DISS. ETH NO. 28574

LATTICE-BASED ZERO-KNOWLEDGE PROOFS UNDER A FEW DOZEN KILOBYTES

A dissertation submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

NGOC KHANH NGUYEN
M. Eng., University of Bristol

born on 10 June 1995
citizen of Poland and Vietnam

accepted on the recommendation of

Prof. Dr. Dennis Hofheinz, examiner
Dr. Vadim Lyubashevsky, co-examiner
Prof. Dr. Ron Steinfeld, co-examiner

2022

ABSTRACT

In preparation for the eventual arrival of quantum computers, there has been a significant amount of work to construct quantum-safe cryptographic primitives, as evidenced by the ongoing NIST PQC Standardization. To ensure post-quantum security, the underlying public-key schemes have to be built based on quantum-safe computational hardness assumptions. In this regard, lattice-based primitives appear to be a leading choice. Indeed, the currently most efficient, in terms of size and speed, quantum-safe basic primitives (e.g. signatures and encryption schemes) are based on the hardness of lattice problems with algebraic structure such as Module-SIS and Module-LWE. As a natural next step, lattice-based cryptography can be thus applied to build more advanced primitives such as zero-knowledge arguments.

In this thesis, we present Lantern, a new lattice-based zero-knowledge protocol with short proofs based on the hardness of Module-SIS and Module-LWE problems. In particular, our framework is suitable for proving lattice-related statements, e.g. proving knowledge of a short vector $\vec{s}$ satisfying $A\vec{s} = \vec{t} \bmod q$. At the core of our constructions lies a more direct and more efficient way to prove that $\vec{s}$ has a small Euclidean norm which, unlike in prior works, does not require proving explicitly that each coefficient of $\vec{s}$ is small, nor any conversion to the CRT representation. Instead, we use the observation that the inner product $\langle \vec{r}, \vec{s} \rangle$ between any two vectors $\vec{r}$ and $\vec{s}$ can be made to appear as a constant coefficient of a product (or sum of products) between polynomials which are functions of $\vec{r}$ and $\vec{s}$. Therefore, by using a polynomial product proof system and hiding all but the constant coefficient, we are able to prove knowledge of the inner product of two vectors (or of a vector with itself) modulo q. Using a cheap "approximate range proof", we can then lift the proof to be over $\mathbb{Z}$ instead of $\mathbb{Z}_q$.

Performance-wise, our framework produces proofs of size 13KB for basic statements which are $2-3$X smaller than prior works. Furthermore, the new proof system can be plugged into constructions of various lattice-based privacy-oriented primitives in a black-box manner. As examples, we instantiate a verifiable encryption scheme as well as ring and group signatures which are significantly more compact than previously the best solutions.

ZUSAMMENFASSUNG

In Vorbereitung auf die Ankunft von Quantencomputern wurde viel an der Entwicklung von quantensicheren kryptografischen Primitiven gearbeitet, wie die laufende NIST PQC-Standardisierung zeigt. Um Post-Quantum-Sicherheit zu gewährleisten, müssen die zugrundeliegenden Public-Key-Verfahren auf der Grundlage von quantensicheren Komplexitätsannahmen aufgebaut werden. In dieser Hinsicht scheinen gitterbasierte Primitive eine gute Wahl zu sein. Die derzeit effizientesten quantensicheren Grundprimitive (z. B. Signaturen und Verschlüsselungsverfahren), was die Größe und Geschwindigkeit betrifft basieren auf der Härte von Gitterproblemen mit algebraischer Struktur, z. B. Module-SIS und Module-LWE. Als natürlicher nächster Schritt kann die gitterbasierte Kryptografie daher zur Entwicklung fortgeschrittener Primitive wie Zero-Knowledge-Arguments eingesetzt werden.

In dieser Arbeit stellen wir Lantern vor, ein neues gitterbasiertes Zero-Knowledge-Argument mit kurzen Beweisen, das auf der Härte von Module-SIS und Module-LWE basiert. Insbesondere eignet sich unser Framework für den Beweis von gitterbezogenen Aussagen, z.B. den Beweis der Kenntnis eines kurzen Vektors $\vec{s}$, der $A\vec{s} = \vec{t} \bmod q$ erfüllt. Der Kern unserer Konstruktionen ist ein direkterer und effizienterer Weg, um zu beweisen, dass $\vec{s}$ eine kleine euklidische Norm hat, wofür weder ein Beweis über die Länge jedes einzelnen Koeffizienten von $\vec{s}$, noch eine Umwandlung in die CRT-Darstellung wie in früheren Arbeiten erforderlich ist. Stattdessen verwenden wir die Beobachtung, dass das Skalarprodukt $\langle \vec{r}, \vec{s} \rangle$ zwischen zwei beliebigen Vektoren $\vec{r}$ und $\vec{s}$ als konstanter Koeffizient eines Produkts (oder einer Summe von Produkten) zwischen Polynomen erscheinen kann, die Funktionen von $\vec{r}$ und $\vec{s}$ sind. Indem wir ein Polynom-Produkt-Beweissystem verwenden und alle Koeffizienten bis auf den konstanten Koeffizienten verstecken, können wir die Kenntnis des Skalarprodukts zweier Vektoren (oder eines Vektors mit sich selbst) modulo q beweisen. Mit Hilfe eines relativ kostengünstigen "Approximate Range-Proofs" kann man dann den Beweis über $\mathbb{Z}$ statt $\mathbb{Z}_q$ führen.

In Bezug auf die Leistung erreicht unser Framework etwa $2-3$-mal kleinere Beweisgrößen als frühere Arbeiten für grundlegende Aussagen, wie z.B. den Nachweis der Kenntnis eines Module-LWE Samples. Darüber hinaus kann das neue Beweissystem in Konstruktionen verschiedener git-

terbasierter privatsphärenorientierter Primitive Blackbox-artig integriert werden. Als Beispiele instanziieren wir ein überprüfbares Verschlüsselungsverfahren sowie Ring- und Gruppensignaturen, die wesentlich kompakter sind als die bisher besten Lösungen.

PUBLICATIONS

Articles fully included in the thesis:

- Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General. In CRYPTO 2022.
  (Full version at `https://ia.cr/2022/284`)
- Vadim Lyubashevsky and Ngoc Khanh Nguyen. BLOOM: Bimodal Lattice One-Out-of-Many Proofs and Applications. In ASIACRYPT 2022.

Publications, which directly led to the main results in this thesis, and thus are partially included:

- Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. Practical Exact Proofs from Lattices: New Techniques to Exploit Fully-Splitting Rings. In ASIACRYPT 2020.
  (Full version at `https://ia.cr/2020/518`)
- Vadim Lyubashevsky, Ngoc Khanh Nguyen, Maxime Plançon, and Gregor Seiler. Shorter Lattice-Based Group Signatures via "Almost Free" Encryption and Other Optimizations. In ASIACRYPT 2021.
  (Full version at `https://ia.cr/2021/1575`)
- Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Practical Lattice-Based Zero-Knowledge Proofs for Integer Relations. In ACM CCS 2020.
  (Full version at `https://ia.cr/2020/1183`)
- Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. SMILE: Set Membership from Ideal Lattices with Applications to Ring Signatures and Confidential Transactions. In CRYPTO 2021.
  (Full version at `https://ia.cr/2021/564`)

Other publications:

- David Bernhard, Ngoc Khanh Nguyen and Bogdan Warinschi: Adaptive Proofs Have Straightline Extractors (in the Random Oracle Model). In ACNS 2017.
  (Full version at `https://ia.cr/2015/712`)

- Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. A Non-PCP Approach to Succinct Quantum- Safe Zero-Knowledge. In: CRYPTO 2020.
  (Full version at `https://ia.cr/2020/737`)

- Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen and Gregor Seiler. More Efficient Amortization of Exact Zero-Knowledge Proofs for LWE. In ESORICS 2021.
  (Full version at `https://ia.cr/2020/1449`)

- Eduard Hauck, Eike Kiltz, Julian Loss and Ngoc Khanh Nguyen. Lattice-Based Blind Signatures, Revisited. In CRYPTO 2020.
  (Full version at `https://ia.cr/2020/769`)

- Dennis Hofheinz and Ngoc Khanh Nguyen. On Tightly Secure Primitives in the Multi-instance Setting. In PKC 2019.
  (Full version at `https://ia.cr/2018/958`)

- Vadim Lyubashevsky, Ngoc Khanh Nguyen and Maxime Plançon. Efficient Lattice-Based Blind Signatures via Gaussian One-Time Signatures. In PKC 2022.
  (Full version at `https://ia.cr/2022/006`)

- Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Shorter Lattice-Based Zero-Knowledge Proofs via One-Time Commitments. In PKC 2021.
  (Full version at `https://ia.cr/2020/1448`)

- Ngoc Khanh Nguyen. On the Non-existence of Short Vectors in Random Module Lattices. In ASIACRYPT 2019.
  (Full version at `https://ia.cr/2019/973`)

- Ngoc Khanh Nguyen and Gregor Seiler. Practical Sublinear Proofs for R1CS from Lattices. In CRYPTO 2022.
  (Full version at `https://ia.cr/2022/1048`)

- Ngoc Khanh Nguyen, Eftychios Theodorakis and Bogdan Warinschi. Lifting Standard Model Reductions to Common Setup Assumptions. In PKC 2022.
  (Full version at `https://ia.cr/2021/888`)

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my main supervisor, Vadim Lyubashevsky, who helped me step into the world of lattice-based cryptography. I am really grateful for his guidance and for the fruitful discussions which led to multiple publications.

Further, I thank my co-supervisor Dennis Hofheinz, who was willing to take me as an external PhD student at ETH Zurich. He was always happy to help and provide feedback whenever needed.

I would like to thank the examination committee: Dennis Hofheinz, Vadim Lyubashevsky and Ron Steinfeld for taking the time to review this thesis and participate in the doctoral exam.

I am very grateful for collaborations with the wonderful people during my PhD journey: Jonathan Bootle, Muhammed F. Esgin, Eduard Hauck, Dennis Hofheinz, Eike Kiltz, Julian Loss, Vadim Lyubashevsky, Maxime Plançon, Gregor Seiler, Eftychios Theodorakis and Bogdan Warinschi. I gained a lot of experience while working together, and I hope we get to collaborate again in the future.

Additionally, I would like to thank my friends and colleagues I met at IBM Research Europe - Zurich: Anja, Anna, Bertram, Cecilia, Gregor, Jesus, Jonathan, Julia, Luca, Maxime, Mike, Muhammed, Patrick Hough, Patrick Steuer, Patrick Towa, Romain, Sebastian, Simon, Vadim and Ward. My stay in Zurich would not haven been as enjoyable without the group hikes, coffee breaks and pool games. Moreover, I thank Sebastian Faller for his feedback on the German version of the abstract.

Last but not least, I owe a debt of gratitude to my parents and friends who helped me throughout my PhD journey. This thesis would not have existed without their tremendous support.

CONTENTS

NOTATION

FREQUENTLY USED SYMBOLS

$\mathbb{N}$	set of natural numbers $\{1, 2, 3, \ldots\}$
$\mathbb{Z}_n$	ring of integers modulo n
$[n]$	$\{1, 2, \ldots, n\}$
κ	security parameter
q	proof system modulus
d	power-of-two, ring dimension
$\mathcal{R}$	ring of integers $\mathbb{Z}[X]/(X^d + 1)$
$\mathcal{R}_q$	ring $\mathbb{Z}_q[X]/(X^d + 1)$
l	number of irreducible factors of $X^d + 1$ modulo q
$\tilde{f}$	constant coefficient of a polynomial $f \in \mathcal{R}$
σ_i	automorphism over $\mathcal{R}$ defined by the map $X \mapsto X^i$ for $i \in \mathbb{Z}_{2d}^\times$
$\mathsf{Aut}(\mathcal{R})$	the automorphism group $\{\sigma_i : i \in \mathbb{Z}_{2d}^\times\}$
$\mathcal{C}$	challenge space over $\mathcal{R}_q$
ω	maximum coefficient in the absolute value of a challenge in $\mathcal{C}$
$D_{\mathfrak{s}}$	discrete Gaussian distribution with standard deviation $\mathfrak{s}$
m_1	length of a "small" committed message $\mathbf{s}_1 \in \mathcal{R}_q^{m_1}$
m_2	length of the randomness for a commitment scheme over $\mathcal{R}_q$
ℓ	length of a (not necessarily small) committed vector $\mathbf{m} \in \mathcal{R}_q^{\ell}$
λ	parameter used for soundness amplification
κ_{MSIS}	dimension of a Module-SIS problem
κ_{MLWE}	dimension of a Module-LWE problem

1

INTRODUCTION

Zero-knowledge proofs form the foundations of many complex privacy-oriented protocols, such as electronic voting, verifiable computation and blockchain. In such applications, it is essential to be able to prove in zero-knowledge, i.e. the proof does not leak any secret information, that one knows how to open a cryptographic commitment, and to prove that the committed values have particular properties or satisfy certain relations.

Recently, more and more zero-knowledge proof techniques have been introduced, each with improvements in proof size, proving time, or verification time. These new constructions are based on a variety of cryptographic assumptions, including the discrete logarithm assumption, pairing-based assumptions, collision-resistant hash functions, and lattice-based assumptions such as (Module-)SIS and (Module-)LWE. However, only constructions from hash-functions or lattices stand any chance of being secure against quantum adversaries. The currently most efficient, in terms of size and speed, quantum-safe basic primitives (e.g. encryption and signature schemes) rely on the hardness of lattice problems with algebraic structure. This is highly evidenced by the fact that majority of the NIST Post-Quantum Competition [NIS] finalists are based on lattices. Lattice-based constructions are therefore natural candidates for more advanced cryptographic tools like zero-knowledge proofs [1].

Lattice-based cryptography relies upon the following fundamental hardness assumption, i.e. it is computationally difficult to find a low-norm vector $\mathbf{s}$ which satisfies

$$\mathbf{As} = \mathbf{t} \bmod q. \tag{1.1}$$

Hence, a natural approach for building privacy-preserving protocols based on the hardness of lattice problems would be to require proving knowledge of a secret vector $\mathbf{s}$ which satisfies the above, or a related, equality. Unlike in the discrete logarithm world, where proving knowledge of a secret s satisfying $g^s = t$ turns out to have a very simple and efficient solution [Sch89], the

1 Technically speaking, the protocols described in the thesis are called *arguments* since their soundness property relies on a computational assumption. However, for simplicity, we use the terms *proof* and *argument* interchangeably.

additional requirement of showing that $\|\mathbf{s}\|$ is small appears to be a major complication for practical lattice cryptography.

Currently, the most efficient lattice-based identification scheme over polynomial rings[2] was proposed by Lyubashevsky [Lyu09; Lyu12], who presented a zero-knowledge proof of knowledge of a vector $\bar{\mathbf{s}}$ and a polynomial c with small coefficients satisfying

$$\mathbf{A}\bar{\mathbf{s}} = c\mathbf{t} \bmod q, \tag{1.2}$$

where $\|\bar{\mathbf{s}}\|$ is some factor (depending on the dimension of $\mathbf{s}$) larger than $\|\mathbf{s}\|$. The protocol enjoys small proof sizes since it achieves negligible soundness error in one-shot, i.e. no repetition is required.

While such *relaxed* proof systems are good enough for constructing efficient basic protocols, such as signature schemes [BG14; Duc+18], the fact that the norm of the extracted $\bar{\mathbf{s}}$ is much larger than the norm of $\mathbf{s}$, along with the presence of the extra factor c in front of $\mathbf{t}$, makes these proofs tricky to use in many other situations. This often results in not giving the resulting scheme the desired functionality, or the protocols employing these proofs being simply less efficient than necessary. Indeed, such constructions are then required to select much larger parameters than needed in order to accommodate the presence of the multiplicand c and the "slack" between the length of the known solution $\mathbf{s}$ and the solution $\bar{\mathbf{s}}$ that one can prove.

Moreover, there are applications where relaxed proof systems are not satisfactory, such as proving integer relations and range proofs. In these protocols one wants to commit to integers, prove that they lie in certain intervals, and prove additive and multiplicative relations between them. In particular, one usually commits to the integers in their binary (or some other small-base) representation, and then proves that the committed message really is a binary vector [Esg+19c; Lib+18]. Hence, it is essential to prove that it does not have any coefficients which come from a larger set.

First lattice-based protocols for exactly proving (1.1) used the combinatorial algorithm of Stern [Ste93] to prove that the L_∞ norm of $\mathbf{s}$ is bounded by revealing a random permutation of $\mathbf{s}$. The main problem with these protocols was that their soundness error was $2/3$, and so they had to be repeated around 200 times to achieve an acceptably small (i.e. 2^{-128}) soundness error. This resulted in proofs for even simple statements being more than 1MB in size [Lin+13], while more interesting constructions produced outputs of size tens of Megabytes [Lib+18; Lib+16; Lib+17]. A significant improvement was shown in [Beu20] by generically combining Stern's protocol with a

2 Namely, rings of the form $\mathcal{R} := \mathbb{Z}[X]/(f(X))$, where $f(X)$ is a monic, irreducible polynomial.

"cut-and-choose" technique to decrease the soundness error of each protocol run (at the cost of higher running times). This allowed proofs for basic statements to be around 200KB in size.

Later on, a more algebraic approach for proving (1.1) combined lattice-based commitments and zero-knowledge proofs of committed values to prove linear relations between the coefficients of $\mathbf{s}$ and also prove a bound on its L_∞ norm. The first such protocols [BLS19; Esg+19a; Yan+19] had proof sizes that were in the order of several hundred kilobytes. These schemes were later significantly improved in [ALS20; ENS20; LNS21a], where it was shown how to very efficiently prove polynomial products over a ring and also linear relations over the CRT slots [3] of committed values. Optimisations of these techniques decreased the proof size for basic statements to around $30 - 50$KB.

The high level idea to prove the L_∞ norm is as follows. For simplicity, suppose we want to prove that $\mathbf{s}$ has coefficients in the set $\{-1, 0, 1\}$. Then, we create a commitment to a polynomial vector $\mathbf{m} = (m_1, \ldots, m_\ell)$ whose CRT slots are the coefficients of $\mathbf{s}$, prove this (linear) relationship and also prove that

$$(m_i - 1) \cdot m_i \cdot (m_i + 1) = 0 \quad \text{for } i = 1, 2, \ldots, \ell. \tag{1.3}$$

By the homomorphic property of the CRT slots, Equation 1.3 is indeed equivalent to the CRT slots of $\mathbf{m}$ being in $\{-1, 0, 1\}$. Note that if $\mathbf{s} \in \mathcal{R}_q^m$ then the vector $\mathbf{m}$ consists of $\ell = m \cdot d/l$ polynomials where $\mathcal{R}_q := \mathcal{R}/(q)$ and l is the number of factors of $f(X)$ modulo q.

There are a few limitations of the aforementioned approach. Firstly, since the CRT slots of $\mathbf{m}$ are small, this implies that the actual coefficients of $\mathbf{m}$ can be large, and thus committing to it requires using a more expensive commitment scheme, e.g. the BDLOP commitment [Bau+18b], which is much more expensive than the standard Ajtai commitment [Ajt96] for long $\mathbf{s}$. There is also an incompatibility between the requirement that the underlying ring has a lot of CRT slots and negligible soundness error of the protocol. Namely, if l is small, then we have to commit to more polynomials because $\mathbf{m}$ gets longer. On the other hand, if we choose l to be large (e.g. $l = d$) then a part of the protocol needs to be repeated for soundness amplification. Another downside is that proving $\|\mathbf{s}\|_\infty \leqslant \alpha$ in general requires committing to 2α extra polynomials. Hence, for vectors

3 We recall that for a polynomial $s \in \mathcal{R}_q := \mathbb{Z}_q[X]/(f(X))$, the Chinese Remainder Theorem (CRT) slots [Esg+19c] of s are coefficients of the vector $(s \bmod (q, f_1(X)), \ldots, s \bmod (q, f_l(X)))$ where $f(X)$ factors into irreducible polynomials $f_1(X), \ldots, f_l(X)$ modulo q. Note that if $f(X)$ splits into linear terms modulo q then CRT slots simply become integers in $\mathbb{Z}_q$.

$\mathbf{s}$ with somewhat-large coefficients, such as ones that are obtained from trapdoor sampling (e.g. [ABB10a; DLP14; MP12]), proving the L_∞ norm becomes significantly costlier. Finally, proving the L_2 norm, rather than the L_∞ one, is very often what one would like to do when constructing proofs for lattice-based primitives. For instance, if one is interested in bounding the norm of a linear combination of $\mathbf{s}$, e.g. for proving no decryption error occurred, then having the L_2 norm seems more optimal. Indeed, given $\|\mathbf{s}\| \leqslant \beta$ over a widely-used ring $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^d + 1)$ where d is a power-of-two, we can bound

$$\|\mathbf{a}^T\mathbf{s}\|_\infty \leqslant \|\mathbf{a}\| \cdot \|\mathbf{s}\| = \|\mathbf{a}\| \cdot \beta$$

whereas given the L_∞ norm bound $\|\mathbf{s}\|_\infty \leqslant \alpha$ we would only deduce that

$$\|\mathbf{a}^T\mathbf{s}\|_\infty \leqslant \|\mathbf{a}\|_1 \cdot \|\mathbf{s}\|_\infty = \|\mathbf{a}\|_1 \cdot \alpha$$

which is usually looser than the former inequality since in practice very few coefficients of $\mathbf{s}$ will be close to α. Another application is proving knowledge of vectors produced by trapdoor sampling because they have a (tightly) bounded L_2 norm but not L_∞ norm.

Outside lattice-based cryptography, there has been a significant advancement in the construction of practical zero-knowledge proof systems, and it has progressed to the point where they can be used routinely to prove relatively large arbitrary arithmetic circuits, thus in particular (1.1). When restricting to (plausibly) quantum-safe protocols, the PCP-type systems like Ligero++ [Bha+20] or Aurora [Ben+19] achieve proof sizes that scale poly-logarithmically with the witness size and only rely on collision-resistant hash functions. As a drawback, they have a concrete base cost in the order of $50 - 100$ Kilobytes. Hence, using lattice-based zero-knowledge proofs for statements of the form (1.1) still seems more advantageous in terms of proof size.

1.1 OUR CONTRIBUTIONS

In this work we propose a simple and general framework, called Lantern[4], for proving statements related to lattice-based cryptography, such as (1.1). Our new protocols do not rely on the CRT slots technique which results in the following two immediate improvements over the current state-of-the-art [ALS20; ENS20; LNS21a]. First, since we do not need to commit to long

4 The name stands for: **la**ttice-based **n**on-in**ter**active zero-k**n**owledge proofs.

Stern-type proofs	3522 KB
Bootle et al. [BLS19]	384 KB
Beullens [Beu20]	233 KB
Ligero [Ame+17]	157 KB
Aurora [Ben+19; Bos+20]	72 KB
Esgin et al. [ENS20]	47 KB
Lyubashevsky et al. [LNS21a]	33 KB
Lantern	13 KB

FIGURE 1.1: Proof length comparison for proving knowledge of short $\mathbf{s}, \mathbf{e}$ satisfying $\mathbf{As} + \mathbf{e} = \mathbf{t} \bmod q$, where $\mathbf{A} \in \mathcal{R}_q^{n \times m}$, $(n, m, d, q) = (16, 16, 64, \approx 2^{32})$, and $\|(\mathbf{s}, \mathbf{e})\| \leqslant \sqrt{2048}$. The protocols from prior works need to make the additional restriction that all the coefficients in $\mathbf{s}, \mathbf{e}$ are from $\{-1, 0, 1\}$. The sizes for the Stern-type proof are taken from [BLS19]. The sizes for Ligero and the scheme from [Beu20] are originally from [Beu20] and are for the matrix $\mathbf{A}$ of height 8.

vectors with large coefficients anymore (e.g. $\mathbf{m}$ in the previous example), we can actually use the Ajtai commitment [Ajt96] which is much cheaper. Secondly, we circumvent the issue of repeating certain (rather expensive) parts of the protocol for boosting soundness. Consequently, our proof sizes become around $2 - 3X$ smaller than prior works for basic statements (see Figure 1.1). In particular, for statements of the form (1.1), the total proof size is $\approx$ 13KB where approximately 8KB of that consists of just the "minimum" commitment (i.e. a commitment to just one element in $\mathcal{R}_q$) and its opening proof. This implies that our construction is quite close to being optimal for any approach that requires creating a commitment to $\mathbf{s}$ using known lattice-based commitment schemes. Since most of the practical lattice-based zero-knowledge proofs for proving knowledge of a witness $\mathbf{s}$ satisfying certain relations follow the commit-and-prove approach and first commit to $\mathbf{s}$, it appears that any significant improvement to our framework (e.g. another factor of two) would require significant improvements in theory of lattices, basing security on stronger assumptions, or simply a different approach.

Our framework is defined in such a way that it can be used out-of-the-box to construct more advanced privacy-preserving primitives. We demonstrate

	Ciphertext Size	Proof Size	Decryption Time Independent of Forgery Time
[LN17]	9KB	9KB	×
[LNS21a]	4KB	33 - 44KB	×
Lantern	1KB	17KB	✓

FIGURE 1.2: The table compares our instantiation of a verifiable encryption scheme from this thesis with [LN17] and [LNS21a]. The latter paper presents a verifiable *decryption* scheme, but the proof size for a verifiable encryption scheme constructed in the same manner would be similar.

the applicability of our protocols with various real-world examples. First, we build an efficient lattice-based verifiable encryption which is on-par with the current state-of-the-art [LN17] in terms of the ciphertext + proof size but overcomes the undesirable problem with the expected decryption time being dependent on the adversary's running time (see Figure 1.2). Furthermore, we show how our framework can be applied to obtain ABB-like group signatures[5] [ABB10b; Lyu+21; PLS18] with signature size $2X$ smaller than the currently most efficient construction [Lyu+21] (see Figure 1.3). Last but not least, we propose a new logarithmic-size lattice-based one-out-of-many proof [GK15] which, using standard techniques, can be transformed into efficient ring and group signatures. Our construction produces signature sizes $\approx 35\%$ smaller while having more than one order of magnitude smaller public keys than the current state-of-the-art lattice-based ring signatures [ESZ21; LNS21b] for a large number of users (see Figure 1.4). We highlight that the one-out-of-many proof, combined with our new proofs of integer relations (see Figure 1.5), can be used to build an efficient lattice-based confidential payment system as in [ESZ21; Esg+19c; LNS21b].

We now give a technical overview of the main building blocks for constructing our framework. For the sake of concreteness, let us define the

5 The main advantage of ABB-like group signatures is the constant signing and verification time as well as constant signature size, i.e. they do not depend on the size of the group.

	Public Key Size	Signature Size	Opening Time Independent of Adversary's Forgery Time
[PLS18]	123KB	581KB	×
[Lyu+21]	96KB	203KB	×
Lantern	48KB	88KB	✓

FIGURE 1.3: Comparison of our ABB-like group signature with prior constructions [Lyu+21; PLS18].

ring $\mathcal{R}_q := \mathbb{Z}_q[X]/(X^d + 1)$ where d is a power-of-two which is a standard choice of a ring in practical lattice-based constructions[6].

1.1.1 *Lattice-Based Hybrid Commitment Scheme*

Our starting point is a new lattice-based commitment scheme, called ABDLOP, which generalises the constructions of Ajtai [Ajt96] and BDLOP [Bau+18b]. Concretely, to commit to a message vector $\mathbf{s}_1 \in \mathcal{R}_q^{m_1}$ with small coefficients, as well as a "full-fledged" polynomial vector $\mathbf{m} \in \mathcal{R}_q^{\ell}$, we sample a randomness vector $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$ and compute:

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

We observe that when $\ell = 0$ (resp. $m_1 = 0$) then this construction ends up being the Ajtai (resp. BDLOP) commitment scheme. In particular, the commitment size does not depend on the length m_1 of $\mathbf{s}_1$ (but it does on ℓ). Hence, our strategy is to commit to long vectors with small coefficients in the "Ajtai" part $\mathbf{s}_1$, e.g. vector $\mathbf{s}$ in Equation 1.1, and commit to a few *garbage* polynomials used for the proofs in the "BDLOP" part $\mathbf{m}$. The opening of the commitment is a pair $(\mathbf{s}_1, \mathbf{s}_2)$[7].

Using similar techniques as in [ALS20] one can show ABDLOP scheme is binding with respect to *weak openings*, i.e. triples $(\mathbf{s}_1, \mathbf{s}_2, c)$ which satisfy:

6 Our protocols are the most efficient here because they utilise a specific automorphism in this ring. However, the high-level ideas can also be made to work for rings which do not have this algebraic structure. We refer to [LNP22b, Section 7] for more details.

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

	sig. sizes for rings of size			hardness	public
	2^6	2^{12}	2^{21}	assumption	key size
Raptor [LAZ19]	81	5161	–	NTRU	0.9
DualRing-LB [Yue+21]	6	106	–	MSIS, MLWE	[2.8, 3.4]
Falafl [BKP20]	32	35	39	MSIS, MLWE	1.9
MatRiCT [Esg+19c]	31	59	148	MSIS, MLWE	[3.4, 22.7]
MatRiCT+ [ESZ21]	11	18	40(?)	MSIS, MLWE	–
SMILE [LNS21b]	18	19	22	MSIS, MLWE	2
Calamari [BKP20]	8	14	23	CSIDH-512	0.06
Lantern	14	15	16	MSIS, MLWE	0.13

FIGURE 1.4: Comparison of the different post-quantum ring signature schemes with approximately 128 bits of security. All the values are given in KB. The signatures sizes for [ESZ21; LNS21b] only approximately correspond to the ring sizes (e.g. 18KB signature size is for the ring of 2^{10} users and not 2^{12}). For DualRing-LB and MatRiCT(+) the user public key size grows in the number of users. Further, we extrapolate the signature size for MatRiCT+ with 2^{21} users from the smaller examples and from MatRiCT.

- $\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A$,
- $c \in \mathcal{R}_q$ is an invertible polynomial with small coefficients,
- $\|c\mathbf{s}_1\|$ and $\|c\mathbf{s}_2\|$ are small.

under the hardness of the Module-SIS assumption. On the other hand, the hiding property of the ABDLOP scheme comes from the fact that if vector $\mathbf{s}_2$ is long enough, then $\begin{bmatrix}\mathbf{A}_2 \\ \mathbf{B}\end{bmatrix}\mathbf{s}_2$ is indistinguishable from a random vector under the Module-LWE assumption.

Proof of knowledge of the ABDLOP commitment opening can be constructed using the standard Schnorr-like sigma protocol [Lyu12] adapted to the lattice setting (see Figure 1.6). Suppose that $\mathcal{C}$ is a challenge space consisting of polynomials with small coefficients such that any difference of two distinct challenges is invertible in $\mathcal{R}_q$. Hence, if one manages to extract two valid transcripts with two different challenges $c, c' \in \mathcal{C}$ then one immediately obtains a weak opening of $(\mathbf{t}_A, \mathbf{t}_B)$.

N	128	512
[LNS20]	25KB	45KB
Lantern	12KB	15KB

N	128	512
[LNS20]	40KB	100KB
Lantern	15KB	21KB

FIGURE 1.5: Proof size comparison for proving integer addition (on the left) and multiplication (on the right). Here, N is the bit-length of the integers. It is worth mentioning that [ESZ21; Esg+19c] also construct efficient proofs of integer addition, alternatively called *balance proofs*, which use similar CRT-packing techniques as [LNS20].

1.1.2 *Product Proofs with Automorphisms*

One of our main building blocks is a proof of linear and higher-degree equations in the committed messages $\mathbf{s}_1, \mathbf{m}$. Namely, we adapt the product proof from [ALS20] to prove that $f(\mathbf{s}_1, \mathbf{m}) = 0$ where $f : \mathcal{R}_q^{m_1+\ell} \to \mathcal{R}_q$ is a polynomial function. For presentation purposes, let us describe how to prove that

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

Generalisation to arbitrary quadratic and higher-degree relations follows immediately.

First of all, consider the *masked opening* $\mathbf{z}_1 := \mathbf{y}_1 + c\mathbf{s}_1$ of $\mathbf{s}_1$ defined in Figure 1.6. Note that

$$\mathbf{z}_1^T \mathbf{z}_1 = c^2 \mathbf{s}_1^T \mathbf{s}_1 + 2c\mathbf{y}_1^T \mathbf{s}_1 + \mathbf{y}_1^T \mathbf{y}_1$$

and hence the coefficient corresponding to the quadratic term c^2 is what we are interested in. We cannot do the same argument with $\mathbf{m}$ since no masked opening of $\mathbf{m}$ was sent. However, we observe that the verifier can compute

$$\mathbf{t}_B - \mathbf{B}\mathbf{z}_2 = -\mathbf{B}\mathbf{y}_2 + c\mathbf{m}$$

which is of the similar form as the masked opening of $\mathbf{s}_1$. Then

$$(\mathbf{t}_B - \mathbf{B}\mathbf{z}_2)^T (\mathbf{t}_B - \mathbf{B}\mathbf{z}_2) = c^2 \mathbf{m}^T \mathbf{m} - 2c\mathbf{y}_2^T \mathbf{B}^T \mathbf{m} + \mathbf{y}_2^T \mathbf{B}^T \mathbf{B}\mathbf{y}_2.$$

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

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

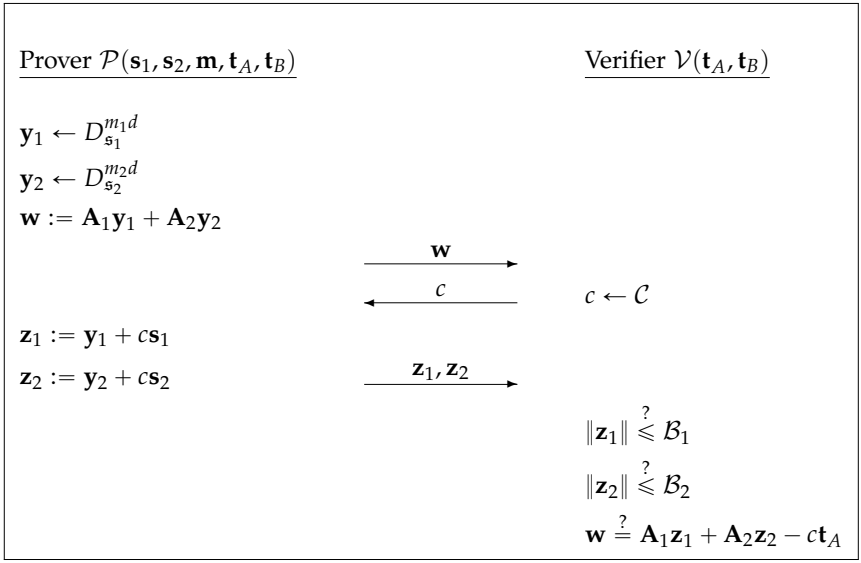

FIGURE 1.6: Proof of knowledge of the ABDLOP commitment opening. Vectors $\mathbf{y}_i$ are sampled from a discrete Gaussian with standard deviation $\mathfrak{s}_i$. We neglect the rejection sampling step for the sake of the overview.

where

$$g_1 := 2(\mathbf{y}_1^T\mathbf{s}_1 - \mathbf{y}_2^T\mathbf{B}^T\mathbf{m}), \quad g_0 := \mathbf{y}_1^T\mathbf{y}_1 + \mathbf{y}_2^T\mathbf{B}^T\mathbf{B}\mathbf{y}_2.$$

The idea is then to additionally send commitments t_i to g_i for $i = 1, 2$ (we will put g_1, g_0 in the "BDLOP" part since their coefficients are large) and prove that

$$\mathbf{z}_1^T\mathbf{z}_1 + (\mathbf{t}_B - \mathbf{B}\mathbf{z}_2)^T(\mathbf{t}_B - \mathbf{B}\mathbf{z}_2) - ct_1 - t_0$$

is a commitment to zero. Finally, in order to reduce the number of garbage commitments, we apply the technique by Attema et al. [ALS20] which does not need to commit to g_0. Consequently, proving a quadratic equation costs an extra commitment to a $\mathcal{R}_q$ element.

In our framework, we need to prove quadratic relations which additionally involve automorphisms σ of $\mathcal{R}$[8]. For instance, we will be interested in equations such as

$$\sigma(\mathbf{s}_1)^T\mathbf{s}_1 + \sigma(\mathbf{m})^T\mathbf{m} = 0 \tag{1.4}$$

8 We denote the group $\mathsf{Aut}(\mathcal{R})$ of automorphisms of $\mathcal{R}$ as $\mathsf{Aut}(\mathcal{R}) := \{\sigma_i : i \in \mathbb{Z}_{2d}^\times\}$ where $\sigma_i : \mathcal{R} \to \mathcal{R}$ is defined by $\sigma_i(X) = X^i$.

where for $\mathbf{x} := (x_1, \ldots, x_n) \in \mathcal{R}_q^n$, we define $\sigma(\mathbf{x}) := (\sigma(x_1), \ldots, \sigma(x_n))$. If we were to apply the approach as above, we would obtain

$$\sigma(\mathbf{z})_1^T \mathbf{z}_1 + \sigma(\mathbf{t}_B - \mathbf{B}\mathbf{z}_2)^T(\mathbf{t}_B - \mathbf{B}\mathbf{z}_2) = cg_{1,0} + \sigma(c)g_{1,1} + g_0 \tag{1.5}$$

where

$$g_{1,0} := \sigma(\mathbf{y})_1^T \mathbf{s}_1 - \sigma(\mathbf{B}\mathbf{y}_2)^T \mathbf{m}, \quad g_{1,1} := \mathbf{y}_1^T \sigma(\mathbf{s}) - (\mathbf{B}\mathbf{y}_2)^T \sigma(\mathbf{m})$$

and g_0 is defined as

$$g_0 := \sigma(\mathbf{y}_1)^T \mathbf{y}_1 + \sigma(\mathbf{B}\mathbf{y}_2)^T \mathbf{B}\mathbf{y}_2.$$

This means that now we would need to commit to both $g_{1,0}$ and $g_{1,1}$. A simple optimisation is to choose a challenge space $\mathcal{C}$ such that $c \in \mathcal{C}$ is stable under automorphism σ, i.e. $\sigma(c) = c$. Then, the expression in (1.5) would be equal to $cg_1 + g_0$ where $g_1 := g_{1,0} + g_{1,1}$. This way, we only commit to one extra polynomial g_1 as in the case previous case. The limitation is, however, the additional condition on the challenge space $\mathcal{C}$ being stable under σ. We show that for typical choices of σ used in this thesis, such a challenge space of an exponential size in the security parameter can still be constructed [9].

1.1.3 *Inner Products of the Polynomial Coefficients*

We propose new techniques to prove inner products between polynomial coefficients of the committed vectors. For instance, suppose we want to prove for some public $\mathbf{a}_1 \in \mathcal{R}_q^{m_1}$ and $\mathbf{a}_2 \in \mathcal{R}_q^{\ell}$ that

$$\langle \mathbf{a}_1, \mathbf{s}_1 \rangle + \langle \mathbf{a}_2, \mathbf{m} \rangle = 0 \pmod{q}$$

where we denote $\langle \mathbf{x}, \mathbf{y} \rangle$ to be the $\mathbb{Z}$-inner product of their corresponding coefficient vectors $\vec{x}$ and $\vec{y}$. The crucial observation here is that $\langle \mathbf{a}_1, \mathbf{s}_1 \rangle + \langle \mathbf{a}_2, \mathbf{m} \rangle \in \mathbb{Z}_q$ is the constant coefficient of the following polynomial in $\mathcal{R}_q$:

$$\sigma_{-1}(\mathbf{a}_1)^T \mathbf{s}_1 + \sigma_{-1}(\mathbf{a}_2)^T \mathbf{m}. \tag{1.6}$$

In other words, we note that for any two polynomials $u, v \in \mathcal{R}_q$, the inner product $\langle u, v \rangle \in \mathbb{Z}_q$ is the constant coefficient of the polynomial $\sigma_{-1}(u)v \in \mathcal{R}_q$ where automorphism σ_{-1} maps $X \mapsto X^{-1}$. Indeed, if we write $u := \sum_{i=0}^{d-1} u_i X^i$ and $v := \sum_{i=0}^{d-1} v_i X^i$, then the constant coefficient of

$$\sigma_{-1}(u)v = \left(\sum_{i=0}^{d-1} u_i X^{-i} \right) \left(\sum_{i=0}^{d-1} v_i X^i \right)$$

9 Recall we still want the difference of any two distinct challenges in $\mathcal{C}$ to be invertible over $\mathcal{R}_q$.

is $u_0v_0 + u_1v_1 + \ldots + u_{d-1}v_{d-1} = \langle u, v \rangle$.

Now, to show that the constant coefficient of (1.6) vanishes, one could simply send that polynomial in the clear and the verifier would check that the constant coefficient is indeed zero. However, that would reveal all the other coefficients of (1.6) – making the scheme not zero-knowledge. Instead, we apply the following strategy described first by Esgin et al. [ENS20]. Namely, we commit to a random polynomial $g \leftarrow \{x \in \mathcal{R}_q : \tilde{x} = 0\}$, where $\tilde{x}$ means the constant coefficient of x. Then, given a challenge $v \leftarrow \mathbb{Z}_q$ from the verifier, we send

$$h := g + v \cdot \left(\sigma_{-1}(\mathbf{a}_1)^T \mathbf{s}_1 + \sigma_{-1}(\mathbf{a}_2)^T \mathbf{m} \right).$$

Then, the verifier checks whether the constant coefficient of h is zero. Since we masked all the other coefficients of (1.6) using g, the verifier gets no sensitive information. Finally, we need to prove that h was well-formed. However, this is just a linear equation in the committed messages $\mathbf{s}_1, \mathbf{m}$ and g and can thus be proved as in Section 1.1.2.

One observes that the soundness error of this approach is $1/q_1$ where q_1 is the smallest prime which divides q. Indeed, a cheating prover might have $\mathbf{s}_1, \mathbf{m}$ such that the constant coefficient of (1.6) is q/q_1 and hope that the challenge v is divisible by q_1. Then, by construction, the constant coefficient of h would still be zero.

In order to exponentially decrease the soundness error, we repeat this part of the protocol, i.e. we commit to extra λ polynomials $g_1, \ldots, g_\lambda \leftarrow \{x \in \mathcal{R}_q : \tilde{x} = 0\}$ and send corresponding λ polynomials $h_1, \ldots, h_\lambda$. Consequently, we reduce the soundness error to $q_1^{-\lambda}$ at the cost of committing to λ garbage polynomials. In the thesis, we also propose an optimisation which relies on certain properties of the σ_{-1} automorphism and reduces the number of garbage polynomials from λ to $\lambda/2$.

We highlight that this strategy can be easily generalised to prove multiple inner products at no extra cost. In particular, one can efficiently prove arbitrary $\mathbb{Z}_q$-linear equations, i.e. that coefficients of $\mathbf{s}_1$ and $\mathbf{m}$ satisfy

$$A_1\vec{s}_1 + A_2\vec{m} = \vec{u}$$

where $\vec{s}_1$ (resp. $\vec{m}$) is the coefficient vector of $\mathbf{s}_1$ (resp. $\mathbf{m}$) and the equation is over $\mathbb{Z}_q$.

The aforementioned approach can also be used to prove inner products between committed vectors. As an example, suppose we want to prove that

$$\langle \mathbf{s}_1, \mathbf{s}_1 \rangle + \langle \mathbf{m}, \mathbf{m} \rangle = 0 \pmod{q}.$$

It is equivalent to prove that the constant coefficient of

$$\sigma_{-1}(\mathbf{s}_1)^T\mathbf{s} + \sigma_{-1}(\mathbf{m})^T\mathbf{m}$$

vanishes. Then, we can proceed as before, i.e. commit to $g \leftarrow \{x \in \mathcal{R}_q : \tilde{x} = 0\}$ and given a challenge $v \leftarrow \mathbb{Z}_q$, we send

$$h := g + v \cdot \left(\sigma_{-1}(\mathbf{s}_1)^T\mathbf{s} + \sigma_{-1}(\mathbf{m})^T\mathbf{m}\right).$$

Now, proving that h is well-formed is simply a quadratic equation (with the σ_{-1} automorphism) in the committed messages $\mathbf{s}_1, \mathbf{m}$ and g and we covered exactly those in the previous subsection.

1.1.4 *Proving Euclidean and Infinity Norms*

The next crucial component of our framework is proving exactly that some of the committed messages satisfy certain norm bounds. Suppose we want to prove $\|\mathbf{s}_1\| \leqslant \beta$. We first recall the "approximate range proof" strategy [GHL21; LNS21a] which only proves the norm approximately. Concretely, we first commit to a small [10] masking vector $\vec{y}$ to ensure zero-knowledge, and then given a random matrix R with coefficients in $\{-1, 0, 1\}$, we output

$$\vec{z} := \vec{y} + R\vec{s}_1 \tag{1.7}$$

and prove that $\vec{z}$ is well-formed (this is just a $\mathbb{Z}_q$-linear relation which can be proven as described above). It can be shown that if $\|\vec{z}\|$ is small, then with an overwhelming probability $\|\mathbf{s}_1\|$ must also be small. This approach is only "approximate" since in the end the prover convinces the verifier that $\|\mathbf{s}_1\| \leqslant \psi \cdot B$ for some approximation factor $\psi > 1$. Even though this is not what we originally wanted to prove, it will be an important building block.

For presentation, suppose first we want to prove $\|\mathbf{s}_1\|^2 = B^2$. One observes that

$$B^2 = \|\mathbf{s}_1\|^2 = \langle \mathbf{s}_1, \mathbf{s}_1 \rangle$$

which boils down to proving an inner product between the secret coefficients as described in Section 1.1.3. The only caveat is that we only proved that $\|\mathbf{s}_1\|^2 = B^2 \pmod q$. This is where we apply the approximate range proof. Indeed, if $(1 + \psi^2) \cdot B^2 < q$ then we can deduce that

$$|B^2 - \|\mathbf{s}_1\|^2| \leqslant B^2 + \psi^2 \cdot B^2 < q$$

10 In our applications, vector $\vec{y}$ will be of dimension 256.

and thus no modulo overflow occured. Hence, $\|\mathbf{s}_1\|^2 = B^2$ holds over integers.

We go back to the old case $\|\mathbf{s}_1\|^2 \leqslant B^2$. Let $\vec{\vartheta} \in \mathbb{Z}_q^d$ be the binary decomposition of $B^2 - \|\mathbf{s}_1\|^2$. We can then commit to the polynomial $\vartheta \in \mathcal{R}_q$, where its coefficient vector is exactly $\vec{\vartheta}$, and prove that

$$\langle \mathbf{s}_1, \mathbf{s}_1 \rangle + \langle \mathsf{pow}(B^2), \theta \rangle = B^2 \pmod q$$

where $\mathsf{pow}(B^2) := \sum_{i=0}^{\lfloor \log B^2 \rfloor} 2^i \cdot X^i \in \mathcal{R}_q$. This is again an inner product equation which can be proven using the techniques from Section 1.1.3. Now, if we can prove that ϑ has binary coefficients, then we can deduce that

$$|B^2 - \|\mathbf{s}_1\|^2 - \langle \mathsf{pow}(B^2), \theta \rangle| \leqslant B^2 + \psi^2 \cdot B^2 + 2B^2 = (3 + \psi^2) \cdot B^2.$$

Hence, if $(3 + \psi^2) \cdot B^2 < q$ then we get that $\|\mathbf{s}_1\|^2 \leqslant B^2$. What we have left is to prove that ϑ has binary coefficients. We make use of the following observation: vector $\vec{b} = (b_1, \ldots, b_n) \in \mathbb{Z}^n$ has binary coefficients if and only if

$$\langle \vec{b}, \vec{b} - \vec{1} \rangle = \sum_{i=1}^{n} b_i(b_i - 1) = 0 \quad \text{over } \mathbb{Z}.$$

Our strategy is thus to prove that

$$\left\langle \vartheta, \vartheta - \sum_{i=0}^{d-1} X^i \right\rangle = 0 \bmod q \tag{1.8}$$

and apply an approximate range proof on ϑ to prove that $\|\vartheta\|$ is relatively small. Then, similarly as before, we deduce that (1.8) holds over integers and by our observation, coefficients of ϑ are indeed binary. It is easy to see that this strategy can be used to perform arbitrary L_∞ proofs, i.e. $\|\mathbf{s}\|_\infty \leqslant \alpha$, by first binary-decomposing the coefficients of $\mathbf{s}$ and proving that the resulting vector is binary.

1.1.5 *Shorter Proofs via Bimodal Gaussians and One-time Commitments*

In order to ensure zero-knowledge property of our schemes, we apply the rejection sampling technique [Lyu12]. The idea is to mask the secret vector, e.g. $c\mathbf{s}_i$ in Figure 1.6, by adding to it a freshly sampled $\mathbf{y}_i$ from a discrete Gaussian $D_{\mathfrak{s}_i}^{m_i d}$ with standard deviation $\mathfrak{s}_i$ and then aborting the protocol with certain probability p_i dependent on $\mathbf{z}_i := \mathbf{y}_i + c\mathbf{s}_i$. Note that

choosing small standard deviation $\mathfrak{s}_i$ results in $\mathbf{z}_i$ having small coefficients which reduces the proof size but also drastically increases the aborting probability p_i. On the other hand, large $\mathfrak{s}_i$ implies bigger coefficients of $\mathbf{z}_i$ which not only increase the proof size but also forces us to pick less optimal parameters for the commitment to satisfy the binding property.

If the protocol contains only one rejection sampling then the standard choice [Lyu12] is to pick $\mathfrak{s}_i = 11 \cdot \|c\mathbf{s}_i\|$. Then, the aborting probability p_i becomes $\approx 2/3$. However, in our framework, we will have *at least* three rejection sampling steps. The first two are for $\mathbf{z}_1, \mathbf{z}_2$ as in Figure 1.6. At least one more will be needed for the approximate range proof, i.e. (1.7). Hence, if we were to set the same parameters for all (three) rejection sampling steps as in [Lyu12] then the total probability of the prover not aborting would be $1/27$. Thus, for run-time purposes, it is important to somehow increase the non-abort probability with no big impact on the standard deviations.

We solve this issue by applying bimodal Gaussian rejection sampling, first introduced by [Duc+13]. The difference from the standard rejection sampling procedure is that we additionally sample a sign $b_i \leftarrow \{-1, 1\}$ and then output $\mathbf{z}_i := \mathbf{y}_i + b_i \cdot c\mathbf{s}_i$. Due to the symmetry of the distribution of $\mathbf{z}_i$, [Duc+13] manage to reduce the standard deviation by one order of magnitude (or for the same standard deviation, significantly reduce the aborting probability p_i). This technique would thus be beneficial for us since we need to deal with (at least) three rejection sampling steps. We explain below how to use bimodal Gaussian rejection sampling in our setting.

Let us first consider the approximate range proof. Now, instead of doing (1.7), we would commit to a sign $b \leftarrow \{-1, 1\}$, send $\vec{z}$ defined by

$$\vec{z} = \vec{y} + b \cdot R\vec{s}_1$$

and prove that $\vec{z}$ is well-formed. We show that, assuming that b is a sign, this equation can be proven directly using our techniques from Section 1.1.3. However, we still need to prove that b is indeed a sign. This can be then done by proving that $(b+1)(b-1) = 0$ over $\mathcal{R}_q$ (Section 1.1.2) and that for all $1 \leqslant i \leqslant d-1$, the constant coefficient of $X^{-i} \cdot b$ is zero (Section 1.1.3).

Further, we focus on the rejection sampling for the randomness of the commitment scheme. Concretely, we compute $\mathbf{z}_2$ as in Figure 1.6 (i.e. we do not sample any additional signs) but we still apply the bimodal Gaussian rejection sampling strategy and reject with certain probability defined in [Duc+13]. Surprisingly, this (naive) strategy works at the potential cost of leaking the value of $\langle \mathbf{z}_2, c\mathbf{s}_2 \rangle \in \mathbb{Z}$. Clearly, leaking some information about the randomness can be dangerous. For example, if one were to repeatedly perform proofs of knowledge for *the same* commitment which leaks

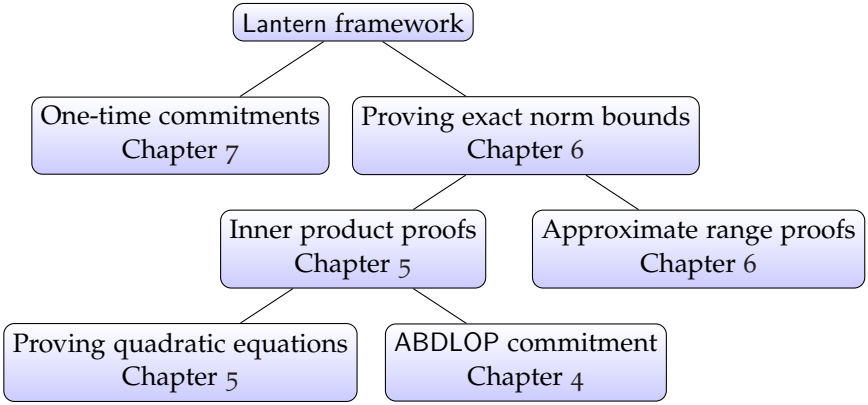

FIGURE 1.7: Main components of the Lantern framework.

something about the same randomness each time, eventually the entire randomness could be recovered by even a passive observer. However, if one looks closer at how the commitments are usually being used in many of the privacy-based protocols, one would notice that the scheme is used to commit to some intermediate value, give a proof-of-knowledge of the value (i.e. proof of knowledge of the commitment randomness), and then discards the commitment. Therefore, only one proof of knowledge is performed and randomness is freshly sampled every time a proof is produced. This is evidenced by the fact that our protocols (where commitment is a part of the proof) are zero-knowledge under the *Extended* Module-LWE problem [AA16] where the inner product of the secret with random vectors are revealed as hints.

1.2 THESIS ORGANISATION

The structure of this thesis is summarised in Figure 1.7. We first describe related works in the area of lattice-based zero-knowledge proofs and constructions of privacy-preserving primitives (e.g. ring and group signatures) in Chapter 2. Next, we cover relevant cryptographic as well as mathematical background in Chapter 3. Chapter 4 introduces the new ABDLOP commitment which is a generalisation of the Ajtai and BDLOP schemes and proposes a zero-knowledge opening proof. The focus on Chapter 5 can be split into two parts. First, we show how to prove linear and higher-degree equations in the committed messages (also the ones involving automor-

phisms). Secondly, we show how to use the tools to prove linear relations and inner products between the coefficients of the committed polynomials. Chapter 6 describes how to prove both Euclidean and infinity norm bounds and proposes a general framework, called Lantern, for proving arbitrary (lattice-related) statements. Further, Chapter 7 presents new techniques to further reduce the proof size by utilising one-time commitments. Next, we demonstrate the importance of our results with real-world applications to privacy-preserving primitives in Chapter 8. The thesis is concluded with Chapter 9 with some discussions and potential future research directions.

2

RELATED WORKS

In this chapter we provide a literature review in the area of lattice-based commitments, zero-knowledge proofs and current state-of-the-art privacy-preserving constructions. For the sake of presentation, we define $\mathcal{R} := \mathbb{Z}[X]/(X^d + 1)$, where d is a power-of-two and $\mathcal{R}_q := \mathcal{R}/(q)$ for a prime q. Also, we denote κ_{MSIS} and κ_{MLWE} to be the module ranks required for Module-SIS and Module-LWE security over the ring $\mathcal{R}_q$ respectively.

2.1 LATTICE-BASED COMMITMENT SCHEMES

Commitment schemes are a powerful tool used in various cryptographic constructions. This primitive allows one to commit to a chosen value with the possibility to reveal it later. There are two main security properties of commitment schemes. The first is called *hiding* meaning that the commitment itself does not reveal any information about the committed value. Second is *binding* which says that a party cannot change the value after they committed to it.

Current state-of-the-art lattice-based commitment schemes can be divided into two types [1]: Hashed-Message Commitments (HMC) [Ajt96] and Unbounded-Message Commitments (UMC) [Bau+18b]. The former one has the property that the sizes of commitments are almost independent of the sizes of the committed values. This comes at the cost of the smaller message space being only polynomials of small norm. On the other hand, the main characteristic of UMC is the unbounded message space, but the commitment size is linear in the size of the message.

HASHED-MESSAGE COMMITMENT. We describe the standard Module-SIS commitment scheme [Ajt96; Bau+18a; KTX08] which was first introduced implicitly in the seminal work by Ajtai [Ajt96]. Concretely, let

$$\mathbf{A}_1 \leftarrow \mathcal{R}_q^{\kappa_{\mathsf{MSIS}} \times m_1}, \quad \mathbf{A}_2 \leftarrow \mathcal{R}_q^{\kappa_{\mathsf{MSIS}} \times (\kappa_{\mathsf{MSIS}} + \kappa_{\mathsf{MLWE}})}$$

be uniformly random matrices as public parameters, where m_1 is the number of elements that one wishes to commit to. A commitment to a

1 We use the terminology from [ESZ21].

vector $\mathbf{s}_1$ involves sampling a random vector $\mathbf{s}_2$ with small coefficients and outputting the commitment vector

$$\mathbf{t} = \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 \in \mathcal{R}_q^{\kappa_{\mathsf{MSIS}}}.$$

To see that the commitment is hiding, observe that the vector $\mathbf{s}_2$ is much longer than the height of the matrix $\mathbf{A}_2$. Hence, under the (knapsack) Module-LWE assumption, $\mathbf{A}_2\mathbf{s}_2$ is indistinguishable from a random vector $\mathbf{u} \leftarrow \mathcal{R}_q^{\kappa_{\mathsf{MSIS}}}$. To prove binding, we note that if one can come up with two (possibly different) pairs $(\mathbf{s}_1, \mathbf{s}_2), (\mathbf{s}_1', \mathbf{s}_2')$ such that

$$\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t} = \mathbf{A}_1\mathbf{s}_1' + \mathbf{A}_2\mathbf{s}_2'$$

then one obtains a Module-SIS solution

$$\begin{bmatrix} \mathbf{s}_1 - \mathbf{s}_1' \\ \mathbf{s}_2 - \mathbf{s}_2' \end{bmatrix}$$

for the matrix $[\mathbf{A}_1\ \mathbf{A}_2]$. Hence, in order to obtain the binding property under the Module-SIS assumption, one can only have a message space consisting of vectors with small polynomial coefficients.

It is easy to see that the commitment is compact, i.e. it does not depend explicitly on the length of the message vector m_1.

UNBOUNDED-MESSAGE COMMITMENT. Next, we recall the BDLOP commitment scheme from [Bau+18b] which allows committing to an arbitrary vector of messages over $\mathcal{R}_q$. Suppose that we want to commit to a message vector $\mathbf{m} \in \mathcal{R}_q^{\ell}$. Then, in the key generation, a uniformly random matrices

$$\mathbf{A}_2 \leftarrow \mathcal{R}_q^{\kappa_{\mathsf{MSIS}} \times (\kappa_{\mathsf{MSIS}} + \kappa_{\mathsf{MLWE}} + \ell)}, \quad \mathbf{B} \leftarrow \mathcal{R}_q^{\ell \times (\kappa_{\mathsf{MSIS}} + \kappa_{\mathsf{MLWE}} + \ell)}$$

are generated and output as public parameters [2]. To commit to the message $\mathbf{m}$, we first sample the randomness vector $\mathbf{s}_2$. Now, there are two parts of the commitment scheme: (i) the binding part and (ii) the message encoding part. Concretely, we compute

$$\begin{aligned} \mathbf{t}_A &= \mathbf{A}_2\mathbf{s}_2, \\ \mathbf{t}_B &= \mathbf{B}\mathbf{s}_2 + \mathbf{m} \end{aligned}$$

2 In practice, one may choose to generate $\mathbf{A}_2, \mathbf{B}$ in a more structured way as in [Bau+18b] since it saves some computation. However, for readability, we write the commitment matrices in the "knapsack" form as above.

where the *top part* $\mathbf{t}_A \in \mathcal{R}_q^{\kappa_{\mathsf{MSIS}}}$ forms the binding part and the *bottom part* $\mathbf{t}_B \in \mathcal{R}_q^{\ell}$ encodes a message vector $\mathbf{m}$.

The hiding property of the BDLOP commitment scheme follows from the fact that if the randomness vector $\mathbf{s}_2 \in \mathcal{R}_q^{\kappa_{\mathsf{MLWE}}+\kappa_{\mathsf{MSIS}}+\ell}$ is long enough, then $\begin{bmatrix}\mathbf{A}_2 \\ \mathbf{B}\end{bmatrix}\mathbf{s}_2$ is computationally indistinguishable from a random vector $\mathbf{u} \leftarrow \mathcal{R}_q^{\kappa_{\mathsf{MSIS}}+\ell}$ under the Module-LWE assumption. On the other hand, to prove binding suppose that one can find two pairs $(\mathbf{s}_2, \mathbf{m}), (\mathbf{s}_2', \mathbf{m}')$ such that

$$\mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A = \mathbf{A}_2\mathbf{s}_2' \quad \text{and} \quad \mathbf{B}\mathbf{s}_2 + \mathbf{m} = \mathbf{t}_B = \mathbf{B}\mathbf{s}_2' + \mathbf{m}'.$$

Then, under the Module-SIS assumption for matrix $\mathbf{A}_2$, we obtain $\mathbf{s}_2 = \mathbf{s}_2'$. Furthermore, from the second equation we also get $\mathbf{m} = \mathbf{m}'$.

One observes that the commitment size as well as the length of the randomness vector $\mathbf{s}_2$ are linear in the length of the message vector. Hence, using this commitment for zero-knowledge proofs is much more expensive than the Ajtai commitment.

2.2 LATTICE-BASED ZERO-KNOWLEDGE PROOFS

Zero-knowledge proofs (ZKP), first introduced by Goldwasser, Micali, and Rackoff [GMR85], is a fundamental building block of various privacy-preserving applications, such as ring/group signatures, anonymous credentials, electronic voting, verifiable computation and cryptocurrencies. In this thesis, we restrict our attention to the ZKP constructions based on the hardness of lattice problems.

Lattice-based zero-knowledge proofs is an active area of current research which can be split into the following two groups. The first one focuses on proving statements tailored to practical applications [ALS20; BLS19; ENS20; Esg+19a; Lyu12; Yan+19], such as ring/group signatures [BKP20; ESZ21; Esg+19c; LNS21b], proving integer relations [LNS20] as well as blockchain [Esg+21; ESZ21; LNS21b]. The main drawback of using most of these protocols is the linear proof size in the length of the witness. Consequently, they are not suitable for proving larger statements. The latter group of works, however, concentrates on building protocols which offer asymptotically sub-linear (ideally poly-logarithmic) proof sizes [AL21; ACK21; Bau+18a; BCS21; Boo+20]. They additionally show how these constructions can be turned into efficient arguments of circuit satisfiability. Unfortunately, in terms of concrete sizes, they seem to be behind the schemes in the first

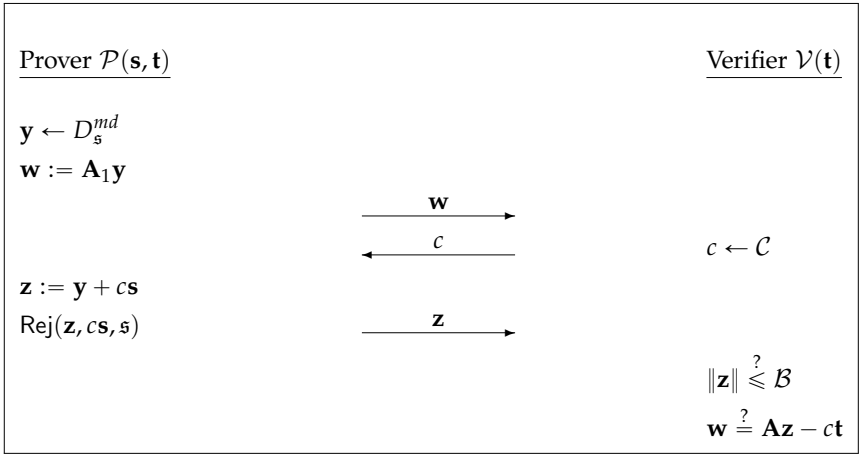

FIGURE 2.1: Identification scheme by Lyubashevsky [Lyu09; Lyu12] . Here, Rej is a rejection sampling algorithm to ensure zero-knowledge property of the protocol.

group with respect to smaller statements, mainly due to parameters which are neglected when doing an asymptotic analysis.

APPROXIMATE PROOFS. The starting point of practical lattice-based zero-knowledge proofs is an identification scheme by Lyubashevsky [Lyu09; Lyu12] which is an adaptation of the well-known Schnorr [Sch89] protocol to the lattice setting. In this scheme, which we sketch out in Figure 2.1, we want to prove knowledge of a secret vector $\mathbf{s} \in \mathcal{R}_q^m$ of small norm which satisfies $\mathbf{A}\mathbf{s} = \mathbf{t}$ over $\mathcal{R}_q$.

Let us consider the soundness property of the protocol in Figure 2.1. Using standard rewinding techniques, we can obtain two accepting transcripts $(\mathbf{w}, c, \mathbf{z})$ and $(\mathbf{w}, c', \mathbf{z}')$ with the same first message $\mathbf{w}$ and distinct challenges $c, c' \in \mathcal{C}$. From the verification equations we deduce that

$$\mathbf{A}(\mathbf{z} - \mathbf{z}') = (c - c')\mathbf{t} \quad \text{and} \quad \|\mathbf{z} - \mathbf{z}'\| \leqslant 2\mathcal{B}.$$

If one were to adapt the strategy from the discrete logarithm setting, then the next step would be to set

$$\bar{\mathbf{s}} := \frac{\mathbf{z} - \mathbf{z}'}{c - c'} \in \mathcal{R}_q^m$$

and conclude that $\mathbf{A}\bar{\mathbf{s}} = \mathbf{t}$. However, this comes with a few major issues. First, we observe that due to the verification condition $\|\mathbf{z}\| \leqslant \mathcal{B}$, it is essential

that coefficients of the challenges in $\mathcal{C}$ are relatively small. Hence, it is unclear that for distinct challenges $c, c' \in \mathcal{C}$ with small coefficients, the difference of $c - c'$ exists over $\mathcal{R}_q$. Even if it does, we have no guarantee that $\bar{\mathbf{s}}$ defined above has small norm since the coefficients $(c - c')^{-1}$ can be actually large.

Benhamouda et al. [Ben+14] showed that the challenge space $\mathcal{C} := \{X^i : i \in \mathbb{Z}_{2d}\} \subseteq \mathcal{R}_q$ *almost* circumvents all the problems discussed above. Namely, each $X^i \in \mathcal{C}$ has small coefficients, the difference $X^i - X^j$, for distinct $0 \leqslant i < j < 2d$, is invertible over $\mathcal{R}_q$ and $2/(X^i - X^j) \in \mathcal{R}_q$ has coefficients between -1 and 1. This implies that $2 \cdot \bar{\mathbf{s}}$ has small norm and $\mathbf{A}(2\bar{\mathbf{s}}) = 2\mathbf{t}$. The drawback of this approach is the size of the challenge space $\mathcal{C}$, which is $2d$, and thus we end up with a large soundness error. Therefore, we would need to further repeat the protocol for soundness amplification. Moreover, it was recently showed by Albrecht and Lai [AL21] that any challenge space that satisfies conditions mentioned above cannot have exponential size in the security parameter and thus any similar approach would require repeating the protocol for boosting soundness.

As noticed by Lyubashevsky, convincing the verifier that $\mathbf{A}\bar{\mathbf{z}} = \bar{c}\mathbf{t}$, for $\bar{\mathbf{z}} := \mathbf{z} - \mathbf{z}'$ and $\bar{c} := c - c'$, is enough for building simple, yet relatively efficient cryptographic primitives, such as signature schemes [Duc+18], verifiable encryption [LN17], and group signatures [BCN18]. As a concrete example, Dilithium signature scheme [Duc+18], which is one of the finalists of the NIST PQC Competition [NIS], produces signatures by essentially applying the Fiat-Shamir transformation [FS86] to the identification scheme from Figure 2.1. Nevertheless, the approach by Lyubashevsky [Lyu09; Lyu12] still does not prove exactly that $\mathbf{As} = \mathbf{t}$ and that $\mathbf{s}$ has small norm.

STERN PROOFS. There is a long line of research using Stern's protocol [Ste93] to exactly prove relations as in (1.1), e.g. [KTX08; Lin+13]. But even for the smallest statements, which for example arise when proving correctness of a Module-LWE sample, the proofs produced by this approach have several Megabytes in size and hence are not really practical. The reason behind this is that a single protocol execution has a very large soundness error of 2/3, and thus many protocol repetitions (in the order of hundreds) are required to reach a negligible soundness error.

PROTOCOLS BASED ON THE CRT SLOTS. More recent constructions by Bootle et al. [BLS19] and Yang et al. [Yan+19] allow proving exactly that

$\mathbf{As} = \mathbf{t}$ and that the coefficients of $\mathbf{s}$ are in a specified range, e.g. binary[3]. The key component of their protocols is the use of so-called CRT (or NTT, which stands for Number Theoretic Transform) slots. Namely, suppose that $q = 1 \pmod{2d}$ and thus $X^d + 1$ can be factored into linear terms [LS18] as follows:

$$X^d + 1 = \prod_{i=0}^{d-1} (X - r_i) \pmod{q}$$

where $r_0, \ldots, r_{d-1} \in \mathbb{Z}_q$ are distinct. Then, for a polynomial $a \in \mathcal{R}_q$, we define "CRT of a" to be the polynomial $\hat{a} \in \mathcal{R}_q$:

$$\hat{a} := \sum_{i=0}^{d-1} \hat{a}_i X^i \quad \text{where} \quad \hat{a}_i := a(r_i).$$

Similarly, we define the "inverse CRT of a" to be the polynomial $\check{a}$ for which CRT is equal to a.

One of the most useful properties of the CRT representation is that for any $a, b, c \in \mathcal{R}_q$: $\vec{a} \circ \vec{b} = \vec{c}$ if and only if $\check{a} \cdot \check{b} = \check{c}$, where $\vec{a}$ (resp. $\vec{b}, \vec{c}$) is the coefficient vector of a (resp. b, c) and $\circ$ is the component-wise product. Hence, to prove that $\mathbf{s} = (s_1, \ldots, s_m)$ has binary coefficients, i.e. $\vec{s}_i \circ (\vec{s}_i - \vec{1}) = \vec{0}$ for $i = 1, 2, \ldots, m$, we need to show that

$$\check{s}_i \cdot (\check{s}_i - 1) = 0 \quad \text{for } i = 1, 2, \ldots, m. \tag{2.1}$$

Since $\check{\mathbf{s}} := (\check{s}_1, \ldots, \check{s}_m)$ might actually have large coefficients, we cannot commit to $\check{\mathbf{s}}$ using the Ajtai commitment. Hence, [BLS19; Yan+19] commit to $\check{\mathbf{s}}$ using the BDLOP construction. Now, we show the intuition to prove (2.1). For simplicity, let us only consider the case $m = 1$. At some point during the protocol, the prover outputs the masked opening $z = y + \mu\check{s}$ to the verifier where $\mu \in \mathcal{R}_q$ is a challenge. Then, the verifier can compute

$$z(z - c) = y^2 + \mu \cdot (2\check{s} - 1)y + \mu^2 \cdot \check{s}(\check{s} - 1).$$

Hence, the idea is to send commitments

$$t_1 = \mathsf{Com}((2\check{s} - 1)y) \quad \text{and} \quad t_0 = \mathsf{Com}(y^2)$$

and given a challenge μ, output z and prove that $z(z - c) - \mu t_1 - t_0$ is a commitment to zero. This implies that $\check{s}(\check{s} - 1) = 0$. Now, if $m > 1$ then this

3 Esgin et al. [Esg+19a] also used the technique of CRT slots but directly in the context of building privacy-oriented primitives.

technique can be amortised so that one always send only two additional commitments instead of $2m$.

We still need to prove $\mathbf{As} = \mathbf{t}$. This becomes especially tricky since we committed to $\check{\mathbf{s}}$ and not $\mathbf{s}$. Here, the key observation is that if $\mu \in \mathbb{Z}_q$ and we send $z_i = y_i + \mu\check{s}_i$ as defined above, then:

$$\hat{\mathbf{z}} = \hat{\mathbf{y}} + \mu\mathbf{s}$$

where $\hat{\mathbf{z}} = (\hat{z}_1, \ldots, \hat{z}_m)$ and similarly for $\hat{\mathbf{y}}$. Thus, the verifier can compute

$$\mathbf{A}\hat{\mathbf{z}} = \mathbf{A}\hat{\mathbf{y}} + \mu\mathbf{t}.$$

Therefore, the idea is for the prover to send $\mathbf{w} := \mathbf{A}\hat{\mathbf{y}}$ in the clear and at the end the verifier checks that $\mathbf{A}\hat{\mathbf{z}} = \mathbf{w} + \mu\mathbf{t}$. Since $\mathbf{w}$ can be computed deterministically from the verification, it does not have to be a part of the non-interactive proof. However, this technique forces the requirement that μ has to be an integer. Consequently, the size of the challenge space of μ is at most q.

The one thing left to do is to show that all the commitments generated are actually valid. The aforementioned protocols use the proof from [Bau+18b] which requires the challenge space to satisfy that any difference of two distinct challenges is invertible. Baum et al. [Bau+18b] resolve this issue by picking a modulus q for which $X^d + 1$ does not split into many factors. Then, using the main result of Lyubashevsky and Seiler [LS18], they can choose an exponentially large challenge space of small polynomials. The invertibility result from [LS18], however, does not apply in the case $q = 1 \pmod{2d}$ and hence [BLS19; Yan+19] have to choose a challenge space $\mathcal{C} := \{X^i : i \in \mathbb{Z}_{2d}\}$. Consequently, the proof needs to be repeated $128/\log 2d$ times to obtain negligible soundness error which significantly increases the total proof size.

UNIFORMITY IN THE CRT SLOTS. Attema et al. [ALS20] generalise the result by Lyubashevsky and Seiler [LS18] and provide a way to compute the min-entropy of a challenge $c \leftarrow \mathcal{C}$ in a fixed CRT slot. Since an element in $\mathcal{R}_q$ is invertible if and only all its CRT slots are non-zero, it would suffice to show that the probability that a random c from the challenge set hits a particular value in a CRT slot is smaller than the targeted soundness error. Note that if c was picked uniformly at random from $\mathcal{R}_q$ then the probability that $c(r_i) = a$ for fixed $i \in \mathbb{Z}_d$ and $a \in \mathbb{Z}_q$ is exactly $1/q$. Attema et al. show that if coefficients of c are chosen from the set $\{-1, 0, 1\}$ then the probability becomes close to $1/q$. This result was later generalised by Esgin et al. [ESZ21] to consider challenges with fixed L_1 norm, and by Esgin et al. [Esg+22] to include challenges with larger coefficients than ternary.

Using the main result of [ALS20] in the setting when $q = 1 \pmod{2d}$, we obtain a proof of a BDLOP commitment opening with soundness error $1/q$ instead of $1/(2d)$ as in prior works. Moreover, Attema et al. propose a new proof of multiplication which has the following two advantages. First, it involves committing to one less polynomial than the degree of the equation. In the case of Equation 2.1, we would then commit to only one polynomial g_1, instead of two as in [BLS19]. More importantly, there is no need to send masked opening z_i of $\check{s}_i$ to the verifier. Since the secret vector $\mathbf{s}$ (and consequently $\check{\mathbf{s}}$) can be relatively long, polynomials $z_1, \ldots, z_m$ constitute a big part of the overall proof size.

EFFICIENT LINEAR PROOFS. Recently, Esgin et al. [ENS20] presented a new approach, which takes inspiration from the univariate sumcheck protocol [Ben+19], to prove $\mathbf{As} = \mathbf{t}$ without sending any masked openings z_i of $\check{s}_i$. Firstly, we can write this equation equivalently as a linear equation over $\mathbb{Z}_q$:

$$A\vec{s} = \vec{t} \pmod{q} \quad \text{where } A = \begin{bmatrix} A_1 & A_2 & \cdots & A_m \end{bmatrix} \in \mathbb{Z}_q^{nd \times md}.$$

The intuition is to let the verifier pick a challenge vector $\vec{\phi} \leftarrow \mathbb{Z}_q^{nd}$ and prove instead

$$\langle A\vec{s} - \vec{t}, \vec{\phi} \rangle = 0. \tag{2.2}$$

By simple transformation, this is equivalent to

$$\langle \vec{s}, A^T\vec{\phi} \rangle - \langle \vec{t}, \vec{\phi} \rangle = 0.$$

Further, Esgin et al. use the following fact. Namely, the sum of the CRT slots of a polynomial $f \in \mathcal{R}_q$ is equal to the (scaled) constant coefficient of f, or alternatively:

$$\sum_{i=0}^{d-1} f(r_i) = \frac{1}{d} \cdot \tilde{f}$$

where $\tilde{f}$ is the constant coefficient of f. Using this observation, it is enough to prove that the constant coefficient of the following polynomial:

$$f := \frac{1}{d} \left(\sum_{i=1}^{m} \check{s}_i \cdot \check{u}_i - \langle \vec{t}, \vec{\phi} \rangle \right)$$

is equal to zero where u_i is the (public) polynomial defined by its coefficient vector $A_i^T\vec{\phi} \in \mathbb{Z}_q^d$ for $i = 1, 2, \ldots, d$. A naive way to prove this statement

would be to send f in the clear and let the verifier check manually that $\tilde{f} = 0$. However, this would also reveal other coefficients of f and potentially reveal some information about $\vec{s}$. Instead, Esgin et al. reveal the constant coefficient of f while masking all the other coefficients. Concretely, the prover will commit to a random polynomial $g \leftarrow \{x \in \mathcal{R}_q : \tilde{x} = 0\}$ and send

$$h := g + f = g + \frac{1}{d}\left(\sum_{i=1}^{m} \check{s}_i \cdot \check{u}_i - \langle \vec{t}, \vec{\phi} \rangle\right).$$

Clearly, if $\tilde{f} = 0$ then also $\tilde{h} = 0$ and this can be checked manually by the verifier. On the other hand, other coefficients of h do not reveal any information about $\vec{s}$. What is left to prove is the well-formedness of h. However, since all $\check{s}_i$ and g are committed, this is just a linear proof in the committed polynomials and can be done identically as in [Bau+18b].

We highlight that if $A\vec{s} \neq \vec{t}$ then (2.2) holds with probability $1/q$. Since this value will be much larger than the targeted soundness error, naively one would need to repeat this part of the protocol $128/\log q$ times, i.e. commit to multiple polynomials $g_1, \ldots, g_{128/\log q}$. However, using certain properties of $\mathcal{R}$-automorphisms, Esgin et al. reduce the number of additional garbage commitments from $128/\log q$ to 1.

Combining the results from [ALS20], Esgin et al. obtain a very efficient proof system for proving statements of the form $\mathbf{As} = \mathbf{t}$ and that the (infinity) norm of $\mathbf{s}$ is small. For basic examples, their protocols enjoy $7 - 8X$ smaller proof sizes than [BLS19; Yan+19]. More recently, Lyubashevsky et al. [LNS21a] improved upon [ENS20] by applying a bimodal-like [Duc+13] rejection sampling strategy which results in the masked openings of the BDLOP randomness having smaller coefficients (by $2 - 3$ bits per coefficients) but each protocol execution reveals one bit of (fresh) randomness. Independently, Tao et al. [TWZ20] showed how to apply the bimodal technique to the BDLOP commitment scheme without any leakage. Consequently, they managed to reduce the standard deviation used for sampling the commitment randomness at the cost of relying on a Module-SIS problem with a larger bound. In our setting, however, this approach has very little improvement over the original opening proof [Bau+18b] with respect to efficiency.

In terms of concrete performance of the aforementioned works, we refer to Table 1.1 for more details.

SUBLINEAR PROOFS. The main bottleneck of the constructions described above is the proof size linear in the length of the committed witness. The

reason is that when proving knowledge of an opening of a BDLOP commitment, one sends a masked opening of the BDLOP randomness which is indeed linear in the size of the message. Hence, for solving more sophisticated statements than (1.1), e.g. circuit satisfiability, practically efficient sublinear-size proof systems are needed. There are several proposals of asymptotically sublinear lattice-based proof systems in the literature [AL21; ACK21; Bau+18a; Boo+20], but their concrete proof sizes are not analyzed in the papers and they are not practically efficient yet.

The first zero-knowledge proof with sublinear communication complexity for arithmetic circuit satisfiability was proposed by Baum et al. [Bau+18a]. At the core of the protocol lies an amortised proof of knowledge of vectors $\mathbf{s}_1, \ldots, \mathbf{s}_n \in \mathcal{R}_q^m$ of small norm, such that $\mathbf{A}\mathbf{s}_i = \mathbf{t}_i$ for $i = 1, 2, \ldots, n$. The total proof size is of the order of $O(n + m)$, hence square-root in the number of secret coefficients $N = nm$. This approach was later generalised by Bootle et al. [Boo+20] who define so-called "levelled commitments" and give $O(N^{1/d})$ size proofs for proving knowledge of a commitment opening with d levels[4]. The main drawback of this construction is that the modulus for the proof system increases exponentially in d and thus considering more than $2 - 3$ levels seems impractical.

Bootle et al. [Boo+20] also proposed the first lattice adaptation of the Bulletproofs [Boo+16; Bün+18] which offers poly-logarithmic proofs. This approach was later improved independently by Attema et al. [ACK21] and Albrecht and Lai [AL21] in terms of tighter soundness analysis and also generalised to a more abstract setting by Bootle et al. [BCS21].

While folding strategy from Bulletproofs is very effective in the discrete logarithm setting and retains asymptotic efficiency in the lattice scenario, they do not combine well with the shortness requirement in lattice cryptography. Consequently, this leads to a concrete blow-up of the parameters as well as the proof size. Informally, it must be possible to invert the folding in the extraction such that the extracted solution vector is still short. For general (small) challenges, this will not be the case. Hence, Bootle et al. [Boo+20] pick monomial challenges X^i so that (a scaled) inverse of a difference of two distinct challenges is still small [Ben+14]. This results in a large soundness error, and hence the protocol needs to be repeated for soundness amplification. Additionally, the length of the extracted solution vector grows by a factor of $O(d^3)$ for *every* level of folding. Then, the parameters must be chosen such that the Module-SIS is hard with respect to the length of the extracted solution vector, resulting in the need for a large

4 The construction by Baum et al. [Bau+18a] can be seen as a 2-level commitment.

modulus q. Concretely, even for ≈ 10 folding steps, the required modulus q would need to be in the order of several hundred bits which results in the proof size being in excess of 100 Megabytes for typical example applications.

POST-QUANTUM SECURITY. The broadly used Fiat-Shamir Transformation [FS86] turns an interactive ZKP into a non-interactive zero-knowledge proof (NIZK) in the random oracle model. In preparation for the eventual arrival of quantum computers, there has been a significant amount of work in understanding the *quantum* security in the quantum random oracle model (QROM) [Bon+11]. Until recently, many aforementioned protocols either (i) were not known to be (in)secure when applying Fiat-Shamir transformation in the QROM or (ii) could be transformed into a QROM secure NIZK using the Unruh transform [Unr15] which leads to a proof size increase by a factor of ≈ 50. Significant progress has been made by Katsumata [Kat21] who proved QROM security of the current state-of-the-art lattice-based zero-knowledge proofs [ALS20; BLS19; ENS20; LNS21a; Yan+19] at the cost of increasing the proof size by only a factor of 2.6. Since the protocols in this thesis have a similar structure as [BLS19; Yan+19], we believe that these techniques can also be applied in our setting.

2.3 LATTICE-BASED PRIVACY-ORIENTED PRIMITIVES

As evidenced in the literature, any development in building efficient lattice-based proofs of (1.1) brings new constructions of privacy-preserving applications from lattices. For instance, Stern proofs [KTX08; Lin+13] were used as a core component for constructing ring signatures [Lib+16], group signatures [Lib+16; Lin+17], pseudo-random functions with applications to e-cash [Lib+17] and proving integer relations [Lib+18]. Since the constructions relied on Stern proofs, their outputs were of the order of several Megabytes.

Once the early works on proving (1.1) based on the CRT technique appeared, significant improvements were made in the area of lattice-based privacy-oriented primitives. As an example, Yang et al. [Yan+19] showed that if one modifies the aforementioned constructions to use their new protocols instead of Stern proofs, one immediately obtains one order of magnitude improvement. Independently, a line of research started by Esgin et al. [Esg+19a; Esg+19b; ESZ21; Esg+19c] uses CRT techniques to build much more practical ring/group signatures and applications to sophisti-

cated primitives such as confidential transactions. The end result [ESZ21] is a Monero-like [NM16] lattice-based payment system where the communication complexity of a transaction is under 30KB.

As expected, the most efficient proof system for (1.1) [ALS20; ENS20; LNS21a] is also getting used in the context of privacy-oriented primitives, e.g. proving integer relations [LNS20], ring signatures and payment systems [LNS21b], group signatures [Lyu+21], blind signatures [LNP22a] and verifiable random functions [Esg+22]. Since the proof system is relatively new, more applications could emerge in the foreseeable future.

Due to the enormous amount of progress in the area of lattice-based privacy-preserving primitives, we restrict our attention to ring and group signatures.

RING SIGNATURES. First introduced by Rivest, Shamir and Tauman-Kalai [RST01], ring signatures allow for anonymous signature generation in a sense that the signer's identity is hidden within a public set of identities, called a ring.

One important aspect of ring signature schemes is the signature size and its growth with respect to the number of identities N in the ring. Lattice-based constructions can thus be split into the following two groups: (i) "linear-size" ring signatures, namely the signature size scales linearly in N and (ii) "logarithmic-size" ring signatures where the signature size is only poly-logarithmic in N (see Figure 1.4 for concrete comparison). Interestingly, the linear-size constructions, such as Raptor [LAZ19] and Dual-Ring [Yue+21], offer very small signature sizes in the range of $4-6$KB for less than 64 identities. However, their performance does not scale well for larger rings (more than 100MB for 2^{12} users).

There has been significant of work in building logarithmic-size ring signatures from lattices [BKP20; ESZ21; Esg+19c; LNS21b]. Most constructions follow the approach by Groth and Kohlweiss [GK15] and propose efficient one-out-of-many proofs in the lattice setting. Intuitively, in the lattice-based one-out-of-many proof, the signer wants to produce a zero-knowledge proof of knowledge of a short vector $\mathbf{s}$ such that

$$\mathbf{As} \in \{\mathbf{t}_1, \ldots, \mathbf{t}_N\} \subseteq \mathcal{R}_q^n. \tag{2.3}$$

In particular, the signing party does not want to reveal any information for which index i, $\mathbf{As} = \mathbf{t}_i$. The currently most efficient lattice-based ring signature for large number of users $N = \beta^k$ has been proposed by Lyubashevsky et al. [LNS21b]. To prove relations of the form (2.3), the authors show a new

way to prove knowledge of a short vector $\vec{s}$ along with the binary vectors $\vec{v}_1, \ldots, \vec{v}_k \in \{0,1\}^\beta$ with exactly one 1, such that

$$A\vec{s} = T(\vec{v}_1 \otimes \cdots \otimes \vec{v}_k)$$

where $\otimes$ is the standard Kronecker product[5]. Here, $\vec{v} := \vec{v}_1 \otimes \cdots \otimes \vec{v}_k \in \mathbb{Z}_q^N$ is exactly the vector that shows the position of a column of T which is equal to $A\vec{s}$. Even though the proof size in [LNS21b] is logarithmic in N, both the prover and verifier time are linear – thus making the protocol impractical to run for very large rings of identities.

GROUP SIGNATURES. First introduced by Chaum and van Heyst [CH91] and later formalised by Bellare et al. [BMW03], group signature scheme is another instance of an anonymous signature. In a group signature, the setup authority uses a master secret key to distribute member secret keys to the members of the group. The members can then use their secret keys to sign messages on behalf of the group. An entity known as the opener (or group manager) also has a special secret key that allows them to obtain the identity of the signer of any message, e.g. in case of a dispute or a misbehaviour.

Most of the early work in trying to construct lattice-based group signatures were efficient only in an asymptotic sense with concrete signature sizes being around 50MB (e.g. [GKV10; Lib+16]). Later on, del Pino et al. [PLS18] proposed a scheme with the signature size of around 580KB in which the parameters and computational complexity of signing and verifying do not depend on the group size.

The advancement of lattice-based zero-knowledge proofs using CRT slots [BLS19; Esg+19a; Yan+19] led to much more efficient constructions of group signatures. For example, recent schemes [Beu+21; ESZ21; Esg+19c] rely on efficient lattice-based OR/one-out-of-many proofs and achieve signature sizes under 100KB for large groups and even less than 20KB for a group of 1024 users. However, the signing and verifying time is linear in the number of users which makes them less attractive to run in practice. Independently, Lyubashevsky et al. [Lyu+21] proposed a group signature that builds upon the framework of [PLS18] and uses the efficient proof system from [ALS20; ENS20] as a building block. Thus, it inherits the property of [PLS18], i.e. a constant signature size as well as signing and verifying independent of the group size. The end result of [Lyu+21] is the group signature of size $\approx$ 200KB.

5 More precisely, we mean $\vec{v}_1 \otimes (\vec{v}_2 \otimes (\vec{v}_3 \cdots \otimes (\vec{v}_{k-1} \otimes \vec{v}_k)))$.

3

PRELIMINARIES

In this chapter, we cover relevant cryptographic as well as mathematical preliminaries that will be frequently used throughout the thesis. We start by introducing notation. Then, we recall necessary mathematical background which includes basic facts from linear algebra, lattices, discrete Gaussian distribution and algebraic number theory. Furthermore, we cover definitions of various cryptographic primitives (e.g. commitment scheme, commit-and-prove functionality) and state our security assumptions.

3.1 NOTATION

Let $\mathbb{Z}_n$ be the set of integers modulo n. Denote $\kappa \in \mathbb{N}$ to be a security parameter. Unless stated otherwise, all algorithms are implicitly given a security parameter in unary. An algorithm here is defined as an interactive Turing machine. Algorithms are randomised and PPT means "probabilistic polynomial time" in the security parameter κ. We describe $(y_1, \dots) \leftarrow \mathcal{A}(1^\kappa, x_1, \dots; r)$ as an event when $\mathcal{A}$ gets $(1^\kappa, x_1, \dots)$ as input, uses fresh random coins r and outputs $(y_1, \dots)$. The joint execution of two algorithms $\mathcal{A}$ and $\mathcal{B}$ is an interactive protocol with private inputs x to $\mathcal{A}$ and y to $\mathcal{B}$ is written as $(a, b) \leftarrow \langle \mathcal{A}(x), \mathcal{B}(y) \rangle$ where a and b are the private outputs of $\mathcal{A}$ and $\mathcal{B}$ respectively. The notation $\mathcal{A}^{(\cdot)}$ means that $\mathcal{A}$ expects a black-box access to some other algorithm.

We write $x \leftarrow S$ when $x \in S$ is sampled uniformly at random from the finite set S and similarly $x \leftarrow D$ when x is sampled according to the discrete distribution D. The statistical distance between two probability distributions X and Y over a countable set D is defined as $\Delta(X, Y) = \sum_{d \in D} |X(d) - Y(d)|$. For integer $n \in \mathbb{N}$, we define $[n] := \{1, 2, \dots, n\}$. A function $v : \mathbb{N} \to \mathbb{R}^{\geqslant 0}$ is negligible if for any $c \in \mathbb{N}$, $\lim_{\kappa \to \infty} v(\kappa)\kappa^c = 0$. For $f, g : \mathbb{N} \to \mathbb{R}^{\geqslant 0}$, we write $f \approx g$ if $|f(\kappa) - g(\kappa)|$ is a negligible function. We say that an event, which is dependent on κ, happens with negligible probability if the probability that the event occurs is negligible in κ. Similarly, an event happens with overwhelming probability if its complement occurs with negligible probability. We write negl to denote an unspecified negligible

function. Similarly, we denote by $\mathsf{poly}(\kappa)$ an unspecified polynomial in κ. We denote log and ln to denote logarithms with base 2 and e respectively.

MODULAR REDUCTION. For an odd (resp. even) integer p, we define $r' = r \bmod^{\pm} p$ to be the unique element r' in the range $-\frac{p-1}{2} \leqslant r' \leqslant \frac{p-1}{2}$ (resp. $-\frac{p}{2} < r' \leqslant \frac{p}{2}$) such that $r' = r \bmod p$. We also denote $r' = r \bmod^{+} p$ to be the unique element r' in the range $0 \leqslant r' < p$ such that $r' = r \bmod p$. When the exact representation is not important, we simply write $r \bmod p$.

MATRICES AND VECTORS. Regular lower-case letters denote elements in $\mathbb{Z}$ and lower-case letters with arrows (resp. upper-case regular letters) represent column vectors (resp. matrices) with coefficients in $\mathbb{Z}$. Given two vectors $\vec{v} = (v_1, \ldots, v_n), \vec{w} = (w_1, \ldots, w_n)$ over $\mathbb{Z}$, we define the inner product as

$$\langle \vec{v}, \vec{w} \rangle := \sum_{i=1}^{n} v_i w_i \in \mathbb{Z}$$

and the component-wise product as $\vec{v} \circ \vec{w} = (v_1 w_1, \ldots, v_n w_n) \in \mathbb{Z}^n$. For a rank-$n$ matrix $S \in \mathbb{R}^{m \times n}$, we define the set $U_S := \{\|S\vec{u}\| : \vec{u} \in \mathbb{R}^n, \|\vec{u}\|_2 = 1\}$. Then, the least (resp. largest) singular value of S is defined as $s_n(S) = \inf U_S$ (resp. $s_1(S) = \sup U_S$).

For an element w in $\mathbb{Z}_p$, we write $\|w\|_\infty$ to mean $|w \bmod^{\pm} p|$. Define the L_∞ and L_α norms for $\vec{w} = (w_1, w_2, \ldots, w_n)$ over $\mathbb{Z}_p$ as follows:

$$\|\vec{w}\|_\infty = \max_{j \in [n]} \|w_j\|_\infty, \quad \|\vec{w}\|_\alpha = \sqrt[\alpha]{\|w_1\|_\infty^\alpha + \ldots + \|w_n\|_\infty^\alpha}.$$

By default, $\|\vec{w}\| := \|\vec{w}\|_2$.

3.2 MATHEMATICAL BACKGROUND

3.2.1 *Lattices*

An n-dimensional lattice $\mathbf{\Lambda}$ is a discrete subgroup of $\mathbb{R}^n$. Suppose $\mathbf{B} = \{\vec{b}_1, \ldots, \vec{b}_m\} \in \mathbb{R}^n$ consists of m linearly independent vectors. Then, the n-dimensional lattice $\mathbf{\Lambda}$ generated by B is defined as

$$\mathbf{\Lambda} = \mathcal{L}(B) = \left\{ \sum_{i=1}^{m} c_i \vec{b}_i : c_1, \ldots, c_m \in \mathbb{Z} \right\}.$$

The determinant of the lattice $\mathbf{\Lambda}$ is defined as $\det(\mathbf{\Lambda}) := \sqrt{\det(B^T B)}$. The minimum distance $\lambda_1^\alpha(\mathbf{\Lambda})$ of a lattice $\mathbf{\Lambda}$ in the L_α norm is the length of the shortest non-zero vector in $\mathbf{\Lambda}$, i.e. $\min_{\vec{x}\in\mathbf{\Lambda}\setminus\{\vec{0}\}} \|\vec{x}\|_\alpha$. Similarly, define $\Lambda_1^\infty(\mathbf{\Lambda})$ for the L_∞ norm. We denote $\lambda_1(\mathbf{\Lambda}) := \lambda_1^2(\mathbf{\Lambda})$, i.e. the shortest non-zero vector in $\mathbf{\Lambda}$ w.r.t. L_2 norm.

We recall the following upper-bound on the shortest non-zero vector in $\mathbf{\Lambda}$ which follows directly from Minkowski's Theorem.

Lemma 3.2.1. *For any n-dimensional lattice* $\mathbf{\Lambda}$, $\lambda_1(\mathbf{\Lambda}) \leqslant \sqrt{n}\det(\mathbf{\Lambda})^{1/n}$.

3.2.2 *Probability Distributions*

DISCRETE GAUSSIAN DISTRIBUTION ON LATTICES. We first define a Gaussian function on $\mathbb{R}^m$ centred at $\vec{v} \in \mathbb{R}^m$ with parameter $\mathfrak{s}$ as:

$$\rho_{\vec{v},\mathfrak{s}}(\vec{x}) := \left(\frac{1}{\sqrt{2\pi\mathfrak{s}^2}}\right)^m \exp\left(-\frac{\|\vec{x}-\vec{v}\|^2}{2\mathfrak{s}^2}\right).$$

When $\vec{v} = \vec{0}$, we just write $\rho_\mathfrak{s}$.

Now, the discrete Gaussian distribution over $\mathbb{Z}^m$ centred at some $\vec{v} \in \mathbb{Z}^m$ with standard deviation $\mathfrak{s}$ is defined as follows:

$$D_{\vec{v},\mathfrak{s}}(\vec{x}) := \frac{\rho_{\vec{v},\mathfrak{s}}(\vec{x})}{\rho_\mathfrak{s}(\mathbb{Z}^m)}.$$

As before, the subscript $\vec{v}$ is omitted when $\vec{v} = 0$.

We recall the following tail bounds from [Ban93; Lyu12].

Lemma 3.2.2. *Let* $m, k > 1, r > 0$ *and* $\vec{v} \in \mathbb{R}^m$. *Then*

1. $\Pr_{z\leftarrow D_\mathfrak{s}}\left[|z| > k\mathfrak{s}\right] \leqslant 2e^{\frac{-k^2}{2}}$.
2. $\Pr_{\vec{z}\leftarrow D_\mathfrak{s}^m}\left[\|\vec{z}\|_2 > k\mathfrak{s}\sqrt{m}\right] \leqslant k^m e^{\frac{m}{2}(1-k^2)}$.
3. $\Pr_{\vec{z}\leftarrow D_\mathfrak{s}^m}\left[|\langle\vec{z},\vec{v}\rangle| > r\right] \leqslant 2e^{-\frac{r^2}{2\|\vec{v}\|^2\mathfrak{s}^2}}$.

BINOMIAL DISTRIBUTION. Next, we recall the binomial distribution.

Definition 3.2.1. The binomial distribution with a positive integer parameter k, written as Bin_k, is the distribution $\sum_{i=1}^k (a_i - b_i)$, where $a_i, b_i \leftarrow \{0,1\}$. The variance of this distribution is $k/2$ and it holds that $\mathsf{Bin}_{k_1} \pm \mathsf{Bin}_{k_2} = \mathsf{Bin}_{k_1+k_2}$.

3.2.3 *Approximate Shortness Test*

A well-known result of Johnson and Lindenstrauss says that any set of n points in m-dimensional Euclidean space can be embedded into a much smaller k-dimensional Euclidean space, where $k = O(\log n)$ and independent of m, so that all pairwise distances are preserved within an arbitrarily small factor. In practical scenarios, such embeddings are simply random projections. Baum and Lyubashevsky [BL17] applied this result in the context of proving shortness of a committed vector $\vec{w} \in \mathbb{Z}^m$. Concretely, the idea is to choose a random rectangular matrix $R \leftarrow \mathsf{Bin}_1^{k\times m}$, where k is only dependent on the security parameter, and prove that the projection $\vec{v} = R\vec{w}$ with respect to R has small norm. We consider two particular norms, i.e. the L_2 and L_∞ norms.

SHORTNESS IN THE L_∞ NORM. Baum and Lyubashevsky [BL17] showed that if $R\vec{w}$ has small coefficients, for a vector $\vec{w}$ over $\mathbb{Z}_q$ and uniformly random binary matrix R, then with high probability $\vec{w}$ must have small coefficients as well. We will generalise their result in two aspects: (i) we show that it also holds when $R\vec{w} + \vec{y}$ has small coefficients, where $\vec{y}$ is an arbitrary vector over $\mathbb{Z}_q$, and (ii) when R is sampled from a distribution centred at 0, i.e. Bin_1. The main advantage of the latter generalisation is that the L_2/L_∞ norm of $R\vec{s}$ decreases significantly.

Lemma 3.2.3. *Let $\vec{w} \in \mathbb{Z}_q^m$ and $\vec{y} \in \mathbb{Z}_q^k$. Then*

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

Proof. Let $y \in \mathbb{Z}_q$. We first focus on proving

$$\Pr_{\vec{r}\leftarrow\mathsf{Bin}_1^k}\left[\|\langle\vec{r},\vec{w}\rangle + y\rangle\|_\infty < \frac{1}{2}\|\vec{w}\|_\infty\right] \leqslant \frac{1}{2}.$$

Let w_i be the coefficient of $\vec{w}$ so that $\|w_i\|_\infty = \|\vec{w}\|_\infty$. Then, one can write $\langle\vec{r},\vec{w}\rangle + y = w_i r_i + a$ for some $a \in \mathbb{Z}_q$. We consider two cases.

CASE 1: $\|a\|_\infty \geqslant \frac{1}{2}\|\vec{w}\|_\infty$. Then, r_i would have to be either 1 or -1 for any chance of $w_i r_i + a$ to be less than $\frac{1}{2}\|\vec{w}\|_\infty$. This implies that

$$\Pr_{\vec{r}\leftarrow\mathsf{Bin}_1^k}\left[\|\langle\vec{r},\vec{w}\rangle + y\rangle\|_\infty < \frac{1}{2}\|\vec{w}\|_\infty \middle| \|a\|_\infty \geqslant \frac{1}{2}\|\vec{w}\|_\infty\right] \leqslant \Pr_{r_i\leftarrow\mathsf{Bin}_1}[|r_i| = 1] = \frac{1}{2}.$$

CASE 2: $\|a\|_\infty < \frac{1}{2}\|\vec{w}\|_\infty$. We will prove that

$$\|a + bw_i\|_\infty \geqslant \frac{1}{2}\|w_i\|_\infty$$

for any $b \in \{-1, 1\}$. Therefore, we have

$$\Pr_{\vec{r}\leftarrow \mathsf{Bin}_1^k}\left[\|\langle \vec{r}, \vec{w}\rangle + y\rangle\|_\infty < \frac{1}{2}\|\vec{w}\|_\infty \middle| \|a\|_\infty < \frac{1}{2}\|\vec{w}\|_\infty\right] \leqslant \Pr_{r_i\leftarrow \mathsf{Bin}_1}[r_i = 0] = \frac{1}{2}$$

which will complete the proof of the lemma.

First, we can assume that $|w_i| \leqslant q/2$ and $|a| < |w_i|/2$. Thus, $\|a + bw_i\|_\infty$ is either equal to $|a + bw_i|$ or $|a + bw_i \pm q|$. In the former case, we immediately have $|a + bw_i| \geqslant |bw_i| - |a| > |w_i|/2$. For the latter case, we can assume for the sake of contradiction that $u = a + bw_i \pm q$ where $|u| < |w_i|/2$. Therefore,

$$q = |\pm q| = |a + bw_i - u| \leqslant |a| + |bw_i| + |u| < |w_i|/2 + |w_i| + |w_i|/2 \leqslant q.$$

This result can then be easily generalised to the matrix setting. Hence, the statement holds.

□

SHORTNESS IN THE L_2 NORM. In many lattice-based scenarios, we are more interested in proving the L_2 norm of a vector rather than its L_∞. Indeed, even the definition of the shortest vector in a lattice is by default over the Euclidean norm. Recently, Gentry et al. [GHL21] propose an analogous result to Lemma 3.2.3 in the L_2 norm. First, they provide a detailed analysis on how to pick parameters α, β such that probabilities

$$\Pr_{R\leftarrow \mathsf{Bin}_\mu^{256\times m}}\left[\|R\vec{w}\|^2 < \|\vec{w}\|^2 \cdot \alpha\right] \text{ and } \Pr_{R\leftarrow \mathsf{Bin}_\mu^{256\times m}}\left[\|R\vec{w}\|^2 > \|\vec{w}\|^2 \cdot \beta\right]$$

are negligible for any $\vec{w} \in \mathbb{Z}^m$. Their analysis relies on the following two heuristics which stem from two lemmas proved by Achlioptas [Acho3]. Firstly, it is shown in [Acho3, Lemma 6.1] that all the respective moments of the distribution $\|R\vec{w}\|$ are largest among $\vec{w}$ of norm $\sqrt{m}$ if $\vec{w} = 1^m$. Further, [Acho3, Lemma 6.3] says that if we change the distribution of R to be the normal distribution with the same mean and variance, then the moments of $\|R\vec{w}\|$ are larger. Consequently, this means the tails of the continuous distribution are fatter, and thus bounding them will imply bounds on the discrete distribution. Note that discretization might cause certain errors, that should become negligible if we look for α, β for which the probabilities above are negligible, e.g. less than 2^{-128}.

Now, the distribution $\|R \cdot 1^d\|$, where entries of R are chosen from the normal distribution with mean 0 and variance $\mu/2$, simply becomes the scaled χ^2 distribution with 256 degrees of freedom, i.e. $\frac{\mu}{2} m \cdot \chi^2[256]$. Hence, we obtain the following (heuristic) generalisation of [GHL21, Corollary 3.2].

Lemma 3.2.4. *Under the heuristic substitution of* Bin_μ *with the normal distribution of variance* $\mu/2$*, for any* $\vec{w} \in \mathbb{Z}^m$*,*

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

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

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

Lemma 3.2.5. *Fix* $m, q \in \mathbb{N}$ *and a bound* $b \leqslant q/41m$*, and let* $\vec{w} \in [\pm q/2]^m$ *with* $\|\vec{w}\| \geqslant b$*, and let* $\vec{y}$ *be an arbitrary vector in* $[\pm q/2]^m$*. Then*

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

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

Claim 3.2.6. Fix $m, q \in \mathbb{N}$ and a bound $b \leqslant q/(41m)$, and let $\vec{w} \in [\pm q/2]^m$ with $\|\vec{w}\| \geqslant b$. Then

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

Proof. Similarly as in the proof of Lemma 3.2.3 we have two cases:

CASE 1: $\|\vec{w}\|_\infty \geqslant q/(4m)$. Let i be an index of an entry in $\vec{w}$ with magnitude at least $q/4m$, and consider any row $\vec{r}$ of R. Then, we can write $\vec{r}^T\vec{w} = r_i w_i + a$ for some $a \in \mathbb{Z}_q$. Note that at most one of the three values $\{0, \pm 1\}$ for r_i yields $|r_i w_i + a \bmod q| < q/(8m)$. Indeed, first suppose that we have $|w_i + a \bmod q| < q/(8m)$ and $|-w_i + a \bmod q| < q/(8m)$. Then, by the triangle inequality, we have $|2w_i \bmod q| < q/(4m)$ which leads to contradiction. Next, assume that for some sign $b \in \{-1, 1\}$, $|bw_i + a \bmod q| < q/(8m)$ and $|a \bmod q| < q/(8m)$. Then, by the triangle inequality we get $|w_i| = |bw_i| < q/(4m)$ which is a contradiction.

Since the total probability of any two of $\{-1, 0, 1\}$ is at least $1/2$ (i.e. $\Pr[0] = 3/8$ and $\Pr[\pm 1] = 1/4$), we have that the probability of $\|R\vec{w} \bmod q\|_\infty < q/(8m)$ is at most 2^{-256}. Moreover, since $b \leqslant q/(41m)$ we get $q/(8m) > b\sqrt{26}$ and therefore

$$\begin{aligned}\Pr_{R \leftarrow \mathsf{Bin}_2^{256\times m}}[\|R\vec{w} \bmod q\| < b\sqrt{26}] &\leqslant \Pr_{R \leftarrow \mathsf{Bin}_2^{256\times m}}[\|R\vec{w} \bmod q\|_\infty < q/8m] \\ &\leqslant 2^{-256}.\end{aligned}$$

CASE 2: $\|\vec{w}\|_\infty < q/(4m)$. Hence, we must have $R\vec{w} \in [\pm q/2]^{256}$, so mod-$q$ reduction has no effect and the statement follows directly from Lemma 3.2.4.

□

Now, suppose for contradiction that for some $\vec{w}, \vec{y}$,

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

which implies

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

is at most 2^{-256}. By the triangle inequality, we have

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

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

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

which is a contradiction with the statement of Lemma 3.2.6.

□

3.2.4 *Power-of-Two Cyclotomic Rings*

Let d be a power-of-two and $K = \mathbb{Q}[X]/(X^d + 1)$ be the $2d$-th cyclotomic field. Denote $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$ to be the ring of integers of K. Suppose that $p \equiv 2l + 1 \pmod{4l}$ for some $l \in \mathbb{N}$. Then, by [LS18, Corollary 1.2], the polynomial $X^d + 1$ factors as:

$$X^d + 1 \equiv \prod_{i=0}^{l-1} (X^{d/l} - r_i) \pmod{p}$$

for distinct $r_i \in \mathbb{Z}_p^*$ where $X^{d/l} - r_i$ are irreducible in the ring $\mathbb{Z}_p[X]$. In other words, the ideal (p) in $\mathcal{R}$ can be uniquely written as a product of prime ideals $(p) = \mathfrak{p}_0\mathfrak{p}_1 \ldots \mathfrak{p}_{l-1}$ where each $\mathfrak{p}_i = (p, X^{d/l} - r_i)$. Define $\zeta = r_0$. Then, $\{r_0, \ldots, r_{l-1}\} = \{\zeta, \zeta^3, \ldots, \zeta^{2l-1}\}$. Without loss of generality, we set $r_i = \zeta^{2i+1}$ for $i = 0, 1, \ldots, l-1$. Finally, denote $\mathcal{R}_p = \mathcal{R}\backslash(p) = \mathbb{Z}_p[X]/(X^d + 1)$.

COEFFICIENT VECTORS AND ROTATION MATRICES. Lower-case letters denote elements in $\mathcal{R}$ or $\mathcal{R}_p$ and bold lower-case (resp. upper-case) letters represent column vectors (resp. matrices) with coefficients in $\mathcal{R}$ or $\mathcal{R}_p$. Let $f = f_0 + f_1 X + \ldots + f_{d-1} X^{d-1}$ be a polynomial in $\mathcal{R}$. Then, we denote $\vec{f} := (f_0, \ldots, f_{d-1}) \in \mathbb{Z}^d$ to be the coefficient vector of f, i.e. we attach an arrow to the letter. Similarly, for $\mathbf{f} = (f_1, \ldots, f_k) \in \mathcal{R}^k$, we write $\vec{f} \in \mathbb{Z}^{kd}$ to mean the concatenation of vectors $\vec{f_1}, \ldots, \vec{f_k}$. We define the rotation (or alternatively, skew-circulant) matrix $\mathsf{Rot}(f)$ as:

$$\mathsf{Rot}(f) = \begin{bmatrix} f_0 & -f_{d-1} & \cdots & -f_1 \\ f_1 & f_0 & \cdots & -f_2 \\ \vdots & \vdots & \cdots & \vdots \\ f_{d-1} & f_{d-2} & \cdots & f_0 \end{bmatrix} \in \mathbb{Z}^{d \times d}.$$

Similarly, for a matrix $\mathbf{F} = (f_{i,j}) \in \mathcal{R}^{n \times m}$, we define

$$\mathsf{Rot}(\mathbf{F}) = \begin{bmatrix} \mathsf{Rot}(f_{1,1}) & \mathsf{Rot}(f_{1,2}) & \ldots & \mathsf{Rot}(f_{1,m}) \\ \vdots & \vdots & \vdots & \vdots \\ \mathsf{Rot}(f_{n,1}) & \mathsf{Rot}(f_{n,2}) & \ldots & \mathsf{Rot}(f_{n,m}) \end{bmatrix} \in \mathbb{Z}^{nd \times md}.$$

One observes that for any $f, g, h \in \mathcal{R}$, $gf = h$ if and only if $\mathsf{Rot}(g)\vec{f} = \vec{h}$.

By default, for a polynomial, we write its i-th coefficient as its corresponding regular font letter subscript i, e.g. $f_{d/2} \in \mathbb{Z}$ is a middle coefficient of $f \in \mathcal{R}$. However, we also define $\tilde{f}$ to be the constant coefficient of f.

Given two vectors $\mathbf{f}, \mathbf{g}$ over $\mathcal{R}$, we denote $\langle \mathbf{f}, \mathbf{g} \rangle$ to be the inner product between their coefficient vectors over $\mathbb{Z}$, i.e. $\langle \mathbf{f}, \mathbf{g} \rangle := \langle \vec{f}, \vec{g} \rangle \in \mathbb{Z}$.

GALOIS AUTOMORPHISMS. Let $\mathsf{Aut}(\mathcal{R}) := \{\sigma_i : i \in \mathbb{Z}_{2d}^{\times}\}$ be the automorphism group of $\mathcal{R}$ where each automorphism $\sigma_i : \mathcal{R} \to \mathcal{R}$ is defined by $\sigma_i(X) = X^i$. Then, G is isomorphic to $\mathbb{Z}_{2d}^{\times} \cong \mathbb{Z}_2 \times \mathbb{Z}_{d/2}$. For a vector $\mathbf{x} = (x_1, \ldots, x_k) \in \mathcal{R}^k$ and any $\sigma \in \mathsf{Aut}(\mathcal{R})$, denote $\sigma(\mathbf{x}) := (\sigma(x_1), \ldots, \sigma(x_k))$ (and similarly $\sigma(\mathbf{X})$ for a matrix $\mathbf{X}$ over $\mathcal{R}$).

NORMS. For $\mathbf{f} \in \mathcal{R}_p^k$, we define the L_α norm of $\mathbf{f}$ as $\|\mathbf{f}\|_\alpha := \|\vec{f}\|_\alpha$. Finally, we define a set $S_k = \{s \in \mathcal{R} : \|s\|_\infty \leqslant k\}$ for $k \in \mathbb{N}$.

In this thesis, we will make use of the following inequalities.

Lemma 3.2.7 ([Mico7]). *Let $c, r \in \mathcal{R}_p$. Then*

$$\|c \cdot r\|_\infty \leqslant \|c\|_\infty \cdot \|r\|_1 \text{ and } \|c \cdot r\|_\infty \leqslant \|c\| \cdot \|r\|.$$

We additionally present an alternative way to bound $\|c\mathbf{r}\|$ which stems from the analysis in [Duc+13, Section 4] and uses the σ_{-1} automorphism.

Lemma 3.2.8. *Let $\mathbf{r} \in \mathcal{R}^\ell$ and $c \in \mathcal{R}$. Then, for any power-of-two k, we have $\|c\mathbf{r}\| \leqslant \sqrt[2k]{\|\sigma_{-1}\left(c^k\right) c^k\|_1} \cdot \|\mathbf{r}\|$.*

Proof. Let $C = \mathsf{Rot}(c) \in \mathbb{Z}^{d \times d}$. We simply want to upper-bound the largest singular norm $s_1(C)$ of the matrix C. We will use the following two facts from linear algebra. Namely, we have that $s_1(C) = \sqrt{s_1(C^T C)}$ and for every power-of-two k,

$$s_1^k(C^T C) = s_1\left((C^T C)^k\right)$$

since $C^T C$ is symmetric. Also, note that for any $u, v \in \mathcal{R}$, $\|uv\| \leqslant \|u\|_1 \cdot \|v\|$, and thus $s_1(\mathsf{Rot}(u)) \leqslant \|u\|_1$. Therefore, using the observation that $C^T = \mathsf{Rot}(\sigma_{-1}(c))$, we deduce that

$$s_1^{2k}(C) = s_1^k\left(C^T C\right) = s_1\left((C^T C)^k\right) = s_1\left(\mathsf{Rot}(\sigma_{-1}(c^k)c^k)\right) \leqslant \|\sigma_{-1}(c^k)c^k\|_1.$$

Hence, $s_1(C) \leqslant \sqrt[2k]{\|\sigma_{-1}\left(c^k\right) c^k\|_1}$ and thus the statement holds. □

INVERTIBILITY OF SHORT POLYNOMIALS. A polynomial $c \in \mathcal{R}_p$ is invertible if and only if for all $i \in \mathbb{Z}_l, c \bmod \mathfrak{p}_i \neq 0$. Lyubashevsky and Seiler [LS18] showed that if c has a small norm then c is invertible over $\mathcal{R}_p$.

Lemma 3.2.9 ([LS18]). *Let $p \equiv 2l+1 \pmod{4l}$ be a prime and $d \geqslant 4$. Then, any $c \in \mathcal{R}_p$ which satisfies either $0 < \|c\|_\infty < \frac{1}{\sqrt{l}} p^{1/l}$ or $0 < \|c\| < p^{1/l}$ is invertible in $\mathcal{R}_p$.*

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

Lemma 3.2.10. *Let $p \equiv 5 \pmod 8$ be a prime. Take any $c \in \mathcal{R}_p$ such that $\sigma_{-1}(c) = c$. Then, c is invertible over $\mathcal{R}_p$ if and only if $c \neq 0$.*

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

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

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

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

Now, we observe that

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

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

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

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

WORKING OVER COMPOSITE MODULUS. In our protocols, we will work over the ring $\mathcal{R}_q := \mathcal{R}/(q)$ where q is a product of odd primes $q_1 < \ldots < q_n$ and each $q_i \equiv 2l + 1 \pmod{4l}$. Usually, $n \in \{1, 2\}$. Then, by the Chinese Remainder Theorem, an element $c \in \mathcal{R}_q$ is invertible if and only if $c \bmod q_i$ is invertible over $\mathcal{R}_{q_i}$ for all $i \in [n]$. Hence, by Lemma 3.2.9, if $0 < \|c\| < {q_1}^{1/l}$ then c is invertible over $\mathcal{R}_q$. Moreover, if each $q_i \equiv 5 \pmod 8$ then we can apply Lemma 3.2.10 which says that if a non-zero $c \in \mathcal{R}_q$ satisfies $\|c\|_\infty < q_1$ and $\sigma_{-1}(c) = c$ then it is invertible over $\mathcal{R}_q$. Additionally, note that if we fix any $a, u \in \mathcal{R}_q$ such that $a \neq 0$ then

$$\Pr_{c \leftarrow \mathcal{R}_q}[ac = u] \leqslant q_1^{-d/l}.$$

WORKING OVER SUBRINGS. Our proof system will natively support equations over the ring $\mathcal{R}$ of dimension d. However, when building various privacy-preserving primitives, it would be more efficient to construct them over a ring $\mathcal{R}' = \mathbb{Z}[X]/(X^{kd} + 1)$ of much larger dimension kd than d where k is also a power-of-two (e.g. to reduce the public key size). Consequently, we would need to be able to prove equations over the larger ring $\mathcal{R}'$ rather than $\mathcal{R}$. Here, we show that equations over $\mathcal{R}'$ can be equivalently written as equations over $\mathcal{R}$.

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

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

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

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

Thus, $(0, \ldots, 0)$ and $(1, 0, \ldots, 0)$ are the additive and multiplicative identities respectively.

Now, we prove the following lemma.

Lemma 3.2.11. *Let $k \geqslant 1$ be a power-of-two. Then, $\mathcal{R}' := \mathbb{Z}[X]/(X^{kd} + 1) \cong S^k$.*

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

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

We claim that ϕ is a ring isomorphism. Bijection follows immediately since one can define the inverse map $\phi^{-1}((a_0, \ldots, a_{k-1})) := \sum_{i=0}^{k-1} a_i X^i \in \mathcal{R}'$. Also, $\phi(1) = (1, 0, \ldots, 0)$.

Now, fix any $a = \sum_{i=0}^{k-1} a_i X^i$ and $b = \sum_{i=0}^{k-1} b_i X^i$ in $\mathcal{R}'$. Clearly, we have $\phi(a) + \phi(b) = \phi(a + b)$. Then, for multiplication, observe that

$$\begin{aligned} ab = \left(\sum_{i=0}^{k-1} a_i X^i\right)\left(\sum_{j=0}^{k-1} b_j X^j\right) &= \sum_{\substack{0 \leqslant i,j < k \\ i+j<k}} a_i b_j X^{i+j} + \sum_{\substack{0 \leqslant i,j<k \\ i+j \geqslant k}} a_i b_j X^k X^{i+j-k} \\ &= \sum_{0 \leqslant i,j<k} \left(a_i b_j X^{\left\lfloor \frac{i+j}{k} \right\rfloor k}\right) X^{(i+j) - \left\lfloor \frac{i+j}{k} \right\rfloor k} \\ &= \sum_{\ell=0}^{k-1} c_\ell X^\ell, \end{aligned}$$

where

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

Hence, by definition of $\star$ we have $\phi(a) \star \phi(b) = \phi(ab)$. □

Example. Suppose we want to transform the equation $ab \equiv c \pmod q$ over $\mathcal{R}'$ into an equivalent system of equations over $\mathcal{R}_q$. First, we know that $ab \equiv c \pmod q$ if and only if there exists some $d \in \mathcal{R}'$ such that $ab = c + qd$. This equation can then be written equivalently as

$$\phi(a) \star \phi(b) = \phi(c) + \phi(q) \star \phi(d). \tag{3.1}$$

Define $\phi(a) = (a_0(X^k), \ldots, a_{k-1}(X^k))$, where each $a_\ell \in \mathcal{R}$ and therefore $a_\ell(X^k) \in S$, and similarly for $\phi(b), \phi(c), \phi(d)$. Also, note that $\phi(q) \star \phi(d) = (qd_0(X^k), \ldots, qd_{k-1}(X^k))$. Then, (3.1) holds if and only if for every $\ell = 0, \ldots, k-1$ we have:

$$\sum_{\substack{0 \leqslant i,j<k \\ i+j \equiv \ell \bmod k}} a_i(X^k) b_j(X^k) X^{\left\lfloor \frac{i+j}{k} \right\rfloor k} = c_\ell(X^k) + qd_\ell(X^k)$$

which is then equivalent to

$$\sum_{\substack{0 \leqslant i,j < k \\ i+j \equiv \ell \bmod k}} a_i b_j X^{\left\lfloor \frac{i+j}{k} \right\rfloor} = c_\ell + q d_\ell$$

over $\mathcal{R}$. Hence, we conclude that $ab \equiv c \pmod q$ if and only if :

$$\forall \ell \in \mathbb{Z}_k, \sum_{\substack{0 \leqslant i,j < k \\ i+j \equiv \ell \bmod k}} a_i b_j X^{\left\lfloor \frac{i+j}{k} \right\rfloor} \equiv c_\ell \pmod q. \tag{3.2}$$

Remark. In various scenarios, apart from proving equations over $\mathcal{R}'$ (or $\mathcal{R}'/(q)$), one also needs to prove that certain vectors have small coefficients. For instance, suppose we want to prove $ab = c$ over $\mathcal{R}'/(q)$ and $\|b\|_\infty \leqslant 1$. It is easy to see that the map ϕ preserves the norm, i.e. $\|\phi(b)\|_\alpha = \|b\|_\alpha$ for $\alpha \in \{1, 2, \ldots, \infty\}$. Hence, in addition to proving (3.2), we would also need to prove that for all $\ell \in \mathbb{Z}_k$, $\|b_\ell\|_\infty \leqslant 1$.

We conclude that we can keep the dimension d suitable for our proof system while having the freedom to pick larger dimension kd when instantiating the primitive.

3.3 CRYPTOGRAPHIC DEFINITIONS

3.3.1 *Security Assumptions*

Security of our constructions relies on the well-known computational lattice problems, namely Module-LWE (MLWE) and Module-SIS (MSIS) [LS15]. Both problems are defined over $\mathcal{R}_q$. Clearly, if we substitute $\mathcal{R}_q$ with $\mathbb{Z}_q$ then these problems become plain SIS [Ajt96] and LWE [Reg09] problems.

Definition 3.3.1 ($\mathsf{MSIS}_{n,m,B}$)**.** Given $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times m}$, the *Module-SIS* problem with parameters $n, m > 0$ and $0 < B < q$ asks to find $\mathbf{z} \in \mathcal{R}_q^m$ such that $\mathbf{Az} = \mathbf{0}$ over $\mathcal{R}_q$ and $0 < \|\mathbf{z}\| \leqslant B$. An algorithm $\mathcal{A}$ is said to have advantage ϵ in solving $\mathsf{MSIS}_{n,m,B}$ if

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

We say that $\mathsf{MSIS}_{n,m,B}$ is hard if for all PPT adversaries $\mathcal{A}$, the advantage in solving $\mathsf{MSIS}_{n,m,B}$ is negligible.

Definition 3.3.2 ($\mathsf{MLWE}_{m,n,\chi}$)**.** The (knapsack) *Module-LWE* problem with parameters $n, m > 0$ and an error distribution χ over $\mathcal{R}$ asks the adversary $\mathcal{A}$ to distinguish between the following two cases: 1) $(\mathbf{A}, \mathbf{As} \bmod q)$ for $\mathbf{A} \leftarrow \mathcal{R}_q^{n\times(n+m)}$, a secret vector $\mathbf{s} \leftarrow \chi^{n+m}$ and 2) $(\mathbf{A}, \mathbf{b}) \leftarrow \mathcal{R}_q^{n\times(n+m)} \times \mathcal{R}_q^n$. Then, $\mathcal{A}$ is said to have advantage ϵ in solving $\mathsf{MLWE}_{m,n,\chi}$ if

$$\begin{aligned}\Big|\Pr\Big[b = 1 \,\Big|\, \mathbf{A} \leftarrow \mathcal{R}_q^{n\times(n+m)};\, \mathbf{s} \leftarrow \chi^{n+m};\, b \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{As} \bmod q)\Big] \\ - \Pr\Big[b = 1 \,\Big|\, \mathbf{A} \leftarrow \mathcal{R}_q^{n\times(n+m)};\, \mathbf{b} \leftarrow \mathcal{R}_q^{n};\, b \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{b})\Big]\Big| \geqslant \epsilon.\end{aligned} \tag{3.3}$$

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

Hardness of MSIS/MLWE problems is often analysed identically as the plain SIS/LWE since, so far, the best known attacks do not make use of the algebraic structure of the polynomial ring [Alk+16]. In order to estimate the practical MSIS hardness, we apply the methodology used in [Duc+18, Appendix C] and [AHJ21, Section 3.4]. Note that solving $\mathsf{MSIS}_{n,m,B}$ is equivalent to finding a non-trivial vector of norm smaller than B in the following ideal lattice

$$\mathbf{\Lambda} = \{\vec{z} \in \mathbb{Z}^{md} : (\mathbf{z} \in \mathcal{R}) \wedge \mathbf{Az} \equiv \mathbf{0} \pmod q\}.$$

In order to find short non-trivial vectors in $\mathbf{\Lambda}$, we apply the Block-Korkine-Zolotarev algorithm (BKZ) [CN11; SE94]. As a subroutine, BKZ uses an algorithm for the shortest vector problem (SVP) in lattices of dimension b, where b is called the block size. If we apply the best known algorithm for solving SVP with no memory constraints by Becker et al. [Bec+16], the time required by BKZ to run on the md-dimensional lattice $\mathbf{\Lambda}$ with block size b is given by $8md \cdot 2^{0.292b+16.4}$. The algorithm outputs a vector of norm $\delta^{md} \det(\mathbf{\Lambda})^{\frac{1}{md}}$ where δ is the root Hermite factor and it is given by

$$\delta = \left(\frac{b(\pi b)^{1/b}}{2\pi e}\right)^{\frac{1}{2(b-1)}}. \tag{3.4}$$

For our usual parameter selection, the probability that a random matrix $\mathbf{A} \in \mathcal{R}_q^{n\times m}$ is of full rank is overwhelming (see [Esg+19c, Appendix C] or the "knapsack" MLWE problem below) and thus $\det(\mathbf{\Lambda}) = q^{nd}$. Next, Micciancio and Regev [MR09] show that

$$\delta^{md} \det(\mathbf{\Lambda})^{\frac{1}{md}} = \delta^{md} q^{\frac{nd}{md}} \geqslant 2^{2\sqrt{nd \log q \log \delta}}$$

level of bit security	80	128	256
root Hermite factor δ	1.0066	1.0044	1.0025

FIGURE 3.1: Values of the root Hermite factor for specific levels of bit security based on Equation 3.4 for full-rank lattices of dimension at least 128.

and the equality holds when $md = \sqrt{nd \log q / \log \delta}$. Hence, given a bound $B < q$ we compute δ from the equation $B = 2^{2\sqrt{nd \log q \log \delta}}$. Next, we calculate the minimum block size b from Equation 3.4 and thus obtain the total time for BKZ to solve $\mathsf{MSIS}_{n,m,B}$. In order to compare with previous works, e.g. [ALS20; BLS19; Esg+19c], we set $\delta = 1.0044$ when aiming for 128-bit security (see Figure 3.1).

Further, we recall that the knapsack MLWE is as hard as the original version of MLWE [Esg+19c; MM11], up to an additive factor which is the probability that a uniformly random matrix $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times (n+m)}$ is singular. Indeed, suppose that q is a product of k primes $q_1 < \cdots < q_k$ and each $q_i \equiv 2l + 1 \pmod{4l}$. Then, Esgin et al. [Esg+19c, Appendix C] show that the probability that a uniformly random matrix $\mathbf{A} \leftarrow \mathcal{R}_q^{n \times (n+m)}$ has full rank is at least

$$\left(1 - q_1^{-(m+1)d/l}\right)^{knl}.$$

In our instantiations, we will pick $(k, d, l) = (2, 64, 2)$ and $q_1 > 2^{15}, n \geqslant 18$. Thus, the value above can be lower-bounded by $1 - 2^{-450}$. Hence, we conclude that the probability that random $\mathbf{A}$ is singular is negligible and thus knapsack MLWE is practically equivalent to the standard Module-LWE.

We estimate the hardness of Module-LWE against known attacks using the LWE estimator by Albrecht et al. [APS15]. Namely, we run the estimator under both "sieving" and "enumeration", and set the final root Hermite factor δ as the largest root Hermite factor returned by the program. Similarly as above, we aim for $\delta = 1.0044$. We remark that parameter m does not play a crucial role in estimating hardness of MLWE as long as it is not too large with respect to $n \cdot d$. Indeed, in our constructions $m \approx n$ and thus the BKW [BKW03] and Arora–Ge [AG11] attacks are not applicable here.

3.3.2 *Commitment Schemes*

A commitment scheme $\mathsf{Com} = (\mathsf{Com.KeyGen}, \mathsf{Com.Commit}, \mathsf{Com.Open})$ is a triple of algorithms described below.

- Com.KeyGen is a PPT algorithm that on input security parameter 1^κ outputs public parameters pp, which specify the message, randomness and commitment spaces $\mathcal{S}_M, \mathcal{S}_R, \mathcal{S}_T$. They also specify an efficiently sampleable probability distribution $\mathcal{D}$ over $\mathcal{S}_R$ and a set of *relaxation factors* $\mathcal{S}_C$[1].
- Com.Commit is a deterministic polynomial-time commitment function, that on input public parameters pp, message $m \in \mathcal{S}_M$ and randomness $r \in \mathcal{S}_R$, outputs a commitment $t \in \mathcal{S}_T$. We write $\mathsf{Com.Commit}(pp, m)$ to denote the PPT algorithm which first samples randomness $r \leftarrow \mathcal{D}$ and then outputs $\mathsf{Com.Commit}(pp, m; r)$.
- Com.Open is a deterministic polynomial-time algorithm that, on input the public parameters pp, a tuple $(m, r, c; t) \in \mathcal{S}_M \times \mathcal{S}_R \times \mathcal{S}_C \times \mathcal{S}_T$ outputs a bit b which indicates "accept" when $b = 1$ and "reject" otherwise.

The latter two algorithms are always given the public parameters, hence for readability we will omit writing pp as an input to Com.Commit and Com.Open.

Definition 3.3.3 (Correctness)**.** We say that a commitment scheme Com is *correct* if there exists $e_{\mathsf{id}} \in \mathcal{S}_C$ such that for all $m \in \mathcal{S}_M$,

$$\Pr\left[\mathsf{Com.Open}(m, r, e_{\mathsf{id}}; t) = 1 : r \leftarrow \mathcal{D}, t = \mathsf{Com.Commit}(m; r)\right] = 1.$$

We now describe two essential properties of commitment schemes, i.e. *hiding* and *binding*. In this thesis we are only interested in their computational variants.

Definition 3.3.4 (Hiding)**.** The commitment scheme is computational hiding if for all PPT adversaries $\mathcal{A}$

$$\Pr\left[\begin{array}{l} pp \leftarrow \mathsf{Com.KeyGen}(1^\kappa); (m_0, m_1) \leftarrow \mathcal{A}(pp); b \leftarrow \{0,1\}; \\ r \leftarrow \mathcal{D}; t \leftarrow \mathsf{Com.Commit}(m_b; r) : \mathcal{A}(t) = b \end{array}\right] \approx \frac{1}{2},$$

where $\mathcal{A}$ outputs $m_0, m_1 \in \mathcal{S}_M$.

Definition 3.3.5 (Binding)**.** The commitment scheme is computational binding if for all PPT adversaries $\mathcal{A}$

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathsf{Com.KeyGen}(1^\kappa); (t, \mathbf{y}_0, \mathbf{y}_1) \leftarrow \mathcal{A}(pp) : \\ m_0 \neq m_1 \text{ and } \mathsf{Com.Open}(\mathbf{y}_0; t) = \mathsf{Com.Open}(\mathbf{y}_1; t) = 1 \end{array}\right] \approx 0,$$

1 This is a crucial property of lattice-based commitment schemes and it will become clear when analysis concrete instantiations in the next chapter.

where $\mathbf{y}_i = (m_i, r_i, c_i)$ for $i = 0, 1$. Moreover, Com is strongly computational binding if we replace $m_0 \neq m_1$ in the definition above with $(m_0, r_0) \neq (m_1, r_1)$.

3.3.3 *Commit-and-Prove Functionality*

Let $R \subseteq \{0,1\}^* \times \{0,1\}^*$ be a binary relation. If $(u, w) \in R$, we say that u is a statement and w is a witness for u. We denote $R(u) = \{w : R(u, w) = 1\}$. In this thesis we only consider NP relations R for which a witness w can be verified in time $\mathsf{poly}(|u|)$ for all $(u, w) \in R$. We also assume that the length of all statements in R are polynomial in the security parameters, i.e. $|u| = \mathsf{poly}(\kappa)$.

A proof system $\Pi = (\mathcal{P}, \mathcal{V})$ for relation R consists of two interactive and stateful PPT algorithms $\mathcal{P}$ and $\mathcal{V}$ which are called the prover and verifier. We write $(tr, b) \leftarrow \langle \mathcal{P}(u, w), \mathcal{V}(u) \rangle$ for running $\mathcal{P}$ and $\mathcal{V}$ on inputs u, w and u respectively and getting communication transcript tr and the verifier's decision bit b. We use the convention that $b = 0$ means reject and $b = 1$ means accept the prover's claim of knowing w such that $(u, w) \in R$. If tr contains a $\perp$ then we say that $\mathcal{P}$ aborts. Unless stated otherwise, we will assume that the first and the last message are sent from a prover. Hence, the protocol between $\mathcal{P}$ and $\mathcal{V}$ has an odd number of rounds.

In this thesis, relations we consider have a very specific form. Roughly speaking, we first commit to a witness w and then prove certain statements about w. More formally, let Com be a commitment scheme and define a relation $R^{(\mathsf{Com})}$ relative to R[2]:

$$R^{(\mathsf{Com})} := \left\{ \begin{array}{c} ((u, pp, t), (w, r, c)) : \\ (u, w) \in R \wedge \mathsf{Com.Open}_{pp}(w, r, c; t) = 1 \end{array} \right\}.$$

In the literature, this approach is called commit-and-prove, e.g. [Can+02; EG14]. Alternatively, one can think of this functionality as a standard proof system where we additionally require the prover to generate and output a commitment t to the witness w in the very first round.

Example. Let $R_{\mathsf{yes}} := \{0,1\}^* \times \{0,1\}^*$. Then, it is easy to find a witness to any statement. Hence by definition, $R_{\mathsf{yes}}^{(\mathsf{Com})}$ becomes simply a relation where the statement contains a commitment and the witness is the commitment opening.

2 Note that even if relation R is trivial, i.e. given u, it is easy to find w so that $(u, w) \in R$, it is not the case for $R^{(\mathsf{Com})}$ if Com is binding and hiding.

Formally, we define the commit-and-prove functionality (in an interactive form) as follows. Namely, a commit-and-prove system for a relation R is a triple $\Pi = (\mathsf{Com}, \mathcal{P}, \mathcal{V})$ where Com is a commitment scheme and $\mathcal{P}$ and $\mathcal{V}$ are interactive and stateful PPT algorithms. Now, we describe security properties of a commit-and-prove system, i.e. completeness, knowledge soundness and simulatability.

Definition 3.3.6 (Completeness)**.** $\Pi = (\mathsf{Com}, \mathcal{P}, \mathcal{V})$ has statistical completeness with correctness error $\epsilon(\kappa)$ if for all adversaries $\mathcal{A}$,

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathsf{Com.KeyGen}(1^\kappa); (u, w) \leftarrow \mathcal{A}(pp); r \leftarrow \mathcal{D} \\ (tr, b) \leftarrow \langle \mathcal{P}(u, pp, t), (w, r)), \mathcal{V}(u, pp, t)\rangle : ((u, pp, t), (w, r, e_{\mathsf{id}})) \in R^{(\mathsf{Com})} \text{ and } b = 0 \end{array}\right]$$

is at most $\epsilon(\kappa) + \mathsf{negl}(\kappa)$ for all $\kappa \in \mathbb{N}$ where $t := \mathsf{Com.Commit}_{pp}(w; r)$.

Next, we introduce the notion of knowledge soundness.

Definition 3.3.7 (Knowledge Soundness)**.** $\Pi = (\mathsf{Com}, \mathcal{P}, \mathcal{V})$ is knowledge sound with knowledge error $\epsilon : \mathbb{N} \to [0, 1)$ if there exists an algorithm $\mathcal{E}$, called a knowledge extractor, with the following property. Namely, given a statement-commitment tuple (u, pp, t) and a black-box oracle access to a probabilistic prover $\mathcal{P}^*$, which convinces the verifier $\mathcal{V}(u, pp, t)$ with probability $\varepsilon > \epsilon(\kappa)$, the extractor runs in an expected polynomial time and with probability at least

$$\frac{\varepsilon - \epsilon(\kappa)}{\mathsf{poly}(\kappa)}$$

outputs either a triple (w, r, c) which satisfies $((u, pp, t), (w, r, c)) \in R^{(\mathsf{Com})}$ or two tuples $(w, r, c), (w', r', c')$ so that $(w, r) \neq (w', r')$ and

$$\mathsf{Com.Open}_{pp}(w, r, c; t) = \mathsf{Com.Open}_{pp}(w, r', c'; t).$$

We observe that an extractor either extracts a witness or breaks the strong binding property of Com. In our examples, winning the strong binding game of a commitment implies solving the MSIS problem. Technically, since knowledge soundness will rely on the computational assumptions, our protocols are *arguments* rather than proofs.

We say the protocol is *public coin* if the verifier's challenges are chosen uniformly at random independently of the prover's messages.

We introduce a new notion called *simulatability*. Informally, it means that there exists an efficient simulator $\mathcal{S}$ which can simulate both the commitment generation and the proof at the same time. The difference between

simulatability and (non-abort special honest-verifier) zero-knowledge is that randomness r is directly generated from an honest party as it would be in the real-world applications rather than chosen from an adversary. This property becomes crucial when using the commitment introduced in Chapter 4.

Definition 3.3.8 (Simulatability)**.** A public-coin commit-and-prove system $\Pi = (\mathsf{Com}, \mathcal{P}, \mathcal{V})$ is said to be simulatable if there exists a PPT simulator $\mathcal{S}$ such that for all PPT stateful adversaries $\mathcal{A}$,

$$\Pr\left[\begin{array}{c} pp \leftarrow \mathsf{Com.KeyGen}(1^\kappa); (u, w, \varrho) \leftarrow \mathcal{A}(pp); r \leftarrow \mathcal{D}; t = \mathsf{Com.Commit}_{pp}(w, r); \\ (tr, b) \leftarrow \langle \mathcal{P}((pp, u, t), (w, r)), \mathcal{V}(pp, u, t; \varrho) \rangle : (u, w) \in R \text{ and } \mathcal{A}(t, tr) = 1 \end{array}\right]$$

$$\approx \Pr\left[\begin{array}{c} pp \leftarrow \mathsf{Com.KeyGen}(1^\kappa); (u, w, \varrho) \leftarrow \mathcal{A}(pp); (t, tr) \leftarrow \mathcal{S}(pp, u, \varrho) : \\ (u, w) \in R \text{ and } \mathcal{A}(t, tr) = 1 \end{array}\right],$$

whenever $\mathcal{P}$ does not abort. Here ϱ is the randomness used by the verifier.

Remark. Let us argue why this notion is useful in practice. First, by definition of simulatability, if we naturally transform the commit-and-prove system Π into a standard proof system Π', where the prover generates and sends the commitment to the witness w in the first round, the simulatability of Π implies non-abort special honest-verifier zero-knowledge (SHVZK) of the proof system Π'. One might wonder why we neglected the issue of simulating the aborted transcripts. Luckily, this type of zero-knowledge definition is enough for most of privacy-oriented applications, because when transforming Π' into a non-interactive protocol using the Fiat-Shamir heuristics [FS86], we can repeat certain parts of the protocol until a non-abort occurs. Since the verifier only sees the non-aborting transcripts, only these should be simulated.

In an interactive setting, a standard approach to modify the protocol to be able to simulate aborting transcripts [Bau+18a; BLS19; Dam+21] is as follows. Namely, we commit to the messages, which would be sent before the prover potentially aborts, and only reveal them if no abort occurs (e.g. when rejection sampling goes through). This method was recently formalised by Damgård et al. [Dam+21] in the context of two-round n-out-of-n and multi-signatures where it is essential to be able to simulate the aborted executions.

3.3.4 *Techniques for Proving Knowledge Soundness*

Various techniques have been developed to prove knowledge soundness property, such as forking lemma [BN06; Bün+18; HKL19; PLS19], splitting lemma [HS03; PS00] or its simplified variant [Dam10]. In this thesis we will apply the strategy from [ACK21; AFK21] to extract transcripts.

Our knowledge extraction approach can be described by the following *collision game* [ACK21]. Let $k \in \mathbb{N}$ and consider a binary matrix $H \in \{0,1\}^{R \times N}$ where $N > k$. Informally, the R rows correspond to the prover's randomness and the N columns correspond to the verifier's randomness, or alternatively, the verifier samples a challenge $c \leftarrow C$ uniformly at random where C has size N. An entry of H equals 1 if and only if the corresponding protocol transcript is accepting. The knowledge extractor will run the following collision game.

1. First, sample $(r, i) \leftarrow [R] \times [N]$ and check if $H(r, i) = 1$. If not, it aborts.
2. If $H(r, i) = 1$, then it samples $i^* \leftarrow [N]$ without replacement until it obtains distinct $i_1^*, \ldots, i_{k-1}^*$ such that $H(r, i_l^*) = 1$ for $l = 1, 2, \ldots, k-1$.

The following lemma states the expected runtime and success probability of the algorithm above.

Lemma 3.3.1 ([ACK21]). *Let $H \in \{0,1\}^{R \times N}$ and define ϵ to be the fraction of 1-entries in H. Then, the expected number of H-entries queried in the collision game is at most k and the probability of the collision-game is at least $\epsilon - \frac{k-1}{N}$.*

Proof. We first focus on the expected number of queries of the collision game. Let X be the number of queries to H. For $r \in [R]$, define ϵ_r to be the fraction of 1-entries in the r-th row. Note that if the entry checked in the first step is of the form $(r, \cdot)$ and equals 1 then the second step can be modelled by a negative hypergeometric distribution. In this case, Attema et al. [ACK21] show that the expected number of draws is at most $(k-1)/\epsilon_r$. Hence, we can compute $\mathbb{E}[X]$ as follows. Let success be the event that the first entry queried by the algorithm is 1 (i.e. first step passes). Then,

$$\begin{aligned}\mathbb{E}[X] &= \frac{1}{|R|} \sum_{j=1}^{R} \mathbb{E}[X|\mathsf{success} \wedge r = j] \cdot \epsilon_j + \mathbb{E}[X|\neg\mathsf{success} \wedge r = j] \cdot (1 - \epsilon_j) \\ &\leqslant \frac{1}{|R|} \sum_{j=1}^{R} \left(1 + \frac{k-1}{\epsilon_j}\right) \cdot \epsilon_j + 1 \cdot (1 - \epsilon_j) = k.\end{aligned}$$

Now, we concentrate on the success probability of the collision game. Let (r, i) be the randomness and challenge sampled in the first step. We want to compute the probability $H(r, i) = 1$ and that there are at least k 1-entries in the r-th row. Let T_j be the fraction of rows which have exactly j 1-entries. Then,

$$\sum_{j=k}^{R} T_j \frac{j}{N} = \left(\sum_{j=0}^{R} T_j \frac{j}{N} \right) - \sum_{j=0}^{k-1} T_j \frac{j}{N} \geqslant \epsilon - (k-1) \cdot \frac{\sum_{j=0}^{k-1} T_j}{N} \geqslant \epsilon - \frac{k-1}{N}$$

which concludes the proof. □

3.3.5 *Rejection Sampling*

In lattice-based zero-knowledge proofs, e.g. [ALS20; BLS19], the prover will want to output a vector $\mathbf{z}$ whose distribution should be independent of a secret randomness vector $\mathbf{r}$, so that $\mathbf{z}$ cannot be used to gain any information on the prover's secret. During the protocol, the prover computes $\mathbf{z} = \mathbf{y} + c\mathbf{r}$ where $\mathbf{r}$ is the randomness used to commit to the prover's secret, $c \leftarrow \mathcal{C}$ is a challenge polynomial, and $\mathbf{y}$ is a "masking" vector. In order to remove the dependency of $\mathbf{z}$ on $\mathbf{r}$, one applies the *rejection sampling* technique [Lyu12].

We first formally define a rejection sampling algorithm as follows.

Definition 3.3.9. A rejection sampling algorithm Rej is an efficient probabilistic algorithm which takes as input a secret $\vec{v} \in \mathbb{Z}^\ell$, masking $\vec{z} \in \mathbb{Z}^\ell$, standard deviation $\mathfrak{s}$ and a repetition rate M. Then, it outputs a bit b. We say that Rej rejects if $b = 1$ and accepts when $b = 0$.

Below we recall commonly used rejection sampling algorithms in the literature.

Lemma 3.3.2 ([Lyu12]). *Let $V \subseteq \mathbb{Z}^\ell$ be a set of polynomials with norm at most T and $\rho \colon V \to [0, 1]$ be a probability distribution. Fix the standard deviation $\mathfrak{s} = \gamma T$ and*

$$M = \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma} + \frac{1}{2\gamma^2} \right).$$

Now, sample $\vec{v} \leftarrow \rho$ and $\vec{y} \leftarrow D^\ell_{\mathfrak{s}}$, set $\vec{z} = \vec{y} + \vec{v}$, and run $b \leftarrow \mathsf{Rej}_0(\vec{z}, \vec{v}, \mathfrak{s}, M)$ as defined in Fig. 3.2. Then, the probability that $b = 0$ is at least $(1 - 2^{-\kappa})/M$ and the distribution of $(\vec{v}, \vec{z})$, conditioned on $b = 0$, is within statistical distance of $2^{-\kappa}$ of the product distribution $\rho \times D^\ell_{\mathfrak{s}}$.

$\underline{\mathsf{Rej}_0(\vec{z}, \vec{v}, \mathfrak{s}, M)}$
$u \leftarrow [0,1)$
If $u > \frac{1}{M} \cdot \exp\left(\frac{-2\langle \vec{z}, \vec{v}\rangle + \|\vec{v}\|^2}{2\mathfrak{s}^2}\right)$
  return 1
Else
  return 0

$\underline{\mathsf{Rej}_1(\vec{z}, \vec{v}, \mathfrak{s}, M)}$
$u \leftarrow [0,1)$
If $u > \frac{1}{M \exp\left(-\frac{\|\vec{v}\|^2}{2\mathfrak{s}^2}\right) \cosh\left(\frac{\langle \vec{z}, \vec{v}\rangle}{\mathfrak{s}^2}\right)}$
  return 1
Else
  return 0

FIGURE 3.2: Standard (left) and bimodal (right) rejection sampling algorithms.

In certain scenarios, we will also use the bimodal Gaussian rejection sampling which was first introduced by Ducas et al. [Duc+13]. The main difference from the standard rejection sampling is that we additionally sample a sign $\beta \leftarrow \{-1, 1\}$ and mask $\vec{v}$ by setting $\vec{z} := \vec{y} + \beta\vec{v}$. Thanks to the reflective symmetry of the distribution of $\vec{z}$, we significantly reduce the standard deviation (e.g. by a factor of 10 if we aim for $M = 3$). The important part is, however, not to reveal any information about the bit β to the verifier. To this end, we apply zero-knowledge proofs to commit to β and prove that $\beta \in \{-1, 1\}$.

Lemma 3.3.3 ([Duc+13]). *Let $V \subseteq \mathbb{Z}^\ell$ be a set of polynomials with norm at most T and $\rho\colon V \to [0,1]$ be a probability distribution. Fix the standard deviation $\mathfrak{s} = \gamma T$ and*

$$M = \exp\left(\frac{1}{2\gamma^2}\right).$$

Now, sample $\vec{v} \leftarrow \rho$ and $\vec{y} \leftarrow D^\ell_{\mathfrak{s}}$ and $\beta \leftarrow \{-1, 1\}$, set $\vec{z} = \vec{y} + \beta\vec{v}$, and run $b \leftarrow \mathsf{Rej}_1(\vec{z}, \vec{v}, \mathfrak{s}, M)$ as defined in Fig. 3.2. Then, the probability that $b = 0$ is at least $1/M$ and the distribution of $(\mathbf{v}, \mathbf{z})$, conditioned on $b = 0$, is identical to the the product distribution $\rho \times D^\ell_{\mathfrak{s}}$.

3.3.6 *Challenge Space*

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

σ	d	l	ω	η	$\lvert S_\omega^\sigma \rvert$	$\lvert \mathcal{C} \rvert$
σ_1	128	2	1	27	2^{202}	2^{201}
σ_{-1}	128	2	2	59	2^{148}	2^{147}
σ_{-1}	64	2	8	140	2^{130}	2^{129}

FIGURE 3.3: Example parameters to instantiate the challenge space $\mathcal{C}$ for a modulus q such that its smallest prime divisor q_1 is greater than 16.

such vectors. We will use the inequality described in Lemma 3.2.8. In order to apply this result, we set the challenge space $\mathcal{C}$ as:

$$\mathcal{C} := \left\{ c \in S_\omega^\sigma : \sqrt{\|\sigma_{-1}(c)c\|_1} \leqslant \eta \right\} \tag{3.5}$$

where

$$S_\omega^\sigma := \{ c \in S_\omega : \sigma(c) = c \}. \tag{3.6}$$

and the $\sigma \in \mathsf{Aut}(\mathcal{R})$ will be specified in our protocols. Also, we denote $\bar{\mathcal{C}} := \{c - c' : c, c' \in \mathcal{C} \text{ and } c \neq c'\}$ to be the set of differences of any two distinct elements in $\mathcal{C}$. In practice, $\sigma \in \{\sigma_1, \sigma_{-1}\}$. We will choose the constant η such that (experimentally) the probability for $c \leftarrow S_\omega^\sigma$ to satisfy $\sqrt{\|\sigma_{-1}(c)c\|_1} \leqslant \eta$ is at least 99%. In our experiments, we observe that the bounds in Lemma 3.2.8 are about $4 - 6$X larger than the actual norms $\|c\mathbf{r}\|$.

For security of our protocols, we need $\omega < \frac{1}{2\sqrt{l}} q_1^{1/l}$ to ensure the invertibility property of the challenge space $\mathcal{C}$, i.e. the difference of any two distinct elements of $\mathcal{C}$ is invertible over $\mathcal{R}_q$ by Lemma 3.2.9. However, if q is a product of primes $q_i \equiv 5 \pmod 8$ and we want the challenges c to be stable under the σ_{-1} automorphism, i.e. $\sigma = \sigma_{-1}$, then we can apply Lemma 3.2.10 and set $\omega < q_1/2$.

Further, to achieve negligible soundness error under the MSIS assumption, we will need $|\mathcal{C}|$ to be exponentially large. In Table 3.3 we propose example parameters to instantiate the challenge space $\mathcal{C}$ for different automorphisms σ. Finally, for implementation purposes, in order to sample from $\mathcal{C}$, we simply generate $c \leftarrow S_\omega^\sigma$ and check whether $\sqrt{\|\sigma_{-1}(c)c\|_1} \leqslant \eta$.

SETTING THE STANDARD DEVIATION. By definition of the challenge space $\mathcal{C}$ and Lemma 3.2.8, if we know that $\|\mathbf{r}\| \leqslant \alpha$, then we can set the standard deviation $\mathfrak{s} := \gamma\eta\alpha$ where $\gamma > 0$ defines the repetition rate M.

4

ABDLOP COMMITMENT SCHEME

In this chapter, we describe a lattice-based hybrid commitment scheme which combines both constructions by Ajtai [Ajt96] and BDLOP [Bau+18b] described briefly in Chapter 2. Namely, we propose our general lattice-based commitment, which we call ABDLOP, in Section 4.1. Further, we describe an argument of knowledge of the ABDLOP commitment opening in Section 4.2. Finally, we show in Section 4.3 how to make use of the compression techniques from Dilithium-G [Duc+17] in order to further reduce the commitment as well as communication size. We remark that although none of the techniques in this chapter are explicitly new, we propose a way to conduct proofs in a generic way, i.e. without differentiating whether we are working with Ajtai or BDLOP commitments.

In the following, denote κ_{MSIS} and κ_{MLWE} to be the module ranks required for MSIS and MLWE security, respectively.

4.1 COMMITMENT CONSTRUCTION

Suppose we want to commit to a vector $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$ where $\mathbf{s}_1$ has a small L_2 norm, i.e. $\|\mathbf{s}_1\| \leqslant \alpha$, but not necessarily $\mathbf{m}$. The intuition here is to commit to $\mathbf{s}_1$ using the Ajtai commitment and $\mathbf{m}$ using the BDLOP commitment. If we were to construct each commitment separately, we would end up with two randomness vectors. We describe a way to generate the two commitments using one randomness vector.

We present our construction of the lattice-based commitment scheme $\mathsf{ABDLOP} = (\mathsf{ABDLOP.KeyGen}, \mathsf{ABDLOP.Commit}, \mathsf{ABDLOP.Open})$ in Figure 4.1. In the key generation, uniformly random matrices

$$\mathbf{A}_1 \leftarrow \mathcal{R}_q^{\kappa_{\mathsf{MSIS}} \times m_1}, \quad \mathbf{A}_2 \leftarrow \mathcal{R}_q^{\kappa_{\mathsf{MSIS}} \times m_2}, \quad \mathbf{B} \leftarrow \mathcal{R}_q^{\ell \times m_2}$$

are generated and output as public parameters. To commit to the message $(\mathbf{s}_1, \mathbf{m})$, we first sample $\mathbf{s}_2 \leftarrow \chi^{m_2}$ and output the commitment $\mathbf{t} = \mathbf{t}_A \parallel \mathbf{t}_B$ defined as:

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{s}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} \mathbf{s}_2 + \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix} \in \mathcal{R}_q^{\kappa_{\mathsf{MSIS}}+\ell}.$$

- $\mathsf{ABDLOP.KeyGen}(1^\kappa)$: select parameters $q, d, \kappa_{\mathsf{MSIS}}, m_1, m_2, \ell, \ell_{\mathsf{ext}}, \nu, \omega, \alpha, B_1, B_2$. Let χ be the uniform distribution on S_ν. Sample uniformly random matrices $\mathbf{A}_1 \leftarrow \mathcal{R}_q^{\kappa_{\mathsf{MSIS}} \times m_1}, \mathbf{A}_2 \leftarrow \mathcal{R}_q^{\kappa_{\mathsf{MSIS}} \times m_2}, \mathbf{B} \leftarrow \mathcal{R}_q^{\ell \times m_2}, \mathbf{B}_{\mathsf{ext}} \leftarrow \mathcal{R}_q^{\ell_{\mathsf{ext}} \times m_2}$.
  Define

$$\begin{aligned} pp.\mathsf{dim} &:= (q, d, \kappa_{\mathsf{MSIS}}, m_1, m_2, \ell, \ell_{\mathsf{ext}}) \\ pp.\mathsf{norms} &:= (\nu, \omega, \alpha, B_1, B_2) \\ pp.\mathsf{mat} &:= (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \mathbf{B}_{\mathsf{ext}}). \end{aligned}$$

  Return $pp = (pp.\mathsf{dim}, pp.\mathsf{norms}, pp.\mathsf{mat})$.

  The public parameters define the following message, randomness, commitment spaces along with randomness distribution and a set of relaxation factors:

$$\mathcal{S}_M = \left\{\mathbf{s}_1 \in \mathcal{R}_q^{m_1} : \|\mathbf{s}_1\| \leqslant \alpha\right\} \times \mathcal{R}_q^{\ell}, \quad \mathcal{S}_C = \bar{\mathcal{C}} \text{ as in Section 3.3.6,}$$

$$\mathcal{S}_T = \mathcal{R}_q^{\kappa_{\mathsf{MSIS}}+\ell}, \quad \mathcal{S}_R = \mathcal{R}_q^{m_2}, \quad \mathcal{D} = \chi^{m_2}.$$

- $\mathsf{ABDLOP.Commit}(\mathbf{s}_1, \mathbf{m}; \mathbf{s}_2)$: Given message $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{S}_M$ and randomness $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$, return $\mathbf{t} = (\mathbf{t}_A, \mathbf{t}_B)$ where

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} := \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{s}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} \mathbf{s}_2 + \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix} \in \mathcal{R}_q^{\kappa_{\mathsf{MSIS}}+\ell}.$$

- $\mathsf{ABDLOP.Open}(\mathbf{s}_1, \mathbf{m}, \mathbf{s}_2, c; \mathbf{t})$: Given $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{S}_M$, $\mathbf{s}_2 \in \mathcal{R}_q^{m_2}$, relaxation factor $c \in \mathcal{R}_q$ and a commitment $\mathbf{t} \in \mathcal{R}_q^{\kappa_{\mathsf{MSIS}}+\ell}$, output 1 if all the conditions below hold:
  1. $\mathsf{ABDLOP.Commit}(\mathbf{s}_1, \mathbf{m}; \mathbf{s}_2) = \mathbf{t}$
  2. $c \in \mathcal{S}_C$
  3. $\|c\mathbf{s}_2\| \leqslant B_2$
  4. $\|c\mathbf{s}_1\| \leqslant B_1$.

FIGURE 4.1: Description of a general lattice-based commitment scheme ABDLOP.

Intuitively, the *top part* $\mathbf{t}_A$ corresponds to binding the commitment as well as encoding the message $\mathbf{s}_1$ and the *bottom part* $\mathbf{t}_B$ encodes the message $\mathbf{m}$. We remark that when $\ell = 0$ (resp. $m_1 = 0$) then ABDLOP becomes the standard Ajtai (resp. BDLOP) commitment. Informally, we will call $\mathbf{s}_1$ (resp. $\mathbf{m}$) the message in the Ajtai (resp. BDLOP) part.

We explain the role of the matrix $\mathbf{B}_{\mathsf{ext}}$. Suppose that a party generates the ABDLOP commitment $(\mathbf{t}_A, \mathbf{t}_B)$ to the messages $(\mathbf{s}_1, \mathbf{m})$ under randomness $\mathbf{s}_2$. Then, if the party wants to commit to an additional message vector $\mathbf{m}_{\mathsf{ext}} \in \mathcal{R}_q^{\ell_{\mathsf{ext}}}$ under the same randomness at some later point in time, they can simply compute $\mathbf{t}_{\mathsf{ext}} := \mathbf{B}_{\mathsf{ext}}\mathbf{s}_2 + \mathbf{m}$ and thus $(\mathbf{t}_A, \mathbf{t}_B \parallel \mathbf{t}_{\mathsf{ext}})$ becomes the commitment to the messages $(\mathbf{s}_1, \mathbf{m} \parallel \mathbf{m}_{\mathsf{ext}})$. This property is directly inherited from the BDLOP commitment scheme and will be frequently used in our protocols.

We now turn to proving security properties of ABDLOP.

Lemma 4.1.1 (Correctness)**.** *If $B_1 \geqslant \alpha$ and $B_2 \geqslant \nu\sqrt{m_2 d}$ then* ABDLOP *is correct.*

Proof. Let $e_{\mathsf{id}} = 1 \in \mathcal{S}_C$ and take any $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{S}_M$. Then, clearly $\|e_{\mathsf{id}}\mathbf{s}_1\| \leqslant \alpha \leqslant B_1$. Also, for $\mathbf{s}_2 \leftarrow \chi^{m_2}$ we have:

$$\|e_{\mathsf{id}}\mathbf{s}_2\| \leqslant \nu\sqrt{m_2 d} \leqslant B_2.$$

Hence, $\mathsf{ABDLOP.Open}\,(\mathbf{s}_1, \mathbf{m}, \mathbf{s}_2, e_{\mathsf{id}}; \mathsf{ABDLOP.Commit}(\mathbf{s}_1, \mathbf{m}; \mathbf{s}_2)) = 1$. □

Lemma 4.1.2 (Hiding)**.** *Suppose that $\kappa_{\mathsf{MLWE}} := m_2 - \kappa_{\mathsf{MSIS}} - \ell \geqslant 0$. Then,* ABDLOP *is computational hiding if $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}}, \kappa_{\mathsf{MSIS}}+\ell, \chi}$ is hard.*

Proof. The statement directly follows from the observation that $\begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} \mathbf{s}_2$ is indistinguishable from a uniformly random vector in $\mathcal{R}_q^{\kappa_{\mathsf{MSIS}}+\ell}$ under the $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}}, \kappa_{\mathsf{MSIS}}+\ell, \chi}$ assumption. □

Lemma 4.1.3 (Binding)**.** ABDLOP *is strongly computational binding under the $\mathsf{MSIS}_{\kappa_{\mathsf{MSIS}}, m_1+m_2, B_{\mathsf{MSIS}}}$ assumption where $B_{\mathsf{MSIS}} = 4\eta\sqrt{B_1^2 + B_2^2}$.*

Proof. Suppose that for two triples $(\mathbf{s}_2, \mathbf{s}_1, \mathbf{m}, c), (\mathbf{s}_2', \mathbf{s}_1', \mathbf{m}', c')$ and a commitment $\mathbf{t} = (\mathbf{t}_A, \parallel \mathbf{t}_B)$ we have

$$\mathsf{BDLOP.Open}(\mathbf{s}_1, \mathbf{m}, \mathbf{s}_2, c; \mathbf{t}) = \mathsf{BDLOP.Open}(\mathbf{s}_1', \mathbf{m}', \mathbf{s}_2', c'; \mathbf{t}) = 1.$$

This implies that

$$\mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2 = \mathbf{t}_A = \mathbf{A}_1\mathbf{s}_1' + \mathbf{A}_2\mathbf{s}_2' \implies \begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix}\begin{bmatrix}\mathbf{s}_1 - \mathbf{s}_1' \\ \mathbf{s}_2 - \mathbf{s}_2'\end{bmatrix} = \mathbf{0}.$$

By the triangle inequality we have

$$\left\| cc' \begin{bmatrix}\mathbf{s}_1 - \mathbf{s}_1' \\ \mathbf{s}_2 - \mathbf{s}_2'\end{bmatrix} \right\| \leqslant \left\| c' \begin{bmatrix}c\mathbf{s}_1 \\ c\mathbf{s}_2\end{bmatrix} \right\| + \left\| c \begin{bmatrix}c'\mathbf{s}_1' \\ c'\mathbf{s}_2'\end{bmatrix} \right\|.$$

Consider the first term on the right-hand side. By definition of the opening algorithm and of $\mathcal{S}_C = \bar{\mathcal{C}}$ we have that $c' = c_0' - c_1'$ where $c_0', c_1' \in \mathcal{C}$ are distinct. Next, by Lemma 3.2.8:

$$\left\| c' \begin{bmatrix}c\mathbf{s}_1 \\ c\mathbf{s}_2\end{bmatrix} \right\| \leqslant \left\| c_0' \begin{bmatrix}c\mathbf{s}_1 \\ c\mathbf{s}_2\end{bmatrix} \right\| + \left\| c_1' \begin{bmatrix}c\mathbf{s}_1 \\ c\mathbf{s}_2\end{bmatrix} \right\| \leqslant 2\eta\sqrt{B_1^2 + B_2^2}.$$

With the same argument for the second term, we deduce that

$$\left\| cc' \begin{bmatrix}\mathbf{s}_1 - \mathbf{s}_1' \\ \mathbf{s}_2 - \mathbf{s}_2'\end{bmatrix} \right\| \leqslant 4\eta\sqrt{B_1^2 + B_2^2}.$$

Hence, we have found a solution to the $\mathsf{MSIS}_{\kappa_{\mathsf{MSIS}}, m_1+m_2, B}$ problem for the matrix $[\mathbf{A}_1\ \mathbf{A}_2]$ where $B = 4\eta\sqrt{B_1^2 + B_2^2}$. Assuming that this problem is hard, we get $\mathbf{s}_1 = \mathbf{s}_1'$ and $\mathbf{s}_2 = \mathbf{s}_2'$. Then, by construction we have $\mathbf{m} = \mathbf{t}_B - \mathbf{B}\mathbf{s}_2 = \mathbf{t}_B - \mathbf{B}\mathbf{s}_2' = \mathbf{m}'$. □

4.2 OPENING PROOF FOR THE ABDLOP COMMITMENT

The key component of proving various properties on a committed message is a proof of knowledge of the commitment opening. Using terminology from Section 3.3.3, we propose a commit-and-prove system $\Pi_{\mathsf{open}} = (\mathsf{ABDLOP}, \mathcal{P}, \mathcal{V})$ for the relation R_{yes} which always outputs 1. In this case, having a statement for R_{yes} is irrelevant, and hence we ignore it.

In the following, we fix the challenge space $\mathcal{C}$ to be as in Section 3.3.6 with respect to the identity automorphism σ_1.

We present the commit-and-prove $\Pi_{\mathsf{open}} = (\mathsf{ABDLOP}, \mathcal{P}, \mathcal{V})$ for relation R_{yes} in Figure 4.2. Prover $\mathcal{P}$ starts by sampling two masking vectors $\mathbf{y}_1 \leftarrow$

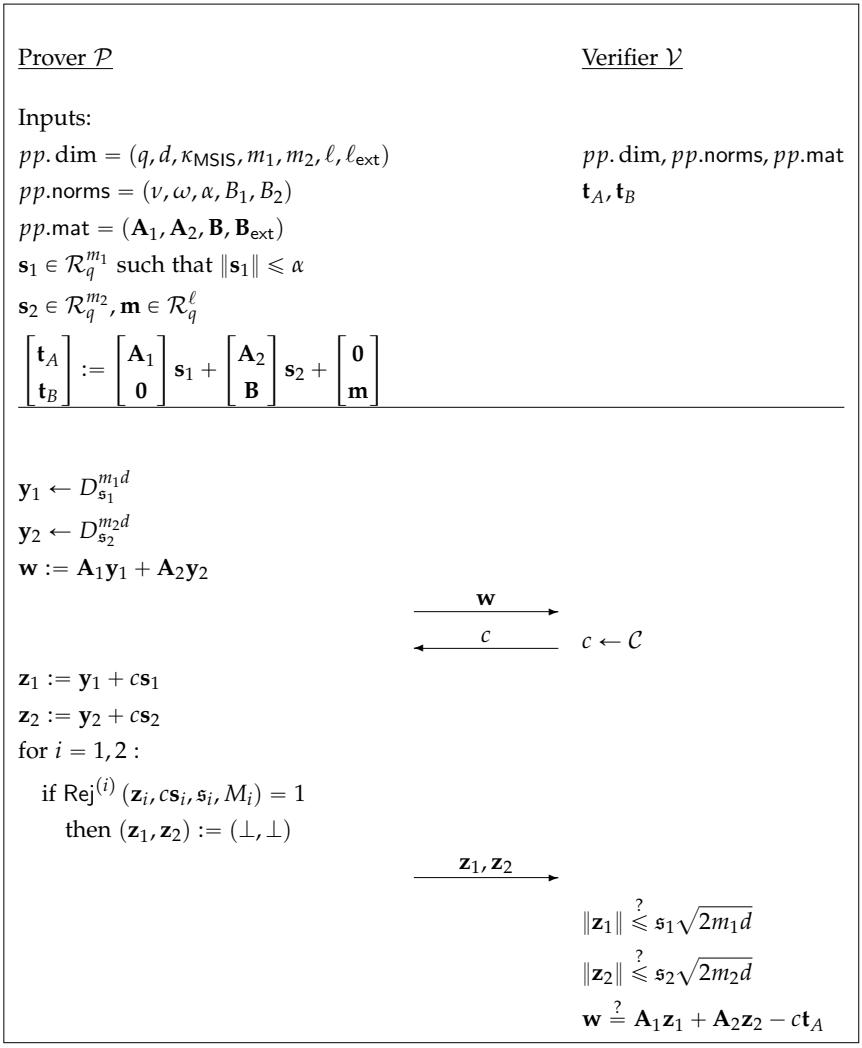

FIGURE 4.2: Commit-and-prove system Π_{open} for the relation R_{yes} which always outputs 1. Here, $\mathsf{Rej}^{(1)}, \mathsf{Rej}^{(2)}$ are rejection sampling algorithms.

$D_{\mathfrak{s}_1}^{m_1 d}, \mathbf{y}_2 \leftarrow D_{\mathfrak{s}_2}^{m_2 d}$ from discrete Gaussians and computes

$$\mathbf{w} := \mathbf{A}_1 \mathbf{y}_1 + \mathbf{A}_2 \mathbf{y}_2.$$

Then, it sends $\mathbf{w}$ to the verifier $\mathcal{V}$. After receiving the challenge $c \leftarrow \mathcal{C}$ from $\mathcal{V}$, the prover computes

$$\mathbf{z}_1 := \mathbf{y}_1 + c\mathbf{s}_1 \text{ and } \mathbf{z}_2 := \mathbf{y}_2 + c\mathbf{s}_2$$

and applies rejection sampling. If it does not abort, $\mathcal{P}$ sends $\mathbf{z}_1, \mathbf{z}_2$. Finally, the verifier checks that coefficients of each $\mathbf{z}_i$ are small and

$$\mathbf{w} \stackrel{?}{=} \mathbf{A}_1 \mathbf{z}_1 + \mathbf{A}_2 \mathbf{z}_2 - c\mathbf{t}_A.$$

4.2.1 *Security Analysis*

We summarise security properties of the protocol in Figure 4.2 below.

Theorem 4.2.1. *Suppose that $m_1 d \geqslant 5\kappa$ and $m_2 d \geqslant 5\kappa$ and let $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ as in Figure 3.2. Fix standard deviations $\mathfrak{s}_1 = \gamma_1 \eta \alpha$ and $\mathfrak{s}_2 = \gamma_1 \eta \nu \sqrt{m_2 d}$ for $\gamma_1, \gamma_2 > 0$ and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) \text{ for } i = 1, 2.$$

Then, Π_{open} for the relation R_{yes} has statistical completeness with correctness error $1 - \frac{1}{M_1 M_2}$.

Proof. We first compute the probability that an honest prover $\mathcal{P}$ does not abort. Note that $\mathfrak{s}_1, \mathfrak{s}_2$ are chosen such that $\mathfrak{s}_1 \geqslant \gamma_1 \|c\mathbf{s}_1\|$ and $\mathfrak{s}_2 \geqslant \gamma_2 \|c\mathbf{s}_2\|$ for any $c \in \mathcal{C}$. Then, by Lemma 3.3.2, $\mathsf{Rej}^{(i)}$ does not abort with probability

$$\frac{1 - 2^{-\kappa}}{M_i}.$$

Hence, the probability that none of the rejection algorithms abort is at least:

$$\frac{(1 - 2^{-\kappa})^2}{M_1 M_2} \geqslant \frac{(1 - 2^{-\kappa+1})}{M_1 M_2} \geqslant \frac{1}{M_1 M_2} - \mathsf{negl}(\kappa).$$

Finally, we turn to checking the verification equations when interacting with an honest prover. By Lemma 3.2.2 and the assumption that

$\min\{m_1 d, m_2 d\} \geqslant 5\kappa$, the first two verification inequalities hold with probability at least $1 - 2^{\kappa-1}$. Then, the last one is true because:

$$\begin{aligned}\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 - c\mathbf{t}_A &= \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2 + c(\mathbf{A}_1\mathbf{r}_1 + \mathbf{A}_2\mathbf{r}_2) - c\mathbf{t}_A \\ &= \mathbf{w} + c\mathbf{t}_A - c\mathbf{t}_A \\ &= \mathbf{w}.\end{aligned}$$

□

Theorem 4.2.2. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *as in Figure 3.2 and suppose* $\kappa_{\mathsf{MLWE}} := m_2 - \kappa_{\mathsf{MSIS}} - \ell$. *Fix standard deviations* $\mathfrak{s}_1 = \gamma_1\eta\alpha$ *and* $\mathfrak{s}_2 = \gamma_1\eta\nu\sqrt{m_2 d}$ *for* $\gamma_1, \gamma_2 > 0$ *and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) \text{ for } i = 1, 2.$$

Then, under the $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}}, \kappa_{\mathsf{MSIS}}+\ell}$ *assumption,* Π_{open} *for the relation* R_{yes} *is simulatable.*

Proof. We describe an efficient simulator $\mathcal{S}$ as follows. First, it samples $\mathbf{z}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1 d}$ and $\mathbf{z}_2 \leftarrow D_{\mathfrak{s}_2}^{m_2 d}$. Finally, $\mathcal{S}$ computes $\mathbf{w} := \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 - c\mathbf{t}_A$ and outputs a simulated transcript $(\mathbf{w}, c, \mathbf{z}_1, \mathbf{z}_2)$ Then, by Lemma 3.3.2, the simulated transcript is statistically close to a real non-aborted one.

Finally, we simulate the commitment by sampling $(\mathbf{t}_A, \mathbf{t}_B) \leftarrow \mathcal{R}_q^{\kappa_{\mathsf{MSIS}}+\ell}$. Then, it is computationally indistinguishable from the actual commitment by the $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}}, \kappa_{\mathsf{MSIS}}+\ell}$ assumption. □

Theorem 4.2.3. *Suppose* $B_1 \geqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ *and* $B_2 \geqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$. *Then,* Π_{open} *for the relation* R_{yes} *is knowledge sound with knowledge error* $1/|\mathcal{C}|$.

Proof. Let $\mathcal{P}^*$ be a probabilistic prover which runs in time at most T and convinces the verifier with probability $\epsilon > |\mathcal{C}|^{-1}$. By Lemma 3.3.1, there is an algorithm $\mathcal{E}$ which runs in expected time at most $2T$ and extracts two accepting transcripts with the same first message $\mathbf{w}$ with probability at least $\epsilon - 1/|\mathcal{C}|$:

$$\mathsf{tr}_i = \left(\mathbf{w}, c^{(i)}, \mathbf{z}_1^{(i)}, \mathbf{z}_2^{(i)}\right) \text{ for } i = 0, 1.$$

Let us define $\bar{c} := c^{(1)} - c^{(0)} \in \bar{\mathcal{C}}$. Note that by definition of the challenge space, $\bar{c}$ is invertible over $\mathcal{R}_q$ and $\|\bar{c}\|_\infty \leqslant 2\omega$. Next, we set

$$\bar{\mathbf{s}}_i := \frac{\mathbf{z}_i^{(1)} - \mathbf{z}_i^{(0)}}{\bar{c}} \text{ for } i = 1, 2 \quad \text{and} \quad \bar{\mathbf{m}} := \mathbf{t}_B - \mathbf{B}\bar{\mathbf{s}}_2.$$

Then, by construction we get

$$\mathbf{A}_1\bar{\mathbf{s}}_1 + \mathbf{A}\bar{\mathbf{s}}_2 = \frac{\left(\mathbf{A}_1\mathbf{z}_1^{(1)} + \mathbf{A}_2\mathbf{z}_2^{(1)}\right) - \left(\mathbf{A}_1\mathbf{z}_1^{(0)} + \mathbf{A}_2\mathbf{z}_2^{(0)}\right)}{\bar{c}} = \frac{\bar{c}\mathbf{t}_A}{\bar{c}} = \mathbf{t}_A$$

and thus $\mathsf{ABDLOP.Commit}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}; \bar{\mathbf{s}}_2) = \mathbf{t}$. Moreover,

$$\|\bar{c}\bar{\mathbf{s}}_1\| = \|\mathbf{z}_1^{(1)} - \mathbf{z}_1^{(0)}\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1 d} \leqslant B_1$$

and similarly

$$\|\bar{c}\bar{\mathbf{s}}_2\| = \|\mathbf{z}_2^{(1)} - \mathbf{z}_2^{(0)}\| \leqslant 2\mathfrak{s}_2\sqrt{2m_2 d} \leqslant B_2.$$

Thus, $\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}) = 1$. □

4.3 IMPROVED OPENING PROOF WITH COMMITMENT COMPRESSION

In this section, we reduce the commitment and communication size by applying compression techniques from Dilithium-G [Duc+17].

4.3.1 *Low/High Order Bits*

In order to reduce the size of the commitment, we need some algorithms that extract "higher-order" and "lower-order" bits of elements in $\mathbb{Z}_q$. The goal is that when given an arbitrary element $r \in \mathbb{Z}_q$ and another small element $z \in \mathbb{Z}_q$, we would like to be able to recover the higher order bits of $r + z$ without needing to store z. The algorithms are exactly as in [Duc+17], and we repeat them for completeness in Figure 4.3. They are described as working on integers modulo q, but one can extend it to (vectors of) polynomials in $\mathcal{R}_q$ by simply being applied individually to each coefficient.

Lemma 4.3.1. *Suppose that q and γ are positive integers satisfying $q \equiv 1 \pmod{\gamma}$. Fix $m := (q-1)/\gamma$. Let $\mathbf{r}$ and $\mathbf{z}$ be vectors of elements in R_q where $\|\mathbf{z}\|_\infty \leqslant \gamma/2$, and let $\mathbf{y}, \mathbf{y}'$ be integral vectors of elements in $(-m/2, m/2]$. Then the $\mathsf{HighBits}_q$, $\mathsf{MakeGHint}_q$, and $\mathsf{UseGHint}_q$ algorithms satisfy the following properties:*

1. $\mathsf{UseGHint}_q(\mathsf{MakeGHint}_q(\mathbf{z}, \mathbf{r}, \gamma), \mathbf{r}, \gamma) = \mathsf{HighBits}_q(\mathbf{r} + \mathbf{z}, \gamma)$.
2. *If* $\mathsf{UseGHint}_q(\mathbf{y}, \mathbf{r}, \gamma) = \mathsf{UseGHint}_q(\mathbf{y}', \mathbf{r}, \gamma)$, *then* $\mathbf{y} = \mathbf{y}'$.

$\underline{\mathsf{Power2Round}_q(r, D)}$
00 $r := r \bmod^+ q$
01 $r_0 := r \bmod^\pm 2^D$
02 **return** $(r - r_0)/2^D$

$\underline{\mathsf{UseGHint}_q(y, r, \gamma)}$
03 $m := (q-1)/\gamma$
04 $r_1 := \mathsf{HighBits}_q(r, \gamma)$
05 **return** $(r_1 + y) \bmod^\pm m$

$\underline{\mathsf{MakeGHint}_q(z, r, \gamma)}$
06 $m = (q-1)/\gamma$
07 $r_1 := \mathsf{HighBits}_q(r, \gamma)$
08 $v_1 := \mathsf{HighBits}_q(r + z, \gamma)$
09 **return** $(v_1 - r_1) \bmod^\pm m$

$\underline{\mathsf{Decompose}_q(r, \gamma)}$
10 $r := r \bmod^+ q$
11 $r_0 := r \bmod^\pm \gamma$
12 **if** $r - r_0 = q - 1$
13 **then** $r_1 := 0$; $r_0 := r_0 - 1$
14 **else** $r_1 := (r - r_0)/\gamma$
15 **return** (r_1, r_0)

$\underline{\mathsf{HighBits}_q(r, \gamma)}$
16 $(r_1, r_0) := \mathsf{Decompose}_q(r, \gamma)$
17 **return** r_1

$\underline{\mathsf{LowBits}_q(r, \gamma)}$
18 $(r_1, r_0) := \mathsf{Decompose}_q(r, \gamma)$
19 **return** r_0

FIGURE 4.3: Supporting algorithms for commitment compression.

4.3.2 *ABDLOP Commitment Compression*

We apply the aforementioned compression techniques in the opening proof presented above. First, we reduce the size of the ABDLOP commitment by not sending the low-order bits of $\mathbf{t}_A$. Namely, for a suitable $D \in \mathbb{N}$ we write

$$\mathbf{t}_A = \mathbf{t}_{A,1} \cdot 2^D + \mathbf{t}_{A,2} \text{ where } \|\mathbf{t}_{A,2}\|_\infty \leqslant 2^{D-1}$$

and only send $\mathbf{t}_{A,1}$. Thus, we reduce the commitment size by $D\kappa_{\mathsf{MSIS}}d$ bits.

Further, instead of sampling uniformly random matrices $\mathbf{A}_2$ and $\mathbf{B}$, we can choose them in the more structured way as originally in [Bau+18b]

$$\begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} := \begin{bmatrix} \mathbf{A}'_2 & \mathbf{I}_{\kappa_{\mathsf{MSIS}}} \\ \mathbf{B}' & \mathbf{0}_{\ell \times \kappa_{\mathsf{MSIS}}} \end{bmatrix} \mathcal{R}_q^{\kappa_{\mathsf{MSIS}} \times m_2}. \tag{4.1}$$

We call this version of the commitment $\mathsf{ABDLOP}_{\mathsf{compress}}$. We show the commit-and-prove system $\Pi_{\mathsf{open-compress}} = (\mathsf{ABDLOP}_{\mathsf{compress}}, \mathcal{P}, \mathcal{V})$ for the relation R_{yes} in Figure 4.4. We recall that R_{yes} is a relation which always outputs 1.

Prover $\mathcal{P}$ starts by sampling vectors $\mathbf{y}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1 d}, \mathbf{y}_{2,1} \leftarrow D_{\mathfrak{s}_2}^{(m_2 - \kappa_{\mathsf{MSIS}})d}$ and $\mathbf{y}_{2,2} \leftarrow D_{\mathfrak{s}_2}^{\kappa_{\mathsf{MSIS}} d}$ from discrete Gaussians and computing $\mathbf{w} = \mathbf{A}_1 \mathbf{y}_1 +$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
Inputs:		
$pp.\dim = (q, d, \kappa_{\mathsf{MSIS}}, m_1, m_2, \ell, \ell_{\mathsf{ext}})$		$pp.\dim$, $pp.\mathsf{norms}$, $pp.\mathsf{mat}$
$pp.\mathsf{norms} = (\nu, \omega, \alpha, B_1, B_2)$		$\mathbf{t}_{A,1}, \mathbf{t}_B$
$pp.\mathsf{mat} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \mathbf{B}_{\mathsf{ext}})$		
$\mathbf{s}_1 \in \mathcal{R}_q^{m_1}$ such that $\|\mathbf{s}_1\| \leqslant \alpha$		
$\mathbf{s}_2 = (\mathbf{s}_{2,1}, \mathbf{s}_{2,2}) \in \mathcal{R}_q^{m_2}, \mathbf{m} \in \mathcal{R}_q^{\ell}$		
$\mathbf{A}_2', \mathbf{B}'$ are defined as in (4.1)		
$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} := \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{s}_1 + \begin{bmatrix} \mathbf{A}_2' \\ \mathbf{B}' \end{bmatrix} \mathbf{s}_{2,1} + \begin{bmatrix} \mathbf{s}_{2,2} \\ \mathbf{m} \end{bmatrix}$		
$\mathbf{t}_{A,1} := \mathsf{Power2Round}_q(\mathbf{t}_A, D)$		
$\mathbf{t}_{A,2} := \mathbf{t}_A - 2^D \mathbf{t}_{A,1}$		
$\mathbf{y}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1 d}$		
$\mathbf{y}_{2,1} \leftarrow D_{\mathfrak{s}_2}^{(m_2 - \kappa_{\mathsf{MSIS}})d}$		
$\mathbf{y}_{2,2} \leftarrow D_{\mathfrak{s}_2}^{\kappa_{\mathsf{MSIS}} d}$		
$\mathbf{w} := \mathbf{A}_1 \mathbf{y}_1 + \mathbf{A}_2' \mathbf{y}_{2,1} + \mathbf{y}_{2,2}$		
$(\mathbf{w}_1, \mathbf{w}_0) := \mathsf{Decompose}_q(\mathbf{w}, \gamma)$		
	$\xrightarrow{\mathbf{w}_1}$	
	$\xleftarrow{c}$	$c \leftarrow \mathcal{C}$
$\mathbf{z}_1 := \mathbf{y}_1 + c\mathbf{s}_1$		
$\mathbf{z}_2 = \begin{bmatrix} \mathbf{z}_{2,1} \\ \mathbf{z}_{2,2} \end{bmatrix} := \begin{bmatrix} \mathbf{y}_{2,1} \\ \mathbf{y}_{2,2} \end{bmatrix} + c \begin{bmatrix} \mathbf{s}_{2,1} \\ \mathbf{s}_{2,2} \end{bmatrix}$		
for $i = 1, 2$:		
if $\mathsf{Rej}_i(\mathbf{z}_i, c\mathbf{s}_i, \mathfrak{s}_i) = 1$		
then $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{h}) := (\perp, \perp, \perp)$		
$\mathbf{z}_{2,2} := \mathbf{z}_{2,2} - c\mathbf{t}_{A,2} - \mathbf{w}_0$		
if $\|(\mathbf{z}_{2,1}, \mathbf{z}_{2,2})\| \leqslant B$		
then $(\mathbf{z}_1, \mathbf{z}_2, \mathbf{h}) := (\perp, \perp, \perp)$		
$\mathbf{h} := \mathsf{MakeGHint}_q(\mathbf{z}_{2,2}, \gamma\mathbf{w}_1 - \mathbf{z}_{2,2}, \gamma)$		
	$\xrightarrow{\mathbf{z}_1, \mathbf{z}_{2,1}, \mathbf{h}}$	
		$\mathbf{w}_1 \overset{?}{=} \mathsf{UseGHint}_q\left(\mathbf{h}, \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2'\mathbf{z}_{2,1} - c \cdot 2^D \mathbf{t}_{A,1}, \gamma\right)$
		$\|\mathbf{z}_1\| \overset{?}{\leqslant} \mathfrak{s}_1 \sqrt{2m_1 d}$
		$\|(\mathbf{z}_{2,1}, \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2'\mathbf{z}_{2,1} - c \cdot 2^D \mathbf{t}_{A,1} - \gamma\mathbf{w}_1)\| \overset{?}{\leqslant} B$
		$\|\mathbf{h}\|_\infty \overset{?}{\leqslant} \frac{q-1}{2\gamma}$

FIGURE 4.4: Commit-and-prove system $\Pi_{\mathsf{open-compress}}$ for the relation R_{yes} using compression techniques from Dilithium-G [Duc+17].

$\mathbf{A}_2'\mathbf{y}_{2,1} + \mathbf{y}_{2,2}$. Additionally, $\mathcal{P}$ calculates $(\mathbf{w}_1, \mathbf{w}_0) = \mathsf{Decompose}_q(\mathbf{w}, \gamma)$ and sends $\mathbf{w}_1$ to the verifier where $q-1$ is divisible by γ.

After receiving a challenge polynomial $c \leftarrow \mathcal{C}$ from $\mathcal{V}$, the prover computes

$$\mathbf{z}_1 = \mathbf{y}_1 + c\mathbf{s}_1 \text{ and } \mathbf{z}_2 = \begin{bmatrix} \mathbf{z}_{2,1} \\ \mathbf{z}_{2,2} \end{bmatrix} := \begin{bmatrix} \mathbf{y}_{2,1} \\ \mathbf{y}_{2,2} \end{bmatrix} + c \begin{bmatrix} \mathbf{s}_{2,1} \\ \mathbf{s}_{2,2} \end{bmatrix}$$

and applies rejection sampling for $\mathbf{z}_1$ and $\mathbf{z}_2$. If it accepts, $\mathcal{P}$ modifies $\mathbf{z}_{2,2} := \mathbf{z}_{2,2} - c\mathbf{t}_{A,2} - \mathbf{w}_0$ and calculates the hint vector

$$\mathbf{h} = \mathsf{MakeGHint}_q\left(\mathbf{z}_{2,2}, \gamma\mathbf{w}_1 - \mathbf{z}_{2,2}, \gamma\right).$$

Finally, the prover sends $(\mathbf{z}_1, \mathbf{z}_{2,1}, \mathbf{h})$. In the last stage, verifier $\mathcal{V}$ checks whether vectors $\mathbf{z}_1$ and $(\mathbf{z}_{2,1}, \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2'\mathbf{z}_{2,1} - c \cdot 2^D\mathbf{t}_{A,1} - \gamma\mathbf{w}_1)$ have small norms and the coefficients of $\mathbf{h}$ are between $-\frac{q-1}{2\gamma}$ and $\frac{q-1}{2\gamma}$ and

$$\mathbf{w}_1 \stackrel{?}{=} \mathsf{UseHint}_q\left(\mathbf{h}, \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2'\mathbf{z}_{2,1} - c \cdot 2^D\mathbf{t}_{A,1}, \gamma\right).$$

As opposed to the standard opening proof, the prover does not send any masked opening of $\mathbf{s}_{2,2}$. Instead, $\mathcal{P}$ sends a vector of hints $\mathbf{h}$ which has much smaller impact on the communication size as opposed to $\mathbf{z}_{2,2}$.

SECURITY ANALYSIS. We first focus on the completeness of the protocol.

Theorem 4.3.2. *Suppose $m_1 d \geqslant 5\kappa, m_2 d \geqslant 5\kappa$ and γ be an even divisor of $q-1$. Fix $\mathfrak{s}_1 = \gamma_1\eta\alpha$ and $\mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d}$ for some $\gamma_1, \gamma_2 > 0$. Then, denote*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) \text{ for } i = 1, 2.$$

and $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ as in Figure 3.2. Also, set

$$B := \mathfrak{s}_2\sqrt{2m_2 d} + \eta 2^{D-1}\sqrt{nd} + \frac{\gamma\sqrt{nd}}{2}.$$

Then, the commit-and-prove system $\Pi_{\mathsf{open-compress}}$ for the relation R_{yes} satisfies statistical completeness with statistical error $1 - 1/(M_1 M_2)$.

Proof. First, if the rejection sampling steps pass, the distributions of $\mathbf{z}_1, \mathbf{z}_2$ are discrete Gaussians centered at 0 with standard deviations $\mathfrak{s}_1$ and $\mathfrak{s}_2$ respectively. Since $m_1 d, m_2 d \geqslant 5\kappa$, we have that

$$\Pr_{\mathbf{z}_i \leftarrow D_{\mathfrak{s}_i}^{m_i}}\left[\|\mathbf{z}_i\| \leqslant \mathfrak{s}_i\sqrt{2m_1 d}\right] \geqslant 1 - 2^{-\kappa} \quad \text{for } i = 1, 2$$

by Lemma 3.2.2 for $t = \sqrt{2}$. Now, since we perturb the vector $\mathbf{z}_{2,2}$, the bound on $\|\mathbf{z}_2\|$ increases slightly. Using the inequalities $\|c\mathbf{t}_{A,2}\| \leqslant \eta\|\mathbf{t}_{A,2}\| = \eta 2^{D-1}\sqrt{\kappa_{\mathsf{MSIS}}d}$ and $\|\mathbf{w}_0\| \leqslant \gamma\sqrt{\kappa_{\mathsf{MSIS}}d}/2$, we get

$$\left\|\begin{bmatrix}\mathbf{z}_{2,1}\\ \mathbf{z}_{2,2} - c\mathbf{t}_{A,2} - \mathbf{w}_0\end{bmatrix}\right\| \leqslant \left\|\begin{bmatrix}\mathbf{z}_{2,1}\\ \mathbf{z}_{2,2}\end{bmatrix}\right\| + \left\|\begin{bmatrix}\mathbf{0}\\ c\mathbf{t}_{A,2}\end{bmatrix}\right\| + \left\|\begin{bmatrix}\mathbf{0}\\ \mathbf{w}_0\end{bmatrix}\right\|$$
$$\leqslant \mathfrak{s}_2\sqrt{2m_2d} + \eta 2^{D-1}\sqrt{nd} + \frac{\gamma\sqrt{nd}}{2} = B.$$

The verification equation on $\|\mathbf{h}\|_\infty$ follows by definition of MakeGHint. Finally, note that

$$\begin{aligned}\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2'\mathbf{z}_{2,1} + \mathbf{z}_{2,2} &= c2^D\mathbf{t}_{A,1} + \mathbf{w} - \mathbf{w}_0\\ &= c2^D\mathbf{t}_{A,1} + \gamma\mathbf{w}_1\end{aligned}$$

and thus

$$\mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2'\mathbf{z}_{2,1} - c2^D\mathbf{t}_{A,1} = \gamma\mathbf{w}_1 - \mathbf{z}_{2,2}.$$

Consequently, by Lemma 4.3.1:

$$\begin{aligned}&\mathsf{UseGHint}_q(\mathbf{h}, \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2'\mathbf{z}_{2,1} - c\cdot 2^D\mathbf{t}_{A,1}, \gamma)\\ &\quad= \mathsf{UseGHint}_q\left(\mathsf{MakeGHint}_q\left(\mathbf{z}_{2,2}, \gamma\mathbf{w}_1 - \mathbf{z}_{2,2}, \gamma\right), \gamma\mathbf{w}_1 - \mathbf{z}_{2,2}, \gamma\right)\\ &\quad= \mathsf{HighBits}_q(\gamma\mathbf{w}_1, \gamma)\\ &\quad= \mathbf{w}_1.\end{aligned}$$

□

Next, we focus on the simulatability property.

Theorem 4.3.3. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *as in Figure 3.2 and fix* $\mathfrak{s}_1 = \gamma_1\eta\alpha$ *and* $\mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2d}$ *for some* $\gamma_1, \gamma_2 > 0$. *Denote*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}}\cdot\frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) \text{ for } i = 1, 2$$

and $\kappa_{\mathsf{MLWE}} := m_2 - \kappa_{\mathsf{MSIS}} - \ell \geqslant 0$. *Then, under the* $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}}, \kappa_{\mathsf{MSIS}}+\ell}$ *assumption, the commit-and-prove system* $\Pi_{\mathsf{open-compress}}$ *for the relation* R_{yes} *is simulatable.*

Proof. As in the proof of Theorem 4.3.3, the algorithm $\mathcal{S}$ simulates the commitment by generating $(\mathbf{t}_A, \mathbf{t}_B) \leftarrow \mathcal{R}_q^{\kappa_{\mathsf{MSIS}}+\ell}$ and computing

$$\mathbf{t}_{A,1} := \mathsf{Power2Round}_q(\mathbf{t}_A, D)$$

under the $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}},\kappa_{\mathsf{MSIS}}+\ell}$ assumption. Furthermore, $\mathcal{S}$ samples $\mathbf{z}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1 d}, \mathbf{z}_2 \leftarrow D_{\mathfrak{s}_2}^{m_2 d}$ and computes

$$\mathbf{w} := \mathbf{A}_1 \mathbf{z}_1 + \mathbf{A}_2' \mathbf{z}_{2,1} + \mathbf{z}_{2,2} - c\mathbf{t}_A.$$

Then, $\mathcal{S}$ calculates $(\mathbf{w}_1, \mathbf{w}_0) := \mathsf{Decompose}_q(\mathbf{w}, \gamma)$. Finally, the hint vector $\mathbf{h}$ can be computed deterministically from all the previous components. □

Now, we turn to proving knowledge soundness.

Theorem 4.3.4. *Suppose $B_1 \geqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ and $B_2 \geqslant 2B$. Then, the commit-and-prove system $\Pi_{\mathsf{open-compress}}$ for the relation R_{yes} is knowledge sound with knowledge error $1/|\mathcal{C}|$.*

Proof. Let $\mathcal{P}^*$ be a probabilistic prover which convinces the verifier with probability $\varepsilon > |\mathcal{C}|^{-1}$. By Lemma 3.3.1, there is an algorithm $\mathcal{E}$ which extracts two accepting transcripts with the same first message $\mathbf{w}_1$ and distinct challenges with probability at least $\epsilon - 1/|\mathcal{C}|$:

$$\mathsf{tr}_i = \left(\mathbf{w}_1, c^{(i)}, \mathbf{z}_1^{(i)}, \mathbf{z}_{2,1}^{(i)}, \mathbf{h}^{(i)}\right) \text{ for } i = 0, 1.$$

Let us define $\bar{c} := c^{(1)} - c^{(0)} \in \bar{\mathcal{C}}$. Note that by definition of the challenge space, $\bar{c}$ is invertible over $\mathcal{R}_q$. Let us define

$$\mathbf{u}^{(i)} := \gamma \mathbf{w}_1 + c^{(i)} \cdot 2^D \mathbf{t}_{A,1} - \mathbf{A}_1 \mathbf{z}_1^{(i)} - \mathbf{A}_2' \mathbf{z}_{2,1}^{(i)}.$$

Thus, we have $\left\| \left(\mathbf{z}_{2,1}^{(i)}, \mathbf{u}^{(i)}\right) \right\| \leqslant B$ for $i = 0, 1$. Then, by combining the two equations on $\mathbf{u}^{(i)}$ we get

$$\mathbf{A}_1 \left(\mathbf{z}_1^{(1)} - \mathbf{z}_1^{(0)}\right) + \mathbf{A}_2' \left(\mathbf{z}_2^{(1)} - \mathbf{z}_2^{(0)}\right) + \left(\mathbf{u}^{(1)} - \mathbf{u}^{(0)}\right) = \bar{c} \cdot 2^D \mathbf{t}_{A,1}.$$

Next, we set

$$\bar{\mathbf{s}}_1 := \frac{\mathbf{z}_1^{(1)} - \mathbf{z}_1^{(0)}}{\bar{c}}, \quad \bar{\mathbf{s}}_2 = \begin{bmatrix} \bar{\mathbf{s}}_{2,1} \\ \bar{\mathbf{s}}_{2,2} \end{bmatrix} := \frac{1}{\bar{c}} \cdot \begin{bmatrix} \mathbf{z}_{2,1}^{(1)} - \mathbf{z}_{2,1}^{(0)} \\ \mathbf{u}^{(1)} - \mathbf{u}^{(0)} \end{bmatrix}$$

and

$$\bar{\mathbf{m}} := \mathbf{t}_B - \mathbf{B}\bar{\mathbf{s}}_2.$$

By construction we obtain $\|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1 d} \leqslant B_1$ and $\|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2B \leqslant B_2$. Hence,

$$\mathsf{ABDLOP.Open}\left(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; (2^D \mathbf{t}_{A,1}, \mathbf{t}_B)\right) = 1.$$

□

5

PROVING LINEAR AND HIGHER-DEGREE RELATIONS BETWEEN COMMITTED MESSAGES

This chapter focuses on proving arbitrary linear and higher-degree equations between committed polynomials by extending the opening proofs presented in Chapter 4. Namely, we first show how to prove knowledge of a message vector $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$ which satisfies

$$f(\mathbf{s}_1, \mathbf{m}) = 0$$

where f is a public $(m_1 + \ell)$-variate polynomial function over $\mathcal{R}_q$. We extend our argument to the case when one wants to prove multiple such relations in parallel, i.e. $f_1(\mathbf{s}_1, \mathbf{m}) = f_2(\mathbf{s}_1, \mathbf{m}) = \ldots = f_N(\mathbf{s}_1, \mathbf{m}) = 0$. Furthermore, we also cover statements where we do not necessarily have $f(\mathbf{m}) = 0$ but one of the coefficients of $f(\mathbf{m})$ is equal to zero. Without loss of generality, we will only consider the constant coefficient.

More precisely, denote $P_n^t(\mathcal{R}_q)$ to be the set of all polynomial functions $f : \mathcal{R}_q^n \to \mathcal{R}_q$ over $\mathcal{R}_q$ of total degree at most t. Then, we are interested in proving the following statements:

- *Single equation.* Given a public polynomial function $f \in P_{m_1+\ell}^t(\mathcal{R}_q)$, prove knowledge of the message vectors $\mathbf{s}_1 \in \mathcal{R}_q^{m_1}$ and $\mathbf{m} \in \mathcal{R}_q^{\ell}$, where $\|\mathbf{s}_1\| \leqslant \alpha$, which satisfy
$$f(\mathbf{s}_1, \mathbf{m}) = 0.$$

- *Many equations.* Given N public polynomial functions $f_i \in P_{m_1+\ell}^t(\mathcal{R}_q)$, prove knowledge of the message vectors $\mathbf{s}_1 \in \mathcal{R}_q^{m_1}$ and $\mathbf{m} \in \mathcal{R}_q^{\ell}$, where $\|\mathbf{s}_1\| \leqslant \alpha$, which satisfy
$$f_i(\mathbf{s}_1, \mathbf{m}) = 0 \text{ for } i = 1, 2, \ldots, N.$$

- *Function evaluations with vanishing constant coefficients.* Given $N + M$ public polynomial functions $f_1, \ldots, f_n, F_1, \ldots, F_M \in P_{m_1+\ell}^t(\mathcal{R}_q)$, prove knowledge of the message vectors $\mathbf{s}_1 \in \mathcal{R}_q^{m_1}$ and $\mathbf{m} \in \mathcal{R}_q^{\ell}$, where $\|\mathbf{s}_1\| \leqslant \alpha$, which satisfy the following:
  1. $f_i(\mathbf{s}_1, \mathbf{m}) = 0$ for $i = 1, 2, \ldots, N$.

2. Denote $x_j := F_j(\mathbf{s}_1, \mathbf{m})$. Then, $\widetilde{x}_1 = \ldots = \widetilde{x}_M = 0$.

We provide our protocols in a commit-and-prove fashion, i.e. we first generate an ABDLOP commitment $\mathbf{t} = (\mathbf{t}_A, \mathbf{t}_B)$ to $(\mathbf{s}_1, \mathbf{m})$ and then prove that the messages satisfy certain relations.

We start by proving linear equations in Section 5.1 by simply extending the argument by Baum et al. [Bau+18b]. Next, we adapt the product proof protocol by Attema et al. [ALS20] to prove general quadratic relations in Section 5.2. In our applications, degree two equations are sufficient, but it will be clear how to generalise the techniques for proving higher-degree relations.

5.1 PROOF OF LINEAR RELATIONS

For convenience, throughout this section we denote $\mathbf{s} := \mathbf{s}_1 \parallel \mathbf{m} \in \mathcal{R}_q^{m_1+\ell}$. Moreover, we represent a linear function $f \in P^1_{m_1+\ell}(\mathcal{R}_q)$ as

$$f(\mathbf{x}) := \mathbf{r}_1^T \mathbf{x} + r_0.$$

where $\mathbf{r}_1 \in \mathcal{R}_q^{m_1+\ell}$ and $r_0 \in \mathcal{R}_q$.

In this section, the challenge space $\mathcal{C}$ is defined as in Section 3.3.6 with the identity automorphism σ_1.

5.1.1 *Single Equation*

Let f be a $(m_1 + \ell)$-variate linear function over $\mathcal{R}_q$. In this subsection, we will be interested in the following relation:

$$R := \{(f, (\mathbf{s}_1, \mathbf{m})) : f(\mathbf{s}_1, \mathbf{m}) = 0\}$$

and the corresponding commit-and-prove relation R_{ABDLOP} as defined in Section 3.3.3.

Let us first consider the opening proof in Figure 4.2. We observe that vector $\mathbf{z}_1 := \mathbf{y} + c\mathbf{s}_1$ "masks" our first message vector $\mathbf{s}_1$. We will informally call $\mathbf{z}_1$ to be a *masked opening* of $\mathbf{s}_1$. Even though this is not the case for $\mathbf{m}$ (i.e. we do not send anything of the form $\mathbf{z}_m = \mathbf{y}_m + c\mathbf{m}$), we observe that

$$c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2 = -\mathbf{B}\mathbf{y}_2 + c\mathbf{m}$$

is a masked opening of $\mathbf{m}$ which can be computed by the verifier. Hence, if we define

$$\mathbf{y} := \begin{bmatrix} \mathbf{y}_1 \\ -\mathbf{B}\mathbf{y}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{z} := \begin{bmatrix} \mathbf{z}_1 \\ c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2 \end{bmatrix}$$

then we have $\mathbf{z} = \mathbf{y} + c\mathbf{s}$. Now, we observe that

$$\mathbf{r}_1^T\mathbf{z} + cr_0 = \mathbf{r}_1^T\mathbf{y} + c\mathbf{r}_1^T\mathbf{s} + cr_0 = \mathbf{r}_1^T\mathbf{y} + c(\mathbf{r}_1^T\mathbf{s} + r_0) = \mathbf{r}_1^T\mathbf{y} + cf(\mathbf{s}) = \mathbf{r}_1^T\mathbf{y}.$$

Hence, in the protocol in Figure 4.2, if we additionally let the prover send $v := \mathbf{r}_1^T\mathbf{y}$ to the verifier $\mathcal{V}$ in the first round, then $\mathcal{V}$ simply has one more verification check

$$\mathbf{r}_1^T\mathbf{z} + cr_0 \stackrel{?}{=} v.$$

As we formally show later, this is sufficient to prove that $f(\mathbf{s}) = 0$.

5.1.2 *Multiple Equations*

Firstly, we observe that proving N linear equations

$$f_i(\mathbf{s}) = 0 \quad \text{for} \quad f_1, \ldots, f_N \in P^1_{m_1+\ell}(\mathcal{R}_q)$$

boils down to proving

$$\mathbf{R}_1\mathbf{s} + \mathbf{r}_0 = \mathbf{0} \tag{5.1}$$

where $\mathbf{R}_1 \in \mathcal{R}_q^{N\times(m_1+\ell)}$ and $\mathbf{r}_0 \in \mathcal{R}_q^N$. Hence, we define the corresponding relation

$$R_{\mathsf{lin}} := \{((\mathbf{R}_1, \mathbf{r}_0), \mathbf{s} := (\mathbf{s}_1, \mathbf{m})) : \mathbf{R}_1\mathbf{s} + \mathbf{r}_0 = \mathbf{0}\}.$$

We extend the approach from the previous subsection naturally. Namely, if (5.1) holds then we have:

$$\mathbf{R}_1\mathbf{z} + c\mathbf{r}_0 = \mathbf{R}_1\mathbf{y} + c\mathbf{R}_1\mathbf{s} + c\mathbf{r}_0 = \mathbf{R}_1\mathbf{y} + c(\mathbf{R}_1\mathbf{s} + \mathbf{r}_0) = \mathbf{R}_1\mathbf{y}.$$

Thus, the prover in the first round of the protocol in Figure 4.2 additionally sends $\mathbf{v} := \mathbf{R}_1\mathbf{y}$. Then, in the end the verifier $\mathcal{V}$ has one more verification check $\mathbf{R}_1\mathbf{z} + c\mathbf{r}_0 \stackrel{?}{=} \mathbf{v}$.

We provide a commit-and-prove system $\Pi_{\mathsf{lin}} := (\mathsf{ABDLOP}, \mathcal{P}, \mathcal{V})$ for the relation R_{lin} in Figure 5.1.

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
Inputs:		
$pp.\dim = (q, d, \kappa_{\mathsf{MSIS}}, m_1, m_2, \ell, \ell_{\mathsf{ext}} := 0)$		$pp.\dim, pp.\mathsf{norms}, pp.\mathsf{mat}$
$pp.\mathsf{norms} = (\nu, \omega, \alpha, B_1, B_2)$		$\mathbf{t}_A, \mathbf{t}_B, \mathbf{R}_1, \mathbf{r}_0$
$pp.\mathsf{mat} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \mathbf{B}_{\mathsf{ext}})$		
$\mathbf{s}_1 \in \mathcal{R}_q^{m_1}, \mathbf{s}_2 \in \mathcal{R}_q^{m_2}, \mathbf{m} \in \mathcal{R}_q^{\ell}$ so that $\|\mathbf{s}_1\| \leqslant \alpha$		
$\begin{bmatrix}\mathbf{t}_A \\ \mathbf{t}_B\end{bmatrix} := \begin{bmatrix}\mathbf{A}_1 \\ \mathbf{0}\end{bmatrix}\mathbf{s}_1 + \begin{bmatrix}\mathbf{A}_2 \\ \mathbf{B}\end{bmatrix}\mathbf{s}_2 + \begin{bmatrix}\mathbf{0} \\ \mathbf{m}\end{bmatrix}$		
$\mathbf{R}_1 \in \mathcal{R}_q^{N \times (m_1+\ell)}, \mathbf{r}_0 \in \mathcal{R}_q^N$		
$\mathbf{y}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1 d}$		
$\mathbf{y}_2 \leftarrow D_{\mathfrak{s}_2}^{m_2 d}$		
$\mathbf{w} := \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2$		
$\mathbf{v} := \mathbf{R}_1 \begin{bmatrix}\mathbf{y}_1 \\ -\mathbf{B}\mathbf{y}_2\end{bmatrix}$		
	$\xrightarrow{\mathbf{w}, \mathbf{v}}$	
	$\xleftarrow{c}$	$c \leftarrow \mathcal{C}$
$\mathbf{z}_1 := \mathbf{y}_1 + c\mathbf{s}_1$		
$\mathbf{z}_2 := \mathbf{y}_2 + c\mathbf{s}_2$		
for $i = 1, 2$:		
if $\mathsf{Rej}^{(i)}(\mathbf{z}_i, c\mathbf{s}_i, \mathfrak{s}_i, M_i) = 1$		
then $(\mathbf{z}_1, \mathbf{z}_2) := (\perp, \perp)$		
	$\xrightarrow{\mathbf{z}_1, \mathbf{z}_2}$	
		$\|\mathbf{z}_1\| \stackrel{?}{\leqslant} \mathfrak{s}_1\sqrt{2m_1 d}$
		$\|\mathbf{z}_2\| \stackrel{?}{\leqslant} \mathfrak{s}_2\sqrt{2m_2 d}$
		$\mathbf{w} \stackrel{?}{=} \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 - c\mathbf{t}_A$
		$\mathbf{v} \stackrel{?}{=} \mathbf{R}_1 \begin{bmatrix}\mathbf{z}_1 \\ c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2\end{bmatrix} + c\mathbf{r}_0$

FIGURE 5.1: Commit-and-prove system Π_{lin} for proving $\mathbf{R}_1 \begin{bmatrix}\mathbf{s}_1 \\ \mathbf{m}\end{bmatrix} + \mathbf{r}_0 = \mathbf{0}$.

5.1.2.1 *Security Analysis*

We summarise security properties of the protocol in Figure 5.1 below.

Theorem 5.1.1. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1\eta\alpha$ *and* $\mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d}$ *for some* $\gamma_1, \gamma_2 > 0$ *and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) \text{ for } i = 1, 2.$$

Suppose that $m_1 d \geqslant 5\kappa$ *and* $m_2 d \geqslant 5\kappa$. *Then, the commit-and-prove system* Π_{lin} *for the relation* R_{lin} *has statistical completeness with correctness error* $1 - \frac{1}{M_1 M_2}$.

Proof. Correctness follows directly from Theorem 4.2.1 and the fact that if $\mathbf{R}_1\mathbf{s} + \mathbf{r}_0 = 0$ then $\mathbf{R}_1\mathbf{z} + c\mathbf{r}_0 = \mathbf{R}_1\mathbf{y}$. □

Theorem 5.1.2. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1\eta\alpha$ *and* $\mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d}$ *for some* $\gamma_1, \gamma_2 > 0$ *and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) \text{ for } i = 1, 2.$$

Suppose $\kappa_{\mathsf{MLWE}} := m_2 - \kappa_{\mathsf{MSIS}} - \ell \geqslant 0$. *Then, the commit-and-prove system* Π_{lin} *for the relation* R_{lin} *is simulatable under the* $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}}, \kappa_{\mathsf{MSIS}}+\ell, \chi}$ *assumption*[1].

Proof. We prove the statement using a hybrid argument. First, we describe an efficient simulator $\mathcal{S}_1$, which knows $\mathbf{s}_1, \mathbf{m}$ and simulates both the commitment and the transcript as follows. Namely, it generates fresh randomness $\mathbf{s}_2 \leftarrow \chi^{m_2}$ and computes $(\mathbf{t}_A, \mathbf{t}_B) = \mathsf{ABDLOP.Commit}(\mathbf{s}_1, \mathbf{m}; \mathbf{s}_2)$. Next, it samples $\mathbf{z}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1 d}$ and $\mathbf{z}_2 \leftarrow D_{\mathfrak{s}_2}^{m_2 d}$. Finally, $\mathcal{S}_1$ computes

$$\mathbf{w} := \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 - c\mathbf{t}_A$$
$$\mathbf{v} := \mathbf{R}_1 \begin{bmatrix} \mathbf{z}_1 \\ c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2 \end{bmatrix} + c\mathbf{r}_0$$

and outputs a simulated transcript $(\mathbf{w}, \mathbf{v}, c, \mathbf{z}_1, \mathbf{z}_2)$ Then, by Lemma 3.3.2, the simulated commitment and transcript by $\mathcal{S}_1$ are statistically close to the honestly generated commitment and non-aborted transcript.

Further, we describe an efficient simulator $\mathcal{S}_2$, which still knows $\mathbf{s}_1, \mathbf{m}$ and simulates both the commitment and the transcript as follows. Namely,

1 Recall that χ is defined to be the uniform distribution on S_ν as described in Figure 4.1.

it executes the $\mathcal{S}_1$ algorithm but instead of generating $(\mathbf{t}_A, \mathbf{t}_B)$ honestly, it samples $\mathbf{u} \leftarrow \mathcal{R}_q^{n+\ell}$ and computes:

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

Now, we observe that under the $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}}, \kappa_{\mathsf{MSIS}}+\ell, \chi}$ assumption, the output distribution of $\mathcal{S}_2$ is computationally indistinguishable from the output distribution of $\mathcal{S}_1$.

Finally, we define our simulator $\mathcal{S}$, which has no access to private information anymore, as follows. Concretely, it executes the $\mathcal{S}_2$ algorithm but instead of generating $(\mathbf{t}_A, \mathbf{t}_B)$ as $\mathcal{S}_2$, it samples $\mathbf{u} \leftarrow \mathcal{R}_q^{n+\ell}$ and sets $(\mathbf{t}_A, \mathbf{t}_B) := \mathbf{u}$. Then, clearly the output distributions of $\mathcal{S}$ and $\mathcal{S}_2$ are identical. Hence, the statement holds by the hybrid argument.

□

Theorem 5.1.3. *Suppose* $B_1 \geqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ *and* $B_2 \geqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$. *Then, the commit-and-prove system* Π_{lin} *for the relation* R_{lin} *is knowledge sound with knowledge error* $1/|\mathcal{C}|$.

Proof. Let $\mathcal{P}^*$ be a probabilistic prover which runs in time a most T and convinces the verifier with probability $\epsilon > |\mathcal{C}|^{-1}$. By Lemma 3.3.1, there is an algorithm $\mathcal{E}$ which runs in expected time at most $2T$ and extracts two accepting transcripts with the same first message $(\mathbf{w}, \mathbf{v})$ with probability at least $\epsilon - 1/|\mathcal{C}|$:

$$\mathsf{tr}_i = \left(\mathbf{w}, \mathbf{v}, c^{(i)}, \mathbf{z}_1^{(i)}, \mathbf{z}_2^{(i)}\right) \text{ for } i = 0, 1.$$

Let us define $\bar{c} := c^{(1)} - c^{(0)}$. By definition of the challenge space, $\bar{c} \in \bar{\mathcal{C}}$ is invertible over $\mathcal{R}_q$. Next, we set

$$\bar{\mathbf{s}}_i := \frac{\mathbf{z}_i^{(1)} - \mathbf{z}_i^{(0)}}{\bar{c}} \text{ for } i = 1, 2 \quad \text{and} \quad \bar{\mathbf{m}} := \mathbf{t}_B - \mathbf{B}\bar{\mathbf{s}}_2.$$

Then, by construction

$$\mathbf{A}_1\bar{\mathbf{s}}_1 + \mathbf{A}\bar{\mathbf{s}}_2 = \frac{\left(\mathbf{A}_1\mathbf{z}_1^{(1)} + \mathbf{A}_2\mathbf{z}_2^{(1)}\right) - \left(\mathbf{A}_1\mathbf{z}_1^{(0)} + \mathbf{A}_2\mathbf{z}_2^{(0)}\right)}{\bar{c}} = \frac{\bar{c}\mathbf{t}_A}{\bar{c}} = \mathbf{t}_A$$

and therefore $\mathsf{ABDLOP.Commit}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2) = \mathbf{t}$. Moreover,

$$\|\bar{c}\bar{\mathbf{s}}_1\| = \|\mathbf{z}_1^{(1)} - \mathbf{z}_1^{(0)}\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1 d} \leqslant B_1$$

and similarly

$$\|\bar{c}\bar{\mathbf{s}}_2\| = \|\mathbf{z}_2^{(1)} - \mathbf{z}_2^{(0)}\| \leqslant 2\mathfrak{s}_2\sqrt{2m_2 d} \leqslant B_2.$$

Thus, $\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}) = 1$.

Finally, from the last verification equation we have

$$\mathbf{R}_1 \begin{bmatrix} \mathbf{z}_1^{(1)} \\ c^{(1)}\mathbf{t}_B - \mathbf{B}\mathbf{z}_2^{(1)} \end{bmatrix} + c^{(1)}\mathbf{r}_0 = \mathbf{R}_1 \begin{bmatrix} \mathbf{z}_1^{(0)} \\ c^{(0)}\mathbf{t}_B - \mathbf{B}\mathbf{z}_2^{(0)} \end{bmatrix} + c\mathbf{r}_0$$

which implies

$$\mathbf{R}_1 \begin{bmatrix} \bar{c}\bar{\mathbf{s}}_1 \\ \bar{c}\mathbf{t}_B - \bar{c}\mathbf{B}\bar{\mathbf{s}}_2 \end{bmatrix} + \bar{c}\mathbf{r}_0 = \mathbf{0}.$$

Again, since $\bar{c}$ is invertible over $\mathcal{R}_q$, we obtain

$$\mathbf{R}_1 \begin{bmatrix} \bar{\mathbf{s}}_1 \\ \bar{\mathbf{m}} \end{bmatrix} + \mathbf{r}_0 = \mathbf{0}.$$

□

5.1.3 *Function Evaluations with Vanishing Constant Coefficients*

In addition to proving the linear relation $\mathbf{R}_1\mathbf{s} + \mathbf{r}_0 = \mathbf{0}$, we now also want to prove that for public $\mathbf{u}_{1,1}, \ldots, \mathbf{u}_{M,1} \in \mathcal{R}_q^{m_1+\ell}$ and $u_{1,0}, \ldots, u_{M,0} \in \mathcal{R}_q$, the constant coefficients of

$$\mathbf{u}_{i,1}^T\mathbf{s} + u_{i,0} \in \mathcal{R}_q \text{ for } i = 1, 2, \ldots, M$$

is equal to zero. We define the corresponding relation as follows:

$$R_{\mathsf{lin-eval}} := \left\{ \begin{array}{c} \Big((\mathbf{R}_1, \mathbf{r}_0, (\mathbf{u}_{i,1}, u_{i,0})_{i\in[M]}), \mathbf{s} := (\mathbf{s}_1, \mathbf{m})\Big) : \\ \mathbf{R}_1\mathbf{s} + \mathbf{r}_0 = \mathbf{0} \wedge \tilde{x}_i = 0 \text{ where } x_i := \mathbf{u}_{i,1}^T\mathbf{s} + u_{i,0} \text{ for } i \in [M] \end{array} \right\}.$$

A naive solution to prove that $\tilde{x}_i = 0$ would be for the prover to simply send x_i to the verifier in the clear and then prove $\mathbf{u}_{i,1}^T\mathbf{s} + u_{i,0} - x_i = 0$ which is a linear equation. Then, the verifier can check itself whether the constant coefficient of x_i is indeed zero. However, the protocol is not simulatable since sending x_i in the clear reveals information about other coefficients apart from the constant one.

We first provide intuition for proving $\tilde{x}_1 = \ldots = \tilde{x}_n = 0$ with soundness error $1/q_1$. To begin with, note that this implies that for any $v_1, \ldots, v_M \in \mathbb{Z}_q$, the constant coefficient of

$$x := \sum_{i=1}^{v} v_i \left(\mathbf{u}_{i,1}^T \mathbf{s} + u_{i,0} \right) \in \mathcal{R}_q$$

is equal to zero. Now, suppose that for some i, the constant coefficient of $\mathbf{u}_{i,1}^T \mathbf{s} + u_{i,0}$ is not equal to 0. Then, if $v_1, \ldots, v_M$ are chosen uniformly at random then with probability at most $1/q_1$ we have $\tilde{x} = 0$. This will be a key observation for soundness. Thus, $v_1, \ldots, v_M$ will be random challenges output by the verifier.

As explained above, we cannot simply reveal all the coefficients of x. Hence, we first commit to a random polynomial $g \leftarrow \{x \in \mathcal{R}_q : \tilde{x} = 0\}$ which also has the constant coefficient equal to zero. Then, we mask other coefficients of x apart from the constant one by outputting:

$$h := g + x = g + \sum_{i=1}^{v} v_i \left(\mathbf{u}_{i,1}^T \mathbf{s} + u_{i,0} \right). \tag{5.2}$$

By construction, $\tilde{h} = 0$ and it can be manually checked by the verifier. Finally, we need to prove that h was constructed correctly. Note that Equation 5.2 is a simple linear relation in the committed messages $\mathbf{s}$ and g and thus can be proven identically as in the previous subsection. Indeed, define vectors $\mathbf{v}_1 := \sum_{i=1}^{M} v_i \mathbf{u}_{i,1}$ and $v_0 := \sum_{i=1}^{M} v_i u_{i,0} - h$. Then, we want to prove the following linear relation:

$$\begin{bmatrix} \mathbf{R}_1 & \mathbf{0}_{N \times 1} \\ \mathbf{v}_1^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ g \end{bmatrix} + \begin{bmatrix} \mathbf{r}_0 \\ v_0 \end{bmatrix} = \mathbf{0}.$$

The intuition for the soundness can be described as follows. If for some $i \in [M]$, the constant coefficient of $\mathbf{u}_{i,1}^T \mathbf{s} + u_{i,0}$ is not equal to 0 and polynomial g was committed before challenges $v_1, \ldots, v_M$ were generated, then with probability at most $1/q_1$ we have $\tilde{h} = 0$.

BOOSTING SOUNDNESS. Often the prime q_1 is too small to guarantee negligible soundness error. We exponentially reduce the soundness error to $q_1^{-\lambda}$ by repeating the strategy above λ times in parallel. Concretely, we first

commit to λ polynomials $\mathbf{g} := (g_1, \ldots, g_\lambda) \leftarrow \{x \in \mathcal{R}_q : \tilde{x} = 0\}^\lambda$. Then, given uniformly random challenges $\mathbf{Y} := (v_{i,j})_{i\in[\lambda],j\in[M]} \leftarrow \mathbb{Z}_q^{\lambda\times M}$, we output:

$$\begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_\lambda \end{bmatrix} := \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_\lambda \end{bmatrix} + \begin{bmatrix} v_{1,1} & \cdots & v_{1,M} \\ v_{2,1} & \cdots & v_{2,M} \\ \vdots & \vdots & \vdots \\ v_{\lambda,1} & \cdots & v_{\lambda,M} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{1,1}^T\mathbf{s} + u_{1,0} \\ \mathbf{u}_{2,1}^T\mathbf{s} + u_{2,0} \\ \vdots \\ \mathbf{u}_{M,1}^T\mathbf{s} + u_{M,0} \end{bmatrix}. \tag{5.3}$$

Thus, the verifier manually checks whether $\tilde{h}_1 = \ldots = \tilde{h}_\lambda = 0$. Finally, to prove well-formedness of $h_1, \ldots, h_\lambda$ we note that (5.3) is again a linear relation in the committed messages. Concretely, we can define the matrix $\mathbf{V}_1$ and the vector $\mathbf{v}_0$ as follows:

$$\mathbf{V}_1 := \begin{bmatrix} v_{1,1} & \cdots & v_{1,M} \\ v_{2,1} & \cdots & v_{2,M} \\ \vdots & \vdots & \vdots \\ v_{\lambda,1} & \cdots & v_{\lambda,M} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{1,1}^T \\ \mathbf{u}_{2,1}^T \\ \vdots \\ \mathbf{u}_{M,1}^T \end{bmatrix} \tag{5.4}$$

and

$$\mathbf{v}_0 := \begin{bmatrix} v_{1,1} & \cdots & v_{1,M} \\ v_{2,1} & \cdots & v_{2,M} \\ \vdots & \vdots & \vdots \\ v_{\lambda,1} & \cdots & v_{\lambda,M} \end{bmatrix} \begin{bmatrix} u_{1,0} \\ u_{2,0} \\ \vdots \\ u_{M,0} \end{bmatrix} - \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_\lambda \end{bmatrix}. \tag{5.5}$$

Then, proving well-formedness of h_i and $\mathbf{R}_1\mathbf{s} + \mathbf{r}_0 = \mathbf{0}$ is equivalent to proving:

$$\begin{bmatrix} \mathbf{R}_1 & \mathbf{0}_{N\times\lambda} \\ \mathbf{V}_1 & \mathbf{I}_\lambda \end{bmatrix} \begin{bmatrix} \mathbf{s} \\ \mathbf{g} \end{bmatrix} + \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{v}_0 \end{bmatrix} = \mathbf{0}. \tag{5.6}$$

We provide a commit-and-prove system $\Pi_{\mathsf{lin-eval}} := (\mathsf{ABDLOP}, \mathcal{P}, \mathcal{V})$ for the relation $R_{\mathsf{lin-eval}}$ in Figure 5.2. Namely, prover $\mathcal{P}$ starts by committing to the vector $\mathbf{g}$, i.e. computing $\mathbf{t}_g := \mathbf{B}_{\mathsf{ext}}\mathbf{s} + \mathbf{g}$[2]. Then, given a challenge matrix $\mathbf{Y} = (v_{i,j})$ from verifier $\mathcal{V}$, the prover outputs the vector $\mathbf{h}$ defined in Equation 5.3. Finally, $\mathcal{P}$ runs the subprotocol Π_{lin} to prove well-formedess of $\mathbf{h}$ as well as $\mathbf{R}_1\mathbf{s} + \mathbf{r}_0 = \mathbf{0}$, or alternatively (5.6). The verifier then checks whether the constant coefficients of $h_1, \ldots, h_\lambda$ are indeed zeroes and if Π_{lin} verifies.

2 As explained in Section 4.1, having the matrix $\mathbf{B}_{\mathsf{ext}}$ allows appending further commitments to $(\mathbf{t}_A, \mathbf{t}_B)$.

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
Inputs:		
$pp.\dim = (q, d, \kappa_{\mathsf{MSIS}}, m_1, m_2, \ell, \ell_{\mathsf{ext}} := \lambda)$		$pp.\dim, pp.\mathsf{norms}, pp.\mathsf{mat}$
$pp.\mathsf{norms} = (\nu, \omega, \alpha, B_1, B_2)$		$\mathbf{t}_A, \mathbf{t}_B, \mathbf{R}_1, \mathbf{r}_0$
$pp.\mathsf{mat} = (\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \mathbf{B}_{\mathsf{ext}})$		$\mathbf{u}_{1,1}, \ldots, \mathbf{u}_{M,1}, u_{1,0}, \ldots, u_{M,0}$
$\mathbf{s}_1 \in \mathcal{R}_q^{m_1}, \mathbf{s}_2 \in \mathcal{R}_q^{m_2}, \mathbf{m} \in \mathcal{R}_q^{\ell}$ so that $\|\mathbf{s}_1\| \leqslant \alpha$		
$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} := \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{s}_1 + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} \mathbf{s}_2 + \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix}$		
$\mathbf{R}_1 \in \mathcal{R}_q^{N \times (m_1 + \ell)}, \mathbf{r}_0 \in \mathcal{R}_q^N$		
$\mathbf{u}_{1,1}, \ldots, \mathbf{u}_{M,1} \in \mathcal{R}_q^{m_1+\ell}, u_{1,0}, \ldots, u_{M,0} \in \mathcal{R}_q$		
$\mathbf{g} \leftarrow \{x \in \mathcal{R}_q : \tilde{x} = 0\}^{\lambda}$		
$\mathbf{t}_g := \mathbf{B}_{\mathsf{ext}} \mathbf{s}_2 + \mathbf{g}$		
	$\xrightarrow{\mathbf{t}_g}$	
	$\xleftarrow{\mathbf{Y}}$	$\mathbf{Y} := (v_{i,j})_{i \in [\lambda], j \in [M]} \leftarrow \mathbb{Z}_q^{\lambda \times M}$
$\mathbf{h} := \mathbf{g} + \mathbf{Y} \begin{bmatrix} \mathbf{u}_{1,1}^T \mathbf{s} + u_{1,0} \\ \mathbf{u}_{2,1}^T \mathbf{s} + u_{2,0} \\ \vdots \\ \mathbf{u}_{M,1}^T \mathbf{s} + u_{M,0} \end{bmatrix}$		
	$\xrightarrow{\mathbf{h}}$	
run Π_{lin} with the following inputs:		accept if (i) Π_{lin} verifies and
$pp.\dim := (q, d, \kappa_{\mathsf{MSIS}}, m_1, m_2, \ell + \lambda, 0)$		(ii) $\tilde{h}_1 = \ldots = \tilde{h}_{\lambda} = 0$
$pp.\mathsf{norms} := pp.\mathsf{norms}$		
$pp.\mathsf{mat} := \left(\mathbf{A}_1, \mathbf{A}_2, \begin{bmatrix} \mathbf{B} \\ \mathbf{B}_{\mathsf{ext}} \end{bmatrix}, \varnothing \right)$		
$(\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m})) := (\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m} \parallel \mathbf{g}))$		
$\mathbf{V}_1 := \mathbf{Y} \begin{bmatrix} \mathbf{u}_{1,1}^T \\ \mathbf{u}_{2,1}^T \\ \vdots \\ \mathbf{u}_{M,1}^T \end{bmatrix}, \mathbf{v}_0 := \mathbf{Y} \begin{bmatrix} u_{1,0} \\ u_{2,0} \\ \vdots \\ u_{M,0} \end{bmatrix} - \mathbf{h}$		
$\mathbf{R}_1 := \begin{bmatrix} \mathbf{R}_1 & \mathbf{0} \\ \mathbf{U}_1 & \mathbf{I}_{\lambda} \end{bmatrix}, \mathbf{r}_0 := \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{u}_0 \end{bmatrix}$		

FIGURE 5.2: Commit-and-prove system $\Pi_{\mathsf{lin-eval}}$ for proving $\mathbf{R}_1 \mathbf{s} + \mathbf{r}_0 = \mathbf{0}$ and that the constant coefficient of $\mathbf{u}_{i,1}^T \mathbf{s} + u_{i,0}$ vanishes for $i = 1, 2, \ldots, M$.

5.1.3.1 *Security Analysis*

We summarise security properties of the protocol in Figure 5.2 below.

Theorem 5.1.4. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1\eta\alpha$ *and* $\mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d}$ *for some* $\gamma_1, \gamma_2 > 0$ *and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) \text{ for } i = 1, 2.$$

Suppose that $m_1 d \geqslant 5\kappa$ *and* $m_2 d \geqslant 5\kappa$*. Then, the commit-and-prove system* $\Pi_{\mathsf{lin-eval}}$ *for the relation* $R_{\mathsf{lin-eval}}$ *has statistical completeness with correctness error* $1 - \frac{1}{M_1 M_2}$.

Proof. Take any $i \in [\lambda]$. Then, if the constant coefficients of g_i and $\mathbf{u}_{j,1}^T\mathbf{s} + u_{j,0}$ for $j \in [M]$ are zeroes, then we must have that the constant coefficient of

$$h_i = g_i + \sum_{j=1}^{M} v_{i,j}\left(\mathbf{u}_{j,1}^T\mathbf{s} + u_{j,0}\right)$$

is also equal to zero. The rest of the correctness argument follows from the proof of Theorem 5.1.1. □

Theorem 5.1.5. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1\eta\alpha$ *and* $\mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d}$ *for some* $\gamma_1, \gamma_2 > 0$ *and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) \text{ for } i = 1, 2.$$

Assume $\kappa_{\mathsf{MLWE}} := m_2 - \kappa_{\mathsf{MSIS}} - \ell - \lambda \geqslant 0$*. Then, the commit-and-prove system* $\Pi_{\mathsf{lin-eval}}$ *for relation* $R_{\mathsf{lin-eval}}$ *is simulatable under the* $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}},\kappa_{\mathsf{MSIS}}+\ell+\lambda,\chi}$ *assumption.*

Proof. The proof is almost identical to the one for Theorem 5.1.2 with the addition that the simulator $\mathcal{S}$ samples $\mathbf{t}_g \leftarrow \mathcal{R}_q^\lambda$ and $\mathbf{h} \leftarrow \{x \in \mathcal{R}_q : \tilde{x} = 0\}^\lambda$. Indeed, since in an honest execution $\mathbf{t}_g$ is chosen uniformly at random from $\{x \in \mathcal{R}_q : \tilde{x} = 0\}^\lambda$, the distribution of the vector $\mathbf{h}$ constructed as in Equation 5.3 is also uniformly random over $\{x \in \mathcal{R}_q : \tilde{x} = 0\}^\lambda$. □

Theorem 5.1.6. *Suppose* $B_1 \geqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ *and* $B_2 \geqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$*. Then, the commit-and-prove system* $\Pi_{\mathsf{lin-eval}}$ *for the relation* $R_{\mathsf{lin-eval}}$ *is knowledge sound with knowledge error* $|\mathcal{C}|^{-1} + q_1^{-\lambda}$.

Proof. Let $\mathcal{P}^*$ be a probabilistic prover which convinces the verifier with probability $\epsilon > |\mathcal{C}|^{-1} + q_1^{-\lambda}$ and runs in time at most T. Define a deterministic algorithm $\mathcal{A}(\rho_P, \rho_E, \mathrm{Y})$ which given randomness $\rho = (\rho_P, \rho_E) \in \mathfrak{R}_P \times \mathfrak{R}_E$ and challenge $\mathrm{Y} \in \mathbb{Z}_q^{\lambda \times M}$ does the following. It first runs $\mathcal{P}^*(\rho_P)$ on randomness ρ_P with challenge Y and stops after the third round. Let $\mathbf{t}_g$ and $\mathbf{h}$ be the output of $\mathcal{P}^*$ in the first and third round respectively. Then, it runs the extractor $\mathcal{E}^*(\rho_E)$ defined in the proof of Theorem 5.1.3 with randomness ρ_E (which runs $\mathcal{P}^*(\rho_P, \mathrm{Y})$ in a black-box way).

We say that $\mathcal{A}$ succeeds if $\mathcal{A}$ outputs $(\mathbf{t}_g, \mathrm{Y}, \mathbf{h}, \bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{g}}, \bar{\mathbf{s}}_2, \bar{c})$ such that $\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{\mathbf{g}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B \parallel \mathbf{t}_g) = 1$ and $\tilde{h}_1 = \ldots = \tilde{h}_\lambda = 0$ and

$$\begin{bmatrix} \mathbf{R}_1 & \mathbf{0}_{N \times \lambda} \\ \mathbf{V}_1 & \mathbf{I}_\lambda \end{bmatrix} \begin{bmatrix} \bar{\mathbf{s}} \\ \bar{\mathbf{g}} \end{bmatrix} + \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{v}_0 \end{bmatrix} = \mathbf{0}$$

where $\mathbf{V}_1, \mathbf{v}_0$ are defined as in (5.4),(5.5) and $\bar{\mathbf{s}} = \bar{\mathbf{s}}_1 \parallel \bar{\mathbf{m}}$. It is easy to see that by Theorem 5.1.3, the probability that $\mathcal{A}$ succeeds for random ρ and Y is at least $\epsilon - 1/|\mathcal{C}|$. Moreover, the expected runtime of $\mathcal{A}(\rho_P, \rho_E, \mathrm{Y})$ for any fixed ρ_P, Y and $\rho_E \leftarrow \mathfrak{R}_E$ is at most $2T$.

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

$$\bar{\mathbf{s}}^{(\rho, \mathrm{Y})} := \left(\bar{\mathbf{s}}_1^{(\rho, \mathrm{Y})}, \bar{\mathbf{m}}^{(\rho, \mathrm{Y})} \right).$$

Finally, we define

$$H' := \left\{ (\rho, \mathrm{Y}) \in H : \exists k \in [M], \text{ const. coeff. of } \mathbf{u}_{k,1}^T \bar{\mathbf{s}}^{(\rho, \mathrm{Y})} + u_{k,0} \text{ is non-zero} \right\}.$$

Then, we have the following claim.

Claim 5.1.7. If $(\rho_P, \rho_E, \mathrm{Y}) \in H$ then

$$\Pr_{(\rho_E', \mathrm{Y}') \leftarrow \mathfrak{R}_E \times \mathbb{Z}_q^{\lambda \times M}} [(\rho_P, \rho_E', \mathrm{Y}') \in H] > 0.$$

Moreover, if $(\rho_P, \rho_E, \mathrm{Y}) \in H'$ then

$$\Pr_{\mathrm{Y}' \leftarrow \mathbb{Z}_q^{\lambda \times M}} \left[\forall i \in [\lambda], \tilde{x}_i = 0 \,\middle|\, x_i := \bar{g}_i^{(\rho, \mathrm{Y})} + \sum_{j=1}^{M} v_{i,j}' \left(\mathbf{u}_{j,1}^T \bar{\mathbf{s}}^{(\rho, \mathrm{Y})} + u_{j,0} \right) \right] \leqslant q_1^{-\lambda}.$$

Proof. First, we observe that if $(\rho_P, \rho_E, \mathrm{Y}) \in H$ then

$$\Pr_{(\rho'_E, \mathrm{Y}') \leftarrow \mathfrak{R}_E \times \mathbb{Z}_q^{\lambda \times M}}[(\rho_P, \rho'_E, \mathrm{Y}') \in H] \geqslant \Pr_{(\rho'_E, \mathrm{Y}') \leftarrow \mathfrak{R}_E \times \mathbb{Z}_q^{\lambda \times M}}[\rho'_E = \rho_E \wedge \mathrm{Y}' = \mathrm{Y}]$$
$$> 0.$$

Now, if the constant coefficient of $\mathbf{u}_{k,1}^T \bar{\mathbf{s}}^{(\rho,\mathrm{Y})} + u_{k,0}$ is non-zero for some k then for fixed i, with probability at most $1/q_1$ the constant coefficient of

$$\bar{g}_i^{(\rho,\mathrm{Y})} + \sum_{j=1}^{M} \upsilon'_{i,j} \left(\mathbf{u}_{j,1}^T \bar{\mathbf{s}}^{(\rho,\mathrm{Y})} + u_{j,0} \right)$$

vanishes. The statement follows by parallel repetition. □

Now, we can define our extractor $\mathcal{E}$. It does the following.

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

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

- $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2) \neq (\bar{\mathbf{s}}'_1, \bar{\mathbf{s}}'_2)$ and

$$\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}) = 1 = \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}'_1, \bar{\mathbf{m}}', \bar{\mathbf{s}}'_2, \bar{c}'; \mathbf{t}).$$

- $\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}) = 1$ and for all $i \in [M]$, the constant coefficient of $\mathbf{u}_{i,1}^T \bar{\mathbf{s}} + u_{i,0}$ is zero where $\bar{\mathbf{s}} := (\bar{\mathbf{s}}_1, \bar{\mathbf{m}})$

where $\mathbf{t} := \mathbf{t}_A \parallel \mathbf{t}_B$. In the first case, we break the binding property of the ABDLOP commitment scheme. On the other hand, we extract the witness in the second case. Then, we have the following claims about $\mathcal{E}$.

Claim 5.1.8. The expected number of calls to $\mathcal{A}$ is at most 2.

Proof. Let X be the expected number of calling $\mathcal{A}$. Take any $i \in \mathfrak{R}_P$ and denote ϵ_i to be the probability that $\mathcal{A}(i, \rho_E, \mathrm{Y})$ succeeds for $\rho_E \leftarrow \mathfrak{R}_E$ and $\mathrm{Y} \leftarrow \mathbb{Z}_q^{\lambda \times M}$. If in the first step of $\mathcal{E}$ algorithm $\mathcal{A}$ succeeds and $\rho_P = i$, then

the expected number of running $\mathcal{A}$ in the second step is at most $1/\epsilon_i$. Next, define E to be the event that $\mathcal{A}$ succeeds in the first step. Then,

$$\begin{aligned}\mathbb{E}[X] &= \frac{1}{|\mathfrak{R}_P|}\sum_{i\in\mathfrak{R}}\mathbb{E}[X|\rho_P = i] \\ &= \frac{1}{|\mathfrak{R}_P|}\sum_{i\in\mathfrak{R}_P}\mathbb{E}[X|\rho_P = i\wedge E]\cdot\epsilon_i + \mathbb{E}[X|\rho_P = i\wedge\neg E]\cdot(1-\epsilon_i) \\ &\leqslant \frac{1}{|\mathfrak{R}_P|}\sum_{i\in\mathfrak{R}_P}\left(1+\frac{1}{\epsilon_i}\right)\cdot\epsilon_i + 1\cdot(1-\epsilon_i) = 2.\end{aligned}$$

□

We conclude from the claim above that the expected runtime of $\mathcal{E}$ is at most $4T$.

Claim 5.1.9. Probability that $\mathcal{E}$ succeeds is at least $\epsilon - 1/|\mathcal{C}| - q_1^{-\lambda}$.

Proof. First, we observe that $\mathcal{E}$ terminates (without aborting) with probability at least $\epsilon - 1/|\mathcal{C}|$. Suppose $\mathcal{E}$ indeed terminates and let us write $(\mathbf{t}_g, \Upsilon, \mathbf{h}, \bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{g}}, \bar{\mathbf{s}}_2, \bar{c})$ and $(\mathbf{t}_g, \Upsilon', \mathbf{h}', \bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{g}}', \bar{\mathbf{s}}_2', \bar{c}')$ to be the respective outputs of $\mathcal{A}$ in the first and second step of $\mathcal{E}$. We have the following three disjoint cases:

Case 1. $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{g}}, \bar{\mathbf{s}}_2) \neq (\bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{g}}', \bar{\mathbf{s}}_2')$ and $\tilde{h}_j = \tilde{h}_j' = 0$ for $j \in [\lambda]$ and

$$\begin{bmatrix}\mathbf{R}_1 & \mathbf{0}_{N\times\lambda}\\ \mathbf{V}_1 & \mathbf{I}_\lambda\end{bmatrix}\begin{bmatrix}\bar{\mathbf{s}}\\ \bar{\mathbf{g}}\end{bmatrix} + \begin{bmatrix}\mathbf{r}_0\\ \mathbf{v}_0\end{bmatrix} = \mathbf{0} \text{ and } \begin{bmatrix}\mathbf{R}_1 & \mathbf{0}_{N\times\lambda}\\ \mathbf{V}_1' & \mathbf{I}_\lambda\end{bmatrix}\begin{bmatrix}\bar{\mathbf{s}}'\\ \bar{\mathbf{g}}'\end{bmatrix} + \begin{bmatrix}\mathbf{r}_0\\ \mathbf{v}_0'\end{bmatrix} = \mathbf{0}$$

and

$$\begin{aligned}1 &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{\mathbf{g}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B \parallel \mathbf{t}_g) \\ &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1', \bar{\mathbf{m}}' \parallel \bar{\mathbf{g}}', \bar{\mathbf{s}}_2', \bar{c}'; \mathbf{t}_A \parallel \mathbf{t}_B \parallel \mathbf{t}_g).\end{aligned}$$

Case 2. $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{g}}, \bar{\mathbf{s}}_2) = (\bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{g}}', \bar{\mathbf{s}}_2')$ and $\tilde{h}_j = \tilde{h}_j' = 0$ for $j \in [\lambda]$ and

$$\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{\mathbf{g}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B \parallel \mathbf{t}_g) = 1$$

and

$$\begin{bmatrix}\mathbf{R}_1 & \mathbf{0}_{N\times\lambda}\\ \mathbf{V}_1 & \mathbf{I}_\lambda\end{bmatrix}\begin{bmatrix}\bar{\mathbf{s}}\\ \bar{\mathbf{g}}\end{bmatrix} + \begin{bmatrix}\mathbf{r}_0\\ \mathbf{v}_0\end{bmatrix} = \mathbf{0} \text{ and } \begin{bmatrix}\mathbf{R}_1 & \mathbf{0}_{N\times\lambda}\\ \mathbf{V}_1' & \mathbf{I}_\lambda\end{bmatrix}\begin{bmatrix}\bar{\mathbf{s}}\\ \bar{\mathbf{g}}\end{bmatrix} + \begin{bmatrix}\mathbf{r}_0\\ \mathbf{v}_0'\end{bmatrix} = \mathbf{0}$$

and for all $i \in [M]$ the constant coefficient of $\mathbf{u}_{i,1}^T \bar{\mathbf{s}} + u_{i,0}$ is zero.

Case 3. $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{g}}, \bar{\mathbf{s}}_2) = (\bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{g}}', \bar{\mathbf{s}}_2')$ and $\tilde{h}_j = \tilde{h}_j' = 0$ for $j \in [\lambda]$ and

$$\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{\mathbf{g}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B \parallel \mathbf{t}_g) = 1$$

and

$$\begin{bmatrix} \mathbf{R}_1 & \mathbf{0}_{N\times\lambda} \\ \mathbf{V}_1 & \mathbf{I}_\lambda \end{bmatrix} \begin{bmatrix} \bar{\mathbf{s}} \\ \bar{\mathbf{g}} \end{bmatrix} + \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{v}_0 \end{bmatrix} = \mathbf{0} \text{ and } \begin{bmatrix} \mathbf{R}_1 & \mathbf{0}_{N\times\lambda} \\ \mathbf{V}_1' & \mathbf{I}_\lambda \end{bmatrix} \begin{bmatrix} \bar{\mathbf{s}} \\ \bar{\mathbf{g}} \end{bmatrix} + \begin{bmatrix} \mathbf{r}_0 \\ \mathbf{v}_0' \end{bmatrix} = \mathbf{0}$$

and there exists $i \in [M]$ so that the constant coefficient of $\mathbf{u}_{i,1}^T \bar{\mathbf{s}} + u_{i,0}$ is non-zero. Here, we define $\mathbf{V}_1', \mathbf{v}_0'$ are as in (5.4) and (5.5) with respect to the challenge Y'.

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

$$\epsilon - 1/|\mathcal{C}| \leqslant \Pr[\mathcal{E} \text{ terminates}] = \Pr[E_1 \vee E_2 \vee E_3]$$

and $\Pr[\mathcal{E} \text{ succeeds}] \geqslant \Pr[E_1 \vee E_2]$. Hence, we only need to upper-bound the probability $\Pr[E_3]$. Define $F(\bar{\mathbf{s}}, \bar{\mathbf{g}})$ to be the event that for all $i \in [\lambda]$, the constant coefficient of

$$\bar{g}_i + \sum_{j=1}^{M} v_{i,j}' \left(\mathbf{u}_{j,1}^T \bar{\mathbf{s}} + u_{j,0} \right)$$

vanishes. We apply Claim 5.1.3.1 as follows:

$$\begin{aligned}
\Pr[E_3] &\leqslant \Pr\left[\begin{array}{c} (\mathcal{A}(\rho, \mathrm{Y}) \text{ succeeds}) \wedge F(\bar{\mathbf{s}}, \bar{\mathbf{g}}) \\ \wedge \left(\exists i \in [M] : \text{ const. coeff. of } \mathbf{u}_{i,1}^T \bar{\mathbf{s}} + u_{i,0} \text{ is non-zero} \right) \end{array} \right] \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho, \mathrm{Y}) \in H'} \Pr_{(\rho_E', \mathrm{Y}') \leftarrow H(\rho_P)} \left[F\left(\bar{\mathbf{s}}^{(\rho, \mathrm{Y})}, \bar{\mathbf{g}}^{(\rho, \mathrm{Y})} \right) \right] \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho, \mathrm{Y}) \in H'} \frac{\Pr_{\mathrm{Y}' \leftarrow \mathbb{Z}_q^{\lambda \times M}} \left[F\left(\bar{\mathbf{s}}^{(\rho, \mathrm{Y})}, \bar{\mathbf{g}}^{(\rho, \mathrm{Y})} \right) \right]}{\Pr_{(\rho_E', \mathrm{Y}') \leftarrow \mathfrak{R}_E \times \mathbb{Z}_q^{\lambda \times M}} [(\rho_E', \mathrm{Y}') \in H(\rho_P)]} \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho, \mathrm{Y}) \in H'} \frac{q_1^{-\lambda} \cdot q^{\lambda M} \cdot |\mathfrak{R}_E|}{|H(\rho_P)|} \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho, \mathrm{Y}) \in H} \frac{q_1^{-\lambda} \cdot q^{\lambda M} \cdot |\mathfrak{R}_E|}{|H(\rho_P)|} \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{\rho_P \in \mathfrak{R}_P} \sum_{(\rho_E, \mathrm{Y}) \in H(\rho_P)} \frac{q_1^{-\lambda} \cdot q^{\lambda M} \cdot |\mathfrak{R}_E|}{|H(\rho_P)|} = q_1^{-\lambda}
\end{aligned}$$

which concludes the proof. □

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

5.1.3.2 *Proving Linear Equations over $\mathbb{Z}_q$*

We apply the commit-and-prove system $\Pi_{\mathsf{lin-eval}}$ to prove linear equations between the polynomial coefficients of $\mathbf{s}_1$ and $\mathbf{m}$. Namely, suppose that we want to prove for public vectors $\mathbf{a}_1 \in \mathcal{R}_q^{m_1}$, $\mathbf{a}_2 \in \mathcal{R}_q^{\ell}$ and $u \in \mathbb{Z}_q$ that:

$$\langle \mathbf{a}_1, \mathbf{s}_1 \rangle + \langle \mathbf{a}_2, \mathbf{m} \rangle \equiv u \pmod{q}. \tag{5.7}$$

In order to use the techniques described in this section, we present the following result.

Lemma 5.1.10. *Let $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_q^{kd}$ and define the polynomial $f = \sigma_{-1}(\mathbf{x})^T \mathbf{y} \in \mathcal{R}_q$. Then, the constant coefficient of f is equal to $\langle \mathbf{x}, \mathbf{y} \rangle$.*

Proof. Denote $\mathbf{x} = (x_1, \ldots, x_k)$ and $\mathbf{y} = (y_1, \ldots, y_k)$. We just need to prove that for every $i \in [k]$, the first coefficient of $f_i := \sigma_{-1}(x_i) y_i \in \mathcal{R}_q$ equals to $\langle x_i, y_i \rangle$. Indeed, let us explicitly write f_i as a product of two polynomials:

$$\left(x_{i,0} - x_{i,d-1} X - \ldots - x_{i,1} X^{d-1}\right) \left(y_{i,0} + y_{i,1} X + \ldots + y_{i,d-1} X^{d-1}\right).$$

Then, it is easy to see that the constant coefficient of $\sigma_{-1}(x_i) y_i$ is equal to

$$x_{i,0} y_{i,0} + x_{i,1} y_{i,1} + \ldots + x_{i,d-1} y_{i,d-1} = \langle x_i, y_i \rangle.$$

□

Using the lemma above, we see that (5.7) holds if and only if the constant coefficient of

$$\sigma_{-1}(\mathbf{a}_1)^T \mathbf{s}_1 + \sigma_{-1}(\mathbf{a}_2)^T \mathbf{m} - u \in \mathcal{R}_q$$

is equal to zero. Hence, we can define the linear function $F : \mathcal{R}_q^{m_1 + \ell} \to \mathcal{R}_q$:

$$F(\mathbf{x}) := \begin{bmatrix} \sigma_{-1}(\mathbf{a}_1)^T & \sigma_{-1}(\mathbf{a}_2)^T \end{bmatrix} \mathbf{x} - u$$

and prove that the constant coefficient of $F(\mathbf{s}_1, \mathbf{m})$ equals to zero using $\Pi_{\mathsf{lin-eval}}$. It is easy to see that this argument extends naturally to the case when one wants to prove multiple equations of the form (5.7).

5.2 PROOFS OF QUADRATIC RELATIONS

We show how to prove quadratic equations between committed messages $(\mathbf{s}_1, \mathbf{m})$ using the ABDLOP commitment. For various applications, we will also need to prove relations between the images of $(\mathbf{s}_1, \mathbf{m})$ under an automorphism $\sigma \in \mathsf{Aut}(\mathcal{R})$, e.g. $\sigma(\mathbf{s}_1)^T \mathbf{s}_1 + \sigma(\mathbf{m})^T \mathbf{m} = 0$ which is a quadratic relation involving σ.

More concretely, let $\sigma \in \mathsf{Aut}(\mathcal{R})$ be a public automorphism over $\mathcal{R}$ of degree k and for presentation purposes define:

$$\langle x \rangle_\sigma := \left(x, \sigma(x), \ldots, \sigma^{k-1}(x)\right) \in \mathcal{R}_q^k \text{ for } x \in \mathcal{R}_q.$$

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

Lemma 5.2.1. *For any $\mathbf{x}, \mathbf{y} \in \mathcal{R}_q^n$ and any $c \in \mathcal{R}_q$ such that $\sigma(c) = c$:*

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

Also, denote $\mathbf{J}_{n,k} := \mathbf{I}_n \otimes \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \in \mathcal{R}_q^{n \times kn}$. Then

$$\mathbf{x} = \mathbf{J}_{n,k} \langle \mathbf{x} \rangle_\sigma.$$

Suppose we have message vectors $\mathbf{s}_1 \in \mathcal{R}_q^{m_1}$ and $\mathbf{m} \in \mathcal{R}_q^{\ell}$ such that $\|\mathbf{s}_1\| \leqslant \alpha$. Then, we consider the following statements:

- *Single quadratic equation with automorphisms.* For a public $k(m_1 + \ell)$-variate quadratic function f over $\mathcal{R}_q$,

$$f\left(\langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma\right) = 0.$$

- *Many quadratic equations with automorphisms.* For N public $k(m_1 + \ell)$-variate quadratic functions $f_1, \ldots, f_N$ over $\mathcal{R}_q$,

$$f_j\left(\langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma\right) = 0 \text{ for } j \in [N].$$

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

– let $x_j := F_j\left(\langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma\right) \in \mathcal{R}_q$ for $j \in [M]$. Then $\tilde{x}_1 = \ldots = \tilde{x}_M = 0$.

Clearly, the statements presented at the beginning of this chapter are a special case when σ is the identity automorphism σ_1.

Remark. Similarly as for [ALS20], our techniques can be easily generalised to prove higher degree relations. Concretely, if we want to prove degree k equations, we end up committing to $k-1$ additional garbage terms. Throughout this thesis, however, we will only consider quadratic relations.

5.2.1 *Single Quadratic Equation with Automorphisms*

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

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

Suppose the prover wants to prove knowledge of the message

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

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

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

where $r_0 \in \mathcal{R}_q, \mathbf{r}_1 \in \mathcal{R}_q^{k(m_1+\ell)}$ and $\mathbf{R}_2 \in \mathcal{R}_q^{k(m_1+\ell)\times k(m_1+\ell)}$. Hence, we define the corresponding relation:

$$R_{\mathsf{quad}} := \left\{ \begin{array}{c} ((\mathbf{R}_2, \mathbf{r}_1, r_0), (\mathbf{s}_1, \mathbf{m})) : \\ \mathbf{s}^T \mathbf{R}_2 \mathbf{s} + \mathbf{r}_1^T \mathbf{s} + r_0 = 0 \text{ where } \mathbf{s} := (\langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma) \end{array} \right\}.$$

In order to prove this relation, let us consider the protocol for proving linear equations over $\mathcal{R}_q$ in Figure 5.1. In the last round, the honest prover sends the *masked openings* $\mathbf{z}_i = c\mathbf{s}_i + \mathbf{y}_i$ of $\mathbf{s}_i$ for $i = 1, 2$ where the challenge space $\mathcal{C}$ is defined as in (3.5) with the σ automorphism. Even though this is not the case for $\mathbf{m}$, we can define the masked opening of $\mathbf{m}$ as

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

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

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

$$\mathbf{y} := \begin{bmatrix} \langle \mathbf{y}_1 \rangle_\sigma \\ -\langle \mathbf{B}\mathbf{y}_2 \rangle_\sigma \end{bmatrix} \in \mathcal{R}_q^{k(m_1+\ell)} \tag{5.8}$$

and

$$\mathbf{z} := \begin{bmatrix} \langle \mathbf{z}_1 \rangle_\sigma \\ \langle \mathbf{z}_m \rangle_\sigma \end{bmatrix} = c \begin{bmatrix} \langle \mathbf{s}_1 \rangle_\sigma \\ \langle \mathbf{m} \rangle_\sigma \end{bmatrix} + \begin{bmatrix} \langle \mathbf{y}_1 \rangle_\sigma \\ -\langle \mathbf{B}\mathbf{y}_2 \rangle_\sigma \end{bmatrix} = c\mathbf{s} + \mathbf{y}. \tag{5.9}$$

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

$$\mathbf{z}^T \mathbf{R}_2 \mathbf{z} + c\mathbf{r}_1^T \mathbf{z} + c^2 r_0 = c^2 \left(\mathbf{s}^T \mathbf{R}_2 \mathbf{s} + \mathbf{r}_1^T \mathbf{s} + r_0 \right) + c g_1 + g_0 \tag{5.10}$$

where polynomials g_1 and g_0 are defined as:

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

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

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

which is a simple transformation of (5.10) when $\mathbf{s}^T \mathbf{R}_2 \mathbf{s} + \mathbf{r}_1^T \mathbf{s} + r_0 = 0$.

We present the commit-and-prove system $\Pi_{\mathsf{quad}} = (\mathsf{ABDLOP}, \mathcal{P}, \mathcal{V})$ for the relation R_{quad} in Figure 5.3. Prover $\mathcal{P}$ starts by sampling masking vectors $\mathbf{y}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1 d}, \mathbf{y}_2 \leftarrow D_{\mathfrak{s}}^{m_2 d}$ and computing $\mathbf{w} = \mathbf{A}_1 \mathbf{y}_1 + \mathbf{A}_2 \mathbf{y}_2$. Then, it calculates $g_1 = \mathbf{s}^T \mathbf{R}_2 \mathbf{y} + \mathbf{y}^T \mathbf{R}_2 \mathbf{s} + \mathbf{r}_1^T \mathbf{y}$, where $\mathbf{y}$ is defined in (5.8), and the commitment $t = \mathbf{b}_{\mathsf{ext}}^T \mathbf{s}_2 + g_1$ to g_1. Finally, the prover sets $v = \mathbf{y}^T \mathbf{R}_2 \mathbf{y} + \mathbf{b}_{\mathsf{ext}}^T \mathbf{y}_2$ and sends $\mathbf{w}, t, v$ to the verifier.

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

Eventually, the verifier checks whether $\mathbf{z}_1$ and $\mathbf{z}_2$ have small norms, $\mathbf{A}_1 \mathbf{z}_1 + \mathbf{A}_2 \mathbf{z}_2 = \mathbf{w} + c\mathbf{t}_A$ and $\mathbf{z}^T \mathbf{R}_2 \mathbf{z} + c\mathbf{r}_1^T \mathbf{z} + c^2 r_0 - f = v$ where $\mathbf{z}$ is defined in (5.9) and f is defined as $f = ct - \mathbf{b}_{\mathsf{ext}}^T \mathbf{z}_2$.

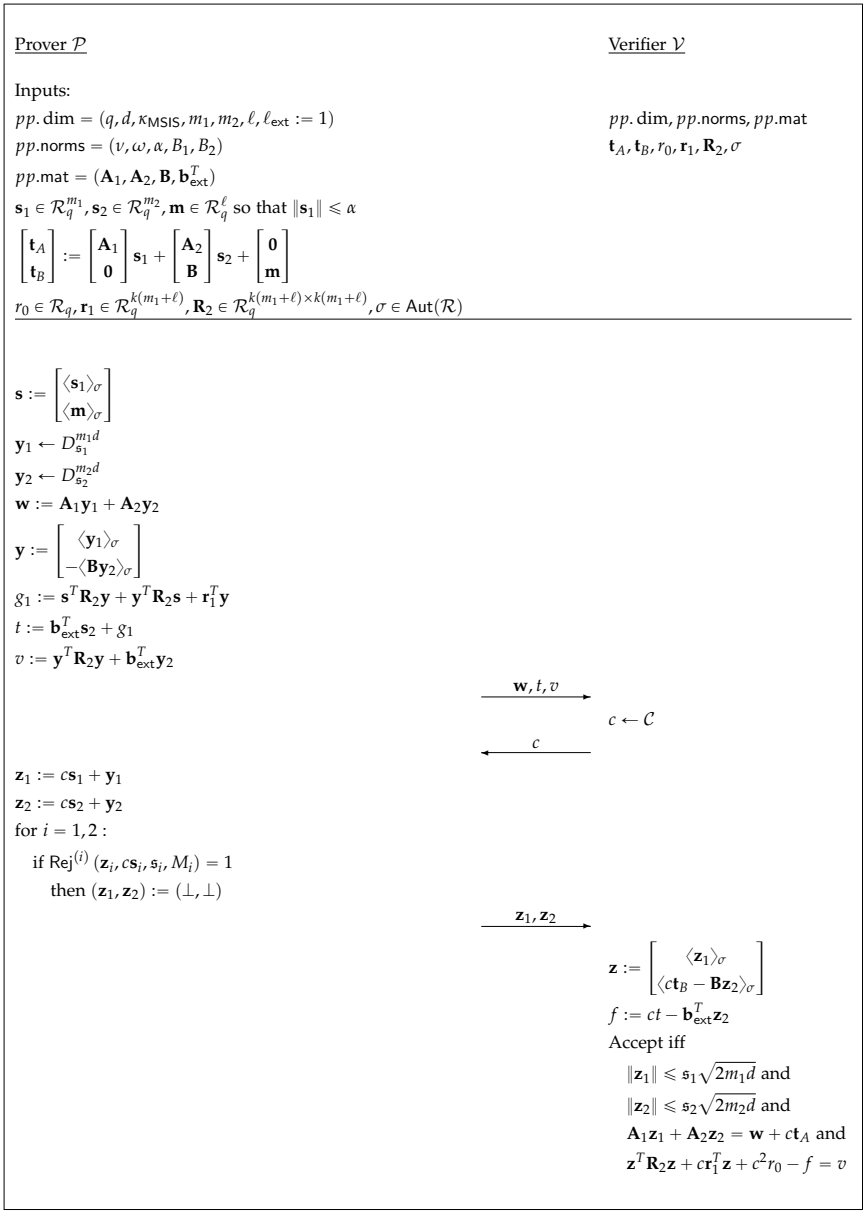

FIGURE 5.3: Commit-and-prove system Π_{quad} for proving $\mathbf{s}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\mathbf{s} + r_0 = 0$. Here, $\mathsf{Rej}^{(1)}, \mathsf{Rej}^{(2)}$ are rejection sampling algorithms.

5.2.1.1 *Security Analysis*

We summarise security properties of the protocol in Figure 5.3 below.

Theorem 5.2.2. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1\eta\alpha$ *and* $\mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d}$ *for some* $\gamma_1, \gamma_2 > 0$ *and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) \text{ for } i = 1, 2.$$

Suppose that $m_1 d \geqslant 5\kappa$ *and* $m_2 d \geqslant 5\kappa$. *Then, the commit-and-prove system* Π_{quad} *for the relation* R_{quad} *has statistical completeness with correctness error* $1 - \frac{1}{M_1 M_2}$.

Proof. To begin with, we bound the norm of $c\mathbf{s}_1$ and $c\mathbf{s}_2$. Note that by Lemma 3.2.8 and the definition of $\mathcal{C}$ in (3.5) we have $\|c\mathbf{s}_1\| \leqslant \alpha\eta$ and $\|c\mathbf{s}_2\| \nu\eta\sqrt{m_2 d}$. Then, by Lemma 3.3.2, the probability that the two rejection sampling algorithms Rej_0 do not abort is at least $1/(M_1 M_2)$. Furthermore, by Lemma 3.2.2 for $t = \sqrt{2}$ and our assumption that $m_1 d, m_2 d \geqslant 5\kappa$, the probability that $\|\mathbf{z}_1\| \leqslant \mathfrak{s}_1\sqrt{2m_1 d}$ and $\|\mathbf{z}_2\| \leqslant \mathfrak{s}_2\sqrt{2m_2 d}$ is overwhelming. The other verification equations hold based on the discussion above. □

Theorem 5.2.3. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1\eta\alpha$ *and* $\mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d}$ *for some* $\gamma_1, \gamma_2 > 0$ *and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) \text{ for } i = 1, 2.$$

Suppose that $\kappa_{\mathsf{MLWE}} := m_2 - \kappa_{\mathsf{MSIS}} - \ell - 1 \geqslant 0$. *Then, the commit-and-prove system* Π_{quad} *for the relation* R_{quad} *is simulatable under the* $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}}, \kappa_{\mathsf{MSIS}}+\ell+1, \chi}$ *assumption.*

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

First, we define a hybrid simulator $\mathcal{S}_1$ which still knows secret information $\mathbf{s}_1, \mathbf{m}$. Given a challenge $c \leftarrow \mathcal{C}$, it honestly generates the commitment $(\mathbf{t}_A, \mathbf{t}_B, t)$ under randomness $\mathbf{s}_2 \leftarrow \chi^{m_2}$. Further, it samples fresh masked opening $\mathbf{z}_1 \leftarrow D_{\mathfrak{s}_1}^{m_1 d}$ and $\mathbf{z}_2 \leftarrow D_{\mathfrak{s}_2}^{m_2 d}$. Finally, it sets $\mathbf{w} := \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 - c\mathbf{t}_A$ and $v := \mathbf{z}^T\mathbf{R}_2\mathbf{z} + c\mathbf{r}_1^T\mathbf{z} + c^2 r_0 - ct + \mathbf{b}_{\mathsf{ext}}^T\mathbf{z}_2$. Then, by Lemma 3.3.2, the distribution of the commitment and a transcript output by $\mathcal{S}_0$ is statistically close to the one in the actual non-aborting protocol.

Further, we describe an efficient simulator $\mathcal{S}_2$, which still knows $\mathbf{s}_1, \mathbf{m}$ and simulates both the commitment and the transcript as follows. Namely,

it executes the $\mathcal{S}_1$ algorithm but instead of generating $(\mathbf{t}_A, \mathbf{t}_B)$ honestly, it samples $\mathbf{u} \leftarrow \mathcal{R}_q^{n+\ell+1}$ and computes:

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \\ t \end{bmatrix} := \mathbf{u} + \begin{bmatrix} \mathbf{A}_1\mathbf{s}_1 \\ \mathbf{m} \\ g_1 \end{bmatrix}. \tag{5.11}$$

Now, we observe that under the $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}},\kappa_{\mathsf{MSIS}}+\ell+1,\chi}$ assumption, the output distribution of $\mathcal{S}_2$ is computationally indistinguishable from the output distribution of $\mathcal{S}_1$.

Finally, we can simply set $\mathcal{S}$ (which does not use any secret information) to proceed identically as $\mathcal{S}_2$ but instead of defining $(\mathbf{t}_A, \mathbf{t}_B, t)$ as in (5.11), it directly samples $(\mathbf{t}_A, \mathbf{t}_B, t) \leftarrow \mathcal{R}_q^{n+\ell+1}$. Then, the output distributions of $\mathcal{S}$ and $\mathcal{S}_1$ are identical. Hence, the statement holds by the hybrid argument. □

Theorem 5.2.4. *Suppose $B_1 \geqslant 2\mathfrak{s}_1\sqrt{2m_1d}$ and $B_2 \geqslant 2\mathfrak{s}_2\sqrt{2m_2d}$. Then, the commit-and-prove system Π_{quad} for the relation R_{quad} is knowledge sound with knowledge error $2|\mathcal{C}|^{-1}$.*

Proof. Let $\mathcal{P}^*$ be a probabilistic prover which runs in time at most T and convinces the verifier with probability $\epsilon > 2|\mathcal{C}|^{-1}$. By Lemma 3.3.1, there is an algorithm $\mathcal{E}$ which runs in expected time at most $3T$ and extracts from $\mathcal{P}^*$ three accepting transcripts with pairwise distinct challenges with probability at least $\epsilon - 2/|\mathcal{C}|$:

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

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

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

By construction, we have $\bar{c} \in \bar{\mathcal{C}}$, $\|\bar{c}\bar{\mathbf{s}}_1\| \leqslant 2\mathfrak{s}_1\sqrt{2m_1d} \leqslant B_1$ and also $\|\bar{c}\bar{\mathbf{s}}_2\| \leqslant 2\mathfrak{s}_2\sqrt{2m_2d} \leqslant B_2$. Moreover, $\mathbf{A}_1\bar{\mathbf{s}}_1 + \mathbf{A}_2\bar{\mathbf{s}}_2 = \mathbf{t}_A$. Further, we define the extracted message vector

$$\bar{\mathbf{m}} := \mathbf{t}_B - \mathbf{B}\bar{\mathbf{s}}_2 \quad \text{and} \quad \bar{g}_1 := t - \mathbf{b}_{\mathsf{ext}}^T\bar{\mathbf{s}}_2.$$

Then, we have

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

Hence, we get $\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B) = 1$.

Next, let $\bar{\mathbf{y}}_i := \mathbf{z}_i^{(1)} - c^{(1)}\bar{\mathbf{s}}_i = \mathbf{z}_i^{(0)} - c^{(0)}\bar{\mathbf{s}}_i$ for $i = 1, 2$. Moreover, consider the third transcript $\mathsf{tr}^{(2)}$ and define $\mathbf{y}_i^{(2)} := \mathbf{z}_i^{(2)} - c^{(2)}\bar{\mathbf{s}}_i$ for $i = 1, 2$. We claim that $(\bar{\mathbf{y}}_1, \bar{\mathbf{y}}_2) = (\mathbf{y}_1^{(2)}, \mathbf{y}_2^{(2)})$ unless $\mathcal{E}$ breaks the binding property of ABDLOP. Indeed, note that:

$$\mathbf{A}_1 \left(\frac{\mathbf{z}_1^{(2)} - \mathbf{z}_1^{(1)}}{c^{(2)} - c^{(1)}} \right) + \mathbf{A}_2 \left(\frac{\mathbf{z}_2^{(2)} - \mathbf{z}_2^{(1)}}{c^{(2)} - c^{(1)}} \right) = \mathbf{t}_A.$$

Hence, unless

$$\frac{\mathbf{z}_1^{(2)} - \mathbf{z}_1^{(1)}}{c^{(2)} - c^{(1)}} = \bar{\mathbf{s}}_1 \quad \text{and} \quad \frac{\mathbf{z}_2^{(2)} - \mathbf{z}_2^{(1)}}{c^{(2)} - c^{(1)}} = \bar{\mathbf{s}}_2,$$

$\mathcal{E}$ finds two different openings to $(\mathbf{t}_A, \mathbf{t}_B)$. Assume this is not the case. Then, we get

$$\mathbf{z}_1^{(2)} - \mathbf{z}_1^{(1)} = (c^{(2)} - c^{(1)})\bar{\mathbf{s}}_1.$$

Note that the term on the left-hand side can be expanded as:

$$\mathbf{z}_1^{(2)} - \mathbf{z}_1^{(1)} = \mathbf{y}_1^{(2)} - \bar{\mathbf{y}}_1 + (c^{(2)} - c^{(1)})\bar{\mathbf{s}}_1.$$

Thus, we conclude that $\mathbf{y}_1^{(2)} - \bar{\mathbf{y}}_1 = 0$. Similarly, we deduce that $\mathbf{y}_2^{(2)} = \bar{\mathbf{y}}_2$.

Finally, let us define the following vectors:

$$\bar{\mathbf{s}} := \begin{bmatrix} \langle \bar{\mathbf{s}}_1 \rangle_\sigma \\ \langle \bar{\mathbf{m}} \rangle_\sigma \end{bmatrix} \text{ and } \bar{\mathbf{y}} := \begin{bmatrix} \langle \bar{\mathbf{y}}_1 \rangle_\sigma \\ -\langle \mathbf{B}\bar{\mathbf{y}}_2 \rangle_\sigma \end{bmatrix}.$$

Then, from the verification equations we have that for $i = 0, 1, 2$:

$$\mathbf{z}^{(i)^T} \mathbf{R}_2 \mathbf{z}^{(i)} + c^{(i)} \mathbf{r}_1^T \mathbf{z}^{(i)} + c^{(i)^2} r_0 - \left(c^{(i)} t - \mathbf{b}_{\mathsf{ext}}^T \mathbf{z}_2^{(i)} \right) = v \tag{5.12}$$

where

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

By expanding Equation 5.12, we obtain

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

where

$$g_1' = \bar{\mathbf{s}}^T \mathbf{R}_2 \bar{\mathbf{y}} + \bar{\mathbf{y}}^T \mathbf{R}_2 \bar{\mathbf{s}} + \mathbf{r}_1^T \bar{\mathbf{y}} - \bar{g}_1$$
$$g_0' = \bar{\mathbf{y}}^T \mathbf{R}_2 \bar{\mathbf{y}} + \mathbf{b}_{\mathsf{ext}}^T \bar{\mathbf{y}}_2 - v.$$

Alternatively, we can write these three equations as follows:

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

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

5.2.2 *Many Quadratic Equations with Automorphisms*

We consider a scenario when the prover wants to simultaneously prove N quadratic relations. Clearly, if one were to prove them separately using the approach from Section 5.2.1, one would end up committing to N garbage polynomials g. Here, we circumvent this issue by linear-combining the N equations into one quadratic equation and prove it using the protocol in Figure 5.3. This results in committing to only one garbage polynomials at the cost of reducing the soundness error by a negligible additive factor.

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

$$f_j(\mathbf{s}) = 0 \text{ for } j \in [N] \text{ where } \mathbf{s} := \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma. \tag{5.13}$$

As before, we can write each function f_j as $f_j(\mathbf{x}) := \mathbf{x}^T \mathbf{R}_{j,2} \mathbf{x} + \mathbf{r}_{j,1} \mathbf{x} + r_{j,0}$. We define the corresponding relation $R_{\mathsf{quad-many}}$ as:

$$\left\{ \begin{array}{c} \Big((\mathbf{R}_{i,2}, \mathbf{r}_{i,1}, r_{i,0})_{i \in [N]}, (\mathbf{s}_1, \mathbf{m}) \Big) : \\ \mathbf{s}^T \mathbf{R}_{i,2} \mathbf{s} + \mathbf{r}_{i,1}^T \mathbf{s} + r_{i,0} = 0 \text{ for } i \in [n] \text{ where } \mathbf{s} := (\langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma) \end{array} \right\}.$$

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

$$f(\mathbf{x}) := \sum_{i=1}^{N} \mu_j f_j(\mathbf{x}) = \mathbf{x}^T \left(\sum_{i=1}^{N} \mu_j \mathbf{R}_{j,2} \right) \mathbf{x} + \left(\sum_{i=1}^{N} \mu_j \mathbf{r}_{j,1}^T \right) \mathbf{x} + \sum_{i=1}^{N} \mu_j r_0$$

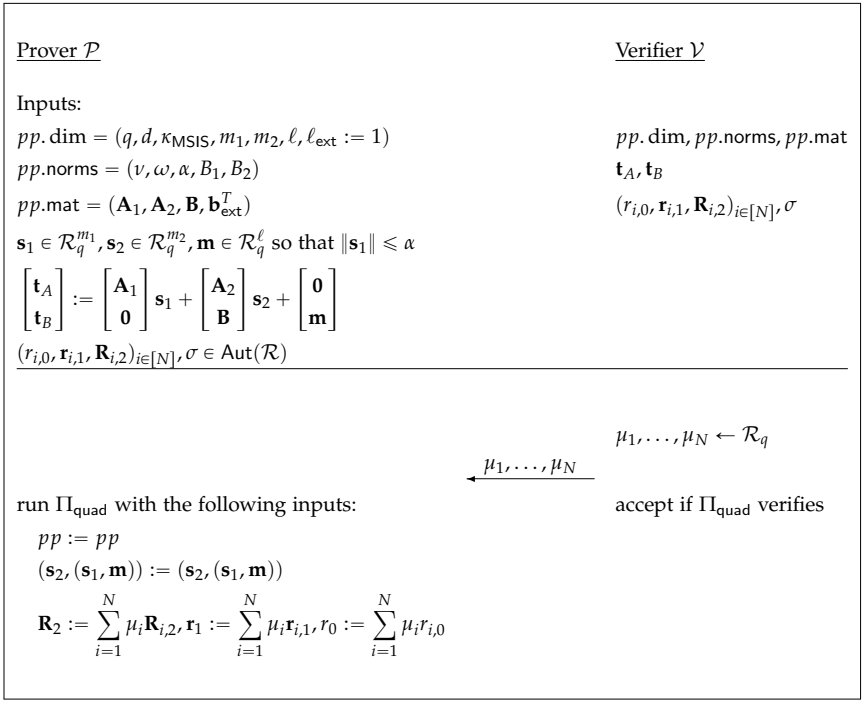

FIGURE 5.4: Commit-and-prove system $\Pi_{\mathsf{quad-many}}$ for proving $\mathbf{s}^T \mathbf{R}_{i,2} \mathbf{s} + \mathbf{r}_{i,1}^T \mathbf{s} + r_{i,0} = 0$ for $i = 1, \ldots, N$ where $\mathbf{s} := \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma$.

and prove that

$$f(\mathbf{s}) = 0 \tag{5.14}$$

using the protocol from Figure 5.3. Now, we observe that if one of the conditions in (5.13) does not hold, then Equation 5.14 is satisfied with probability at most $q_1^{-d/l}$[3].

We present the commit-and-prove system $\Pi_{\mathsf{quad-many}} = (\mathsf{ABDLOP}, \mathcal{P}, \mathcal{V})$ for the relation $R_{\mathsf{quad-many}}$ in Figure 5.4. Since correctness and simulatability will be implicitly included in the more general case in Section 5.2.3, we only focus on knowledge soundness.

Theorem 5.2.5. *Suppose $B_1 \geqslant 2\mathfrak{s}_1\sqrt{2m_1d}$ and $B_2 \geqslant 2\mathfrak{s}_2\sqrt{2m_2d}$. Then, the commit-and-prove system $\Pi_{\mathsf{quad-many}}$ for the relation $R_{\mathsf{quad-many}}$ is knowledge sound with knowledge error $2|\mathcal{C}|^{-1} + q_1^{-d/l}$.*

3 Recall that l is the number of factors that $X^d + 1$ splits into modulo q.

Proof. Let $\mathcal{P}^*$ be a probabilistic prover which convinces the verifier with probability $\varepsilon > 2|\mathcal{C}|^{-1} + q_1^{-d/l}$ and runs in time at most T. We define a deterministic algorithm $\mathcal{A}(\rho, \boldsymbol{\mu})$ which given randomness $\rho \in \mathfrak{R}$ and a challenge $\boldsymbol{\mu} \in \mathcal{R}_q^N$, it does the following. It simply runs the extractor $\mathcal{E}^*(\rho)$ from the proof of Theorem 5.2.4 with randomness ρ which then calls $\mathcal{P}^*(\boldsymbol{\mu})$ in a black-box way.

We say that $\mathcal{A}$ succeeds if $\mathcal{A}$ outputs $(\boldsymbol{\mu}, \bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c})$ such that

$$\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B) = 1$$

and

$$\bar{\mathbf{s}}^T \left(\sum_{i=1}^{N} \mu_j \mathbf{R}_{j,2} \right) \bar{\mathbf{s}} + \left(\sum_{i=1}^{N} \mu_j \mathbf{r}_{j,1}^T \right) \bar{\mathbf{s}} + \sum_{i=1}^{N} \mu_j r_0 = 0$$

where $\bar{\mathbf{s}} := \langle \bar{\mathbf{s}}_1 \parallel \bar{\mathbf{m}} \rangle$. Note that $\mathcal{E}^*$ could also extract two different openings of $(\mathbf{t}_A, \mathbf{t}_B)$. But then, $\mathcal{A}$ would break the binding property of ABDLOP. For presentation, we will assume this never occurs.

From Theorem 5.2.4 we know that the expected runtime of $\mathcal{A}$ for any $\boldsymbol{\mu}$ and $\rho \leftarrow \mathfrak{R}$ is at most $3T$ and the probability that $\mathcal{A}$ succeeds for random ρ and $\boldsymbol{\mu}$ is at least $\epsilon - 2/|\mathcal{C}|$.

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

$$\bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})} := \left\langle \bar{\mathbf{s}}_1^{(\rho,\boldsymbol{\mu})} \parallel \bar{\mathbf{m}}^{(\rho,\boldsymbol{\mu})} \right\rangle_\sigma .$$

Finally, we define

$$H' := \left\{ (\rho, \boldsymbol{\mu}) \in H : \exists j \in [N], \quad \bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})^T} \mathbf{R}_{j,2} \bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})} + \mathbf{r}_{j,1}^T \bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})} + r_{j,0} \neq 0 \right\}.$$

Then, we have the following claim.

Claim 5.2.6. If $(\rho, \boldsymbol{\mu}) \in H$ then

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

Moreover, if $(\rho, \boldsymbol{\mu}) \in H'$ then

$$\Pr_{\boldsymbol{\mu}' \leftarrow \mathcal{R}_q^N} \left[\bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})^T} \left(\sum_{i=1}^{N} \mu_j' \mathbf{R}_{j,2} \right) \bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})} + \left(\sum_{i=1}^{N} \mu_j' \mathbf{r}_{j,1}^T \right) \bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})} + \sum_{i=1}^{N} \mu_j' r_0 = 0 \right]$$

is at most $q_1^{-d/l}$.

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

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

Now, if $\bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})^T} \mathbf{R}_{j,2} \bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})} + \mathbf{r}_{j,1}^T \bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})} + r_{j,0} \neq 0$ for some j, then with probability at most $q_1^{-d/l}$ we have

$$\sum_{i=1}^{N} \mu_i' \left(\bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})^T} \mathbf{R}_{i,2} \bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})} + \mathbf{r}_{i,1}^T \bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})} + r_{i,0} \right) = 0.$$

Hence, the claim follows. □

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

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

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

- $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2) \neq (\bar{\mathbf{s}}_1', \bar{\mathbf{s}}_2')$ and

$$\begin{aligned} 1 &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B) \\ &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{s}}_2', \bar{c}'; \mathbf{t}_A \parallel \mathbf{t}_B). \end{aligned}$$

- $\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B) = 1$ and for all $i \in [N]$, $\bar{\mathbf{s}}^T \mathbf{R}_{i,2} \bar{\mathbf{s}} + \mathbf{r}_{i,1}^T \bar{\mathbf{s}} + r_{i,0} = 0$ where $\bar{\mathbf{s}} := \langle \bar{\mathbf{s}}_1 \parallel \bar{\mathbf{m}} \rangle$.

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

Claim 5.2.7. The expected number of calls to $\mathcal{A}$ is at most 2.

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

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

□

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

Claim 5.2.8. Probability that $\mathcal{E}$ succeeds is at least $\epsilon - 2/|\mathcal{C}| - q_1^{-d/l}$.

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

Case 1. $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2) \neq (\bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{s}}_2')$ and

$$\sum_{i=1}^{N} \mu_i \left(\bar{\mathbf{s}}^T \mathbf{R}_{i,2} \bar{\mathbf{s}} + \mathbf{r}_{i,1}^T \bar{\mathbf{s}} + r_{i,0} \right) = 0 \quad \text{and} \quad \sum_{i=1}^{N} \mu_i' \left(\bar{\mathbf{s}}'^T \mathbf{R}_{i,2} \bar{\mathbf{s}}' + \mathbf{r}_{i,1}^T \bar{\mathbf{s}}' + r_{i,0} \right) = 0$$

and

$$\begin{aligned} 1 &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B) \\ &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{s}}_2', \bar{c}'; \mathbf{t}_A \parallel \mathbf{t}_B). \end{aligned}$$

Case 2. $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2) = (\bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{s}}_2')$ and $\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B) = 1$ and

$$\sum_{i=1}^{N} \mu_i \left(\bar{\mathbf{s}}^T \mathbf{R}_{i,2} \bar{\mathbf{s}} + \mathbf{r}_{i,1}^T \bar{\mathbf{s}} + r_{i,0} \right) = 0 \quad \text{and} \quad \sum_{i=1}^{N} \mu_i' \left(\bar{\mathbf{s}}^T \mathbf{R}_{i,2} \bar{\mathbf{s}} + \mathbf{r}_{i,1}^T \bar{\mathbf{s}} + r_{i,0} \right) = 0$$

and for all $i \in [N]$, $\bar{\mathbf{s}}^T \mathbf{R}_{i,2} \bar{\mathbf{s}} + \mathbf{r}_{i,1}^T \bar{\mathbf{s}} + r_{i,0} = 0$.

Case 3. $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2) = (\bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{s}}_2')$ and $\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B) = 1$ and

$$\sum_{i=1}^{N} \mu_i \left(\bar{\mathbf{s}}^T \mathbf{R}_{i,2} \bar{\mathbf{s}} + \mathbf{r}_{i,1}^T \bar{\mathbf{s}} + r_{i,0} \right) = 0 \quad \text{and} \quad \sum_{i=1}^{N} \mu_i' \left(\bar{\mathbf{s}}^T \mathbf{R}_{i,2} \bar{\mathbf{s}} + \mathbf{r}_{i,1}^T \bar{\mathbf{s}} + r_{i,0} \right) = 0$$

and there exists $i \in [N]$, such that $\bar{\mathbf{s}}^T \mathbf{R}_{i,2} \bar{\mathbf{s}} + \mathbf{r}_{i,1}^T \bar{\mathbf{s}} + r_{i,0} \neq 0$.

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

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

and

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

Hence, we only need to upper-bound the probability $\Pr[E_3]$. We apply Claim 5.2.6 as follows:

$$\begin{aligned}
\Pr[E_3] &\leqslant \Pr\left[\begin{array}{c}(\mathcal{A}(\rho,\boldsymbol{\mu}) \text{ succeeds}) \wedge \left(\sum_{i=1}^{N} \mu'_i \left(\bar{\mathbf{s}}^T \mathbf{R}_{i,2} \bar{\mathbf{s}} + \mathbf{r}_{i,1}^T \bar{\mathbf{s}} + r_{i,0}\right) = 0\right) \\ \wedge \left(\exists i \in [N] : \bar{\mathbf{s}}^T \mathbf{R}_{i,2} \bar{\mathbf{s}} + \mathbf{r}_{i,1}^T \bar{\mathbf{s}} + r_{i,0} \neq 0\right)\end{array}\right] \\
&\leqslant \frac{1}{|\mathfrak{R}| \cdot q^{Nd}} \sum_{(\rho,\boldsymbol{\mu}) \in H'} \Pr_{\boldsymbol{\mu}'}\left[\sum_{i=1}^{N} \mu'_i \left(\bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})^T} \mathbf{R}_{i,2} \bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})} + \mathbf{r}_{i,1}^T \bar{\mathbf{s}}^{(\rho,\boldsymbol{\mu})} + r_{i,0}\right) = 0\right] \\
&\leqslant \frac{1}{|\mathfrak{R}| \cdot q^{Nd}} \sum_{(\rho,\boldsymbol{\mu}) \in H'} q_1^{-d/l} \\
&\leqslant \frac{1}{|\mathfrak{R}| \cdot q^{Nd}} \sum_{(\rho,\boldsymbol{\mu}) \in \mathfrak{R} \times \mathcal{R}_q^N} q_1^{-d/l} \leqslant q_1^{-d/l}.
\end{aligned}$$

□

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

5.2.3 *Polynomial Evaluations with Vanishing Constant Coefficients*

Suppose we want to prove simultaneously N quadratic relations (i.e. (5.13)) and *additionally* prove that for quadratic $k(m_1 + \ell)$-variate polynomials $F_1, \ldots, F_M$, evaluations $F_j(\langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma)$ have the constant coefficient equal to zero. Concretely, denote the quadratic function

$$F_j(\mathbf{x}) := \mathbf{x}^T \mathbf{R}'_{j,2} \mathbf{x} + \mathbf{r}'^T_{j,1} \mathbf{x} + r'_{j,0} \text{ for } j \in [M].$$

We define the correspond relation

$$R_{\mathsf{quad-eval}} = \left\{ \begin{array}{c} \left(\left((\mathbf{R}_{i,2}, \mathbf{r}_{i,1}, r_{i,0})_{i\in[N]}, (\mathbf{R}'_{j,2}, \mathbf{r}'_{j,1}, r'_{j,0})_{j\in[M]}\right), (\mathbf{s}_1, \mathbf{m})\right) : \\ \forall i \in [N], \mathbf{s}^T \mathbf{R}_{i,2} \mathbf{s} + \mathbf{r}_{i,1}^T \mathbf{s} + r_{i,0} = 0 \text{ and } \forall j \in [M], \tilde{x}_j = 0 \\ \text{where } \mathbf{s} := (\langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma) \text{ and } x_j := \mathbf{s}^T \mathbf{R}'_{j,2} \mathbf{s} + \mathbf{r}'^T_{j,1} \mathbf{s} + r'_{j,0} \end{array} \right\}. \tag{5.15}$$

For simplicity we first present an approach with soundness error $1/q_1$. We apply the strategy from Section 5.1.3 and first commit to a random masking polynomial $g \leftarrow \{x \in \mathcal{R}_q : \tilde{x} = 0\}$. Then, given random challenges $v_1, \ldots, v_M \leftarrow \mathbb{Z}_q$, we send

$$h := g + \sum_{j=1}^{M} v_j \left(\mathbf{s}^T \mathbf{R}'_{j,2} \mathbf{s} + \mathbf{r}'^T_{j,1} \mathbf{s} + r'_{j,0}\right) \tag{5.16}$$

to the verifier where $\mathbf{s} := \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma$ as before. Then, it simply checks whether the constant coefficient of h is indeed equal to zero. What is left to prove is that h is well-formed, i.e. (5.16) holds. Clearly, Equation 5.16 is a quadratic relation in the committed messages. Indeed, note that it is equivalent to:

$$\langle \mathbf{s}_1 \parallel \mathbf{m} \parallel g \rangle_\sigma^T \hat{\mathbf{R}}_{N+1,2} \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel g \rangle_\sigma + \hat{\mathbf{r}}_{N+1,1}^T \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel g \rangle_\sigma + \hat{r}_{N+1,0} = 0$$

where

$$\hat{\mathbf{R}}_{N+1,2} := \begin{bmatrix} \sum_{j=1}^M v_j \mathbf{R}'_{j,2} & \mathbf{0}_{k(m_1+\ell)\times k} \\ \mathbf{0}_{k\times k(m_1+\ell)} & \mathbf{0}_{k\times k} \end{bmatrix} \in \mathcal{R}_q^{k(m_1+\ell+1)\times k(m_1+\ell+1)}$$

$$\hat{\mathbf{r}}_{N+1,1}^T := \begin{bmatrix} \sum_{j=1}^M v_j \mathbf{r}'^T_{j,1} & 1 & \mathbf{0}_{1\times(k-1)} \end{bmatrix} \in \mathcal{R}_q^{k(m_1+\ell+1)}$$

$$\hat{r}_{N+1,0} := \sum_{j=1}^M v_j r'_{j,0} - h \in \mathcal{R}_q.$$

We provide intuition for the soundness argument. Assume that the verifier is convinced that h is of the correct form (5.16) and $\widetilde{h} = 0$. Also, note that a cheating prover committed to g before seeing the challenges $v_1, \ldots, v_M$. Hence, if for some $j \in [M]$, the constant coefficient of $\mathbf{s}^T \mathbf{R}'_{j,2} \mathbf{s} + \mathbf{r}'^T_{j,1} \mathbf{s} + r'_{j,0}$ is non-zero, then the cheating prover has probability at most $1/q_1$ of guessing the constant coefficient of $\sum_{j=1}^M v_j \left(\mathbf{s}^T \mathbf{R}'_{j,2} \mathbf{s} + \mathbf{r}'^T_{j,1} \mathbf{s} + r'_{j,0} \right)$.

Recall that we also need to prove (5.13), i.e.

$$\langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma^T \mathbf{R}_{i,2} \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma + \mathbf{r}_{i,1}^T \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma + r_{i,0} = 0 \tag{5.17}$$

for $i = 1, \ldots, N$. Note that each such quadratic equation can be equivalently written as:

$$\langle \mathbf{s}_1 \parallel \mathbf{m} \parallel g \rangle_\sigma^T \hat{\mathbf{R}}_{i,2} \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel g \rangle_\sigma + \hat{\mathbf{r}}_{i,1}^T \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel g \rangle_\sigma + \hat{r}_{i,0} = 0 \tag{5.18}$$

where

$$\hat{\mathbf{R}}_{i,2} := \begin{bmatrix} \mathbf{R}_{i,2} & \mathbf{0}_{k(m_1+\ell)\times k} \\ \mathbf{0}_{k\times k(m_1+\ell)} & \mathbf{0}_{k\times k} \end{bmatrix} \in \mathcal{R}_q^{k(m_1+\ell+1)\times k(m_1+\ell+1)}$$

$$\hat{\mathbf{r}}_{i,1}^T := \begin{bmatrix} \mathbf{r}_{i,1}^T & \mathbf{0}_{1\times k} \end{bmatrix} \in \mathcal{R}_q^{k(m_1+\ell+1)}$$

$$\hat{r}_{i,0} := r_{i,0} \in \mathcal{R}_q.$$

Hence, we end up with proving $N+1$ quadratic equations of the form (5.18) for $i = 1, \ldots, N+1$ and can thus apply the protocol $\Pi_{\mathsf{quad-many}}$.

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

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

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

$$\begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_\lambda \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_\lambda \end{bmatrix} + \begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,M} \\ \vdots & \vdots & \cdots & \vdots \\ v_{\lambda,1} & v_{\lambda,2} & \cdots & v_{\lambda,M} \end{bmatrix} \begin{bmatrix} \mathbf{s}^T\mathbf{R}'_{1,2}\mathbf{s} + \mathbf{r}'^T_{1,1}\mathbf{s} + r'_{1,0} \\ \mathbf{s}^T\mathbf{R}'_{2,2}\mathbf{s} + \mathbf{r}'^T_{2,1}\mathbf{s} + r'_{2,0} \\ \vdots \\ \mathbf{s}^T\mathbf{R}'_{M,2}\mathbf{s} + \mathbf{r}'^T_{M,1}\mathbf{s} + r'_{M,0} \end{bmatrix} \tag{5.19}$$

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

As before, we still need to prove that vector $\mathbf{h}$ was constructed correctly. We reduce this problem to proving quadratic relations. Let us fix $i \in [\lambda]$. Then,

$$h_i := g_i + \sum_{j=1}^{M} v_{i,j}\left(\mathbf{s}^T\mathbf{R}'_{j,2}\mathbf{s} + \mathbf{r}'^T_{j,1}\mathbf{s} + r'_{j,0}\right)$$

is equivalent to

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

where

$$\begin{aligned} \hat{\mathbf{R}}_{N+i,2} &:= \begin{bmatrix} \sum_{j=1}^M v_{i,j}\mathbf{R}'_{j,2} & \mathbf{0}_{k(m_1+\ell)\times k\lambda} \\ \mathbf{0}_{k\lambda\times k(m_1+\ell)} & \mathbf{0}_{k\lambda\times k\lambda} \end{bmatrix} \in \mathcal{R}_q^{k(m_1+\ell+\lambda)\times k(m_1+\ell+\lambda)} \\ \hat{\mathbf{r}}_{N+i,1}^T &:= \begin{bmatrix} \sum_{j=1}^M v_{i,j}\mathbf{r}'^T_{j,1} & \mathbf{e}_i^T \end{bmatrix} \in \mathcal{R}_q^{k(m_1+\ell+\lambda)} \\ \hat{r}_{N+i,0} &:= \sum_{j=1}^{M} v_{i,j}r'_{j,0} - h_i \in \mathcal{R}_q. \end{aligned} \tag{5.20}$$

and $\mathbf{e}_i^T = \begin{bmatrix} \mathbf{0}_{1\times k(i-1)} & 1 & \mathbf{0}_{1\times k(\lambda-i+1)-1} \end{bmatrix} \in \mathcal{R}_q^{k\lambda}$ is the binary vector which has exactly one 1 in the $(k(i-1)+1)$-th position.

Further, we need to prove (5.13), or alternatively (5.17) for $i = 1, \ldots, N$. Similarly as before, Equation 5.17 can be written equivalently as

$$\langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{g} \rangle_\sigma^T \hat{\mathbf{R}}_{i,2} \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{g} \rangle_\sigma + \hat{\mathbf{r}}_{i,1}^T \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{g} \rangle_\sigma + \hat{r}_{i,0} = 0 \tag{5.21}$$

where $\hat{\mathbf{R}}_{i,2}, \hat{\mathbf{r}}_{i,1}^T, \hat{r}_{i,0}$ are defined as follows

$$\begin{aligned} \hat{\mathbf{R}}_{i,2} &:= \begin{bmatrix} \mathbf{R}_{i,2} & \mathbf{0}_{k(m_1+\ell)\times k\lambda} \\ \mathbf{0}_{k\lambda \times k(m_1+\ell)} & \mathbf{0}_{k\lambda \times k\lambda} \end{bmatrix} \in \mathcal{R}_q^{k(m_1+\ell+\lambda)\times k(m_1+\ell+\lambda)} \\ \hat{\mathbf{r}}_{i,1}^T &:= \begin{bmatrix} \mathbf{r}_{i,1}^T & \mathbf{0}_{1\times k\lambda} \end{bmatrix} \in \mathcal{R}_q^{k(m_1+\ell+\lambda)} \\ \hat{r}_{i,0} &:= r_{i,0} \in \mathcal{R}_q. \end{aligned} \tag{5.22}$$

Hence, we reduce the problem to proving $N + \lambda$ quadratic equations in $\langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{g} \rangle_\sigma$ and can thus run the commit-and-prove system $\Pi_{\mathsf{quad-many}}$.

We present the commit-and-prove system $\Pi_{\mathsf{quad-eval}} = (\mathsf{ABDLOP}, \mathcal{P}, \mathcal{V})$ for the relation $R_{\mathsf{quad-eval}}$ in Figure 5.5.

5.2.3.1 *Security Analysis*

We summarise security properties of the protocol in Figure 5.5 below.

Theorem 5.2.9. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1 \eta \alpha$ *and* $\mathfrak{s}_2 = \gamma_2 \eta \nu \sqrt{m_2 d}$ *for some* $\gamma_1, \gamma_2 > 0$ *and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2} \right) \text{ for } i = 1, 2.$$

Suppose that $m_1 d \geqslant 5\kappa$ *and* $m_2 d \geqslant 5\kappa$. *Then, the commit-and-prove system* $\Pi_{\mathsf{quad-eval}}$ *for the relation* $R_{\mathsf{quad-eval}}$ *has statistical completeness with correctness error* $1 - \frac{1}{M_1 M_2}$.

Proof. Take any $i \in [\lambda]$. Then, if the constant coefficients of g_i and $\mathbf{s}^T \mathbf{R}'_{j,2} \mathbf{s} + \mathbf{r}'^T_{j,1} \mathbf{s} + r'_{j,0}$ are all zeroes for $j \in [M]$ and each $v_{i,j}$ is an integer, then we must have that the constant coefficient of h_i also zero. The rest of the correctness argument follows from Theorem 5.2.2. □

Theorem 5.2.10. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1 \eta \alpha$ *and* $\mathfrak{s}_2 = \gamma_2 \eta \nu \sqrt{m_2 d}$ *for some* $\gamma_1, \gamma_2 > 0$ *and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2} \right) \text{ for } i = 1, 2.$$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
Inputs:		
$pp.\dim = (q, d, \kappa_{\mathsf{MSIS}}, m_1, m_2, \ell, \ell_{\mathsf{ext}} := \lambda + 1)$		$pp.\dim, pp.\mathsf{norms}, pp.\mathsf{mat}$
$pp.\mathsf{norms} = (\nu, \omega, \alpha, B_1, B_2)$		$\mathbf{t}_A, \mathbf{t}_B, \sigma$
$pp.\mathsf{mat} = \left(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \begin{bmatrix}\mathbf{B}_{\mathsf{ext}} \\ \mathbf{b}_{\mathsf{ext}}^T\end{bmatrix}\right)$		$(r_{i,0}, \mathbf{r}_{i,1}, \mathbf{R}_{i,2})_{i\in[N]}$
$\mathbf{s}_1 \in \mathcal{R}_q^{m_1}, \mathbf{s}_2 \in \mathcal{R}_q^{m_2}, \mathbf{m} \in \mathcal{R}_q^{\ell}$ so that $\|\mathbf{s}_1\| \leqslant \alpha$		$(r'_{i,0}, \mathbf{r}'_{i,1}, \mathbf{R}'_{i,2})_{i\in[M]}$
$\begin{bmatrix}\mathbf{t}_A \\ \mathbf{t}_B\end{bmatrix} := \begin{bmatrix}\mathbf{A}_1 \\ \mathbf{0}\end{bmatrix}\mathbf{s}_1 + \begin{bmatrix}\mathbf{A}_2 \\ \mathbf{B}\end{bmatrix}\mathbf{s}_2 + \begin{bmatrix}\mathbf{0} \\ \mathbf{m}\end{bmatrix}$		
$(r_{i,0}, \mathbf{r}_{i,1}, \mathbf{R}_{i,2})_{i\in[N]}, (r'_{i,0}, \mathbf{r}'_{i,1}, \mathbf{R}'_{i,2})_{i\in[M]}$		
$\sigma \in \mathsf{Aut}(\mathcal{R})$		
$\mathbf{s} := \begin{bmatrix}\langle \mathbf{s}_1 \rangle_\sigma \\ \langle \mathbf{m} \rangle_\sigma\end{bmatrix}$		
$\mathbf{g} := (g_1, \ldots, g_\lambda) \leftarrow \{x : \mathcal{R}_q : \bar{x} = 0\}^\lambda$		
$\mathbf{t}_g := \mathbf{B}_{\mathsf{ext}}\mathbf{s}_2 + \mathbf{g}$		
	$\xrightarrow{\mathbf{t}_g}$	
		$(v_{i,j}) \leftarrow \mathbb{Z}_q^{\lambda\times M}$
	$\xleftarrow{(v_{i,j})_{i\in[\lambda],j\in[M]}}$	
for $i \in [\lambda]$:		
$h_i := g_i + \sum_{j=1}^{M} v_{i,j}\left(\mathbf{s}^T\mathbf{R}'_{j,2}\mathbf{s} + \mathbf{r}'^T_{j,1}\mathbf{s} + r'_{j,0}\right)$		
	$\xrightarrow{h_1, \ldots, h_\lambda}$	
run $\Pi_{\mathsf{quad-many}}$ with the following inputs:		accept if:
$pp.\dim := (q, d, \kappa_{\mathsf{MSIS}}, m_1, m_2, \ell + \lambda, 1), pp.\mathsf{norms} := pp.\mathsf{norms}$		(i) $\Pi_{\mathsf{quad-many}}$ verifies
$pp.\mathsf{mat} := \left(\mathbf{A}_1, \mathbf{A}_2, \begin{bmatrix}\mathbf{B} \\ \mathbf{B}_{\mathsf{ext}}\end{bmatrix}, \mathbf{b}_{\mathsf{ext}}^T\right)$		(ii) $\tilde{h}_1 = \ldots = \tilde{h}_\lambda = 0$
$(\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m})) := (\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m} \parallel \mathbf{g}))$		
for $i \in [N]$:		
$\mathbf{R}_{i,2} := \begin{bmatrix}\mathbf{R}_{i,2} & \mathbf{0}_{k(m_1+\ell)\times k\lambda} \\ \mathbf{0}_{k\lambda\times k(m_1+\ell)} & \mathbf{0}_{k\lambda\times k\lambda}\end{bmatrix}, \mathbf{r}_{i,1} := \begin{bmatrix}\mathbf{r}_{i,1} \\ \mathbf{0}_{k\lambda\times 1}\end{bmatrix}$		
$r_{i,0} := r_{i,0}$		
for $i \in [\lambda]$:		
$\mathbf{R}_{N+i,2} := \begin{bmatrix}\sum_{j=1}^{M} v_{i,j}\mathbf{R}'_{j,2} & \mathbf{0}_{k(m_1+\ell)\times k\lambda} \\ \mathbf{0}_{k\lambda\times k(m_1+\ell)} & \mathbf{0}_{k\lambda\times k\lambda}\end{bmatrix}$		
$\mathbf{r}_{N+i,1} := \begin{bmatrix}\sum_{j=1}^{M} v_{i,j}\mathbf{r}'_{j,1} \\ \mathbf{e}_i\end{bmatrix}, r_{N+i,0} := \sum_{j=1}^{M} v_{i,j}r'_{j,0} - h_i$		

FIGURE 5.5: Commit-and-prove system $\Pi_{\mathsf{quad-eval}}$ for the relation $R_{\mathsf{quad-eval}}$. Here, $\mathbf{e}_i \in \mathcal{R}_q^{k\lambda}$ is the binary vector which has exactly one 1 in the $(k(i-1)+1)$-th position.

Suppose $\kappa_{\mathsf{MLWE}} := m_2 - \kappa_{\mathsf{MSIS}} - \ell - \lambda - 1 \geqslant 0$. *Then, the commit-and-prove system* Π_{quad} *for relation* R_{quad} *is simulatable under the* $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}},\kappa_{\mathsf{MSIS}}+\ell+\lambda+1,\chi}$ *assumption.*

Proof. The proof is almost identical to the one for Theorem 5.2.3 with the addition that the simulator $\mathcal{S}$ samples $\mathbf{t}_g \leftarrow \mathcal{R}_q^\lambda$ and $\mathbf{h} \leftarrow \{x \in \mathcal{R}_q : \tilde{x} = 0\}^\lambda$. Indeed, note that since in an honest execution $\mathbf{t}_g$ is chosen uniformly at random from $\{x \in \mathcal{R}_q : \tilde{x} = 0\}^\lambda$, the distribution of the vector $\mathbf{h}$ constructed as in Equation 5.19 is still uniformly random over $\{x \in \mathcal{R}_q : \tilde{x} = 0\}^\lambda$. □

Theorem 5.2.11. *Suppose* $B_1 \geqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ *and* $B_2 \geqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$. *Then, the commit-and-prove system* $\Pi_{\mathsf{quad-eval}}$ *for the relation* $R_{\mathsf{quad-eval}}$ *is knowledge sound with knowledge error* $2|\mathcal{C}|^{-1} + q_1^{-d/l} + q_1^{-\lambda}$.

Proof. Let $\mathcal{P}^*$ be a probabilistic prover which runs in time at most T and convinces the verifier with probability $\epsilon > 2|\mathcal{C}|^{-1} + q_1^{-d/l} + q_1^{-\lambda}$. Define a deterministic algorithm $\mathcal{A}(\rho_P, \rho_E, \mathrm{Y})$ which given randomness $\rho = (\rho_P, \rho_E) \in \mathfrak{R}_P \times \mathfrak{R}_E$ and challenge $\mathrm{Y} \in \mathbb{Z}_q^{\lambda \times M}$ does the following. It first runs $\mathcal{P}^*(\rho_P)$ on randomness ρ_P with challenge Y and stops after the third round. Let $\mathbf{t}_g$ and $\mathbf{h}$ be the output of $\mathcal{P}^*$ in the first and third round respectively. Then, it runs the extractor $\mathcal{E}^*(\rho_E)$ defined in the proof of Theorem 5.2.5 with randomness ρ_E (which runs $\mathcal{P}^*(\rho_P, \mathrm{Y})$ in a black-box way).

We say that $\mathcal{A}$ succeeds if $\mathcal{A}$ outputs $(\mathbf{t}_g, \mathrm{Y}, \mathbf{h}, \bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{g}}, \bar{\mathbf{s}}_2, \bar{c})$ such that $\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{\mathbf{g}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B \parallel \mathbf{t}_g) = 1$ and $\tilde{h}_1 = \ldots = \tilde{h}_\lambda = 0$ and for all i,

$$h_i = \bar{g}_i + \sum_{j=1}^{M} v_{i,j} \left(\bar{\mathbf{s}}^T \mathbf{R}'_{j,2} \bar{\mathbf{s}} + \mathbf{r}'^T_{j,1} \bar{\mathbf{s}} + r'_{j,0} \right)$$

and

$$\bar{\mathbf{s}}^T \mathbf{R}_{j,2} \bar{\mathbf{s}} + \mathbf{r}_{j,1}^T \bar{\mathbf{s}} + r_{j,0} = 0 \text{ for } j \in [N]$$

where $\bar{\mathbf{s}} = \langle \bar{\mathbf{s}}_1 \parallel \bar{\mathbf{m}} \rangle_\sigma$. As before, we assume that $\mathcal{E}^*$ does not break the binding property of ABDLOP since if it did, then so does $\mathcal{A}$ (and later on $\mathcal{E}$). Clearly, by Theorem 5.2.5, the probability that $\mathcal{A}$ succeeds for random ρ and Y is at least $\epsilon - 2/|\mathcal{C}| - q_1^{-d/l}$. Moreover, the expected runtime $\mathcal{A}(\rho_P, \rho_E, \mathrm{Y})$ for any fixed ρ_P, Y and $\rho_E \leftarrow \mathfrak{R}_E$ is at most $6T$.

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

to be the $\bar{\mathbf{s}}_1$ part of the output of $\mathcal{A}(\rho, \mathrm{Y})$ (and similarly for other variables) and denote

$$\bar{\mathbf{s}}^{(\rho,\mathrm{Y})} := \left\langle \bar{\mathbf{s}}_1^{(\rho,\mathrm{Y})}, \bar{\mathbf{m}}^{(\rho,\mathrm{Y})} \right\rangle_{\sigma}.$$

Finally, we define

$$H' := \left\{ \begin{array}{c} (\rho, \mathrm{Y}) \in H : \exists j \in [M], \text{ const. coeff. of} \\ \bar{\mathbf{s}}^{(\rho,\mathrm{Y})^T} \mathbf{R}'_{j,2} \bar{\mathbf{s}}^{(\rho,\mathrm{Y})} + \mathbf{r}'^T_{j,1} \bar{\mathbf{s}}^{(\rho,\mathrm{Y})} + r'_{j,0} \text{ is non-zero} \end{array} \right\}.$$

Then, we have the following claim which follows identically as in Claim 5.1.3.1.

Claim 5.2.12. If $(\rho_P, \rho_E, \mathrm{Y}) \in H$ then

$$\Pr_{(\rho'_E, \mathrm{Y}') \leftarrow \mathfrak{R}_E \times \mathbb{Z}_q^{\lambda \times M}} [(\rho_P, \rho'_E, \mathrm{Y}') \in H] > 0.$$

Moreover, if $(\rho_P, \rho_E, \mathrm{Y}) \in H'$ then

$$\Pr_{\mathrm{Y}' \leftarrow \mathbb{Z}_q^{\lambda \times M}} \left[\forall i \in [\lambda], \tilde{x}_i = 0 \middle| x_i := \bar{g}_i^{(\rho,\mathrm{Y})} + \sum_{j=1}^{M} v'_{i,j} \left(\bar{\mathbf{s}}^{(\rho,\mathrm{Y})^T} \mathbf{R}'_{j,2} \bar{\mathbf{s}}^{(\rho,\mathrm{Y})} + \mathbf{r}'^T_{j,1} \bar{\mathbf{s}}^{(\rho,\mathrm{Y})} + r'_{j,0} \right) \right]$$

is at most $q_1^{-\lambda}$.

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

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

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

- $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2) \neq (\bar{\mathbf{s}}'_1, \bar{\mathbf{s}}'_2)$ and

$$\begin{aligned} 1 &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B) \\ &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}'_1, \bar{\mathbf{m}}', \bar{\mathbf{s}}'_2, \bar{c}'; \mathbf{t}_A \parallel \mathbf{t}_B). \end{aligned}$$

- $\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B) = 1$ and for all $j \in [N]$, $\bar{\mathbf{s}}^T \mathbf{R}_{j,2} \bar{\mathbf{s}} + \mathbf{r}_{j,1}^T \bar{\mathbf{s}} + r_{j,0} = 0$ and for all $i \in [M]$, the constant coefficient of $\bar{\mathbf{s}}^T \mathbf{R}'_{i,2} \bar{\mathbf{s}} + \mathbf{r}'^T_{i,1} \bar{\mathbf{s}} + r'_{i,0}$ is zero where $\bar{\mathbf{s}} := (\bar{\mathbf{s}}_1, \bar{\mathbf{m}})$.

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

Claim 5.2.13. The expected number of calls to $\mathcal{A}$ is at most 2.

The proof follows identically as in Claim 5.2.7. We conclude that the expected runtime of $\mathcal{E}$ is at most $12T$.

Claim 5.2.14. Probability that $\mathcal{E}$ succeeds is at least $\epsilon - 2/|\mathcal{C}| - q^{-d/l} - q_1^{-\lambda}$.

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

Case 1. $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{g}}, \bar{\mathbf{s}}_2) \neq (\bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{g}}', \bar{\mathbf{s}}_2')$ and for $i \in [\lambda]$, $\tilde{h}_i = \tilde{h}_i' = 0$ and

$$\begin{cases} h_i = \bar{g}_i + \sum_{j=1}^{M} v_{i,j} \left(\bar{\mathbf{s}}^T \mathbf{R}_{j,2}' \bar{\mathbf{s}} + \mathbf{r}_{j,1}'^T \bar{\mathbf{s}} + r_{j,0}' \right) \\ h_i' = \bar{g}_i' + \sum_{j=1}^{M} v_{i,j}' \left(\bar{\mathbf{s}}'^T \mathbf{R}_{j,2}' \bar{\mathbf{s}}' + \mathbf{r}_{j,1}'^T \bar{\mathbf{s}}' + r_{j,0}' \right) \end{cases}$$

and for all $j \in [N]$,

$$\bar{\mathbf{s}}^T \mathbf{R}_{j,2} \bar{\mathbf{s}} + \mathbf{r}_{j,1}^T \bar{\mathbf{s}} + r_{j,0} = 0 \text{ and } \bar{\mathbf{s}}'^T \mathbf{R}_{j,2} \bar{\mathbf{s}} + \mathbf{r}_{j,1}^T \bar{\mathbf{s}}' + r_{j,0} = 0$$

and

$$\begin{aligned} 1 &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{\mathbf{g}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B \parallel \mathbf{t}_g) \\ &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1', \bar{\mathbf{m}}' \parallel \bar{\mathbf{g}}', \bar{\mathbf{s}}_2', \bar{c}'; \mathbf{t}_A \parallel \mathbf{t}_B \parallel \mathbf{t}_g). \end{aligned}$$

Case 2. $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{g}}, \bar{\mathbf{s}}_2) = (\bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{g}}', \bar{\mathbf{s}}_2')$ and for $i \in [\lambda]$, $\tilde{h}_i = \tilde{h}_i' = 0$ and

$$\begin{cases} h_i = \bar{g}_i + \sum_{j=1}^{M} v_{i,j} \left(\bar{\mathbf{s}}^T \mathbf{R}_{j,2}' \bar{\mathbf{s}} + \mathbf{r}_{j,1}'^T \bar{\mathbf{s}} + r_{j,0}' \right) \\ h_i' = \bar{g}_i + \sum_{j=1}^{M} v_{i,j}' \left(\bar{\mathbf{s}}^T \mathbf{R}_{j,2}' \bar{\mathbf{s}} + \mathbf{r}_{j,1}'^T \bar{\mathbf{s}} + r_{j,0}' \right) \end{cases}$$

and

$$\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{\mathbf{g}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B \parallel \mathbf{t}_g) = 1$$

and for all $j \in [N]$, $\bar{\mathbf{s}}^T \mathbf{R}_{j,2} \bar{\mathbf{s}} + \mathbf{r}_{j,1}^T \bar{\mathbf{s}} + r_{j,0} = 0$ and for all $j \in [M]$, the constant coefficient of $\bar{\mathbf{s}}^T \mathbf{R}_{j,2}' \bar{\mathbf{s}} + \mathbf{r}_{j,1}'^T \bar{\mathbf{s}} + r_{j,0}'$ is zero.

Case 3. $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{g}}, \bar{\mathbf{s}}_2) = (\bar{\mathbf{s}}'_1, \bar{\mathbf{m}}', \bar{\mathbf{g}}', \bar{\mathbf{s}}'_2)$ and for $i \in [\lambda]$, $\tilde{h}_i = \tilde{h}'_i = 0$ and

$$\begin{cases} h_i = \bar{g}_i + \sum_{j=1}^{M} v_{i,j} \left(\bar{\mathbf{s}}^T \mathbf{R}'_{j,2} \bar{\mathbf{s}} + \mathbf{r}'^T_{j,1} \bar{\mathbf{s}} + r'_{j,0} \right) \\ h'_i = \bar{g}_i + \sum_{j=1}^{M} v'_{i,j} \left(\bar{\mathbf{s}}^T \mathbf{R}'_{j,2} \bar{\mathbf{s}} + \mathbf{r}'^T_{j,1} \bar{\mathbf{s}} + r'_{j,0} \right) \end{cases}$$

and

$$\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{\mathbf{g}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B \parallel \mathbf{t}_g) = 1$$

and for all $j \in [N]$, $\bar{\mathbf{s}}^T \mathbf{R}_{j,2} \bar{\mathbf{s}} + \mathbf{r}_{j,1}^T \bar{\mathbf{s}} + r_{j,0} = 0$ and there exists $j \in [M]$ such that the constant coefficient of $\bar{\mathbf{s}}^T \mathbf{R}'_{j,2} \bar{\mathbf{s}} + \mathbf{r}'^T_{j,1} \bar{\mathbf{s}} + r'_{j,0}$ is non-zero.

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

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

and $\Pr[\mathcal{E} \text{ succeeds}] \geqslant \Pr[E_1 \vee E_2]$. Hence, we only need to upper-bound the probability $\Pr[E_3]$. Define $F(\bar{\mathbf{s}}, \bar{\mathbf{g}})$ to be the event that for all $i \in [\lambda]$, the constant coefficient of

$$\bar{g}_i + \sum_{j=1}^{M} v'_{i,j} \left(\bar{\mathbf{s}}^T \mathbf{R}'_{j,2} \bar{\mathbf{s}} + \mathbf{r}'^T_{j,1} \bar{\mathbf{s}} + r'_{j,0} \right)$$

vanishes. Now, by Claim 5.2.12 we obtain:

$$\begin{aligned}
\Pr[E_3] &\leqslant \Pr\left[\begin{array}{c} (\mathcal{A}(\rho, \mathrm{Y}) \text{ succeeds}) \wedge F(\bar{\mathbf{s}}, \bar{\mathbf{g}}) \wedge \\ \left(\exists j \in [M] : \text{ const. coeff. of } \bar{\mathbf{s}}^T \mathbf{R}'_{j,2} \bar{\mathbf{s}} + \mathbf{r}'^T_{j,1} \bar{\mathbf{s}} + r'_{j,0} \text{ is non-zero} \right) \end{array} \right] \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho, \mathrm{Y}) \in H'} \Pr_{(\rho'_E, \mathrm{Y}') \leftarrow H(\rho_P)} \left[F\left(\bar{\mathbf{s}}^{(\rho, \mathrm{Y})}, \bar{\mathbf{g}}^{(\rho, \mathrm{Y})} \right) \right] \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho, \mathrm{Y}) \in H'} \frac{\Pr_{\mathrm{Y}' \leftarrow \mathbb{Z}_q^{\lambda \times M}} \left[F\left(\bar{\mathbf{s}}^{(\rho, \mathrm{Y})}, \bar{\mathbf{g}}^{(\rho, \mathrm{Y})} \right) \right]}{\Pr_{(\rho'_E, \mathrm{Y}') \leftarrow \mathfrak{R}_E \times \mathbb{Z}_q^{\lambda \times M}} [(\rho'_E, \mathrm{Y}') \in H(\rho_P)]} \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho, \mathrm{Y}) \in H'} \frac{q_1^{-\lambda} \cdot q^{\lambda M} \cdot |\mathfrak{R}_E|}{|H(\rho_P)|} \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho, \mathrm{Y}) \in H} \frac{q_1^{-\lambda} \cdot q^{\lambda M} \cdot |\mathfrak{R}_E|}{|H(\rho_P)|} \\
&\leqslant \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{\rho_P \in \mathfrak{R}_P} \sum_{(\rho_E, \mathrm{Y}) \in H(\rho_P)} \frac{q_1^{-\lambda} \cdot q^{\lambda M} \cdot |\mathfrak{R}_E|}{|H(\rho_P)|} = q_1^{-\lambda}.
\end{aligned}$$

□

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

5.2.3.2 *Reducing the Number of Garbage Commitments*

The approach in Section 5.2.3 requires us to commit to λ additional polynomials g_i in order to have $\approx q_1^{-\lambda}$ soundness error. Here, we consider a special case when $\sigma := \sigma_{-1}{}^4$ and show how to reduce this number by a factor of two for free. In particular, will use the following property of σ_{-1}.

Lemma 5.2.15. *Define the σ_{-1}-trace map $\mathsf{Tr} : \mathcal{R}_q \mapsto \mathcal{R}_q$ as*

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

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

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

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

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

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

$$a^{(i)} := \sum_{j=1}^{M} v_{2i-1,j}\left(\mathbf{s}^T\mathbf{R}'_{j,2}\mathbf{s} + \mathbf{r}'^T_{j,1}\mathbf{s} + r'_{j,0}\right)$$

$$b^{(i)} := \sum_{j=1}^{M} v_{2i,j}\left(\mathbf{s}^T\mathbf{R}'_{j,2}\mathbf{s} + \mathbf{r}'^T_{j,1}\mathbf{s} + r'_{j,0}\right)$$

and apply Lemma 5.2.15 to simultaneously prove that the constant coefficient of both elements in $\mathcal{R}_q$ is equal to zero. Concretely, we prove that the constant *and* middle coefficient of each

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

4 Thus its degree k is equal to 2.

is equal to zero.

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

$$h_i = g_i + \mathsf{Tr}\left(a^{(i)}\right) + X^{d/2}\mathsf{Tr}\left(b^{(i)}\right) \text{ for } i \in [\lambda/2]. \tag{5.23}$$

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

Finally, we need to prove that all $h_1, \ldots, h_{\lambda/2}$ are well-formed. First, we observe that there is an efficiently computable matrix $\mathbf{U} \in \mathcal{R}_q^{2(m_1+\ell)\times 2(m_1+\ell)}$ such that for all $\mathbf{x} \in \mathcal{R}_q^{m_1+\ell}$:

$$\sigma(\langle \mathbf{x} \rangle_\sigma) = \mathbf{U}\langle \mathbf{x} \rangle_\sigma.$$

Hence, we have the following lemma.

Lemma 5.2.16. *Let* $\mathbf{R}_2 \in \mathcal{R}_q^{2(m_1+\ell)\times 2(m_1+\ell)}, \mathbf{r}_1 \in \mathcal{R}_q^{2(m_1+\ell)}$ *and* $r_0 \in \mathcal{R}_q$. *Then*

$$\mathsf{Tr}\left(\mathbf{s}^T\mathbf{R}_2\mathbf{s} + \mathbf{r}_1^T\mathbf{s} + r_0\right) = \mathbf{s}^T\mathbf{V}_2\mathbf{s} + \mathbf{v}_1^T\mathbf{s} + v_0$$

where

$$\begin{aligned}
\mathbf{V}_2 &:= 2^{-1}\left(\mathbf{R}_2 + \mathbf{U}^T\sigma(\mathbf{R}_2)\mathbf{U}\right), \\
\mathbf{v}_1 &:= 2^{-1}\left(\mathbf{r}_1 + \mathbf{U}^T\sigma(\mathbf{r}_1)\right) \\
v_0 &:= 2^{-1}\left(r_0 + \sigma(r_0)\right).
\end{aligned}$$

Proof. The proof follows directly from the fact that $\sigma(\mathbf{s}) = \mathbf{U}\mathbf{s}$. □

By applying Lemma 5.2.16, we note that Equation 5.23 can be written equivalently as:

$$\langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{g} \rangle_\sigma^T \hat{\mathbf{R}}_{N+i,2} \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{g} \rangle_\sigma + \hat{\mathbf{r}}_{N+i,1}^T \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{g} \rangle_\sigma + \hat{r}_{N+i,0} = 0 \tag{5.24}$$

where $\hat{\mathbf{R}}_{N+i,2}, \hat{\mathbf{r}}_{N+i,1}^T, \hat{r}_{N+i,0}$ are defined as follows

$$\begin{aligned}
\hat{\mathbf{R}}_{N+i,2} &:= \begin{bmatrix} \sum_{j=1}^{M} 2^{-1}(v_{2i-1,j} + X^{d/2} v_{2i,j}) \left(\mathbf{R}'_{j,2} + \mathbf{U}^T \sigma(\mathbf{R}'_{j,2}) \mathbf{U} \right) & \mathbf{0}_{2(m_1+\ell)\times\lambda} \\ \mathbf{0}_{\lambda\times 2(m_1+\ell)} & \mathbf{0}_{\lambda\times\lambda} \end{bmatrix} \\
\hat{\mathbf{r}}_{N+i,1}^T &:= \begin{bmatrix} \sum_{j=1}^{M} 2^{-1}(v_{2i-1,j} + X^{d/2} v_{2i,j}) \left(\mathbf{r}'^T_{j,1} + \sigma(\mathbf{r}'^T_{j,1}) \mathbf{U} \right) & \mathbf{e}_i^T \end{bmatrix} \\
\hat{r}_{N+i,0} &:= \sum_{j=1}^{M} 2^{-1}(v_{2i-1,j} + X^{d/2} v_{2i,j}) \left(r'_{j,0} + \sigma(r'_{j,0}) \right) - h_i.
\end{aligned} \tag{5.25}$$

and $\mathbf{e}_i^T = \begin{bmatrix} \mathbf{0}_{1\times 2(i-1)} & 1 & \mathbf{0}_{1\times(\lambda-2i+1)} \end{bmatrix} \in \mathcal{R}_q^\lambda$ is the binary vector which has exactly one 1 in the $(2(i-1)+1)$-th position. Not to mention the fact that we also need to prove (5.17) for $i = 1, \ldots, N$. Identically as before, Equation 5.17 can be written equivalently as

$$\langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{g} \rangle_\sigma^T \hat{\mathbf{R}}_{i,2} \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{g} \rangle_\sigma + \hat{\mathbf{r}}_{i,1}^T \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{g} \rangle_\sigma + \hat{r}_{i,0} = 0 \tag{5.26}$$

where $\hat{\mathbf{R}}_{i,2}, \hat{\mathbf{r}}_{i,1}^T, \hat{r}_{i,0}$ are defined as follows

$$\begin{aligned}
\hat{\mathbf{R}}_{i,2} &:= \begin{bmatrix} \mathbf{R}_{i,2} & \mathbf{0}_{2(m_1+\ell)\times\lambda} \\ \mathbf{0}_{\lambda\times 2(m_1+\ell)} & \mathbf{0}_{\lambda\times\lambda} \end{bmatrix} \in \mathcal{R}_q^{2(m_1+\ell+\lambda/2)\times 2(m_1+\ell+\lambda/2)} \\
\hat{\mathbf{r}}_{i,1}^T &:= \begin{bmatrix} \mathbf{r}_{i,1}^T & \mathbf{0}_{1\times\lambda} \end{bmatrix} \in \mathcal{R}_q^{2(m_1+\ell+\lambda/2)} \\
\hat{r}_{i,0} &:= r_{i,0} \in \mathcal{R}_q.
\end{aligned} \tag{5.27}$$

Hence, we reduce the problem to proving $N + \lambda/2$ quadratic equations in $\langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{g} \rangle_\sigma$ and can thus run the commit-and-prove system $\Pi_{\mathsf{quad-many}}$. We present the commit-and-prove system

$$\Pi_{\mathsf{quad-eval}}^{(-1)} = (\mathsf{ABDLOP}, \mathcal{P}, \mathcal{V})$$

for the relation $R_{\mathsf{quad-eval}}$ in Figure 5.6. Below, we state the security properties of $\Pi_{\mathsf{quad-eval}}^{(-1)}$. We omit the proofs since they are almost identical to the ones presented in Section 5.2.3.1.

Theorem 5.2.17. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1 \eta \alpha$ *and* $\mathfrak{s}_2 = \gamma_2 \eta \nu \sqrt{m_2 d}$ *for some* $\gamma_1, \gamma_2 > 0$ *and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2} \right) \text{ for } i = 1, 2.$$

Prover $\mathcal{P}$ | Verifier $\mathcal{V}$

Inputs:

$pp.\dim = (q, d, \kappa_{\mathsf{MSIS}}, m_1, m_2, \ell, \ell_{\mathsf{ext}} := \lambda/2 + 1)$ | $pp.\dim, pp.\mathsf{norms}$

$pp.\mathsf{norms} = (\nu, \omega, \alpha, B_1, B_2)$ | $pp.\mathsf{mat}$

$pp.\mathsf{mat} = \left(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \begin{bmatrix}\mathbf{B}_{\mathsf{ext}} \\ \mathbf{b}_{\mathsf{ext}}^T\end{bmatrix}\right)$ | $\mathbf{t}_A, \mathbf{t}_B, \sigma := \sigma_{-1}$

$\mathbf{s}_1 \in \mathcal{R}_q^{m_1}, \mathbf{s}_2 \in \mathcal{R}_q^{m_2}, \mathbf{m} \in \mathcal{R}_q^{\ell}$ so that $\|\mathbf{s}_1\| \leqslant \alpha$ | $(r_{i,0}, \mathbf{r}_{i,1}, \mathbf{R}_{i,2})_{i\in[N]}$

$\begin{bmatrix}\mathbf{t}_A \\ \mathbf{t}_B\end{bmatrix} := \begin{bmatrix}\mathbf{A}_1 \\ \mathbf{0}\end{bmatrix}\mathbf{s}_1 + \begin{bmatrix}\mathbf{A}_2 \\ \mathbf{B}\end{bmatrix}\mathbf{s}_2 + \begin{bmatrix}\mathbf{0} \\ \mathbf{m}\end{bmatrix}$ | $(r'_{i,0}, \mathbf{r}'_{i,1}, \mathbf{R}'_{i,2})_{i\in[M]}$

$(r_{i,0}, \mathbf{r}_{i,1}, \mathbf{R}_{i,2})_{i\in[N]}, (r'_{i,0}, \mathbf{r}'_{i,1}, \mathbf{R}'_{i,2})_{i\in[M]}$

$\sigma := \sigma_{-1} \in \mathsf{Aut}(\mathcal{R})$

$\mathbf{s} := \begin{bmatrix}\langle \mathbf{s}_1 \rangle_\sigma \\ \langle \mathbf{m} \rangle_\sigma\end{bmatrix}$

$\mathbf{g} := (g_1, \ldots, g_{\lambda/2}) \leftarrow \{x : \mathcal{R}_q : \tilde{x} = 0\}^{\lambda/2}$

$\mathbf{t}_g := \mathbf{B}_{\mathsf{ext}}\mathbf{s}_2 + \mathbf{g}$

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

$(v_{i,j}) \leftarrow \mathbb{Z}_q^{\lambda/2 \times M}$

$\xleftarrow{(v_{i,j})_{i\in[\lambda/2], j\in[M]}}$

for $i \in [\lambda/2]$:

$h_i := g_i + \sum_{j=1}^{M} (v_{2i-1,j} + X^{d/2} v_{2i,j}) \mathsf{Tr}\left(\mathbf{s}^T \mathbf{R}'_{j,2}\mathbf{s} + \mathbf{r}'^T_{j,1}\mathbf{s} + r'_{j,0}\right)$

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

run $\Pi_{\mathsf{quad-many}}$ with the following inputs: | accept if:

$pp.\dim := (q, d, \kappa_{\mathsf{MSIS}}, m_1, m_2, \ell + \lambda, 1), pp.\mathsf{norms} := pp.\mathsf{norms}$ | (i) $\Pi_{\mathsf{quad-many}}$ verifies

$pp.\mathsf{mat} := \left(\mathbf{A}_1, \mathbf{A}_2, \begin{bmatrix}\mathbf{B} \\ \mathbf{B}_{\mathsf{ext}}\end{bmatrix}, \mathbf{b}_{\mathsf{ext}}^T\right)$ | (ii) $\tilde{h}_1 = \ldots = \tilde{h}_\lambda = 0$

$(\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m})) := (\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m} \parallel \mathbf{g}))$

for $i \in [N]$:

$\mathbf{R}_{i,2} := \begin{bmatrix}\mathbf{R}_{i,2} & \mathbf{0}_{2(m_1+\ell)\times\lambda} \\ \mathbf{0}_{\lambda\times 2(m_1+\ell)} & \mathbf{0}_{\lambda\times\lambda}\end{bmatrix}, \mathbf{r}_{i,1} := \begin{bmatrix}\mathbf{r}_{i,1} \\ \mathbf{0}_{\lambda\times 1}\end{bmatrix}, r_{i,0} := r_{i,0}$

for $i \in [\lambda/2]$:

$\mathbf{R}_{N+i,2} := \begin{bmatrix}\sum_{j=1}^{M} 2^{-1}(v_{2i-1,j} + X^{d/2}v_{2i,j})\left(\mathbf{R}'_{j,2} + \mathbf{U}^T\sigma(\mathbf{R}'_{j,2})\mathbf{U}\right) & \mathbf{0}_{2(m_1+\ell)\times\lambda} \\ \mathbf{0}_{\lambda\times 2(m_1+\ell)} & \mathbf{0}_{\lambda\times\lambda}\end{bmatrix}$

$\mathbf{r}_{N+i,1} := \begin{bmatrix}\sum_{j=1}^{M} 2^{-1}(v_{2i-1,j} + X^{d/2}v_{2i,j})\left(\mathbf{r}'_{j,1} + \mathbf{U}^T\sigma(\mathbf{r}'_{j,1})\right) \\ \mathbf{e}_i\end{bmatrix}$

$r_{N+i,0} := \sum_{j=1}^{M} 2^{-1}(v_{2i-1,j} + X^{d/2}v_{2i,j})\left(r'_{j,0} + \sigma(r'_{j,0})\right) - h_i$

FIGURE 5.6: Commit-and-prove system $\Pi^{(-1)}_{\mathsf{quad-eval}}$ for the relation $R_{\mathsf{quad-eval}}$ with $\sigma := \sigma_{-1}$. Here, $\mathbf{e}_i \in \mathcal{R}_q^\lambda$ is the binary vector which has exactly one 1 in the $(2(i-1)+1)$-th position.

Suppose that $m_1 d \geqslant 5\kappa$ *and* $m_2 d \geqslant 5\kappa$. *Then, the commit-and-prove system* $\Pi^{(-1)}_{\mathsf{quad-eval}}$ *for the relation* $R_{\mathsf{quad-eval}}$ *has statistical completeness with correctness error* $1 - \frac{1}{M_1 M_2}$.

Theorem 5.2.18. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1 \eta \alpha$ *and* $\mathfrak{s}_2 = \gamma_2 \eta \nu \sqrt{m_2 d}$ *for some* $\gamma_1, \gamma_2 > 0$ *and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) \text{ for } i = 1, 2.$$

Suppose $\kappa_{\mathsf{MLWE}} := m_2 - \kappa_{\mathsf{MSIS}} - \ell - \lambda/2 - 1 \geqslant 0$. *Then,* $\Pi^{(-1)}_{\mathsf{quad-eval}}$ *for relation* $R_{\mathsf{quad-eval}}$ *is simulatable under the* $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}}, \kappa_{\mathsf{MSIS}}+\ell+\lambda/2+1, \chi}$ *assumption.*

Theorem 5.2.19. *Suppose* $B_1 \geqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ *and* $B_2 \geqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$. *Then, the commit-and-prove system* $\Pi^{(-1)}_{\mathsf{quad-eval}}$ *for the relation* $R_{\mathsf{quad-eval}}$ *is knowledge sound with knowledge error* $2|\mathcal{C}|^{-1} + q_1^{-d/l} + q_1^{-\lambda}$.

5.2.3.3 *Proving Inner Products over* $\mathbb{Z}_q$

We apply the commit-and-prove system $\Pi^{(-1)}_{\mathsf{quad-eval}}$ to prove inner products between the polynomial coefficients of $\mathbf{s}_1$ and $\mathbf{m}$. Concretely, let $\mathbf{V}_1, \mathbf{V}_2 \in \mathcal{R}_q^{n \times 2(m_1+\ell)}$, $\mathbf{v}_1, \mathbf{v}_2 \in \mathcal{R}_q^n$ and $u \in \mathbb{Z}_q$ be public. Denote $\sigma := \sigma_{-1}$ and $\mathbf{s} := \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma$ as before. Then, we want to prove that

$$\langle \mathbf{V}_1 \mathbf{s} - \mathbf{v}_1, \mathbf{V}_2 \mathbf{s} - \mathbf{v}_2 \rangle \equiv u \pmod{q}. \tag{5.28}$$

Similarly as in Section 5.1.3.2, we apply Lemma 5.1.10 to deduce that (5.28) holds if and only if the constant coefficient of the polynomial

$$\sigma\left(\mathbf{V}_1 \mathbf{s} - \mathbf{v}_1\right)^T \left(\mathbf{V}_2 \mathbf{s} - \mathbf{v}_2\right) - u$$

is equal to zero. Now, using the fact that $\sigma(\mathbf{s}) = \mathbf{U}\mathbf{s}$ described in Section 5.2.3.2, we obtain

$$\begin{aligned}\sigma\left(\mathbf{V}_1 \mathbf{s} - \mathbf{v}_1\right)^T \left(\mathbf{V}_2 \mathbf{s} - \mathbf{v}_2\right) - u &= \left(\sigma(\mathbf{V}_1)\mathbf{U}\mathbf{s} - \sigma(\mathbf{v}_1)\right)^T \left(\sigma(\mathbf{V}_2)\mathbf{U}\mathbf{s} - \sigma(\mathbf{v}_2)\right) - u \\ &= \mathbf{s}^T \mathbf{R}_2' \mathbf{s} + \mathbf{r}_1'^T \mathbf{s} + r_0'\end{aligned}$$

where

$$\begin{aligned}\mathbf{R}_2' &:= \mathbf{U}^T \sigma(\mathbf{V}_1)^T \sigma(\mathbf{V}_2)\mathbf{U} \\ \mathbf{r}_1'^T &:= -\sigma(\mathbf{v}_2)^T \sigma(\mathbf{V}_1)\mathbf{U} - \sigma(\mathbf{v}_1)^T \sigma(\mathbf{V}_2)\mathbf{U} \\ r_0' &:= \sigma(\mathbf{v}_1)^T \sigma(\mathbf{v}_2) - u.\end{aligned}$$

Since this is a quadratic relation on $\mathbf{s}$, we can directly apply $\Pi^{(-1)}_{\mathsf{quad-eval}}$ to prove that the constant coefficient of $\sigma \left(\mathbf{V}_1\mathbf{s} - \mathbf{v}_1\right)^T \left(\mathbf{V}_2\mathbf{s} - \mathbf{v}_2\right) - u$ is equal to zero. The approach extends naturally if we want to prove multiple equations of the form (5.28).

6

TOOLBOX FOR PROVING NORM BOUNDS

Often lattice relations combine two types of statements. First, we want to prove that the committed messages are a solution to some public (e.g. linear or quadratic) equation. This has already been covered in Chapter 5. The second one, however, focuses on proving that the secret messages have small coefficients. One simple yet important example is a proof of knowledge of a MSIS solution. Namely, we want to prove knowledge of a vector $\mathbf{s} \in \mathcal{R}_q^m$ such that:

$$\mathbf{A}\mathbf{s} = \mathbf{u} \quad \text{and} \quad \|\mathbf{s}\| \leqslant B$$

where $\mathbf{A} \in \mathcal{R}_q^{n \times m}$, $\mathbf{u} \in \mathcal{R}_q^n$ and bound B are public.

In this chapter, we concentrate on proving that the norms of committed messages are below certain public bounds. At the core of our techniques lie the so-called *approximate range proofs* (ARP) described in Section 6.1. To briefly show the intuition, suppose we have a vector $\mathbf{s}$ such that

$$\|\mathbf{s}\| \leqslant B \tag{6.1}$$

where bound B is public. Approximate range proofs allow us, not to prove exactly that $\|\mathbf{s}\| \leqslant B$, but to prove that for some known approximation constant $\psi_2 \geqslant 1$, $\|\mathbf{s}\| \leqslant \psi_2 \cdot B$. Similarly, we can prove for the infinity norm that $\|\mathbf{s}\|_\infty \leqslant \psi_\infty \cdot B$ where $\psi_\infty \geqslant 1$. Intuitively, the primary goal of using ARP is to prove that certain equations over $\mathcal{R}_q$ hold also over $\mathcal{R}$ by showing that no wrap-around modulo q occurs.

We apply approximate range proofs and the techniques presented in Chapter 5 to prove smallness of a vector in both L_∞ and L_2 norm. Concretely, in Section 6.2 we propose a novel method to prove that a polynomial vector consists of binary coefficients. For proving larger ranges, one can simply binary-decompose the vector and prove that the longer vector is binary. However, we observe that in applications it is sufficient to only have binary proofs. Furthermore, Section 6.3 focuses on proving (6.1) *exactly*, i.e. without any approximation constants as in the case of ARP.

Finally, we combine the newly introduced methods with techniques for proving linear and quadratic relations from Chapter 5. The end goal of this chapter is thus a general toolbox for proving various relations which (i)

hold over $\mathcal{R}_q$ (e.g. quadratic relations) or (ii) are related to the smallness of the secret polynomials (e.g. proving the L_2 norm).

6.1 APPROXIMATE RANGE PROOFS

In this section we provide techniques for proving that the L_P norm of polynomial vector $\mathbf{s}$, which satisfies $\|\mathbf{s}\| \leqslant B$, is at most $\psi \cdot B$ where ψ is a public constant that does not depend on B. We will consider the two cases $P \in \{2, \infty\}$. Since in practice ψ will be much larger than 1, we only prove shortness of $\mathbf{s}$ *approximately*. In order to apply the results from Section 3.2.3, we set $\kappa = 128$ and aim for 128-bit security. We assume that $d \leqslant 256$ which will always be the case in our instantiations.

Concretely, let $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$ such that $\|\mathbf{s}_1\| \leqslant \alpha$. We initially want to prove that $\mathbf{s}_1, \mathbf{m}$ satisfy the following:

$$\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\| \leqslant B$$

where $\mathbf{D}_s \in \mathcal{R}_q^{n\times m_1}, \mathbf{D}_m \in \mathcal{R}_q^{n\times \ell}$ and $\mathbf{u} \in \mathcal{R}_q^n$ are public. We can define the corresponding relation R_{arp} as

$$R_{\mathsf{arp}} := \{((\mathbf{D}_s, \mathbf{D}_m, \mathbf{u}, B), (\mathbf{s}_1, \mathbf{m})) : \|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\| \leqslant B\}. \tag{6.2}$$

We additionally introduce the *relaxed* relation $R_{\mathsf{arp}}^{(P,\psi)}$

$$R_{\mathsf{arp}}^{(P,\psi)} := \{((\mathbf{D}_s, \mathbf{D}_m, \mathbf{u}, B), (\mathbf{s}_1, \mathbf{m})) : \|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\|_P \leqslant \psi \cdot B\}.$$

Clearly, for any $P \geqslant 2$ and $\psi \geqslant 1$ we have $R_{\mathsf{arp}} \subseteq R_{\mathsf{arp}}^{(P,\psi)}$.

GENERAL STRATEGY. We prove shortness of $\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}$ using the following template. First, we embed its coefficient vector into a 256-dimensional Euclidean space under a random projection. Concretely, let $\mathbf{s} := \mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}$ and $\vec{s} \in \mathbb{Z}_q^{nd}$ be its coefficient vector. We commit to a random masking vector $\mathbf{y}_3 \in \mathcal{R}_q^{256/d}$ [1] and given a random matrix $R \leftarrow \mathsf{Bin}_1^{256\times nd}$ as a challenge, we output the polynomial vector $\mathbf{z}_3 \in \mathcal{R}_q^{256/d}$ such that

$$\vec{z}_3 := \vec{y}_3 + R\vec{s}. \tag{6.3}$$

1 Note that the subscript 3 comes from the fact that $\mathbf{y}_1, \mathbf{y}_2$ are already defined in Figure 5.3 which we use as a black box.

If we also apply rejection sampling on $\vec{z}_3$ then revealing it to the verifier leaks no information about $\mathbf{s}$. Now, the prover simply needs to prove well-formedness of $\mathbf{z}_3$ which is a $\mathbb{Z}_q$-linear relation in $\mathbf{s}_1$, $\mathbf{m}$ and $\mathbf{y}_3$.

Finally, the verifier checks whether $\mathbf{z}_3$ has small coefficients. Then, by Lemmas 3.2.3 and 3.2.5, if $\mathbf{z}_3$ is indeed small and of the correct form then $\mathbf{s}$ must also be small. Since there is a separation between the L_∞ and L_2 norms, we will consider these cases separately.

6.1.1 *Approximate Infinity Norm Proof*

We follow the strategy described above. More precisely, we start by committing to the messages $(\mathbf{s}_1, \mathbf{m})$ using the ABDLOP commitment defined in Section 4.1. Namely, we sample the randomness $\mathbf{s}_2 \leftarrow \chi^{m_2}$ and compute

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

Next, we commit to a random masking vector $\mathbf{y}_3 \leftarrow D^{256}_{\mathfrak{s}_3}$, i.e.

$$\mathbf{t}_y := \mathbf{B}_y \mathbf{s}_2 + \mathbf{y}_3.$$

Then, given a random matrix $R \leftarrow \mathsf{Bin}_1^{256 \times nd}$ as a challenge from the verifier, we compute the polynomial vector $\mathbf{z}_3$ defined by $\vec{z}_3 := \vec{y}_3 + R\vec{s}$. If the rejection sampling algorithm does not abort then we output $\mathbf{z}_3$.

Now, we need to prove that $\mathbf{z}_3$ was well-formed. Let $i \in [256]$ and define $\mathbf{r}_i \in \mathcal{R}_q^{(m_1+\ell)}$ to be the polynomial vector so that its coefficient vector is the i-th row of R. Also, denote $\mathbf{e}_i \in \mathcal{R}_q^{256/d}$ to be the polynomial vector such that its coefficient vector consists of all zeroes and one 1 in the i-th position. Then, (6.3) holds if and only if for all $i = 1, \ldots, 256$ we have

$$\begin{aligned} \langle \mathbf{e}_i, \mathbf{z}_3 \rangle &= \langle \mathbf{e}_i, \mathbf{y}_3 \rangle + \langle \mathbf{r}_i, \mathbf{D}_s \mathbf{s}_1 + \mathbf{D}_m \mathbf{m} + \mathbf{u} \rangle \\ &= \left\langle \begin{bmatrix} \sigma_{-1}(\mathbf{D}_s)^T \mathbf{r}_i \\ \sigma_{-1}(\mathbf{D}_m)^T \mathbf{r}_i \\ \mathbf{e}_i \end{bmatrix}, \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \\ \mathbf{y}_3 \end{bmatrix} \right\rangle + \langle \mathbf{r}_i, \mathbf{u} \rangle \end{aligned} \tag{6.4}$$

where we used the property that $\langle \mathbf{r}_i, \mathbf{D}_s \mathbf{s}_1 \rangle = \langle \sigma_{-1}(\mathbf{D}_s)^T \mathbf{r}_i, \mathbf{s}_1 \rangle$ which follows directly from Lemma 5.1.10. Now, this equation is equivalent to the constant coefficient of the following polynomial being equal to zero:

$$\begin{bmatrix} \sigma_{-1}(\mathbf{r}_i)^T \mathbf{D}_s & \sigma_{-1}(\mathbf{r}_i)^T \mathbf{D}_m & \sigma_{-1}(\mathbf{e}_i)^T \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \\ \mathbf{y}_3 \end{bmatrix} + \langle \mathbf{r}_i, \mathbf{u} \rangle - \langle \mathbf{e}_i, \mathbf{z}_3 \rangle \in \mathcal{R}_q.$$

Proving 256 such statements can be easily done using Π_{eval} from Figure 5.6. Indeed, if we define $\sigma := \sigma_{-1}$ and use Lemma 5.2.1 then we need to prove for $i = 1, \ldots, 256$ that the constant coefficient of

$$\mathbf{r}'^T_{i,1} \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{y}_3 \rangle_\sigma + r'_{i,0} \in \mathcal{R}_q$$

vanishes where

$$\mathbf{r}'^T_1 := \begin{bmatrix} \sigma_{-1}(\mathbf{r}_i)^T \mathbf{D}_s & \sigma_{-1}(\mathbf{r}_i)^T \mathbf{D}_m & \sigma_{-1}(\mathbf{e}_i)^T \end{bmatrix} \mathbf{J}_{m_1+\ell+256/d,2} \in \mathcal{R}_q^{2(m_1+\ell+256/d)}$$
$$r'_0 := \langle \mathbf{r}_i, \mathbf{u} \rangle - \langle \mathbf{e}_i, \mathbf{z}_3 \rangle \in \mathcal{R}_q.$$

Finally, the verifier accepts if the verification equations in Π_{eval} hold and if $\|\mathbf{z}_3\|_\infty \leqslant \sqrt{2\kappa}\mathfrak{s}_3$. By Lemma 3.2.2 and the union bound, the probability that the infinity norm of $\mathbf{z}_3 \leftarrow D^{256}_{\mathfrak{s}_3}$ is greater than $\sqrt{2\kappa}\mathfrak{s}_3$ is at most $256 \cdot 2e^{-\kappa}$.

Now, if $\mathbf{z}_3$ is well-formed and $\|\mathbf{z}_3\|_\infty \leqslant \sqrt{2\kappa}\mathfrak{s}_3$ then by Lemma 3.2.3 we deduce that

$$\|\mathbf{D}_s \mathbf{s}_1 + \mathbf{D}_m \mathbf{m} + \mathbf{u}\|_\infty = \|\mathbf{s}\|_\infty \leqslant 2\sqrt{2\kappa}\mathfrak{s}_3.$$

As before, we can denote the standard deviation $\mathfrak{s}_3 := \gamma_3 T_3$ where $\gamma_3 > 0$ dictates the repetition rate of the rejection sampling and T_3 is an upper-bound on $\|R\vec{s}\|$. Note that by Lemma 3.2.4, $\|R\vec{s}\| \leqslant \sqrt{337}B$ with an overwhelming probability, and thus we can set $\mathfrak{s}_3 := \gamma_3\sqrt{337}B$. Therefore, we conclude that

$$\|\mathbf{D}_s \mathbf{s}_1 + \mathbf{D}_m \mathbf{m} + \mathbf{u}\|_\infty \leqslant 2\gamma_3\sqrt{2\kappa \cdot 337} \cdot B.$$

UTILISING BIMODAL GAUSSIANS. In order to reduce the value of γ_3 and thus the approximation factor on the right-hand side, we apply the bimodal rejection sampling. Namely, we also commit to a sign $\beta \leftarrow \{-1, 1\}$:

$$t_\beta := \mathbf{b}^T_\beta \mathbf{s}_2 + \beta$$

and then compute $\mathbf{z}_3$ in the following way:

$$\vec{z}_3 := \vec{y}_3 + \beta \cdot R\vec{s}.$$

Then, the equation above holds if and only if for all $i = 1, \ldots, 256$ we have

$$\begin{aligned}
\langle \mathbf{e}_i, \mathbf{z}_3 \rangle &= \langle \mathbf{e}_i, \mathbf{y}_3 \rangle + \beta \cdot \langle \mathbf{r}_i, \mathbf{D}_s \mathbf{s}_1 + \mathbf{D}_m \mathbf{m} + \mathbf{u} \rangle \\
&= \left\langle \begin{bmatrix} \beta \cdot \sigma_{-1}(\mathbf{D}_s)^T \mathbf{r}_i \\ \beta \cdot \sigma_{-1}(\mathbf{D}_m)^T \mathbf{r}_i \end{bmatrix}, \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \end{bmatrix} \right\rangle + \left\langle \begin{bmatrix} \mathbf{e}_i \\ \langle \mathbf{r}_i, \mathbf{u} \rangle \end{bmatrix}, \begin{bmatrix} \mathbf{y}_3 \\ \beta \end{bmatrix} \right\rangle \\
&= \left\langle \begin{bmatrix} \mathbf{0}_{m_1 \times (m_1+\ell+256/d)} & \sigma_{-1}(\mathbf{D}_s)^T \mathbf{r}_i \\ \mathbf{0}_{\ell \times (m_1+\ell+256/d)} & \sigma_{-1}(\mathbf{D}_m)^T \mathbf{r}_i \\ \mathbf{0}_{(256/d+1) \times (m_1+\ell+256/d)} & \mathbf{0}_{(256/d+1) \times 1} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \\ \mathbf{y}_3 \\ \beta \end{bmatrix}, \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \\ \mathbf{y}_3 \\ \beta \end{bmatrix} \right\rangle \\
&\qquad + \left\langle \begin{bmatrix} \mathbf{0}_{m_1 \times 1} \\ \mathbf{0}_{\ell \times 1} \\ \mathbf{e}_i \\ \langle \mathbf{r}_i, \mathbf{u} \rangle \end{bmatrix}, \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \\ \mathbf{y}_3 \\ \beta \end{bmatrix} \right\rangle
\end{aligned}$$

which by Lemmas 5.1.10 and 5.2.1 is equivalent to the constant coefficient of the following polynomial being equal to zero:

$$\langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{y}_3 \parallel \beta \rangle_\sigma^T \mathbf{R}'_{i,2} \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{y}_3 \parallel \beta \rangle_\sigma + \mathbf{r}'^T_{i,1} \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{y}_3 \parallel \beta \rangle_\sigma \in \mathcal{R}_q$$

where

$$\mathbf{R}'_{i,2} := \mathbf{J}^T \begin{bmatrix} \mathbf{0}_{(m_1+\ell+256/d) \times m_1} & \mathbf{0}_{(m_1+\ell+256/d) \times \ell} & \mathbf{0}_{(m_1+\ell+256/d) \times (256/d+1)} \\ \sigma_{-1}(\mathbf{r}_i)^T \mathbf{D}_s & \sigma_{-1}(\mathbf{r}_i)^T \mathbf{D}_m & \mathbf{0}_{1 \times (256/d+1)} \end{bmatrix} \mathbf{J}$$

$$\mathbf{r}'^T_{i,1} := \begin{bmatrix} \mathbf{0}_{1 \times m_1} & \mathbf{0}_{1 \times \ell} & \sigma_{-1}(\mathbf{e}_i)^T & \sigma_{-1}(\langle \mathbf{r}_i, \mathbf{u} \rangle) \end{bmatrix} \mathbf{J}$$

and $\mathbf{J} := \mathbf{J}_{m_1+\ell+256/d+1,2}$ as defined in Lemma 5.2.1. Here, we used the property of the matrix $\mathbf{J}$ that

$$\begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \\ \mathbf{y}_3 \\ \beta \end{bmatrix} = \mathbf{J} \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{y}_3 \parallel \beta \rangle_\sigma.$$

As before, proving such statements can be easily done using Π_{eval} from Figure 5.6.

We still need to prove that $\beta \in \{-1, 1\}$. We apply the following simple fact.

Fact 6.1.1. Let $b \in \mathcal{R}_q$. Then, $b \in \{-1, 1\}$ if and only if $b^2 = 1$ and the constant coefficients of $X \cdot b, \ldots, X^{d-1} \cdot b \in \mathcal{R}_q$ are all zeroes.

Hence, we prove that $\beta^2 = 1$ which is equivalent to

$$\begin{bmatrix} \mathbf{s}_1^T & \mathbf{m}^T & \mathbf{y}_3^T & \beta \end{bmatrix} \begin{bmatrix} \mathbf{0}_{(m_1+\ell+256/d)\times(m_1+\ell+256/d)} & \mathbf{0}_{(m_1+\ell+256/d)\times 1} \\ \mathbf{0}_{1\times(m_1+\ell+256/d)} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \\ \mathbf{y}_3 \\ \beta \end{bmatrix}$$

being equal to 1. As a quadratic relation over $\langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{y}_3 \parallel \beta \rangle_\sigma$, we can write $\beta^2 = 1$ as:

$$\langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{y}_3 \parallel \beta \rangle_\sigma^T \mathbf{R}_{1,2} \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{y}_3 \parallel \beta \rangle_\sigma + r_{1,0} = 0$$

where

$$\begin{aligned} \mathbf{R}_{1,2} &:= \mathbf{J}^T \begin{bmatrix} \mathbf{0}_{(m_1+\ell+256/d)\times(m_1+\ell+256/d)} & \mathbf{0}_{(m_1+\ell+256/d)\times 1} \\ \mathbf{0}_{1\times(m_1+\ell+256/d)} & 1 \end{bmatrix} \mathbf{J} \\ r_{1,0} &:= -1. \end{aligned}$$

Next, we need to show that the constant coefficient of $X^i \cdot \beta$ equals zero for $i \in [d-1]$. Equivalently, the constant coefficient of

$$\begin{aligned} X^i \cdot \beta &= \begin{bmatrix} \mathbf{0}_{1\times(m_1+\ell+256/d)} & X^i \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \\ \mathbf{y}_3 \\ \beta \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0}_{1\times(m_1+\ell+256/d)} & X^i \end{bmatrix} \mathbf{J} \langle \mathbf{s}_1 \parallel \mathbf{m} \parallel \mathbf{y}_3 \parallel \beta \rangle_\sigma \end{aligned}$$

vanishes. Hence, we reduced the problem of proving shortness of $\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}$ approximately to proving various quadratic relations.

We present the commit-and-prove system $\Pi_{\mathsf{arp}}^{\infty} = (\mathsf{ABDLOP}, \mathcal{P}, \mathcal{V})$ for the relation R_{arp} in Figure 6.1. In the protocol, we run Π_{eval} defined in Section 5.2.3 for proving quadratic relations.

6.1.1.1 *Security Analysis*

We provide security properties of the commit-and-prove $\Pi_{\mathsf{arp}}^{\infty}$ below.

Prover $\mathcal{P}$ | Verifier $\mathcal{V}$

Inputs:

$pp.\dim = (q, d, \kappa_{\mathsf{MSIS}}, m_1, m_2, \ell, \ell_{\mathsf{ext}} := 256/d + \lambda/2 + 2)$ | $pp.\dim, pp.\mathsf{norms}$

$pp.\mathsf{norms} = (\nu, \omega, \alpha, B_1, B_2)$ | $pp.\mathsf{mat}$

$$pp.\mathsf{mat} = \left(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \begin{bmatrix} \mathbf{B}_y \\ \mathbf{b}_\beta^T \\ \mathbf{B}_{\mathsf{ext}} \\ \mathbf{b}_{\mathsf{ext}}^T \end{bmatrix} \right)$$ | $\mathbf{t}_A, \mathbf{t}_B$

$\mathbf{s}_1 \in \mathcal{R}_q^{m_1}, \mathbf{s}_2 \in \mathcal{R}_q^{m_2}, \mathbf{m} \in \mathcal{R}_q^{\ell}$ so that $\|\mathbf{s}_1\| \leqslant \alpha$ | $\mathbf{D}_s, \mathbf{D}_m, \mathbf{u}$

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

$\mathbf{D}_s \in \mathcal{R}_q^{n \times m_1}, \mathbf{D}_m \in \mathcal{R}_q^{n \times \ell}, \mathbf{u} \in \mathcal{R}_q^n$

$\mathbf{s} := \mathbf{D}_s \mathbf{s}_1 + \mathbf{D}_m \mathbf{m} + \mathbf{u}, \|\mathbf{s}\| \leqslant B$

$\mathbf{y}_3 \leftarrow D_{\mathfrak{s}_3}^{256}$

$\beta \leftarrow \{-1, 1\}$

$\mathbf{t}_y := \mathbf{B}_y \mathbf{s}_2 + \mathbf{y}_3$

$t_\beta := \mathbf{b}_\beta^T \mathbf{s}_2 + \beta$

$\xrightarrow{\mathbf{t}_y, t_\beta}$

$R \leftarrow \mathsf{Bin}_2^{256 \times nd}$

$\xleftarrow{R}$

compute $\mathbf{z}_3 \in \mathcal{R}_q^{256/d}$ so that $\vec{z}_3 := \vec{y}_3 + \beta \cdot R\vec{s}$

if $\mathsf{Rej}^{(3)}(\vec{z}_3, R\vec{s}, \mathfrak{s}_3, M_3) = 1$

then $\mathbf{z}_3 := \perp$

$\xrightarrow{\mathbf{z}_3}$

run Π_{eval} with the following inputs: | accept if:

$pp.\dim := (q, d, \kappa_{\mathsf{MSIS}}, m_1, m_2, \ell + 256/d + 1, \lambda/2 + 1), pp.\mathsf{norms} := pp.\mathsf{norms}$ | (i) Π_{eval} verifies

$$pp.\mathsf{mat} := \left(\mathbf{A}_1, \mathbf{A}_2, \begin{bmatrix} \mathbf{B} \\ \mathbf{B}_y \\ \mathbf{b}_\beta^T \end{bmatrix}, \begin{bmatrix} \mathbf{B}_{\mathsf{ext}} \\ \mathbf{b}_{\mathsf{ext}}^T \end{bmatrix} \right)$$ | (ii) $\|\mathbf{z}_3\|_\infty \leqslant \sqrt{2\kappa}\mathfrak{s}_3$

$(\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m})) := (\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m} \parallel \mathbf{y}_3 \parallel \beta))$

$\mathbf{J} := \mathbf{J}_{\ell+256/d+1,2}$ defined in Lemma 5.2.1, $(\mathbf{e}_i)_{i \in [256]}$ defined as in (6.4)

$$\mathbf{R}_{1,2} := \mathbf{J}^T \begin{bmatrix} \mathbf{0}_{(m_1+\ell+256/d) \times (m_1+\ell+256/d)} & \mathbf{0}_{(m_1+\ell+256/d) \times 1} \\ \mathbf{0}_{1 \times (m_1+\ell+256/d)} & 1 \end{bmatrix} \mathbf{J},$$

$\mathbf{r}_{1,1} := \mathbf{0}_{2(m_1+\ell+256/d) \times 1}, r_{1,0} := -1$

for $i \in [256]$:

$$\mathbf{R}'_{i,2} := \mathbf{J}^T \begin{bmatrix} \mathbf{0}_{(m_1+\ell+256/d) \times m_1} & \mathbf{0}_{(m_1+\ell+256/d) \times \ell} & \mathbf{0}_{(m_1+\ell+256/d) \times (256/d+1)} \\ \sigma_{-1}(\mathbf{r}_i)^T \mathbf{D}_s & \sigma_{-1}(\mathbf{r}_i)^T \mathbf{D}_m & \mathbf{0}_{1 \times (256/d+1)} \end{bmatrix} \mathbf{J}$$

$$\mathbf{r}'_{i,1} := \mathbf{J}^T \begin{bmatrix} \mathbf{0}_{m_1 \times 1} \\ \mathbf{0}_{\ell \times 1} \\ \sigma_{-1}(\mathbf{e}_i) \\ \sigma_{-1}(\langle \mathbf{r}_i, \mathbf{u} \rangle) \end{bmatrix}, r'_{i,0} := 0$$

for $i \in [d-1]$:

$$\mathbf{R}'_{256+i,2} := \mathbf{0}_{2(m_1+\ell+256/d+1) \times 2(m_1+\ell+256/d+1)}, \mathbf{r}'_{256+i,1} := \mathbf{J}^T \begin{bmatrix} \mathbf{0}_{(m_1+\ell+256/d) \times 1} \\ X^i \end{bmatrix}$$

$r'_{256+i,0} = -\langle \mathbf{e}_i, \mathbf{z}_3 \rangle$

FIGURE 6.1: Commit-and-prove system $\Pi_{\mathsf{arp}}^{\infty}$ for the relation R_{arp}.

Theorem 6.1.2. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *and* $\mathsf{Rej}^{(3)} = \mathsf{Rej}_1$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1\eta\alpha, \mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d}$ *and* $\mathfrak{s}_3 = \gamma_3\sqrt{337}B$ *for some* $\gamma_1, \gamma_2, \gamma_3 > 0$ *and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) \text{ for } i = 1,2 \quad \text{and} \quad M_3 := \exp\left(\frac{1}{2\gamma_3^2}\right).$$

Suppose that $m_1 d \geqslant 5\kappa$ *and* $m_2 d \geqslant 5\kappa$*. Then, the commit-and-prove system* $\Pi^{\infty}_{\mathsf{arp}}$ *for the relation* R_{arp} *has statistical completeness with correctness error* $1 - \frac{1}{M_1 M_2 M_3} + 2^{-128}$.

Proof. First, note that $\|R\vec{s}\| \leqslant \sqrt{337}B$ with probability at least $1 - 2^{-128}$ by Lemma 3.2.4. Assuming this inequality holds, the probability that an honest prover succeeds in all three rejection sampling algorithms is $1/(M_1 M_2 M_3)$ by Lemmas 3.3.2 and 3.3.3. In terms of verification equations, $\|\mathbf{z}_3\|_\infty > \sqrt{2\kappa}\mathfrak{s}_3$ with probability at most $256 \cdot 2e^{-\kappa}$ by Lemma 3.2.2. All the other verification equations hold by the discussion above. □

Theorem 6.1.3. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *and* $\mathsf{Rej}^{(3)} = \mathsf{Rej}_1$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1\eta\alpha, \mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d}$ *and* $\mathfrak{s}_3 = \gamma_3\sqrt{337}B$ *for some* $\gamma_1, \gamma_2, \gamma_3 > 0$ *and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) \text{ for } i = 1,2 \quad \text{and} \quad M_3 := \exp\left(\frac{1}{2\gamma_3^2}\right).$$

Suppose $\kappa_{\mathsf{MLWE}} := m_2 - \kappa_{\mathsf{MSIS}} - \ell - \lambda/2 - 256/d - 2 \geqslant 0$*. Then, under the* $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}}, \kappa_{\mathsf{MSIS}}+\ell+\lambda/2+256/d+2, \chi, \mathcal{C}, D^d_{\mathfrak{s}_2}}$ *assumption,* $\Pi^{\infty}_{\mathsf{arp}}$ *for relation* R_{arp} *is simulatable.*

Proof. The simulator $\mathcal{S}$ simply simulates $\mathbf{z}_3$ by picking $\mathbf{z}_3 \leftarrow D^{256}_{\mathfrak{s}_3}$ and following the simulator in Theorem 5.2.18. Thus, the statement holds by Lemma 3.3.3 and the aforementioned theorem. □

As described above, we now show that the commit-and-prove system $\Pi^{\infty}_{\mathsf{arp}}$ for the relation $R^{(\infty,\psi)}_{\mathsf{arp}}$ (and not R_{arp}) is knowledge sound where ψ is a public approximation factor.

Theorem 6.1.4. *Suppose* $B_1 \geqslant 2\mathfrak{s}_1\sqrt{2m_1 d}$ *and* $B_2 \geqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$*. Let* $\mathfrak{s}_3 := \gamma_3\sqrt{337}B$ *and* $\psi := 2\gamma_3\sqrt{337 \cdot 2\kappa}$ *for* $\gamma_3 > 0$*. Then, the commit-and-prove system* $\Pi^{\infty}_{\mathsf{arp}}$ *for the relation* $R^{(\infty,\psi)}_{\mathsf{arp}}$ *is knowledge sound with knowledge error*

$$2|\mathcal{C}|^{-1} + q_1^{-d/l} + q_1^{-\lambda} + 2^{-256}.$$

Proof. Let $\mathcal{P}^*$ be a probabilistic prover which runs in time at most T and convinces the verifier with probability $\epsilon > 2|\mathcal{C}|^{-1} + q_1^{-d/l} + q_1^{-\lambda} + 2^{-256}$. Define a deterministic algorithm $\mathcal{A}(\rho_P, \rho_E, R)$ which given randomness $\rho = (\rho_P, \rho_E) \in \mathfrak{R}_P \times \mathfrak{R}_E$ and challenge $R \in \{-1, 0, 1\}^{256 \times nd}$ does the following. It first runs $\mathcal{P}^*(\rho_P)$ on randomness ρ_P with challenge R and stops after the third round. Let $\mathbf{t}_y, t_\beta$ and $\mathbf{z}_3$ be the output of $\mathcal{P}^*$ in the first and third round respectively. Then, it runs the extractor $\mathcal{E}^*(\rho_E)$ defined in the proof of Theorem 5.2.19 with randomness ρ_E (which runs $\mathcal{P}^*(\rho_P, R)$ in a black-box way).

We say that $\mathcal{A}$ succeeds if $\mathcal{A}$ outputs $(\mathbf{t}_y, t_\beta, R, \mathbf{z}_3, \bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{y}}_3, \bar{\beta}, \bar{\mathbf{s}}_2, \bar{c})$ such that

$$\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{\mathbf{y}}_3 \parallel \bar{\beta}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B \parallel \mathbf{t}_y \parallel t_\beta) = 1$$

and $\|\mathbf{z}_3\| \leqslant \sqrt{2\kappa}\mathfrak{s}_3$ and $\vec{z}_3 = \bar{\beta} \cdot R\vec{s} + \vec{y}_3$ where $\mathbf{s} := \mathbf{D}_s\bar{\mathbf{s}}_1 + \mathbf{D}_m\bar{\mathbf{m}} + \mathbf{u}$ and $\bar{\beta} \in \{-1, 1\}$. As before, we assume that $\mathcal{E}^*$ does not break the binding property of ABDLOP since if it did, then so does $\mathcal{A}$ (and later on $\mathcal{E}$). Clearly, by Theorem 5.2.19, the probability that $\mathcal{A}$ succeeds for random ρ and R is at least $\epsilon - 2/|\mathcal{C}| - q_1^{-d/l} - q_1^{-\lambda}$. Moreover, the expected runtime $\mathcal{A}(\rho_P, \rho_E, R)$ for any fixed ρ_P, R and $\rho_E \leftarrow \mathfrak{R}_E$ is at most $12T$.

We introduce the following notation. Let $H \subseteq \mathfrak{R}_P \times \mathfrak{R}_E \times \{-1, 0, 1\}^{256 \times nd}$ be the set of triples (ρ, R) such that $\mathcal{A}(\rho, R)$ succeeds. Also, define $H(\rho_P)$ to be the set of all (ρ_E, R) for which $(\rho_P, \rho_E, R) \in H$. For fixed $(\rho, R) \in H$, denote $\bar{\mathbf{s}}_1^{(\rho,R)}$ to be the $\mathbf{s}_1$ part of the output of $\mathcal{A}(\rho, R)$ (and similarly for other variables) and denote

$$\mathbf{s}^{(\rho,R)} := \mathbf{D}_s\bar{\mathbf{s}}_1^{(\rho,R)} + \mathbf{D}_m\bar{\mathbf{m}}^{(\rho,R)} + \mathbf{u}.$$

Finally, we define

$$H' := \left\{(\rho, R) \in H : \|\mathbf{s}^{(\rho,R)}\|_\infty > 2\sqrt{2\kappa}\mathfrak{s}_3\right\}.$$

Then, we have the following claim.

Claim 6.1.5. If $(\rho_P, \rho_E, R) \in H$ then

$$\Pr_{(\rho'_E, R') \leftarrow \mathfrak{R}_E \times \mathsf{Bin}_2^{256 \times nd}}[(\rho_P, \rho'_E, R') \in H] > 0.$$

Moreover, if $(\rho_P, \rho_E, R) \in H'$ then

$$\Pr_{R' \leftarrow \mathsf{Bin}_2^{256 \times nd}}\left[\left\|\vec{y}_3^{(\rho,R)} + R'\vec{s}^{(\rho,R)}\right\|_\infty \leqslant \sqrt{2\kappa}\mathfrak{s}_3\right] \leqslant 2^{-256}.$$

Proof. The first part of the claim is identical as e.g. Claim 5.1.3.1. The second one follows directly from definition of H' and Lemma 3.2.3. □

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

1. Sample $\rho = (\rho_P, \rho_E) \leftarrow \mathfrak{R}_P \times \mathfrak{R}_E$ and $R \in \mathsf{Bin}_2^{256\times nd}$ and run $\mathcal{A}(\rho, R)$. If $\mathcal{A}(\rho, R)$ does not succeed, abort.
2. If $\mathcal{A}(\rho, R)$ succeeds, run $\mathcal{A}(\rho_P, \rho'_E, R')$ for the same prover randomness ρ_P but fresh $\rho'_E \leftarrow \mathfrak{R}_E$ and $R' \leftarrow \mathsf{Bin}_2^{256\times nd}$ until $\mathcal{A}$ succeeds.

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

- $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2) \neq (\bar{\mathbf{s}}'_1, \bar{\mathbf{s}}'_2)$ and

$$\begin{aligned} 1 &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B) \\ &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}'_1, \bar{\mathbf{m}}', \bar{\mathbf{s}}'_2, \bar{c}'; \mathbf{t}_A \parallel \mathbf{t}_B). \end{aligned}$$

- $\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B) = 1$ and

$$\|\mathbf{D}_s\bar{\mathbf{s}}_1 + \mathbf{D}_m\bar{\mathbf{m}} + \mathbf{u}\| \leqslant 2\sqrt{2\kappa}\mathfrak{s}_3 = \psi B.$$

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

Claim 6.1.6. The expected number of calls to $\mathcal{A}$ is at most 2.

The proof follows identically as in Claim 5.2.7. We conclude that the expected runtime of $\mathcal{E}$ is at most $24T$.

Claim 6.1.7. Probability that $\mathcal{E}$ succeeds is at least $\epsilon - 2/|\mathcal{C}| - q_1^{-d/l} - q_1^{-\lambda} - 2^{-256}$.

Proof. First, we observe that $\mathcal{E}$ terminates (without aborting) with probability at least $\epsilon - 2/|\mathcal{C}| - q_1^{-d/l} - q_1^{-\lambda}$. Suppose $\mathcal{E}$ indeed terminates and let us write $(\mathbf{t}_y, t_\beta, R, \mathbf{z}_3, \bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{y}}_3, \bar{\mathbf{s}}_2, \bar{c})$ and $(\mathbf{t}_y, t_\beta, R', \mathbf{z}'_3, \bar{\mathbf{s}}'_1, \bar{\mathbf{m}}', \bar{\mathbf{y}}'_3, \bar{\mathbf{s}}'_2, \bar{c}')$ to be the respective outputs of $\mathcal{A}$ in the first and second step of $\mathcal{E}$. We have the following three disjoint cases:

Case 1. $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{y}}_3, \bar{\beta}, \bar{\mathbf{s}}_2) \neq (\bar{\mathbf{s}}'_1, \bar{\mathbf{m}}', \bar{\mathbf{y}}'_3, \bar{\beta}', \bar{\mathbf{s}}'_2)$, $\bar{\beta}, \bar{\beta}' \in \{-1, 1\}$ and $\|\mathbf{z}_3\|_\infty \leqslant \sqrt{2\kappa}\mathfrak{s}_3$, $\|\mathbf{z}'_3\|_\infty \leqslant \sqrt{2\kappa}\mathfrak{s}_3$ and

$$\vec{z}_3 = \bar{\beta} \cdot R\vec{s} + \vec{y}_3 \quad \text{and} \quad \vec{z}'_3 = \bar{\beta}' \cdot R'\vec{s}' + \vec{y}'_3$$

and

$$\begin{aligned}1 &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{\mathbf{y}}_3 \parallel \bar{\beta}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B \parallel t_\beta) \\ &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1', \bar{\mathbf{m}}' \parallel \bar{\mathbf{y}}_3' \parallel \bar{\beta}', \bar{\mathbf{s}}_2', \bar{c}'; \mathbf{t}_A \parallel \mathbf{t}_B \parallel t_\beta).\end{aligned}$$

where

$$\mathbf{s} := \|\mathbf{D}_s\bar{\mathbf{s}}_1 + \mathbf{D}_m\bar{\mathbf{m}} + \mathbf{u}\|, \quad \mathbf{s} := \|\mathbf{D}_s\bar{\mathbf{s}}_1' + \mathbf{D}_m\bar{\mathbf{m}}' + \mathbf{u}\|.$$

Case 2. $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{y}}_3, \bar{\beta}, \bar{\mathbf{s}}_2) = (\bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{y}}_3', \bar{\beta}', \bar{\mathbf{s}}_2')$ and $\bar{\beta} \in \{-1, 1\}$ and $\|\mathbf{z}_3\|_\infty \leqslant \sqrt{2\kappa}\mathfrak{s}_3, \|\mathbf{z}_3'\|_\infty \leqslant \sqrt{2\kappa}\mathfrak{s}_3$ and

$$\vec{z}_3 = \bar{\beta} \cdot R\vec{s} + \vec{y}_3 \quad \text{and} \quad \vec{z}_3' = \bar{\beta} \cdot R'\vec{s} + \vec{y}_3$$

and

$$1 = \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{\mathbf{y}}_3 \parallel \bar{\beta}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B \parallel t_\beta).$$

and $\|\vec{s}\| \leqslant 2\sqrt{2\kappa}\mathfrak{s}_3$ where $\mathbf{s} := \|\mathbf{D}_s\bar{\mathbf{s}}_1 + \mathbf{D}_m\bar{\mathbf{m}} + \mathbf{u}\|$.

Case 3. $(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{y}}_3, \bar{\beta}, \bar{\mathbf{s}}_2) = (\bar{\mathbf{s}}_1', \bar{\mathbf{m}}', \bar{\mathbf{y}}_3', \bar{\beta}', \bar{\mathbf{s}}_2')$ and $\bar{\beta} \in \{-1, 1\}$ and $\|\mathbf{z}_3\|_\infty \leqslant \sqrt{2\kappa}\mathfrak{s}_3, \|\mathbf{z}_3'\|_\infty \leqslant \sqrt{2\kappa}\mathfrak{s}_3$ and

$$\vec{z}_3 = \bar{\beta} \cdot R\vec{s} + \vec{y}_3 \quad \text{and} \quad \vec{z}_3' = \bar{\beta} \cdot R'\vec{s} + \vec{y}_3$$

and

$$1 = \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{\mathbf{y}}_3 \parallel \bar{\beta}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B \parallel t_\beta).$$

and $\|\vec{s}\| > 2\sqrt{2\kappa}\mathfrak{s}_3$ where $\mathbf{s} := \|\mathbf{D}_s\bar{\mathbf{s}}_1 + \mathbf{D}_m\bar{\mathbf{m}} + \mathbf{u}\|$.

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

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

and

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

Hence, we only need to upper-bound the probability $\Pr[E_3]$. Clearly, we have

$$\Pr[E_3] \leqslant \Pr\left[(\mathcal{A}(\rho, R) \text{ succeeds}) \wedge \left(\left\|\vec{y}_3 + R'\vec{s}\right\|_\infty \leqslant \sqrt{2\kappa}\mathfrak{s}_3\right) \wedge \left(\|\mathbf{s}\| > 2\sqrt{2\kappa}\mathfrak{s}_3\right)\right].$$

Define $\mathcal{D}(\rho^*, R^*) := \Pr_{(\rho,R) \leftarrow \mathfrak{R}_P \times \mathfrak{R}_E \times \mathsf{Bin}_2^{256 \times nd}}[(\rho, R) = (\rho^*, R^*)]$ for fixed $(\rho^*, R^*) \in \mathfrak{R}_P \times \mathfrak{R}_E \times \{-1, 0, 1\}^{256 \times nd}$.

Now, by Claim 6.1.5 we obtain:

$$\begin{aligned}
\Pr[E_3] &\leqslant \sum_{(\rho^*,R^*)\in H'} \Pr_{(\rho'_E,R')\leftarrow H(\rho^*_P)}\left[\left\|\vec{y}_3^{(\rho^*,R^*)} + R'\vec{s}^{(\rho^*,R^*)}\right\|_\infty \leqslant \sqrt{2\kappa}\mathfrak{s}_3\right]\cdot\mathcal{D}(\rho^*,R^*)\\
&\leqslant \sum_{(\rho^*,R^*)\in H'} \frac{\Pr_{R'\leftarrow \mathsf{Bin}_2^{256\times nd}}\left[\left\|\vec{y}_3^{(\rho^*,R^*)} + R'\vec{s}^{(\rho^*,R^*)}\right\|_\infty \leqslant \sqrt{2\kappa}\mathfrak{s}_3\right]}{\Pr_{(\rho'_E,R')\leftarrow\mathfrak{R}_E\times\mathsf{Bin}_2^{256\times nd}}[(\rho'_E,R')\in H(\rho^*_P)]}\cdot\mathcal{D}(\rho^*,R^*)\\
&\leqslant \sum_{(\rho^*,R^*)\in H'} \frac{2^{-256}}{\Pr_{(\rho'_E,R')\leftarrow\mathfrak{R}_E\times\mathsf{Bin}_2^{256\times nd}}[(\rho'_E,R')\in H(\rho^*_P)]}\cdot\mathcal{D}(\rho^*,R^*)\\
&\leqslant \sum_{(\rho^*,R^*)\in H} \frac{2^{-256}}{\Pr_{(\rho'_E,R')\leftarrow\mathfrak{R}_E\times\mathsf{Bin}_2^{256\times nd}}[(\rho'_E,R')\in H(\rho^*_P)]}\cdot\mathcal{D}(\rho^*,R^*)\\
&\leqslant \sum_{\rho^*_P\in\mathfrak{R}_P}\sum_{(\rho^*_E,R^*)\in H(\rho^*_P)} \frac{2^{-256}}{\Pr_{(\rho'_E,R')\leftarrow\mathfrak{R}_E\times\mathsf{Bin}_2^{256\times nd}}[(\rho'_E,R')\in H(\rho^*_P)]}\cdot\mathcal{D}(\rho^*,R^*)\\
&\leqslant 2^{-256}\cdot\sum_{\rho^*_P\in\mathfrak{R}_P} \frac{\sum_{(\rho^*_E,R^*)\in H(\rho^*_P)}\mathcal{D}(\rho^*,R^*)}{\Pr_{(\rho'_E,R')\leftarrow\mathfrak{R}_E\times\mathsf{Bin}_2^{256\times nd}}[(\rho'_E,R')\in H(\rho^*_P)]}\\
&\leqslant 2^{-256}.
\end{aligned}$$

□

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

6.1.2 *Approximate Euclidean Norm Proof*

Suppose we want to prove the L_2 norm approximately, i.e. $\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\| \leqslant \psi\cdot B$. The approach is almost identical to the one presented in Section 6.1.1 with the only change being that the verifier checks if the L_2 norm of the message $\mathbf{z}_3$. Indeed, instead of requiring $\|\mathbf{z}_3\|_\infty \leqslant \sqrt{2\kappa}\mathfrak{s}_3$, we check whether $\|\mathbf{z}_3\| \leqslant \varrho\mathfrak{s}_3\sqrt{256}$ where $\varrho\in\mathbb{R}^+$ satisfies

$$\varrho^{256}\cdot e^{128(1-\varrho^2)} = 2^{-\kappa}.$$

For example, when $\kappa = 128$ then we can set $\varrho = 1.64$. Clearly, if $\mathbf{z}_3 \leftarrow D^{256}_{\mathfrak{s}_3}$ then by Lemma 3.2.2, the probability that $\|\mathbf{z}_3\| \leqslant \varrho\mathfrak{s}_3\sqrt{256}$ is at least $1-2^{-\kappa}$.

Assume that we proved the well-formedness of $\mathbf{z}_3$ as before, namely

$$\vec{z}_3 = \vec{y}_3 + \beta\cdot R\vec{s}.$$

Now we can apply Lemma 3.2.5. Concretely, with probability at least $1 - 2^{-128}$, we conclude that

$$\|\vec{s}\| = \|\beta \cdot \vec{s}\| \leqslant \frac{2}{\sqrt{26}} \cdot \varrho \mathfrak{s}_3 \sqrt{256}.$$

As before, we will set $\mathfrak{s}_3 = \gamma_3 \sqrt{337} B$ by Lemma 3.2.4 and thus we proved the norm of $\mathbf{s}$ approximately where $\psi := 2\sqrt{\frac{337 \cdot 256}{26}} \cdot \varrho \cdot \gamma_3$.

One of the main differences from the infinity norm case is that in order to use Lemma 3.2.5, we need large enough modulus q so that no wrap-around occurs. Namely, q should satisfy

$$q \geqslant 41 \cdot nd \cdot \frac{2}{\sqrt{26}} \cdot \varrho \mathfrak{s}_3 \sqrt{256}.$$

We present the commit-and-prove system $\Pi^2_{\mathsf{arp}} = (\mathsf{ABDLOP}, \mathcal{P}, \mathcal{V})$ for the relation R_{arp} in Figure 6.2. In the protocol, we run Π_{eval} defined in Section 5.2.3 for proving quadratic relations.

6.1.2.1 *Security Analysis*

As discussed above, the only difference to $\Pi^{\infty}_{\mathsf{arp}}$ is that the verifier checks the shortness of the vector $\mathbf{z}_3$ in the L_2 rather than the L_∞ norm. Thus, the security analysis is almost identical to the results presented in Section 6.1.1.1. Hence, we provide security properties of the commit-and-prove Π^2_{arp} and omit the proofs.

Theorem 6.1.8. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *and* $\mathsf{Rej}^{(3)} = \mathsf{Rej}_1$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1 \eta \alpha, \mathfrak{s}_2 = \gamma_2 \eta \nu \sqrt{m_2 d}$ *and* $\mathfrak{s}_3 = \gamma_3 \sqrt{337} B$ *for some* $\gamma_1, \gamma_2, \gamma_3 > 0$ *and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) \text{ for } i = 1,2 \quad \text{and} \quad M_3 := \exp\left(\frac{1}{2\gamma_3^2}\right).$$

Suppose that $m_1 d \geqslant 5\kappa$ *and* $m_2 d \geqslant 5\kappa$. *Then, the commit-and-prove system* Π^2_{arp} *for the relation* R_{arp} *has statistical completeness with correctness error* $1 - \frac{1}{M_1 M_2 M_3} + 2^{-128}$.

Theorem 6.1.9. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *and* $\mathsf{Rej}^{(3)} = \mathsf{Rej}_1$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1 \eta \alpha, \mathfrak{s}_2 = \gamma_2 \eta \nu \sqrt{m_2 d}$ *and* $\mathfrak{s}_3 = \gamma_3 \sqrt{337} B$ *for some* $\gamma_1, \gamma_2, \gamma_3 > 0$ *and define*

$$M_i := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) \text{ for } i = 1,2 \quad \text{and} \quad M_3 := \exp\left(\frac{1}{2\gamma_3^2}\right).$$

Prover $\mathcal{P}$ | Verifier $\mathcal{V}$

Inputs:

$pp.\dim = (q, d, \kappa_{\mathsf{MSIS}}, m_1, m_2, \ell, \ell_{\mathsf{ext}} := 256/d + \lambda/2 + 2)$ | $pp.\dim$, $pp.\mathsf{norms}$

$pp.\mathsf{norms} = (\nu, \omega, \alpha, B_1, B_2)$ | $pp.\mathsf{mat}$

$$pp.\mathsf{mat} = \left(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \begin{bmatrix} \mathbf{B}_y \\ \mathbf{b}_\beta^T \\ \mathbf{B}_{\mathsf{ext}} \\ \mathbf{b}_{\mathsf{ext}}^T \end{bmatrix} \right)$$ | $\mathbf{t}_A, \mathbf{t}_B$

$\mathbf{s}_1 \in \mathcal{R}_q^{m_1}, \mathbf{s}_2 \in \mathcal{R}_q^{m_2}, \mathbf{m} \in \mathcal{R}_q^{\ell}$ so that $\|\mathbf{s}_1\| \leqslant \alpha$ | $\mathbf{D}_s, \mathbf{D}_m, \mathbf{u}$

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

$\mathbf{D}_s \in \mathcal{R}_q^{n \times m_1}, \mathbf{D}_m \in \mathcal{R}_q^{n \times \ell}, \mathbf{u} \in \mathcal{R}_q^n$ | $\varrho^{256} \cdot e^{128(1-\varrho^2)} = 2^{-\kappa}$

$\mathbf{s} := \mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}, \|\mathbf{s}\| \leqslant B$

$\mathbf{y}_3 \leftarrow D_{\mathfrak{s}_3}^{256}$

$\beta \leftarrow \{-1, 1\}$

$\mathbf{t}_y := \mathbf{B}_y\mathbf{s}_2 + \mathbf{y}_3$

$t_\beta := \mathbf{b}_\beta^T\mathbf{s}_2 + \beta$

$\xrightarrow{\mathbf{t}_y, t_\beta}$

$R \leftarrow \mathsf{Bin}_2^{256 \times nd}$

$\xleftarrow{R}$

compute $\mathbf{z}_3 \in \mathcal{R}_q^{256/d}$ so that $\vec{z}_3 := \vec{y}_3 + \beta \cdot R\vec{s}$

if $\mathsf{Rej}^{(3)}(\vec{z}_3, R\vec{s}, \mathfrak{s}_3, M_3) = 1$

then $\mathbf{z}_3 := \perp$

$\xrightarrow{\mathbf{z}_3}$

run Π_{eval} with the following inputs: | accept if:

$pp.\dim := (q, d, \kappa_{\mathsf{MSIS}}, m_1, m_2, \ell + 256/d + 1, \lambda/2 + 1), pp.\mathsf{norms} := pp.\mathsf{norms}$ | (i) Π_{eval} verifies

$$pp.\mathsf{mat} := \left(\mathbf{A}_1, \mathbf{A}_2, \begin{bmatrix} \mathbf{B} \\ \mathbf{B}_y \\ \mathbf{b}_\beta^T \end{bmatrix}, \begin{bmatrix} \mathbf{B}_{\mathsf{ext}} \\ \mathbf{b}_{\mathsf{ext}}^T \end{bmatrix} \right)$$ | (ii) $\|\mathbf{z}_3\| \leqslant \varrho\mathfrak{s}_3\sqrt{256}$

$(\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m})) := (\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m} \parallel \mathbf{y}_3 \parallel \beta))$

$\mathbf{J} := \mathbf{J}_{\ell+256/d+1,2}$ defined in Lemma 5.2.1, $(\mathbf{e}_i)_{i\in[256]}$ defined as in (6.4)

$$\mathbf{R}_{1,2} := \mathbf{J}^T \begin{bmatrix} \mathbf{0}_{(m_1+\ell+256/d)\times(m_1+\ell+256/d)} & \mathbf{0}_{(m_1+\ell+256/d)\times 1} \\ \mathbf{0}_{1\times(m_1+\ell+256/d)} & 1 \end{bmatrix} \mathbf{J},$$

$\mathbf{r}_{1,1} := \mathbf{0}_{2(m_1+\ell+256/d)\times 1}, r_{1,0} := -1$

for $i \in [256]$:

$$\mathbf{R}'_{i,2} := \mathbf{J}^T \begin{bmatrix} \mathbf{0}_{(m_1+\ell+256/d)\times m_1} & \mathbf{0}_{(m_1+\ell+256/d)\times \ell} & \mathbf{0}_{(m_1+\ell+256/d)\times(256/d+1)} \\ \sigma_{-1}(\mathbf{r}_i)^T\mathbf{D}_s & \sigma_{-1}(\mathbf{r}_i)^T\mathbf{D}_m & \mathbf{0}_{1\times(256/d+1)} \end{bmatrix} \mathbf{J}$$

$$\mathbf{r}'_{i,1} := \mathbf{J}^T \begin{bmatrix} \mathbf{0}_{m_1\times 1} \\ \mathbf{0}_{\ell\times 1} \\ \sigma_{-1}(\mathbf{e}_i) \\ \sigma_{-1}(\langle \mathbf{r}_i, \mathbf{u} \rangle) \end{bmatrix}, r'_{i,0} := 0$$

for $i \in [d-1]$:

$$\mathbf{R}'_{256+i,2} := \mathbf{0}_{2(m_1+\ell+256/d+1)\times 2(m_1+\ell+256/d+1)}, \mathbf{r}'_{256+i,1} := \mathbf{J}^T \begin{bmatrix} \mathbf{0}_{(m_1+\ell+256/d)\times 1} \\ X^i \end{bmatrix}$$

$r'_{256+i,0} = -\langle \mathbf{e}_i, \mathbf{z}_3 \rangle$

FIGURE 6.2: Commit-and-prove system Π_{arp}^2 for the relation R_{arp}.

Suppose $\kappa_{\mathsf{MLWE}} := m_2 - \kappa_{\mathsf{MSIS}} - \ell - \lambda/2 - 256/d - 2 \geqslant 0$. *Then, under the* $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}},\kappa_{\mathsf{MSIS}}+\ell+\lambda/2+256/d+2,\chi,\mathcal{C},D^d_{\mathfrak{s}_2}}$ *assumption,* Π^2_{arp} *for relation* R_{arp} *is simulatable.*

Theorem 6.1.10. *Suppose* $B_1 \geqslant 2\mathfrak{s}_1\sqrt{2m_1d}$ *and* $B_2 \geqslant 2\mathfrak{s}_2\sqrt{2m_2d}$. *Let* $\mathfrak{s}_3 := \gamma_3\sqrt{337}B$ *and* $\psi := 2\sqrt{\frac{337\cdot 256}{26}}\varrho\gamma_3$ *for* $\gamma_3 > 0$. *If*

$$q \geqslant 41 \cdot nd \cdot \frac{2}{\sqrt{26}} \cdot \varrho\mathfrak{s}_3\sqrt{256}$$

then, the commit-and-prove system Π^2_{arp} *for the relation* $R^{(2,\psi)}_{\mathsf{arp}}$ *is knowledge sound with knowledge error* $2|\mathcal{C}|^{-1} + q_1^{-d/l} + q_1^{-\lambda} + 2^{-128}$.

The soundness proof is almost identical to the proof of Theorem 6.1.4 with the only difference being that we apply Lemma 3.2.5 instead of Lemma 3.2.3. This has two consequences. Namely, (i) we need large enough modulus q to use Lemma 3.2.5 and (ii) the constant term in the knowledge error becomes 2^{-128} instead of 2^{-256}.

Remark. As expected, this approximate Euclidean norm proof is tighter than just proving the L_∞ norm proof from Section 6.1.1 and deducing the L_2 norm by using trivial inequalities. Indeed, if one were to do the naive method, one would end up with proving $\|\mathbf{s}\|_\infty \leqslant 2\gamma_3\sqrt{337 \cdot 2\kappa}$ and thus

$$\|\mathbf{s}\| \leqslant 2\gamma_3\sqrt{337 \cdot 2\kappa}\sqrt{nd}$$

which is dependent on the dimension of the vector $\mathbf{s}$. As shown in Theorem 6.1.10, we can prove the L_2 norm of $\mathbf{s}$ approximately, where the approximation factor ψ is independent of dimension of $\mathbf{s}$. However, this comes at the cost of an additional condition on the proof system modulus q. Hence, when it is not important to prove the norm of $\mathbf{s}$ tightly, i.e. we are fine with a relatively large approximation factor, then it is more beneficial to prove the Euclidean norm through the L_∞ norm proof from Section 6.1.1.

6.2 PROVING EXACT SHORTNESS IN THE INFINITY NORM

In this section, we present a way to prove exactly that coefficients of a secret vector $\mathbf{s}$ lie in a certain (public) range. At the core of our techniques is a new proof that coefficients of $\mathbf{s}$ are binary. Note that if one were interested in proving larger ranges, e.g. that coefficients of $\mathbf{s}$ are in $\{-1, 0, 1\}$, then one

could simply binary-decompose $\mathbf{s}$ into a larger vector $\mathbf{s}'$ and prove that $\mathbf{s}'$ has binary coefficients instead.

Concretely, let $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$ such that $\|\mathbf{s}_1\| \leqslant \alpha$. We initially want to prove that the coefficients of the following vector are binary

$$\mathbf{s} := \mathbf{D}_s \mathbf{s}_1 + \mathbf{D}_m \mathbf{m} + \mathbf{u} \in \mathcal{R}_q^n$$

where $\mathbf{D}_s \in \mathcal{R}_q^{n \times m_1}, \mathbf{D}_m \in \mathcal{R}_q^{n \times \ell}$ and $\mathbf{u} \in \mathcal{R}_q^n$ are public. In order to prove such statements, we rely on the following simple observation.

Lemma 6.2.1. *Let $\vec{s} \in \mathbb{Z}^m$. Then, $\vec{s} \in \{0,1\}^m$ if and only if $\langle \vec{s}, \vec{s} - \vec{1} \rangle = 0$.*

Proof. Clearly, if $\vec{s} \in \{0,1\}^m$ then for each $i \in [m]$, $s_i(s_i - \vec{1}) = 0$ and thus $\langle \vec{s}, \vec{s} - \vec{1} \rangle = 0$. On the other hand, note that

$$\forall a \in \mathbb{Z}, a(a-1) \geqslant 0$$

and the equality holds if and only if $a \in \{0,1\}$. Hence, if $\vec{s} \notin \{0,1\}^m$ then $\langle \vec{s}, \vec{s} - \vec{1} \rangle > 0$. □

We denote

$$\mathbf{x} := (1 + X + \ldots + X^{d-1}, \ldots, 1 + X + \ldots + X^{d-1}) \in \mathcal{R}_q^n \tag{6.5}$$

to be the vector such that $\vec{x} = 1^{nd} \in \mathbb{Z}^{nd}$. Then, Lemma 6.2.1 says that $\mathbf{s}$ has binary coefficients if and only if $\langle \mathbf{s}, \mathbf{s} - \mathbf{x} \rangle = 0$ over integers. We observe that if one were to prove the inner product modulo q then by Lemma 5.1.10, this corresponds to proving that the constant coefficient of $\sigma(\mathbf{s} - \mathbf{x})^T \mathbf{s}$, which can be equivalently written as

$$\begin{bmatrix} \mathbf{s}_1^T & \mathbf{m}^T & \sigma(\mathbf{s}_1)^T & \sigma(\mathbf{m})^T \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \sigma(\mathbf{D}_s)^T \mathbf{D}_s & \sigma(\mathbf{D}_s)^T \mathbf{D}_m & \mathbf{0} & \mathbf{0} \\ \sigma(\mathbf{D}_m)^T \mathbf{D}_s & \sigma(\mathbf{D}_m)^T \mathbf{D}_m & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \\ \sigma(\mathbf{s}_1) \\ \sigma(\mathbf{m}) \end{bmatrix}$$

$$+ \begin{bmatrix} \sigma(\mathbf{u} - \mathbf{x})^T \mathbf{D}_s & \sigma(\mathbf{u} - \mathbf{x})^T \mathbf{D}_m & \mathbf{u}^T \sigma(\mathbf{D}_s) & \mathbf{u}^T \sigma(\mathbf{D}_m) \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \\ \sigma(\mathbf{s}_1) \\ \sigma(\mathbf{m}) \end{bmatrix}$$

$$+ \sigma(\mathbf{u} - \mathbf{x})^T \mathbf{u}$$

vanishes where $\sigma := \sigma_{-1}$ as usual. Proving such a statement was already covered in Section 5.2.3. Finally, in order to prove that $\langle \mathbf{s}, \mathbf{s} - \mathbf{x} \rangle = 0$ holds over integers and modulo q, we apply the approximate L_2 proof on $\mathbf{s}$. Since its coefficients are binary, we have $\|\mathbf{s}\| \leqslant \sqrt{nd}$. The proof system in Section 6.1.2 allows us to prove that

$$\|\mathbf{s}\| \leqslant \psi \cdot \sqrt{nd} = 2\sqrt{\frac{337 \cdot 256}{26}} \varrho \gamma_3 \cdot \sqrt{nd}.$$

Now, note that for any $\vec{a} \in \mathbb{R}^m$ such that $\|\vec{a}\| \leqslant B$ we have:

$$\left| \sum_{i=1}^{m} a_i(a_i - 1) \right| \leqslant \sum_{i=1}^{m} a_i^2 + \sum_{i=1}^{m} |a_i| \leqslant \sum_{i=1}^{m} a_i^2 + m\sqrt{\frac{\sum_{i=1}^{m} a_i^2}{m}} \leqslant B^2 + B\sqrt{m}. \quad (6.6)$$

Here we used the inequality of arithmetic and geometric means.

Hence, we want to set the modulus q so that the extracted $\mathbf{s}$ satisfies $|\langle \mathbf{s}, \mathbf{s} - \mathbf{x} \rangle| < q$. Namely, we pick q such that

$$|\langle \mathbf{s}, \mathbf{s} - \mathbf{x} \rangle| \leqslant \psi^2 nd + \psi nd = \psi nd(\psi + 1) < q.$$

If this is the case and $\langle \mathbf{s}, \mathbf{s} - \mathbf{x} \rangle = 0$ modulo q then we can deduce that $\langle \mathbf{s}, \mathbf{s} - \mathbf{x} \rangle = 0$ over integers. This proves that the coefficients of the extracted $\mathbf{s}$ are binary.

6.3 PROVING EXACT SHORTNESS IN THE EUCLIDEAN NORM

This section focuses on proving exactly that a secret vector $\mathbf{s}$ satisfies $\|\mathbf{s}\| \leqslant B$ for some bound $B < \sqrt{q}$, without any approximation factors. For simplicity, we first consider a simple case – proving that $\|\mathbf{s}\| = B$.

Concretely, let $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$ such that $\|\mathbf{s}_1\| \leqslant \alpha$. We initially want to prove that

$$\|\mathbf{s}\| = B \text{ where } \mathbf{s} := \mathbf{D}_s \mathbf{s}_1 + \mathbf{D}_m \mathbf{m} + \mathbf{u} \in \mathcal{R}_q^n$$

and $\mathbf{D}_s \in \mathcal{R}_q^{n \times m_1}, \mathbf{D}_m \in \mathcal{R}_q^{n \times \ell}$ and $\mathbf{u} \in \mathcal{R}_q^n$ are public.

The main observation in this section is that $\|\mathbf{s}\|^2 = \langle \mathbf{s}, \mathbf{s} \rangle$ and thus in order to prove $\|\mathbf{s}\| = B$ modulo q, we simply need to prove that the constant coefficient of the polynomial $\sigma(\mathbf{s})^T \mathbf{s} - B^2$ vanishes. We can do that using

techniques from Section 5.2.3. Indeed, the polynomial can be equivalently written as

$$\begin{bmatrix} \mathbf{s}_1^T & \mathbf{m}^T & \sigma(\mathbf{s}_1)^T & \sigma(\mathbf{m})^T \end{bmatrix} \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \sigma(\mathbf{D}_s)^T\mathbf{D}_s & \sigma(\mathbf{D}_s)^T\mathbf{D}_m & \mathbf{0} & \mathbf{0} \\ \sigma(\mathbf{D}_m)^T\mathbf{D}_s & \sigma(\mathbf{D}_m)^T\mathbf{D}_m & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \\ \sigma(\mathbf{s}_1) \\ \sigma(\mathbf{m}) \end{bmatrix}$$

$$+ \begin{bmatrix} \sigma(\mathbf{u})^T\mathbf{D}_s & \sigma(\mathbf{u})^T\mathbf{D}_m & \mathbf{u}^T\sigma(\mathbf{D}_s) & \mathbf{u}^T\sigma(\mathbf{D}_m) \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \\ \sigma(\mathbf{s}_1) \\ \sigma(\mathbf{m}) \end{bmatrix} + \sigma(\mathbf{u})^T\mathbf{u} - B^2.$$

To prove that $\langle \mathbf{s}, \mathbf{s} \rangle = B^2$ over integers, we show $|\langle \mathbf{s}, \mathbf{s} \rangle - B^2| < q$ for a large enough proof system modulus q. Using the approximate range proof from Section 6.1.2, we can convince the verifier that $\|\mathbf{s}\| \leqslant \psi \cdot B$. Thus, we pick q such that

$$|\langle \mathbf{s}, \mathbf{s} \rangle - B^2| \leqslant \psi^2 \cdot B^2 + B^2 = (\psi^2 + 1)B^2 < q.$$

This implies that $\|\mathbf{s}\|^2 = B^2$ over integers.

Next, suppose we want to prove that $\|\mathbf{s}\| \leqslant B$, i.e. relation R_{arp} in (6.2). The idea is to commit to the binary representation of the difference $B^2 - \|\mathbf{s}\|^2$ and prove that it has binary coefficients. Namely, for $0 < x < \sqrt{q}$, define $\mathsf{pow}(x)$ as

$$\mathsf{pow}(x) := \sum_{i=0}^{\lfloor \log x \rfloor} 2^i \cdot X^i \in \mathcal{R}_q \tag{6.7}$$

and $\vartheta \in \mathcal{R}_q$ be the binary polynomial such that

$$\langle \mathsf{pow}(B^2), \vartheta \rangle = B^2 - \|\mathbf{s}\|^2. \tag{6.8}$$

We will commit to ϑ and prove that (i) it contains binary coefficients and (ii) Equation 6.8 holds over integers. These two statements imply that $0 \leqslant \langle \mathsf{pow}(B^2), \vartheta \rangle = B^2 - \|\mathbf{s}\|^2$ which is what we want.

The first statement can be proven using the techniques in Section 6.2. For the latter one, we follow the strategy as before, i.e. we want to prove that (6.8) holds over $\mathbb{Z}_q$ and that no wrap-around modulo q occurs. Hence, we show that the constant coefficient of the polynomial

$$\sigma\left(\mathsf{pow}(B^2)\right) \cdot \vartheta + \sigma(\mathbf{s})^T\mathbf{s} - B^2$$

is equal to zero. Again, this is a quadratic relation (with an automorphism) which can be proven as in Section 5.2.3. This proves that (6.8) is true over $\mathbb{Z}_q$. Next, by doing the approximate L_2 proof on $\mathbf{s}$, i.e. proving $\|\mathbf{s}\| \leqslant \psi \cdot B$ and assuming that ϑ has binary coefficients, we want to choose q such that

$$|\langle \mathsf{pow}(B^2), \vartheta \rangle + \|\mathbf{s}\|^2 - B^2| \leqslant (\psi^2 + 3) \cdot B^2 < q.$$

Then, we are sure there is no overflow modulo q and thus (6.8) holds over integers.

6.4 TOOLBOX FOR PROVING LATTICE RELATIONS

We describe a general protocol to prove various quadratic relations by combining all the techniques introduced in the previous sections. Namely, we want to prove knowledge of the secret messages $(\mathbf{s}_1, \mathbf{m}) \in \mathcal{R}_q^{m_1+\ell}$ which satisfy all the following conditions (below we define $\sigma := \sigma_{-1}$):

1. *Quadratic relations over $\mathcal{R}_q$ with automorphisms.* For $i \in [N]$ and public triples $(\mathbf{R}_{i,2}, \mathbf{r}_{i,1}, r_{i,0}) \in \mathcal{R}_q^{2(m_1+\ell)\times 2(m_1+\ell)} \times \mathcal{R}_q^{2(m_1+\ell)} \times \mathcal{R}_q$, we have:

$$\langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma^T \mathbf{R}_{i,2} \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma + \mathbf{r}_{i,1}^T \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma + r_{i,0} = 0. \tag{6.9}$$

2. *Quadratic relations over $\mathbb{Z}_q$ with automorphisms.* For $i \in [M]$ and public triples $(\mathbf{R}'_{i,2}, \mathbf{r}'_{i,1}, r'_{i,0}) \in \mathcal{R}_q^{2(m_1+\ell)\times 2(m_1+\ell)} \times \mathcal{R}_q^{2(m_1+\ell)} \times \mathcal{R}_q$:

$$\text{const. coeff. of } \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma^T \mathbf{R}'_{i,2} \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma + \mathbf{r}'^T_{i,1} \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma + r'_{i,0} \text{ equals } 0. \tag{6.10}$$

3. *Shortness in the infinity norm.* For public $\mathbf{P}_s \in \mathcal{R}_q^{n_{\mathsf{bin}} \times m_1}, \mathbf{P}_m \in \mathcal{R}_q^{n_{\mathsf{bin}} \times \ell}$ and $\mathbf{f} \in \mathcal{R}_q^{n_{\mathsf{bin}}}$, the following polynomial vector has binary coefficients

$$\mathbf{P}_s \mathbf{s}_1 + \mathbf{P}_m \mathbf{m} + \mathbf{f} \in \{0,1\}^{n_{\mathsf{bin}} \cdot d} \tag{6.11}$$

4. *Shortness in the Euclidean norm.* For $i \in [Z]$, public bound $\mathcal{B}_i < \sqrt{q}$ and $\mathbf{E}_s^{(i)} \in \mathcal{R}_q^{n_i \times m_1}, \mathbf{E}_m^{(i)} \in \mathcal{R}_q^{n_i \times \ell}$ and $\mathbf{v}^{(i)} \in \mathcal{R}_q^{n_i}$, we have:

$$\|\mathbf{E}_s^{(i)} \mathbf{s}_1 + \mathbf{E}_m^{(i)} \mathbf{m} + \mathbf{v}^{(i)}\| \leqslant \mathcal{B}_i.$$

This is equivalent to additionally proving knowledge of the binary polynomial $\vartheta_i \in \mathcal{R}$ such that

$$\langle \mathsf{pow}(\mathcal{B}_i^2), \vartheta_i \rangle = \mathcal{B}_i^2 - \left\| \mathbf{E}_s^{(i)} \mathbf{s}_1 + \mathbf{E}_m^{(i)} \mathbf{m} + \mathbf{v}^{(i)} \right\|^2 \text{ over } \mathbb{Z}. \tag{6.12}$$

5. *Approximate Shortness.* For a public bound $\mathcal{B}'$ and $\mathbf{D}_s \in \mathcal{R}_q^{n' \times m_1}, \mathbf{D}_m \in \mathcal{R}_q^{n' \times \ell}$ and $\mathbf{u} \in \mathcal{R}_q^{n'}$, we have:

$$\|\mathbf{D}_s \mathbf{s}_1 + \mathbf{D}_m \mathbf{m} + \mathbf{u}\| \leqslant \mathcal{B}'. \tag{6.13}$$

However, we are fine with convincing the verifier that

$$\|\mathbf{D}_s \mathbf{s}_1 + \mathbf{D}_m \mathbf{m} + \mathbf{u}\|_\infty \leqslant \psi \cdot \mathcal{B}' \tag{6.14}$$

where $\psi > 1$ is a public approximation factor.

We define the corresponding relation as

$$R_{\mathsf{toolbox}} = \left\{ \begin{array}{c} (u, (\mathbf{s}_1, \mathbf{m}, \vartheta_1, \ldots, \vartheta_Z)) : (\forall i \in [N], (6.9)) \wedge (\forall i \in [M], (6.10)) \\ (6.11) \wedge (\forall i \in [Z], (6.12)) \wedge (6.13) \end{array} \right\} \tag{6.15}$$

where

$$u := \left(\begin{array}{ccc} (\mathbf{R}_{i,2}, \mathbf{r}_{i,1}, r_{i,0})_{i \in [N]}, & (\mathbf{R}'_{i,2}, \mathbf{r}'_{i,1}, r'_{i,0})_{i \in [M]}, & (\mathbf{P}_s, \mathbf{P}_m, \mathbf{f}), \\ \left(\mathbf{E}_s^{(i)}, \mathbf{E}_m^{(i)}, \mathbf{v}^{(i)}, \mathcal{B}_i\right)_{i \in [Z]}, & & \left(\mathbf{D}_s, \mathbf{D}_m, \mathbf{u}, \mathcal{B}'\right) \end{array} \right). \tag{6.16}$$

As mentioned in the fifth statement, we are only interested in proving the norm approximately. Hence, our protocol will be sound with respect to the following relaxed relation:

$$R^{\psi}_{\mathsf{toolbox}} = \left\{ \begin{array}{c} (u, (\mathbf{s}_1, \mathbf{m}, \vartheta_1, \ldots, \vartheta_Z)) : (\forall i \in [N], (6.9)) \wedge (\forall i \in [M], (6.10)) \\ (6.11) \wedge (\forall i \in [Z], (6.12)) \wedge (6.14) \end{array} \right\} \tag{6.17}$$

where the statement u is defined identically as above.

In order to prove the aforementioned statements, we commit to the secrets $((\mathbf{s}_1, \vartheta_1, \ldots, \vartheta_Z), \mathbf{m})$ using the ABDLOP commitment from Section 4.1. Concretely, we sample randomness $\mathbf{s}_2 \leftarrow \chi^{m_2}$ and compute

$$\begin{bmatrix} \mathbf{t}_A \\ \mathbf{t}_B \end{bmatrix} := \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \vartheta_1 \\ \vdots \\ \vartheta_Z \end{bmatrix} + \begin{bmatrix} \mathbf{A}_2 \\ \mathbf{B} \end{bmatrix} \mathbf{s}_2 + \begin{bmatrix} \mathbf{0} \\ \mathbf{m} \end{bmatrix}$$

For proving the third and fourth statements, we apply the approximate range proof strategy for both L_2 and L_∞ norms. This requires us to commit

to two masking vectors $\mathbf{y}_3 \leftarrow D^{256}_{\mathfrak{s}_3}, \mathbf{y}_4 \leftarrow D^{256}_{\mathfrak{s}_4}$ and two signs $\beta_3, \beta_4 \leftarrow \{-1, 1\}$ for bimodal Gaussian rejection sampling. Concretely,

$$\begin{bmatrix} \mathbf{t}_y \\ \mathbf{t}_\beta \end{bmatrix} := \begin{bmatrix} \mathbf{B}_y \\ \mathbf{B}_\beta \end{bmatrix} \mathbf{s}_2 + \begin{bmatrix} \mathbf{y}_3 \\ \mathbf{y}_4 \\ \beta_3 \\ \beta_4 \end{bmatrix}$$

Overall, $(\mathbf{t}_A, \mathbf{t}_B \parallel \mathbf{t}_y \parallel \mathbf{t}_\beta)$ is the ABDLOP commitment to $(\mathbf{s}_1^*, \mathbf{m}^*)$ where

$$\mathbf{s}_1^* := \mathbf{s}_1 \parallel \vartheta_1 \parallel \cdots \parallel \vartheta_Z \qquad \text{and} \qquad \mathbf{m}^* := \mathbf{m} \parallel \mathbf{y}_3 \parallel \mathbf{y}_4 \parallel \beta_3 \parallel \beta_4.$$

Clearly, if $\|\mathbf{s}_1\| \leqslant \alpha$ then $\|\mathbf{s}_1^*\| \leqslant \sqrt{\alpha^2 + Zd}$. Our goal will be to reduce all the five statements above to proving quadratic relations in $(\mathbf{s}_1^*, \mathbf{m}^*)$, or more concretely, in $\langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma$. We cover them one by one, but first we start with introducing notation.

6.4.1 *Notation*

To begin with, we recall the $\mathbf{U}$ matrix defined in Section 5.2.3.2. Namely, $\mathbf{U} \in \mathcal{R}_q^{2(m_1+Z+\ell+512/d+2)\times 2(m_1+Z+\ell+512/d+2)}$ is the public matrix such that

$$\sigma(\langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma) = \mathbf{U}\langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma.$$

Now, we will write all the variables in $(\mathbf{s}_1^*, \mathbf{m}^*)$ as a linear combination of elements in $\langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma$. We start with $(\mathbf{s}_1, \mathbf{m})$ and observe that

$$\begin{aligned} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \end{bmatrix} &= \begin{bmatrix} \mathbf{I}_{m_1} & \mathbf{0}_{m_1\times Z} & \mathbf{0}_{m_1\times \ell} & \mathbf{0}_{m_1\times(512/d+2)} \\ \mathbf{0}_{\ell\times m_1} & \mathbf{0}_{\ell\times Z} & \mathbf{I}_\ell & \mathbf{0}_{\ell\times(512/d+2)} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1^* \\ \mathbf{m}^* \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{I}_{m_1} & \mathbf{0}_{m_1\times Z} & \mathbf{0}_{m_1\times \ell} & \mathbf{0}_{m_1\times(512/d+2)} \\ \mathbf{0}_{\ell\times m_1} & \mathbf{0}_{\ell\times Z} & \mathbf{I}_\ell & \mathbf{0}_{\ell\times(512/d+2)} \end{bmatrix} \mathbf{J}\langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma \end{aligned} \tag{6.18}$$

where $\mathbf{J} := \mathbf{J}_{m_1+Z+\ell+512/d+2,2}$ as defined in Lemma 5.2.1. Similarly, we have

$$\langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma = \begin{bmatrix} \mathbf{I}_{2m_1} & \mathbf{0}_{2m_1\times 2Z} & \mathbf{0}_{2m_1\times 2\ell} & \mathbf{0}_{2m_1\times 2(512/d+2)} \\ \mathbf{0}_{2\ell\times 2m_1} & \mathbf{0}_{2\ell\times 2Z} & \mathbf{I}_{2\ell} & \mathbf{0}_{2\ell\times 2(512/d+2)} \end{bmatrix} \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma. \tag{6.19}$$

For convenience, we define

$$\mathbf{K}_s := \begin{bmatrix} \mathbf{I}_{m_1} & \mathbf{0}_{m_1\times Z} & \mathbf{0}_{m_1\times \ell} & \mathbf{0}_{m_1\times (512/d+2)} \\ \mathbf{0}_{\ell\times m_1} & \mathbf{0}_{\ell\times Z} & \mathbf{I}_{\ell} & \mathbf{0}_{\ell\times (512/d+2)} \end{bmatrix} \mathbf{J}$$

and

$$\mathbf{K}_{s,\sigma} := \begin{bmatrix} \mathbf{I}_{2m_1} & \mathbf{0}_{2m_1\times 2Z} & \mathbf{0}_{2m_1\times 2\ell} & \mathbf{0}_{2m_1\times 2(512/d+2)} \\ \mathbf{0}_{2\ell\times 2m_1} & \mathbf{0}_{2\ell\times 2Z} & \mathbf{I}_{2\ell} & \mathbf{0}_{2\ell\times 2(512/d+2)} \end{bmatrix}.$$

Further, for $i \in [Z]$ we have

$$\vartheta_i = \begin{bmatrix} \mathbf{0}_{1\times (m_1+i-1)} & 1 & \mathbf{0}_{1\times (Z-i+\ell+512/d+2)} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1^* \\ \mathbf{m}^* \end{bmatrix}.$$

Hence, denote

$$\mathbf{k}_{\vartheta_i}^T := \begin{bmatrix} \mathbf{0}_{1\times (m_1+i-1)} & 1 & \mathbf{0}_{1\times (Z-i+\ell+512/d+2)} \end{bmatrix} \mathbf{J}.$$

Next, note that

$$\begin{bmatrix} \mathbf{y}_3 \\ \mathbf{y}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{256/d\times (m_1+Z+m_1)} & \mathbf{I}_{256/d} & \mathbf{0}_{256/d\times 256/d} & \mathbf{0}_{256/d\times 2} \\ \mathbf{0}_{256/d\times (m_1+Z+m_1)} & \mathbf{0}_{256/d} & \mathbf{I}_{256/d\times 256/d} & \mathbf{0}_{256/d\times 2} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1^* \\ \mathbf{m}^* \end{bmatrix}.$$

Thus, we define matrices $\mathbf{K}_{y_3}$ and $\mathbf{K}_{y_4}$ as follows:

$$\begin{bmatrix} \mathbf{K}_{y_3} \\ \mathbf{K}_{y_4} \end{bmatrix} := \begin{bmatrix} \mathbf{0}_{256/d\times (m_1+Z+m_1)} & \mathbf{I}_{256/d} & \mathbf{0}_{256/d\times 256/d} & \mathbf{0}_{256/d\times 2} \\ \mathbf{0}_{256/d\times (m_1+Z+m_1)} & \mathbf{0}_{256/d} & \mathbf{I}_{256/d\times 256/d} & \mathbf{0}_{256/d\times 2} \end{bmatrix} \mathbf{J}.$$

Finally, we focus on β_3, β_4. Clearly

$$\begin{bmatrix} \beta_3 \\ \beta_4 \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{1\times (m_1+Z+m_1+512/d)} & 1 & 0 \\ \mathbf{0}_{1\times (m_1+Z+m_1+512/d)} & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{s}_1^* \\ \mathbf{m}^* \end{bmatrix}.$$

Consequently, we define vectors $\mathbf{k}_{\beta_3}, \mathbf{k}_{\beta_4}$ as follows:

$$\begin{bmatrix} \mathbf{k}_{\beta_3}^T \\ \mathbf{k}_{\beta_4}^T \end{bmatrix} := \begin{bmatrix} \mathbf{0}_{1\times (m_1+Z+m_1+512/d)} & 1 & 0 \\ \mathbf{0}_{1\times (m_1+Z+m_1+512/d)} & 0 & 1 \end{bmatrix} \mathbf{J}.$$

In conclusion, we obtain the following equalities:

$$\begin{aligned}
&\begin{bmatrix}\mathbf{s}_1\\ \mathbf{m}\end{bmatrix} = \mathbf{K}_s\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma\\
&\langle \mathbf{s}_1 \parallel \mathbf{m}\rangle_\sigma = \mathbf{K}_{s,\sigma}\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma\\
&\forall i \in [Z], \vartheta_i = \mathbf{k}_{\vartheta_i}^T\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma\\
&\mathbf{y}_3 = \mathbf{K}_{y_3}\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma\\
&\mathbf{y}_4 = \mathbf{K}_{y_4}\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma\\
&\beta_3 = \mathbf{k}_{\beta_3}^T\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma\\
&\beta_4 = \mathbf{k}_{\beta_4}^T\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma.
\end{aligned}$$

6.4.2 *Proving Quadratic Relations*

We concentrate on the first two statements. Using the notation above, we observe that proving Equation 6.9 is equivalent to proving

$$\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma^T \mathbf{K}_{s,\sigma}^T \mathbf{R}_{i,2} \mathbf{K}_{s,\sigma} \langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma + \mathbf{r}_{i,1}^T \mathbf{K}_{s,\sigma} \langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma + r_{i,0} = 0 \quad (6.20)$$

which is a quadratic equation in $\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma$ covered in Section 5.2.1. Similarly, for (6.10) we need to show that the constant coefficient of

$$\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma^T \mathbf{K}_{s,\sigma}^T \mathbf{R}'_{i,2} \mathbf{K}_{s,\sigma} \langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma + \mathbf{r}'^T_{i,1} \mathbf{K}_{s,\sigma} \langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma + r'_{i,0} \quad (6.21)$$

is equal to zero, which we discussed in Section 5.2.3. In conclusion, we can reduce the first two statements to proving quadratic relations covered by Π_{eval} from Section 5.2.3.

6.4.3 *Proving Exact Shortness*

We now focus on the third and fourth statement. We follow the strategy described in Sections 6.2 and 6.3 and first start by proving that

$$\begin{aligned}
&\mathbf{P}_s \mathbf{s}_1 + \mathbf{P}_m \mathbf{m} + \mathbf{f}\\
&\mathbf{E}_s^{(i)} \mathbf{s}_1 + \mathbf{E}_m^{(i)} \mathbf{m} + \mathbf{v}^{(i)} \text{ for } i = 1, 2, \ldots, Z\\
&\boldsymbol{\vartheta} := (\vartheta_1, \ldots, \vartheta_Z)
\end{aligned}$$

are approximately small in the L_2 norm using the techniques from Section 6.1.2. Namely, define $n_{\mathsf{ex}} := n_{\mathsf{bin}} + \sum_{i=1}^{Z} n_i + Z$ and

$$\mathbf{s}_3 := \begin{bmatrix} \mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f} \\ \mathbf{E}_s^{(1)}\mathbf{s}_1 + \mathbf{E}_m^{(1)}\mathbf{m} + \mathbf{v}^{(1)} \\ \vdots \\ \mathbf{E}_s^{(Z)}\mathbf{s}_1 + \mathbf{E}_m^{(Z)}\mathbf{m} + \mathbf{v}^{(Z)} \\ \boldsymbol{\vartheta} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_s & \mathbf{P}_m & \mathbf{0}_{n_{\mathsf{bin}} \times Z} \\ \mathbf{E}_s^{(1)} & \mathbf{E}_m^{(1)} & \mathbf{0}_{n_1 \times Z} \\ \vdots & \vdots & \vdots \\ \mathbf{E}_s^{(Z)} & \mathbf{E}_m^{(Z)} & \mathbf{0}_{n_Z \times Z} \\ \mathbf{0}_{Z \times m_1} & \mathbf{0}_{Z \times \ell} & \mathbf{I}_Z \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \\ \boldsymbol{\vartheta} \end{bmatrix} + \begin{bmatrix} \mathbf{f} \\ \mathbf{v}^{(1)} \\ \vdots \\ \mathbf{v}^{(Z)} \end{bmatrix} \tag{6.22}$$

where $\mathbf{s}_3 \in \mathcal{R}_q^{n_{\mathsf{ex}}}$. Then,

$$\|\mathbf{s}_3\| \leqslant \sqrt{(n_{\mathsf{bin}} + Z)d + \sum_{i=1}^{Z} \mathcal{B}_i^2}.$$

Now, given a matrix $R \leftarrow \mathsf{Bin}_2^{256 \times n_{\mathsf{ex}} d}$ from the verifier as a challenge, we compute the polynomial vector $\mathbf{z}_3 \in \mathcal{R}_q^{256/d}$ which satisfies

$$\vec{z}_3 := \vec{y}_3 + \beta_3 \cdot R\vec{s}_3. \tag{6.23}$$

After applying the bimodal rejection sampling, we output $\mathbf{z}_3$. The verifier checks whether $\|\mathbf{z}_3\| \leqslant \varrho\mathfrak{s}_3\sqrt{256}$ where ϱ satisfies $\varrho^{256} \cdot e^{128(1-\varrho^2)} = 2^{-\kappa}$. Then, we need to prove well-formedness of $\mathbf{z}_3$. For $i \in [256]$, denote $\mathbf{e}_i \in \mathcal{R}_q^{256/d}$ to be the binary polynomial vector such that $\vec{e}_i \in \{0,1\}^{256}$ and it has one 1 exactly in the i-th position. Also, let $\mathbf{r}_i \in \mathcal{R}_q^{n_{\mathsf{ex}}}$ be the vector so that $\vec{r}_i$ is the i-th row of the matrix R. Then, (6.23) holds if and only for all $i \in [256]$:

$$\langle \mathbf{z}_3, \mathbf{e}_i \rangle = \langle \mathbf{y}_3, \mathbf{e}_i \rangle + \beta_3 \cdot \langle \mathbf{r}_i, \mathbf{s}_3 \rangle.$$

That is, the constant coefficient of the following polynomial vanishes:

$$\sigma(\mathbf{e}_i)^T\mathbf{y}_3 + \beta_3 \cdot \sigma(\mathbf{r}_i)^T\mathbf{s}_3 - \langle \mathbf{z}_3, \mathbf{e}_i \rangle$$

which can be written equivalently using the notation from Section 6.4.1 as

$$\langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma^T \mathbf{R}'_{M+i,2} \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma + \mathbf{r}'^T_{M+i,1} \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma + r'_{M+i,0}$$

where

$$\begin{aligned}
\mathbf{R}'_{M+i,2} &:= \mathbf{k}_{\beta_3}\sigma(\mathbf{r}_i)^T \begin{bmatrix} \mathbf{P}_s & \mathbf{P}_m & \mathbf{0}_{n_{\mathsf{bin}}\times Z} \\ \mathbf{E}_s^{(1)} & \mathbf{E}_m^{(1)} & \mathbf{0}_{n_1\times Z} \\ \vdots & \vdots & \vdots \\ \mathbf{E}_s^{(Z)} & \mathbf{E}_m^{(Z)} & \mathbf{0}_{n_Z\times Z} \\ \mathbf{0}_{Z\times m_1} & \mathbf{0}_{Z\times \ell} & \mathbf{I}_Z \end{bmatrix} \begin{bmatrix} \mathbf{K}_s \\ \mathbf{k}_{\vartheta_1}^T \\ \cdots \\ \mathbf{k}_{\vartheta_Z}^T \end{bmatrix} \\
\mathbf{r}'^T_{M+i,1} &:= \sigma(\mathbf{e}_i)^T\mathbf{K}_{y_3} + \sigma(\mathbf{r}_i)^T \begin{bmatrix} \mathbf{f} \\ \mathbf{v}^{(1)} \\ \vdots \\ \mathbf{v}^{(Z)} \end{bmatrix} \mathbf{k}_{\beta_3}^T \\
r'_{M+i,0} &:= -\langle \mathbf{z}_3, \mathbf{e}_i\rangle.
\end{aligned} \tag{6.24}$$

Hence, we prove this relation using the techniques from Section 5.2.3.

Not to mention the fact that we also need to prove that $\beta_3 \in \{-1, 1\}$. We do this by proving that $\beta_3^2 = 1$ and the constant coefficient of $X^i\beta_3$ is equal to zero for $i = 1, \ldots, d-1$. The former statement can be written as a quadratic equation:

$$\beta_3^2 - 1 = \langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma^T \mathbf{R}_{N+1,2}\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma + \mathbf{r}_{N+1,1}^T\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma + r_{N+1,0} = 0$$

where

$$\mathbf{R}_{N+1,2} := \mathbf{k}_{\beta_3}\mathbf{k}_{\beta_3}^T, \quad \mathbf{r}_{N+1,1} = \mathbf{0}, \quad r_{N+1,0} = -1. \tag{6.25}$$

To prove the latter statement, note that $X^i \cdot \beta_3$ equals to

$$\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma^T \mathbf{R}'_{M+256+i,2}\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma + \mathbf{r}'^T_{M+256+i,1}\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma + r'_{M+256+i,0}$$

for

$$\mathbf{R}_{M+256+i,2} := \mathbf{0}, \quad \mathbf{r}_{M+256+i,1} = X^i\mathbf{k}_{\beta_3}^T, \quad r_{M+256+i,0} = 0. \tag{6.26}$$

Now, assuming well-formedness of $\mathbf{z}_3$, we can convince the verifier that

$$\|\mathbf{s}_3\| \leqslant \mathcal{B}_{\mathsf{arp}} := \psi \cdot \sqrt{(n_{\mathsf{bin}} + Z)d + \sum_{i=1}^{Z} \mathcal{B}_i^2}$$

for some public $\psi > 1$. In particular, we get that

$$\begin{aligned}
&\|\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}\| \leqslant \mathcal{B}_{\mathsf{arp}}, \\
&\forall i \in [Z], \left\|\mathbf{E}_s^{(i)}\mathbf{s}_1 + \mathbf{E}_m^{(i)}\mathbf{m} + \mathbf{v}^{(i)}\right\| \leqslant \mathcal{B}_{\mathsf{arp}} \text{ and} \|\vartheta_i\| \leqslant \mathcal{B}_{\mathsf{arp}}.
\end{aligned}$$

6.4.3.1 *Infinity Norm*

Now, we prove that $\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}$ and $\vartheta_1, \ldots, \vartheta_Z$ have binary coefficients. To prove the former statement, we show

$$\langle \mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}, \mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f} - \mathbf{x} \rangle = 0 \text{ over } \mathbb{Z}$$

where

$$\mathbf{x} := (1 + X + \ldots + X^{d-1}, \ldots, 1 + X + \ldots + X^{d-1}) \in \mathcal{R}_q^{n_{\mathsf{bin}}}.$$

We first prove this equation modulo q. By Lemma 5.1.10, this boils down to showing that the constant coefficient of

$$\begin{bmatrix} \sigma(\mathbf{s}_1)^T & \sigma(\mathbf{m})^T \end{bmatrix} \begin{bmatrix} \sigma(\mathbf{P}_s)^T\mathbf{P}_s & \sigma(\mathbf{P}_s)^T\mathbf{P}_m \\ \sigma(\mathbf{P}_m)^T\mathbf{P}_s & \sigma(\mathbf{P}_m)^T\mathbf{P}_m \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \end{bmatrix}$$
$$+ \begin{bmatrix} \sigma(\mathbf{s}_1)^T & \sigma(\mathbf{m})^T \end{bmatrix} \begin{bmatrix} \sigma(\mathbf{P}_s)^T \\ \sigma(\mathbf{P}_m)^T \end{bmatrix} (\mathbf{f} - \mathbf{x}) + \sigma(\mathbf{f})^T \begin{bmatrix} \mathbf{P}_s & \mathbf{P}_m \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \end{bmatrix} + \langle \mathbf{f}, \mathbf{f} - \mathbf{x} \rangle$$

vanishes. Now, we use the property of the $\mathbf{U}$ matrix defined in Section 6.4.1, i.e.

$$\begin{bmatrix} \sigma(\mathbf{s}_1) \\ \sigma(\mathbf{m}) \end{bmatrix} = \sigma\left(\mathbf{K}_s\right) \sigma\left(\langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma\right) = \sigma\left(\mathbf{K}_s\right) \mathbf{U} \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma.$$

Then, the polynomial above can be written equivalently using the notation from Section 6.4.1 as

$$\langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma^T \mathbf{R}'_{M+256+d,2} \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma + \mathbf{r}'^T_{M+256+d,1} \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma + r'_{M+256+d,0}$$

where

$$\begin{aligned} \mathbf{R}'_{M+256+d,2} &:= \mathbf{U}^T \sigma(\mathbf{K}_s)^T \begin{bmatrix} \sigma(\mathbf{P}_s)^T\mathbf{P}_s & \sigma(\mathbf{P}_s)^T\mathbf{P}_m \\ \sigma(\mathbf{P}_m)^T\mathbf{P}_s & \sigma(\mathbf{P}_m)^T\mathbf{P}_m \end{bmatrix} \mathbf{K}_s \\ \mathbf{r}'^T_{M+256+d,1} &:= (\mathbf{f}^T - \mathbf{x}^T) \begin{bmatrix} \sigma(\mathbf{P}_s) & \sigma(\mathbf{P}_m) \end{bmatrix} \sigma(\mathbf{K}_s)\mathbf{U} + \sigma(\mathbf{f})^T \begin{bmatrix} \mathbf{P}_s & \mathbf{P}_m \end{bmatrix} \mathbf{K}_s \\ r'_{M+256+d,0} &:= \langle \mathbf{f}, \mathbf{f} - \mathbf{x} \rangle. \end{aligned} \tag{6.27}$$

Thus, we prove this relation using the techniques from Section 5.2.3. Now, assuming that

$$\mathcal{B}_{\mathsf{arp}}^2 + \mathcal{B}_{\mathsf{arp}}\sqrt{n_{\mathsf{bin}}d} < q$$

we can deduce similarly as in (6.6) that

$$|\langle \mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}, \mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f} - \mathbf{x}\rangle| < q.$$

Therefore, we conclude that the inner product equals zero over integers. By Lemma 6.2.1, this implies that the vector $\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}$ has binary coefficients.

Next, we focus on $\vartheta_1, \ldots, \vartheta_Z$. Similarly as before, we want to prove that $\langle \vartheta_i, \vartheta_i - \sum_{i=0}^{d-1} X^i \rangle = 0$ over integers. In order to prove it in $\mathbb{Z}_q$, we need to show that the constant coefficient of

$$\sigma(\vartheta_i)\left(\vartheta_i - \sum_{i=0}^{d-1} X^i\right)$$

is equal to zero. Using the notation from Section 6.4.1, we can write this polynomial as

$$\langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma^T \mathbf{R}'_{M+256+d+i,2} \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma + \mathbf{r}'^T_{M+256+d+i,1} \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma$$

where

$$\mathbf{R}'_{M+256+d+i,2} := \mathbf{U}^T \sigma(\mathbf{k}_{\vartheta_i}) \mathbf{k}_{\vartheta_i}^T, \quad \mathbf{r}'^T_{M+256+d+i,1} := -\left(\sum_{i=0}^{d-1} X^i\right) \cdot \sigma(\mathbf{k}_{\vartheta_i})^T \mathbf{U}. \tag{6.28}$$

Thus, we prove this relation using the techniques from Section 5.2.3. Now, if

$$\mathcal{B}_{\mathsf{arp}}^2 + \mathcal{B}_{\mathsf{arp}}\sqrt{d} < q$$

then we conclude that $\langle \vartheta_i, \vartheta_i - \sum_{i=0}^{d-1} X^i \rangle = 0$ over integers. Hence, ϑ_i has binary coefficients.

6.4.3.2 *Euclidean Norm*

Now, we turn to proving that for every $i \in [Z]$,

$$\langle \mathsf{pow}(\mathcal{B}_i^2), \vartheta_i \rangle = \mathcal{B}_i^2 - \left\| \mathbf{E}_s^{(i)}\mathbf{s}_1 + \mathbf{E}_m^{(i)}\mathbf{m} + \mathbf{v}^{(i)} \right\|^2 \text{ over } \mathbb{Z}_q.$$

Note that this implies that the equation holds over integers if the modulus q is appropriately large. Indeed, we observe

$$\left| \langle \mathsf{pow}(\mathcal{B}_i^2), \vartheta_i \rangle + \left\| \mathbf{E}_s^{(i)}\mathbf{s}_1 + \mathbf{E}_m^{(i)}\mathbf{m} + \mathbf{v}^{(i)} \right\|^2 - \mathcal{B}_i^2 \right| \leqslant 3\mathcal{B}_i^2 + \mathcal{B}_{\mathsf{arp}}^2.$$

Hence, if $q > 3\mathcal{B}_i^2 + \mathcal{B}_{\mathsf{arp}}^2$ for all $i = 1, 2, \ldots, Z$ then we are done.

To prove the initial equation over $\mathbb{Z}_q$, we apply the strategy from Section 6.3 and write

$$\langle \mathsf{pow}(\mathcal{B}_i^2), \vartheta_i \rangle + \langle \mathbf{E}_s^{(i)} \mathbf{s}_1 + \mathbf{E}_m^{(i)} \mathbf{m} + \mathbf{v}^{(i)}, \mathbf{E}_s^{(i)} \mathbf{s}_1 + \mathbf{E}_m^{(i)} \mathbf{m} + \mathbf{v}^{(i)} \rangle - \mathcal{B}_i^2 = 0.$$

This is equivalent to proving that the constant coefficient of the following polynomial equals to zero

$$\begin{aligned}
&\begin{bmatrix} \sigma(\mathbf{s}_1)^T & \sigma(\mathbf{m})^T \end{bmatrix} \begin{bmatrix} \sigma(\mathbf{E}_s^{(i)})^T \mathbf{E}_s^{(i)} & \sigma(\mathbf{E}_s^{(i)})^T \mathbf{E}_m^{(i)} \\ \sigma(\mathbf{E}_m^{(i)})^T \mathbf{E}_s^{(i)} & \sigma(\mathbf{E}_m^{(i)})^T \mathbf{E}_m^{(i)} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \end{bmatrix} \\
&+ \begin{bmatrix} \sigma(\mathbf{s}_1)^T & \sigma(\mathbf{m})^T \end{bmatrix} \begin{bmatrix} \sigma(\mathbf{E}^{(i)})^T \\ \sigma(\mathbf{E}_m^{(i)})^T \end{bmatrix} \mathbf{v}^{(i)} + \sigma(\mathbf{v}^{(i)})^T \begin{bmatrix} \mathbf{E}^{(i)} & \mathbf{E}_m^{(i)} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \end{bmatrix} + \langle \mathbf{v}^{(i)}, \mathbf{v}^{(i)} \rangle \\
&+ \sigma \left(\mathsf{pow}(\mathcal{B}_i^2) \right) \vartheta_i - \mathcal{B}_i^2.
\end{aligned}$$

This polynomial can be alternatively written using the notation from Section 6.4.1 as:

$$\begin{aligned}
\langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma^T \mathbf{R}'_{M+256+d+Z+i,2} \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma + \mathbf{r}'^T_{M+256+d+Z+i,1} \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma \\
+ r'_{M+256+d+Z+i,0} \in \mathcal{R}_q
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{R}'_{M+256+d+Z+i,2} &:= \mathbf{U}^T \sigma(\mathbf{K}_s)^T \begin{bmatrix} \sigma(\mathbf{E}_s^{(i)})^T \mathbf{E}_s^{(i)} & \sigma(\mathbf{E}_s^{(i)})^T \mathbf{E}_m^{(i)} \\ \sigma(\mathbf{E}_m^{(i)})^T \mathbf{E}_s^{(i)} & \sigma(\mathbf{E}_m^{(i)})^T \mathbf{E}_m^{(i)} \end{bmatrix} \mathbf{K}_s \\
\mathbf{r}'^T_{M+256+d+Z+i,1} &:= \mathbf{v}^{(i)^T} \begin{bmatrix} \sigma(\mathbf{E}_s^{(i)}) & \sigma(\mathbf{E}_m^{(i)}) \end{bmatrix} \sigma(\mathbf{K}_s) \mathbf{U} + \sigma(\mathbf{v}^{(i)})^T \begin{bmatrix} \mathbf{E}_s^{(i)} & \mathbf{E}_m^{(i)} \end{bmatrix} \mathbf{K}_s \\
&\quad + \sigma \left(\mathsf{pow}(\mathcal{B}_i^2) \right) \mathbf{k}_{\vartheta_i}^T \\
r'_{M+256+d+Z+i,0} &:= \langle \mathbf{v}^{(i)}, \mathbf{v}^{(i)} \rangle - \mathcal{B}_i^2.
\end{aligned} \tag{6.29}$$

Therefore, we prove this relation using the techniques from Section 5.2.3. Now, if (6.12) holds and ϑ_i has binary coefficients then we conclude that

$$0 \leqslant \langle \mathsf{pow}(\mathcal{B}_i^2), \vartheta_i \rangle = \mathcal{B}_i^2 - \left\| \mathbf{E}_s^{(i)} \mathbf{s}_1 + \mathbf{E}_m^{(i)} \mathbf{m} + \mathbf{v}^{(i)} \right\|^2$$

and we are done.

6.4.4 *Approximate Shortness*

Finally, we deal with the last statement, i.e. proving approximately that

$$\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\| \leqslant \mathcal{B}'.$$

The proof follows the first part of Section 6.4.3. Concretely, define

$$\mathbf{s}_4 := \mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u} = \begin{bmatrix}\mathbf{D}_s & \mathbf{D}_m\end{bmatrix}\begin{bmatrix}\mathbf{s}_1 \\ \mathbf{m}\end{bmatrix} + \mathbf{u} \in \mathcal{R}_q^{n'}. \tag{6.30}$$

Now, given a matrix $R' \leftarrow \mathsf{Bin}_2^{256\times n'd}$ from the verifier as a challenge, we compute the polynomial vector $\mathbf{z}_4 \in \mathcal{R}_q^{256/d}$ which satisfies

$$\vec{z}_4 := \vec{y}_4 + \beta_4 \cdot R'\vec{s}_4. \tag{6.31}$$

After applying the bimodal rejection sampling, we output $\mathbf{z}_4$. Since we are interested in proving the L_∞ norm approximately, the verifier checks whether $\|\mathbf{z}_4\|_\infty \leqslant \sqrt{2\kappa}\mathfrak{s}_4$ as in Section 6.1.1. Then, we need to prove well-formedness of $\mathbf{z}_4$. For $i \in [256]$, denote $\mathbf{e}_i \in \mathcal{R}_q^{256/d}$ to be the binary polynomial vector such that $\vec{e}_i \in \{0,1\}^{256}$ and it has one 1 exactly in the i-th position identically as in Section 6.4.3. Also, let $\mathbf{r}'_i \in \mathcal{R}_q^{n'}$ be the vector so that $\vec{r}'_i$ is the i-th row of the matrix R'. Then, (6.31) holds if and only for all $i \in [256]$:

$$\langle \mathbf{z}_4, \mathbf{e}_i\rangle = \langle \mathbf{y}_4, \mathbf{e}_i\rangle + \beta_4 \cdot \langle \mathbf{r}'_i, \mathbf{s}_4\rangle.$$

Alternatively, the constant coefficient of the following polynomial vanishes:

$$\sigma(\mathbf{e}_i)^T\mathbf{y}_4 + \beta_4 \cdot \sigma(\mathbf{r}'_i)^T\mathbf{s}_4 - \langle \mathbf{z}_4, \mathbf{e}_i\rangle$$

which can be written equivalently using the notation from Section 6.4.1 as

$$\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma^T \mathbf{R}'_{M+256+d+2Z+i,2}\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma + \mathbf{r}'^T_{M+256+d+2Z+i,1}\langle \mathbf{s}_1^* \parallel \mathbf{m}^*\rangle_\sigma + r'_{M+256+d+2Z+i,0}$$

where

$$\begin{aligned}
\mathbf{R}'_{M+256+d+2Z+i,2} &:= \mathbf{k}_{\beta_4}\sigma(\mathbf{r}'_i)^T\begin{bmatrix}\mathbf{D}_s & \mathbf{D}_m\end{bmatrix}\mathbf{K}_s \\
\mathbf{r}'^T_{M+256+d+2Z+i,1} &:= \sigma(\mathbf{e}_i)^T\mathbf{K}_{y_4} + \sigma(\mathbf{r}'_i)^T\mathbf{u}\mathbf{k}_{\beta_4}^T \\
r'_{M+256+d+2Z+i,0} &:= -\langle \mathbf{z}_4, \mathbf{e}_i\rangle.
\end{aligned} \tag{6.32}$$

Finally, we need to prove that $\beta_4 \in \{-1, 1\}$. As before, we do this by proving that $\beta_4^2 = 1$ and the constant coefficient of $X^i\beta_4$ is equal to zero for $i = 1, \ldots, d-1$. The first statement can be written as a quadratic equation:

$$\beta_4^2 - 1 = \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma^T \mathbf{R}_{N+2,2} \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma + \mathbf{r}_{N+2,1}^T \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma + r_{N+2,0} = 0$$

where

$$\mathbf{R}_{N+2,2} := \mathbf{k}_{\beta_4} \mathbf{k}_{\beta_4}^T, \quad \mathbf{r}_{N+2,1} = \mathbf{0}, \quad r_{N+2,0} = -1. \tag{6.33}$$

To prove the second one, observe that $X^i \cdot \beta_4$ equals to

$$\begin{aligned} \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma^T \mathbf{R}'_{M+512+d+2Z+i,2} \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma \\ + \mathbf{r}'^T_{M+512+d+2Z+i,1} \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma \\ + r'_{M+512+d+2Z+i,0} \end{aligned}$$

where

$$\mathbf{R}_{M+512+d+2Z+i,2} := \mathbf{0}, \quad \mathbf{r}_{M+512+d+2Z+i,1} = X^i \mathbf{k}_{\beta_4}^T, \quad r_{M+512+d+2Z+i,0} = 0. \tag{6.34}$$

Hence, we prove all the necessary relations using the techniques from Section 5.2.3.

6.4.5 *Main Protocol*

We summarise the strategies presented in the previous subsections and propose a commit-and-prove system $\Pi_{\mathsf{tbox}} = (\mathsf{ABDLOP}, \mathcal{P}, \mathcal{V})$ for the relation R_{toolbox} in Figure 6.3. In the protocol, we run Π_{eval} defined in Section 5.2.3 for proving quadratic relations.

We provide security properties of the commit-and-prove $\Pi_{\mathsf{tbox}}^{\infty}$ below.

Theorem 6.4.1. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *and* $\mathsf{Rej}^{(3)} = \mathsf{Rej}^{(4)} = \mathsf{Rej}_1$ *as defined in Figure 3.2. Fix standard deviations*

$$\mathfrak{s}_1 = \gamma_1 \eta \sqrt{\alpha^2 + Zd}, \qquad \mathfrak{s}_2 = \gamma_2 \eta \nu \sqrt{m_2 d},$$

$$\mathfrak{s}_3 = \gamma_3 \sqrt{337} \sqrt{(n_{\mathsf{bin}} + Z)d + \sum_{i=1}^{Z} \mathcal{B}_i^2}, \qquad \mathfrak{s}_4 = \gamma_4 \sqrt{337} \mathcal{B}'$$

for some $\gamma_1, \gamma_2, \gamma_3, \gamma_4 > 0$ *and define*

$$M_i := \begin{cases} \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2} \right) & \text{for } i = 1, 2 \\ \exp\left(\frac{1}{2\gamma_i^2} \right) & \text{for } i = 3, 4. \end{cases}$$

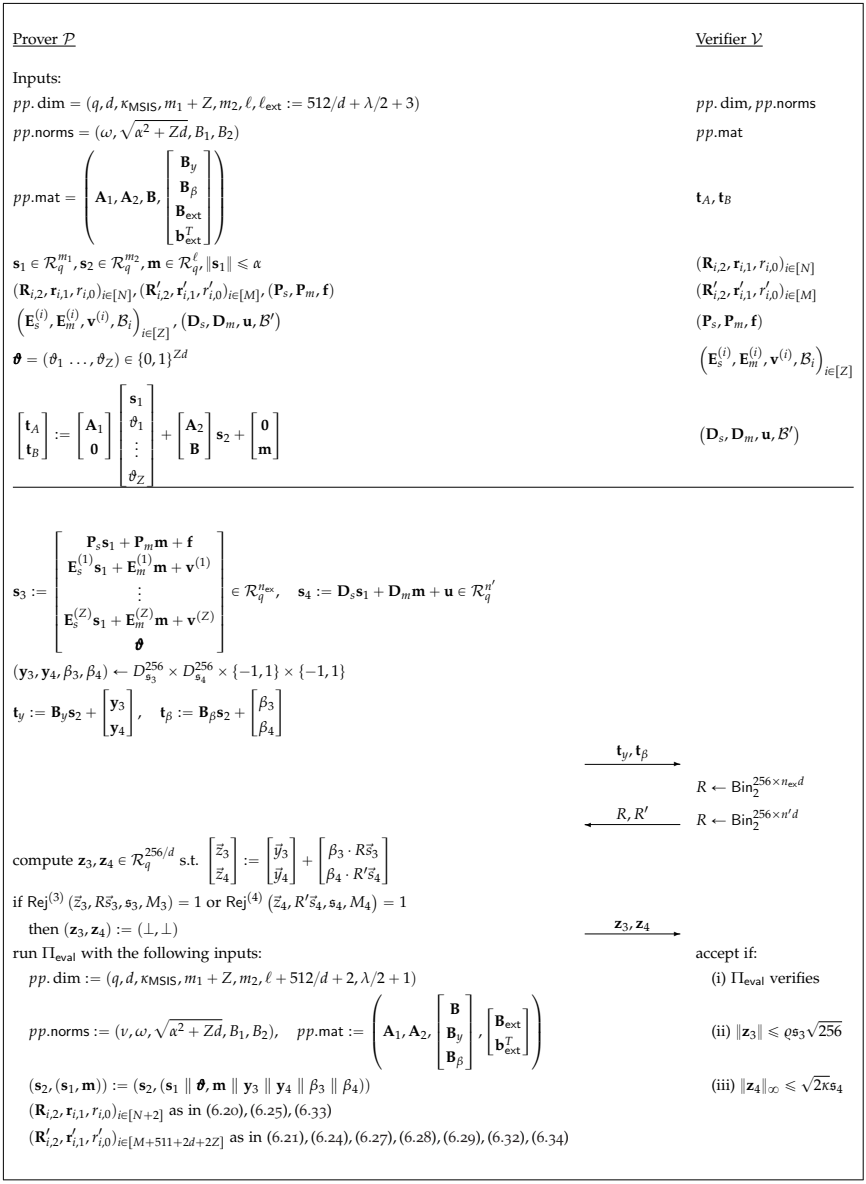

FIGURE 6.3: Commit-and-prove system Π_{tbox} for the relation R_{toolbox} in (6.15). Here, we define $n_{\mathsf{ex}} := n_{\mathsf{bin}} + \sum_{i=1}^{Z} n_i + Z$ and ϱ which satisfies $\varrho^{256} \cdot e^{128(1-\varrho^2)} = 2^{-\kappa}$.

Suppose that $(m_1 + Z)d \geqslant 5\kappa$ *and* $m_2 d \geqslant 5\kappa$. *Then, the commit-and-prove system* Π_{tbox} *for the relation* R_{toolbox} *has statistical completeness with correctness error* $1 - \frac{1}{M_1 M_2 M_3 M_4} + 2^{-127}$.

Proof. First of all, note that

$$\|R\vec{s}_3\| \leqslant \sqrt{337} \cdot \sqrt{(n_{\mathsf{bin}} + Z)d + \sum_{i=1}^{Z} \mathcal{B}_i^2} \text{ and } \|R'\vec{s}_4\| \leqslant \sqrt{337}\mathcal{B}'$$

with probability at least $1 - 2^{-127}$ by Lemma 3.2.4 and the union bound. Assuming these inequalities hold, the probability that an honest prover succeeds in all four rejection sampling algorithms is $1/(M_1 M_2 M_3 M_4)$ by Lemmas 3.3.2 and 3.3.3. In terms of verification equations, $\|\mathbf{z}_3\| > \varrho\mathfrak{s}_3\sqrt{256}$ or $\|\mathbf{z}_4\|_\infty > \sqrt{2\kappa}\mathfrak{s}_4$ with probability at most $256 \cdot 2e^{-\kappa} + 2^{-\kappa}$ by Lemma 3.2.2. All the other verification equations hold by the discussion above. □

Theorem 6.4.2. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}^{(2)} = \mathsf{Rej}_0$ *and* $\mathsf{Rej}^{(3)} = \mathsf{Rej}^{(4)} = \mathsf{Rej}_1$ *as defined in Figure 3.2. Fix standard deviations*

$$\mathfrak{s}_1 = \gamma_1\eta\sqrt{\alpha^2 + Zd}, \qquad \mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d},$$
$$\mathfrak{s}_3 = \gamma_3\sqrt{337}\sqrt{(n_{\mathsf{bin}} + Z)d + \sum_{i=1}^{Z} \mathcal{B}_i^2}, \qquad \mathfrak{s}_4 = \gamma_4\sqrt{337}\mathcal{B}'$$

for some $\gamma_1, \gamma_2, \gamma_3, \gamma_4 > 0$ *and define*

$$M_i := \begin{cases} \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_i} + \frac{1}{2\gamma_i^2}\right) & \text{for } i = 1, 2 \\ \exp\left(\frac{1}{2\gamma_i^2}\right) & \text{for } i = 3, 4. \end{cases}$$

Suppose $\kappa_{\mathsf{MLWE}} := m_2 - \kappa_{\mathsf{MSIS}} - \ell - \lambda/2 - 512/d - 3 \geqslant 0$. *Then, under the* $\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}}, \kappa_{\mathsf{MSIS}} + \ell + \lambda/2 + 512/d + 3, \chi, \mathcal{C}, D^d_{\mathfrak{s}_2}}$ *assumption,* Π_{tbox} *for relation* R_{tbox} *is simulatable.*

Proof. Identically as in Theorems 6.1.3 and 6.1.9, the simulator $\mathcal{S}$ simulates $\mathbf{z}_3, \mathbf{z}_4$ by picking $(\mathbf{z}_3, \mathbf{z}_4) \leftarrow D^{256}_{\mathfrak{s}_3} \times D^{256}_{\mathfrak{s}_4}$ and then follows the simulator in Theorem 5.2.18. □

As we already mentioned, we now show that the commit-and-prove system Π_{tbox} for the relation $R^{\psi}_{\mathsf{toolbox}}$ (and not R_{toolbox}) is knowledge sound where ψ is a public approximation factor.

Theorem 6.4.3. *Suppose* $B_1 \geqslant 2\mathfrak{s}_1\sqrt{2(m_1+Z)d}$ *and* $B_2 \geqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$. *Let*

$$\mathfrak{s}_3 = \gamma_3\sqrt{337}\sqrt{(n_{\mathsf{bin}}+Z)d + \sum_{i=1}^{Z}\mathcal{B}_i^2}, \qquad \mathfrak{s}_4 = \gamma_4\sqrt{337}\mathcal{B}'$$

$$\psi := 2\gamma_4\sqrt{337\cdot 2\kappa}, \qquad \mathcal{B}_{\mathsf{arp}} := 2\sqrt{\frac{256}{26}}\varrho\mathfrak{s}_3$$

for $\gamma_3, \gamma_4 > 0$. *If* q *satisfies the following conditions*

$$q \geqslant 41\cdot\left(n_{\mathsf{bin}} + \sum_{i=1}^{Z} n_i + Z\right) d\cdot\mathcal{B}_{\mathsf{arp}}, \quad \text{to use Lemma 3.2.5}$$

$$q > \mathcal{B}_{\mathsf{arp}}^2 + \mathcal{B}_{\mathsf{arp}}\sqrt{n_{\mathsf{bin}} d}, \quad \text{to prove } \mathbf{P}_s + \mathbf{P}_m + \mathbf{f} \text{ has binary coeff.}$$

$$q > \mathcal{B}_{\mathsf{arp}}^2 + \mathcal{B}_{\mathsf{arp}}\sqrt{d}, \quad \text{to prove } \vartheta_1, \ldots, \vartheta_Z \text{ have binary coeff.}$$

$$\forall i \in [Z], q > 3\mathcal{B}_i^2 + \mathcal{B}_{\mathsf{arp}}^2, \quad \text{to prove (6.12)}$$

Then, the commit-and-prove system Π_{tbox} *for the relation* R_{tbox}^{ψ} *is knowledge sound with knowledge error*

$$2|\mathcal{C}|^{-1} + q_1^{-d/l} + q_1^{-\lambda} + 2^{-128} + 2^{-256}$$

under the $\mathsf{MSIS}_{\kappa_{\mathsf{MSIS}}, m_1+m_2, B}$ *assumption where* $B = 4\eta\sqrt{B_1^2 + B_2^2}$.

Proof. Let $\mathcal{P}^*$ be a probabilistic prover which runs in time at most T and convinces the verifier with probability $\epsilon > 2|\mathcal{C}|^{-1} + q_1^{-d/l} + q_1^{-\lambda} + 2^{-128} + 2^{-256}$. Then, similarly as in the proof of Theorems 6.1.4 and 6.1.10, we can define an extractor $\mathcal{E}$ which in expected runtime of at most $24T$ either solves the MSIS problem or extracts $(\bar{\mathbf{s}}_1^*, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2)$ and $\bar{c} \in \bar{\mathcal{C}}$ such that all the conditions below hold

1. $\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1^*, \bar{\mathbf{m}}; \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B) = 1$.
2. $\|\mathbf{D}_s\bar{\mathbf{s}}_1 + \mathbf{D}_m\bar{\mathbf{m}} + \mathbf{u}\|_\infty \leqslant 2\gamma_4\sqrt{337\cdot 2\kappa}\mathcal{B}' = \psi\cdot\mathcal{B}'$.
3. Let $\bar{\boldsymbol{\vartheta}} := \bar{\vartheta}_1 \parallel \cdots \parallel \bar{\vartheta}_Z$. Then,

$$\left\|\begin{bmatrix} \mathbf{P}_s\bar{\mathbf{s}}_1 + \mathbf{P}_m\bar{\mathbf{m}} + \mathbf{f} \\ \mathbf{E}_s^{(1)}\bar{\mathbf{s}}_1 + \mathbf{E}_m^{(1)}\bar{\mathbf{m}} + \mathbf{v}^{(1)} \\ \vdots \\ \mathbf{E}_s^{(Z)}\bar{\mathbf{s}}_1 + \mathbf{E}_m^{(Z)}\bar{\mathbf{m}} + \mathbf{v}^{(Z)} \\ \bar{\boldsymbol{\vartheta}} \end{bmatrix}\right\| \leqslant 2\sqrt{\frac{256}{26}}\varrho\mathfrak{s}_3 = \mathcal{B}_{\mathsf{arp}}$$

4. $\langle \mathbf{P}_s\bar{\mathbf{s}}_1 + \mathbf{P}_m\bar{\mathbf{m}} + \mathbf{f}, \mathbf{P}_s\bar{\mathbf{s}}_1 + \mathbf{P}_m\bar{\mathbf{m}} + \mathbf{f} - \mathbf{x}\rangle = 0$.
5. $\left\langle \vartheta_i, \vartheta_i - \left(\sum_{i=0}^{d-1} X^i\right)\right\rangle = 0$
6. $\langle \mathsf{pow}(\mathcal{B}_i^2), \vartheta_i\rangle + \langle \mathbf{E}_s^{(i)}\bar{\mathbf{s}}_1 + \mathbf{E}_m^{(i)}\bar{\mathbf{m}} + \mathbf{v}^{(i)}, \mathbf{E}_s^{(i)}\bar{\mathbf{s}}_1 + \mathbf{E}_m^{(i)}\bar{\mathbf{m}} + \mathbf{v}^{(i)}\rangle - \mathcal{B}_i^2 = 0$
7. For all $i \in [N]$, $\bar{\mathbf{s}}^T\mathbf{R}_{i,2}\bar{\mathbf{s}} + \mathbf{r}_{i,1}^T\bar{\mathbf{s}} + r_{i,0} = 0$ where $\bar{\mathbf{s}} := \bar{\mathbf{s}}_1 \parallel \bar{\mathbf{m}}$
8. For all $i \in [M]$, the constant coefficient of $\bar{\mathbf{s}}^T\mathbf{R}'_{i,2}\bar{\mathbf{s}} + \mathbf{r}'^T_{i,1}\bar{\mathbf{s}} + r'_{i,0}$ is zero

with probability at least $\epsilon - 2|\mathcal{C}|^{-1} - q_1^{-d/l} - q_1^{-\lambda} - 2^{-128} - 2^{-256}$ where

$$\bar{\mathbf{s}}_1^* := \bar{\mathbf{s}}_1 \parallel \bar{\vartheta}_1 \parallel \cdots \parallel \bar{\vartheta}_Z, \quad \mathbf{x} := \left(\sum_{i=0}^{d-1} X^i, \ldots, \sum_{i=0}^{d-1} X^i\right) \in \mathcal{R}_q^{n_{\mathsf{bin}}}$$

Now, by the assumptions on q and the fact that Statement 3 holds, we deduce that Statements $4, 5, 6$ hold over integers. Hence, we conclude that $\mathbf{P}_s\bar{\mathbf{s}}_1 + \mathbf{P}_m\bar{\mathbf{m}} + \mathbf{f}, \vartheta_1, \ldots, \vartheta_Z$ have binary coefficients as well as

$$\mathcal{B}_i - \|\mathbf{E}_s^{(i)}\bar{\mathbf{s}}_1 + \mathbf{E}_m^{(i)}\bar{\mathbf{m}} + \mathbf{v}^{(i)}\| \geqslant 0 \text{ for } i = 1, 2, \ldots, Z.$$

Thus, we conclude the proof. □

6.4.6 *Packing Signs*

Recall that we commit to each sign β_3 and β_4 separately. We can reduce the proof size by committing to both of them in the following way. Namely, we compute

$$\beta := \beta_3 + X^{d/2}\beta_4 \in \mathcal{R}_q$$

and commit to β:

$$t_\beta := \mathbf{b}_\beta^T\mathbf{s} + \beta.$$

In order to prove certain properties of β_3 and β_4, we observe that:

$$\beta_3 = 2^{-1} \cdot (\beta + \sigma(\beta)) \text{ and } \beta_4 = 2^{-1} \cdot (X^{d/2}\beta + \sigma(X^{d/2}\beta)).$$

Then, for example, to prove that β_3 is a sign, we show that

$$\beta_3^2 - 1 = \left(2^{-1} \cdot (\beta + \sigma(\beta))\right)^2 - 1 = 4^{-1} \cdot (\beta^2 + 2\sigma(\beta)\beta + \sigma(\beta)^2) - 1 = 0$$

and the constant coefficient of

$$X^i \cdot \beta_3 = X^i \cdot 2^{-1} \cdot (\beta + \sigma(\beta))$$

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

6.4.7 *Simplified Versions of the Framework Protocol*

In certain applications, we will not use the commit-and-prove system Π_{tbox} in its full capacity. For instance, it will not be necessary to do any infinity norm proofs or to prove shortness approximately. In the former case, this boils down to simply not adding relations (6.27) as an input to Π_{eval} and we remove one condition on q in Theorem 6.4.3. For the latter statement, we can omit the approximate range proof part, i.e. not commit to $\mathbf{y}_4, \beta_4$ and not send $\mathbf{z}_4$. Moreover, the relations described in (6.32),(6.33) and (6.34) as well as the improvement from Section 6.4.6 become irrelevant.

To conclude, it is easy to modify the protocol in Figure 6.3 in order to prove relations suitable for various applications.

6.5 NON-INTERACTIVE COMMIT-AND-PROVE FUNCTIONALITY

The broadly used Fiat-Shamir Transformation [DFM20; FS86] turns a public-coin interactive argument into a non-interactive argument in the random oracle model. The approach is to compute the i-th challenge c_i as a hash of the i-th prover message a_i as well as some part of the previous communication transcript (including the statement u itself). Then, if $\pi = (a_1, a_2, \ldots)$ is a proof then the verifier can manually recompute challenges c_i from π and a statement u.

We apply the multi-round Fiat-Shamir transformation for the protocol in Figure 6.3 to obtain a non-interactive commit-and-prove functionality for R_{toolbox}. Let $\mathsf{Lantern} = (\mathsf{ABDLOP}, \mathsf{Lantern.Prove}, \mathsf{Lantern.Verify})$ be the non-interactive commit-and-prove system where Lantern.Prove and Lantern.Verify are defined in Algorithms 1 and 4 respectively. Both algorithms use a subroutine ComputeRelations in Algorithm 3 respectively. Informally, ComputeRelations builds new relations analogously as in Figure 5.6.

We apply the standard optimisation where we do not send $\mathbf{w}$ and v but only the challenge c instead. Indeed, the verifier can compute $\mathbf{w}, v$ directly from the verification equations and then check whether

$$c \stackrel{?}{=} \mathcal{H}_4(u, pp, \mathbf{t}_A, \mathbf{t}_B, \mathbf{t}_y, \mathbf{t}_\beta, \mathbf{z}_3, \mathbf{z}_4, \mathbf{t}_g, \mathbf{h}, \mathbf{w}, t, v).$$

Security analysis of the non-interactive version of our commit-and-prove system can be derived similarly as in the interactive case. For example, since Π_{quad} defined in Figure 5.3 (which is the last black-box protocol called by Π_{tbox}) is a 3-special-sound Σ-protocol, its non-interactive version via

Algorithm 1 First Part of Lantern.Prove

Input: u as in (6.16), $(pp, \mathbf{t}_A, \mathbf{t}_B), (\mathbf{s}_1, \mathbf{m}, \boldsymbol{\vartheta} = (\vartheta_1, \ldots, \vartheta_Z)), \mathbf{s}_2$
Output: $\pi = \left(\mathbf{t}_y, \mathbf{t}_\beta, \mathbf{z}_3, \mathbf{z}_4, \mathbf{t}_g, \mathbf{h}, t, c, \mathbf{z}_1, \mathbf{z}_2\right)$

$(\mathbf{y}_3, \mathbf{y}_4, \beta_3, \beta_4) \leftarrow D_{\mathbf{s}_3}^{256} \times D_{\mathbf{s}_4}^{256} \times \{-1, 1\} \times \{-1, 1\}$
$\mathbf{t}_y = \mathbf{B}_y \mathbf{s}_2 + \begin{bmatrix} \mathbf{y}_3 \\ \mathbf{y}_4 \end{bmatrix}, \quad \mathbf{t}_\beta = \mathbf{B}_\beta \mathbf{s}_2 + \begin{bmatrix} \beta_3 \\ \beta_4 \end{bmatrix}$
$(R, R') = \mathcal{H}_1(u, pp, \mathbf{t}_A, \mathbf{t}_B, \mathbf{t}_y, \mathbf{t}_\beta)$
define $\mathbf{s}_3, \mathbf{s}_4$ as in (6.22),(6.30)
compute $\mathbf{z}_3, \mathbf{z}_4 \in \mathcal{R}_q^{256/d}$ s.t. $\begin{bmatrix} \vec{z}_3 \\ \vec{z}_4 \end{bmatrix} = \begin{bmatrix} \vec{y}_3 \\ \vec{y}_4 \end{bmatrix} + \begin{bmatrix} \beta_3 \cdot R\vec{s}_3 \\ \beta_4 \cdot R'\vec{s}_4 \end{bmatrix}$
for $i \in \{3, 4\}$ **do**
    if $\mathsf{Rej}^{(i)}(\vec{z}_i, R\vec{s}_i, \mathbf{s}_i, M_i) = 1$ **then**
        $(\mathbf{z}_3, \mathbf{z}_4) = (\perp, \perp)$
    end if
end for
$\mathbf{s}_1^* = \mathbf{s}_1 \parallel \boldsymbol{\vartheta} \in \mathcal{R}_q^{m_1+Z}$
$\mathbf{m}^* = \mathbf{m} \parallel \mathbf{y}_3 \parallel \mathbf{y}_4 \parallel \beta_3 \parallel \beta_4 \parallel \in \mathcal{R}_q^{\ell+512/d+2}$
$\mathbf{s}^* = \langle \mathbf{s}_1^* \parallel \mathbf{m}^* \rangle_\sigma$ ▷ Compute relations for the following $\mathbf{s}_1^*$ and $\mathbf{m}^*$
$(\mathbf{R}_{i,2}, \mathbf{r}_{i,1}, r_{i,0})_{i \in [N+2]}$ as in (6.20), (6.25), (6.33)
$(\mathbf{R}'_{i,2}, \mathbf{r}'_{i,1}, r'_{i,0})_{i \in [M+511+2d+2Z]}$ as in $\left\{ \begin{array}{c} (6.21), (6.24), (6.27), (6.28), \\ (6.29), (6.32), (6.34) \end{array} \right\}$
run Algorithm 2

Fiat-Shamir transform is also knowledge sound by [AFK21, Theorem 1]. Then, proving knowledge soundness for the next building blocks, such as $\Pi_{\mathsf{quad-many}}, \Pi_{\mathsf{quad-eval}}$ and eventually Π_{tbox} follows almost identically as in the interactive setting.

6.5.1 *Commitment and Proof Size*

We provide a general strategy on instantiating the non-interactive commit-and-prove functionality Lantern (or its interactive version in Figure 6.3). As before, we pick the challenge space $\mathcal{C}$ as described in Section 3.3.6 with respect to the automorphism σ_{-1}. Further, we choose λ and l such that terms $q_1^{-\lambda}$ and $q_1^{-d/l}$ are negligible.

Algorithm 2 Second Part of Lantern.Prove

$\mathbf{g} = (g_1, \ldots, g_{\lambda/2}) \leftarrow \{x : \mathcal{R}_q : \tilde{x} = 0\}$
$\mathbf{t}_g = \mathbf{B}_{\text{ext}}\mathbf{s}_2 + \mathbf{g}$
$\Upsilon = (v_{i,j}) = \mathcal{H}_2(u, pp, \mathbf{t}_A, \mathbf{t}_B, \mathbf{t}_y, \mathbf{t}_\beta, \mathbf{z}_3, \mathbf{z}_4, \mathbf{t}_g)$
for $i \in [\lambda/2]$ **do**
    $h_i = g_i + \sum_{j=1}^{M}(v_{2i-1,j} + X^{d/2}v_{2i,j})\mathsf{Tr}\left(\mathbf{s}^{*T}\mathbf{R}'_{j,2}\mathbf{s}^* + \mathbf{r}'^{T}_{j,1}\mathbf{s}^* + r'_{j,0}\right)$
end for
$\hat{\mathbf{s}}_1 = \mathbf{s}_1 \parallel \boldsymbol{\vartheta} \in \mathcal{R}_q^{m_1+Z}$
$\hat{\mathbf{m}} = \mathbf{m} \parallel \mathbf{y}_3 \parallel \mathbf{y}_4 \parallel \beta_3 \parallel \beta_4 \parallel \mathbf{g} \in \mathcal{R}_q^{\ell+512/d+2+\lambda/2}$
$\hat{\mathbf{s}} = \langle \hat{\mathbf{s}} \parallel \hat{\mathbf{m}} \rangle_\sigma$ ▷ Compute relations for the following $\hat{\mathbf{s}}_1$ and $\hat{\mathbf{m}}$
$\left(\mathbf{R}^\dagger_{i,2}, \mathbf{r}^\dagger_{i,1}, r^\dagger_{i,0}\right) \leftarrow \mathsf{ComputeRelations}\left((\mathbf{R}_{i,2}, \mathbf{r}_{i,1}, r_{i,0}), (\mathbf{R}'_{i,2}, \mathbf{r}'_{i,1}, r'_{i,0}), \Upsilon, \mathbf{h}\right)$
$(\mu_1, \ldots, \mu_N) = \mathcal{H}_3(u, pp, \mathbf{t}_A, \mathbf{t}_B, \mathbf{t}_y, \mathbf{t}_\beta, \mathbf{z}_3, \mathbf{z}_4, \mathbf{t}_g, \mathbf{h})$
$\left(\mathbf{R}^\dagger_2, \mathbf{r}^\dagger_1, r^\dagger_0\right) = \left(\sum_{i=1}^{N}\mu_i\mathbf{R}^\dagger_{i,2}, \sum_{i=1}^{N}\mu_i\mathbf{r}^\dagger_{i,1}, \sum_{i=1}^{N}\mu_i r^\dagger_{i,0}\right)$
$\mathbf{y}_1 \leftarrow D^{m_1 d}_{\mathfrak{s}_1}$
$\mathbf{y}_2 \leftarrow D^{m_2 d}_{\mathfrak{s}_2}$
$\mathbf{w} = \mathbf{A}_1\mathbf{y}_1 + \mathbf{A}_2\mathbf{y}_2$
$\mathbf{y} = \begin{bmatrix} \langle \mathbf{y}_1 \rangle_\sigma \\ -\langle \mathbf{B}\mathbf{y}_2 \rangle_\sigma \end{bmatrix}$
$g^* = \mathbf{s}^T\mathbf{R}^\dagger_2\mathbf{y} + \mathbf{y}^T\mathbf{R}^\dagger_2\mathbf{s} + \mathbf{r}^{\dagger T}_1\mathbf{y}$
$t = \mathbf{b}^T_{\text{ext}}\mathbf{s}_2 + g^*$
$v = \mathbf{y}^T\mathbf{R}^\dagger_2\mathbf{y} + \mathbf{b}^T_{\text{ext}}\mathbf{y}_2$
$c = \mathcal{H}_4(u, pp, \mathbf{t}_A, \mathbf{t}_B, \mathbf{t}_y, \mathbf{t}_\beta, \mathbf{z}_3, \mathbf{z}_4, \mathbf{t}_g, \mathbf{h}, \mathbf{w}, t, v)$
$\mathbf{z}_1 = c\mathbf{s}_1 + \mathbf{y}_1$
$\mathbf{z}_2 = c\mathbf{s}_2 + \mathbf{y}_2$
for $i \in \{1, 2\}$ **do**
    if $\mathsf{Rej}^{(i)}(\mathbf{z}_i, c\mathbf{s}_i, \mathfrak{s}_i, M_i) = 1$ **then**
        $(\mathbf{z}_1, \mathbf{z}_2) = (\perp, \perp)$
    end if
end for

Algorithm 3 ComputeRelations

Input: $(\mathbf{R}_{i,2}, \mathbf{r}_{i,1}, r_{i,0})_{i\in[N+2]}, (\mathbf{R}'_{i,2}, \mathbf{r}'_{i,1}, r'_{i,0})_{i\in[M+511+2d+2Z]}, \Upsilon = (v_{i,j})$

Output: $(\mathbf{R}^{\dagger}_{i,2}, \mathbf{r}^{\dagger}_{i,1}, r^{\dagger}_{i,0})_{i\in[N+2+\lambda/2]}$

$\hat{n} = m_1 + Z + \ell + 512/d + 2$ ▷ Length of $\hat{\mathbf{s}}_1 \parallel \hat{\mathbf{m}}$

$\hat{M} = M + 511 + 2d + 2Z$ ▷ Number of quadratic relations over $\mathbb{Z}_q$

compute $\mathbf{U} \in \mathcal{R}_q^{2\hat{n}\times 2\hat{n}}$ such that for all $\mathbf{x} \in \mathcal{R}_q^{\hat{n}}, \sigma(\langle \mathbf{x}\rangle_\sigma) = \mathbf{U}\langle \mathbf{x}\rangle_\sigma$

for $i \in [N+2]$ **do**

$$\mathbf{R}^{\dagger}_{i,2} = \begin{bmatrix} \mathbf{R}_{i,2} & \mathbf{0}_{2\hat{n}\times\lambda} \\ \mathbf{0}_{\lambda\times 2\hat{n}} & \mathbf{0}_{\lambda\times\lambda} \end{bmatrix}$$

$$\mathbf{r}^{\dagger}_{i,1} := \begin{bmatrix} \mathbf{r}_{i,1} \\ \mathbf{0}_{\lambda\times 1} \end{bmatrix}$$

$r^{\dagger}_{i,0} = r_{i,0}$

end for

for $i \in [\lambda/2]$ **do**

$$\mathbf{e}_i = \begin{bmatrix} \mathbf{0}_{2(i-1)\times 1} \\ 1 \\ \mathbf{0}_{(\lambda-2i+1)\times 1} \end{bmatrix}$$

$$\mathbf{R}^{\dagger}_{N+i,2} := \begin{bmatrix} \sum_{j=1}^{\hat{M}} \frac{(v_{2i-1,j} + X^{d/2} v_{2i,j})\left(\mathbf{R}'_{j,2} + \mathbf{U}^T \sigma(\mathbf{R}'_{j,2})\mathbf{U}\right)}{2} & \mathbf{0}_{2\hat{n}\times\lambda} \\ \mathbf{0}_{\lambda\times 2\hat{n}} & \mathbf{0}_{\lambda\times\lambda} \end{bmatrix}$$

$$\mathbf{r}^{\dagger}_{N+i,1} := \begin{bmatrix} \sum_{j=1}^{\hat{M}} \frac{(v_{2i-1,j} + X^{d/2} v_{2i,j})\left(\mathbf{r}'_{j,1} + \mathbf{U}^T \sigma(\mathbf{r}'_{j,1})\right)}{2} \\ \mathbf{e}_i \end{bmatrix}$$

$$r^{\dagger}_{N+i,0} := \sum_{j=1}^{\hat{M}} \frac{(v_{2i-1,j} + X^{d/2} v_{2i,j})\left(r'_{j,0} + \sigma(r'_{j,0})\right)}{2} - h_i$$

end for

Algorithm 4 Lantern.Verify

Input: $pp, \mathbf{t}_A, \mathbf{t}_B, u$ as in 6.16, $\pi = \left(\mathbf{t}_y, \mathbf{t}_\beta, \mathbf{z}_3, \mathbf{z}_4, \mathbf{t}_g, \mathbf{h}, t, c, \mathbf{z}_1, \mathbf{z}_2\right)$

Output: $b \in \{0,1\}$

if $\|\mathbf{z}_3\| > \rho\mathfrak{s}_3\sqrt{256} \vee \|\mathbf{z}_4\|_\infty > \sqrt{2\kappa}\mathfrak{s}_4 \vee \exists i \in [\lambda/2], \tilde{h}_i \neq 0$ **then**
  return 0
end if
$(R, R') = \mathcal{H}_1(u, pp, \mathbf{t}_A, \mathbf{t}_B, \mathbf{t}_y, \mathbf{t}_\beta)$
$(\mathbf{R}_{i,2}, \mathbf{r}_{i,1}, r_{i,0})_{i\in[N+2]}$ as in (6.20), (6.25), (6.33)
$(\mathbf{R}'_{i,2}, \mathbf{r}'_{i,1}, r'_{i,0})_{i\in[M+511+2d+2Z]}$ as in $\left\{\begin{array}{c}(6.21), (6.24), (6.27), (6.28),\\ (6.29), (6.32), (6.34)\end{array}\right\}$
$\Upsilon = (v_{i,j}) = \mathcal{H}_2(u, pp, \mathbf{t}_A, \mathbf{t}_B, \mathbf{t}_y, \mathbf{t}_\beta, \mathbf{z}_3, \mathbf{z}_4, \mathbf{t}_g)$
$\left(\mathbf{R}^\dagger_{i,2}, \mathbf{r}^\dagger_{i,1}, r^\dagger_{i,0}\right) \leftarrow \mathsf{ComputeRelations}\left((\mathbf{R}_{i,2}, \mathbf{r}_{i,1}, r_{i,0}), (\mathbf{R}'_{i,2}, \mathbf{r}'_{i,1}, r'_{i,0}), \Upsilon, \mathbf{h}\right)$
$(\mu_1, \ldots, \mu_N) = \mathcal{H}_3(u, pp, \mathbf{t}_A, \mathbf{t}_B, \mathbf{t}_y, \mathbf{t}_\beta, \mathbf{z}_3, \mathbf{z}_4, \mathbf{t}_g, \mathbf{h})$
$\left(\mathbf{R}^\dagger_2, \mathbf{r}^\dagger_1, r^\dagger_0\right) = \left(\sum_{i=1}^N \mu_i \mathbf{R}^\dagger_{i,2}, \sum_{i=1}^N \mu_i \mathbf{r}^\dagger_{i,1}, \sum_{i=1}^N \mu_i r^\dagger_{i,0}\right)$
$\mathbf{B}^\dagger = \begin{bmatrix}\mathbf{B}\\ \mathbf{B}_y\\ \mathbf{B}_\beta\\ \mathbf{B}_{\mathsf{ext}}\end{bmatrix}, \quad \mathbf{t}^\dagger = \begin{bmatrix}\mathbf{t}_B\\ \mathbf{t}_y\\ \mathbf{t}_\beta\\ \mathbf{t}_g\end{bmatrix}$
$\mathbf{z} = \begin{bmatrix}\langle \mathbf{z}_1 \rangle_\sigma\\ \langle c\mathbf{t}^\dagger - \mathbf{B}^\dagger \mathbf{z}_2 \rangle_\sigma\end{bmatrix}$
$f = ct - \mathbf{b}^T_{\mathsf{ext}}\mathbf{z}_2$
$\mathbf{w}^\dagger = \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}\mathbf{z}_2 - c\mathbf{t}_A, \quad v^\dagger = \mathbf{z}^T\mathbf{R}^\dagger\mathbf{z} + c\mathbf{r}_1^{\dagger T}\mathbf{z} + c^2 r_0^\dagger - f$
if $c \neq \mathcal{H}_4(u, pp, \mathbf{t}_A, \mathbf{t}_B, \mathbf{t}_y, \mathbf{t}_\beta, \mathbf{z}_3, \mathbf{z}_4, \mathbf{t}_g, \mathbf{h}, \mathbf{w}^\dagger, t, v^\dagger) \vee \|\mathbf{z}_1\| > \mathfrak{s}_1\sqrt{2m_1} \vee \|\mathbf{z}_2\| > \mathfrak{s}_2\sqrt{2m_2}$ **then**
  return 0
else
  return 1
end if

There are now 4 rejection sampling algorithms: each to mask $c\mathbf{s}_1, c\mathbf{s}_2, R\vec{s}_3$ and $R'\vec{s}_4$ respectively. Denote $\mathfrak{s}_i = \gamma_i T_i$ where T_1, T_2, T_3, T_4 are the upper-bounds on $\|c\mathbf{s}_1\|, \|c\mathbf{s}_2\|, \|R\vec{s}_3\|$ and $\|R'\vec{s}_4\|$ respectively. The non-aborting probability of the prover is

$$\approx \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_1} + \frac{1}{2\gamma_1^2} + \sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_2} + \frac{1}{2\gamma_2^2} + \frac{1}{2\gamma_3^2} + \frac{1}{2\gamma_4^2}\right)^{-1}.$$

Then, as in Theorem 6.4.1, we define

$$\mathfrak{s}_1 = \gamma_1\eta\sqrt{\alpha^2 + Zd}, \qquad \mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d},$$
$$\mathfrak{s}_3 = \gamma_3\sqrt{337}\sqrt{(n_{\mathsf{bin}} + Z)d + \sum_{i=1}^{Z}\mathcal{B}_i^2}, \qquad \mathfrak{s}_4 = \gamma_4\sqrt{337}\mathcal{B}'$$

Now we set κ_{MSIS} and m_2 such that the MLWE and MSIS from Theorems 6.4.2 and 6.4.3 are hard against known attacks. Here, we assume that MLWE is as hard as plain MLWE. We measure the hardness with the root Hermite factor δ and aim for $\delta \approx 1.0044$.

As discussed above, messages $\mathbf{w}$ and v need not be included in the output as they are uniquely determined by the remaining components. Moreover, all the challenges apart from c can be computed as a hash of the previous components of the proof. On the other hand, sending c requires at most $\lceil \log(2\kappa + 1) \rceil \cdot d$ bits.

As "full-sized" elements of $\mathcal{R}_q$, we have $\mathbf{t}_A, \mathbf{t}_B, \mathbf{t}_y, \mathbf{t}_\beta, \mathbf{t}_g, t$ and h_i. Therefore, we have in total $\kappa_{\mathsf{MSIS}} + \ell + 512/d + \lambda + 3$ full-sized elements of $\mathcal{R}_q$, which altogether costs at most $(\kappa_{\mathsf{MSIS}} + \ell + 512/d + \lambda + 3)\, d\lceil \log q \rceil$ bits. If we further apply the optimisation described in Section 6.4.6, the total cost is at most $(\kappa_{\mathsf{MSIS}} + \ell + 512/d + \lambda + 2)\, d\lceil \log q \rceil$ bits.

Now, the only remaining part are the vectors $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4$. We can encode them using the Huffman coding. Concretely, suppose that $z \leftarrow D_{\mathfrak{s}}$. Then, we can write

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

where $z_0 = z \bmod^{\pm} 2^{\delta+1}$. Since the expected absolute value of z is $\mathfrak{s}$ and assuming that $2^\delta \approx \mathfrak{s}$, the value of z_0 is close to being uniformly random between -2^δ and 2^δ. Due to the discrete Gaussian tails, the tails of the distribution of z_1 decrease very fast. Hence, the idea is to send z_0 in the clear (which has $\delta + 1$ bits) and then encode z_1 using the Huffman coding. If we assume that $\mathfrak{s} = 2^\delta$ and the tails of z_1 are the same as in the normal

distribution centred at zero [2], then the above compression requires on average approximately 1.57 bits to represent z_1. Thus, the total representation of z requires on average $\approx 2.57 + \delta$ bits. Applying this strategy to $\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4$, the overall commitment and proof length is around

$$(n + \ell + 512/d + \lambda + 2)\, d\lceil \log q \rceil + \lceil \log(2\kappa + 1) \rceil \cdot d + m_2 d \cdot (2.57 + \lceil \log \mathfrak{s}_2 \rceil)$$
$$+ (m_1 + Z)d \cdot (2.57 + \lceil \log \mathfrak{s}_1 \rceil) + 256 \cdot (2.57 + \lceil \log \mathfrak{s}_3 \rceil) + 256 \cdot (2.57 + \lceil \log \mathfrak{s}_4 \rceil)$$

bits.

Finally, we can further reduce the commitment and proof size by applying the compression techniques described in Section 4.3. The only change from the previous case is the introduction of the variables D (for cutting low-order bits of the commitment $\mathbf{t}_A$) and γ (for cutting low-order bits of $\mathbf{w}$ which allows us not to send some part of the masked opening $\mathbf{z}_2$ of the commitment randomness $\mathbf{s}_2$). Then, by Theorems 4.3.2 and 4.3.4, we choose $\kappa_{\mathsf{MSIS}}, m_2$ and D, γ so that the $\mathsf{MSIS}_{\kappa_{\mathsf{MSIS}}, m_1+m_2, B}$ is hard for $B := 4\eta \cdot \sqrt{B_1^2 + B_2^2}$ where

$$B_1 = 2\mathfrak{s}_1\sqrt{2m_1 d} \quad \text{and} \quad B_2 = 2\mathfrak{s}_2\sqrt{2m_2 d} + 2^D \eta \sqrt{\kappa_{\mathsf{MSIS}} d} + \gamma\sqrt{\kappa_{\mathsf{MSIS}} d}.$$

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

Now, we provide an asymptotic analysis of bounding the size of the hint vector $\mathbf{h}$. First, note that the coefficient vector $\mathbf{h}$ with high probability satisfies $\|\mathbf{h}\|_\infty \leqslant \|\mathsf{HighBits}_q(c\mathbf{t}_{A,2} - \mathbf{z}_{2,2})\|_\infty$ (here we assume the low-order bits $\mathbf{w}_0$ of $\mathbf{w}$ do not cause the increase in the high-order bits). Then, $\|c\mathbf{t}_{A,2} + \mathbf{z}_{2,2}\|_\infty \leqslant 2^{D-1}\omega d + 16\mathfrak{s}_2$ with an overwhelming probability by Lemma 3.2.2. Hence, we conclude that (with high probability) the coefficients of $\mathbf{h}$ are between $-x$ and x where

$$x := \left\lceil \frac{2^{D-1}\omega d + 16\mathfrak{s}_2}{\gamma} \right\rceil. \tag{6.35}$$

For our parameters, the standard deviation $\mathfrak{s}_2$ will be much smaller than γ and thus x will be close to $2^{D-1}\omega d/\gamma$. Finally, by picking D such that

2 This assumption is needed so that we can compute the frequencies for the Huffman coding.

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

TABLE 6.1: Prefix-free encoding [Duc+17].

$2^{D-1}\omega d < \gamma$, we conclude that the coefficients of $\mathbf{h}$ are between -1 and 1 with high probability. Assuming heuristically that they follow a binomial distribution, we encode $\mathbf{h}$ using a prefix-free encoding [3] [Duc+17] as shown in Table 6.1. As computed in [Duc+17], encoding a coefficient of $\mathbf{h}$ requires on average ≈ 2.25 bits.

The final proof size including compression becomes:

$$\begin{aligned}&\kappa_{\mathsf{MSIS}} d(\lceil \log q \rceil - D) + (\ell + 512/d + \lambda + 2)\, d\lceil \log q \rceil + \lceil \log(2\omega+1) \rceil \cdot d \\ &+ (m_1 + Z)d \cdot (2.57 + \lceil \log \mathfrak{s}_1 \rceil) + m_2 d \ \cdot (2.57 + \lceil \log \mathfrak{s}_2 \rceil) \\ &+ 2.25 \cdot \kappa_{\mathsf{MSIS}} d + 256 \cdot (2.57 + \lceil \log \mathfrak{s}_3 \rceil) + 256 \cdot (2.57 + \lceil \log \mathfrak{s}_4 \rceil) \text{ bits.}\end{aligned}$$

6.5.1.1 *Skipping the Non-Exact Norm Proof*

In certain applications, we will not perform any non-exact norm proofs, as described in Section 6.4.7. In this scenario, we do not send the commitments $\mathbf{y}_4, \beta_4$ and the masked opening $\mathbf{z}_4$. Also, the packing technique from Section 6.4.6 becomes pointless. In conclusion, the proof size for this case becomes:

$$\begin{aligned}&\kappa_{\mathsf{MSIS}} d(\lceil \log q \rceil - D) + (\ell + 256/d + \lambda + 2)\, d\lceil \log q \rceil + \lceil \log(2\omega+1) \rceil \cdot d \\ &+ (m_1 + Z)d \cdot (2.57 + \lceil \log \mathfrak{s}_1 \rceil) + m_2 d \ \cdot (2.57 + \lceil \log \mathfrak{s}_2 \rceil) \\ &+ 2.25 \cdot \kappa_{\mathsf{MSIS}} d + 256 \cdot (2.57 + \lceil \log \mathfrak{s}_3 \rceil) \text{ bits.}\end{aligned}$$

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

7

SHORTER PROOFS VIA ONE-TIME COMMITMENTS

In order to provide zero-knowledge (or more precisely, simulatability) for the protocols in Chapters 5 and 6, we apply rejection sampling to avoid leaking any information about the short message $\mathbf{s}_1$ and a randomness vector $\mathbf{s}_2$. As described in Sections 3.3.5 and 3.3.6, if one wants to use the Gaussian rejection sampling procedure [Lyu12], then the coefficients of $\mathbf{z}_i$ output in the proof are around $\gamma_i \cdot \eta \|\mathbf{s}_i\|$ – here η is the constant dependent on the challenge space and $\gamma_i > 0$ determines the repetition rate. Indeed, by the reasoning in Section 6.5.1 one would need to repeat at least [1]

$$M := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \left(\frac{1}{\gamma_1} + \frac{1}{\gamma_2}\right) + \frac{1}{2\gamma_1^2} + \frac{1}{2\gamma_2^2}\right)$$

times to obtain an accepting transcript. In terms of concrete parameters, if we main for 128-bit security then by setting $\gamma_1 = \gamma_2 = \sqrt{\frac{2(128+1)}{\log(e)}} \approx 13$, we obtain $M \approx 7.5$. Hence, the coefficients of $\mathbf{z}_i$ are about $13 \cdot 140$ [2] larger than coefficients of $\mathbf{s}_1$.

The increased coefficient size implies that the proof $(\mathbf{z}_1, \mathbf{z}_2)$ is noticeably larger than the message and randomness themselves. However, it seems necessary because leaking some information about the message or randomness can be dangerous. For instance, if one were to repeatedly perform proofs of knowledge for the same commitment and leak something about the same randomness $\mathbf{s}_2$ each time, eventually the entire $\mathbf{s}_2$ could be recovered.

Interestingly, the role of the commitments in many of the privacy-based primitives, such as group signatures [PLS18], is to commit to some intermediate messages $\mathbf{m}$ under fresh randomness $\mathbf{s}_2$ and give a proof-of-knowledge of $\mathbf{m}$ and that they satisfy certain relations. This means that the output of the primitive is a commitment *and* a proof. Consequently, every new output contains a commitment with fresh randomness $\mathbf{s}_2$. In this case, it is not immediately clear whether some leakage of the randomness vector is problematic. Nevertheless, it would be good to have a technique

1 For the sake of the overview, we ignore the terms related to $\mathbf{z}_3, \mathbf{z}_4$.
2 We use the value of η from Figure 3.3 for $d = 64$.

which lowers the proof size, and concurrently allows one to understand exactly how the hiding property of the commitment scheme is affected by the leakage. Similar analysis can also be applied for the ABDLOP commitments.

As discussed above, if one wants to avoid leaking any information about the randomness $\mathbf{s}_2$, then the "masked opening" $\mathbf{z}_2$ of $\mathbf{s}_2$ will have coefficients around $\gamma_2 \cdot \eta \|\mathbf{s}_2\|$. On the other end of the spectrum, if one simply sends $\mathbf{z}_2 = \mathbf{s}_2$ in the clear, then obviously the coefficients of $\mathbf{z}_2$ have size at most $\|\mathbf{s}_2\|_\infty$ but the whole randomness is leaked. Our contribution in this chapter is finding a middle ground and showing that by applying bimodal Gaussian rejection sampling on $\mathbf{z}_2$, i.e. use Rej_1 instead of Rej_0 defined in Figure 3.2, we reduce the coefficient size of $\mathbf{z}_2$ by a factor of $O(\gamma_2)$. We achieve this improvement at the cost of (potentially) leaking the inner product $\langle \mathbf{s}_2, c\mathbf{z}_2 \rangle \in \mathbb{Z}$ where $c \in \mathcal{C}$ is a challenge. Hence, we show that the simulatability property of our protocols relies on the *Extended*-MLWE assumption, first introduced by Alperin-Sheriff and Apon [AA16], which in addition to the plain MLWE, it reveals the inner products of the secret with public vectors to the adversary.

Similar results were proposed recently by Lyubashevsky et al. [LNS21a]. We describe the main differences. Firstly, for the same standard deviation, we obtain a repetition rate which is two times smaller. This is because in [LNS21a] the prover only sends $\mathbf{z}_2$ if the inner product $\langle \mathbf{z}_2, c\mathbf{s}_2 \rangle$ is non-negative which happens with probability at least $1/2$. This means that even an honest verifier learns the sign of the inner product. Although our protocol relies on a stronger variant of the Module-LWE assumption, where the adversary is given the whole inner product of the secret with a random vector rather than just the sign, the honest verifier in our case is not given explicit information about the inner product itself.

7.1 BIMODAL GAUSSIAN REJECTION SAMPLING ON THE RANDOMNESS

As evidenced in the case of signature schemes [Duc+13], applying bimodal Gaussians significantly reduces the standard deviation used for rejection sampling[3]. We attempt to follow the same methodology for our protocols.

In our constructions, we apply a rejection sampling procedure to mask a secret vector $\vec{v}$ by first sampling $\vec{y}$ from a discrete Gaussian with standard deviation $\mathfrak{s}$, and then computing $\vec{z} := \vec{v} + \vec{y}$. By Lemma 3.3.2, if we additionally run $\mathsf{Rej}_0(\vec{z}, \vec{v}, \mathfrak{s}, M)$, then the distribution of $\vec{z}$ is indistinguishable to the one where we simply sample $\vec{z}$ from a discrete Gaussian and output $\vec{z}$ with

3 One can compare Lemma 3.3.3 to Lemma 3.3.2 to see the difference.

$\mathcal{A}(\vec{v})$
01 $\vec{y} \leftarrow D^m_{\mathfrak{s}}$
02 $\vec{z} := \vec{y} + \vec{v}$
03 output $(\vec{z}, \vec{v})$ with prob. $\frac{\exp\left(\frac{\|\vec{v}\|^2}{2\mathfrak{s}^2}\right)}{M\cosh(\langle \vec{z}, \vec{v}\rangle/\mathfrak{s}^2)}$

$\mathcal{F}(\vec{v})$
01 $\vec{y} \leftarrow D^m_{\mathfrak{s}}$
02 $(\vec{z}_+, \vec{z}_-) := (\mathsf{sign}(\langle \vec{y}, \vec{v}\rangle) \cdot \vec{y}, -\mathsf{sign}(\langle \vec{y}, \vec{v}\rangle) \cdot \vec{y})$
03 $p := \frac{\exp\left(\frac{2|\langle \vec{y}, \vec{v}\rangle|}{\mathfrak{s}^2}\right)}{\exp\left(\frac{2|\langle \vec{y}, \vec{v}\rangle|}{\mathfrak{s}^2}\right)+1}$
04 $\vec{z} := \begin{cases} \vec{z}_+ \text{ with prob. } p \\ \vec{z}_- \text{ with prob. } 1-p \end{cases}$
05 output $(\vec{z}, \vec{v})$ with prob. $\frac{1}{M}$

FIGURE 7.1: Algorithms $\mathcal{A}$ and $\mathcal{F}$ for Lemma 7.1.1. We define $\mathsf{sign}(x) = 1$ if $x \geqslant 0$ and -1 otherwise.

probability $1/M$. Here, it is important that one could generate $\vec{z}$ without having any information on $\vec{v}$.

Now, suppose that instead of Rej_0, we run Rej_1. It is now a natural question to ask whether there is a way to simulate the $\vec{z}$ by having as little information on $\vec{v}$ as possible. We answer this question positively and show that this distribution is simulatable given only the inner product $\langle \vec{z}, \vec{v}\rangle$ of $\vec{z}$ and $\vec{v}$. We summarise our observation with the following lemma.

Lemma 7.1.1. *Let $\vec{v} \in \mathbb{Z}^m$ be a vector of norm T. Fix $\mathfrak{s} \geqslant \gamma T$ and $M \geqslant \exp\left(\frac{1}{2\gamma^2}\right)$. Then the distributions of the outputs of $\mathcal{A}(\vec{v})$ and $\mathcal{F}(\vec{v})$ defined in Figure 7.1 are identical. Moreover, the probability that $\mathcal{A}$ outputs something is exactly $1/M$.*

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

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

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

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

Now, we focus on $\mathcal{F}(\vec{v})$. We see that by construction, $\langle \vec{z}_+, \vec{v}\rangle \geqslant 0$ and $\langle \vec{z}_-, \vec{v}\rangle \leqslant 0$. Let us consider three separate cases. First, suppose $\vec{z}$ satisfies

$\langle \vec{z}, \vec{v} \rangle > 0$. Informally, we want to compute the probability that $\vec{y} = \pm\vec{z}$ and $\mathcal{F}$ picks $\vec{z}_+$. Then,

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

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

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

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

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

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

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

□

7.2 EXTENDED-MLWE

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

Definition 7.2.1 (Extended-MLWE)**.** The *Extended*-MLWE problem with parameters n, m and distribution χ, ξ_c, ξ_y over $\mathcal{R}$ asks the adversary $\mathcal{A}$ to distinguish between the following two cases: 1) $(\mathbf{B}, \mathbf{Bs}, c, \mathbf{y}, \langle c\mathbf{s}, \mathbf{y} \rangle)$ and 2) $(\mathbf{B}, \mathbf{u}, c, \mathbf{y}, \langle c\mathbf{s}, \mathbf{y} \rangle)$ for $\mathbf{B} \leftarrow \mathcal{R}_q^{m \times (n+m)}$, a secret vector $\mathbf{s} \leftarrow \chi^{n+m}$, uniformly

random vector $\mathbf{u} \in \mathcal{R}_q^m$ and $(c, \mathbf{y}) \leftarrow \xi_c \times \xi_z^{n+m}$. Then, $\mathcal{A}$ is said to have advantage ϵ in solving Extended-MLWE$_{n,m,\chi,\xi_c,\xi_y}$ if

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

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

We note that the (Module-)LWE problem with various side information has already been discussed in prior work e.g. [AP12; Dac+20; Dod+10]. As far as we are aware, this new variant of MLWE is the closest to the Extended Module-LWE problems defined by Lyubashevsky et al. [LNS21a], Alperin-Sheriff and Apon [AA16], Alperin-Sheriff and Peikert [AP12] and Boudgoust et al. [Bou+21].

We observe that [AA16] describes a similar problem with the two differences: (i) there is no c involved (assume that $c = 1$) and (ii) the hint is an arbitrary $\mathbb{Q}$-linear function on the "error" part $\mathbf{e}$ of the secret $\mathbf{s}$ (in particular, it could be $\langle \mathbf{e}, \mathbf{y}\rangle \in \mathbb{Z}$ where $\mathbf{y} \leftarrow \xi_y^m$). Alperin-Sheriff and Apon show that their Extended-MLWE problem can be reduced to plain MLWE if the errors come from a discrete Gaussian with a large enough standard deviation. The proof strategy was later extended by Boudgoust et al. [Bou+21] who define another Extended-MLWE problem. This time, however, the hint becomes a whole polynomial $\langle \mathbf{e}, \mathbf{y}\rangle \in \mathcal{R}$. Finally, the only difference between our problem and the one in [LNS21a] is that the adversary is given the whole inner product $\langle c\mathbf{s}, \mathbf{y}\rangle$ instead of its sign.

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

7.3 APPLICATIONS

For presentation, we apply our new rejection sampling strategy on the commit-and-prove system $\Pi_{\mathsf{lin}} = (\mathsf{ABDLOP}, \mathcal{P}, \mathcal{V})$ for the relation R_{lin} in Figure 5.1. However, it can be almost identically applied to all other systems described in Chapters 5 and 6.

Concretely, we substitute Theorems 5.1.1 and 5.1.2 with the following results.

Theorem 7.3.1. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}_0$ *and* $\mathsf{Rej}^{(2)} = \mathsf{Rej}_1$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1\eta\alpha$ *and* $\mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d}$ *for some* $\gamma_1, \gamma_2 > 0$ *and define*

$$M_1 := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_1} + \frac{1}{2\gamma_1^2}\right) \text{ and } M_2 := \exp\left(\frac{1}{2\gamma_2^2}\right).$$

Then, the commit-and-prove system Π_{lin} *for the relation* R_{lin} *has statistical completeness with correctness error* $1 - \frac{1}{M_1 M_2}$.

Proof. The proof follows directly from Lemma 7.1.1 which says that Rej_1 does not abort with probability $1/M_2$. □

Theorem 7.3.2. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}_0$ *and* $\mathsf{Rej}^{(2)} = \mathsf{Rej}_1$ *as defined in Figure 3.2. Fix standard deviations* $\mathfrak{s}_1 = \gamma_1\eta\alpha$ *and* $\mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d}$ *for some* $\gamma_1 > 0, \gamma_2 > 0$ *and define*

$$M_1 := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_1} + \frac{1}{2\gamma_1^2}\right) \text{ and } M_2 := \exp\left(\frac{1}{2\gamma_2^2}\right).$$

Suppose $\kappa_{\mathsf{MLWE}} := m_2 - \kappa_{\mathsf{MSIS}} - \ell \geqslant 0$. *Then, the commit-and-prove system* Π_{lin} *for the relation* R_{lin} *is simulatable under the Extended-*$\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}}, \kappa_{\mathsf{MSIS}}+\ell, \chi, \mathcal{C}, D^d_{\mathfrak{s}_2}}$ *assumption.*

Proof. Similarly as before, we prove the statement using a hybrid argument. First, we describe an efficient simulator $\mathcal{S}_1$, which knows $\mathbf{s}_1, \mathbf{m}$ and simulates both the commitment and the transcript as follows. It generates fresh randomness $\mathbf{s}_2 \leftarrow \chi^{m_2}$ and a masking vector $\mathbf{y}_2 \leftarrow D^{m_2 d}_{\mathfrak{s}_2}$ and computes $(\mathbf{t}_A, \mathbf{t}_B) = \mathsf{ABDLOP.Commit}(\mathbf{s}_1, \mathbf{m}; \mathbf{s}_2)$ and $\mathbf{z}_2 = \mathbf{y}_2 + c\mathbf{s}_2$. It aborts if $\mathsf{Rej}_1(\mathbf{z}_2, c\mathbf{s}_2, \mathfrak{s}_2, M_2) = 1$. Next, $\mathcal{S}_1$ samples $\mathbf{z}_1 \leftarrow D^{m_1 d}_{\mathfrak{s}_1}$. Finally, $\mathcal{S}_1$ computes

$$\mathbf{w} := \mathbf{A}_1\mathbf{z}_1 + \mathbf{A}_2\mathbf{z}_2 - c\mathbf{t}_A$$
$$\mathbf{v} := \mathbf{R}_1 \begin{bmatrix} \mathbf{z}_1 \\ c\mathbf{t}_B - \mathbf{B}\mathbf{z}_2 \end{bmatrix} + c\mathbf{r}_0$$

and outputs a simulated transcript $(\mathbf{w}, \mathbf{v}, c, \mathbf{z}_1, \mathbf{z}_2)$ with probability $1/M_1$. Then, by Lemma 3.3.2, the non-aborted simulated commitment and transcript by $\mathcal{S}_1$ are statistically close to the honestly generated commitment and non-aborted transcript.

Next, we describe an efficient simulator $\mathcal{S}_2$, which still knows $\mathbf{s}_1, \mathbf{m}$ and simulates both the commitment and the transcript in the following way. It executes the $\mathcal{S}_1$ algorithm but instead of constructing $\mathbf{z}_2$ honestly as in the protocol, $\mathcal{S}_2$ samples $\mathbf{y}_2 \leftarrow D_{\mathfrak{s}_2}^{m_2 d}$ and defines $\mathbf{z}_+ := \mathsf{sign}(\langle c\mathbf{s}_2, \mathbf{y}_2 \rangle) \cdot \mathbf{y}_2$ and $\mathbf{z}_- := -\mathbf{z}_+$. Then, it sets $\mathbf{z}_2 := \mathbf{z}_+$ with probability p and $\mathbf{z}_2 := \mathbf{z}_-$ with probability $1 - p$ where

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

It then continues with probability $1/M_2$. By Lemma 7.1.1, the non-aborted simulated commitment and transcript by $\mathcal{S}_1$ and $\mathcal{S}_2$ are identical.

Further, we describe an efficient simulator $\mathcal{S}_3$, which still knows $\mathbf{s}_1, \mathbf{m}$ and simulates both the commitment and the transcript as follows. Namely, it executes the $\mathcal{S}_2$ algorithm but instead of generating $(\mathbf{t}_A, \mathbf{t}_B)$ honestly, it samples $\mathbf{u} \leftarrow \mathcal{R}_q^{n+\ell}$ and computes:

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

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

Finally, we define our simulator $\mathcal{S}$, which has no access to private information anymore, as follows. Concretely, it executes the $\mathcal{S}_3$ algorithm but instead of generating $(\mathbf{t}_A, \mathbf{t}_B)$ as $\mathcal{S}_3$, it samples $\mathbf{u} \leftarrow \mathcal{R}_q^{n+\ell}$ and sets $(\mathbf{t}_A, \mathbf{t}_B) := \mathbf{u}$. Also, it does not perform any abort operations. Then, clearly the output distribution of $\mathcal{S}$ is identical to the non-aborted output of $\mathcal{S}_3$. Hence, the statement holds by the hybrid argument. □

In conclusion, for the same standard deviation $\mathfrak{s}_2$ we manage to reduce the repetition rate by a factor of $\exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_2}\right)$. For instance, by applying the new rejection sampling technique in the protocol described in Figure 6.3, an honest prover convinces the verifier with probability

$$\approx \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_1} + \frac{1}{2\gamma_1^2} + \frac{1}{2\gamma_2^2} + \frac{1}{2\gamma_3^2} + \frac{1}{2\gamma_4^2}\right)^{-1}.$$

8

APPLICATIONS

In this chapter, we show how to make use of our techniques developed in Chapter 6 for proving norms in real-world applications. Concretely, we apply our framework for proving knowledge of a Module-LWE secret in Section 8.1, verifiable encryption in Section 8.2 and proving integer relations in Section 8.3. Further, we focus on building more complex privacy-preserving primitives such as group and ring signatures in Section 8.4 and 8.5. In order to show significance of our results, we compare our efficiency with relevant prior work. We additionally provide SAGE [The22] scripts which compute parameters for the examples described in this chapter:

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

8.1 PROVING KNOWLEDGE OF A MODULE-LWE SECRET

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

$$\mathbf{A}\mathbf{s} + \mathbf{e} = \mathbf{u} \pmod{q} \tag{8.1}$$

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

We propose the following solution using the framework developed in Section 6.4. Simply, we commit to $\mathbf{s}$ and prove that

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

In Figure 8.1 we show to properly instantiate the commit-and-prove system Π_{tbox} to prove knowledge of a Module-LWE secret.

Remark. We note that [ENS20; LNS21a] could not avoid committing to $\mathbf{e}$ without having additional commitments. Indeed, previous work efficiently prove smallness of a vector $\mathbf{s}$, e.g. $\|\mathbf{s}\|_\infty \leqslant 1$, by committing to its coefficient vector $\vec{s}$ using NTT slots and then proving that

$$\vec{s} \circ (\vec{s} - \vec{1}) \circ (\vec{s} + \vec{1}) = \vec{0}.$$

variable	description	instantiation
N	# of quadratic equations over $\mathcal{R}_q$	0
M	# of quadratic equations over $\mathbb{Z}_q$	0
n_{bin}	length of the vector to prove binary coefficients	0
Z	# of exact norm proofs	1
–	approximate shortness proof	✗
$\mathbf{s}_1$	message in the Ajtai part	$\mathbf{s}$
$\mathbf{m}$	committed message in the BDLOP part	$\varnothing$
$\mathbf{E}_s^{(1)}$	matrix for proving $\|\mathbf{E}_s^{(1)}\mathbf{s}_1 + \mathbf{E}_m^{(1)}\mathbf{m} + \mathbf{v}^{(1)}\| \leqslant \mathcal{B}_1$	$\begin{bmatrix}\mathbf{I}_{\mathsf{m}} \\ \mathbf{A}\end{bmatrix}$
$\mathbf{E}_m^{(1)}$	matrix for proving $\|\mathbf{E}_s^{(1)}\mathbf{s}_1 + \mathbf{E}_m^{(1)}\mathbf{m} + \mathbf{v}^{(1)}\| \leqslant \mathcal{B}_1$	$\varnothing$
$\mathbf{v}^{(1)}$	vector for proving $\|\mathbf{E}_s^{(1)}\mathbf{s}_1 + \mathbf{E}_m^{(1)}\mathbf{m} + \mathbf{v}^{(1)}\| \leqslant \mathcal{B}_1$	$-\begin{bmatrix}\mathbf{0} \\ \mathbf{u}\end{bmatrix}$
$\mathcal{B}_1$	upper-bound on $\|\mathbf{E}_s^{(1)}\mathbf{s}_1 + \mathbf{E}_m^{(1)}\mathbf{m} + \mathbf{v}^{(1)}\|$	B

FIGURE 8.1: Instantiation of the protocol in Figure 6.3 for proving $\mathbf{As} + \mathbf{e} = \mathbf{u} \pmod q$ and $\|(\mathbf{s}, \mathbf{e})\| \leqslant B$. The variables in the first two columns refer to the ones defined in Section 6.4 and the ones in the last column refer to the parameters in this section. Here, $\varnothing$ denotes an empty vector/matrix.

If one were not to commit to $\mathbf{e}$, then one would need to prove an equation of the form

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

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

8.1.1 *Parameters*

We instantiate our protocol for the case when $q \approx 2^{32}$ and $\mathsf{n} = \mathsf{m} = 1024/d$ similarly [BLS19; ENS20; LNS21a] using the methodology in Section 6.5.1. We provide a summary of our parameter selection in Table 8.2.

parameters	description	value
q	prime modulus	$2^{32} - 99$
d	ring dimension of $\mathcal{R}$	64
l	# factors $X^d + 1$ splits into mod q	2
n	height of the $\mathbf{A}$ matrix	16
m	width of the $\mathbf{A}$ matrix	16
γ_1	rejection sampling constant for $c\mathbf{s}_1$	10
γ_2	rejection sampling constant for $c\mathbf{s}_2$	1
γ_3	rejection sampling constant for the ARP	6
ω	maximum coefficient of a challenge in $\mathcal{C}$	8
κ_{MSIS}	height of matrices $\mathbf{A}_1, \mathbf{A}_2$ in ABDLOP	19
m_1	length of the message $\mathbf{s}_1$ in the "Ajtai" part	16
ℓ	length of the message $\mathbf{m}$ in the "BDLOP" part	0
λ	$2 \cdot$ (# of $g_j \in \mathcal{R}_q$ for boosting soundness)	4
m_2	length of the randomness $\mathbf{s}_2$ in ABDLOP	47
ν	randomness $\mathbf{s}_2$ is sampled from $S_\nu^{m_2}$	1
γ	parameter to cut low-order bits of $\mathbf{w}$	65526
D	number of low-order bits cut from $\mathbf{t}_A$	3
	repetition rate	7
	commitment + proof size	13.1KB

FIGURE 8.2: Parameter selection for proving $\mathbf{As} + \mathbf{e} = \mathbf{u} \pmod q$ and $\|(\mathbf{s}, \mathbf{e})\| \leqslant \sqrt{2048}$ using the protocol in Figure 6.3

Let us pick prime $q := 2^{32} - 99$ (i.e. $q = q_1$) and set $d = 64, l = 2$ and $B = \sqrt{2048}$[1]. Then we define the randomness distribution χ as a uniform one over S_1. For the challenge space, we set $\omega = 8$ and $\eta = 140$ as in Figure 3.3. Then, any difference of two distinct challenges in $\mathcal{C}$ is invertible over $\mathcal{R}_q$. Also, for $q \approx 2^{32}$, we choose $\lambda = 4$. Thus, $q^{-d/l} < q^{-\lambda} \approx 2^{-128}$.

There are three rejection sampling algorithms: one to mask $c\mathbf{s}_1$, another one to mask $c\mathbf{s}_2$ and the last one to mask $\|R\vec{s}_3\|$. Denote $\mathfrak{s}_i = \gamma_i T_i$ where T_1, T_2, T_3 are the upper-bounds on $\|c\mathbf{s}_1\|, \|c\mathbf{s}_2\|$ and $\|R\vec{s}_3\|$ respectively. The repetition rate in our case, using the optimised rejection sampling in Chapter 7, is at least

$$\exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_1} + \frac{1}{2\gamma_1^2} + \frac{1}{2\gamma_2^2} + \frac{1}{2\gamma_3^2}\right).$$

The rate in [LNS21a] is around 7 hence we set $\gamma_1 = 10, \gamma_2 = 1$ and $\gamma_3 = 6$. Finally, the total communication size has been computed as in Section 6.5.1.1.

8.2 VERIFIABLE ENCRYPTION

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

$$\begin{bmatrix} \mathbf{t}_0 \\ t_1 \end{bmatrix} := \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ \lfloor \frac{p}{2} \rceil m \end{bmatrix} \tag{8.2}$$

over $\mathcal{R}_p := \mathbb{Z}_p[X]/(X^d + 1)$ where $(\mathbf{A}, \mathbf{b}) \in \mathcal{R}_p^{\mathsf{n} \times \mathsf{m}} \times \mathcal{R}_p^{\mathsf{n}}$ is the public key [2]. Let B be an upper-bound on $\mathbf{r}$ such that the probability that $\|\mathbf{r}\| > B$ for $\mathbf{r} \leftarrow \xi^{\mathsf{m}}$ is negligible. Then, in the verifiable encryption scenario, we want to prove knowledge of $\mathbf{r} \in \mathcal{R}^{\mathsf{m}}$ and $m \in \mathcal{R}$ such that (i) Equation 8.2 is satisfied over $\mathcal{R}_p$, (ii) $\|\mathbf{r}\| \leqslant B$ and (iii) $m \in \{0,1\}^d$.

The high-level idea is to commit to $(\mathbf{r}, m)$ using the ABDLOP commitment modulo q and prove these three statements. Note that the latter two have

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

2 Recall that all coefficients of the terms involved in (8.2) are between $-p/2$ and $p/2$.

already been covered in Section 6.4. Hence, from now on we focus on proving the first statement.

We first observe that if q is divisible by p then (8.2) can be transformed into a linear equation modulo q and can be proven as described in Section 6.4. However, in practical instantiations p will be significantly small relative to q (e.g. $p = 3329$ in Kyber). Consequently, if q has a small prime divisor p then by Theorem 6.4.3, we would need to commit to more garbage polynomials g_i in order to keep the soundness error negligible. Moreover, for implementation purposes one might want p to be a prime such that $X^d + 1$ splits into many factors modulo p (e.g. $p = 3329$). In this case, if p divides q, then the challenge space $\mathcal{C}$ does not have the invertibility property which is necessary for the soundness proof. In Figure 8.4 we propose an example instantiation for the case when q is divisible by p (see parameter set II).

Now, suppose that p is co-prime to q. Then, (8.2) is true if and only if there exists a vector $\mathbf{v} \in \mathcal{R}^{n+1}$ such that

$$\begin{bmatrix} \mathbf{t}_0 \\ t_1 \end{bmatrix} := \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^T \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{0} \\ \lfloor \frac{p}{2} \rceil m \end{bmatrix} + p\mathbf{v} \tag{8.3}$$

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

$$\mathbf{v} := p^{-1} \cdot \left(\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{b}^T & \lfloor \frac{p}{2} \rceil \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ m \end{bmatrix} - \begin{bmatrix} \mathbf{t}_0 \\ t_1 \end{bmatrix} \right) \in \mathcal{R}_q^n \tag{8.4}$$

has norm at most $B_v := (B\sqrt{md}/2 + 1)\sqrt{(n+1)d}$. Since this expression is linear in the committed messages $\mathbf{r}$ and m, we can apply the protocol in Figure 6.3 to prove its norm. As we will show below, it is enough to prove an approximate bound, i.e. $\|\mathbf{v}\|_\infty \leqslant \psi \cdot B_v$, where $\psi := 2 \cdot \sqrt{2\kappa \cdot 337}\gamma_4$, as described in Section 6.4.3. Indeed, in the soundness argument we would extract a pair $(\mathbf{r}^*, m^*)$ which satisfies

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

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

$$\begin{bmatrix} \mathbf{t}_0 \\ t_1 \end{bmatrix} \equiv \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^T \end{bmatrix} \mathbf{r}^* + \begin{bmatrix} \mathbf{0} \\ \lfloor \frac{p}{2} \rceil m^* \end{bmatrix} + p\mathbf{v}^* \pmod q. \tag{8.5}$$

Thus,

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

Hence, if q is bigger than the right-hand side of this inequality, then we conclude that Equation (8.5) holds over integers. In particular $(\mathbf{t}_0, t_1)$ is a valid encryption of m under randomness $\mathbf{r}$ over $\mathcal{R}_p$.

In Figure 8.3 we instantiate the protocol from Figure 6.3 for verifiable encryption as described above.

Remark. Note that the current state-of-the-art lattice based verifiable encryption [LN17], which is used in e.g. [Lyu+21; PLS18], only provide *relaxed* verifiable encryption. Namely, the soundness argument only guarantees knowledge of a message and randomness corresponding to the ciphertext $(\bar{c}\mathbf{t}_0, \bar{c}t_1)$, where $\bar{c} \in \mathcal{R}_p$ is called a relaxation factor. More importantly, $\bar{c}$ is not known to the decryptor and thus it guesses a $\bar{c}$ and attempts to recover the ciphertext $(\bar{c}\mathbf{t}_0, \bar{c}t_1)$. Consequently, the prior works had to equate the decryption time with the adversary's running time. Here, since we commit to $\mathbf{r}$ and m using a separate ABDLOP commitment, we circumvent the relaxation factor by proving exact norms on $\mathbf{r}$ and $m \in \{0,1\}^d$.

8.2.0.1 *Parameters*

We provide our parameters choices[3] in Figure 8.4. For the ciphertext modulus and dimensions, we follow the Kyber instantiation. In particular, we set $d = 64$, $n = 8, m = 18$ and $\mathbf{b} = \mathbf{A}^T\mathbf{s} + \mathbf{e}$ where the secret key $\mathbf{s}$ and error $\mathbf{e}$ come from Bin_2^{8d} and Bin_2^{18d} respectively. For the randomness distribution, fix $\xi := \mathsf{Bin}_2^d$. Hence, we can set the upper-bound B on the norm of $\mathbf{r} \leftarrow \xi^K$ as $B = 2\sqrt{md}$ and thus $B_v = (md+1)\sqrt{(n+1)d}$.

To compute the decryption error probability, we want to calculate the probability that for $\mathbf{r}, \mathbf{e} \leftarrow \mathsf{Bin}_2^{md}$, $\|\langle \mathbf{r}, \mathbf{e} \rangle\|_\infty < q/4$. First, we compute that for any $\vec{r}, \vec{e} \leftarrow \mathsf{Bin}_2^{md}$, the probability that $\|\langle \vec{r}, \vec{e} \rangle\|_\infty > 800$ is less than 2^{-360}.

3 One can also instantiate the encryption scheme over a larger ring, e.g. $\mathcal{R}' := \mathbb{Z}[X]/(X^{256}+1)$. Then, in order to apply our proof system over a smaller ring $\mathcal{R}$, one would first map the equations to work over $\mathcal{R}$ rather than $\mathcal{R}'$ as described in Section 3.2.4.

variable	description	instantiation
N	# of quadratic equations over $\mathcal{R}_q$	0
M	# of quadratic equations over $\mathbb{Z}_q$	0
n_{bin}	length of the vector to prove bin. coeff.	1
Z	# of exact norm proofs	1
–	approximate shortness proof	✓
$\mathbf{s}_1$	committed message in the Ajtai part	$\mathbf{r} \parallel m$
$\mathbf{m}$	committed message in the BDLOP part	$\varnothing$
$\mathbf{P}_s$	matrix to prove $\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}$ has bin. coeff.	$[\mathbf{0}\ 1]$
$\mathbf{P}_m$	matrix to prove $\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}$ has bin. coeff.	$\varnothing$
$\mathbf{f}$	vector to prove $\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}$ has bin. coeff.	$\mathbf{0}$
$\mathbf{E}_s^{(1)}$	matrix for proving $\|\mathbf{E}_s^{(1)}\mathbf{s}_1 + \mathbf{E}_m^{(1)}\mathbf{m} + \mathbf{v}^{(1)}\| \leqslant \mathcal{B}_1$	$[\mathbf{I}_{\mathsf{m}}\ \mathbf{0}]$
$\mathbf{E}_m^{(1)}$	matrix for proving $\|\mathbf{E}_s^{(1)}\mathbf{s}_1 + \mathbf{E}_m^{(1)}\mathbf{m} + \mathbf{v}^{(1)}\| \leqslant \mathcal{B}_1$	$\varnothing$
$\mathbf{v}^{(1)}$	vector for proving $\|\mathbf{E}_s^{(1)}\mathbf{s}_1 + \mathbf{E}_m^{(1)}\mathbf{m} + \mathbf{v}^{(1)}\| \leqslant \mathcal{B}_1$	$\mathbf{0}$
$\mathcal{B}_1$	upper-bound on $\|\mathbf{E}_s^{(1)}\mathbf{s}_1 + \mathbf{E}_m^{(1)}\mathbf{m} + \mathbf{v}^{(1)}\|$	B
$\mathbf{D}_s$	matrix for proving $\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\| \leqslant \mathcal{B}'$	$\frac{1}{p} \cdot \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{b}^T & \lfloor \frac{p}{2} \rceil \end{bmatrix}$
$\mathbf{D}_m$	matrix for proving $\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\| \leqslant \mathcal{B}'$	$\varnothing$
$\mathbf{u}$	vector for proving $\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\| \leqslant \mathcal{B}'$	$\frac{1}{p} \cdot \begin{bmatrix} \mathbf{t}_0 \\ t_1 \end{bmatrix}$
$\mathcal{B}'$	upper-bound on $\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\|$	B_v

FIGURE 8.3: Instantiation of the protocol in Figure 6.3 for verifiable encryption. The variables in the first two columns refer to the ones defined in Section 6.4 and the ones in the last column refer to the parameters in this section. Here, $B_v := (B\sqrt{\mathsf{m}d}/2 + 1)\sqrt{(\mathsf{n}+1)d}$ and $\varnothing$ denotes an empty vector/matrix.

parameters	description	I	II
p	encryption modulus	3329	3253
n	height of $\mathbf{A}$	8	8
m	width of $\mathbf{A}$	18	18
ξ	ξ^K is the rand. dist. of $\mathbf{r}$	Bin_2^d	Bin_2^d
q	proof system modulus	$\approx 2^{35}$	$\approx 2^{32}$
d	dimension of $\mathcal{R}$	64	64
l	# factors $X^d + 1$ splits into mod q	2	2
γ_1	rej. samp. constant for $c\mathbf{s}_1$	32	9.5
γ_2	rej. samp. constant for $c\mathbf{s}_2$	1	1
γ_3	rej. samp. constant for exact ARP	16	6
γ_4	rej. samp. constant for non-exact ARP	0.7	–
w	max. coeff. of a challenge in $\mathcal{C}$	8	8
κ_{MSIS}	height of $\mathbf{A}_1, \mathbf{A}_2$ in ABDLOP	20	19
m_1	length of the "Ajtai" message $\mathbf{s}_1$	19	11
ℓ	length of the "BDLOP" message $\mathbf{m}$	0	0
λ	$2 \cdot$ (# of $g_j \in \mathcal{R}_q$ for boosting soundness)	4	12
m_2	length of randomness $\mathbf{s}_2$	54	51
ν	randomness $\mathbf{s}_2$ is sampled from $S_\nu^{m_2}$	1	1
γ	parameter to cut low-order bits of $\mathbf{w}$	113302	28822
D	number of low-order bits cut from $\mathbf{t}_A$	8	6
	repetition rate	7	7
	ciphertext size	1KB	1KB
	commitment + proof size	17.2KB	15.0KB

FIGURE 8.4: Parameter selection, ciphertext and proof sizes for verifiable encryption. For the second parameter set we choose $q := 1320301 \cdot 3253$. Since p divides q, we do not need to do an approximate range proof of $\mathbf{v}$ as for I. Consequently, we can pick smaller modulus q and apply a similar strategy as in Section 8.1. In particular, we do not commit to the whole vector $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_0) \in \mathcal{R}_q^{\mathsf{m}-\mathsf{n}} \times \mathcal{R}_q^{\mathsf{n}}$, but only a part of it, i.e. the vector $\mathbf{r}_1$.

Then, by the union-bound, the probability that $\|\langle \mathbf{r}, \mathbf{e} \rangle\|_\infty > 800$ is still at most 2^{-300}. Hence, in our parameter selection, we will pick a prime p larger than 3200.

The rest of the parameters are chosen similarly Section 6.5.1. Finally, we need to check that

$$q \approx 2^{35} > p \cdot \left(B\sqrt{\mathsf{m}d}/2 + 1 + (B\sqrt{\mathsf{m}d}/2 + 1)\sqrt{(\mathsf{n}+1)d}\psi \right).$$

The term on the right-hand side is less than 2^{35} thus the inequality holds.

8.3 PROVING INTEGER RELATIONS

This section focuses on proving integer relations using the framework developed in Section 6.4. We start by proving integer addition in Section 8.3.1 and then move to proving multiplication in Section 8.3.2. We highlight that the relations we are interested in hold over integers, i.e. no wrap-around modulo q occurs.

8.3.1 *Integer Addition*

In this subsection we provide an efficient commit-and-prove system for addition on the committed integers. Specifically, given commitments to integers a, b, c (depending on the application, some of these values can be given out in the clear), we want to prove that $a + b = c$. In order to consider both positive and negative values, we use the two's complement representation. Namely, let n be a power of two and suppose $\mathsf{n} = kd$ for $k \geqslant 1$. Suppose $a, b, c \in [-2^{\mathsf{n}-1}, 2^{\mathsf{n}-1} - 1]$ and we want to prove $a + b = c$. Then, a can be represented in two's complement as n bits $a_0, \ldots, a_{\mathsf{n}-1} \in \{0, 1\}$ which satisfy

$$a = -a_{\mathsf{n}-1} 2^{\mathsf{n}-1} + \sum_{i=0}^{\mathsf{n}-2} a_i 2^i.$$

Similarly, we write

$$b = -b_{\mathsf{n}-1} 2^{\mathsf{n}-1} + \sum_{i=0}^{\mathsf{n}-2} b_i 2^i \quad \text{and} \quad c = -c_{\mathsf{n}-1} 2^{\mathsf{n}-1} + \sum_{i=0}^{\mathsf{n}-2} c_i 2^i.$$

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

$$\hat{a} := -a_{\mathsf{n}-1} X^{\mathsf{n}-1} + \sum_{i=0}^{\mathsf{n}-2} a_i X^i$$

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

$$\hat{a}(X) + \hat{b}(X) = \hat{c}(X) + (2 - X)\hat{f}(X) \tag{8.6}$$

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

Lemma 8.3.1. *The polynomial $\hat{f} \in \mathbb{Z}[X]$ defined above has coefficients in $\{0, 1\}$.*

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

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

and thus

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

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

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

and therefore

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

We conclude that $f_i \in \{0, 1\}$. Finally, focus on $f_{\mathsf{n}-2}$. We know that

$$-f_{\mathsf{n}-2} = (-a_{\mathsf{n}-1}) + (-b_{\mathsf{n}-1}) - (-c_{\mathsf{n}-1}) = c_{\mathsf{n}-1} - a_{\mathsf{n}-1} - b_{\mathsf{n}-1}.$$

Now, we claim that $0 \leqslant a_{\mathsf{n}-1} + b_{\mathsf{n}-1} - c_{\mathsf{n}-1} \leqslant 1$ which concludes the proof. Indeed, first note that $a_{\mathsf{n}-1} + b_{\mathsf{n}-1} - c_{\mathsf{n}-1} \leqslant 1$ since otherwise $a_{\mathsf{n}-1} = b_{\mathsf{n}-1} = 1$ and $c_{\mathsf{n}-1} = 0$. By definition of two's complement, this implies that $a, b < 0$ and $0 \leqslant c$. Thus, $a + b < 0 \leqslant c = a + b$ which is a contradiction. Next, we show that $a_{\mathsf{n}-1} + b_{\mathsf{n}-1} - c_{\mathsf{n}-1} \neq -1$. If it were the case, then $a_{\mathsf{n}-1} = b_{\mathsf{n}-1} = 0$ and $c_{\mathsf{n}-1} = 1$. However, then $a + b \geqslant 0 > c$ which leads to contradiction. Hence,

$$0 \leqslant f_{\mathsf{n}-2} = a_{\mathsf{n}-1} + b_{\mathsf{n}-1} - c_{\mathsf{n}-1} \leqslant 1.$$

□

Our strategy will be to prove (8.6). We do it by first proving the equation over $\mathcal{R}'_q := \mathbb{Z}_q[X]/(X^{\mathsf{n}} + 1) = \mathbb{Z}_q[X]/(X^{kd} + 1)$ and then showing that no modulo q and $X^{\mathsf{n}} + 1$ wrap-around occurs. Let $\hat{x} \in \mathcal{R}'_q$ be an inverse of $2 - X$. Such inverse exists if $2^{kd} + 1$ is not divisible by q which will be the

case in our instantiations. Consider the $\phi : \mathcal{R}'_q \to \mathcal{R}^k_q$ map described in Section 3.2.4, i.e.

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

We showed in Lemma 3.2.11 that (8.6) is equivalent to

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

For simplicity denote

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

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

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

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

1. *$\hat{a}, \hat{b}$ and $\hat{c}$ are well-formed.* We need to show that all the coefficients of $\hat{a} + X^{n-1}, \hat{b} + X^{n-1}, \hat{c} + X^{n-1}$ are binary. Note that this is equivalent to proving that $\hat{a}_0, \ldots, \hat{a}_{k-2}, \hat{a}_{k-1} + X^{d-1} \in \mathcal{R}_q$ all have binary coefficients and similarly for $\hat{b}, \hat{c}$.

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

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

   has binary coefficients.

3. *No overflow modulo $X^n + 1$.* Recall that we prove Equation 8.6 over $\mathcal{R}'_q$. In order to conclude that the equation holds over integers, we prove that there is no overflow modulo q and $X^n + 1$. The first statements above make sure no wrap-around modulo q occurs when $q \geqslant 7$. For the latter issue, note that it is enough to prove that the highest degree

coefficient of $\hat{f}$ is equal to zero. This is done by proving that the constant coefficient of

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

is equal to zero.

It is now easy to see that all the statements can be directly proven using our framework developed in Section 6.4. Namely, set $\mathbf{s}_1 := \phi(\hat{a}) \parallel \phi(\hat{b}) \parallel \phi(\hat{c})$ and $\mathbf{m} = \varnothing$. For presentation, denote $\mathbf{k}_{a,i}, \mathbf{k}_{b,i}, \mathbf{k}_{c,i} \in \mathcal{R}_q^{3k}$ as

$$\mathbf{k}_{a,i}^T = \begin{bmatrix} \mathbf{0}_{1\times i} & 1 & \mathbf{0}_{1\times(3k-i-1)} \end{bmatrix}, \quad \mathbf{k}_{b,i}^T = \begin{bmatrix} \mathbf{0}_{1\times(k+i)} & 1 & \mathbf{0}_{1\times(2k-i-1)} \end{bmatrix}$$
$$\mathbf{k}_{c,i}^T = \begin{bmatrix} \mathbf{0}_{1\times(2k+i)} & 1 & \mathbf{0}_{1\times(k-i-1)} \end{bmatrix}.$$

Then, for any $i \in \mathbb{Z}_k$ we have

$$\mathbf{k}_{a,i}^T \mathbf{s}_1 = \hat{a}_i, \quad \mathbf{k}_{b,i}^T \mathbf{s}_1 = \hat{b}_i, \quad \mathbf{k}_{c,i}^T \mathbf{s}_1 = \hat{c}_i.$$

Next, define the following matrices

$$\mathbf{K}_a := \begin{bmatrix} \mathbf{k}_{a,0}^T \\ \vdots \\ \mathbf{k}_{a,k-1}^T \end{bmatrix}, \quad \mathbf{K}_b := \begin{bmatrix} \mathbf{k}_{b,0}^T \\ \vdots \\ \mathbf{k}_{b,k-1}^T \end{bmatrix}, \quad \mathbf{K}_c := \begin{bmatrix} \mathbf{k}_{c,0}^T \\ \vdots \\ \mathbf{k}_{c,k-1}^T \end{bmatrix}$$

and

$$\mathbf{K}_f := \begin{bmatrix} \sum_{\substack{0 \leqslant i,j < k \\ i+j \equiv 0 \bmod k}} \hat{x}_i X^{\left\lfloor \frac{i+j}{k} \right\rfloor} \left(\mathbf{k}_{a,j}^T + \mathbf{k}_{b,j}^T - \mathbf{k}_{c,j}^T \right) \\ \vdots \\ \sum_{\substack{0 \leqslant i,j < k \\ i+j \equiv k-1 \bmod k}} \hat{x}_i X^{\left\lfloor \frac{i+j}{k} \right\rfloor} \left(\mathbf{k}_{a,j}^T + \mathbf{k}_{b,j}^T - \mathbf{k}_{c,j}^T \right) \end{bmatrix}.$$

Hence, to prove the first two statements, we want to prove that $\mathbf{P}_s\mathbf{s}_1 + \mathbf{f}$ has binary coefficients, where

$$\mathbf{P}_s := \begin{bmatrix} \mathbf{K}_a \\ \mathbf{K}_b \\ \mathbf{K}_c \\ \mathbf{K}_f \end{bmatrix} \in \mathcal{R}_q^{4k \times 3k}, \quad \mathbf{f} := \begin{bmatrix} \mathbf{0}_{(k-1)\times 1} \\ X^{d-1} \\ \mathbf{0}_{(k-1)\times 1} \\ X^{d-1} \\ \mathbf{0}_{(k-1)\times 1} \\ X^{d-1} \\ \mathbf{0}_{k\times 1} \end{bmatrix} \in \mathcal{R}_q^{4k}. \tag{8.7}$$

Finally, the third statement is equivalent to proving that the constant coefficient of $\langle \mathbf{s}_1 \rangle_\sigma^T \mathbf{R}'_{1,2} \langle \mathbf{s}_1 \rangle_\sigma + \mathbf{r}'^T_{1,1} \langle \mathbf{s}_1 \rangle_\sigma + r'_{1,0}$ is equal to zero where $\mathbf{R}'_{1,2} := \mathbf{0}_{6k\times 6k}$, $r'_{1,0} = 0$ and

$$\mathbf{r}'_{1,1} := X^{-d+1} \cdot \sum_{\substack{0 \leqslant i,j < k \\ i+j \equiv k-1 \bmod k}} \hat{x}_i \cdot \mathbf{J}_{3k,2}^T \left(\mathbf{k}_{a,j} + \mathbf{k}_{b,j} - \mathbf{k}_{c,j} \right) X^{\left\lfloor \frac{i+j}{k} \right\rfloor} \tag{8.8}$$

where the matrix $\mathbf{J}_{3k,2}$ defined in Lemma 5.2.1 satisfies $\mathbf{s}_1 = \mathbf{J}_{3k,2} \langle \mathbf{s}_1 \rangle_\sigma$.

In Figure 8.5 we instantiate the protocol from Figure 6.3 for integer addition as described above. Then, we present the proof sizes for various values of n in Figure 8.6. For each instance, we choose $(q, d, l) = (\approx 2^{32}, 64, 2)$ and set the standard deviations so that the overall repetition rate is at most 7. Other parameters are selected similarly as in the previous examples.

8.3.2 *Integer Multiplication*

We show how to prove knowledge of integers a, b, c such that $ab = c$. We first present a non-optimal solution which can be done by directly applying the framework in Section 6.4. Then, we describe a way to reduce the proof size at the cost of slightly extending our framework in Section 8.3.2.1.

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

$$a = -a_{\mathsf{n}-1} 2^{\mathsf{n}-1} + \sum_{i=0}^{\mathsf{n}-2} a_i 2^i, \qquad b = -b_{\mathsf{n}-1} 2^{\mathsf{n}-1} + \sum_{i=0}^{\mathsf{n}-2} b_i 2^i$$

variable	description	instantiation
N	# of quadratic equations over $\mathcal{R}_q$	0
M	# of quadratic equations over $\mathbb{Z}_q$	1
n_{bin}	length of the vector to prove bin. coeff.	$4k$
Z	# of exact norm proofs	0
–	approximate shortness proof	✗
$\mathbf{s}_1$	committed message in the Ajtai part	$\phi(\hat{a}) \parallel \phi(\hat{b}) \parallel \phi(\hat{c})$
$\mathbf{m}$	committed message in the BDLOP part	$\varnothing$
$\mathbf{R}'_{1,2}$	matrix used for the quad. equation over $\mathbb{Z}_q$	$\mathbf{0}_{6k\times 6k}$
$\mathbf{r}'_{1,1}$	vector used for the quad. equation over $\mathbb{Z}_q$	(8.8)
$r'_{1,0}$	constant used for the quad. equation over $\mathbb{Z}_q$	0
$\mathbf{P}_s$	matrix to prove $\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}$ has bin. coeff.	(8.7)
$\mathbf{P}_m$	matrix to prove $\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}$ has bin. coeff.	$\varnothing$
$\mathbf{f}$	vector to prove $\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}$ has bin. coeff.	(8.7)

FIGURE 8.5: Instantiation of the protocol in Figure 6.3 for proving n-bit integer addition where $\mathsf{n} = kd$. The variables in the first two columns refer to the ones defined in Section 6.4 and the ones in the last column refer to the parameters in this section. Here, $\varnothing$ denotes an empty vector/matrix.

and

$$c = -c_{2\mathsf{n}-1}2^{2\mathsf{n}-1} + \sum_{i=0}^{2\mathsf{n}-2} c_i 2^i.$$

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

$$\hat{a}(X) = a_0 + a_1 X + \cdots + a_{\mathsf{n}-2}X^{\mathsf{n}-2} - a_{\mathsf{n}-1}X^{\mathsf{n}-1} \in \mathbb{Z}[X]$$

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

$$\hat{a}(X)\hat{b}(X) - \hat{c}(X) = (2 - X)\hat{f}(X). \tag{8.9}$$

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

Lemma 8.3.2. *Let $\hat{f}$ be the polynomial of degree at most $2\mathsf{n}-2$ defined above. Then, for each coefficient f_k of $\hat{f}$ corresponding to X^k, $|f_k| \leqslant \mathsf{n}+1$.*

n	proof size
64	10.8KB
128	11.6KB
512	14.4KB

FIGURE 8.6: Proof size comparison for proving integer addition $a + b = c$ for $a, b, c \in [-2^{n-1}, 2^{n-1} - 1]$.

Proof. We first show $f_0 \in \{-1, 0, 1\}$. Consider Equation 8.9 for $X = 0$. Then, we have $a_0 b_0 - c_0 = 2f_0$. Since $-2 \leqslant a_0 b_0 - c_0 \leqslant 2$, we get $|f_0| \leqslant 1$.

In general, by considering the k-th coefficient of $\hat{a}\hat{b} - \hat{c}$ and $(2 - X)\hat{f}$ for $k > 0$, we have the following equality:

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

Hence, by the triangle inequality:

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

Thus, $|f_1| \leqslant n/2 + 1$. Then, one can show by induction that

$$|f_k| \leqslant (n+1)(1/2 + 1/4 + 1/8 + \ldots + 1/2^k) + 1/2^k < (n+1) + 1/2^k$$

for $k \geqslant 1$. Since $f_k \in \mathbb{Z}$, we have $|f_k| \leqslant n + 1$. □

Unlike in Lemma 8.3.1, the coefficients of $\hat{f}$ are much bigger than $\{0, 1\}$ but still small compared to q (if q is much larger than $n + 1$ which will be the case). However, in order to show that no modulo q overflow occurs, we just need to prove shortness of $\hat{f}$ approximately.

Similarly as in the integer addition proof, we want to prove Equation 8.9 over $\mathbb{Z}[X]$. In order to do so, we consider this equation over $\mathcal{R}'_q := \mathbb{Z}_q[X]/(X^{2n} + 1) = \mathbb{Z}_q[X]/(X^{kd} + 1)$. Namely, consider the $\phi : \mathcal{R}'_q \to \mathcal{R}^k_q$ map described in Section 3.2.4, i.e.

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

As shown in Lemma 3.2.11, (8.9) over $\mathcal{R}'_q$ is equivalent to

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

For simplicity denote

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

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

$$\forall \iota \in \mathbb{Z}_k, \sum_{\substack{0 \leqslant i,j < k \\ i+j \equiv \iota \bmod k}} \hat{a}_i \hat{b}_j X^{\left\lfloor \frac{i+j}{k} \right\rfloor} - \hat{c}_\iota = \sum_{\substack{0 \leqslant i,j < k \\ i+j \equiv \iota \bmod k}} \hat{x}_i \hat{f}_j X^{\left\lfloor \frac{i+j}{k} \right\rfloor}. \tag{8.10}$$

Now, in order to conclude that (8.9) holds over $\mathbb{Z}[X]$, we need to show that no wrap-around modulo q and $X^{2n} + 1$ occurs. For the first issue, we show that coefficients of $\hat{a} + X^{n-1}, \hat{b} + X^{n-1}$ and $\hat{c} + X^{2n-1}$ are binary (by definition of two's complement). As for $\hat{f}$, we conduct an approximate shortness proof to show that $\hat{f}$ has sufficiently small coefficients so that no modulo q overflow happens. Next, in order to make sure there is no wrap-around modulo $X^{2n} + 1$, we prove that the degree of $\hat{a}$ and $\hat{b}$ are at most $n - 1$ and the degree of $\hat{f}$ is at most $2n - 2$.

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

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

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

   are zeroes for $i \in \mathbb{Z}_{d/2}$ and $j \in \mathbb{Z}_k$.

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

3. *Equation 8.9 holds over $\mathcal{R}'_q$.* We simply prove k quadratic equations (8.10).

4. *No overflow modulo q.* We prove approximately that $\hat{f}$ has small coefficients. By Lemma 8.3.2, $\|\phi(\hat{f})\| \leqslant \mathcal{B}' := (n+1)\sqrt{2n} = (kd/2+1)\sqrt{kd}$. We can convince the verifier that $\|\hat{f}\|_\infty = \|\phi(\hat{f})\|_\infty \leqslant \psi \cdot \mathcal{B}'$ for some approximation factor. If

$$q > 2n + 1 + 3\psi \cdot \mathcal{B}'$$

and we proved that that $\hat{a}, \hat{b}, \hat{c}$ all have ternary coefficients, then (8.9) holds over $\mathbb{Z}$ and no wrap-around modulo q occurs.

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

It is now clear that all the statements can be directly proven using our framework developed in Section 6.4. Namely, define $\mathbf{s}_1 := \phi(\hat{a}) \parallel \phi(\hat{b}) \parallel \phi(\hat{c})$ and $\mathbf{m} = \phi(\hat{f})$. The reason to set $\mathbf{m}$ this way is because the coefficients of $\hat{f}$ are much larger than the coefficients of $\hat{a}, \hat{b}, \hat{c}$.

We introduce the following notation. First, recall that matrix $\mathbf{J}_{4k,2}$ defined in Lemma 5.2.1 satisfies:

$$\mathbf{J}_{4k,2}\langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \end{bmatrix}.$$

Next, denote $\mathbf{k}_{a,i}, \mathbf{k}_{b,i}, \mathbf{k}_{c,i}, \mathbf{k}_{f,i} \in \mathcal{R}_q^{4k}$ as

$$\mathbf{k}_{a,i}^T = \begin{bmatrix} \mathbf{0}_{1\times i} & 1 & \mathbf{0}_{1\times(4k-i-1)} \end{bmatrix}, \quad \mathbf{k}_{b,i}^T = \begin{bmatrix} \mathbf{0}_{1\times(k+i)} & 1 & \mathbf{0}_{1\times(3k-i-1)} \end{bmatrix}$$
$$\mathbf{k}_{c,i}^T = \begin{bmatrix} \mathbf{0}_{1\times(2k+i)} & 1 & \mathbf{0}_{1\times(2k-i-1)} \end{bmatrix}, \quad \mathbf{k}_{f,i}^T = \begin{bmatrix} \mathbf{0}_{1\times(3k+i)} & 1 & \mathbf{0}_{1\times(k-i-1)} \end{bmatrix}.$$

Then, for any $i \in \mathbb{Z}_k$ we have

$$\mathbf{k}_{a,i}^T \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \end{bmatrix} = \hat{a}_i, \quad \mathbf{k}_{b,i}^T \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \end{bmatrix} = \hat{b}_i, \quad \mathbf{k}_{c,i}^T \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \end{bmatrix} = \hat{c}_i, \quad \mathbf{k}_{f,i}^T \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \end{bmatrix} = \hat{f}_i.$$

Next, define the following matrices

$$\mathbf{K}_a := \begin{bmatrix} \mathbf{k}_{a,0}^T \\ \vdots \\ \mathbf{k}_{a,k-1}^T \end{bmatrix}, \quad \mathbf{K}_b := \begin{bmatrix} \mathbf{k}_{b,0}^T \\ \vdots \\ \mathbf{k}_{b,k-1}^T \end{bmatrix}, \quad \mathbf{K}_c := \begin{bmatrix} \mathbf{k}_{c,0}^T \\ \vdots \\ \mathbf{k}_{c,k-1}^T \end{bmatrix}, \quad \mathbf{K}_f := \begin{bmatrix} \mathbf{k}_{f,0}^T \\ \vdots \\ \mathbf{k}_{f,k-1}^T \end{bmatrix}.$$

Therefore, to prove the first two statements, we want to prove that $\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}$ has binary coefficients, where

$$\begin{bmatrix} \mathbf{P}_s & \mathbf{P}_m \end{bmatrix} := \begin{bmatrix} \mathbf{K}_a \\ \mathbf{K}_b \\ \mathbf{K}_c \end{bmatrix} \in \mathcal{R}_q^{3k \times 4k}, \quad \mathbf{f} := \begin{bmatrix} \mathbf{0}_{(k/2-1)\times 1} \\ X^{d-1} \\ \mathbf{0}_{(k-1)\times 1} \\ X^{d-1} \\ \mathbf{0}_{(3k/2-1)\times 1} \\ X^{d-1} \end{bmatrix} \in \mathcal{R}_q^{3k}. \tag{8.11}$$

Not to mention the fact that we need to show the constant coefficients of $X^{-i-d/2} \cdot \hat{a}_j$ and $X^{-i-d/2} \cdot \hat{b}_j$ are zeroes for $i \in \mathbb{Z}_{d/2}$ and $j \in \mathbb{Z}_k$. Hence, we define triples $(\mathbf{R}_{\iota,2}, \mathbf{r}_{\iota,1}, r_{\iota,0})$ for $\iota \in \mathbb{Z}_{kd/2}$ as follows. Let us write $\iota = i \cdot k + j$ where $i \in \mathbb{Z}_{d/2}$ and $j \in \mathbb{Z}_k$. Then, we define

$$\mathbf{R}_{\iota,2} := \mathbf{0}_{8k\times 8k}, \quad \mathbf{r}_{\iota,1}^T := X^{-i-d/2} \cdot \mathbf{k}_{a,j}^T \mathbf{J}_{4k,2}, \quad r_{\iota,0} := 0. \tag{8.12}$$

Similarly, we denote triples $(\mathbf{R}_{kd/2+\iota,2}, \mathbf{r}_{kd/2+\iota,1}, r_{kd/2+\iota,0})$ for $\iota \in \mathbb{Z}_{kd/2}$ as

$$\mathbf{R}_{kd/2+\iota,2} := \mathbf{0}_{8k\times 8k}, \quad \mathbf{r}_{kd/2+\iota,1}^T := X^{-i-d/2} \cdot \mathbf{k}_{b,j}^T \mathbf{J}_{4k,2}, \quad r_{kd/2+\iota,0} := 0. \tag{8.13}$$

Further, to prove the third statement, we define triples $(\mathbf{R}_{\iota,2}, \mathbf{r}_{\iota,1}, r_{\iota,0})$ for $\iota \in \mathbb{Z}_k$ as follows:

$$\begin{aligned} \mathbf{R}_{\iota,2} &:= \sum_{\substack{0 \leqslant i,j < k \\ i+j \equiv \iota \bmod k}} X^{\left\lfloor \frac{i+j}{k} \right\rfloor} \mathbf{J}_{4k,2}^T \mathbf{k}_{a,i} \mathbf{k}_{b,j}^T \mathbf{J}_{4k,2} \\ \mathbf{r}_{\iota,1}^T &:= -\left(\mathbf{k}_{c,\iota}^T + \sum_{\substack{0 \leqslant i,j < k \\ i+j \equiv \iota \bmod k}} X^{\left\lfloor \frac{i+j}{k} \right\rfloor} \hat{x}_i \mathbf{k}_{f,j}^T \right) \mathbf{J}_{4k,2} \\ r_{\iota,0} &= 0. \end{aligned} \tag{8.14}$$

Next, in order to prove the norm of $\phi(\hat{f})$ approximately, we define $\mathbf{D}_s = \mathbf{0}_{k\times 3k}$, $\mathbf{D}_m = \mathbf{I}_k$ and $\mathbf{u} = \mathbf{0}$. Then,

$$\|\phi(\hat{f})\| = \|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\| \leqslant \mathcal{B}' = (kd/2+1)\sqrt{kd}.$$

Finally, we focus on the last statement. Namely, we want to prove that the constant coefficient $X^{-d+1}\cdot\hat{f}_{k-1}$ vanishes. To this end, we define a tuple $(\mathbf{R}'_{kd,2}, \mathbf{r}'_{kd,1}, r'_{kd,0})$ where

$$\mathbf{R}'_{kd,2} := \mathbf{0}_{8k\times 8k}, \quad \mathbf{r}'_{kd,1} := X^{-d+1}\mathbf{J}^T_{4k,2}\mathbf{k}_{f,k-1}, \quad r'_{kd,0} = 0. \tag{8.15}$$

In Figure 8.7 we instantiate the protocol from Figure 6.3 for integer multiplication as described above. Then, we present the proof sizes for various values of n in Figure 8.8. For each instance, we choose $(d,l) = (64,2)$ and set the standard deviations such that the overall repetition rate is at most 7. We show that for our parameter selection $q > 2^{30}$ is large enough to make sure no overflow modulo q occurs. Suppose for concreteness that $\mathsf{n} = 512$. Then, $k = 16$ and $\mathcal{B}' = (kd/2+1)\sqrt{kd} = 16416$. If we pick $\gamma_4 = 32$ then by Theorem 6.4.3 we set $\psi := 2\gamma_4\sqrt{337\cdot 2\kappa} \leqslant 2\cdot 10^4$. Hence, we obtain

$$2\mathsf{n} + 1 + 3\psi\mathcal{B}' < 2^{30} < q.$$

Hence, there is no wrap-around modulo q.

8.3.2.1 *Is Committing to the Carry Polynomials Necessary?*

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

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

Now, note that our framework natively *only* supports proving shortness in the L_∞/L_2 norm of *linear* functions in $\mathbf{s}_1, \mathbf{m}$. The reason is that when applying approximate range proofs, we introduce a sign β in order to use bimodal Gaussian rejection sampling. Having this additional unknown β turns a linear equation into a quadratic one which we know how to prove

variable	description	instantiation
N	# of quadratic equations over $\mathcal{R}_q$	k
M	# of quadratic equations over $\mathbb{Z}_q$	$kd+1$
n_{bin}	length of the vector to prove bin. coeff.	$3k$
Z	# of exact norm proofs	0
–	approximate shortness proof	✓
$\mathbf{s}_1$	committed message in the Ajtai part	$\phi(\hat{a}) \parallel \phi(\hat{b}) \parallel \phi(\hat{c})$
$\mathbf{m}$	committed message in the BDLOP part	$\phi(\hat{f})$
$\mathbf{R}_{\iota,2}$	matrix used for the quad. equation over $\mathcal{R}_q$	(8.14)
$\mathbf{r}_{\iota,1}$	vector used for the quad. equation over $\mathcal{R}_q$	(8.14)
$r_{\iota,0}$	constant used for the quad. equation over $\mathcal{R}_q$	(8.14)
$\mathbf{R}'_{\iota,2}$	matrix used for the quad. equation over $\mathbb{Z}_q$	(8.12),(8.13),(8.15)
$\mathbf{r}'_{\iota,1}$	vector used for the quad. equation over $\mathbb{Z}_q$	(8.12),(8.13),(8.15)
$r'_{\iota,0}$	constant used for the quad. equation over $\mathbb{Z}_q$	(8.12),(8.13),(8.15)
$\mathbf{P}_s$	matrix to prove $\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}$ has bin. coeff.	(8.11)
$\mathbf{P}_m$	matrix to prove $\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}$ has bin. coeff.	(8.11)
$\mathbf{f}$	vector to prove $\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}$ has bin. coeff.	(8.11)
$\mathbf{D}_s$	matrix to prove $\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\| \leqslant \mathcal{B}'$	$\mathbf{0}_{k\times 3k}$
$\mathbf{D}_m$	matrix to prove $\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\| \leqslant \mathcal{B}'$	$\mathbf{I}_k$
$\mathbf{u}$	vector to prove $\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\| \leqslant \mathcal{B}'$	$\mathbf{0}_{k\times 1}$
$\mathcal{B}'$	bound of $\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\|$	$(kd/2+1)\sqrt{kd}$

FIGURE 8.7: Instantiation of the protocol in Figure 6.3 for proving n-bit integer multiplication where $2\mathsf{n} = kd$. The variables in the first two columns refer to the ones defined in Section 6.4 and the ones in the last column refer to the parameters in this section.

n	$\lceil \log q \rceil$	proof size
64	32	14.2KB
128	32	16.1KB
512	33	26.6KB

n	$\lceil \log q \rceil$	proof size
64	32	13.5KB
128	32	14.6KB
512	33	21.0KB

FIGURE 8.8: Proof size comparison for proving integer multiplication $ab = c$ for $a, b \in [-2^{n-1}, 2^{n-1} - 1]$ and $c \in [-2^{2n-1}, 2^{2n-1} - 1]$. We present two approaches: one which commits to $\phi(\hat{f})$ (on the left) and the one explained in Section 8.3.2.1 (on the right).

from Section 5.2. Observe that for integer addition $F_1, \ldots, F_k$ were indeed linear. However, for integer multiplication $F_1, \ldots, F_k$ become quadratic and thus our framework cannot be used directly.

We circumvent this problem and still not commit to $\phi(\hat{f})$ by simply removing the bimodal Gaussian rejection sampling when proving approximate shortness of $\phi(\hat{f})$. Then, we do not commit to β_4 from Figure 6.3 and thus we can prove shortness of a quadratic expression in $\mathbf{s}_1, \mathbf{m}$. The drawback is a slightly increased repetition rate due to the standard rejection sampling, i.e. Lemma 3.3.2. Concretely, when $\gamma_4 = 32$, our rejection rate increases by a factor of

$$\exp\left(\sqrt{\frac{2(\kappa+1)}{\log e}} \cdot \frac{1}{\gamma_4}\right) \approx 1.52 \quad \text{where } \kappa = 128$$

and therefore other standard deviations need to be adjusted in order to maintain the repetition rate equal to 7. We include the optimised proof sizes in Figure 8.8.

8.4 CONSTANT SIZE GROUP SIGNATURE

We apply our proof system to contruct an ABB-like [ABB10a] group signature following the works by del Pino et al. [PLS18] and Lyubashevsky et al. [Lyu+21]. Our construction inherits a big advantage from [Lyu+21; PLS18], namely signature generation and verification time do not depend on the size of the group and the signature itself is constant. Since, the techniques are almost identical as in the aforementioned previous works, we only sketch the scheme and refer to [Lyu+21] for more details. In this subsection, we work over the larger ring $\mathcal{R}_{kd} := \mathbb{Z}[X]/(X^{kd} + 1)$ where

$k \geqslant 1$ is a power-of-two. Then, define $\mathcal{R}'_{kd,p} := \mathcal{R}_{kd}/(p)$ for an integer p. The benefit of having a larger ring than $\mathcal{R}$ is a small public key size of our group signature. Operations in the construction will be over $\mathcal{R}_{kd,p}$ where p is prime.

8.4.1 *Overview*

Let $G \subseteq \mathcal{R}_{kd,p}$ be the identity space. To begin with, the group manager samples $\mathbf{A} \leftarrow \mathcal{R}_{kd,p}^{n\times(n+m)}$, $\mathbf{B}' \leftarrow \mathcal{R}_{kd,p}^{n\times\tau n}$, randomness matrix $\mathbf{R} \leftarrow S_{kd,1}^{(n+m)\times\tau n}$, where

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

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

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

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

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

such that

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

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

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

$$\left[\mathbf{A}|\mathbf{B} + i\mathbf{G}|\mathbf{B}'\right] \begin{bmatrix} \mathbf{s}_1^{(i)} \\ \mathbf{s}_2^{(i)} \\ \mathbf{s}_3^{(i)} \end{bmatrix} = \mathbf{u}, \quad \left\| \begin{bmatrix} \mathbf{s}_1^{(i)} \\ \mathbf{s}_2^{(i)} \\ \mathbf{s}_3^{(i)} \end{bmatrix} \right\| \leqslant B := \mathfrak{s}\sqrt{2((2\tau+1)n+m)kd}, \quad i \in G. \tag{8.16}$$

For the bound B we used Lemma 3.2.2 for $t = \sqrt{2}$.

In order to be able to open the group signature scheme, we will add a verifiable encryption to the signature. Namely, we want the signer to encrypt their identity i, using a public key associated to a decryption key that the group manager possesses, and prove that this encryption is indeed

of their identity. We do this exactly as described in Section 8.2 with a prime $p_{\mathsf{enc}} := 3329$. Similarly, all the dimensions and bounds included in that section will be written with subscript enc.

8.4.2 *Efficient Proof of* (8.16)

To begin with, note that relations over $\mathcal{R}_{kd,p}$ such as the first one in Equation (8.16) can be written equivalently over our usual subring $\mathcal{R}_p$. Indeed, as shown in Section 3.2.4 and demonstrated in the previous examples, arbitrary relations over $\mathcal{R}_{kd,p}$ can be proven by showing that some corresponding relations over $\mathcal{R}_p$ hold true.

Secondly, we observe that if we choose a proof system modulus q to be divisible by p and commit to $(i, \mathbf{s}_1^{(i)}, \mathbf{s}_2^{(i)}, \mathbf{s}_3^{(i)})$ in the "Ajtai" part of the ABDLOP commitment then the first statement in (8.16) is simply a system of quadratic equations in the committed messages. Indeed, we pick $q = q_1 p$ where $q_1 < p$ and then prove an equivalent quadratic relation over $\mathcal{R}_q$, namely:

$$q_1 \left[\mathbf{A}|\mathbf{B} + i\mathbf{G}|\mathbf{B}'\right] \begin{bmatrix} \mathbf{s}_1^{(i)} \\ \mathbf{s}_2^{(i)} \\ \mathbf{s}_3^{(i)} \end{bmatrix} = q_1 \left[\mathbf{A}|\mathbf{B}|\mathbf{G}|\mathbf{B}'\right] \begin{bmatrix} \mathbf{s}_1^{(i)} \\ \mathbf{s}_2^{(i)} \\ i\mathbf{s}_2^{(i)} \\ \mathbf{s}_3^{(i)} \end{bmatrix} = q_1 \mathbf{u}. \tag{8.17}$$

Further, the second statement is about norms which is covered in Section 6.4. Next, we define the identity space G. It should be designed so that we can efficiently prove that $i \in G$ (third statement). Let $\mathcal{B}$ be the set of non-zero binary polynomials in $\mathcal{R}_p$. Then, we define the identity space[4] as

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

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

Note that the space G is constructed in such a way that when we map equations over $\mathcal{R}_{kd,p}$ to $\mathcal{R}_p^k$, then we only need to commit to one polynomial

4 Previous works [Lyu+21; PLS18] define the identity space G to be a set of integers $\mathbb{Z}_p$ since it was easier to prove set membership $i \in G$ with their proof system. Here, we make a small modification and set the identity space to be a subset of binary polynomials with fixed norm.

$i \in \mathcal{R}_p$ using our ABDLOP commitment instead of k polynomials, i.e. $i(X^k) \in \mathcal{R}_{kd,p}$. Similarly, we only need to send an encryption of i over $\mathcal{R}_p$ instead of $i(X^k)$. Hence, for such a set G, proving $i(X^k) \in G$ is equivalent to proving that i has binary coefficients and $\langle i, \sum_{j=0}^{d-1} X^j \rangle = \mathsf{w}$ which is covered in Section 6.4.

In summary, we show in Figures 8.9 and 8.10 how to instantiate the protocol in Figure 6.3 to construct a group signature.

8.4.2.1 *Parameters*

We present our parameter selection in Figure 8.11 for a group signature instantiation which achieves security level 111. We start by setting $p = 2^{38} - 107$ and $q = (2^{26} - 371) \cdot p \approx 2^{64}$. Then, we choose $d = 64, k = 8$ and $l = 2$, thus $\mathcal{R}_{kd,p} = \mathbb{Z}[X]/(X^{512} + 1)$. Next, let $\mathsf{n} = 2, \mathsf{m} = 3$ and $\tau = 5$, hence $g = \lfloor p^{1/5} \rfloor$. Further, we pick large enough standard deviation $\mathfrak{s}$ used for trapdoor sampling. We know from [MP12] that $\mathfrak{s} \geqslant 2(s_1(\mathbf{R}) + 1)\sqrt{g^2 + 1}$ where s_1 is the operator norm. Note that if $\mathbf{R}$ did not have a polynomial structure, i.e $R \leftarrow \{-1, 0, 1\}^{(\mathsf{n}+\mathsf{m})kd \times \tau \mathsf{n} kd}$, we could use upper-bounds for norms of random subgaussian matrices, e.g. [MP12, Lemma 2.9]. Namely, we would obtain the following bound

$$s_1(R) \leqslant \sqrt{(\mathsf{n} + \mathsf{m})kd} + \sqrt{\tau \mathsf{n} kd} + 6 \approx 128$$

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

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

and thus we set

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

Further, we describe how we choose n and m, i.e. the height and the width of the matrix $\mathbf{A}$. Concretely, in the traceability proof, the challenger sets

$$\mathbf{B} := \mathbf{A}\mathbf{R} - i^*\mathbf{G} \quad \text{and} \quad \mathbf{B}' = \mathbf{A}\mathbf{R}'$$

where $\mathbf{R}, \mathbf{R}' \leftarrow S_{kd,1}^{(\mathsf{n}+\mathsf{m}) \times \tau \mathsf{n}}$ and $i^* \leftarrow G$. Additionally, it samples

$$sk^{\mathsf{gm}} := (\mathbf{s}_1^{\mathsf{gm}}, \mathbf{s}_2^{\mathsf{gm}}, \mathbf{s}_3^{\mathsf{gm}}) \leftarrow D_{\mathfrak{s}}^{((2\tau+1)\mathsf{n}+\mathsf{m})kd}$$

and computes

$$\mathbf{u} := \left[\mathbf{A} | \mathbf{A}\mathbf{R} | \mathbf{A}\mathbf{R}'\right] sk^{\mathsf{gm}}.$$

variable	description	instantiation
N	# of quadratic equations over $\mathcal{R}_q$	n
M	# of quadratic equations over $\mathbb{Z}_q$	1
n_{bin}	length of the vector to prove bin. coeff.	1
Z	# of exact norm proofs	2
–	approximate shortness proof	✓
$\mathbf{s}_1$	committed message in the Ajtai part	$(\mathbf{s}_1^{(i)}, \mathbf{s}_2^{(i)}, \mathbf{s}_3^{(i)}, \mathbf{r}_{\mathsf{enc}}, i)$
$\mathbf{m}$	committed message in the BDLOP part	$\varnothing$
$\mathbf{R}_{\iota,2}$	matrix used for the quad. equation over $\mathcal{R}_q$	to prove (8.17)
$\mathbf{r}_{\iota,1}$	vector used for the quad. equation over $\mathcal{R}_q$	to prove (8.17)
$r_{\iota,0}$	const. used for the quad. equation over $\mathcal{R}_q$	to prove (8.17)
$\mathbf{R}'_{\iota,2}$	matrix used for the quad. equation over $\mathbb{Z}_q$	$\mathbf{0}$
$\mathbf{r}'_{\iota,1}$	vector used for the quad. equation over $\mathbb{Z}_q$	$\begin{bmatrix} \mathbf{0} \\ \sigma_{-1}\left(\sum_{i=0}^{d-1} X^i\right) \\ 0 \end{bmatrix}$
$r'_{\iota,0}$	constant used for the quad. equation over $\mathbb{Z}_q$	$-\mathsf{w}$
$\mathbf{P}_s$	matrix to prove $\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}$ has bin. coeff.	$\mathbf{0}$
$\mathbf{P}_m$	matrix to prove $\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}$ has bin. coeff.	$\begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix}$
$\mathbf{f}$	vector to prove $\mathbf{P}_s\mathbf{s}_1 + \mathbf{P}_m\mathbf{m} + \mathbf{f}$ has bin. coeff.	0

FIGURE 8.9: Instantiation of the protocol in Figure 6.3 for the group signature. The instantiation is further explained in Figure 8.10. Variables in the first two columns refer to the ones defined in Section 6.4 and the ones in the last column refer to the parameters in this subsection. Variables with subscript enc are defined for the verifiable encryption in Section 8.2. Triple $(\mathbf{R}'_{1,2}, \mathbf{r}'_{1,1}, r_{1,0})$ corresponds to proving that the sum of coefficients of polynomial i is equal to exactly w. On the other hand, triple $(\mathbf{P}_s, \mathbf{P}_m, \mathbf{f})$ corresponds to proving that polynomial i has binary coefficients.

variable	description	instantiation
$\mathbf{E}_s^{(1)}$	matrix for $\|\mathbf{E}_s^{(1)}\mathbf{s}_1 + \mathbf{E}_m^{(1)}\mathbf{m} + \mathbf{v}^{(1)}\| \leqslant \mathcal{B}_1$	$[\mathbf{I}_{(2\tau+1)\mathsf{n}+\mathsf{m}}\ \mathbf{0}]$
$\mathbf{E}_m^{(1)}$	matrix for $\|\mathbf{E}_s^{(1)}\mathbf{s}_1 + \mathbf{E}_m^{(1)}\mathbf{m} + \mathbf{v}^{(1)}\| \leqslant \mathcal{B}_1$	$\varnothing$
$\mathbf{v}^{(1)}$	vector for $\|\mathbf{E}_s^{(1)}\mathbf{s}_1 + \mathbf{E}_m^{(1)}\mathbf{m} + \mathbf{v}^{(1)}\| \leqslant \mathcal{B}_1$	$\mathbf{0}$
$\mathcal{B}_1$	upper-bound on $\|\mathbf{E}_s^{(1)}\mathbf{s}_1 + \mathbf{E}_m^{(1)}\mathbf{m} + \mathbf{v}^{(1)}\|$	B
$\mathbf{E}_s^{(2)}$	matrix for $\|\mathbf{E}_s^{(2)}\mathbf{s}_1 + \mathbf{E}_m^{(2)}\mathbf{m} + \mathbf{v}^{(2)}\| \leqslant \mathcal{B}_2$	$[\mathbf{0}\ \mathbf{I}_{\mathsf{m}_{\mathsf{enc}}}\ \mathbf{0}]$
$\mathbf{E}_m^{(2)}$	matrix for $\|\mathbf{E}_s^{(2)}\mathbf{s}_1 + \mathbf{E}_m^{(2)}\mathbf{m} + \mathbf{v}^{(2)}\| \leqslant \mathcal{B}_2$	$\varnothing$
$\mathbf{v}^{(2)}$	vector for $\|\mathbf{E}_s^{(2}\mathbf{s}_1 + \mathbf{E}_m^{(2)}\mathbf{m} + \mathbf{v}^{(2)}\| \leqslant \mathcal{B}_2$	$\mathbf{0}$
$\mathcal{B}_2$	upper-bound on $\|\mathbf{E}_s^{(2)}\mathbf{s}_1 + \mathbf{E}_m^{(2)}\mathbf{m} + \mathbf{v}^{(2)}\|$	B
$\mathbf{D}_s$	matrix to prove $\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\| \leqslant \mathcal{B}'$	$\frac{1}{p_{\mathsf{enc}}} \cdot \begin{bmatrix} \mathbf{0} & \mathbf{A}_{\mathsf{enc}} & 0 \\ \mathbf{0} & \mathbf{b}_{\mathsf{enc}}^T & \lfloor \frac{p_{\mathsf{enc}}}{2} \rceil \end{bmatrix}$
$\mathbf{D}_m$	matrix to prove $\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\| \leqslant \mathcal{B}'$	$\varnothing$
$\mathbf{u}$	vector to prove $\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\| \leqslant \mathcal{B}'$	$\frac{1}{p_{\mathsf{enc}}} \cdot \begin{bmatrix} \mathbf{t}_0 \\ t_1 \end{bmatrix}$
$\mathcal{B}'$	bound of $\|\mathbf{D}_s\mathbf{s}_1 + \mathbf{D}_m\mathbf{m} + \mathbf{u}\|$	$B_{v,\mathsf{enc}}$

FIGURE 8.10: Instantiation of the protocol in Figure 6.3 for the group signature. Triples $(\mathbf{E}_s^{(1)}, \mathbf{E}_m^{(1)}, \mathbf{v}^{(1)}, \mathcal{B}_1)$ and $(\mathbf{E}_s^{(2)}, \mathbf{E}_m^{(2)}, \mathbf{v}^{(2)}, \mathcal{B}_2)$ correspond to proving exactly $\|(\mathbf{s}_1^{(i)}, \mathbf{s}_2^{(i)}, \mathbf{s}_3^{(i)})\| \leqslant B$ and $\|\mathbf{r}_{\mathsf{enc}}\| \leqslant B_{\mathsf{enc}}$ respectively. The last triple $(\mathbf{D}_s, \mathbf{D}_m, \mathbf{u}, \mathcal{B}')$ corresponds to proving approximately that $\|\mathbf{v}_{\mathsf{enc}}\| \leqslant B_{v,\mathsf{enc}} := (B_{\mathsf{enc}}\sqrt{\mathsf{m}_{\mathsf{enc}}d}/2 + 1)\sqrt{(\mathsf{n}_{\mathsf{enc}} + 1)d}$ where $\mathbf{v}_{\mathsf{enc}}$ is defined in (8.4).

It will hope that an adversary forges a signature for the identity i^*[5]. In that case, we can extract from the forged signature the secret vector $sk_{i^*} = (\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{\mathbf{s}}_3)$ such that

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

and thus

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

5 Hence, there is a $1/|G|$ security loss.

is a MSIS solution for the matrix $\mathbf{A}$ [6]. Also, with high probability we have $\mathbf{s} \neq 0$ since sk^{gm} was chosen independently by the challenger. Now, we need to bound the norm of $\mathbf{s}$. In order to do so, we will use the property that for any $\mathbf{x} \in \mathcal{R}_p^{\tau\mathsf{n}}$, $\|\mathbf{R}\mathbf{x}\| \leqslant s_1(\mathbf{R})\|\mathbf{x}\| \leqslant \psi\|\mathbf{x}\|$. Thus, we can bound the norm of $\mathbf{s}$ defined above using the Cauchy-Schwarz inequality as follows:

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

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

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

Hence

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

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

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

Last but not least, we observe that including a verifiable encryption from Section 8.2 does not have a significant impact on the total signature size. Indeed, identity i is already committed using the ABDLOP scheme and additionally committing to the randomness $\mathbf{r}$ (in the "Ajtai part") does not increase the commitment size. Hence, the only extra cost consists of: (i) a ciphertext, (ii) masked opening of the randomness $\mathbf{r}$, (iii) commitments and masked openings to polynomials involved in the approximate range proof for $\mathbf{v}$ in (8.4). For our instantiation, the verifiable encryption costs $\approx$ 6.5KB compared to 17.3KB shown in Figure 8.4.

6 Since we prove the norm of sk_i* exactly, there is no relaxation factor c in front of the vector $\mathbf{u}$ as in previous works.

parameters	description	value
p	modulus for the group signature	$2^{38} - 107$
d	ring dimension for of $\mathcal{R}$	64
k	kd is the ring dimension of $\mathcal{R}'$	8
N	height of the $\mathbf{A}$ matrix	2
M	$\mathsf{n} + \mathsf{m}$ is the width of the $\mathbf{A}$ matrix	3
τ	$\tau\mathsf{n}$ is the width of the gadget matrix $\mathbf{G}$	5
w	#1's in the identity $i \in G$	5
p_{enc}	encryption modulus	3329
N_{enc}	height of $\mathbf{A}_{\mathsf{enc}}$	4
K_{enc}	width of $\mathbf{A}_{\mathsf{enc}}$	9
ξ_{enc}	ξ_{enc}^K is the randomness distribution of $\mathbf{r}_{\mathsf{enc}}$	Bin_2^d
q	modulus for the proof system	$\approx 2^{64}$
l	# factors $X^d + 1$ splits into mod q	2
γ_1	rejection sampling constant for $c\mathbf{s}_1$	9
γ_2	rejection sampling constant for $c\mathbf{s}_2$	1.2
γ_3	rejection sampling constant exact ARP	2.5
γ_4	rejection sampling constant for non-exact ARP	12
ω	maximum coefficient of a challenge in $\mathcal{C}$	8
κ_{MSIS}	height of matrices $\mathbf{A}_1, \mathbf{A}_2$ in ABDLOP	25
m_1	length of the message $\mathbf{s}_1$ in the "Ajtai" part	219
ℓ	length of the message $\mathbf{m}$ in the "BDLOP" part	0
λ	number of garbage $g_j \in \mathcal{R}_q$ for boosting soundness	6
m_2	length of the randomness $\mathbf{s}_2$ in ABDLOP	78
ν	randomness $\mathbf{s}_2$ is sampled from $S_\nu^{m_2}$	1
γ	parameter to cut low-order bits of $\mathbf{w}$	$\approx 2^{35}$
D	number of low-order bits cut from $\mathbf{t}_A$	27
	signature size	87.5KB
	public key size	47.5KB
	secret key size	6.3KB

FIGURE 8.11: Parameter selection and concrete sizes for the group signature.

8.5 ONE-OUT-OF-MANY PROOF

8.5.1 *Overview*

In this section we construct an efficient logarithmic-size one-out-of-many proof [GK15] with applications to lattice-based ring and group signatures using techniques from Section 6.4 as the building block. The one-out-of-many proof considers the following problem. Informally, we want to prove knowledge of an opening to some commitment contained in a public set S without revealing any information about the commitment itself. In the lattice setting, we we would like to prove knowledge of a short vector such that $\mathbf{As} \in S$, where S is a public set $S = \{\mathbf{p}_1, \ldots, \mathbf{p}_t\} \subseteq \mathcal{R}_q^n$ of size $t = d \cdot \mathfrak{d}^k$. In this section we assume that $\mathbf{s} \in \{0,1\}^{md}$ has binary coefficients and $d = \mathfrak{l} \cdot \mathfrak{d}$ for $\mathfrak{l} \in \mathbb{N}$.

We now use the observation from [Boo+15; Esg+19b; GK15] that $\mathbf{As} \in S$ if and only if there exists a binary vector $\vec{v} \in \{0,1\}^t$ with exactly one 1 such that

$$\begin{bmatrix} \vec{p}_1 & \vec{p}_2 & \cdots & \vec{p}_t \end{bmatrix} \vec{v} = A\vec{s} \tag{8.18}$$

where $A = \mathsf{Rot}(\mathbf{A})$ is the the rotation matrix of $\mathbf{A}$. One could then directly prove knowledge of $\vec{s}$ and $\vec{v}$ which satisfy conditions above using the protocol from Section 6.4. However, the proof size grows linearly in t since we would commit to the whole vector $\vec{v}$.

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

$$\vec{v} = \vec{v}_1 \otimes \vec{v}_2 \otimes \cdots \otimes \vec{v}_{k+1} := \vec{v}_1 \otimes (\vec{v}_2 \otimes (\cdots \otimes (\vec{v}_k \otimes \vec{v}_{k+1}))) \,. \tag{8.19}$$

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

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

In the end, we want to commit to $\mathbf{s}$ and polynomials $u_1, \ldots, u_k, v_{k+1} \in \mathcal{X}$ such that $\vec{u}_i = \vec{v}_i \parallel 0^{d-\mathfrak{d}} \in \mathbb{Z}_q^{d}$[7] for $i \in [k]$ and prove

$$P(\vec{v}_1 \otimes \cdots \otimes \vec{v}_{k+1}) = A\vec{s} \tag{8.20}$$

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

where $P \in \mathbb{Z}_q^{n \times t}$ is the matrix on the left-hand side of (8.18). We formally define the corresponding relation:

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

We now describe a commit-and-prove system for relation R_{oom} using the ABDLOP commitment. Suppose that $k \geqslant 1$, otherwise one can prove this relation directly using the framework from 6.4.

First, note that proving $u_1, \ldots, u_k, v_{k+1} \in \mathcal{X}$ and $\mathbf{s} \in \{0,1\}^{md}$ can be done directly using the techniques from Section 6.4 hence we focus first on (8.20). Our strategy to prove this equation with $k-1$ tensor products would be somehow to reduce it to proving an equation of the same form with only $k-2$ tensor products. Then, by recursion, we will end up with a system of linear equations with no tensor products involved and thus we can apply the methods presented in Section 6.4.

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

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

Note that if (8.20) was not true, then these $\mathfrak{l}$ equations above would hold with probability at most $q_1^{-\mathfrak{l}}$. Now, if we write

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

then by simple algebraic manipulation we obtain

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

where

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

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

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

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

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

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

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

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

Lemma 8.5.1. *Let $i \in [\mathfrak{l}]$. Then, the constant coefficient of $X^{(i-1)\mathfrak{d}} u_1 \sigma(w) \in \mathcal{R}_q$ is equal to $\langle \vec{v}_1, \vec{w}_i \rangle$.*

Proof. First, we note that $\langle \vec{v}_1, \vec{w}_i \rangle = \langle X^{(i-1)\mathfrak{d}} u_1, w \rangle$. Here, we used the fact that the coefficient vector of u_1 is of the form $\vec{v}_1 \parallel 0^{d-\mathfrak{d}}$. Then, by Lemma 5.1.10, $\langle X^{(i-1)\mathfrak{d}} u_1, w \rangle$ is the constant coefficient of $X^{(i-1)\mathfrak{d}} u_1 \sigma(w)$. □

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

$$\vec{w} = \begin{bmatrix} P_{1,1} \\ \vdots \\ P_{1,\mathfrak{l}} \end{bmatrix} (\vec{v}_2 \otimes \cdots \otimes \vec{v}_{k+1}). \tag{8.21}$$

Thus, we reduce the one-out-of-many problem to proving knowledge of a tuple $(\mathbf{s}, u_1, \ldots, u_k, v_{k+1}, w)$ which satisfies the following conditions:

- $\mathbf{s} \in \{0,1\}^{\mathfrak{m}d}$
- $P_1(\vec{v}_2 \otimes \cdots \otimes \vec{v}_{k+1}) = \vec{w}$
- for all $i \in [\mathfrak{l}]$, the constant coefficient of $X^{(i-1)\mathfrak{d}} u_1 \sigma(w) - \sigma(\mathbf{a}_i)^T \mathbf{s}$ is zero
- $u_1, \ldots, u_k, v_{k+1} \in \mathcal{X}$

where

$$\vec{u}_i := \vec{v}_i \parallel 0^{d-\mathfrak{d}} \text{ for } i \in [k] \quad \text{and} \quad P_1 := \begin{bmatrix} P_{1,1} \\ \vdots \\ P_{1,\mathfrak{l}} \end{bmatrix} \in \mathbb{Z}_q^{d \times d\mathfrak{d}^{k-1}}.$$

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

We can define the correspond relation as:

$$R := \left\{ \begin{array}{c} \Big((P_1, (\mathbf{a}_i)_{i\in[\mathfrak{l}]}), (\mathbf{s}, u_1, \ldots, u_k, v_{k+1}, w) \Big) : \mathbf{s} \in \{0,1\}^{\mathfrak{m}d} \wedge P_1(\vec{v}_2 \otimes \cdots \otimes \vec{v}_{k+1}) = \vec{w} \\ \wedge \forall i \in [\mathfrak{l}], \text{ const coeff. of } X^{(i-1)\mathfrak{d}} u_1 \sigma(w) - \sigma(\mathbf{a}_i)^T \mathbf{s} \text{ is zero} \\ \wedge u_1, \ldots, u_k, v_{k+1} \in \mathcal{X} \text{ where } \vec{u}_i := \vec{v}_i \parallel 0^{d-\mathfrak{d}} \end{array} \right\}.$$

8.5.1.1 *Intermediate Relations*

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

$$R_j := \left\{ \begin{array}{c} \Big((P_j \in \mathbb{Z}_q^{d \times d\mathfrak{d}^{k-j}}, (\mathbf{a}_i)_{i\in[\mathfrak{l}]}, (\varphi_{\iota,i})_{\iota\in[j-1], i\in[\mathfrak{l}]}), (\mathbf{s}, u_1, \ldots, u_k, v_{k+1}, w_1, \ldots, w_j) \Big) : \\ \mathbf{s} \in \{0,1\}^{\mathfrak{m}d} \wedge P_j(\vec{v}_{j+1} \otimes \cdots \otimes \vec{v}_{k+1}) = \vec{w}_j \\ \wedge \forall i \in [\mathfrak{l}], \text{ const coeff. of } X^{(i-1)\mathfrak{d}} u_1 \sigma(w_1) - \sigma(\mathbf{a}_i)^T \mathbf{s} \text{ is zero} \\ \wedge \forall \iota \in [j-1], i \in [\mathfrak{l}], \text{ const coeff. of } X^{(i-1)\mathfrak{d}} u_{\iota+1} \sigma(w_{\iota+1}) - \sigma(\varphi_{\iota,i}) w_\iota \text{ is zero} \\ \wedge u_1, \ldots, u_k, v_{k+1} \in \mathcal{X} \text{ where } \vec{u}_i := \vec{v}_i \parallel 0^{d-\mathfrak{d}} \end{array} \right\}. \tag{8.22}$$

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

8.5.1.2 *Base Case*

We first show how to prove R_k only using the methods described in Section 6.4. Namely, we define

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

We also introduce the matrix $\mathbf{J} := \mathbf{J}_{\mathfrak{m}+2k+1,2}$ as in Lemma 5.2.1 which satisfies:

$$\mathbf{J}\langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma = \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{m} \end{bmatrix}.$$

First, we prove $P_k \vec{v}_{k+1} = \vec{w}_k$. This is equivalent to proving that the constant coefficient of $\sigma(p_{k,i}) v_{k+1} - X^{-i+1} w_k$ is zero for all $i \in [d]$, where $p_{k,i} \in \mathcal{R}_q$ is the polynomial such that its coefficient vector is the i-th row of P_k. Hence, we define

$$\mathbf{R}'_{i,2} = \mathbf{0}_{2(\mathfrak{m}+2k+1)\times 2(\mathfrak{m}+2k+1)}, \quad \mathbf{r}'_{i,1} := \mathbf{J}^T \begin{bmatrix} \mathbf{0}_{(\mathfrak{m}+k)\times 1} \\ \sigma(p_{k,i}) \\ \mathbf{0}_{(k-1)\times 1} \\ -X^{-i+1} \end{bmatrix}, \quad r'_{i,0} = 0. \tag{8.23}$$

Then,

$$\mathbf{r}_{i,1}^T \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma = \sigma(p_{k,i}) u_k - X^{-i+1} w_k.$$

The next thing to prove is that the constant coefficient of $X^{(i-1)\mathfrak{d}} u_1 \sigma(w_1) - \sigma(\mathbf{a}_i)^T \mathbf{s}$ is zero for $i \in [\mathfrak{l}]$. Thus, we define

$$\begin{aligned} \mathbf{R}'_{d+i,2} &:= X^{(i-1)\mathfrak{d}} \begin{bmatrix} \mathbf{0}_{(2\mathfrak{m}+2k+3)\times 1} \\ 1 \\ \mathbf{0}_{2(k-1)\times 1} \end{bmatrix} \begin{bmatrix} \mathbf{0}_{2\mathfrak{m}\times 1} & 1 & \mathbf{0}_{(4k+1)\times 1} \end{bmatrix}, \\ \mathbf{r}'_{d+i,1} &:= \mathbf{J}^T \begin{bmatrix} -\sigma(\mathbf{a}_i) \\ \mathbf{0}_{(2k+1)\times 1} \end{bmatrix}, \qquad r'_{d+i,0} := 0. \end{aligned} \tag{8.24}$$

Then,

$$\begin{aligned} \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma^T \mathbf{R}'_{d+i,2} \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma + \mathbf{r}'^T_{d+i,1} \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma + r_{d+i,0} \\ = X^{(i-1)\mathfrak{d}} u_1 \sigma(w)_1 - \sigma(\mathbf{a}_i)^T \mathbf{s}. \end{aligned}$$

Further, we proceed to proving that for all $\iota \in [k-1]$ and $i \in [\mathfrak{l}]$, the constant coefficient of $X^{(i-1)\mathfrak{d}} u_{\iota+1} \sigma(w_{\iota+1}) - \sigma(\varphi_{\iota,i}) w_\iota$ is zero. Hence, we define

$$\begin{aligned} \mathbf{R}'_{d+\iota\mathfrak{l}+i,2} &:= X^{(i-1)\mathfrak{d}} \begin{bmatrix} \mathbf{0}_{(2\mathfrak{m}+2k+2\iota+3)\times 1} \\ 1 \\ \mathbf{0}_{(2k-2\iota-2)\times 1} \end{bmatrix} \begin{bmatrix} \mathbf{0}_{2(\mathfrak{m}+\iota)\times 1} & 1 & \mathbf{0}_{(4k-2\iota+1)\times 1} \end{bmatrix}, \\ \mathbf{r}'_{d+\iota\mathfrak{l}+i,1} &:= \mathbf{J}^T \begin{bmatrix} \mathbf{0}_{\mathfrak{m}+k+\iota} \\ -\sigma(\varphi_{\iota,i}) \\ \mathbf{0}_{(k-\iota)\times 1} \end{bmatrix}, \qquad r'_{d+\iota\mathfrak{l}+i,0} := 0. \end{aligned} \tag{8.25}$$

Thus,

$$\begin{aligned} \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma^T \mathbf{R}'_{d+\iota\mathfrak{l}+i,2} \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma + \mathbf{r}'^T_{d+\iota\mathfrak{l}+i,1} \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma + r_{d+\iota\mathfrak{l}+i,0} \\ = X^{(i-1)\mathfrak{d}} u_{\iota+1} \sigma(w_{\iota+1}) - \sigma(\varphi_{\iota,i}) w_\iota. \end{aligned}$$

Next, we prove that the coefficients of $\mathbf{s}, u_1, \ldots, u_k$ are binary. We simply define:

$$\mathbf{P}_s := \mathbf{I}_{\mathfrak{m}+k+1}, \quad \mathbf{P}_m := \mathbf{0}_{(\mathfrak{m}+k+1)\times k}, \quad \mathbf{f} := \mathbf{0}_{(\mathfrak{m}+k+1)\times 1}. \tag{8.26}$$

Then,

$$\mathbf{P}_s \mathbf{s}_1 + \mathbf{P}_m \mathbf{m} + \mathbf{f} = \mathbf{s} \parallel u_1 \parallel \cdots \parallel u_k \parallel v_{k+1}$$

is a binary vector. Further, we prove that the last $d - \mathfrak{d}$ coefficients of u_i are all zeroes for all $i \in [k]$. This is done by proving that for all $0 \leqslant \iota < d - \mathfrak{d}$, the constant coefficient of $X^{-\iota-\mathfrak{d}} u_i$ is equal to zero. Hence, we define:

$$\begin{aligned} \mathbf{R}'_{d+k\mathfrak{l}+\iota k+i,2} &:= \mathbf{0}_{2(\mathfrak{m}+2k+1)\times 2(\mathfrak{m}+2k+1)}, \\ \mathbf{r}'_{d+k\mathfrak{l}+\iota k+i,1} &:= \mathbf{J}^T \begin{bmatrix} \mathbf{0}_{\mathfrak{m}+i-1} \\ X^{-\iota-\mathfrak{d}} \\ \mathbf{0}_{(2k-i+1)\times 1} \end{bmatrix}, \qquad r'_{d+k\mathfrak{l}+\iota k+i,0} := 0. \end{aligned} \tag{8.27}$$

and by construction

$$\begin{aligned} \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma^T \mathbf{R}'_{d+k\mathfrak{l}+\iota k+i,2} \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma + \mathbf{r}'^T_{d+k\mathfrak{l}+\iota k+i,1} \langle \mathbf{s}_1 \parallel \mathbf{m} \rangle_\sigma + r_{d+k\mathfrak{l}+\iota k+i,0} \\ = X^{-\iota-\mathfrak{d}} u_i. \end{aligned}$$

Last but not least, we have to prove that each $u_1, \ldots, u_k, v_{k+1}$ contains exactly one 1. This is done by proving that the constant coefficients of

$$\sigma\left(\sum_{\iota=0}^{d-1} X^\iota\right) \cdot u_i \text{ for } i = 1, 2, \ldots, k, \quad \text{and} \quad \sigma\left(\sum_{\iota=0}^{d-1} X^\iota\right) \cdot v_{k+1}$$

vanish. Therefore, we define for $i \in [k+1]$:

$$\begin{aligned} \mathbf{R}'_{d+k(\mathfrak{l}+d-\mathfrak{d})+i,2} &:= \mathbf{0}_{2(\mathfrak{m}+2k+1)\times 2(\mathfrak{m}+2k+1)}, \\ \mathbf{r}'_{d+k(\mathfrak{l}+d-\mathfrak{d})+i,1} &:= \mathbf{J}^T \begin{bmatrix} \mathbf{0}_{\mathfrak{m}+i-1} \\ \sigma\left(\sum_{\iota=0}^{d-1} X^\iota\right) \\ \mathbf{0}_{(2k-i+1)\times 1} \end{bmatrix}, \qquad r'_{d+k(\mathfrak{l}+d-\mathfrak{d})+i,0} := 0. \end{aligned} \tag{8.28}$$

We present the commit-and-prove system $\Pi_k = (\mathsf{ABDLOP}, \mathcal{P}, \mathcal{V})$ for relation R_k in Figure 8.12. Here, we apply Π_{tbox} defined in Figure 6.3 without doing any approximate norm proof as described in Section 6.4.7.

8.5.1.3 *Recursive Step*

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

$$P_j(\vec{v}_{j+1} \otimes \cdots \otimes \vec{v}_{k+1}) = \vec{w}_j. \tag{8.29}$$

Prover $\mathcal{P}$	Verifier $\mathcal{V}$
Inputs:	
$pp.\dim = (q, d, \kappa_{\mathsf{MSIS}}, \mathsf{m}+k+1, m_2, k, \ell_{\mathsf{ext}} := 256/d + \lambda/2 + 2)$	$pp.\dim, pp.\mathsf{norms}$
$pp.\mathsf{norms} = (\omega, \sqrt{\mathsf{m}d+k+1}, B_1, B_2)$	$pp.\mathsf{mat}$
$pp.\mathsf{mat} = \left(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \begin{bmatrix}\mathbf{B}_y \\ \mathbf{B}_\beta \\ \mathbf{B}_{\mathsf{ext}} \\ \mathbf{b}_{\mathsf{ext}}^T\end{bmatrix}\right)$	$\mathbf{t}_A, \mathbf{t}_B$
$\mathbf{s}_1 := \mathbf{s} \parallel u_1 \parallel \cdots \parallel u_k \parallel v_{k+1} \in \{0,1\}^{(\mathsf{m}+k+1)d}, \mathbf{s}_2 \in \mathcal{R}_q^{m_2}$	$P_k \in \mathbb{Z}_q^{d\times d}$
$\mathbf{m} = (w_1, \ldots, w_k)$	$(\mathbf{a}_i)_{i\in[\mathfrak{l}]}$
$P_k \in \mathbb{Z}_q^{d\times d}, (\mathbf{a}_i)_{i\in[\mathfrak{l}]}, (\varphi_{\iota,i})_{\iota\in[k-1],i\in[\mathfrak{l}]}$	$(\varphi_{\iota,i})_{\iota\in[k-1],i\in[\mathfrak{l}]}$
$\begin{bmatrix}\mathbf{t}_A \\ \mathbf{t}_B\end{bmatrix} := \begin{bmatrix}\mathbf{A}_1 \\ \mathbf{0}\end{bmatrix}\mathbf{s}_1 + \begin{bmatrix}\mathbf{A}_2 \\ \mathbf{B}\end{bmatrix}\mathbf{s}_2 + \begin{bmatrix}\mathbf{0} \\ \mathbf{m}\end{bmatrix}$	
run Π_{tbox} with the following inputs:	accept if:
$pp := pp$	(i) Π_{tbox} verifies
$(\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m})) := (\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m}))$	
$(\mathbf{R}'_{i,2}, \mathbf{r}'_{i,1}, r'_{i,0})_{i\in[d+k(\mathfrak{l}+d-\mathfrak{d}+1)+1]}$ as in (8.23), (8.24), (8.25), (8.27), (8.28)	
$(\mathbf{P}_s, \mathbf{P}_m, \mathbf{f})$ as in (8.26)	

FIGURE 8.12: Commit-and-prove system Π_k for the relation R_k. Here, we use Π_{tbox} defined in Figure 6.3 but without an approximate norm proof.

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

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

Note that if (8.20) was not true, then these $\mathfrak{l}$ equations above would hold with probability at most $q_1^{-\mathfrak{l}}$. Now, if we write

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

then we have

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

where

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

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

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

Then, we need to show that for all i,

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

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

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

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

$$\vec{w}_{j+1} = P_{j+1}(\vec{v}_{j+2} \otimes \cdots \otimes \vec{v}_{k+1}) \quad \text{where} \quad P_{j+1} := \begin{bmatrix} P_{j+1,1} \\ \vdots \\ P_{j+1,\mathfrak{l}} \end{bmatrix} \in \mathbb{Z}_q^{d \times d\mathfrak{d}^{k-j-1}}. \tag{8.30}$$

Therefore, we reduced proving (8.29) to proving that

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

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

We provide a commit-and-prove system $\Pi_j = (\mathsf{ABDLOP}, \mathcal{P}, \mathcal{V})$ for relation R_j in Figure 8.13. The prover proceeds as described above and eventually runs Π_{j+1}.

In terms of security analysis, correctness follows by Theorem 6.4.1 and the argument presented above. Then, for simulatability, we observe that before running Π_{tbox}, the prover only sends the "bottom part" commitments to w_i and these (as a part of the whole ABDLOP commitment) can be simulated as in Theorem 6.4.2. Hence, we obtain the following results.

Prover $\mathcal{P}$ | Verifier $\mathcal{V}$

Inputs:

$pp.\dim = (q, d, \kappa_{\mathsf{MSIS}}, \mathsf{m}+k+1, m_2, j, \ell_{\mathsf{ext}} := k-j+256/d+\lambda/2+2)$ | $pp.\dim, pp.\mathsf{norms}$

$pp.\mathsf{norms} = (\omega, \sqrt{\mathsf{m}d+k+1}, B_1, B_2)$ | $pp.\mathsf{mat}$

$$pp.\mathsf{mat} = \left(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B}, \begin{bmatrix} \mathbf{B}_y \\ \mathbf{B}_\beta \\ \mathbf{b}_w^T \\ \mathbf{B}_{\mathsf{ext}} \\ \mathbf{b}_{\mathsf{ext}}^T \end{bmatrix}\right) \quad | \quad \mathbf{t}_A, \mathbf{t}_B$$

$\mathbf{s}_1 := \mathbf{s} \parallel u_1 \parallel \cdots \parallel u_k \parallel v_{k+1} \in \{0,1\}^{(\mathsf{m}+k+1)d}, \mathbf{s}_2 \in \mathcal{R}_q^{m_2}$ | $P_j \in \mathbb{Z}_q^{d \times d\mathfrak{d}^{k-j}}$

$\vec{u}_i := \vec{v}_i \parallel 0^{d-\mathfrak{d}}$ for $i = 1, 2, \ldots, k$ | $(\mathbf{a}_i)_{i\in[\mathfrak{l}]}$

$\mathbf{m} = (w_1, \ldots, w_j)$ | $(\varphi_{\iota,i})_{\iota\in[j-1],i\in[\mathfrak{l}]}$

$P_j \in \mathbb{Z}_q^{d\times d\mathfrak{d}^{k-j}}, (\mathbf{a}_i)_{i\in[\mathfrak{l}]}, (\varphi_{\iota,i})_{\iota\in[j-1],i\in[\mathfrak{l}]}$

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

$\xleftarrow{\vec{\varphi}_{j,i}}$ | $\vec{\varphi}_{j,1}, \ldots, \vec{\varphi}_{j,\mathfrak{l}} \leftarrow \mathbb{Z}_q^d$

$P_j := \begin{bmatrix} P_{j,1} & P_{j,2} & \cdots & P_{j,\mathfrak{d}} \end{bmatrix}$

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

$P_{j+1} := \begin{bmatrix} P_{(j+1,1)} \\ \vdots \\ P_{(j+1,\mathfrak{l})} \end{bmatrix} \in \mathbb{Z}_q^{d\times d\mathfrak{d}^{k-j-1}}$

$\vec{w}_{j+1} := P_{j+1}(\vec{v}_{j+2} \otimes \cdots \otimes \vec{v}_{k+1})$

$t_w := \mathbf{b}_w^T \mathbf{s}_2 + w_{j+1}$ | $\xrightarrow{t_w}$

run Π_{j+1} with the following inputs: | accept if:

- $pp.\dim = (q, d, \kappa_{\mathsf{MSIS}}, \mathsf{m}+k+1, m_2, j+1, \ell_{\mathsf{ext}}-1)$ | (i) Π_{j+1} verifies
- $pp.\mathsf{norms} = (\omega, \sqrt{\mathsf{m}d+k+1}, B_1, B_2)$
- $$pp.\mathsf{mat} = \left(\mathbf{A}_1, \mathbf{A}_2, \begin{bmatrix} \mathbf{B} \\ \mathbf{b}_w^T \end{bmatrix}, \begin{bmatrix} \mathbf{B}_y \\ \mathbf{B}_\beta \\ \mathbf{B}_{\mathsf{ext}} \\ \mathbf{b}_{\mathsf{ext}}^T \end{bmatrix}\right)$$
- $(\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m})) := (\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m} \parallel w_{j+1}))$
- $P_{j+1} \in \mathbb{Z}_q^{d\times d\mathfrak{d}^{k-j-1}}, (\mathbf{a}_i)_{i\in[\mathfrak{l}]}, (\varphi_{\iota,i})_{\iota\in[j],i\in[\mathfrak{l}]}$

FIGURE 8.13: Commit-and-prove system Π_j for the relation R_j defined in (8.22).

Theorem 8.5.2. *Fix $1 \leqslant j \leqslant k$ and let $\mathsf{Rej}^{(1)} = \mathsf{Rej}_0$ and $\mathsf{Rej}^{(2)} = \mathsf{Rej}^{(3)} = \mathsf{Rej}_1$ as defined in Figure 3.2. Fix standard deviations*

$$\mathfrak{s}_1 = \gamma_1 \eta \sqrt{\mathsf{m}d + k + 1}, \quad \mathfrak{s}_2 = \gamma_2 \eta \nu \sqrt{m_2 d}, \quad \mathfrak{s}_3 = \gamma_3 \sqrt{337 \cdot (\mathsf{m}d + k + 1)}$$

for some $\gamma_1, \gamma_2, \gamma_3 > 0$ and define

$$M_1 := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_1} + \frac{1}{2\gamma_1^2}\right) \quad \text{and} \quad M_i := \exp\left(\frac{1}{2\gamma_i^2}\right) \text{ for } i = 2, 3.$$

Suppose that $(\mathsf{m} + k + 1)d \geqslant 5\kappa$ and $m_2 d \geqslant 5\kappa$. Then, the commit-and-prove system Π_j for the relation R_j has statistical completeness with correctness error $1 - \frac{1}{M_1 M_2 M_3} + 2^{-127}$.

Theorem 8.5.3. *Fix $1 \leqslant j \leqslant k$ and let $\mathsf{Rej}^{(1)} = \mathsf{Rej}_0$ and $\mathsf{Rej}^{(2)} = \mathsf{Rej}^{(3)} = \mathsf{Rej}_1$ as defined in Figure 3.2. Fix standard deviations*

$$\mathfrak{s}_1 = \gamma_1 \eta \sqrt{\mathsf{m}d + k + 1}, \quad \mathfrak{s}_2 = \gamma_2 \eta \nu \sqrt{m_2 d}, \quad \mathfrak{s}_3 = \gamma_3 \sqrt{337 \cdot (\mathsf{m}d + k + 1)}$$

for some $\gamma_1, \gamma_2, \gamma_3 > 0$ and define

$$M_1 := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}} \cdot \frac{1}{\gamma_1} + \frac{1}{2\gamma_1^2}\right) \quad \text{and} \quad M_i := \exp\left(\frac{1}{2\gamma_i^2}\right) \text{ for } i = 2, 3.$$

Suppose $\kappa_{\mathsf{MLWE}} := m_2 - \kappa_{\mathsf{MSIS}} - (k - j) - \lambda/2 - 256/d - 2 \geqslant 0$. Then, under the Extended-$\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}}, \kappa_{\mathsf{MSIS}} + (k-j) + \lambda/2 + 256/d + 2, \chi, \mathcal{C}, D^d_{\mathfrak{s}_2}}$ assumption, the commit-and-prove system Π_j for relation R_j is simulatable.

Finally, we consider knowledge soundness.

Theorem 8.5.4. *Fix $1 \leqslant j \leqslant k$ and assume $k = O(\log \kappa)$. Suppose $B_1 \geqslant 2\mathfrak{s}_1 \sqrt{2(\mathsf{m} + k + 1)d}$ and $B_2 \geqslant 2\mathfrak{s}_2 \sqrt{2m_2 d}$. Let*

$$\mathfrak{s}_3 = \gamma_3 \sqrt{337(\mathsf{m}d + k + 1)}, \qquad \mathcal{B}_{\mathsf{arp}} := 2\sqrt{\frac{256}{26}} \varrho \mathfrak{s}_3$$

for $\gamma_3 > 0$. If q satisfies the following conditions

$$q \geqslant 41 \cdot (\mathsf{m} + k + 1)\, d \cdot \mathcal{B}_{\mathsf{arp}}, \qquad \text{to use Lemma 3.2.5}$$
$$q > \mathcal{B}^2_{\mathsf{arp}} + \mathcal{B}_{\mathsf{arp}} \sqrt{(\mathsf{m} + k + 1)d}, \qquad \text{to prove } \mathbf{P}_s + \mathbf{P}_m + \mathbf{f} \text{ has binary coeff.}$$

Then, the commit-and-prove system Π_{j} *for the relation* R_{j} *is knowledge sound with knowledge error*

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

Moreover, the extractor makes expected at most $2^{k-j} \cdot \mathsf{poly}(\kappa)$ *queries to the prover.*

Proof. We prove the statement by induction. First, consider $j = k$. Then, knowledge soundness follows directly from Theorem 6.4.3 (without an approximate norm proof) and the corresponding extractor makes at most expected $\mathsf{poly}(\kappa)$ queries to the prover.

Now, assume that $\Pi_{\mathrm{j}+1}$ is knowledge sound with knowledge error

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

for some $j+1 \leqslant k$. Also, denote $\mathcal{E}^*$ as the knowledge extractor for Π_{j+1} from the induction hypothesis.

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

We say that $\mathcal{A}$ succeeds if $\mathcal{A}$ outputs $((\varphi_{j,i}), t_w, \bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{w}_{j+1}, \bar{\mathbf{s}}_2, \bar{c})$ such that

$$\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}} \parallel \bar{w}_{j+1}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B \parallel t_w) = 1$$

and

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

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

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

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

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

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

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

- $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2) \neq (\bar{\mathbf{s}}'_1, \bar{\mathbf{s}}'_2)$ and

$$\begin{aligned} 1 &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B) \\ &= \mathsf{ABDLOP.Open}(\bar{\mathbf{s}}'_1, \bar{\mathbf{m}}', \bar{\mathbf{s}}'_2, \bar{c}'; \mathbf{t}_A \parallel \mathbf{t}_B). \end{aligned}$$

- $\mathsf{ABDLOP.Open}(\bar{\mathbf{s}}_1, \bar{\mathbf{m}}, \bar{\mathbf{s}}_2, \bar{c}; \mathbf{t}_A \parallel \mathbf{t}_B) = 1$ and

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

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

Claim 8.5.5. The expected number of calls to $\mathcal{A}$ is at most 2.

The proof follows identically as in Claim 5.2.7. We conclude that the expected runtime of $\mathcal{E}$ is at most $2^{k-j} \cdot \mathsf{poly}(\kappa) \cdot T$.

Claim 8.5.6. Probability that $\mathcal{E}$ succeeds is at least

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

One proves the statement similarly as e.g. Claim 5.2.8. The key idea here is that if

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

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

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

for random challenges $\vec{\varphi}'_{j,i}$. Then, we know these $\mathfrak{l}$ equations hold by construction of the matrix P_{j+1} and the relation R_{j+1}. Hence, $\mathcal{E}$ succeeds with probability at most the difference of $\mathcal{A}$ succeeding and $q_1^{-\mathfrak{l}}$.

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

8.5.2 *Commit-and-Prove System for R_{oom}*

Recall that Section 8.5.1 presents a way to reduce proving relation R_{oom} to R_1. Further, in Section 8.5.1.1 we propose a commit-and-prove system for relation R_1. Hence, we formally describe the commit-and-prove system Π_{oom} for relation R_{oom} in Figure 8.14. Below we state security properties of Π_{oom}, however we omit the proofs since they are almost identical to the ones included in Section 8.5.1.1.

Theorem 8.5.7. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}_0$ *and* $\mathsf{Rej}^{(2)} = \mathsf{Rej}^{(3)} = \mathsf{Rej}_1$ *as defined in Figure 3.2. Fix standard deviations*

$$\mathfrak{s}_1 = \gamma_1\eta\sqrt{\mathsf{m}d+k+1}, \quad \mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d}, \quad \mathfrak{s}_3 = \gamma_3\sqrt{337\cdot(\mathsf{m}d+k+1)}$$

for some $\gamma_1, \gamma_2, \gamma_3 > 0$ and define

$$M_1 := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}}\cdot\frac{1}{\gamma_1}+\frac{1}{2\gamma_1^2}\right) \quad \textit{and} \quad M_i := \exp\left(\frac{1}{2\gamma_i^2}\right) \textit{ for } i=2,3.$$

Suppose that $(\mathsf{m}+k+1)d \geqslant 5\kappa$ and $m_2 d \geqslant 5\kappa$. Then, the commit-and-prove system Π_{oom} for the relation R_{oom} has statistical completeness with correctness error $1-\frac{1}{M_1M_2M_3}+2^{-127}$.

Theorem 8.5.8. *Let* $\mathsf{Rej}^{(1)} = \mathsf{Rej}_0$ *and* $\mathsf{Rej}^{(2)} = \mathsf{Rej}^{(3)} = \mathsf{Rej}_1$ *as defined in Figure 3.2. Fix standard deviations*

$$\mathfrak{s}_1 = \gamma_1\eta\sqrt{\mathsf{m}d+k+1}, \quad \mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2 d}, \quad \mathfrak{s}_3 = \gamma_3\sqrt{337\cdot(\mathsf{m}d+k+1)}$$

for some $\gamma_1, \gamma_2, \gamma_3 > 0$ and define

$$M_1 := \exp\left(\sqrt{\frac{2(\kappa+1)}{\log(e)}}\cdot\frac{1}{\gamma_1}+\frac{1}{2\gamma_1^2}\right) \quad \textit{and} \quad M_i := \exp\left(\frac{1}{2\gamma_i^2}\right) \textit{ for } i=2,3.$$

Suppose $\kappa_{\mathsf{MLWE}} := m_2 - \kappa_{\mathsf{MSIS}} - k - \lambda/2 - 256/d - 2 \geqslant 0$. Then, under the Extended-$\mathsf{MLWE}_{\kappa_{\mathsf{MLWE}},\kappa_{\mathsf{MSIS}}+k+\lambda/2+256/d+2,\chi,\mathcal{C},D^d_{\mathfrak{s}_2}}$ assumption, Π_{oom} for relation R_{oom} is simulatable.

Theorem 8.5.9. *Assume $k = O(\log\kappa)$. Suppose $B_1 \geqslant 2\mathfrak{s}_1\sqrt{2(\mathsf{m}+k+1)d}$ and $B_2 \geqslant 2\mathfrak{s}_2\sqrt{2m_2 d}$. Let*

$$\mathfrak{s}_3 = \gamma_3\sqrt{337(\mathsf{m}d+k+1)}, \qquad \mathcal{B}_{\mathsf{arp}} := 2\sqrt{\frac{256}{26}}\varrho\mathfrak{s}_3$$

Prover $\mathcal{P}$ | Verifier $\mathcal{V}$

Inputs:

$pp.\dim = (q, d, \kappa_{\mathsf{MSIS}}, \mathsf{m}+k+1, m_2, 0, \ell_{\mathsf{ext}} := k + 256/d + \lambda/2 + 2)$ | $pp.\dim, pp.\mathsf{norms}$

$pp.\mathsf{norms} = (\omega, \sqrt{\mathsf{m}d+k+1}, B_1, B_2)$ | $pp.\mathsf{mat}$

$$pp.\mathsf{mat} = \left(\mathbf{A}_1, \mathbf{A}_2, \varnothing, \begin{bmatrix} \mathbf{B}_y \\ \mathbf{B}_\beta \\ \mathbf{b}_w^T \\ \mathbf{B}_{\mathsf{ext}} \\ \mathbf{b}_{\mathsf{ext}}^T \end{bmatrix} \right) \quad | \quad \mathbf{t}_A$$

$\mathbf{s}_1 := \mathbf{s} \parallel u_1 \parallel \cdots \parallel u_k \parallel v_{k+1} \in \{0,1\}^{(\mathsf{m}+k+1)d}, \mathbf{s}_2 \in \mathcal{R}_q^{m_2}$ | $P \in \mathbb{Z}_q^{\mathsf{n}d \times d\mathfrak{d}^k}$

$\vec{u}_i := \vec{v}_i \parallel 0^{d-\mathfrak{d}}$ for $i = 1, 2, \ldots, k$

$P \in \mathbb{Z}_q^{\mathsf{n}d \times d\mathfrak{d}^k}$

$\mathbf{t}_A := \mathbf{A}_1\mathbf{s}_1 + \mathbf{A}_2\mathbf{s}_2$

$\xleftarrow{\vec{\varphi}_{0,i}}$ | $\vec{\varphi}_{0,1}, \ldots, \vec{\varphi}_{0,\mathfrak{l}} \leftarrow \mathbb{Z}_q^{\mathsf{n}d}$

$$P := \begin{bmatrix} P_{0,1} & P_{0,2} & \cdots & P_{0,\mathfrak{d}} \end{bmatrix}$$

$$P_{1,i} := \begin{bmatrix} \vec{\varphi}_{0,i}^T P_{0,1} \\ \vdots \\ \vec{\varphi}_{0,i}^T P_{0,\mathfrak{d}} \end{bmatrix} \in \mathbb{Z}_q^{\mathfrak{d} \times d\mathfrak{d}^{k-1}} \text{ for } i \in [\mathfrak{l}]$$

$$P_1 := \begin{bmatrix} P^{(1,1)} \\ \vdots \\ P^{(1,\mathfrak{l})} \end{bmatrix} \in \mathbb{Z}_q^{d \times d\mathfrak{d}^{k-1}}$$

$\vec{w}_1 := P_1(\vec{v}_2 \otimes \cdots \otimes \vec{v}_{k+1})$

$t_w := \mathbf{b}_w^T\mathbf{s}_2 + w_1$ | $\xrightarrow{t_w}$

run Π_1 with the following inputs: | accept if:

$pp.\dim = (q, d, \kappa_{\mathsf{MSIS}}, \mathsf{m}+k+1, m_2, 1, \ell_{\mathsf{ext}} - 1)$ | (i) Π_1 verifies

$pp.\mathsf{norms} = (\omega, \sqrt{\mathsf{m}d+k+1}, B_1, B_2)$

$$pp.\mathsf{mat} = \left(\mathbf{A}_1, \mathbf{A}_2, \mathbf{b}_w^T, \begin{bmatrix} \mathbf{B}_y \\ \mathbf{B}_\beta \\ \mathbf{B}_{\mathsf{ext}} \\ \mathbf{b}_{\mathsf{ext}}^T \end{bmatrix} \right)$$

$(\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m})) := (\mathbf{s}_2, (\mathbf{s}_1, \mathbf{m} \parallel w_1))$

$P_1, (\mathbf{a}_i)_{i \in [\mathfrak{l}]}$, where $\vec{a}_i := A^T \vec{\varphi}_{0,i} \in \mathbb{Z}_q^{\mathsf{m}d}$

FIGURE 8.14: Commit-and-prove system Π_{oom} for the relation R_{oom}.

for $\gamma_3 > 0$. If q satisfies the following conditions

$$q \geqslant 41 \cdot (\mathsf{m}+k+1)\, d \cdot \mathcal{B}_{\mathsf{arp}}, \qquad \text{to use Lemma 3.2.5}$$
$$q > \mathcal{B}_{\mathsf{arp}}^2 + \mathcal{B}_{\mathsf{arp}}\sqrt{(\mathsf{m}+k+1)d}, \qquad \text{to prove } \mathbf{P}_s + \mathbf{P}_m + \mathbf{f} \text{ has binary coeff.}$$

Then, the commit-and-prove system Π_{oom} for the relation R_{oom} is knowledge sound with knowledge error

$$kq_1^{-\mathfrak{t}} + 2|\mathcal{C}|^{-1} + q_1^{-d/l} + q_1^{-\lambda} + 2^{-128}.$$

Let us consider the total proof size of our one-out-of-many proof. Based on Section 6.5.1 and highlighting that we do not perform any approximate norm proof, the total proof size becomes

$$\begin{aligned}&\kappa_{\mathsf{MSIS}} d(\lceil \log q \rceil - D) + (k + 256/d + \lambda + 2)\, d\lceil \log q \rceil + \lceil \log(2\omega+1) \rceil \cdot d \\ &+ (\mathsf{m}+k+1)d \cdot (2.57 + \lceil \log \mathfrak{s}_1 \rceil) + m_2 d \ \cdot (2.57 + \lceil \log \mathfrak{s}_2 \rceil) \\ &+ 2.25 \cdot \kappa_{\mathsf{MSIS}} d + 256 \cdot (2.57 + \lceil \log \mathfrak{s}_3 \rceil) \text{ bits.}\end{aligned}$$

which is logarithmic in the size of the set $\mathfrak{t} = d\mathfrak{d}^k$.

8.5.3 *Logarithmic-Size Ring Signature*

We sketch out the folklore approach to transform an one-out-of-many proof into a ring signature [Boo+15; Esg+19b; GK15; LNS21b]. Suppose we have a ring of $\mathfrak{t}$ users. Each user $i \in [\mathfrak{t}]$ has their associated private-public key $(\mathsf{sk}_i, \mathsf{pk}_i)$ such that $\mathsf{sk}_i := \mathbf{s}^{(i)} \leftarrow \{0,1\}^{\mathsf{m}d}$ and $\mathsf{pk}_i := \mathbf{A}\mathbf{s}^{(i)} \bmod p$ where $\mathbf{A} \leftarrow \mathcal{R}_p^{\mathsf{n}\times\mathsf{m}}$ and p is a modulus for the ring signature.

Now, user i signs a message by producing a non-interactive one-out-of-many proof, i.e. proof of knowledge of a vector $\mathbf{s}^{(i)}$ such that $\mathbf{s}^{(i)} \in \{0,1\}^{\mathsf{m}d}$ and

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

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

Anonymity property of the ring signature follows directly from simulatability of Π_{oom}. In order to argue unforgeability with respect to insider collusion, we proceed as in [Esg+19b, Theorem 3] and [LNS21b, Theorem C.4]. Namely, the reduction picks a uniformly random user j and sets a uniformly random public key $\mathsf{pk}_j \leftarrow \mathcal{R}_p^{\mathsf{n}}$ (under the $\mathsf{MLWE}_{\mathsf{n},\mathsf{m}-\mathsf{n},\mathfrak{D}}$ assumption where $\mathfrak{D}$ is the distribution over $\mathcal{R}_p$ so that each coefficient is sampled

t	k	proof size
2^6	0	13.9 KB
2^{12}	2	14.7 KB
2^{21}	5	16.2 KB

FIGURE 8.15: Ring signature sizes for $\mathsf{t} = d \cdot \mathfrak{d}^k$ users. For all parameter sets, we choose $(p, \mathsf{n}, \mathsf{m}, d, \mathfrak{d}) = (65437, 1, 12, 64, 8)$ and $q = 65437 \cdot 65629 \approx 2^{32}$.

uniformly at random from $\{0,1\}$). If there is any signing query to j, then the reduction simulates the one-out-of-many proof. Finally, the reduction will hope that: (i) the adversary does not make a corruption query to j and (ii) it forges a signature exactly for the public key pk_j. In this case, one can extract a secret key $\mathbf{s}^* \in \{0,1\}^{\mathsf{m}d}$ such that $\mathbf{A}\mathbf{s}^* = \mathsf{pk}_j$. Thus, $(\mathbf{s}^*, -1)$ is a non-zero vector of norm at most $\sqrt{\mathsf{m}d+1}$ which is a Module-SIS solution for the matrix $[\mathbf{A} \mid \mathsf{pk}_j] \in \mathcal{R}_p^{\mathsf{n}\times(\mathsf{m}+1)}$ and thus the reduction solves $\mathsf{MSIS}_{\mathsf{n},\mathsf{m}+1,\sqrt{\mathsf{m}d+1}}$.

In Figure 8.15, we present ring signature sizes for various rings of size between 2^6 and 2^{21}. We set $(p, \mathsf{n}, \mathsf{m}) = (65437, 1, 12)$ so that both the $\mathsf{MLWE}_{\mathsf{n},\mathsf{m}-\mathsf{n},\mathfrak{D}}$ and the $\mathsf{MSIS}_{\mathsf{n},\mathsf{m}+1,\sqrt{\mathsf{m}d+1}}$ problems are hard. Namely, since there is a reduction loss of $1/\mathsf{t}$, we pick the root Hermite factor $\delta \approx 1.0039$ for $\mathsf{MSIS}_{\mathsf{n},\mathsf{m}+1,\sqrt{\mathsf{m}d+1}}$ which should be enough for rings of size at most 2^{24}. In regard to $\mathsf{MLWE}_{\mathsf{n},\mathsf{m}-\mathsf{n},\mathfrak{D}}$, we aim for the root Hermite factor $\delta \approx 1.0044$ as in prior works. For such parameters, the user public key (resp. secret key) has size 128B (resp. 96B) which is more than one order of magnitude smaller than the public key in [LNS21b]. Next, we pick $(d, \mathfrak{d}, \mathfrak{l}) = (64, 8, 16)$. In all instantiations we picked $q_1 := p = 65437$ and $q_2 := 65629$ such that the proof system modulus $q = q_1 q_2 \approx 2^{32}$ and the repetition rate is ≈ 7 as in the previous examples.

9

CONCLUSION

In this thesis, we studied the problem of producing efficient zero-knowledge proofs for statements related to lattice-based cryptography. Our proposed framework Lantern performs very well compared to prior work with around a factor of $2-3X$ improvement over the previous works for basic statements, such as proving knowledge of a Module-LWE sample. Our protocol has the advantage over prior works in a sense that it does not rely on the CRT technique anymore. In particular, we can choose a prime q such that X^d+1 does not split into many factors modulo q (even two) at no extra cost[1]. This comes with a huge benefit that we do not need to repeat any (costly) part of the protocol for soundness amplification, thus making our proofs *one-shot*.

We provide new technical tools for proving various relations in the committed messages, such as inner products (involving either one or two secret vectors) and norm bounds which make use of the algebraic properties of the $\mathcal{R}_q$-automorphism σ_{-1}. We believe that they can be of independent interest for building more advanced privacy-preserving protocols.

As a final objective of the thesis, we applied our framework as a (black-box) building block to construct more efficient privacy-oriented primitives, such as verifiable encryption, proving integer relations, ring signatures and group signatures. As evidenced in e.g. [ESZ21; Esg+19c; LNS21b; TW04], these components can be used further for designing more sophisticated protocols, such as cryptocurrencies or secure e-voting.

9.1 FUTURE RESEARCH DIRECTIONS

IMPLEMENTATION. Basic primitives based on lattices (e.g. encryption and signature schemes) are renowned for their fast runtimes. Indeed, the operations involved in lattice constructions have been shown to be readily ported to more constrained devices. This opens up the possibility of quantum-safe zero-knowledge proofs being used in "daily" interactions, e.g. credit card transactions, where operations should take (significantly) less time than a second.

1 This choice of a modulus might, however, have an impact on the protocol implementation. We leave this aspect as a future work.

It is an open question how our protocol performs in terms of the computational complexity. We recall that for the previous state-of-the-art lattice-based proof system by Esgin et al. [ENS20], the prover (resp. verifier) runtime is about 3.5ms (resp. 0.4ms). Unfortunately, we cannot precisely extrapolate their results to our setting due to the following two reasons. First, our modulus is not "NTT-friendly", meaning $X^d + 1$ does not split into many small factors. Hence, we cannot apply the standard fast algorithms for polynomial multiplication[2] as in [ENS20]. Secondly, Esgin et al. use uniform rejection sampling and thus do not require efficient algorithms for sampling from discrete Gaussians. This is not the case in our framework since we explicitly provide new results for Gaussian rejection sampling.

FURTHER APPLICATIONS. A clear future direction is using our framework in the context of other privacy-oriented applications. Indeed, some currently most efficient lattice-based schemes, e.g. e-cash [Deo+20] or group encryption with full dynamicity and message filtering policy [Pan+21], are still based on the protocol by Yang et al. [Yan+19], and surprisingly not the works which significantly build upon it [ENS20; LNS21a]. The reason is that Yang et al. present a general protocol for proving so-called "instance relations", i.e. prove knowledge of a vector $\vec{s}$ over $\mathbb{Z}_q$ such that $A\vec{s} = \vec{t}$ and for each triple (i, j, k) of indices in a fixed set $\mathcal{M}$, we have $s_i \cdot s_j = s_k$[3]. These specific statements were not considered explicitly in [ENS20], nor in this thesis, and thus our framework cannot be applied in such applications out-of-the-box. This raises a question whether our protocols can be easily modified to prove "instance relations" which would consequently improve the efficiency of [Deo+20; Pan+21].

SUBLINEAR PROOFS. Asymptotically, our framework provides proofs which are linear in the number of committed messages. Hence, it is not very suitable for proving larger statements, such as circuit satisfiability.

As discussed in Chapter 2, various lattice-based protocols with asymptotically succinct proofs have been introduced. However, these constructions fall short in practice since they require very large parameters to instantiate, especially in comparison to PCP/IOP-type constructions [Ben+19; Bha+20; COS20] which are also (plausibly) post-quantum. We recall that the aforementioned schemes offer proofs in the order of 100KB for proving arbitrary circuits with millions of gates. The bottleneck of the PCP-type

2 However, there is a recent work by Chung et al. [Chu+21] which provides fast polynomial multiplication for "NTT-unfriendly" rings and might be useful for our setting.

3 Clearly, if $\mathcal{M}$ contains triples of the form (i, i, i) then $\vec{s}$ has binary coefficients.

constructions is arguably the prover runtime which in the order of tens of seconds for even small instances. This is clearly evidenced by the work by Boschini et al. [Bos+20] who built a group signature using the Aurora proof system [Ben+19]. Namely, they computed that proving knowledge of a Module-LWE sample takes around 40 seconds on a standard laptop. What is worse, they could not successfully run the full signing algorithm, even with the help of Google Cloud large-memory machines due to very large memory requirements. This raises a very important future research direction, from both theoretical and practical point of view, i.e. to construct a *concretely* efficient sublinear-size lattice-based zero-knowledge proof system which enjoys fast implementation and small memory requirements while producing comparable proof sizes to the PCP-type systems.

BIBLIOGRAPHY

[Ach03] Dimitris Achlioptas. "Database-friendly random projections: Johnson-Lindenstrauss with binary coins". In: *J. Comput. Syst. Sci.* 66.4 (2003), 671.

[ABB10a] Shweta Agrawal, Dan Boneh, and Xavier Boyen. "Efficient Lattice (H)IBE in the Standard Model". In: *EUROCRYPT*. 2010, 553.

[ABB10b] Shweta Agrawal, Dan Boneh, and Xavier Boyen. "Lattice Basis Delegation in Fixed Dimension and Shorter-Ciphertext Hierarchical IBE". In: *CRYPTO*. 2010, 98.

[Ajt96] Miklós Ajtai. "Generating Hard Instances of Lattice Problems (Extended Abstract)". In: *STOC*. 1996, 99.

[AL21] Martin R. Albrecht and Russell W. F. Lai. "Subtractive Sets over Cyclotomic Rings - Limits of Schnorr-Like Arguments over Lattices". In: *CRYPTO (2)*. Vol. 12826. Lecture Notes in Computer Science. Springer, 2021, 519.

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. *On the concrete hardness of Learning with Errors*. Cryptology ePrint Archive, Report 2015/046. https://ia.cr/2015/046. 2015.

[AHJ21] Nabil Alkeilani Alkadri, Patrick Harasser, and Christian Janson. *BlindOR: An Efficient Lattice-Based Blind Signature Scheme from OR-Proofs*. Cryptology ePrint Archive, Report 2021/1385. https://ia.cr/2021/1385. 2021.

[Alk+16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. "Post-quantum Key Exchange - A New Hope". In: *USENIX Security Symposium*. USENIX Association, 2016, 327.

[AA16] Jacob Alperin-Sheriff and Daniel Apon. *Dimension-Preserving Reductions from LWE to LWR*. Cryptology ePrint Archive, Report 2016/589. https://ia.cr/2016/589. 2016.

[AP12] Jacob Alperin-Sheriff and Chris Peikert. "Circular and KDM Security for Identity-Based Encryption". In: *Public Key Cryptography*. Vol. 7293. Lecture Notes in Computer Science. Springer, 2012, 334.

[Ame+17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. "Ligero: Lightweight Sublinear Arguments Without a Trusted Setup". In: *ACM Conference on Computer and Communications Security*. ACM, 2017, 2087.

[AG11] Sanjeev Arora and Rong Ge. "New Algorithms for Learning in Presence of Errors". In: *ICALP (1)*. 2011, 403.

[ACK21] Thomas Attema, Ronald Cramer, and Lisa Kohl. "A Compressed Σ-Protocol Theory for Lattices". In: *CRYPTO (2)*. Vol. 12826. Lecture Notes in Computer Science. Springer, 2021, 549.

[AFK21] Thomas Attema, Serge Fehr, and Michael Klooß. "Fiat-Shamir Transformation of Multi-Round Interactive Proofs". In: *IACR Cryptol. ePrint Arch.* (2021), 1377.

[ALS20] Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler. "Practical Product Proofs for Lattice Commitments". In: *CRYPTO (2)*. Vol. 12171. Lecture Notes in Computer Science. Springer, 2020, 470.

[BG14] Shi Bai and Steven D. Galbraith. "An Improved Compression Technique for Signatures Based on Learning with Errors". In: *CT-RSA*. 2014, 28.

[Ban93] Wojciech Banaszczyk. "New bounds in some transference theorems in the geometry of numbers". In: *Mathematische Annalen* 296 (1993), 625.

[Bau+18a] Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. "Sub-linear Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits". In: *CRYPTO*. 2018, 669.

[Bau+18b] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. "More Efficient Commitments from Structured Lattice Assumptions". In: *SCN*. 2018, 368.

[BL17] Carsten Baum and Vadim Lyubashevsky. "Simple Amortized Proofs of Shortness for Linear Relations over Polynomial Rings". In: *IACR Cryptology ePrint Archive* 2017 (2017), 759.

[Bec+16] Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. "New directions in nearest neighbor searching with applications to lattice sieving". In: *SODA*. SIAM, 2016, 10.

[BMW03] Mihir Bellare, Daniele Micciancio, and Bogdan Warinschi. "Foundations of Group Signatures: Formal Definitions, Simplified Requirements, and a Construction Based on General Assumptions". In: *EUROCRYPT*. 2003, 614.

[BN06] Mihir Bellare and Gregory Neven. "Multi-signatures in the plain public-Key model and a general forking lemma". In: *ACM Conference on Computer and Communications Security*. 2006, 390.

[Ben+19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. "Aurora: Transparent Succinct Arguments for R1CS". In: *EUROCRYPT (1)*. Vol. 11476. Lecture Notes in Computer Science. Springer, 2019, 103.

[Ben+14] Fabrice Benhamouda, Jan Camenisch, Stephan Krenn, Vadim Lyubashevsky, and Gregory Neven. "Better Zero-Knowledge Proofs for Lattice Encryption and Their Application to Group Signatures". In: *Advances in Cryptology - ASIACRYPT 2014*. Ed. by Palash Sarkar and Tetsu Iwata. Vol. 8873. Lecture Notes in Computer Science. Springer, 2014, 551.

[Beu20] Ward Beullens. "Sigma Protocols for MQ, PKP and SIS, and Fishy Signature Schemes". In: *EUROCRYPT (3)*. Vol. 12107. Lecture Notes in Computer Science. Springer, 2020, 183.

[Beu+21] Ward Beullens, Samuel Dobson, Shuichi Katsumata, Yi-Fu Lai, and Federico Pintore. "Group Signatures and More from Isogenies and Lattices: Generic, Simple, and Efficient". In: *IACR Cryptol. ePrint Arch.* (2021), 1366.

[BKP20] Ward Beullens, Shuichi Katsumata, and Federico Pintore. "Calamari and Falafl: Logarithmic (Linkable) Ring Signatures from Isogenies and Lattices". In: *ASIACRYPT (2)*. Vol. 12492. Lecture Notes in Computer Science. Springer, 2020, 464.

[Bha+20] Rishabh Bhadauria, Zhiyong Fang, Carmit Hazay, Muthuramakrishnan Venkitasubramaniam, Tiancheng Xie, and Yupeng Zhang. "Ligero++: A New Optimized Sublinear IOP". In: *CCS*. ACM, 2020, 2025.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. "Noise-tolerant learning, the parity problem, and the statistical query model". In: *J. ACM* 50.4 (2003), 506.

[Bon+11] Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. "Random Oracles in a Quantum World". In: *ASIACRYPT*. Vol. 7073. Lecture Notes in Computer Science. Springer, 2011, 41.

[Boo+15] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Essam Ghadafi, Jens Groth, and Christophe Petit. "Short Accountable Ring Signatures Based on DDH". In: *ESORICS (1)*. Vol. 9326. Lecture Notes in Computer Science. Springer, 2015, 243.

[Boo+16] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. "Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting". In: *EUROCRYPT (2)*. Vol. 9666. Lecture Notes in Computer Science. Springer, 2016, 327.

[BCS21] Jonathan Bootle, Alessandro Chiesa, and Katerina Sotiraki. "Sumcheck Arguments and Their Applications". In: *CRYPTO (1)*. Vol. 12825. Lecture Notes in Computer Science. Springer, 2021, 742.

[Boo+20] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. "A Non-PCP Approach to Succinct Quantum-Safe Zero-Knowledge". In: *CRYPTO (2)*. Vol. 12171. Lecture Notes in Computer Science. Springer, 2020, 441.

[BLS19] Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. "Algebraic Techniques for Short(er) Exact Lattice-Based Zero-Knowledge Proofs". In: *CRYPTO (1)*. Vol. 11692. Lecture Notes in Computer Science. Springer, 2019, 176.

[Bos+18] Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM". In: *2018 IEEE European Symposium on Security and Privacy, EuroS&P*. 2018, 353.

[BCN18] Cecilia Boschini, Jan Camenisch, and Gregory Neven. "Relaxed Lattice-Based Signatures with Short Zero-Knowledge Proofs". In: *ISC*. Vol. 11060. Lecture Notes in Computer Science. Springer, 2018, 3.

[Bos+20] Cecilia Boschini, Jan Camenisch, Max Ovsiankin, and Nicholas Spooner. "Efficient Post-quantum SNARKs for RSIS and RLWE and Their Applications to Privacy". In: *PQCrypto*. Vol. 12100. Lecture Notes in Computer Science. Springer, 2020, 247.

[Bou+21] Katharina Boudgoust, Corentin Jeudy, Adeline Roux-Langlois, and Weiqiang Wen. "On the Hardness of Module-LWE with Binary Secret". In: *CT-RSA*. Vol. 12704. Lecture Notes in Computer Science. Springer, 2021, 503.

[Bün+18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Gregory Maxwell. "Bulletproofs: Short Proofs for Confidential Transactions and More". In: *IEEE Symposium on Security and Privacy*. 2018, 315.

[Can+02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. "Universally composable two-party and multi-party secure computation". In: *STOC*. ACM, 2002, 494.

[CH91] David Chaum and Eugène van Heyst. "Group Signatures". In: *EUROCRYPT*. Ed. by Donald W. Davies. Vol. 547. Lecture Notes in Computer Science. Springer, 1991, 257.

[CN11] Yuanmi Chen and Phong Q. Nguyen. "BKZ 2.0: Better Lattice Security Estimates". In: *ASIACRYPT*. Vol. 7073. Lecture Notes in Computer Science. Springer, 2011, 1.

[COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. "Fractal: Post-quantum and Transparent Recursive Proofs from Holography". In: *EUROCRYPT (1)*. Vol. 12105. Lecture Notes in Computer Science. Springer, 2020, 769.

[Chu+21] Chi-Ming Marvin Chung, Vincent Hwang, Matthias J. Kannwischer, Gregor Seiler, Cheng-Jhih Shih, and Bo-Yin Yang. "NTT Multiplication for NTT-unfriendly Rings New Speed Records for Saber and NTRU on Cortex-M4 and AVX2". In: *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2021.2 (2021), 159.

[DAn+18] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. "Saber: Module-LWR Based Key Exchange, CPA-Secure Encryption and CCA-Secure KEM". In: *AFRICACRYPT*. 2018, 282.

[Dac+20] Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. "LWE with Side Information: Attacks and Concrete Security Estimation". In: *CRYPTO (2)*. Vol. 12171. Lecture Notes in Computer Science. Springer, 2020, 329.

[Dam10] Ivan Damgård. *On Σ-Protocols*. Lecture on Cryptologic Protocol Theory; Faculty of Science, University of Aarhus. 2010.

[Dam+21] Ivan Damgård, Claudio Orlandi, Akira Takahashi, and Mehdi Tibouchi. "Two-Round n-out-of-n and Multi-signatures and Trapdoor Commitment from Lattices". In: *Public Key Cryptography (1)*. Vol. 12710. Lecture Notes in Computer Science. Springer, 2021, 99.

[Deo+20] Amit Deo, Benoit Libert, Khoa Nguyen, and Olivier Sanders. "Lattice-Based E-Cash, Revisited". In: *ASIACRYPT (2)*. Vol. 12492. Lecture Notes in Computer Science. Springer, 2020, 318.

[Dod+10] Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert, and Vinod Vaikuntanathan. "Public-Key Encryption Schemes with Auxiliary Inputs". In: *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*. Ed. by Daniele Micciancio. 2010.

[DFM20] Jelle Don, Serge Fehr, and Christian Majenz. "The Measure-and-Reprogram Technique 2.0: Multi-Round Fiat-Shamir and More". In: *CoRR* abs/2003.05207 (2020).

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

[Duc+13] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. "Lattice Signatures and Bimodal Gaussians". In: *CRYPTO (1)*. 2013, 40.

[Duc+18] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme". In: *IACR Trans. Cryptogr. Hardw. Embed. Syst.* 2018.1 (2018), 238.

[DLP14] Léo Ducas, Vadim Lyubashevsky, and Thomas Prest. "Efficient Identity-Based Encryption over NTRU Lattices". In: *ASIACRYPT*. 2014, 22.

[EG14] Alex Escala and Jens Groth. "Fine-Tuning Groth-Sahai Proofs". In: *Public Key Cryptography*. Vol. 8383. Lecture Notes in Computer Science. Springer, 2014, 630.

[Esg+21] Muhammed F. Esgin, Veronika Kuchta, Amin Sakzad, Ron Steinfeld, Zhenfei Zhang, Shifeng Sun, and Shumo Chu. "Practical Post-quantum Few-Time Verifiable Random Function with Applications to Algorand". In: *Financial Cryptography (2)*. Vol. 12675. Lecture Notes in Computer Science. Springer, 2021, 560.

[ENS20] Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. "Practical Exact Proofs from Lattices: New Techniques to Exploit Fully-Splitting Rings". In: *ASIACRYPT (2)*. 2020, 259.

[Esg+22] Muhammed F. Esgin, Ron Steinfeld, Dongxi Liu, and Sushmita Ruj. "Efficient Hybrid Exact/Relaxed Lattice Proofs and Applications to Rounding and VRFs". In: *IACR Cryptol. ePrint Arch.* (2022), 141.

[Esg+19a] Muhammed F. Esgin, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. "Lattice-Based Zero-Knowledge Proofs: New Techniques for Shorter and Faster Constructions and Applications". In: *CRYPTO (1)*. Vol. 11692. Lecture Notes in Computer Science. Springer, 2019, 115.

[Esg+19b] Muhammed F. Esgin, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, and Dongxi Liu. "Short Lattice-Based One-out-of-Many Proofs and Applications to Ring Signatures". In: *ACNS*. Vol. 11464. Lecture Notes in Computer Science. Springer, 2019, 67.

[ESZ21] Muhammed F. Esgin, Ron Steinfeld, and Raymond K. Zhao. *MatRiCT+: More Efficient Post-Quantum Private Blockchain Payments*. Cryptology ePrint Archive, Report 2021/545. `https://ia.cr/2021/545`. 2021.

[Esg+19c] Muhammed F. Esgin, Raymond K. Zhao, Ron Steinfeld, Joseph K. Liu, and Dongxi Liu. "MatRiCT: Efficient, Scalable and Post-Quantum Blockchain Confidential Transactions Protocol". In: *CCS*. ACM, 2019, 567.

[FS86] Amos Fiat and Adi Shamir. "How to Prove Yourself: Practical Solutions to Identification and Signature Problems". In: *CRYPTO*. 1986, 186.

[GHL21] Craig Gentry, Shai Halevi, and Vadim Lyubashevsky. "Practical Non-interactive Publicly Verifiable Secret Sharing with Thousands of Parties". In: *IACR Cryptol. ePrint Arch.* (2021), 1397.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract)". In: *STOC*. ACM, 1985, 291.

[GKV10] S. Dov Gordon, Jonathan Katz, and Vinod Vaikuntanathan. "A Group Signature Scheme from Lattice Assumptions". In: *Advances in Cryptology - ASIACRYPT 2010*. 2010, 395.

[GK15] Jens Groth and Markulf Kohlweiss. "One-Out-of-Many Proofs: Or How to Leak a Secret and Spend a Coin". In: *EUROCRYPT*. 2015, 253.

[HKL19] Eduard Hauck, Eike Kiltz, and Julian Loss. "A Modular Treatment of Blind Signatures from Identification Schemes". In: *EUROCRYPT (3)*. Vol. 11478. Lecture Notes in Computer Science. Springer, 2019, 345.

[HS03] Javier Herranz and Germán Sáez. "Forking Lemmas for Ring Signature Schemes". In: *INDOCRYPT*. Vol. 2904. Lecture Notes in Computer Science. Springer, 2003, 266.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. "NTRU: A Ring-Based Public Key Cryptosystem". In: *ANTS*. 1998, 267.

[Kat21] Shuichi Katsumata. "A New Simple Technique to Bootstrap Various Lattice Zero-Knowledge Proofs to QROM Secure NIZKs". In: *CRYPTO (2)*. Vol. 12826. Lecture Notes in Computer Science. Springer, 2021, 580.

[KTX08] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. "Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems". In: *ASIACRYPT*. 2008, 372.

[LS15] Adeline Langlois and Damien Stehlé. "Worst-case to average-case reductions for module lattices". In: *Des. Codes Cryptography* 75.3 (2015), 565.

[Lib+18] Benoit Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. "Lattice-Based Zero-Knowledge Arguments for Integer Relations". In: *CRYPTO (2)*. Vol. 10992. Lecture Notes in Computer Science. Springer, 2018, 700.

[Lib+16] Benoit Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. "Zero-Knowledge Arguments for Lattice-Based Accumulators: Logarithmic-Size Ring Signatures and Group Signatures Without Trapdoors". In: *EUROCRYPT (2)*. Vol. 9666. Lecture Notes in Computer Science. Springer, 2016, 1.

[Lib+17] Benoit Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. "Zero-Knowledge Arguments for Lattice-Based PRFs and Applications to E-Cash". In: *ASIACRYPT (3)*. Vol. 10626. Lecture Notes in Computer Science. Springer, 2017, 304.

[Lin+13] San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. "Improved Zero-Knowledge Proofs of Knowledge for the ISIS Problem, and Applications". In: *PKC*. 2013, 107.

[Lin+17] San Ling, Khoa Nguyen, Huaxiong Wang, and Yanhong Xu. "Lattice-Based Group Signatures: Achieving Full Dynamicity with Ease". In: *ACNS*. Vol. 10355. Lecture Notes in Computer Science. Springer, 2017, 293.

[LAZ19] Xingye Lu, Man Ho Au, and Zhenfei Zhang. "Raptor: A Practical Lattice-Based (Linkable) Ring Signature". In: *ACNS*. Vol. 11464. Lecture Notes in Computer Science. Springer, 2019, 110.

[Lyu09] Vadim Lyubashevsky. "Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures". In: *ASIACRYPT*. 2009, 598.

[Lyu12] Vadim Lyubashevsky. "Lattice Signatures Without Trapdoors". In: *EUROCRYPT*. 2012, 738.

[LN17] Vadim Lyubashevsky and Gregory Neven. "One-Shot Verifiable Encryption from Lattices". In: *EUROCRYPT*. 2017.

[LNP22a] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. "Efficient Lattice-Based Blind Signatures via Gaussian One-Time Signatures". In: *Public Key Cryptography (2)*. Vol. 13178. Lecture Notes in Computer Science. Springer, 2022, 498.

[LNP22b] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. "Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General". In: *IACR Cryptol. ePrint Arch.* (2022), 284.

[Lyu+21] Vadim Lyubashevsky, Ngoc Khanh Nguyen, Maxime Plançon, and Gregor Seiler. "Shorter Lattice-Based Group Signatures via "Almost Free" Encryption and Other Optimizations". In: *ASIACRYPT (4)*. Vol. 13093. Lecture Notes in Computer Science. Springer, 2021, 218.

[LNS20] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. "Practical Lattice-Based Zero-Knowledge Proofs for Integer Relations". In: *CCS*. ACM, 2020, 1051.

[LNS21a] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. "Shorter Lattice-Based Zero-Knowledge Proofs via One-Time Commitments". In: *Public Key Cryptography (1)*. Vol. 12710. Lecture Notes in Computer Science. Springer, 2021, 215.

[LNS21b] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. "SMILE: Set Membership from Ideal Lattices with Applications to Ring Signatures and Confidential Transactions". In: *CRYPTO (2)*. Vol. 12826. Lecture Notes in Computer Science. Springer, 2021, 611.

[LS18] Vadim Lyubashevsky and Gregor Seiler. "Short, Invertible Elements in Partially Splitting Cyclotomic Rings and Applications to Lattice-Based Zero-Knowledge Proofs". In: *EUROCRYPT (1)*. Vol. 10820. Lecture Notes in Computer Science. Springer, 2018, 204.

[Mic07] Daniele Micciancio. "Generalized Compact Knapsacks, Cyclic Lattices, and Efficient One-Way Functions". In: *Computational Complexity* 16.4 (2007), 365.

[MM11] Daniele Micciancio and Petros Mol. "Pseudorandom Knapsacks and the Sample Complexity of LWE Search-to-Decision Reductions". In: *CRYPTO*. Vol. 6841. Lecture Notes in Computer Science. Springer, 2011, 465.

[MP12] Daniele Micciancio and Chris Peikert. "Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller". In: *EUROCRYPT*. 2012, 700.

[MR09] Daniele Micciancio and Oded Regev. "Lattice-based cryptography". In: *Post-quantum cryptography*. Springer, 2009, 147.

[NIS] NIST. *NIST Post-Quantum Cryptography Standardization*. `https://csrc.nist.gov/Projects/post-quantum-cryptography`.

[NM16] Shen Noether and Adam Mackenzie. "Ring Confidential Transactions". In: *Ledger* 1 (2016), 1.

[Pan+21] Jing Pan, Xiaofeng Chen, Fangguo Zhang, and Willy Susilo. "Lattice-Based Group Encryption with Full Dynamicity and Message Filtering Policy". In: *ASIACRYPT (4)*. Vol. 13093. Lecture Notes in Computer Science. Springer, 2021, 156.

[PLS18] Rafaël del Pino, Vadim Lyubashevsky, and Gregor Seiler. "Lattice-Based Group Signatures and Zero-Knowledge Proofs of Automorphism Stability". In: *ACM Conference on Computer and Communications Security*. ACM, 2018, 574.

[PLS19] Rafaël del Pino, Vadim Lyubashevsky, and Gregor Seiler. "Short Discrete Log Proofs for FHE and Ring-LWE Ciphertexts". In: *Public Key Cryptography (1)*. Vol. 11442. Lecture Notes in Computer Science. Springer, 2019, 344.

[PS00] David Pointcheval and Jacques Stern. "Security Arguments for Digital Signatures and Blind Signatures". In: *J. Cryptol.* 13.3 (2000), 361.

[Reg09] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *J. ACM* 56.6 (2009).

[RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. "How to Leak a Secret". In: *ASIACRYPT*. Vol. 2248. Lecture Notes in Computer Science. Springer, 2001, 552.

[The22] The Sage Developers. *SageMath, the Sage Mathematics Software System (Version 9.3)*. `https://www.sagemath.org`. 2022.

[Sch89] Claus-Peter Schnorr. "Efficient Identification and Signatures for Smart Cards". In: *CRYPTO*. 1989, 239.

[SE94] Claus-Peter Schnorr and M. Euchner. "Lattice basis reduction: Improved practical algorithms and solving subset sum problems". In: *Math. Program.* 66 (1994), 181.

[Ste93] Jacques Stern. "A New Identification Scheme Based on Syndrome Decoding". In: *CRYPTO*. 1993, 13.

[TWZ20] Yang Tao, Xi Wang, and Rui Zhang. "Short Zero-Knowledge Proof of Knowledge for Lattice-Based Commitment". In: *PQCrypto*. Vol. 12100. Lecture Notes in Computer Science. Springer, 2020, 268.

[TW04] Patrick P. Tsang and Victor K. Wei. "Short Linkable Ring Signatures for E-Voting, E-Cash and Attestation". In: *IACR Cryptol. ePrint Arch.* (2004), 281.

[Unr15] Dominique Unruh. "Non-Interactive Zero-Knowledge Proofs in the Quantum Random Oracle Model". In: *EUROCRYPT (2)*. Vol. 9057. Lecture Notes in Computer Science. Springer, 2015, 755.

[Yan+19] Rupeng Yang, Man Ho Au, Zhenfei Zhang, Qiuliang Xu, Zuoxia Yu, and William Whyte. "Efficient Lattice-Based Zero-Knowledge Arguments with Standard Soundness: Construction and Applications". In: *CRYPTO (1)*. Vol. 11692. Lecture Notes in Computer Science. Springer, 2019, 147.

[Yue+21] Tsz Hon Yuen, Muhammed F. Esgin, Joseph K. Liu, Man Ho Au, and Zhimin Ding. "DualRing: Generic Construction of Ring Signatures with Efficient Instantiations". In: *CRYPTO (1)*. Vol. 12825. Lecture Notes in Computer Science. Springer, 2021, 251.