - c'_{1,k-1})\big) \\ \big(t_{0,1} + c_{0,0} & , & -(t_{d-1,1} + c_{d-1,0}) & , \cdots , & -(t_{1,1} + c_{1,0})\big) \\ & & \vdots & & \\ \big(t_{0,k-1} + c_{0,k-2} & , & -(t_{d-1,k-1} + c_{d-1,k-2}) & , \cdots , & -(t_{1,k-1} + c_{1,k-2})\big) \end{pmatrix}. \tag{8}$$

In a second step, we show the action of the σ after the Encoding protocol.

$$\mathsf{Encode}(t) = \begin{pmatrix} \big(t_{0,0} - c_{0,k-1} & , & t_{1,0} - c_{1,k-1} & , \cdots , & t_{d-1,0} - c_{d-1,k-1}\big) \\ \big(t_{0,1} + c_{0,0} & , & t_{1,1} + c_{1,0} & , \cdots , & t_{d-1,1} + c_{d-1,0}\big) \\ & & \vdots & & \\ \big(t_{0,k-1} + c_{0,k-2} & , & t_{1,k-1} + c_{1,k-2} & , \cdots , & a_{d-1,k-1} + c_{d-1,k-2}\big) \end{pmatrix}.$$

Now we apply the automorphism row per row by keeping fixed the constant coefficients and negarotating others.

$$\sigma\Big(\mathsf{Encode}(t)\Big) = \begin{pmatrix} \big(t_{0,0} - c_{0,k-1} \quad , \quad -(t'_{d-1,0} - c'_{d-1,k-1}) \quad , \cdots , \quad -(t'_{1,0} - c'_{1,k-1})\big) \\ \big(t_{0,1} + c_{0,0} \quad , \quad -(t_{d-1,1} + c_{d-1,0}) \quad , \cdots , \quad -(t_{1,1} + c_{1,0})\big) \\ \vdots \\ \big(t_{0,k-1} + c_{0,k-2} \ , -(t_{d-1,k-1} + c_{d-1,k-2}) \ , \cdots , -(t_{1,k-1} + c_{1,k-2})\big) \end{pmatrix}. \tag{10}$$

We now see the equality between the two structures (8) and (10) that implies the equality $\mathbf{G}^{-1}\mathbf{A}\vec{t} = \mathbf{A}\mathbf{G}^{-1}\vec{t}$. By considering m_1 elements $\mathbf{s} \in \mathcal{R}_p^{m_1}$, we have the same property: $(\mathbf{I}_{m_1} \otimes \mathbf{G}^{-1}\mathbf{A})\mathbf{s} = (\mathbf{I}_{m_1} \otimes \mathbf{A}\mathbf{G}^{-1})\mathbf{s}$.

Now, we show that the equality also works over randomized encoding. We remind that we have $\tilde{\mathbf{t}} = \hat{\mathbf{t}} + \mathfrak{p}\mathbf{t}$ for $\mathfrak{p} := X^d - b$. We then view its vector of coefficients as $\mathsf{Rot}_{kd}(X^d - b) \cdot \mathsf{Coeffs}(\mathbf{e})$ with the aim of showing that we have the invariance over the polynomial $\mathfrak{p}$: $\sigma(\tilde{\mathbf{t}}) = \sigma(\hat{\mathbf{t}}) \mod X^d - b$. By using the homomorphic property of the automorphism $\sigma(\tilde{\mathbf{t}}) = \sigma(\hat{\mathbf{t}}) + \sigma(\mathfrak{p}\mathbf{e})$, we just need to prove that $\sigma(\mathfrak{p}\mathbf{e}) = 0 \mod X^d - b$ i.e. a multiple of $X^d - b$. We remind the following Kronecker product property $\mathbf{AC} \otimes \mathbf{BD} = (\mathbf{A} \otimes \mathbf{B}) \cdot (\mathbf{C} \otimes \mathbf{D})$ for well sized matrices. Expanding the matrix $\mathsf{Rot}_{kd}(X^d - b)$, we can easily see that: $\mathsf{Rot}_{kd}(X^d - b) = \mathsf{Rot}_d(X^{\frac{d}{k}} - b) \otimes \mathbf{I}_k = \mathsf{Rot}_k(X - b) \otimes \mathbf{I}_d$. By denoting $\mathbf{B} = \mathsf{Rot}_k(X - b)$ and using Kronecker product properties, we have:

$$\begin{aligned} \mathsf{Coeffs}(\sigma(\mathfrak{p}\mathbf{e})) &= (\mathbf{I}_k \otimes \mathbf{A}) \cdot (\mathbf{B} \otimes \mathbf{I}_d) \cdot \mathsf{Coeffs}(\mathbf{e}) \\ &= (\mathbf{B} \otimes \mathbf{I}_d) \cdot (\mathbf{I}_k \otimes \mathbf{A}) \cdot \mathsf{Coeffs}(\mathbf{e}) \\ &= \mathsf{Rot}_{kd}(X^d - b) \cdot \mathsf{Coeffs}(\sigma(\mathbf{e})) \\ &= \mathsf{Coeffs}(\mathfrak{p}\sigma(\mathbf{e})) \end{aligned}$$

This ensures as well that $\sigma(\mathbf{e})$ using $\mathfrak{p}_m$. This proof is also valid for any message $\mathbf{s} \in \mathcal{R}_p^{m_1}$ using the equality $\vec{\sigma}(\mathbf{s}) = (\mathbf{I}_{m_1} \otimes \mathbf{A})\vec{s}$. □

C Comparison with the original framework

C.1 Details of the different sets of parameters

We chose to maintain the same ring degree $d = 64$ (rather than 128) because it does not significantly impact the overall comparison or ratio between our implementation and that of [LNP22], which relies on [LSS24]. Parameter sets 1 and 2 represent typical executions involving a low number of both bounded and unbounded messages, closely reflecting classical use cases of the framework, but with two different ring moduli. Parameter set 3 simulates an execution with a high number of bounded messages only, while set 4 focuses on a high number of unbounded messages only. Notably, the set 4 was not supported in the original construction without affecting the size of the transcript in $\log(p)$. Finally, parameter set 5 simulates a typical execution involving a high number of messages, reflecting use cases in more complex cryptographic applications such as aggregate signatures.

Sets of parameters						
Variable	**Description**	**Set 1**	**Set 2**	**Set 3**	**Set 4**	**Set 5**
λ	Security parameter	128				
d	Ring degree	64				
$\log p$	Log of the ring moduli	60	120	80	80	60
$m_{1\mathsf{LNP}}$	Number of low-norm messages	10	10	100	0	100
ℓ_{LNP}	Number of unbounded messages	2	2	0	20	20
$m_{1\mathsf{new}}$	Number of messages $= m_{1\mathsf{LNP}} + \ell_{\mathsf{LNP}}$	12	12	100	20	120
N	Number of quadratic equations to prove	3				
Security parameters						
κ	$\|c\|_\infty \leq \kappa$ with $c \in \mathcal{C}$	8				
η	$\|c\mathbf{r}\| \leq \eta\|\mathbf{r}\|$ with $c \in \mathcal{C}$ and $\mathbf{r}$ a vector of $\mathcal{R}$	140				
n	Height of $\mathbf{A}_1$ and $\mathbf{A}_2$	32	38	35	34	34
m_2	Length of the randomness $\mathbf{s}_2$	71	114	86	85	73
Dilithium compression						
γ	Largest parameter to cut low-order bits of $\mathbf{w}$	See in implementation				
D	Number of low-order bits cut from $\mathbf{t_A}$	31	61	42	41	32
Computed parameters						
b	Base decomposition	See in implementation				
p	Prime $p := b^2 + 1$ congruent to 5 modulo 8					
σ_1	SD of the randomized encoding of $\mathbf{s}_1$					
$\sqrt{2}\mathfrak{s}_1$	SD of the randomized encoding of the mask $\mathbf{y}_{\mathsf{rand}}$					
σ_2	SD of the randomness $\mathbf{s}_2$					
$\mathfrak{s}_2$	SD of the mask of $\mathbf{y}_2$					

Table 2: Sets of parameters involved in the proof frameworks. The first section of parameters are manually set, others are either computed using `/scripts` of our code inspired by the LaZer Library or using LWE-Estimator [APS15].

C.2 Comparison table

Size of the transcripts (in kilobytes KB)						
Protocol	Protocol/Phase	Param. 1	Param. 2	Param. 3	Param. 4	Param. 5
[LNP22]	**Opening proof**	13.22	20.77	30.53	×	41.05
	+ 3 Quad.	13.93	21.86	31.31	×	41.68
	+ 3 Eval.	15.16	22.96	32.09	×	42.93
Ours	**Opening proof**	24.23	45.69	103.19	37.33	96.52
	+ 3 Quad.	24.83	46.76	103.95	38.08	97.12
	+ 3 Eval.	25.42	47.82	104.70	38.83	97.72
Mean times (in million cycles)						
[LNP22]	**KeyGen**	7.63	10.32	19.32	×	39.35
	Commit	7.29	20.89	31.33	×	39.42
	Prove	324.22	352.22	4289.68	×	4462.29
	Verify	18.54	39.17	1744.10	×	1629.12
Ours	**KeyGen**	19.01	47.04	66.30	29.55	62.01
	Commit	19.86	86.54	102.47	45.73	67.15
	Prove	59.45	134.26	4248.67	220.98	3961.24
	Verify	41.74	96.79	3897.43	180.31	3624.34

Table 3: The first part of the Table explains the size of the transcripts in kilobytes for different parameter sets explained in Appendix C.1. In the second part, the results show our benchmarks for the mean running time of 100 executions of each phase for an execution of different instances of the protocol for the 5 different sets of parameters detailed in Table 2. The running time of the Prove part of [LNP22] using the [LSS24] implementation takes into account several executions due to rejection sampling.