Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

Tiril Stjernberg

# Lattice-Based Zero-Knowledge Proofs From Commitments

Master's thesis in Applied Physics and Mathematics
Supervisor: Kristian Gjøsteen
June 2023

NTNU
Norwegian University of
Science and Technology

Tiril Stjernberg

# Lattice-Based Zero-Knowledge Proofs From Commitments

Master's thesis in Applied Physics and Mathematics
Supervisor: Kristian Gjøsteen
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU
Norwegian University of
Science and Technology

# Abstract

The main purpose of this thesis is to study the state of the art lattice-based zero-knowledge protocol that was given by Lyubashevsky, Nguyen and Plançon. We start by introducing the cryptographic definitions and mathematical theory that we need. In order to fully understand the scheme, we look at some of the previous lattice-based zero-knowledge schemes that lead up to it. All of these protocols uses lattice-based commitment schemes, and we dedicate a chapter to look at two commitment schemes for lattice elements and their opening proofs. In the final chapter we end up with a practical protocol that can be used in a variety of lattice-based cryptographic systems.

# Sammendrag

Hovedformålet med denne oppgaven er å studere den nyeste lattice baserte zero-knowledge protokollen som ble introdusert av Lyubashevsky, Nguyen og Plançon. Vi starter med å introdusere de kryptografiske definisjonene og den matematiske teorien vi trenger. For å oppnå full forståelse av protokollen, ser vi på noen tidligere lattice baserte zero-knowledge protokoller som førte frem til denne. Alle disse protokollene bruker lattice baserte commitment systemer, og vi vier ett kapittel til å se på to ulike commitment systemer for lattice elementer og tilhørende bevis av åpning. I det siste kapittelet kommer vi frem til en praktisk protokoll som kan bli brukt i en mengde ulike lattice baserte kryptografiske systemer.

# Contents

# Chapter 1

# Introduction

Public key cryptography bases its security on mathematical problems that are hard to solve. Traditionally, the factorization problem and the discrete logarithm problem have been used, as the best known techniques for solving these problems would take so much time that the scheme is practically secure. However, the field of quantum computing poses a huge threat to such public-key cryptography, as it could rather easily break the security of these schemes. This is due to Shor's algorithm, developed by Peter Shor in 1994 [25], that when run on an efficient quantum computer, could solve the factorization problem and the discrete logarithm problem in much less time.

Hence the need for post-quantum cryptography, cryptographic schemes that are secure even with the existence of powerful quantum computers, arise. One of the currently strong candidates is lattice-based cryptography. This is because there are lattice-problems believed to be hard to solve even for a quantum computer, all the while lattices allow for the construction of practical schemes.

One cryptographic system that is important in many applications is zero-knowledge proofs. These are schemes that allows a prover to prove that a given statement is true, or that it knows some secret information, without revealing any additional information. Such schemes can be used for instance to enforce honest behaviour while preserving privacy, or in verification. It is thus clear that we need zero-knowledge proofs for lattice relations.

One of the fundamental hardness assumptions that lattice-based cryptography is built upon, is that it is difficult to find vector $\boldsymbol{s}$ of low norm satisfying $\boldsymbol{A}\boldsymbol{s} = \boldsymbol{u}$ over the ring $R_q$. Hence many lattice-based protocols will have to be able to prove knowledge of such an $\boldsymbol{s}$. But it turns out that proving that $\|\boldsymbol{s}\|$ is small is hard to do practically. One of the first attempts in constructing such a proof used Stern's protocol [26] in a lattice setting. Due to a soundness error of 2/3 these protocols had to be repeated so many times that the proofs reached several megabytes in size, and were hence very unpractical.

When Baum et al. introduced a more efficient lattice-based commitment scheme for vectors over $R_q$ in [4], papers like [28, 6] soon used it to build lattice-based zero-knowledge proofs with lower soundness error by exploiting the security properties of the commitment scheme. The proofs consisted of a proof of linear relations and a product proof of committed values to show that the secret has small coefficients, and were only several hundred kilobytes in size. These schemes were significantly improved in terms of proof size in [3, 10, 17], each by optimizing parts of the scheme. Attema et al. [3] introduced a more practical product proof of committed values, and Esgin et al. introduced a more efficient proof of linear relations. Lyubashevsky et at. [17] further optimized the schemes, for instance by using a new version of rejection sampling. At this point the main obstacle in reducing the proof sizes were the size of the commitments and opening proofs. Recently

Lyubashevsky, Nguyen and Plançon [16] further improved these schemes by using a new commitment scheme, making the proof sizes even smaller.

The goal of this paper is to study the state of the art lattice-based zero-knowledge proof of knowledge protocol that is given by Lyubashevsky Lyubashevsky, Nguyen and Plançon in [16]. We focus or attention on proofs that uses commitment schemes, and we look at how such schemes have developed over the past few years, in order to understand the complete picture of how the latest schemes are constructed. In Chapters 2 and 3 we present the cryptographic and mathematical theory that is necessary for this paper. We note that several subsections here are taken from, or very similar to, the corresponding chapters of the Specialization Project [27], as this paper considered some of the same topics. Further, we in Chapter 4 present the commitment schemes that we use in the paper, accompanied by a full security analysis and proofs of opening.

In Chapter 5 we explain three different zero-knowledge proofs of lattice relations, that all are important in understanding the latest one. We start with the scheme by Bootle et al. [6], that proves knowledge of a $\vec{s}$ with coefficients in $\{0, 1, 2\}$ satisfying $A\vec{s} = \vec{u}$ over $\mathbb{Z}_q$, by using commitments and $\mathsf{NTT}$ coefficients. We note that this is a more general statement, but by choosing $A$ to have a certain structure, this is equivalent to proving knowledge of $\boldsymbol{s}$ satisfying $\boldsymbol{As} = \boldsymbol{u}$ over $R_q$. Next, we investigate the scheme by Attema et al. [3], which is a more efficient scheme for proving multiplicative relations of committed values. Finally, we consider the scheme by Esgin et al. [10], which is a more efficient scheme for proving linear relations.

We then look at the main building blocks for the final scheme in Chapter 6. The goal is to understand how to construct schemes for proving many quadratic equations in $\boldsymbol{s}$ and that many polynomial evaluations in $\boldsymbol{s}$ have no constant coefficients. Then, we explain how such a scheme can be used to prove norm bounds on lattice elements, and give the final general state of the art protocol in Chapter 7.

# Chapter 2

# Cryptographic background

In this chapter we introduce the theory and definitions we need in order to build zero-knowledge protocols and commitment schemes. We start in Section 2.1 by introducing the basic notation and definitions that we need in order to define the cryptographic schemes and security properties that we will use in this paper. In Section 2.2 we define commitment schemes, before we in Section 2.3 define the notion of zero-knowledge proofs of knowledge. We note that Section 2.1 and parts of Section 2.3 are taken from [27].

## 2.1 Notation and definitions

If $X$ is a set, we use the notation $x \xleftarrow{\$} X$ to denote that $x$ is chosen uniformly at random from $X$. If $U$ is a distribution, the notation $x \xleftarrow{\$} U$ will denote that $x$ is randomly chosen according to $U$. We will also denote by $a \leftarrow b$ that $a$ gets assigned the value $b$. We write $i \in [n]$ to mean $i = 1, \ldots, n$.

In order to formally define the security of cryptographic schemes, we use the concepts of negligibility, statistical distance and indistinguishability. The term negligible is used to indicate that something is 'too small to matter'. In this paper we will use the term negligible when something very small, but we will also use the following asymptotic definition.

**Definition 1** ([8])**.** $\varepsilon(l)$ is *negligible in* $l$ if for any polynomial $p$, $\varepsilon(l) \leq 1/p(l)$ for all large enough $l$.

In order to say something about how similar two probability distributions are, we need to define statistical distance.

**Definition 2** ([8])**.** For two probability distributions $U$ and $V$, the *statistical distance* between them is

$$\Delta(X, Y) = \sum_y |U(y) - V(y)|,$$

where $U(y)$ and $V(y)$ denotes the probabilities $U$ and $V$ assigns to $y$, respectively.

Let $U$ be a probabilistic algorithm. We will now denote by $U_x$ the probability distribution of $U$'s output when run on input $x$. We can now give a formal definition of indistinguishability.

**Definition 3** ([8])**.** Given two probabilistic algorithms, or families of distributions, $U$ and $V$, we say that $U$ and $V$ are:

– *Perfectly indistinguishable* if $U_x$ and $V_x$ have the same probability distribution, $U_x = V_x$, for every $x$.

– *Statistically indistinguishable* if the statistical distance between $U_x$ and $V_x$ is negligible in the length of $x$ for every $x$.

– *Computationally indistinguishable* if for every algorithm $D$ that can run in $\text{poly}(n)$ time, the advantage in determining which distribution an output $x$ of length $n$ is from, is negligible in $n$. Namely that $|p_{U,D}(x) - p_{V,D}(x)| = \text{neg}(n)$, where $p_{U,D}(x)$ and $p_{V,D}(x)$ denotes the probability that $D$ guesses $U$ when $x$ is from $U$ and $V$ when $x$ is from $V$, respectively.

For two probabilistic algorithms $U$ and $V$, we will denote by $x \xleftarrow{\$} U^V(y)$ the output of $U$ on input $y$, when $U$ is given black-box access to $V$.

## 2.2 Commitment schemes

Commitment schemes are one of the fundamental cryptographic primitives, and can be used in a variety of cryptographic protocols. Commitment schemes allows for one to commit to a value while keeping it hidden, with the ability to reveal the committed value later. In this paper we will use such schemes as a part of zero-knowledge protocols. We now give a formal definition of a commitment scheme, which was first introduced by Blum [5].

**Definition 4** ([4])**.** A *commitment scheme* consists of algorithms $(\mathsf{KeyGen}, \mathsf{Commit}, \mathsf{Open})$ for key generation, commitment and opening, respectively, such that

– $\mathsf{KeyGen}(1^\lambda)$ is a probabilistic algorithm that on input a security parameter $1^\lambda$ outputs the public parameters $\mathsf{pp} \in \{0,1\}^{\mathsf{poly}(\lambda)}$, including a randomness space $\chi$

– $\mathsf{Commit}(\mathsf{pp}, m, r)$ is a probabilistic algorithm that on input the public parameters $\mathsf{pp}$ and a message $m$, draws $r \xleftarrow{\$} \chi$ and outputs a commitment $c \in \{0,1\}^{\mathsf{poly}(\lambda)}$ under the randomness

– $\mathsf{Open}(\mathsf{pp}, m, r, c)$ is a deterministic algorithm that on input the public parameters $\mathsf{pp}$, a message $m$, a commitment and opening $c, r \in \{0,1\}^{\mathsf{poly}(\lambda)}$, outputs a bit $b \in \{0,1\}$

We continue by defining the properties that we want our scheme to satisfy. It is important that the message one commits to is the only message that the commitment can open to. This ensures that a party cannot change the message after they have committed to it, and is called the binding property. It is also crucial that the commitment does not reveal anything about the message. This is called the hiding property. Also, we require that honestly generated commitments are accepted by $\mathsf{Open}$.

**Definition 5.** The commitment scheme $(\mathsf{KeyGen}, \mathsf{Commit}, \mathsf{Open})$ is *complete* if

$$\Pr\left[\mathsf{Open}(\mathsf{pp}, m, r, c) = 1 \;\middle|\; \mathsf{pp} \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda), c \xleftarrow{\$} \mathsf{Commit}(\mathsf{pp}, m, r)\right] = 1,$$

where the probability is taken over the randomness of $\mathsf{KeyGen}$ and $\mathsf{Commit}$.

**Definition 6.** The commitment scheme $(\mathsf{KeyGen}, \mathsf{Commit}, \mathsf{Open})$ is *hiding* if an algorithm $\mathcal{A}$ cannot distinguish which of two chosen messages a commitment is to. We say that the advantage of an

algorithm $\mathcal{A}$ in breaking the hiding property is

$$\left| \Pr \left[ b' = b \;\middle|\; \begin{array}{l} \mathsf{pp} \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda), (m_0, m_1) \xleftarrow{\$} \mathcal{A}(\mathsf{pp}), b \xleftarrow{\$} \{0,1\}, \\ c \xleftarrow{\$} \mathsf{Commit}(\mathsf{pp}, m_b, r), b' \xleftarrow{\$} \mathcal{A}(c) \end{array} \right] - \frac{1}{2} \right|,$$

where the probability is taken over the randomness of KeyGen and Commit.

**Definition 7.** The commitment scheme (KeyGen, Commit, Open) is *binding* if an algorithm $\mathcal{A}$ cannot find two valid openings to a commitment for different messages. We say that an algorithm $\mathcal{A}$ has probability

$$\Pr \left[ \begin{array}{l} m \neq m', \mathsf{Open}(\mathsf{pp}, m, r, c) = 1 \\ \mathsf{Open}(\mathsf{pp}, m', r', c) = 1, \end{array} \;\middle|\; \mathsf{pp} \xleftarrow{\$} \mathsf{KeyGen}(1^\lambda), (m, m', r, r', c) \xleftarrow{\$} \mathcal{A}(\mathsf{pp}) \right],$$

in breaking the binding property, where the probability is taken over the randomness of KeyGen.

## 2.3 Zero-knowledge protocols

Zero-knowledge protocols are important cryptographic primitives that allows a party to prove that a given statement is true, or that it knows some secret, without revealing anything other than that the statement is true. Such schemes are used in many cryptographic protocols, for instance to enforce honest behaviour, in digital voting and in verification.

In order to define zero-knowledge protocols, we start by introducing the concept of an interactive proof system, as it is defined in [8]. Suppose that you have the interactive algorithms $\mathcal{P}$ (prover) and $\mathcal{V}$ (verifier), and a language $L \subset \{0,1\}^*$. ($\mathcal{P}$,$\mathcal{V}$) are now given the input $x$. Through interactions between $\mathcal{P}$ and $\mathcal{V}$, the prover will claim that $x \in L$ and the verifier will try to determine whether this is true or not. The interaction between the prover and the verifier ends with the verifier outputting either accept or reject, indicating whether $(\mathcal{P}, \mathcal{V})$ accepts or rejects $x$. We will in this paper denote by $(\mathcal{P}, \mathcal{V})$ honest algorithms that follow the protocol, and by $(\mathcal{P}^*, \mathcal{V}^*)$ possibly dishonest algorithms.

**Definition 8.** $(\mathcal{P}, \mathcal{V})$ is an *interactive proof system* for the language $L$ if the following properties hold.

**Completeness:** for all $x \in L$, the probability that $(\mathcal{P}, \mathcal{V})$ accepts $x$ is non-negligible.

**Soundness:** for all $x \notin L$ and for any prover $\mathcal{P}^*$, the probability that $(\mathcal{P}^*, \mathcal{V})$ accepts $x$ is negligible in the length of $x$.

We can extend this definition to a situation where the prover tries to convince the verifier that it has 'knowledge' related to the publicly given $x$. For languages $L \subset \{0,1\}^*$ and $W \subset \{0,1\}^*$, we can define a relation $\mathcal{R} \subseteq L \times W$ to be such that if $(x, w) \in \mathcal{R}$ for $x \in L$ and $w \in W$, then $w$ is called a witness for $x$. Suppose now that the prover and the verifier are given a public $x$, and that the prover is given a secret $w$. $\mathcal{P}$ now tries to convince $\mathcal{V}$ that $w$ is a witness for $x$, and $\mathcal{V}$ either accepts or rejects this fact. We now give the definition of a proof of knowledge, following the idea in [20].

**Definition 9.** The interactive protocol $(\mathcal{P}, \mathcal{V})$ is a *proof of knowledge* for the relation $\mathcal{R}$ if it satisfies the following properties

**Completeness:** for all $(x, w) \in \mathcal{R}$, the probability that $(\mathcal{P}, \mathcal{V})$ accepts $x$ when $\mathcal{P}$ has input $w$, is non-negligible.

**Knowledge soundness:** there exists an algorithm $K$, called a knowledge extractor, such that for any prover $\mathcal{P}^*$ with non-negligible probability of making $\mathcal{V}$ accept, if $K$ can interact with $\mathcal{P}^*$, then $w \xleftarrow{\$} K^{\mathcal{P}^*}(x)$ is such that $(x, w) \in \mathcal{R}$ with non-negligible probability. If we for every $x \in L$ have that

$$\Pr[(x, w) \in \mathcal{R} \mid w \xleftarrow{\$} K^{\mathcal{P}^*}(x)] \geq \Pr[(\mathcal{P}^*, \mathcal{V}) \text{ accepts } x] - \varepsilon,$$

we say that the protocol has soundness error $\varepsilon$.

The knowledge soundness property ensures that the prover actually possesses the knowledge they claim to have, by ensuring that it is hard to create proofs of incorrect statements. We note that when a protocol has a large soundness error, the protocol can be repeated to reduce this error. This will however increse the total proof size.

### 2.3.1 Zero-knowledge

Goldwasser, Micali and Rackoff introduced the concept of zero-knowledge in [11]. Zero-knowledge is a property that guarantees that the prover does not reveal any additional information. We denote by $t \xleftarrow{\$} (\mathcal{P}(x, w), \mathcal{V}(x))$ the transcript produced by the protocol $(\mathcal{P}, \mathcal{V})$ on respective inputs $x$ and $w$.

**Definition 10.** An interactive proof of knowledge $(\mathcal{P}, \mathcal{V})$ is *zero-knowledge* if for any verifier $\mathcal{V}^*$ there exists a probabilistic simulator $\mathcal{S}$ such that for an adversary $\mathcal{A}$, the advantage in distinguishing the output of $(\mathcal{P}, \mathcal{V}^*)$ from the output of $\mathcal{S}$,

$$\left| \Pr\left[ b' = b \mid b \xleftarrow{\$} \{0,1\}, t_0 \xleftarrow{\$} (\mathcal{P}(x, w), \mathcal{V}^*(x)), t_1 \xleftarrow{\$} \mathcal{S}(x), b' \xleftarrow{\$} \mathcal{A}(t_b) \right] - \frac{1}{2} \right|,$$

is negligible [20].

One is usually interested in computational zero-knowledge, but one can also achieve perfect or statistical zero-knowledge, depending on the type of indistinguishability we have between $(\mathcal{P}, \mathcal{V}^*)$ and $\mathcal{S}$ [8]. This property can also be relaxed into the setting where we only consider an honest verifier $\mathcal{V}$, meaning that the verifier follows the protocol.

**Definition 11.** An interactive proof of knowledge $(\mathcal{P}, \mathcal{V})$ is *honest-verifier zero-knowledge* if there exists a probabilistic simulator $\mathcal{S}$ such that for an adversary $\mathcal{A}$, the advantage in distinguishing the output of $(\mathcal{P}, \mathcal{V})$ from the output of $\mathcal{S}$,

$$\left| \Pr\left[ b' = b \mid b \xleftarrow{\$} \{0,1\}, t_0 \xleftarrow{\$} (\mathcal{P}(x, w), \mathcal{V}(x)), t_1 \xleftarrow{\$} \mathcal{S}(x), b' \xleftarrow{\$} \mathcal{A}(t_b) \right] - \frac{1}{2} \right|,$$

is negligible.

### 2.3.2 Σ-protocols

A common type of zero-knowledge proofs of knowledge is the Σ-protocol. These are interactive protocols that work in the following three move manner [7].

1. $\mathcal{P}$ sends a message $a$ to $\mathcal{V}$
2. $\mathcal{V}$ sends a random string $e$ to $\mathcal{P}$
3. $\mathcal{P}$ sends a reply $z$, and $\mathcal{V}$ decides to accept or reject based on $x, a, e, z$

We require Σ-protocols to satisfy the same completeness and zero-knowledge properties as previous, but we can define a special version of soundness.

**Definition 12.** The above proof of knowledge is a Σ-*protocol* for the relation $\mathcal{R}$ if it satisfies the completeness and honest-verifier zero-knowledge properties as defined previously, and the following soundness property.

**Special soundness:** There exists an algorithm $E$, called a special extractor, such that for any $x$ with accepting transcripts $(a, e, z)$ and $(a, e', z')$ with $e \neq e'$, then $w \xleftarrow{\$} E(x, (a, e, z), (a, e', z'))$ is such that $(x, w) \in \mathcal{R}$ with overwhelming probability.

Many of the protocols presented in this paper will have this, or a very similar, structure. For part two of the protocol, we will often call the random string $e$ a challenge. We will also specify beforehand a challenge space, a space for which the verifier can draw the challenge from. This space have to be defined in such a manner that the protocol will satisfy the soundness property. We note that we in this paper will not specify the relation the protocols are for, but rather state what the protocols prove.

### 2.3.3 Commit-and-prove simulatability

When commitments are used as a part of zero-knowledge protocols, we have to be able to simulate the commitments in order to achieve zero-knowledge. This is usually not a problem, since the hiding property of commitment schemes makes sure that commitments look uniformly random. However, in some protocols we wish to create intermediate commitments under the same randomness, and this cannot be simulated in a zero-knoweldge manner.

We therefore introduce a new form of simulatability that was introduced by Lyubashevsky et al. in [17], called commit-and-prove simulatability. This instead makes sure that the view of the commitment and the protocol output is computationally indistinguishable for all committed messages. In practice this means that one can no longer reuse the commitment, but this is not a problem for applications, since commitments never needs to be reused in practice.

**Definition 13** ([17])**.** An interactive proof of knowledge $(\mathcal{P}, \mathcal{V})$ is *commit-and-prove simulatable* for the relation $\mathcal{R}$ if there exists simulators $\mathsf{SimCom}$ and $\mathsf{SimProve}$ such that for all adversaries $\mathcal{A}$:

$$\Pr\left[\begin{array}{r|l}(x, m_1, \ldots, m_n) \xleftarrow{\$} \mathcal{A}, r_1, \ldots, r_n \xleftarrow{\$} \chi,\ \forall i, c_i = \mathsf{Commit}(m_i, r_i), & (x, (m_1, \ldots, m_n)) \in \mathcal{R} \\ t \xleftarrow{\$} \Big(\mathcal{P}\big(x, (m_1, r_1), \ldots, (m_n, r_n)\big), \mathcal{V}(x)\Big) & \wedge\ \mathcal{A}(c_1, \ldots, c_n, t) = 1\end{array}\right]$$

$$\approx \Pr\left[\begin{array}{r|l}(x, m_1, \ldots, m_n) \xleftarrow{\$} \mathcal{A}, c_1, \ldots, c_n \xleftarrow{\$} \mathsf{SimCom}(x), & (x, (m_1, \ldots, m_n)) \in \mathcal{R} \\ t \xleftarrow{\$} \mathsf{SimProve}(x, c_1, \ldots, c_n) & \wedge\ \mathcal{A}(c_1, \ldots, c_n, t) = 1\end{array}\right]$$

where $\chi$ is a probability distribution on the randomness space.

### 2.3.4 Proving knowledge soundness

There are many techniques for proving that a protocol is knowledge sound. In this paper we will use one such technique for extracting transcripts, a collision game introduced by Attema et al. in [2]. The strategy starts by letting $H \in \{0,1\}^{R \times M}$ be a binary matrix where the $R$ rows correspond to the prover's randomness and the $M$ columns corresponds to the verifier's randomnes, when the challenge space is of size $M$. We can denote the entry corresponding to randomness $r$ and challenge $c \in \mathcal{C}$ by $H(r,c)$, and this entry is equal to 1 if and only if the corresponding protocol transcript is accepting. One can now define the following extractor $\mathcal{E}$.

1. First, $\mathcal{E}$ samples $(r,i) \xleftarrow{\$} [R] \times [M]$. It checks is $H(r,i) = 1$, and aborts if not.
2. If $H(r,i) = 1$, then it samples $i^* \xleftarrow{\$} [M]$ without replacement until it obtains distinct $i_1^*, \ldots, i_{k-1}^*$ such that $H(r, i_\ell^*) = 1$ for $\ell = 1, \ldots, k-1$.

We now give a result from [2] that states the expected run time and success probability of $\mathcal{E}$.

**Lemma 1.** Let $H \in \{0,1\}^{R \times M}$ and define $\varepsilon$ to be the fraction of 1-entries in $H$. Then, the expected number of $H$-entries queried in the collision game defined above is at most $k$ and the success probability of the collision game is at least $\varepsilon - \frac{k-1}{M}$.

We can thus define such an extractor $\mathcal{E}$ in our soundness proofs to obtain a given number of valid transcripts, with a given success probability.

Lastly, we will explain the heavy rows argument, which can be used in soundness proofs to determine how likely it is and how long it will take to obtain accepting transcripts [7]. We say that a row of $H$ is heavy if it has a fraction of at least $\varepsilon/2$ 1's. By this definition, more than half of the 1's must lie in a heavy row. If we now let $H'$ be the sub-matrix of all rows that are not heavy, we can denote by $h'$ and $h$ the number of entries in $H'$ and $H$, respectively. The number of 1's in $H$ is by assumption $h\varepsilon$, and thus the number of 1's in $H'$ must be less that $h'\varepsilon/2$. We now denote by $g$ the number of 1's in heavy rows, and see that it must satisfy

$$g > h\varepsilon - h'\varepsilon/2 \geq h\varepsilon - h\varepsilon/2 = h\varepsilon/2.$$

If we assume that $\varepsilon$ is such that this implies that a heavy row has at least two 1's, we can find two 1's in the same row as explained in the following.

We start by first randomly searching $H$ for a 1-entry. The expected number of tries to get a 1 is $1/\varepsilon$. There is now a probability of at least $1/2$ that this 1 lies in a heavy row. If this is true, and we continue searching in this row, we will find another 1-entry in one try with probability at least $\varepsilon/2 - 1/|\mathcal{C}|$. Hence the expected number of tries it takes to find another 1-entry is

$$\frac{1}{\varepsilon/2 - 1/|\mathcal{C}|}.$$

We can thus use this heavy rows argument in soundness proofs to say how likely it is to find accepting transcripts by querying a prover, and how much time it is expected to take.

# Chapter 3

# Mathematical background

In this chapter we present all the mathematical theory that we will use in this paper. We start by introducing lattices and the hard lattice problems that are the foundation for lattice cryptography. Then we introduce the rings that we will actually work over in the rest of this paper and its hard problems, which looks very similar to the lattice problems. We then introduce rejection sampling in Section 3.3, which is an important part of all the protocols in this paper. In Section 3.4 we explain approximate range proofs, which will be used in proving norm bounds.

Section 3.5 introduces Galois automorphisms, and Section 3.6 explains the number theoretic transform, which will both be used to construct more efficient zero-knowledge protocols. Finally, in Section 3.7 we give the results that is used to construct the challenge spaces we use in this paper. We note that Section 3.1 on lattices and parts of the Sections 3.2 and 3.3 are taken directly from [27].

## 3.1 Lattice algebra

There are many ways to define a lattice, but we follow the definitions given in [12], and restrict our analysis to $\mathbb{Z}^n$.

**Definition 14.** A *lattice* $\mathcal{L}$ is the set of all linear combinations with integer coefficients of a given set of linearly independent points in $\mathbb{Z}^n$. Any such $\mathcal{L}$ is spanned by a basis $\boldsymbol{B} = \{\boldsymbol{b}_1, ..., \boldsymbol{b}_d\}$, such that $\mathcal{L} = \left\{ \sum_{i=1}^{d} x_i \boldsymbol{b}_i \mid x_i \in \mathbb{Z} \right\}$. We then say that $\mathcal{L}$ has *rank* $d$, and that it is *full rank* if $d = n$.

For a lattice $\mathcal{L} \subseteq \mathbb{Z}^n$ we define the inner product $\langle \cdot, \cdot \rangle$ and norms $\| \cdot \|_p$ as the usual vector inner product and norms on $\mathbb{Z}^n$. For lattice vectors we mostly consider the $\ell_2$ norm, and denote this by $\| \cdot \|$ for the rest of this section.

Lattices $\mathcal{L} \subseteq \mathbb{Z}^n$ has the property that for every point in the lattice, there exist an open ball around it in which there are no other points in the lattice. This is called the discreteness property, and it implies that any lattices of rank at least 1 has a non-zero lattice point that is closest to the origin [12]. The norm of this point is the smallest possible distance between two points in the lattice.

**Definition 15.** For a lattice $\mathcal{L} \subseteq \mathbb{Z}^n$ of rank $d$, the *first successive minimum* of the lattice is

$$\lambda_1(\mathcal{L}) = \min\{\|\boldsymbol{x}\| \mid \boldsymbol{x} \in \mathcal{L}, \boldsymbol{x} \neq 0\}.$$

We also define the *i-th successive minimum*, for $i = 2, ..., d$ to be

$$\lambda_i(\mathcal{L}) = \min\{\max\{\|\boldsymbol{x}_1\|, ..., \|\boldsymbol{x}_i\|\} \mid \boldsymbol{x}_1, ..., \boldsymbol{x}_i \in \mathcal{L} \text{ are linearly independent } \}.$$

From now on we only consider full rank lattices $\mathcal{L}$. We recall the first successive minimum property, and notice that finding a non-zero lattice vector with norm equal to, or sufficiently close to, the first successive minimum is a natural problem.

**Definition 16.** Given a basis of a lattice $\mathcal{L}$, the *shortest vector problem* ($\mathsf{SVP}$) is to find $\boldsymbol{y} \in \mathcal{L}$ such that $\|\boldsymbol{y}\| = \lambda_1(\mathcal{L})$. If we are also given an approximation factor $\gamma \geq 1$, the *approximate shortest vector problem* ($\mathsf{SVP}_\gamma$) is to find $\boldsymbol{y} \in \mathcal{L}$ such that $0 < \|\boldsymbol{y}\| \leq \gamma\lambda_1(\mathcal{L})$. Clearly $\mathsf{SVP} = \mathsf{SVP}_1$.

The $\mathsf{SVP}_\gamma$ problem is known to be NP-hard for $\gamma \approx \sqrt{d}$ [12]. We now extend the $\mathsf{SVP}_\gamma$ problem to finding short sets of lattice vectors.

**Definition 17.** Given a basis of a lattice $\mathcal{L}$ and an approximation factor $\gamma > 1$, the *shortest independent vector problem* ($\mathsf{SIVP}_\gamma$) is to find a linearly independent set $\{\boldsymbol{y}_1, ..., \boldsymbol{y}_d\}$ such that $\max_i \|\boldsymbol{y}_i\| \leq \gamma\lambda_d(\mathcal{L})$.

The $\mathsf{SIVP}_\gamma$ problem is NP-hard for $\gamma = d^{1/\log\log d}$ [12]. We now introduce some hard problems that were used to construct the first lattice-based schemes. The following problem was first introduced by Ajtai [1].

**Definition 18.** Given an integer $q$, $\boldsymbol{A} \in \mathbb{Z}_q^{m\times n}$, and a $\beta < q$, the *small integer solutions problem* ($\mathsf{SIS}_{q,n,\beta}$) is to find $\boldsymbol{y} \in \mathbb{Z}^n$ such that $\boldsymbol{A}\boldsymbol{y} \equiv 0 \pmod q$ and $\|\boldsymbol{y}\| \leq \nu$.

The worst-case $\mathsf{SIVP}_\gamma$ problem can be reduced to the $\mathsf{SIS}$ problem, and hence the $\mathsf{SIS}$ problem is at least as hard as the worst-case $\mathsf{SIVP}_\gamma$ problem [21].

In order to define the learning with errors problem, which was first introduced by Regev [23], we must introduce some notation. For an integer $q$, $\boldsymbol{s} \in \mathbb{Z}_q^n$ and a probability distribution $\psi$ on $\mathbb{Z}_q$, we can define a new probability distribution $A_{\boldsymbol{s},\psi}$ on $\mathbb{Z}_q^n \times \mathbb{Z}_q$. This distribution is sampled by taking $\boldsymbol{a} \in \mathbb{Z}_q^n$ uniformly at random, taking $e$ according to $\psi$, and then returning $(\boldsymbol{a}, \langle\boldsymbol{a}, \boldsymbol{s}\rangle + e) \pmod q$.

**Definition 19.** Given $n$, $q$, a probability distribution $\psi$ on $\mathbb{Z}_q$, and any number of independent samples from $A_{\boldsymbol{s},\psi}$, the *learning with errors problem* ($\mathsf{LWE}_{q,\chi}$) is to find $\boldsymbol{s}$.

The worst-case $\mathsf{SIVP}_\gamma$ problem can be reduced to the $\mathsf{LWE}$ problem, which means that $\mathsf{LWE}$ problem is at least as hard as the worst-case $\mathsf{SIVP}_\gamma$ problem [23]. We are more interested in the decision version of the $\mathsf{LWE}$ problem, which is to distinguish a sample from $A_{\boldsymbol{s},\psi}$ from a truly uniform $(\boldsymbol{a}, \boldsymbol{b}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$. This problem is as hard as solving the $\mathsf{LWE}$ problem for appropriate choices of $q$ [24].

There are many other hard lattice problems with respective reductions to the $\mathsf{SIS}$ and $\mathsf{LWE}$ problems, but for this paper the ones presented in this section suffices to illustrate that these problems are believed to be hard.

## 3.2 The rings $R$ and $R_q$

Many of the early lattice-based cryptographic schemes were based on the $\mathsf{SIS}$ and $\mathsf{LWE}$ problems. One disadvantage of cryptographic schemes that are based on these problems, is that their key sizes

are very large. Lyubashevsky, Peikert and Regev introduced a new environment for which more efficient variants of the LWE and SIS problems can be defined [18]. In this section we will explain this environment, and give the hard problems corresponding to SIS and LWE, namely the MSIS and MLWE problems.

Let $f(X) = X^N + 1 \in \mathbb{Z}[X]$, for an $N$ that is a power of 2. Let $R = \mathbb{Z}[X]/\langle f(X)\rangle$ and $R_q = \mathbb{Z}_q[X]/\langle f(X)\rangle$. The elements of these rings are polynomials of degree at most $N-1$, and in $R_q$ all coefficients are restricted to $[-(q-1)/2, (q-1)/2]$. In these rings addition is defined to be component-wise in the coefficients, and multiplication is defined as regular polynomial multiplication modulo $f(X)$. We call $R$ and $R_q$ module lattices.

Since elements of these rings can be written as $a = \sum_{i=0}^{N-1} a_i X^i$, for $a_i \in \mathbb{Z}$ or $a_i \in \mathbb{Z}_q$, respectively, we can define the $\ell_1, \ell_2$ and $\ell_\infty$ lengths as

$$\|a\|_1 = \sum_{i=0}^{N-1} |a_i|, \ \|a\|_2 = \sqrt{\sum_{i=0}^{N-1} a_i^2}, \text{ and } \|a\|_\infty = \max_{i \in 0,\ldots,N-1} |a_i|.$$

For a vector of ring elements $\boldsymbol{a} = (a_1, \ldots, a_k) \in R^k$, we define the $\ell_1, \ell_2$ and $\ell_\infty$ lengths as

$$\|\boldsymbol{a}\|_1 = \sum_{i=1}^{k} \|a_i\|_1, \ \|\boldsymbol{a}\|_2 = \sqrt{\sum_{i=1}^{k} \|a_i\|_2^2}, \text{ and } \|\boldsymbol{a}\|_\infty = \max_{i \in 1,\ldots,k} \|a_i\|_\infty.$$

We write $\|\boldsymbol{a}\| := \|\boldsymbol{a}\|_2$ throughout the rest of this paper. One useful relation between these norms is that $\|\boldsymbol{a}\| \leq \sqrt{kN}\|\boldsymbol{a}\|_\infty$. For the hardness and correctness properties of cryptographic schemes based on problems in this environment, it is often required to use elements of small norms. We therefore introduce the sets $S_i$ and $S_i^k$ of elements of $R$ and $R^k$ respectively, that has $\ell_\infty$ length at most $i$. We also use the notation $R_q^\times$ to denote the set of all elements of $R_q$ that are invertible.

In this paper we will denote by $\vec{a} = (a_0, \ldots, a_{N-1}) \in \mathbb{Z}_q^N$ the coefficient vector of a ring element $a \in R_q$, such that $a_i$ is the coefficient corresponding to $X^i$ of $a$. We use the notation $\tilde{a} := a_0 \in \mathbb{Z}_q$ for the constant coefficient of $a$. We notice that the standard vector norms of a coefficient vector $\vec{a}$ will be the same as the corresponding norm of the ring element $a$. Lastly, we define the inner product of two vectors of ring elements to be the inner product of their corresponding coefficient vectors, $\langle \boldsymbol{a}, \boldsymbol{b}\rangle := \langle \vec{a}, \vec{b}\rangle$.

Throughout this paper we will use the following important result from [19] on how the polynomial $X^N + 1$ factors modulo $q$, and that shows how to choose $q$ such that all elements with small norms are invertible in $R_q$.

**Lemma 2.** Let $N \geq k > 1$ be powers of 2 and let $q = 2k + 1 \pmod{4k}$ be a prime. Then the polynomial $X^N + 1$ factors as

$$X^N + 1 \equiv \prod_{j=1}^{k} (X^{N/k} - \zeta_j) \pmod{q}$$

where $\zeta_j \in \mathbb{Z}_q$ are the $2k$-th roots of unity in $\mathbb{Z}_q$, and $X^{N/k} - \zeta_j$ are irreducible in the ring $\mathbb{Z}_q[X]$.

Furthermore, any $y$ in $\mathbb{Z}_q[X]/(X^N+1)$ that satisfies either

$$0 < \|y\|_\infty < \frac{1}{\sqrt{k}} q^{1/k}$$
$$0 < \|y\| < q^{1/k}$$

has an inverse in $\mathbb{Z}_q[X]/(X^N+1)$.

We are now ready to define the problems that are the foundation for all the schemes we will consider in this paper. It is clear that these problems are the equivalent problems to SIS and decision LWE, only over the module lattice $R_q$ instead.

**Definition 20.** Given $\boldsymbol{A} \xleftarrow{\$} R_q^{n\times m}$, the *module small integer solutions problem* ($\mathsf{MSIS}_{n,m,B}$) is to find $\boldsymbol{z} \in R_q^m$ such that $\boldsymbol{Az} = \boldsymbol{0}$ over $R_q$ and $0 < \|\boldsymbol{z}\| \leq B$. For an algorithm $\mathcal{A}$ we say that he advantage in solving the problem is

$$\Pr[0 < \|\boldsymbol{z}\| \leq B \wedge \boldsymbol{Az} = \boldsymbol{0} \mid \boldsymbol{A} \xleftarrow{\$} R_q^{n\times m}, \boldsymbol{z} \xleftarrow{\$} \mathcal{A}(\boldsymbol{A})]$$

**Definition 21.** The *module learning with errors problem* with error distribution $\chi$ over $R$ ($\mathsf{MLWE}_{m,n,\chi}$) is to distinguish $(\boldsymbol{A}, \boldsymbol{As} \mod q)$ for $\boldsymbol{A} \xleftarrow{\$} R_q^{n\times m}$ and secret $\boldsymbol{s} \xleftarrow{\$} \chi^m$ from truly random $(\boldsymbol{A}, \boldsymbol{b}) \xleftarrow{\$} R_q^{n\times m} \times R_q^n$. For an algorithm $\mathcal{A}$ we say that he advantage in solving the problem is

$$\Big| \Pr\left[b = 1 \mid \boldsymbol{A} \xleftarrow{\$} R_q^{n\times m}, \boldsymbol{s} \xleftarrow{\$} \chi^m, b \xleftarrow{\$} \mathcal{A}(\boldsymbol{A}, \boldsymbol{As} \mod q)\right]$$
$$- \Pr\left[b = 1 \mid \boldsymbol{A} \xleftarrow{\$} R_q^{n\times m}, \boldsymbol{b} \xleftarrow{\$} R_q^n, b \xleftarrow{\$} \mathcal{A}(\boldsymbol{A}, \boldsymbol{b})\right] \Big|$$

Langlois and Stehlé gives in [13] reductions from worst-case $\mathsf{SIVP}_\gamma$ problems restricted to module lattices to both the MSIS and MLWE problem. This illustrates the hardness of these problems.

We also give an extended version of the MLWE problem, that we need for the security analysis of protocols that uses the subset rejection sampling method by Lyubashevsky et al. [17], which is defined in the next section. This rejection sampling algorithm reveals the sign of $\langle \boldsymbol{z}, c\boldsymbol{s}\rangle$ when applied to $\boldsymbol{z} = c\boldsymbol{s} + \boldsymbol{y}$. Hence we need a problem that is still hard with this information given, for simulation of such schemes.

**Definition 22** ([17])**.** The *extended module learning with errors problem* ($\mathsf{Extended\text{-}MLWE}_{m,\lambda,\chi,\mathcal{C},\mathfrak{s}}$) with parameters $m, \lambda > 0$, probability distribution $\chi$ over $R_q$, challenge space $\mathcal{C} \subseteq R_q$ and the standard deviation $\mathfrak{s}$, asks the adversary $\mathcal{A}$ to distinguish between the following two cases:

1. $\Big(\boldsymbol{B}, \boldsymbol{Bs}, c, \boldsymbol{z}, \mathsf{sign}(\langle \boldsymbol{z}, c\boldsymbol{s}\rangle)\Big)$ for $\boldsymbol{B} \xleftarrow{\$} R_q^{m\times(m+\lambda)}$, a secret vector $\boldsymbol{s} \xleftarrow{\$} \chi^{m+\lambda}$, $\boldsymbol{z} \xleftarrow{\$} D_{R^{m+\lambda},\mathfrak{s}}$ and $c \xleftarrow{\$} \mathcal{C}$
2. $\Big(\boldsymbol{B}, \boldsymbol{u}, c, \boldsymbol{z}, \mathsf{sign}(\langle \boldsymbol{z}, c\boldsymbol{s}\rangle)\Big)$ for $\boldsymbol{B} \xleftarrow{\$} R_q^{m\times(m+\lambda)}$, $\boldsymbol{u} \xleftarrow{\$} R_q^m$, $\boldsymbol{z} \xleftarrow{\$} D_{R^{m+\lambda},\mathfrak{s}}$ and $c \xleftarrow{\$} \mathcal{C}$,

where $\mathsf{sign}(a) = 1$ if $a \geq 0$ and 0 otherwise. For an algorithm $\mathcal{A}$ we say that he advantage in solving the problem is

$$\Big| \Pr\Big[b = 1 \mid \boldsymbol{B} \xleftarrow{\$} R_q^{m\times(m+\lambda)}, \boldsymbol{r} \xleftarrow{\$} \chi^{m+\lambda}, \boldsymbol{z} \xleftarrow{\$} D_{R^{m+\lambda},\mathfrak{s}}, c \xleftarrow{\$} \mathcal{C}, b \xleftarrow{\$} \mathcal{A}(\boldsymbol{B}, \boldsymbol{Br}, \boldsymbol{z}, c, s)\Big]$$

$$- \Pr\Big[b = 1 \mid \boldsymbol{B} \xleftarrow{\$} R_q^{m\times(m+\lambda)}, \boldsymbol{u} \xleftarrow{\$} R_q^{m}, \boldsymbol{z} \xleftarrow{\$} D_{R^{m+\lambda},\mathfrak{s}}, c \xleftarrow{\$} \mathcal{C}, b \xleftarrow{\$} \mathcal{A}(\boldsymbol{B}, \boldsymbol{u}, \boldsymbol{z}, c, s)\Big]\Big|,$$

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

It is shown in [17] that the hardness of this problem can be reduced to the $\mathsf{LWE}$ problem.

## 3.3 The Gaussian distribution and rejection sampling

In zero-knowledge protocols for lattice relations, one often wants to output a linear combination of a secret $\boldsymbol{s}$, say $\boldsymbol{z} = c\boldsymbol{s} + \boldsymbol{y}$. But then it is important to make $\boldsymbol{z}$ independent of $\boldsymbol{s}$, to make sure that no information is revealed. In order to achieve this, we use rejection sampling, a method that was first introduced by Lyubashevsky in [14, 15]. We start by defining the Gaussian distribution that is used for rejection sampling.

**Definition 23.** The *discrete Gaussian distribution* over the lattice $R^k$ centered at some $\boldsymbol{v} \in R^k$ with standard deviation $\mathfrak{s}$ is defined as

$$D_{R^k,\boldsymbol{v},\mathfrak{s}}(\boldsymbol{z}) = e^{\frac{-\|\boldsymbol{z}-\boldsymbol{v}\|^2}{2\mathfrak{s}^2}} \Big/ \sum_{\boldsymbol{w}\in R^k} e^{\frac{-\|\boldsymbol{w}\|^2}{2\mathfrak{s}^2}}$$

For the rest of this paper we will use the notation $\boldsymbol{y} \xleftarrow{\$} D_{R^k,\mathfrak{s}}$ to indicate that $\boldsymbol{y}$ was chosen according to $D_{R^k,0,\mathfrak{s}}$. We now introduce an important tail-bound lemma for this distribution, Lemma 4.4 in [15].

**Lemma 3.** For any $\delta > 1$ we have that

1. $\Pr[|z| > \delta\mathfrak{s} \mid z \xleftarrow{\$} D_{R,\mathfrak{s}}] \leq 2\exp(\frac{-\delta^2}{2})$
2. $\Pr[\|\boldsymbol{z}\| > \delta\mathfrak{s}\sqrt{kN} \mid \boldsymbol{z} \xleftarrow{\$} D_{R^k,\mathfrak{s}}] < \delta^{kN}\exp\left(\frac{kN}{2}(1-\delta^2)\right)$

Rejection sampling is used in protocols where the prover draws a masking vector $\boldsymbol{y} \xleftarrow{\$} D_{R^k,\mathfrak{s}}$, and upon receiving a challenge $c$ from the verifier, want so send $\boldsymbol{z} = c\boldsymbol{s} + \boldsymbol{y}$ to the verifier. By performing the rejection sampling algorithm on $\boldsymbol{z}$, the dependency of $\boldsymbol{z}$ on $\boldsymbol{s}$ is removed, or in other words $\boldsymbol{z}$ will be statistically close to $D_{R^k,\mathfrak{s}}$, so that the prover can send $\boldsymbol{z}$ without revealing anything about $\boldsymbol{s}$.

The standard Gaussian rejection sampling procedure from [15], $\mathsf{Rej}_1$, is shown in Algorithm 1. There have been proposed several versions of rejection sampling after this algorithm was introduced. Such new versions introduce some modifications to the original algorithm, with the goal of reducing the standard deviation used in the protocols. This way we are able to achieve lower bounds on the elements drawn from $D_{R^k,\mathfrak{s}}$.

One such version of rejection sampling was proposed by Lyubashevsky et al. in [17], and it modifies

**Algorithm 1** $\mathsf{Rej}_1(\boldsymbol{z}, c\boldsymbol{s}, \mathfrak{s})$

1: $u \xleftarrow{\$} [0, 1)$
2: **if** $u > \frac{1}{M}\exp(\frac{-2\langle \boldsymbol{z}, \boldsymbol{s}\rangle + \|\boldsymbol{s}\|^2}{2\mathfrak{s}^2})$ **then**
3: return 1 (*reject*)
4: **else**
5: return 0 (*accept*)
6: **end if**

$\mathsf{Rej}_1$ by forcing $\boldsymbol{z}$ to satisfy $\langle \boldsymbol{z}, \boldsymbol{s}\rangle \geq 0$. This has the effect that the standard deviation can be reduced by more than a factor of 10, while keeping the expected number of rejections the same. We notice that this rejection sampling algorithm will reveal one bit of the secret, but in cases where this does not matter, one can use this algorithm. This rejection sampling variant, $\mathsf{Rej}_2$, is shown in Algorithm 2. We call this version subset rejection sampling.

**Algorithm 2** $\mathsf{Rej}_2(\boldsymbol{z}, \boldsymbol{s}, \mathfrak{s})$

1: **if** $\langle \boldsymbol{z}, \boldsymbol{s}\rangle < 0$ **then**
2: return 1 (*reject*)
3: **end if**
4: $u \xleftarrow{\$} [0, 1)$
5: **if** $u > \frac{1}{M}\exp(\frac{-2\langle \boldsymbol{z}, \boldsymbol{s}\rangle + \|\boldsymbol{s}\|^2}{2\mathfrak{s}^2})$ **then**
6: return 1 (*reject*)
7: **else**
8: return 0 (*accept*)
9: **end if**

The final rejection sampling variant we will use is the bimodal rejection sampling, which was first introduced by Ducas et al. in [9]. The main difference of this variant, is that we sample a sign $\beta \xleftarrow{\$} \{-1, 1\}$ and compute $\boldsymbol{z}$ as $\boldsymbol{z} = \boldsymbol{y} + \beta c\boldsymbol{s}$ instead. This significantly reduces the standard deviation. This rejection sampling variant, $\mathsf{Rej}_0$, is shown in Algorithm 3.

**Algorithm 3** $\mathsf{Rej}_0(\boldsymbol{z}, \boldsymbol{s}, \mathfrak{s})$

1: $u \xleftarrow{\$} [0, 1)$
2: **if** $u > \frac{1}{M\exp\left(\frac{-\|\boldsymbol{s}\|^2}{2\mathfrak{s}^2}\right)\cosh\left(\frac{\langle \boldsymbol{z}, \boldsymbol{s}\rangle}{\mathfrak{s}^2}\right)}$ **then**
3: return 1 (*reject*)
4: **else**
5: return 0 (*accept*)
6: **end if**

These algorithms have the properties stated in the following Lemma, which gives the repetition rates $M$ and rejection probabilities of the protocols, and states that the output $\boldsymbol{z}$ looks independent of $\boldsymbol{s}$.

**Lemma 4.** Let $V \subseteq R^k$ be such that all elements have $\ell_2$ norm less than $T$, $\mathfrak{s} \in \mathbb{R}$ such that $\mathfrak{s} = \gamma T$ and $h : V \to [0, 1]$ be a probability distribution. Then, the following statements hold.

1. Let $M = \exp(14/\gamma + 1/(2\gamma^2))$. Now, sample $\boldsymbol{s} \xleftarrow{\$} h$ and $\boldsymbol{y} \xleftarrow{\$} D_{R^k,\mathfrak{s}}$, set $\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{s}$ and run $b \leftarrow \mathsf{Rej}_1(\boldsymbol{z}, \boldsymbol{s}, \mathfrak{s})$ as defined in Algorithm 1. Then, the probability that $b = 0$ is at least $(1-2^{-128})/M$ and the distribution of $(\boldsymbol{s}, \boldsymbol{z})$ conditioned on $b = 0$, is within statistical distance of $2^{-128}$ of the product distribution $h \times D_{R^k,\mathfrak{s}}$.

2. Let $M = \exp(1/(2\gamma^2))$. Now, sample $\boldsymbol{s} \xleftarrow{\$} h$ and $\boldsymbol{y} \xleftarrow{\$} D_{R^k,\mathfrak{s}}$, set $\boldsymbol{z} = \boldsymbol{y} + \boldsymbol{s}$ and run $b \leftarrow \mathsf{Rej}_2(\boldsymbol{z}, \boldsymbol{s}, \mathfrak{s})$ as defined in Algorithm 2. Then, the probability that $b = 0$ is at least $1/(2M)$ and the distribution of $(\boldsymbol{s}, \boldsymbol{z})$ conditioned on $b = 0$, is identical to the distribution of $\mathcal{F}$ where $\mathcal{F}$ is defined as follows: sample $\boldsymbol{s} \xleftarrow{\$} h$, $\boldsymbol{z} \xleftarrow{\$} D_{R^{kN},\mathfrak{s}}$ conditioned on $\langle \boldsymbol{s}, \boldsymbol{z} \rangle \geq 0$ and output $(\boldsymbol{s}, \boldsymbol{z})$.

3. Let $M = \exp(1/(2\gamma^2))$. Now, sample $\boldsymbol{s} \xleftarrow{\$} h$, $\beta \xleftarrow{\$} \{-1, 1\}$ and $\boldsymbol{y} \xleftarrow{\$} D_{R^k,\mathfrak{s}}$, set $\boldsymbol{z} = \boldsymbol{y} + \beta\boldsymbol{s}$ and run $b \leftarrow \mathsf{Rej}_0(\boldsymbol{z}, \boldsymbol{s}, \mathfrak{s})$ as defined in Algorithm 3. Then, the probability that $b = 0$ is at least $1/M$ and the distribution of $(\boldsymbol{s}, \boldsymbol{z})$ conditioned on $b = 0$, is identical to the product distribution $h \times D_{R^k,\mathfrak{s}}$.

## 3.4 The binomial distribution and approximate range proofs

One tool that we will use for both proving norm bounds, and that there are no overflow modulo $q$ in certain equations, are so called approximate range proofs. These say that if you draw a matrix $R$ from a binomial distribution, then the projection $R\vec{w} + \vec{y}$ has approximately the same norm as $\vec{w}$, for some vector $\vec{w}$ and masking vector $\vec{y}$ over $\mathbb{Z}_q$. This applies both for the $\ell_2$ and the $\ell_\infty$ norm. We start by defining the binomial distribution that we will use.

**Definition 24** ([16])**.** The binomial distribution with a positive integer parameter $\kappa$, written as $\mathsf{Bin}_\kappa$, is the distribution $\sum_{i=1}^{\kappa}(a_i - b_i)$ where $a_i, b_i \xleftarrow{\$} \{0, 1\}$. The variance of the distribution is $\kappa/2$ and it holds that $\mathsf{Bin}_{\kappa_1} \pm \mathsf{Bin}_{\kappa_2} = \mathsf{Bin}_{\kappa_1+\kappa_2}$.

We now introduce a result that we will use in order to prove approximate shortness in the $\ell_\infty$ norm. The idea is that if we choose a random matrix $R \xleftarrow{\$} \mathsf{Bin}_1^{k\times m}$ and $\vec{y} \in \mathbb{Z}_q^k$, we can prove approximate shortness of $\vec{w} \in \mathbb{Z}^m$ by proving that $R\vec{w}+\vec{y}$ is short in the $\ell_\infty$ length. This is sufficient by the following Lemma.

**Lemma 5** ([17])**.** Let $\vec{w} \in \mathbb{Z}_q^m$ and $\vec{y} \in \mathbb{Z}_q^k$. Then

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

We can use the same idea to prove approximate shortness in the $\ell_2$ norm, by utilizing the following Lemma instead.

**Lemma 6** ([17])**.** Fix $m, P \in \mathbb{N}$ and a bound $b \leq P/41m$, and let $\vec{w} \in [\pm P/2]^m$ with $\|\vec{w}\| \geq b$, and let $\vec{y}$ be an arbitrary vector in $[\pm P/2]^m$. Then

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

When we use relations of the form $R\vec{w} + \vec{y}$ in zero-knowledge protocols, the upper bound on $\|R\vec{w}\|$ will determine the standard deviation we can use for rejection sampling. Hence we need a way to bound the $\ell_2$ norm of $R\vec{w}$. For this we will use the following result.

**Lemma 7.** For any $\vec{w} \in \mathbb{Z}^m$ we have

$$\Pr_{R \xleftarrow{\$} \mathsf{Bin}_\kappa^{256 \times m}} [\|R\vec{w}\|^2 > \|\vec{w}\|^2 \cdot 337 \cdot \kappa] \leq 2^{-128}$$

## 3.5 Galois automorphism

One of the tools that have been used to create more efficient lattice-based zero-knowledge protocols, are Galois automorphisms. These automorphisms have properties that we can use to create protocols with larger challenge spaces, and they can be used to reduce the soundness error of protocols. They can also be used to construct functions whose constant coefficients equals desired inner products.

We let $N \geq k > 1$ be powers of 2 and let $q = 2k + 1 \pmod{4k}$ be a prime. By Lemma 2 we then have the factorization

$$X^N + 1 = (X^{N/k} - \zeta_1) \cdot \ldots \cdot (X^{N/k} - \zeta_k),$$

where $\zeta_i$ are the primitive $2k$-th roots of unity in $\mathbb{Z}_q$ and all $X^{N/k} - \zeta_i$ are irreducible modulo $q$. We now consider the group of automorphisms of $R_q$, $\mathsf{Aut}(R_q)$, and note that this is isomorphic to $\mathbb{Z}_{2N}^\times$ by the isomorphism

$$i \mapsto \sigma_i \ : \ \mathbb{Z}_{2N}^\times \to \mathsf{Aut}(R_q),$$

where $\sigma_i$ is defined by $\sigma_i(X) = X^i$. We call these Galois automorphisms. We now consider the prime ideal $(X^{N/k} - \zeta)$, for some $\zeta \in \mathbb{Z}_q$. In a suitable extension field of $\mathbb{Z}_q$, the roots of $X^{N/k} - \zeta^{i^{-1}}$ will also be roots of $X^{iN/k} - \zeta$. Hence we get that for all $i \in \mathbb{Z}_{2N}^\times$

$$\sigma_i(X^{N/k} - \zeta) = (X^{iN/k} - \zeta) = (X^{N/k} - \zeta^{i^{-1}}).$$

This then implies that for $f \in R_q$,

$$\sigma_i\Big(f \mod (X^{N/k} - \zeta)\Big) = \sigma_i(f) \mod (X^{N/k} - \zeta^{i^{-1}}).$$

We can now use Lemma 2.4 of [19] derive that the cyclic subgroup $\langle 2k + 1 \rangle \subset \mathbb{Z}_{2N}^\times$ has order $N/k$, and thus stabilizes every prime ideal $(X^{N/k} - \zeta)$ since $\zeta$ is a primitive $2k$-th root of unity, and of order $2k$. Thus $\mathbb{Z}_{2N}^\times / \langle 2k+1 \rangle$ has order $k$, and will act transitively on the $k$ prime ideals $(X^{N/k} - \zeta)$, meaning that we can index the prime ideals by $i \in \mathbb{Z}_{2N}^\times / \langle 2k + 1 \rangle$ and thus write

$$(X^N + 1) = \prod_{i \in \mathbb{Z}_{2N}^\times / \langle 2k+1 \rangle} (X^{N/k} - \zeta^i).$$

If we for $l$ such that $l|k$ let $i$ run over $\langle 2k/l + 1 \rangle / \langle 2k + 1 \rangle$, then the product of the $l$ prime ideals $(X^{N/k} - \zeta^i)$ will be

$$\prod_{i \in \langle 2k/l+1 \rangle / \langle 2k+1 \rangle} (X^{N/k} - \zeta^i) = (X^{lN/k} - \zeta^l).$$

Using this, we can partition the $k$ prime ideals into $k/l$ groups of $l$ ideals,

$$(X^N+1) = \prod_{j\in\mathbb{Z}_{2N}^\times/\langle 2k/l+1\rangle} (X^{lN/k} - \zeta^{jl}) = \prod_{j\in\mathbb{Z}_{2N}^\times/\langle 2k/l+1\rangle} \prod_{i\in\langle 2k/l+1\rangle/\langle 2k+1\rangle} (X^{N/k} - \zeta^{ij}).$$

We note that we have the isomorphism $\mathbb{Z}_{2N}^\times/\langle 2k/l+1\rangle \cong \mathbb{Z}_{2k/l}^\times$, and that $\big((2k/l+1)^i\big)_{i=1,\ldots,l-1}$ form a complete set of representatives for $\langle 2k/l+1\rangle/\langle 2k+1\rangle$. Hence for $\sigma = \sigma_{2k/l+1} \in \mathsf{Aut}(R_q)$, we can instead instead write

$$(X^N+1) = \prod_{j\in\mathbb{Z}_{2k/l}^\times} \prod_{i=0}^{l-1} \sigma^i\Big(X^{N/k} - \zeta^j\Big).$$

We also present an important result that we get by using the automorphism $\sigma_{-1} \in \mathsf{Aut}(R_q)$. We define the map $\mathsf{T} : \mathbb{Z}^{kN} \times \mathbb{Z}^{kN} \to R$ as

$$\mathsf{T}(\vec{a},\vec{b}) := \sum_{i=0}^{k-1} \sigma_{-1}\Big(\sum_{j=0}^{N-1} a_{iN+j}X^j\Big) \cdot \Big(\sum_{j=0}^{N-1} b_{iN+j}X^j\Big) \in R,$$

given vectors $\vec{a} = (a_0,\ldots,a_{kN-1})$ and $\vec{b} = (b_0,\ldots,b_{kN-1})$. We now have a simple and very useful property of $\mathsf{T}$, that we will use to prove inner products.

**Lemma 8.** Let $\vec{a},\vec{b} \in \mathbb{Z}^{kN}$ for $k \geq 1$. Then the constant coefficient of $\mathsf{T}\big(\vec{a},\vec{b}\big)$ is equal to $\langle \vec{a},\vec{b}\rangle$.

## 3.6 Number theoretic transform

Some of the first lattice-based zero-knowledge protocols that were constructed by using commitment schemes, also relied on the number theoretic transform. Suppose we choose parameters such that we by Lemma 2 have the isomorphism

$$\mathbb{Z}_q[X]/(X^N+1) \cong \prod_{i\in\mathbb{Z}_{2k}^\times} \mathbb{Z}_q[X]/(X^{N/k} - \zeta^i).$$

We can then define the number theoretic transform (NTT) of polynomial $a \in R_q$, $\hat{a}$, to be the image of $a$ under this isomorphism. Namely, $\hat{a} = \mathsf{NTT}(a) = (\hat{a}_i)_{i\in\mathbb{Z}_{2k}^\times}$ where $\hat{a}_i = a \mod (X^{N/k} - \zeta^i)$. Due to the Chinese remainder theorem, the inverse of this map exists, and we denote this by $\check{a} = \mathsf{NTT}^{-1}(a)$.

We now give some of the very useful properties the $\mathsf{NTT}$ representation that we will use in this paper.

- For any $a,b,c \in R_q$ we have that $ab = c$ if and only if $\hat{a} \circ \hat{b} = \hat{c}$, where $\circ$ denotes column-wise multiplication.
- $\mathsf{NTT}(ab) = \mathsf{NTT}(a) \circ \mathsf{NTT}(b)$
- An element $a \in R_q$ is invertible if an only if all its $\mathsf{NTT}$ coefficients are non-zero [3].

We also present a useful Lemma by Esgin et al. in [10] that states that the scaled sum of the $\mathsf{NTT}$ coefficients of a polynomial is equal to its $N/k$ first coefficients.

**Lemma 9.** Let $a \in R_q$. Then $\frac{1}{k}\sum_{i\in\mathbb{Z}_{2k}^\times} \hat{a}_i = a_0 + a_1X + \ldots + a_{N/k-1}X^{N/k-1}$, when we lift the $\hat{a}_i$ to $\mathbb{Z}_q[X]$.

## 3.7 Challenge spaces

The zero-knowledge protocols we present in this paper need to have challenge spaces for which the difference of any two challenges is invertible. This is crucial for the soundness property of the protocols. The size of the challenge space will also determine the soundness error of the protocol, and hence the total size of the proof. It is thus important to construct as large challenge spaces $\mathcal{C}$ as possible, while also making sure that all elements in the set of differences of challenge elements, $\overline{\mathcal{C}} = \{c - c' \mid c, c' \in \mathcal{C}, c \neq c'\}$, are invertible. In this section we present some results that can be used in order to construct challenge spaces with the desired properties.

**Lemma 10** ([6])**.** Let $q$ be such that $X^N + 1$ splits into linear factors. Then the polynomials $X^i - X^j \in R_q$ for $i \not\equiv j \pmod{2N}$ are invertible.

*Proof.* Let $\zeta \in \mathbb{Z}_q$ be one of the primitive $2N$-th roots of unity such that $X^N + 1 \equiv \prod_{j=1}^{N}(X - \zeta_j)$. Then $X^i - X^j \mod (X - \zeta) = \zeta^i - \zeta^j$. This is zero in $\mathbb{Z}_q$ if and only if $i \equiv j \pmod{2N}$, and hence $X^i - X^j$ is invertible when $i \not\equiv j \pmod{2N}$. □

We present an important result from [16] that builds upon Lemma 2 with $k = 2$.

**Lemma 11** ([16])**.** Let $q = 5 \pmod 8$ be a prime. Take any $c \in R_q$ such that $\sigma_{-1}(c) = c$. Then, $c$ is invertible over $R_q$ if and only if $c \neq 0$.

*Proof.* We get from Lemma 2 that

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

for some $r \in \mathbb{Z}_q$ such that $X^{N/2} \pm r$ are irreducible modulo $q$. From the assumption that $\sigma_1(c) = c$, we get that $c$ can be written as

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

Thus we get

$$c \mod q \mod (X^{N/2} \pm r) = c_0 + \sum_{i=1}^{N/2-1} (c_i \pm r c_{N/2-i}) X^i, \tag{3.1}$$

and so if $c \neq 0$, then at least one of the coefficients $c_0, \ldots, c_{N/2-1} \in \mathbb{Z}_q$ is non-zero. Suppose now that $c_i \neq 0$. We now have two cases:

- if $i = N/4$, then $c_i \pm r c_{N/2-i} = c_{N/4} \pm r c_{N/4}$ must be non-zero, since $r \neq \pm 1$.
- if $i \neq N/4$, then for any sign $b \in \{-1, 1\}$, either $c_i - b r c_{N/2-i}$ or $c_{N/2-i} - b r c_i$ is non-zero. We can see this by the fact that if we assume that both are zero, namely $c_i = b r c_{N/2-i}$ and $c_{N/2-i} = b r c_i$, we get that

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

  which is a contradiction, since $c_i \neq 0$.

Hence we get that both $c \mod q \mod (X^{N/2} - r)$ and $c \mod q \mod (X^{N/2} + r)$ are non-zero, and by the Chinese Remainder Theorem, we get that $c$ has an inverse in $R_q$. □

Since the zero-knowledge protocols in this paper uses rejection sampling algorithms with secrets of the form $c\boldsymbol{r}$ for a challenge $c \in \mathcal{C}$ and $\boldsymbol{r} \in R_q^\ell$, we need a way to bound $\|c\boldsymbol{r}\|$, so that we can set the standard deviation for rejection sampling. We now introduce such a bound, that involves the $\sigma_{-1}$ automorphism.

**Lemma 12** ([16])**.** Let $\boldsymbol{r} \in R^\ell$ and $c \in R$. Then, for any $k$ that is a power of 2, we have that

$$\|c\boldsymbol{r}\| \leq \sqrt[2k]{\|\sigma_{-1}(c^k)c^k\|_1}\|\boldsymbol{r}\|.$$

*Proof.* Let $C \in \mathbb{Z}^{d\times d}$ be the rotation matrix of $c = c_0 + c_1X + \ldots + c_{d-1}X^{d-1}$:

$$C = \mathsf{Rot}(c) = \begin{bmatrix} c_0 & -c_{d-1} & \ldots & -c_1 \\ c_1 & c_0 & \ldots & -c_2 \\ \vdots & \vdots & \ldots & \vdots \\ c_{d-1} & c_{d-2} & \ldots & c_0 \end{bmatrix}.$$

We now want to upper-bound the operator norm $\|C\|$ of the matrix $C$. To achieve this, we will use that $\|C\| = \sqrt{\|C^TC\|}$ and that for any $k$ that is a power of two we have $\|C^TC\|^k = \|(C^TC)^k\|$, since $C^TC$ is symmetric. We also note that $\|\mathsf{Rot}(u)\| \leq \|(u)\|_1$ for all $u \in R$. We can now use the observation that $C^T = \mathsf{Rot}(\sigma_{-1}(c))$, to deduce that

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

Hence we have that $\|C\| \leq \sqrt[2k]{\|\sigma_{-1}(c^k)c^k\|_1}$, and hence the statement holds. □

# Chapter 4

# Lattice based commitment schemes

The first lattice based commitment scheme that was used to construct cryptographic primitives was the standard Ajtai commitment scheme [1]. This scheme was originally was defined over $\mathbb{Z}_q$, but is easily expanded to $R_q$ so that it bases its security on the $\mathsf{MSIS}$ and $\mathsf{MLWE}$ problems. We now give an informal description of this scheme.

In order to commit to a vector $\boldsymbol{s}_1 \in R_q^{m_1}$ for which $\|\boldsymbol{s}_1\|$ is small, we let $\boldsymbol{A}_1 \xleftarrow{\$} R_q^{n \times m_1}$ and $\boldsymbol{A}_2 \xleftarrow{\$} R_q^{n \times m_2}$ be public parameters, such that $m_2$ is much larger than $n$. We then sample randomness $\boldsymbol{s}_2 \xleftarrow{\$} R_q^{m_2}$ such that $\boldsymbol{s}_2$ has small coefficients, and output the commitment vector

$$\boldsymbol{t} = \boldsymbol{A}_1 \boldsymbol{s}_1 + \boldsymbol{A}_2 \boldsymbol{s}_2 \in R_q^n. \tag{4.1}$$

We notice that $\boldsymbol{A}_2 \boldsymbol{s}_2$ is indistinguishable from a truly uniform vector if the $\mathsf{MLWE}_{m_2-n,n}$ problem is hard. This ensures the hiding property of the scheme, since the whole commitment vector will then be indistinguishable uniformly random. In order to see that the scheme is binding, we assume that we are able to come up with $(\boldsymbol{s}_1, \boldsymbol{s}_2) \neq (\boldsymbol{s}_1', \boldsymbol{s}_2')$ such that

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

This implies that

$$\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_1 - \boldsymbol{s}_1' \\ \boldsymbol{s}_2 - \boldsymbol{s}_2' \end{bmatrix} = 0$$

for a non-zero vector $\begin{bmatrix} \boldsymbol{s}_1 - \boldsymbol{s}_1' \\ \boldsymbol{s}_2 - \boldsymbol{s}_2' \end{bmatrix}$ that by construction has small coefficients. Hence we have a $\mathsf{MSIS}_{n,m_1+m_2}$ solution for the matrix $\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{bmatrix}$.

The main disadvantage of this commitment scheme is that it only allows for small message spaces, since the messages must be of small norm. On the other hand, the size of the commitment does not depend the size of the message $m_1$. It depends on the parameter $n$, which has to be chosen large enough so that the $\mathsf{MSIS}$ and $\mathsf{MLWE}$ problems are hard.

It is clear that more practical commitment schemes that can commit to arbitrary vectors are needed for many applications. Baum et al. presented such a practical scheme with unbounded message space in [4]. The disadvantage of this scheme is however that the commitment size is linear in the

message size, which means that commitments potentially can be very large. We present this scheme in Section 4.1, together with a zero-knowledge proof of opening knowledge.

In constructing zero-knowledge proofs of lattice elements using this commitment scheme, the commitment size has eventually become the biggest obstacle in reducing the proof sizes. In response to this Lyubashevsky et al. recently proposed a new commitment scheme in [16]. This new scheme combines the commitment schemes by Ajtai and Baum et al., in order to exploit the advantages of both schemes. We present this scheme, together with a zero-knowledge proof of opening knowledge, in Section 4.2.

## 4.1 BDLOP commitment scheme

We present the commitment scheme of Baum et al. [4], which is a practical scheme that allows one to commit to vectors over $R_q$. The main purpose of this commitment scheme is to be used in zero-knowledge protocols, and so we need to define a suitable challenge space that will ensure soundness of such protocols. We define the challenge space for the commitment scheme as

$$\mathcal{C} = \{c \in R_q \mid \|c\|_\infty = 1, \|c\|_1 = \kappa\}$$

for a parameter $\kappa$ that determines the size of $\mathcal{C}$. We also assume that $q$ is chosen such that all elements of $\ell_\infty$ norm at most 2 are invertible in $R_q$, as per Lemma 2. We can then define the set of differences as $\overline{\mathcal{C}} = \{c - c' \mid c, c' \in \mathcal{C}, c \neq c'\}$, for which all elements will be invertible.

We also note that there is no efficient zero-knowledge protocol for simply proving knowledge of the message and the randomness that was used to commit. Such protocols can only prove something weaker, and we will account for this in the the opening algorithm of the scheme, so that the scheme will still be binding with respect to such relaxed openings. We now explain the algorithms KeyGen, Commit and Open for this scheme.

BDLOP.KeyGen: In order to create the public parameters that can be used to commit to messages $\boldsymbol{m} \in R_q^\ell$, we sample matrices $\boldsymbol{A}_1 \xleftarrow{\$} R_q^{n \times k}$ and $\boldsymbol{A}_2 \xleftarrow{\$} R_q^{\ell \times k}$. We also select a constant $\beta$ that defines the randomness distribution $\chi^k$, where $\chi$ is the uniform distribution on $S_\beta$.

BDLOP.Commit: In order to commit to a message $\boldsymbol{m} \in R_q^\ell$ with the public parameters $(\boldsymbol{A}_1, \boldsymbol{A}_2)$, we start by sampling randomness $\boldsymbol{r} \xleftarrow{\$} \chi^k$ and output the commitment

$$\mathsf{BDLOP.Commit}(\boldsymbol{m}; \boldsymbol{r}) := \begin{bmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix} \boldsymbol{r} + \begin{bmatrix} \boldsymbol{0}^n \\ \boldsymbol{m} \end{bmatrix}. \tag{4.2}$$

Since there are no efficient zero-knowledge proofs for proving knowledge of $\boldsymbol{r}$ and $\boldsymbol{m}$ that satisfies (4.2), we let the opening algorithm also take a challenge $f \in \overline{\mathcal{C}}$ as input. An honest prover will simply just put $f = 1$.

BDLOP.Open: For a commitment $\boldsymbol{t} = \begin{bmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \end{bmatrix}$, $(\boldsymbol{m}, \boldsymbol{r}, f) \in R_q^\ell \times R_q^k \times \overline{\mathcal{C}}$ is a valid opening if

$$f \begin{bmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix} \boldsymbol{r} + f \begin{bmatrix} \boldsymbol{0}^n \\ \boldsymbol{m} \end{bmatrix}, \tag{4.3}$$

and $\|r_i\| \leq 4\mathfrak{s}\sqrt{N}$ for all $i$. $\mathsf{BDLOP.Open}(\boldsymbol{m}, \boldsymbol{r}, f, \boldsymbol{t})$ outputs 1 if $(\boldsymbol{m}, \boldsymbol{r}, f)$ is a valid opening of $\boldsymbol{t}$, and 0 if not.

We now explain how we can set the standard deviation. Since $\|\boldsymbol{r}\| \leq \beta\sqrt{kN}$, and $\|c\|_1 = \kappa$ for all $c \in \mathcal{C}$, we have that $\|c\boldsymbol{r}\| \leq \kappa\beta\sqrt{kN}$. Hence, in accordance with Lemma 4, we use standard deviation $\mathfrak{s} = \gamma\kappa\beta\sqrt{kN}$ for a $\gamma > 0$.

**Variants of the scheme**
We now explain two variants of this commitment scheme that we will use.

**Variant 1.** For randomly chosen polynomials $b_{i,j} \in R_q$ we define $\boldsymbol{B} \in R_q^{5\times 6}$ as

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{b}_0 \\ \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4 \end{bmatrix} = \begin{bmatrix} 1 & b_{0,2} & b_{0,3} & b_{0,4} & b_{0,5} & b_{0,6} \\ 0 & 1 & 0 & 0 & 0 & b_{1,6} \\ 0 & 0 & 1 & 0 & 0 & b_{2,6} \\ 0 & 0 & 0 & 1 & 0 & b_{3,6} \\ 0 & 0 & 0 & 0 & 1 & b_{4,6} \end{bmatrix}$$

We can now commit to messages $\boldsymbol{m} = (m_1, m_2, m_3, m_4)^T \in R_q^4$ by sampling a random vector $\boldsymbol{r} \xleftarrow{\$} S_\beta^6$ and compute the commitment as

$$\boldsymbol{t} = \begin{bmatrix} \boldsymbol{t}_0 \\ \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \\ \boldsymbol{t}_3 \\ \boldsymbol{t}_4 \end{bmatrix} = \boldsymbol{B} \cdot \boldsymbol{r} + \begin{bmatrix} 0 \\ m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix}$$

**Variant 2.** In order to commit to a message vector $\boldsymbol{m} = (m_1, \ldots, m_\ell) \in R_q^\ell$, we draw a uniformly random matrix $\boldsymbol{B}_0 \xleftarrow{\$} R_q^{\mu\times(\lambda+\mu+\ell)}$ and vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_\ell \xleftarrow{\$} R_q^{\lambda+\mu+\ell}$. We then sample randomness $\boldsymbol{r} \xleftarrow{\$} \chi^{(\lambda+\mu+\ell)N}$ and compute the commitment as

$$\begin{aligned} \boldsymbol{t}_0 &= \boldsymbol{B}_0\boldsymbol{r}, \\ \boldsymbol{t}_i &= \langle \boldsymbol{b}_i, \boldsymbol{r}\rangle + m_i \text{ for } i = 1, \ldots, \ell. \end{aligned}$$

### 4.1.1 Hiding and binding

We now prove that this scheme has the desired security properties.

**Theorem 1.** Suppose that $\mathfrak{s} = \gamma\kappa\beta\sqrt{kN}$ for some $\gamma > 0$. Then the $\mathsf{BDLOP}$ commitment scheme is complete.

*Proof.* Given public parameters $\begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix} \xleftarrow{\$} \mathsf{BDLOP.KeyGen}$ and a message $\boldsymbol{m} \in R_q^\ell$, then an honestly generated commitment $\begin{bmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \end{bmatrix}$ with randomness $\boldsymbol{r}$ will obviously satisfy (4.3) with $f = 1$. Since the randomness $\boldsymbol{r}$ was sampled from $S_\beta^k$, we have that $\|r_i\| \leq \sqrt{N} \cdot \|r_i\|_\infty \leq \sqrt{N}\beta$ for all $i = 1, \ldots, k$. Since $\mathfrak{s} = \gamma\kappa\beta\sqrt{kN}$, $\beta$ is clearly smaller than $4\mathfrak{s}$, and so we must further have that $\|r_i\| \leq 4\mathfrak{s}\sqrt{N}$. Hence $\mathsf{BDLOP.Open}(\boldsymbol{m}, \boldsymbol{r}, 1, \mathsf{BDLOP.Commit}(\boldsymbol{m}; \boldsymbol{r})) = 1$, and the scheme is complete. □

We show now the the hiding property of the commitment scheme can be reduced to the MLWE problem. This will imply that breaking the hiding property is at least as hard as solving the MLWE problem, and so if we assume the the latter problem is hard, then the scheme must be hiding.

**Theorem 2.** If there exists an algorithm $\mathcal{A}$ that has advantage $\varepsilon$ in breaking the hiding property of the BDLOP commitment scheme, then there exists an algorithm $\mathcal{A}'$ that runs in the same time and has advantage $\varepsilon$ in solving the $\mathsf{MLWE}_{k,n+\ell,\chi}$ problem.

*Proof.* We now show how we can construct such an algorithm $\mathcal{A}'$. Suppose that $\mathcal{A}'$ is given a $\mathsf{MLWE}_{k,n+\ell,\chi}$-instance $\boldsymbol{A} = \begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{A}_2\end{bmatrix} \in R_q^{(n+\ell)\times k}$, $\boldsymbol{b} \in R_q^{n+\ell}$. $\mathcal{A}'$ now sets up for running the hiding game with $\mathcal{A}$, and outputs $(\boldsymbol{A}_1, \boldsymbol{A}_2)$ as the public parameters.

If $\mathcal{A}'$ now receives $\boldsymbol{m}_0, \boldsymbol{m}_1 \in R_q^\ell$ form $\mathcal{A}$, $\mathcal{A}'$ samples $b \xleftarrow{\$} \{0,1\}$ and sends the commitment of $\boldsymbol{m}_b$,

$$\begin{bmatrix}\boldsymbol{t}_1\\ \boldsymbol{t}_2\end{bmatrix} = \boldsymbol{b} + \begin{bmatrix}\boldsymbol{0}^n\\ \boldsymbol{m}_b\end{bmatrix},$$

to $\mathcal{A}$. If $\mathcal{A}$ responds with $b' = b$, then $\mathcal{A}'$ outputs 1, and 0 if not. We now have two cases:

- If $\mathcal{A}'$ was given a truly uniformly random $\boldsymbol{b}$, then the output $\begin{bmatrix}\boldsymbol{t}_1\\ \boldsymbol{t}_2\end{bmatrix}$ is independent of $\boldsymbol{m}_b$, and thus the probability that $\mathcal{A}$ outputs the correct $b'$ is exactly $1/2$.
- If $\mathcal{A}'$ was given $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{r}$, for some $\boldsymbol{r} \xleftarrow{\$} \chi^k$, then the output commitment is

  $$\begin{bmatrix}\boldsymbol{t}_1\\ \boldsymbol{t}_2\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{A}_2\end{bmatrix}\boldsymbol{r} + \begin{bmatrix}\boldsymbol{0}^n\\ \boldsymbol{m}_b\end{bmatrix} = \mathsf{BDLOP.Commit}(\boldsymbol{m}_b; \boldsymbol{r}),$$

  and by assumption $\mathcal{A}$ outputs the correct $b'$ with probability $1/2 + \varepsilon$.

Hence the advantage of $\mathcal{A}'$ in solving the $\mathsf{MLWE}_{k,n+\ell,\chi}$ problem is $\varepsilon$. □

We now show that the binding property of the commitment scheme can be reduced to the MSIS problem. Again, this will imply that breaking the binding property of the scheme is at least as hard as the MSIS problem, and thus the scheme is binding if the problem is hard.

**Theorem 3.** If there is an algorithm $\mathcal{A}$ who can break the binding property of the BDLOP commitment scheme with probability $\varepsilon$, then there is an algorithm $\mathcal{A}'$ with advantage $\varepsilon$ in solving the $\mathsf{MSIS}_{n,k,B}$ problem, for $B = 16\mathfrak{s}\sqrt{\kappa N}$.

*Proof.* We now show how we can construct such an algorithm $\mathcal{A}'$. Suppose that $\mathcal{A}'$ is given an $\mathsf{MSIS}_{n,k,B}$-instance $\boldsymbol{A}_1 \in R_q^{n\times k}$, for which it is supposed to find a vector $\boldsymbol{y} \in R_q^k$ such that $\boldsymbol{A}_1\boldsymbol{y} = 0$ and $\|\boldsymbol{y}\| \leq B$. $\mathcal{A}'$ creates a random $\boldsymbol{A}_2 \xleftarrow{\$} R_q^{\ell\times k}$ and outputs $(\boldsymbol{A}_1, \boldsymbol{A}_2)$ as public parameters. By assumption, with probability $\varepsilon$, $\mathcal{A}$ is able to come up with a commitment $\begin{bmatrix}\boldsymbol{t}_1\\ \boldsymbol{t}_2\end{bmatrix}$ with two different valid openings, $(\boldsymbol{m}, \boldsymbol{r}, f)$ and $(\boldsymbol{m}', \boldsymbol{r}', f')$ such that $\boldsymbol{m} \neq \boldsymbol{m}'$. This then implies that

$$f\begin{bmatrix}\boldsymbol{t}_1\\ \boldsymbol{t}_2\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{A}_2\end{bmatrix}\boldsymbol{r} + f\begin{bmatrix}\boldsymbol{0}^n\\ \boldsymbol{m}\end{bmatrix},$$
$$f'\begin{bmatrix}\boldsymbol{t}_1\\ \boldsymbol{t}_2\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{A}_2\end{bmatrix}\boldsymbol{r}' + f'\begin{bmatrix}\boldsymbol{0}^n\\ \boldsymbol{m}'\end{bmatrix}.$$

If we multiply the first equation by $f'$ and the second by $f$, we get that the left hand sides are both the same, and thus we get that

$$\begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix} f'\boldsymbol{r} + f'f \begin{bmatrix} \boldsymbol{0}^n \\ \boldsymbol{m} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix} f\boldsymbol{r}' + ff' \begin{bmatrix} \boldsymbol{0}^n \\ \boldsymbol{m'} \end{bmatrix}.$$

By subtracting and splitting this up, we get

$$\begin{aligned} \boldsymbol{A}_1(f'\boldsymbol{r} - f\boldsymbol{r}') &= \boldsymbol{0}^n \\ \boldsymbol{A}_2(f'\boldsymbol{r} - f\boldsymbol{r}') + (ff'\boldsymbol{m} - ff'\boldsymbol{m}') &= \boldsymbol{0}^\ell \end{aligned}$$

We note that $ff'(\boldsymbol{m} - \boldsymbol{m}') \neq \boldsymbol{0}^\ell$, since $f$ and $f'$ are invertible and $\boldsymbol{m} \neq \boldsymbol{m}'$. Then the last equation implies that $(f'\boldsymbol{r} - f\boldsymbol{r}') \neq \boldsymbol{0}^k$, which would imply that we have a solution to the $\mathsf{MSIS}_{n,k,B}$ problem for matrix $\boldsymbol{A}_1$ if $\|(f'\boldsymbol{r} - f\boldsymbol{r}')_i\| \leq B$ for all $i$.

By the definition of the challenge space we have that $\|f\| \leq 2\sqrt{\kappa}$ for every $f \in \overline{\mathcal{C}}$. We also have that $\|r_i\| \leq 4\mathfrak{s}\sqrt{N}$ for every polynomial of the vector $\boldsymbol{r}$, since it is a valid opening. The same holds for $\boldsymbol{r}'$, and hence we get that $\|f'r_i\|, \|fr_i'\| \leq 8\mathfrak{s}\sqrt{\kappa N}$. This implies that we have the bound $\|(f'\boldsymbol{r} - f\boldsymbol{r}')_i\| \leq 16\mathfrak{s}\sqrt{\kappa N}$ for all $i$, and thus $\mathcal{A}'$ has advantage $\varepsilon$ in solving the $\mathsf{MSIS}_{n,k,16\mathfrak{s}\sqrt{\kappa N}}$ problem. □

### 4.1.2 Opening proof

In order to be able to use commitments as a part of zero-knowledge protocols, we need a zero-knowledge proof of knowledge of a valid opening to a commitment, in order to show that the commitments are constructed correctly, without revealing anything about the message or the randomness. We will now describe such a zero-knowledge proof of opening to a commitment. This scheme will be almost identical to the 'Fiat-Shamir with aborts' protocol from [15], and we note that only non-aborting transcripts are honest-verifier zero-knowledge. This is however not a problem, since in most applications one uses non-interactive versions of the protocol. The opening proof is given as $\Pi_{\mathsf{open}}$ in Figure 4.1.

**Theorem 4.** Suppose that $\mathfrak{s} = \gamma\kappa\beta\sqrt{kN}$ for a $\gamma > 0$. Then the protocol $\Pi_{\mathsf{open}}$ is complete, meaning that the honest prover convinces the verifier with probability

$$\approx \frac{1}{\exp(14/\gamma + 1/(2\gamma^2))}.$$

*Proof.* An honest prover can answer correctly according to the protocol for every challenge $c$, since it knows $\boldsymbol{r}$. By Lemma 4, the probability that $\mathsf{Rej}_1(\boldsymbol{z}, c\boldsymbol{r}, \mathfrak{s})$ does not abort is at least

$$\frac{1}{\exp(14/\gamma + 1/(2\gamma^2))},$$

and $\boldsymbol{z}$ is within statistical distance of $2^{-128}$ from $D_{R^k,\mathfrak{s}}$. We can then use the tail-bound Lemma 3 with $\delta = 2$ to deduce that, except with negligible probability $2^{N/2}\exp(-3N/2)$, $\|z_i\| < 2\mathfrak{s}\sqrt{N}$ for all components $z_i$ of $\boldsymbol{z}$. Therefore, the verifier will with overwhelming probability accept when an honest prover does not abort, and the theorem holds. □

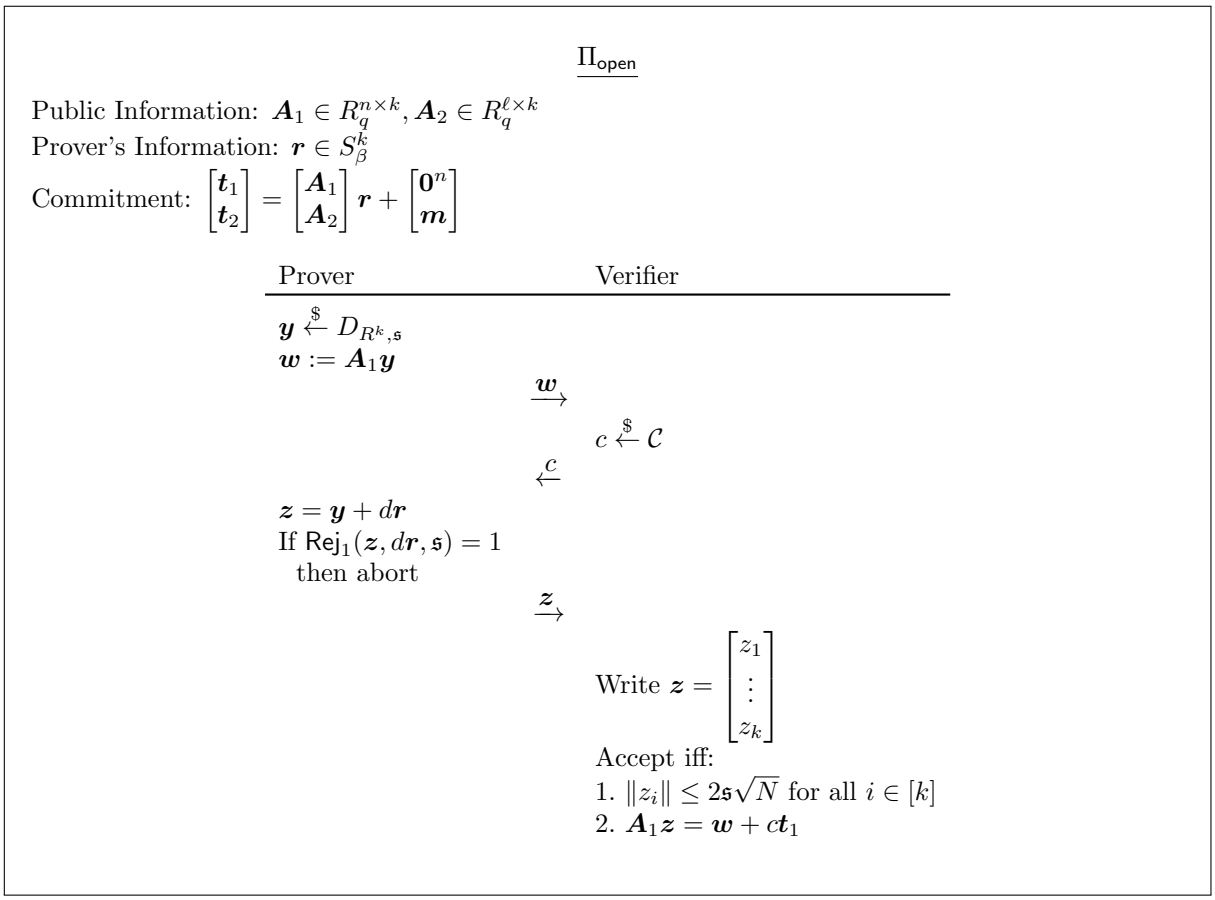



Figure 4.1: Proof of knowledge $\Pi_{\mathsf{open}}$ of $(\boldsymbol{m}, \boldsymbol{r}, f) \in R_q^\ell \times R_q^k \times \overline{\mathcal{C}}$ satisfying $f\boldsymbol{t}_1 = \boldsymbol{A}_1\boldsymbol{r}$, $f\boldsymbol{t}_2 = \boldsymbol{A}_2\boldsymbol{r} + f\boldsymbol{m}$ and $\|r_i\| \le 4\mathfrak{s}\sqrt{N}$ for all $i \in [k]$.

**Theorem 5.** The protocol $\Pi_{\mathsf{open}}$ satisfies the honest-verifier zero-knowledge property, meaning that there exists a simulator $\mathcal{S}$, that without access to secret information outputs a simulation of a non-aborting transcript of the protocol between an honest prover and verifier, which has statistical distance at most $2^{-128}$ to the actual transcript.

*Proof.* We can construct this simulator $\mathcal{S}$ by letting it draw a random $c$ from $\mathcal{C}$ and sample a random $\boldsymbol{z}$ from $D_{R^k,\mathfrak{s}}$, and by setting $\boldsymbol{w} = \boldsymbol{A}_1\boldsymbol{z} - d\boldsymbol{t}_1$. Since the $\boldsymbol{z}$ in a real protocol is within statistical distance of $2^{-128}$ from $D_{R^k,\mathfrak{s}}$ according to Lemma 4, and $\boldsymbol{z}$ in independent of $c$, the simulated $\boldsymbol{z}$ will be statistically indistinguishable from the one in the real protocol.

By construction of $\boldsymbol{w}$, we get the desired relation $\boldsymbol{A}_1\boldsymbol{z} = \boldsymbol{w} + d\boldsymbol{t}_1$. By the tail bound in Lemma 3 with $\delta = 2$, we also have, except with negligible probability $2^{N/2}\exp(-3N/2)$, that $\|z_i\| < 2\mathfrak{s}\sqrt{N}$ for all components $z_i$ of $\boldsymbol{z}$. Hence the simulation is within statistical distance of $2^{-128}$ from a real transcript, and $\Pi_{\mathrm{OPEN}}$ is honest-verifier zero-knowledge. □

**Theorem 6.** The protocol $\Pi_{\mathsf{open}}$ satisfies the special soundness property, meaning that given a

commitment $\boldsymbol{t}$ and a pair of different transcripts for $\Pi_{\mathsf{open}}$, $(\boldsymbol{w}, c, \boldsymbol{z}), (\boldsymbol{w}, c', \boldsymbol{z}')$ where $c \neq c'$, we can extract a valid opening $(\boldsymbol{m}, \boldsymbol{r} = \begin{bmatrix} r_1 \\ \vdots \\ r_k \end{bmatrix}, f)$ of $\boldsymbol{t}$, with $\|r_i\| \leq 4\mathfrak{s}\sqrt{N}$ and $f \in \overline{\mathcal{C}}$.

*Proof.* If we have two different valid transcripts for different challenges $c, c'$, we can easily compute $f = (c - c') \in \overline{\mathcal{C}}$ and $\boldsymbol{r} = \begin{bmatrix} r_1 \\ \vdots \\ r_k \end{bmatrix} = \boldsymbol{z} - \boldsymbol{z}'$, for which $\boldsymbol{A}_1\boldsymbol{r} = f\boldsymbol{t}_1$. This will hold since $\boldsymbol{A}_1\boldsymbol{z} = \boldsymbol{w} + c\boldsymbol{t}_1$ and $\boldsymbol{A}_1\boldsymbol{z}' = \boldsymbol{w} + c'\boldsymbol{t}_1$. We can now define the message $\boldsymbol{m} = \boldsymbol{t}_2 - f^{-1}\boldsymbol{A}_2\boldsymbol{r}$. Since the components of $\boldsymbol{z}$ and $\boldsymbol{z}'$ are bounded by $2\mathfrak{s}\sqrt{N}$ in the $\ell_2$ norm, we have that $\|r_i\| \leq \|z_i\| + \|z_i'\| \leq 4\mathfrak{s}\sqrt{N}$. Because we defined $\boldsymbol{m}$ by $f\boldsymbol{m} = f\boldsymbol{t}_2 - \boldsymbol{A}_2\boldsymbol{r}$, we also have that $\begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \end{bmatrix} \boldsymbol{r} + f \begin{bmatrix} \boldsymbol{0}^n \\ \boldsymbol{m} \end{bmatrix} = f \begin{bmatrix} \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \end{bmatrix}$, and hence $(\boldsymbol{m}, \boldsymbol{r}, f)$ is a valid opening of $\boldsymbol{t}$. □

## 4.2 ABDLOP commitment scheme

We recall that the main disadvantage of the BDLOP commitment scheme is that the commitments are much more expensive than the standard Ajtai commitments [1]. The new scheme presented by Lyubashevsky et al. in [16] allows for using the much cheaper Ajtai commitment scheme for the parts of the message that are small. This new commitment scheme, called ABDLOP, is a combination of the Ajtai and the BDLOP commitment scheme. The complete description of the scheme is given by Nguyen in [22].

This commitment scheme uses a more efficient challenge space, that is constructed by using Galois automorphisms $\sigma$. We remember that this determines the standard deviation we are able to use for rejection sampling. For a fixed $k$ that is a power of two, we define the challenge space for the commitment scheme as

$$\mathcal{C} = \{c \in S_\omega \mid \sigma(c) = c, \ \sqrt[2k]{\|\sigma_{-1}(c^k)c^k\|_1} \leq \eta\},$$

where $\omega$ is chosen such that we by Lemma 2 ensure that all elements of $\overline{\mathcal{C}} = \{c - c' \mid c, c' \in \mathcal{C}, c \neq c'\}$ are invertible, and $\eta$ is chosen such that for a $c \xleftarrow{\$} S_\omega$ we have $\sqrt[2k]{\|\sigma_{-1}(c^k)c^k\|_1} \leq \eta$ with high probability. We can then bound multiplication by challenge elements according to Lemma 12. Another advantage of this challenge space is that if we choose $\sigma := \sigma_{-1}$, we can use Lemma 11 to ensure that elements of $\overline{\mathcal{C}}$ is invertible, instead of Lemma 2. This is a common choice, and it increases the size of the challenge space.

We remember that we can commit to vectors $\boldsymbol{s}_1$ of small norm as in (4.1), and to vectors $\boldsymbol{m}$ of larger norm as in (4.2). We now divide the vector $\boldsymbol{s}$ we want to commit to into a smaller Ajtai part and a larger BDLOP part, namely $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{m})$. This allows us to commit to $\boldsymbol{s}_1$ in the Ajtai part and to $\boldsymbol{m}$ in the BDLOP part under the same randomness $\boldsymbol{s}_2$. We now explain the algorithms KeyGen, Commit and Open for this scheme.

ABDLOP.KeyGen: In order to create public parameters that can be used to commit to messages $(\boldsymbol{s}_1, \boldsymbol{m}) \in R_q^{m_1+\ell}$, we sample matrices $\boldsymbol{A}_1 \xleftarrow{\$} R_q^{n \times m_1}$, $\boldsymbol{A}_2 \xleftarrow{\$} R_q^{n \times m_2}$, $\boldsymbol{B} \xleftarrow{\$} R_q^{\ell \times m_2}$, $\boldsymbol{B}_{\mathsf{ext}} \xleftarrow{\$} R_q^{\ell_{\mathsf{ext}} \times m_2}$.

Here $\ell_{\mathsf{ext}}$ defines how many additional polynomials we can commit to later, under the same randomness. We also select parameters $\nu, \alpha$ that defines the message space $\mathcal{S}_M = \{\boldsymbol{s}_1 \in R_q^{m_1} \mid \|\boldsymbol{s}_1\| \leq \alpha\} \times R_q^{\ell}$, and the randomness distribution $\mathcal{D} = \chi^{m_2}$, where $\chi$ is the uniform distribution on $S_\nu$. Lastly we choose the bounds $B_1$ and $B_2$.

ABDLOP.Commit: In order to commit to a message $(\boldsymbol{s}_1, \boldsymbol{m}) \in \mathcal{S}_M$, we sample randomness $\boldsymbol{s}_2 \xleftarrow{\$} \chi^{m_2}$, and output the commitment

$$\mathsf{ABDLOP.Commit}(\boldsymbol{s}_1, \boldsymbol{m}; \boldsymbol{s}_2) := \begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{s}_2 + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \end{bmatrix} \in R_q^{n+\ell}.$$

ABDLOP.Open: For a commitment $\boldsymbol{t} \in R_q^{n+\ell}$, $((\boldsymbol{s}_1, \boldsymbol{m}), \boldsymbol{s}_2, c) \in \mathcal{S}_M \times R_q^{m_2} \times R_q$ is a valid opening if

- $\mathsf{ABDLOP.Commit}(\boldsymbol{s}_1, \boldsymbol{m}; \boldsymbol{s}_2) = \boldsymbol{t}$
- $c \in \overline{\mathcal{C}}$
- $\|c\boldsymbol{s}_2\| \leq B_2$
- $\|c\boldsymbol{s}_1\| \leq B_1$

$\mathsf{ABDLOP.Open}((\boldsymbol{s}_1, \boldsymbol{m}), \boldsymbol{s}_2, c, \boldsymbol{t})$ outputs 1 if $((\boldsymbol{s}_1, \boldsymbol{m}), \boldsymbol{s}_2, c)$ is a valid opening of $\boldsymbol{t}$, and 0 if not.

In some of the zero-knowledge protocols that will be constructed using this commitment scheme, there will be a need for creating additional commitments under the same randomness, as a part of the protocol. By using $\boldsymbol{B}_{\mathsf{ext}}$, we can create such additions in the BDLOP part of the commitment scheme. This can be done by computing $\boldsymbol{t}_{\mathsf{ext}} := \boldsymbol{B}_{\mathsf{ext}}\boldsymbol{s}_2 + \boldsymbol{m}_{\mathsf{ext}}$. Then $(\boldsymbol{t}_A, \boldsymbol{t}_B \| \boldsymbol{t}_{\mathsf{ext}})$ is a commitment to $(\boldsymbol{s}_1, \boldsymbol{m} \| \boldsymbol{m}_{\mathsf{ext}})$. We note that it is also possible to do this separately for several messages, by defining several such extension matrices $\boldsymbol{B}_{\mathsf{ext}}$.

### 4.2.1 Hiding and binding

We now prove that the ABDLOP commitment scheme satisfies the desired security properties.

**Theorem 7.** Suppose that $B_1 \geq \alpha$ and $B_2 \geq \nu\sqrt{m_2 N}$. Then the ABDLOP commitment scheme is complete.

*Proof.* Let $1 \in \overline{\mathcal{C}}$ and take any $(\boldsymbol{s}_1, \boldsymbol{m}) \in \mathcal{S}_M$. Then by the definition of $\mathcal{S}_M$ and by assumption, $\|1\boldsymbol{s}_1\| \leq \alpha \leq B_1$. We also have that for $\boldsymbol{s}_2 \xleftarrow{\$} \chi^{m_2}$, $\|\boldsymbol{s}_2\| \leq \nu\sqrt{m_2 N}$ since $\|\boldsymbol{s}_2\|_\infty \leq \nu$. Thus we also have that

$$\|1\boldsymbol{s}_2\| \leq \nu\sqrt{m_2 N} \leq B_2.$$

Hence $\mathsf{ABDLOP.Open}(\boldsymbol{s}_1, \boldsymbol{m}, \boldsymbol{s}_2, 1; \mathsf{ABDLOP.Commit}(\boldsymbol{s}_1, \boldsymbol{m}; \boldsymbol{s}_2)) = 1$, and the scheme is complete. □

We now prove the hiding property of the commitment scheme can be reduced to the MLWE problem, thus proving that the scheme is hiding.

**Theorem 8.** Suppose that $m_2 - n - \ell \geq 0$. Then, if there exists an algorithm $\mathcal{A}$ that has advantage $\varepsilon$ in breaking the hiding property of the ABDLOP commitment scheme, then there exists an algorithm $\mathcal{A}'$ that has advantage $\varepsilon$ in solving the $\mathsf{MLWE}_{m_2, n+\ell, \chi}$ problem.

*Proof.* Suppose that $\mathcal{A}'$ is given an $\mathsf{MLWE}_{m_2,n+\ell,\chi}$ instance $\begin{bmatrix}\boldsymbol{A}_2\\ \boldsymbol{B}\end{bmatrix} \in R_q^{(n+\ell)\times m_2}, \boldsymbol{b} \in R_q^{n+\ell}$. Now $\mathcal{A}'$ outputs public parameters $\boldsymbol{A}_1 \xleftarrow{\$} R_q^{n\times m_1}, \boldsymbol{A}_2, \boldsymbol{B}$. Upon receiving $(\boldsymbol{s}_{1,0}, \boldsymbol{m}_0), (\boldsymbol{s}_{1,1}, \boldsymbol{m}_1)$ from $\mathcal{A}$, $\mathcal{A}'$ samples $b \xleftarrow{\$} \{0,1\}$ sends the commitment of $(\boldsymbol{s}_{1,b}, \boldsymbol{m}_b)$

$$\begin{bmatrix}\boldsymbol{t}_A\\ \boldsymbol{t}_B\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{s}_{1,b} + \boldsymbol{b} + \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{m}_b\end{bmatrix}.$$

If $\mathcal{A}$ responds with $b' = b$, then $\mathcal{A}'$ outputs 1, and 0 if not. We now have two cases:

- If $\mathcal{A}'$ was given a randomly generated $\boldsymbol{b}$, then the output $\begin{bmatrix}\boldsymbol{t}_A\\ \boldsymbol{t}_B\end{bmatrix}$ is independent of $(\boldsymbol{s}_{1,b}, \boldsymbol{m}_b)$, and the probability that $\mathcal{A}$ outputs the correct $b'$ is exactly $1/2$.
- If $\mathcal{A}'$ was given $\boldsymbol{b} = \begin{bmatrix}\boldsymbol{A}_2\\ \boldsymbol{B}\end{bmatrix}\boldsymbol{s}_2$, for some $\boldsymbol{s}_2 \xleftarrow{\$} \chi^{m_2}$, then the output commitment is

  $$\begin{bmatrix}\boldsymbol{t}_A\\ \boldsymbol{t}_B\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{s}_1 + \begin{bmatrix}\boldsymbol{A}_2\\ \boldsymbol{B}\end{bmatrix}\boldsymbol{s}_2 + \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{m}\end{bmatrix} = \mathsf{ABDLOP.Commit}(\boldsymbol{s}_{1,b}, \boldsymbol{m}_b; \boldsymbol{s}_2)$$

  and by assumption $\mathcal{A}$ outputs the correct $b'$ with probability $1/2 + \varepsilon$.

Therefore $\mathcal{A}'$ will have advantage $\varepsilon$ in solving the $\mathsf{MLWE}_{m_2,n+\ell,\chi}$ problem. □

We now prove that the binding property of the commitment scheme can be reduced to the MSIS problem, thus proving that the scheme is binding.

**Theorem 9.** If there exists an algorithm $\mathcal{A}$ who can break the binding property of the ABDLOP commitment scheme with probability $\varepsilon$, then there exists an algorithm $\mathcal{A}'$ with advantage $\varepsilon$ in solving the $\mathsf{MSIS}_{n,m_1+m_2,B}$ problem, where $B = 4\eta\sqrt{B_1^2 + B_2^2}$.

*Proof.* Suppose that $\mathcal{A}'$ is given an $\mathsf{MSIS}_{n,m_1+m_2,B}$ instance $\begin{bmatrix}\boldsymbol{A}_1 & \boldsymbol{A}_2\end{bmatrix} \leftarrow R_q^{n\times(m_1+m_2)}$. $\mathcal{A}'$ now outputs $\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{B} \xleftarrow{\$} R_q^{\ell\times m_2}$ as public parameters, according to the commitment scheme. By assumption $\mathcal{A}$ is now, with probability $\varepsilon$, able to come up with a commitment $\boldsymbol{t} = \begin{bmatrix}\boldsymbol{t}_A\\ \boldsymbol{t}_B\end{bmatrix}$ with two valid openings $(\boldsymbol{s}_2, \boldsymbol{s}_1, \boldsymbol{m}, c), (\boldsymbol{s}_2', \boldsymbol{s}_1', \boldsymbol{m}', c')$, where $\boldsymbol{m} \neq \boldsymbol{m}'$. This implies that

$$\boldsymbol{A}_1\boldsymbol{s}_1 + \boldsymbol{A}_2\boldsymbol{s}_2 = \boldsymbol{t}_A = \boldsymbol{A}_1\boldsymbol{s}_1' + \boldsymbol{A}_2\boldsymbol{s}_2'.$$

We can now rearrange this equation to get that

$$\begin{bmatrix}\boldsymbol{A}_1 & \boldsymbol{A}_2\end{bmatrix}\begin{bmatrix}\boldsymbol{s}_1 - \boldsymbol{s}_1'\\ \boldsymbol{s}_2 - \boldsymbol{s}_2'\end{bmatrix} = \boldsymbol{0},$$

for a non-zero matrix $\begin{bmatrix}\boldsymbol{s}_1 - \boldsymbol{s}_1'\\ \boldsymbol{s}_2 - \boldsymbol{s}_2'\end{bmatrix}$. Thus, if the $\ell_2$ norm of $\begin{bmatrix}\boldsymbol{s}_1 - \boldsymbol{s}_1'\\ \boldsymbol{s}_2 - \boldsymbol{s}_2'\end{bmatrix}$ is bounded by $B$, we have a solution to the $\mathsf{MSIS}_{n,m_1+m_2,B}$ problem for $\begin{bmatrix}\boldsymbol{A}_1 & \boldsymbol{A}_2\end{bmatrix}$. We will use Lemma 12 in order to obtain the desired bound, so we start by using the triangle inequality to get

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

Since we have $c, c' \in \overline{\mathcal{C}}$, we must have that $c = c_0 - c_1$ and $c' = c'_0 - c'_1$ for some $c_0, c_1, c'_0, c'_1 \in \mathcal{C}$. By construction of the commitment scheme, we have that $\left\| \begin{bmatrix} c\boldsymbol{s}_1 \\ c\boldsymbol{s}_2 \end{bmatrix} \right\|, \left\| \begin{bmatrix} c'\boldsymbol{s}'_1 \\ c'\boldsymbol{s}'_2 \end{bmatrix} \right\| \leq \sqrt{B_1^2 + B_2^2}$, and by definition of the challenge space we have that $\sqrt[2k]{\|\sigma_{-1}(c^k)c^k\|_1} \leq \eta$ for all $c \in \mathcal{C}$. We now use Lemma 12 and obtain

$$\left\| c' \begin{bmatrix} c\boldsymbol{s}_1 \\ c\boldsymbol{s}_2 \end{bmatrix} \right\| \leq \left\| c'_0 \begin{bmatrix} c\boldsymbol{s}_1 \\ c\boldsymbol{s}_2 \end{bmatrix} \right\| + \left\| c'_1 \begin{bmatrix} c\boldsymbol{s}_1 \\ c\boldsymbol{s}_2 \end{bmatrix} \right\| \leq 2\eta\sqrt{B_1^2 + B_2^2}$$
$$\left\| c \begin{bmatrix} c'\boldsymbol{s}'_1 \\ c'\boldsymbol{s}'_2 \end{bmatrix} \right\| \leq \left\| c_0 \begin{bmatrix} c'\boldsymbol{s}'_1 \\ c'\boldsymbol{s}'_2 \end{bmatrix} \right\| + \left\| c_1 \begin{bmatrix} c'\boldsymbol{s}'_1 \\ c'\boldsymbol{s}'_2 \end{bmatrix} \right\| \leq 2\eta\sqrt{B_1^2 + B_2^2}$$

Hence we have that

$$\left\| cc' \begin{bmatrix} \boldsymbol{s}_1 - \boldsymbol{s}'_1 \\ \boldsymbol{s}_2 - \boldsymbol{s}'_2 \end{bmatrix} \right\| \leq 4\eta\sqrt{B_1^2 + B_2^2} = B,$$

and $\mathcal{A}'$ has advantage $\varepsilon$ in solving the $\mathsf{MSIS}_{n,m_1+m_2,B}$ problem, where $B = 4\eta\sqrt{B_1^2 + B_2^2}$. □

### 4.2.2 Opening proof

We need a proof of knowledge of a valid opening to a commitment also for this commitment scheme. The idea for this scheme is to perform rejection sampling on both the randomness $\boldsymbol{s}_2$, and on the message $\boldsymbol{s}_1$. This means that we draw two masking vectors $\boldsymbol{y}_1$ to mask $\boldsymbol{s}_1$ and $\boldsymbol{y}_2$ to mask $\boldsymbol{s}_2$, and perform rejection sampling on both. This is so that the prover can send $\boldsymbol{w} := \boldsymbol{A}_1\boldsymbol{y}_1 + \boldsymbol{A}_2\boldsymbol{y}_2$ to the verifier, whom can check if $\boldsymbol{A}_1\boldsymbol{z}_1 + \boldsymbol{A}_2\boldsymbol{z}_2 - c\boldsymbol{t}_A = \boldsymbol{w}$.

Since the commitment scheme and opening proof will mostly be used to prove that committed values satisfies some relation, and then never using the commitment again, we can then use the more efficient rejection sampling algorithm $\mathsf{Rej}_2$ on the randomness $\boldsymbol{s}_2$. This means that we will leak one bit of the randomness, but this will not be a problem. If we however want to use the protocol and not throw out the commitment, we can simply just use $\mathsf{Rej}_1$ on $\boldsymbol{s}_2$ as well.

We now explain how to set the standard deviations for this scheme. Since we know that $\|\boldsymbol{s}_1\| \leq \alpha$, we get by the definition of $\mathcal{C}$ and Lemma 12, that $\|c\boldsymbol{s}_1\| \leq \eta\alpha$ for $c \in \mathcal{C}$. Similarly, since $\|\boldsymbol{s}_2\|_\infty \leq \nu$, we get that $\|c\boldsymbol{s}_2\| \leq \eta\nu\sqrt{m_2N}$ for $c \in \mathcal{C}$. So, according to Lemma 4 we set the standard deviations $\mathfrak{s}_1 = \gamma_1\eta\alpha$ and $\mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2N}$ for some $\gamma_1, \gamma_2 > 0$. The opening proof is given as $\Pi'_{\mathsf{open}}$ in Figure 4.2.

**Theorem 10.** Suppose that $\mathfrak{s}_1 = \gamma_1\eta\alpha$ and $\mathfrak{s}_2 = \gamma_2\eta\nu\sqrt{m_2N}$ for some $\gamma_1, \gamma_2 > 0$. Then the protocol $\Pi'_{\mathsf{open}}$ is complete, meaning that if $m_1, m_2 > 640/N$ then the honest prover convinces the verifier with probability

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

*Proof.* By the definition of $\mathcal{C}$ and Lemma 12, and since $\|\boldsymbol{s}_1\| \leq \alpha, \|\boldsymbol{s}_2\| \leq \nu\sqrt{m_2N}$, we have the bounds

$$\|c\boldsymbol{s}_1\| \leq \eta\alpha, \quad \|c\boldsymbol{s}_1\| \leq \eta\nu\sqrt{m_2N}.$$

$\Pi'_{\mathsf{open}}$

Public Information: $\boldsymbol{A}_1 \in R_q^{n\times m_1}$, $\boldsymbol{A}_2 \in R_q^{n\times m_2}$, $\boldsymbol{B} \in R_q^{\ell\times m_2}$.
Prover's Information: $(\boldsymbol{s}_1, \boldsymbol{m}, \boldsymbol{s}_2) \in R_q^{m_1+\ell+m_2}$ so that $\|\boldsymbol{s}_1\| \le \alpha$ and $\|\boldsymbol{s}_2\|_\infty \le \nu$
Commitment: $\boldsymbol{t} = \begin{bmatrix}\boldsymbol{t}_A\\ \boldsymbol{t}_B\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{s}_1 + \begin{bmatrix}\boldsymbol{A}_2\\ \boldsymbol{B}\end{bmatrix}\boldsymbol{s}_2 + \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{m}\end{bmatrix}$

| Prover | | Verifier |
|---|---|---|
| $\boldsymbol{y}_1 \xleftarrow{\$} D_{R^{m_1},\mathfrak{s}_1}$ | | |
| $\boldsymbol{y}_2 \xleftarrow{\$} D_{R^{m_2},\mathfrak{s}_2}$ | | |
| $\boldsymbol{w} := \boldsymbol{A}_1\boldsymbol{y}_1 + \boldsymbol{A}_2\boldsymbol{y}_2$ | | |
| | $\xrightarrow{\boldsymbol{w}}$ | |
| | | $c \xleftarrow{\$} \mathcal{C}$ |
| | $\xleftarrow{c}$ | |
| $\boldsymbol{z}_1 = c\boldsymbol{s}_1 + \boldsymbol{y}_1$ | | |
| $\boldsymbol{z}_2 = c\boldsymbol{s}_2 + \boldsymbol{y}_2$ | | |
| for $i = 1, 2$ : | | |
| if $\mathsf{Rej}_i(\boldsymbol{z}_i, c\boldsymbol{s}_i, \mathfrak{s}_i) = 1$ | | |
| then abort | | |
| | $\xrightarrow{\boldsymbol{z}_1, \boldsymbol{z}_2}$ | |
| | | Accept iff: |
| | | 1. $\|\boldsymbol{z}_1\| \le \mathfrak{s}_1\sqrt{2m_1N}$ |
| | | 2. $\|\boldsymbol{z}_2\| \le \mathfrak{s}_2\sqrt{2m_2N}$ |
| | | 3. $\boldsymbol{A}_1\boldsymbol{z}_1 + \boldsymbol{A}_2\boldsymbol{z}_2 - c\boldsymbol{t}_A = \boldsymbol{w}$ |

Figure 4.2: Proof of knowledge $\Pi'_{\mathsf{open}}$ of $(\boldsymbol{s}_1, \boldsymbol{s}_2, \overline{c}) \in R_q^{m_1} \times R_q^{m_2} \times \overline{\mathcal{C}}$ such that $\boldsymbol{A}_1\boldsymbol{s}_1 + \boldsymbol{A}_2\boldsymbol{s}_2 = \boldsymbol{t}_A$, $\boldsymbol{B}\boldsymbol{s}_2 + \boldsymbol{m} = \boldsymbol{t}_B$ and $\|\overline{c}\boldsymbol{s}_i\| \le 2\mathfrak{s}_i\sqrt{2m_iN}$ for $i = 1, 2$.

We can use Lemma 4 to see that the probability that both $\mathsf{Rej}_1(\boldsymbol{z}_1, c\boldsymbol{s}_1, \mathfrak{s}_1)$ and $\mathsf{Rej}_2(\boldsymbol{z}_2, c\boldsymbol{s}_2, \mathfrak{s}_2)$ do not abort is at least

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

We can now use the tail-bound in Lemma 3 with $\delta = \sqrt{2}$ to see that $\|\boldsymbol{z}_1\| \le \mathfrak{s}_1\sqrt{2m_1N}$, except with probability $\sqrt{2}^{m_1N}\exp(-m_1N/2)$, and that $\|\boldsymbol{z}_2\| \le \mathfrak{s}_2\sqrt{2m_2N}$, except with probability $\sqrt{2}^{m_2N}\exp(-m_2N/2)$. Under the assumption that $m_1, m_2 \ge 640/N$, these probabilities are negligible. The last verification equation follows from

$$\begin{aligned}\boldsymbol{A}_1\boldsymbol{z}_1 + \boldsymbol{A}_2\boldsymbol{z}_2 &= \boldsymbol{A}_1\boldsymbol{y}_1 + \boldsymbol{A}_2\boldsymbol{y}_2 + c\boldsymbol{A}_1\boldsymbol{s}_1 + c\boldsymbol{A}_2\boldsymbol{s}_2\\ &= \boldsymbol{w} + c\boldsymbol{t}_A,\end{aligned}$$

and hence the Theorem holds. □

**Theorem 11.** Suppose that $\mathfrak{s}_1 = \gamma_1\eta\alpha$ and $\mathfrak{s}_2 = \gamma_2\nu\eta\sqrt{m_2 N}$ for some $\gamma_1, \gamma_2 > 0$. Then the protocol $\Pi'_{\mathsf{open}}$ is honest-verifier zero-knowledge, meaning that there exists a simulator $\mathcal{S}$ that outputs a simulation of a non-aborting transcript of the protocol between the honest prover and verifier, that is indistinguishable from the real transcript.

*Proof.* According to Lemma 4, the $\boldsymbol{z}_i$'s in the real protocol is within statistical distance of $2^{-128}$ from $D_{R^{m_i},\mathfrak{s}_i}$, and independent of $c$. So we can construct the simulator $\mathcal{S}$ in the following manner. It samples $\boldsymbol{z}_1 \xleftarrow{\$} D_{R^{m_1},\mathfrak{s}_1}$, $\boldsymbol{z}_2 \xleftarrow{\$} D_{R^{m_2},\mathfrak{s}_2}$ and $c \xleftarrow{\$} \mathcal{C}$. It then computes $\boldsymbol{w} := \boldsymbol{A}_1\boldsymbol{z}_1 + \boldsymbol{A}_2\boldsymbol{z}_2 - c\boldsymbol{t}_A$. Finally it outputs the simulated transcript $(\boldsymbol{w}, c, \boldsymbol{z}_1, \boldsymbol{z}_2)$.

From construction of $\boldsymbol{w}$ we clearly get that $\boldsymbol{A}_1\boldsymbol{z}_1 + \boldsymbol{A}_2\boldsymbol{z}_2 - c\boldsymbol{t}_A = \boldsymbol{w}$. We can now use the tail-bound in Lemma 3 with $\delta = \sqrt{2}$ to see that $\|\boldsymbol{z}_1\| \leq \mathfrak{s}_1\sqrt{2m_1 N}$ and $\|\boldsymbol{z}_2\| \leq \mathfrak{s}_2\sqrt{2m_2 N}$ with overwhelming probability, by the same argumentation as in the proof of Theorem 10. Hence simulated transcript is statistically close to a real non-aborted transcript, and $\Pi'_{\mathsf{open}}$ is honest-verifier zero-knowledge. □

**Theorem 12.** Suppose that $B_1 \geq 2\mathfrak{s}_1\sqrt{2m_1 N}$ and $B_2 \geq 2\mathfrak{s}_2\sqrt{2m_2 N}$. Then the protocol $\Pi'_{\mathsf{open}}$ is knowledge sound, meaning that there is an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a probabilistic prover $\mathcal{P}^*$, which runs in time at most $T$ and convinces $\mathcal{V}$ with probability $\varepsilon > 1/|\mathcal{C}|$, the extractor $\mathcal{E}$ with probability at least $\varepsilon - 1/|\mathcal{C}|$ outputs $(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{s}}_2, \overline{\boldsymbol{m}}) \in R_q^{m_1+m_2+\ell}$ and $\overline{c} \in \overline{\mathcal{C}}$ such that for $\boldsymbol{t} = \mathsf{ABDLOP.Commit}(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{\boldsymbol{s}}_2)$, we have $\mathsf{ABDLOP.Open}(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{\boldsymbol{s}}_2, \overline{c}, \boldsymbol{t}) = 1$.

*Proof.* In order to prove knowledge soundness we use the collision game strategy introduced in Section 2.3.4. We let $H \in \{0,1\}^{R\times|\mathcal{C}|}$ be the binary matrix where the $R$ rows correspond to the prover's randomness and $|\mathcal{C}|$ columns correspond to the different choices for the challenge $c$, and let $H(r, c)$ denote the entry corresponding to randomness $r$ and challenge $c \in \mathcal{C}$. By assumption there are $\varepsilon$ 1-entries in $H$. We define the following extractor:

1. $\mathcal{E}$ first samples fresh randomness $r$ and challenge $c^{(0)} \xleftarrow{\$} \mathcal{C}$. Then it checks if $H(r, c^{(0)}) = 1$, and aborts if not.
2. Otherwise, $\mathcal{E}$ samples along row $r$ without replacement until it finds a $c^{(1)}$ such that $H(r, c^{(0)}) = H(r, c^{(1)}) = 1$ and $c^{(0)}, c^{(1)}$ are distinct.

If we assume that $\mathcal{E}$ can check values of each entry in $H$ in time most $T$, then Lemma 1 states that the expected time of $\mathcal{E}$ is at most $2T$ and that $\mathcal{E}$ extracts two valid transcripts with probability at least $\varepsilon - 1/|\mathcal{C}|$. We denote the valid transcripts as

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

We define $\overline{c} = c^{(1)} - c^{(0)} \in \overline{\mathcal{C}}$, which by definition of the challenge space is invertible and such that $\|\overline{c}\|_\infty \leq 2\omega$. We can now define

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

Since the transcripts are valid, we have that $\boldsymbol{A}_1\boldsymbol{z}_1^{(i)} + \boldsymbol{A}_2\boldsymbol{z}_2^{(i)} = \boldsymbol{w} + c^{(i)}\boldsymbol{t}_A$ for $i = 0, 1$, and thus we get

$$\begin{aligned}\boldsymbol{A}_1\overline{\boldsymbol{s}}_1 + \boldsymbol{A}_2\overline{\boldsymbol{s}}_2 &= \boldsymbol{A}_1(\boldsymbol{z}_1^{(1)}/\overline{c} - \boldsymbol{z}_1^{(0)}/\overline{c}) + \boldsymbol{A}_2(\boldsymbol{z}_2^{(1)}/\overline{c} - \boldsymbol{z}_2^{(0)}/\overline{c}) \\ &= c^{(1)}\boldsymbol{t}_A/\overline{c} - c^{(0)}\boldsymbol{t}_A/\overline{c} \\ &= \boldsymbol{t}_A.\end{aligned}$$

And since $\boldsymbol{t}_B = \boldsymbol{B}\overline{\boldsymbol{s}}_2 + \overline{\boldsymbol{m}}$ by construction, we thus have that $\mathsf{ABDLOP.Commit}(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{\boldsymbol{s}}_2) = \boldsymbol{t}$.

By construction and Lemma 3 we have $\|\overline{c}\overline{\boldsymbol{s}}_1\| \leq 2\mathfrak{s}_1\sqrt{2m_1N} \leq B_1$, $\|\overline{c}\overline{\boldsymbol{s}}_2\| \leq 2\mathfrak{s}_2\sqrt{2m_2N} \leq B_2$. Hence we will get that $\mathsf{ABDLOP.Open}(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{\boldsymbol{s}}_2, \overline{c}, \boldsymbol{t}) = 1$. □

# Chapter 5

# Zero-knowledge schemes using BDLOP commitments

Shortly after the introduction of the BDLOP commitment scheme, it was put into use to construct zero-knowledge proofs of knowledge of a short vector $\vec{s}$ satisfying

$$A\vec{s} = \vec{u} \text{ over } \mathbb{Z}_q, \tag{5.1}$$

for public $A \in \mathbb{Z}_q^{m \times n}$ and $\vec{u} \in \mathbb{Z}_q^m$. This is in fact a more general statement, but if we let the matrix $A$ have a certain structure describing linear relations over $R_q$, then this is equivalent to $\boldsymbol{As} = \boldsymbol{u}$ over $R_q$, for a matrix $\boldsymbol{A}$ and vectors $\boldsymbol{s}, \boldsymbol{u}$ over $R_q$. Schemes for proving (5.1) will consist of a proof of knowledge of such an $\vec{s}$ and a proof showing that $\vec{s}$ is short.

In this chapter we will explain some of the main constructions that led to the state of the art lattice-based zero-knowledge proofs. We start in Section 5.1 by explaining one of the very first zero-knowledge schemes that used the BDLOP commitment scheme. This scheme uses a product proof of committed values to prove that the coefficients of $\vec{s}$ are in $\{0, 1, 2\}$. In the next Section 5.2, we explain a more efficient product proof for committed values. Lastly, in Section 5.3 we explain a more efficient method for proving knowledge of an $\vec{s}$ satisfying (5.1). These can be used in the first scheme to make it more efficient.

The state of the art schemes by Lyubashevsky et al. [16] uses ideas from all the schemes presented in this chapter. We omit the security analysis of the schemes in this chapter, because we want to focus on the ideas and how they build upon each other, rather than the details of the security. However for completeness, the full security analysis is given in the appendix.

## 5.1 Zero-knowledge using commitments and NTT coefficients

Bootle et. al. proposed in [6] the first lattice based proof system that significantly outperformed the Stern type proofs for proving knowledge of a short $\vec{s}$ satisfying $A\vec{s} = \vec{u}$ over $\mathbb{Z}_q$. The idea behind their protocol is to choose $q$ such that $q \equiv 1 \pmod{2N}$, which by Lemma 2 implies that $X^N + 1$ splits into linear factors modulo $q$. The NTT of a polynomial in $R_q$ will then be a vector of dimension $N$. We denote by 0, 1 and 2 the elements in $R_q$ that are the inverse NTT of the $N$-dimensional vectors of all 0's, 1's and 2's, respectively.

We notice that showing that a vector $\vec{s}$ has coefficients in $\{0,1,2\}$, is equivalent to showing that $\vec{s} \circ (\vec{s}-\vec{1}) \circ (\vec{s}-\vec{2}) = \vec{0}$. Thus we see that proving knowledge of a vector $\vec{s}$ with coefficients in $\{0,1,2\}$ that satisfies $A\vec{s} = \vec{u}$ over $\mathbb{Z}_q$, is equivalent to proving knowledge of a polynomial $s \in R_q$, possibly with large coefficients, such that

$$s(s-1)(s-2) = 0 \text{ and } A\hat{s} = \vec{u} \text{ over } \mathbb{Z}_q, \tag{5.2}$$

because $s(s-1)(s-2) = 0$ if and only if $\hat{s} \circ (\hat{s}-\hat{1}) \circ (\hat{s}-\hat{2}) = \hat{0}$.

We now explain how the protocol for proving knowledge of such an $s \in R_q$ satisfying (5.2) is constructed. The prover starts by sampling a masking vector $y \xleftarrow{\$} R_q$ and sends $\vec{w} = A\hat{y} \mod q$ to the verifier. Upon receiving a challenge $c \in \mathbb{Z}_q \subset R$ from the verifier, it outputs $z = y + cs$. Since $c$ is an integer, this is equivalent to $\hat{z} = \hat{y} + c\hat{s}$. The verifier can now check that

$$A\hat{z} = \vec{w} + c\vec{u}.$$

However, if one rewinds and obtains for a challenge $c' \neq c$ another equation, $A\hat{z}' = \vec{w} + c'\vec{u}$, we get that

$$A(\hat{z} - \hat{z}') = (c - c')\vec{u}. \tag{5.3}$$

There are two problems with this in order for it to prove (5.2). Firstly, we don't know that $\hat{z} - \hat{z}'$ has coefficients in $\{0,1,2\}$, and secondly $c - c'$ is not necessarily equal to 1. The solution to both of these problems involves committing to $y$ and $s$. We note that since $s$ might actually have large coefficients, we cannot use the Ajtai commitment scheme, and so we use version 1 of the BDLOP commitment scheme.

To solve the second problem, the prover in the first step also makes the commitments $\boldsymbol{t}_1 = \boldsymbol{b}_1\boldsymbol{r} + y$ and $\boldsymbol{t}_2 = \boldsymbol{b}_2\boldsymbol{r} + s$ to $y$ and $s$. We have for any challenge $c \in R$ that

$$\boldsymbol{t}_1 + c\boldsymbol{t}_2 = (\boldsymbol{b}_1 + c\boldsymbol{b}_2)\boldsymbol{r} + y + cs,$$

is a commitment to the message $y + cs$ with vector $\boldsymbol{b}_1 + c\boldsymbol{b}_2$. So upon receiving a challenge $c \in \mathbb{Z}_q$ from the verifier, the prover now proves that $\boldsymbol{t}_1 + c\boldsymbol{t}_2$ is a commitment to $z$. This implies that $z = y + cs$, because of the binding property of the commitment scheme. Hence the rewinding argument will now yield another equation $z' = y + c's$, and we get that $z - z' = (c - c')s$. This implies that

$$\hat{z} - \hat{z}' = (c - c')\hat{s}.$$

Plugging this into Equation (5.3) gives $A(c - c')\hat{s} = (c - c')\vec{u}$. Since $c, c' \in \mathbb{Z}_q$ and $c \neq c'$, we can divide out $c - c'$, and this then implies that

$$A\hat{s} = \vec{u},$$

as desired. The first problem is now reduced to showing that the coefficients of $\hat{s}$ are in $\{0,1,2\}$. The initial idea for this proof stems from the observation that

$$\begin{aligned} z(z-c)(z-2c) &= (y + cs)(y + c(s-1))(y + c(s-2)) \\ &= y^3 + 3y^2(s-1)c + y(3s^2 - 6s + 2)c^2 + s(s-1)(s-2)c^3. \end{aligned}$$

We notice that the last coefficient of the above polynomial in $c$ is exactly what we want to prove equals 0. So, if the prover in the first step also makes the commitments

$$\begin{aligned}\boldsymbol{t}_3 &= \boldsymbol{b}_3\boldsymbol{r} + y^3,\\ \boldsymbol{t}_4 &= \boldsymbol{b}_4\boldsymbol{r} + 3y^2(s-1),\\ \boldsymbol{t}_5 &= \boldsymbol{b}_5\boldsymbol{r} + y(3s^2-6s+2),\end{aligned}$$

it can upon receiving the challenge $c$, prove that $\boldsymbol{t}_3 + c\boldsymbol{t}_4 + c^2\boldsymbol{t}_5$ is indeed a commitment to $z(z-c)(z-2c)$. This would imply that the above polynomial is quadratic in $c$, and therefore that $s(s-1)(s-2) = 0$. However, this argument can be optimized in such a way that the prover can commit to one less polynomial. If we instead consider the expression

$$\begin{aligned}(z-c)(z-2c)s &= ((y+cs)^2 - 3c(y+cs) + 2c^2)s\\ &= y^2s + (2ys^2 - 3ys)c + (s^3 - 3s^2 + 2s)c^2\\ &= y^2s + (zy - y^2)(2s-3) + s(s-1)(s-2)c^2\\ &= zy(2s-3) - y^2(s-3) + s(s-1)(s-2)c^2,\end{aligned}$$

where we used that $cys = (z-y)y = zy - y^2$, we notice that the prover can instead make the commitments

$$\begin{aligned}\boldsymbol{t}_3 &= \boldsymbol{b}_3\boldsymbol{r} + y(2s-3)\\ \boldsymbol{t}_4 &= \boldsymbol{b}_4\boldsymbol{r} + y^2(s-3)\end{aligned}$$

and prove that $(z-c)(z-2c)\boldsymbol{t}_2 - z\boldsymbol{t}_3 + \boldsymbol{t}_4$ is a commitment to 0, to imply that $s(s-1)(s-2) = 0$.

We notice that the commitments are constructed as a part of the protocol, and we will sample the coefficients of the randomness vector $\boldsymbol{r}$ from the distribution $\chi$ on $\{-1, 0, 1\}$, where $\pm 1$ have probability 5/16, and 0 has probability 6/16. We will use the protocol $\Pi_{\mathsf{open}}$ to prove that all the commitments are valid, by drawing a $\boldsymbol{y}' \xleftarrow{\$} D_{R^6,\mathfrak{s}}$, computing $\boldsymbol{w}' = \boldsymbol{b}_0\boldsymbol{y}'$, and upon receiving a challenge $f \xleftarrow{\$} \mathcal{C}$ from the verifier, output $\boldsymbol{z}' = \boldsymbol{y}' + f\boldsymbol{r}$ if the rejection sampling algorithm on $\boldsymbol{z}'$ does not abort. The verifier can then check if $\boldsymbol{z}'$ is small enough and if $\boldsymbol{b}_0\boldsymbol{z}' = \boldsymbol{w}' + f\boldsymbol{t}_0$.

What remains now is to explain how we prove that $\boldsymbol{t}_1 + c\boldsymbol{t}_2$ and $(z-c)(z-2c)\boldsymbol{t}_2 - z\boldsymbol{t}_3 + \boldsymbol{t}_4$ are commitments to $z$ and 0, respectively. This is done by the prover computing

$$\boldsymbol{x}_1 = (\boldsymbol{b}_1 + c\boldsymbol{b}_2)\boldsymbol{y}' \tag{5.4}$$

$$\boldsymbol{x}_2 = ((z-c)(z-2c)\boldsymbol{b}_2 - z\boldsymbol{b}_3 + \boldsymbol{b}_4)\boldsymbol{y}'. \tag{5.5}$$

The verifier can then check if

$$(\boldsymbol{b}_1 + c\boldsymbol{b}_2)\boldsymbol{z}' + fz \stackrel{?}{=} \boldsymbol{x}_1 + f(\boldsymbol{t}_1 + c\boldsymbol{t}_2), \tag{5.6}$$

$$((z-c)(z-2c)\boldsymbol{b}_2 - z\boldsymbol{b}_3 + \boldsymbol{b}_4)\boldsymbol{z}' \stackrel{?}{=} \boldsymbol{x}_2 + f((z-c)(z-2c)\boldsymbol{t}_2 - z\boldsymbol{t}_3 + \boldsymbol{t}_4), \tag{5.7}$$

which would only hold if the commitments actually are to $z$ and 0.

Since $q$ is chosen such that $X^N + 1$ splits into linear factors modulo $q$, the largest challenge space that can be used for the commitment opening proof is

$$\mathcal{C} = \{0, X^i \mid 0 \leq i < 2N\}.$$

It is guaranteed by Lemma 10 that all elements in the set of differences $\overline{\mathcal{C}}$ are invertible. We therefore use the standard deviation $\mathfrak{s} = \gamma\sqrt{6N}$, as explained in Section 4.1, since $\kappa = 1$ for this challenge space, and $\beta = 1$, $k = 6$ for the randomness space.

This means that the largest challenge space one can use for the commitment validity proof is of size $2N + 1$, and that the largest challenge space one can use for proving knowledge of $s$ is of size $q$. Hence the commitment validity proof would need to be repeated $128/\log(2N)$ times order achieve 128-bit security, which increases the total proof size.

$\underline{\Pi_{\mathsf{NTT}}}$

Public Information: $A \in \mathbb{Z}_q^{m \times N}, \vec{u} = A\hat{\boldsymbol{s}} \in \mathbb{Z}_q^m, \boldsymbol{b}_0, \ldots, \boldsymbol{b}_4 \in R_q^6$
Prover's Information: $\hat{\boldsymbol{s}} \in \{0,1,2\}^N$

| Prover | | Verifier |
|---|---|---|
| $y \xleftarrow{\$} R_q$ | | |
| $\boldsymbol{r} \xleftarrow{\$} \chi^{6N}$ | | |
| $\boldsymbol{t} = \begin{bmatrix} \boldsymbol{t}_0 \\ \boldsymbol{t}_1 \\ \boldsymbol{t}_2 \\ \boldsymbol{t}_3 \\ \boldsymbol{t}_4 \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_0 \\ \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \boldsymbol{b}_3 \\ \boldsymbol{b}_4 \end{bmatrix} \boldsymbol{r} + \begin{bmatrix} 0 \\ y \\ s \\ y(2s-3) \\ y^2(s-3) \end{bmatrix}$ | | |
| $\vec{w} = A\hat{y}$ | $\xrightarrow{\boldsymbol{t}, \vec{w}}$ | |
| | $\xleftarrow{c}$ | $c \xleftarrow{\$} \mathbb{Z}_q$ |
| $z = y + cs$ | | |
| $\boldsymbol{y}' \xleftarrow{\$} D_{R^6,\mathfrak{s}}$ | | |
| $\boldsymbol{w}' = \boldsymbol{b}_0\boldsymbol{y}'$ | | |
| Define $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ as in (5.4) and (5.5) | | |
| | $\xrightarrow{z, \boldsymbol{w}', \boldsymbol{x}_1, \boldsymbol{x}_2}$ | |
| | $\xleftarrow{f}$ | $f \xleftarrow{\$} \mathcal{C}$ |
| $\boldsymbol{z}' = \boldsymbol{y}' + f\boldsymbol{r}$ | | |
| If $\mathsf{Rej}_1(\boldsymbol{z}', f\boldsymbol{r}, \mathfrak{s}) = 1$, abort | $\xrightarrow{\boldsymbol{z}'}$ | |
| | | Accept iff:<br>1. $\|\boldsymbol{z}'\| \leq B = \mathfrak{s}\sqrt{12n}$<br>2. $A\hat{z} \stackrel{?}{=} \vec{w} + c\vec{u}$<br>3. $\boldsymbol{b}_0\boldsymbol{z}' \stackrel{?}{=} \boldsymbol{w}' + f\boldsymbol{t}_0$<br>4. (5.6) and (5.7) holds |

Figure 5.1: Proof of knowledge $\Pi_{\mathsf{NTT}}$ of $\vec{s} \in \mathbb{Z}_q^N$ with coefficients in $\{0,1,2\}$ satisfying $A\vec{s} = \vec{u}$ over $\mathbb{Z}_q$.

The protocol is given as $\Pi_{\mathsf{NTT}}$ in Figure 5.1. We give the full security analysis of the protocol in Appendix A.

## 5.2 Automorphism opening and product proof

One key observation to get around the problem with small challenge space that we encountered in the previous section, is that we don't always need $\overline{c} \in \overline{\mathcal{C}}$ to be invertible. It is actually sufficient that the probability that $\overline{c}$ is not invertible, is less than the targeted soundness error. If we then use the fact that an element in $R_q$ is invertible if and only if all its $\mathsf{NTT}$ coefficients are non-zero, we notice that in order to determine the probability that $\overline{c} \in \overline{\mathcal{C}}$ is invertible, one can just determine the probability that a random $c \in \mathcal{C}$ hits a given $\mathsf{NTT}$ coefficient. We now explain how we can construct challenge spaces for which we can compute this probability.

We start by choosing parameters according to Lemma 2 such that

$$R_q = \mathbb{Z}_q[X]/(X^N + 1) \cong \prod_{i \in \mathbb{Z}_{2k}^\times} \mathbb{Z}_q[X]/(X^{N/k} - \zeta^i),$$

where $\zeta \in \mathbb{Z}_q$ is a $2k$-th root of unity. We can now define the challenge space to be all polynomials of degree $N$ with coefficients in $\{-1, 0, 1\}$. We let $C$ be the distribution over $\mathcal{C} = \{-1, 0, 1\}^N \subset R_q$ such that the coefficients of a challenge $c \in \mathcal{C}$ are independently identically distributed, with 0 having probability $p$ and $\pm 1$ having probability $(1 - p)/2$ each.

For a challenge $c \xleftarrow{\$} C$ and $i \in \mathbb{Z}_{2k}^\times$, we can now study the distribution of $c \mod (X^{N/k} - \zeta^i)$. The following Lemma, that is proved by Attema et al. in [3], states that this distribution is independent of $i$.

**Lemma 13** ([3])**.** Let $a \in R_q$ be a random polynomial with coefficients independently and identically distributed. Then $R_q/(X^{N/k} - \zeta^i) \cong R_q/(X^{N/k} - \zeta^j)$, and $a \mod (X^{N/k} - \zeta^i)$ and $a \mod (X^{N/k} - \zeta^j)$ are identically distributed for all $i, j \in \mathbb{Z}_{2k}^\times$.

Hence we can focus on the case where $i = 1$. It is obvious that for $c \xleftarrow{\$} C$, all coefficients follow the same distribution over $\mathbb{Z}_q$. Attema et al. also proves the following upper bound on the probability of the coefficients.

**Lemma 14** ([3])**.** Let the random variable $Y$ over $\mathbb{Z}_q$ be defined as above. Then for all $x \in \mathbb{Z}_q$,

$$\Pr[Y = x] \leq M := \frac{1}{q} + \frac{2k}{q} \sum_{j \in \mathbb{Z}_q^\times / \langle \zeta \rangle} \prod_{l=0}^{k-1} |p + (1 - p)\cos(2\pi j y \zeta^l / q)|$$

### 5.2.1 Opening proof

Attema et al. [3] explains how a new opening proof for commitments can be constructed with this challenge space, using the automorphism from Section 3.5. Suppose that the prover knows an opening to the commitment

$$\begin{aligned} \boldsymbol{t}_0 &= \boldsymbol{B}_0 \boldsymbol{r}, \\ \boldsymbol{t}_1 &= \langle \boldsymbol{b}_1, \boldsymbol{r} \rangle + m. \end{aligned}$$

for $\boldsymbol{B}_0 \in R_q^{\mu\times(\lambda+\mu+1)}$, $\boldsymbol{b}_1 \in R_q^{\lambda+\mu+1}$ and $\boldsymbol{r} \in R_q^{\lambda+\mu+1}$. The idea is the same as in $\Pi_{\mathsf{open}}$, but the initial problem is that when we no longer require $c - c'$ to be invertible, we can no longer use the prover replies $\boldsymbol{z}, \boldsymbol{z}'$ for different challenges $c, c'$ to obtain $\boldsymbol{y}_\mathsf{e}$ and $\boldsymbol{r}_\mathsf{e}$ such that

$$\boldsymbol{z} = \boldsymbol{y}_\mathsf{e} + c\boldsymbol{r}_\mathsf{e} \text{ and } \boldsymbol{z}' = \boldsymbol{y}_\mathsf{e} + c'\boldsymbol{r}_\mathsf{e}.$$

Here we by subscript $\mathsf{e}$ mean to indicate extracted values. What we can do instead is to make sure that for every $i \in \mathbb{Z}_{2k}^\times$, we have an accepting transcript pair with a challenge difference that is non-zero modulo $X^{N/k} - \zeta^i$. So suppose one can rewind and obtain $k$ pairs of prover replies $(\boldsymbol{z}_i, \boldsymbol{z}_i')$ for challenge pairs $(c_i, c_i')$, that satisfy

$$\overline{c_i} = c_i - c_i' \not\equiv 0 \pmod{(X^{N/k} - \zeta^i)}.$$

Then, if we have that all these transcripts have the same prover commitment $\boldsymbol{w}$ and are accepting, meaning that $\boldsymbol{B}_0\boldsymbol{z}_i = \boldsymbol{w} + c_i\boldsymbol{t}_0$ and $\boldsymbol{B}_0\boldsymbol{z}_i' = \boldsymbol{w} + c_i'\boldsymbol{t}_0$ for all $i$, we can compute

$$\boldsymbol{z}_i \equiv (\boldsymbol{y}_\mathsf{e})_i + c_i(\boldsymbol{r}_\mathsf{e})_i \text{ and } \boldsymbol{z}_i' \equiv (\boldsymbol{y}_\mathsf{e})_i + c_i'(\boldsymbol{r}_\mathsf{e})_i \pmod{(X^{N/k} - \zeta^i)}.$$

From this we can define

$$\begin{aligned}(\boldsymbol{r}_\mathsf{e})_i &= \frac{\boldsymbol{z}_i - \boldsymbol{z}_i'}{\overline{c_i}} \pmod{(X^{N/k} - \zeta^i)}, \text{ and}\\ (\boldsymbol{y}_\mathsf{e})_i &= \frac{c_i\boldsymbol{z}_i' - c_i'\boldsymbol{z}_i}{\overline{c_i}} \pmod{(X^{N/k} - \zeta^i)}.\end{aligned}$$

If we let $\boldsymbol{r}_\mathsf{e}$ and $\boldsymbol{y}_\mathsf{e}$ over $R_q$ be the inverse $\mathsf{NTT}$ of the $(\boldsymbol{r}_\mathsf{e})_i$ and $(\boldsymbol{y}_\mathsf{e})_i$, meaning that $\boldsymbol{r}_\mathsf{e} \equiv (\boldsymbol{r}_\mathsf{e})_i \mod (X^{N/k} - \zeta^i)$ and $\boldsymbol{y}_\mathsf{e} \equiv (\boldsymbol{y}_\mathsf{e})_i \mod (X^{N/k} - \zeta^i)$ for all $i \in \mathbb{Z}_{2k}^\times$, then it must hold that

$$\boldsymbol{z}_i = \boldsymbol{y}_\mathsf{e} + c_i\boldsymbol{r}_\mathsf{e} \text{ and } \boldsymbol{z}_i' = \boldsymbol{y}_\mathsf{e} + c_i'\boldsymbol{r}_\mathsf{e}$$

for all $i$, and that $\boldsymbol{B}_0\boldsymbol{r}_\mathsf{e} = \boldsymbol{t}_0$ and $\boldsymbol{B}_0\boldsymbol{y}_\mathsf{e} = \boldsymbol{w}$. We prove this in the following Lemma.

**Lemma 15.** If we have obtained $k$ pairs of accepting transcripts with commitment $\boldsymbol{w}$ as in the preceding paragraph, then every accepting transcript $(\boldsymbol{w}, c, \boldsymbol{z})$ must be such that $\boldsymbol{z} = \boldsymbol{y}_\mathsf{e} + c\boldsymbol{r}_\mathsf{e}$ where $\boldsymbol{y}_\mathsf{e}$ and $\boldsymbol{r}_\mathsf{e}$ are the vectors computed above independently from $c$, or we obtain an $\mathsf{MSIS}_{\mu,\lambda+\mu+1,8\kappa B}$ solution for $\boldsymbol{B}_0$ where $\kappa$ is a bound on the $\ell_1$ norm of the challenges. Moreover, we have $\boldsymbol{B}_0\boldsymbol{r}_\mathsf{e} = \boldsymbol{t}_0$ and $\boldsymbol{B}_0\boldsymbol{y}_\mathsf{e} = \boldsymbol{w}$.

*Proof.* We define $\boldsymbol{y}_\mathsf{e}'$ by $\boldsymbol{z} = \boldsymbol{y}_\mathsf{e}' + c\boldsymbol{r}_\mathsf{e}$ and fix some $i \in \{1, \ldots, k\}$. From the verification equations we get that

$$\boldsymbol{B}_0(\boldsymbol{z}_i - \boldsymbol{z}_i') = \overline{c}_i\boldsymbol{t}_0 \text{ and } \boldsymbol{B}_0(\boldsymbol{z} - \boldsymbol{z}_i) = (c - c_i)\boldsymbol{t}_0$$

since all transcripts are accepting. This implies that $(c - c_i)\boldsymbol{B}_0(\boldsymbol{z}_i - \boldsymbol{z}_i') = \overline{c}_i\boldsymbol{B}_0(\boldsymbol{z} - \boldsymbol{z}_i)$. Since we have that $\|\overline{c}_i\|_1 \leq 2\kappa$ and $\|\overline{c}_i\boldsymbol{z}\| \leq 2B$, we get that $\|((c - c_i)(\boldsymbol{z}_i - \boldsymbol{z}_i') - \overline{c}_i(\boldsymbol{z} - \boldsymbol{z}_i))\| \leq 8\kappa B$. Thus we either have that $((c - c_i)(\boldsymbol{z}_i - \boldsymbol{z}_i') - \overline{c}_i(\boldsymbol{z} - \boldsymbol{z}_i))$ is an $\mathsf{MSIS}_{\mu,\lambda+\mu+1,8\kappa B}$ solution for $\boldsymbol{B}_0$ or that $(c - c_i)(\boldsymbol{z}_i - \boldsymbol{z}_i') = \overline{c}_i(\boldsymbol{z} - \boldsymbol{z}_i)$. We focus on the latter case.

We now use $\boldsymbol{z} = \boldsymbol{y}_\mathsf{e}' + c\boldsymbol{r}_\mathsf{e}$, $\boldsymbol{z}_i = (\boldsymbol{y}_\mathsf{e})_i + c_i(\boldsymbol{r}_\mathsf{e})_i$ and $\boldsymbol{z}_i' = (\boldsymbol{y}_\mathsf{e})_i + c_i'(\boldsymbol{r}_\mathsf{e})_i$ in the above equality, and get that

$$\begin{aligned}&(c - c_i)\overline{c}_i(\boldsymbol{r}_\mathsf{e})_i \equiv \overline{c}_i(\boldsymbol{y}_\mathsf{e}' - (\boldsymbol{y}_\mathsf{e})_i + (c - c_i)(\boldsymbol{r}_\mathsf{e})_i) \pmod{X^{N/k} - \zeta^i}\\ &\Leftrightarrow \overline{c}_i(\boldsymbol{y}_\mathsf{e} - (\boldsymbol{y}_\mathsf{e})_i) \equiv 0 \pmod{X^{N/k} - \zeta^i}\\ &\Leftrightarrow \boldsymbol{y}_\mathsf{e}' \equiv (\boldsymbol{y}_\mathsf{e})_i \equiv \boldsymbol{y}_\mathsf{e} \pmod{X^{N/k} - \zeta^i}\end{aligned}$$

since $\overline{c}_i \mod (X^{N/k} - \zeta^i) \neq 0$. Since this holds for all $i$, we hence have that $\boldsymbol{y}'_{\mathsf{e}} = \boldsymbol{y}_{\mathsf{e}}$ and the first part of the lemma holds.

From the construction of $\boldsymbol{r}_{\mathsf{e}}$ and $\boldsymbol{y}_{\mathsf{e}}$, and from the verification equations it follows that

$$\boldsymbol{B}_0\boldsymbol{r}_{\mathsf{e}} \equiv \boldsymbol{B}_0(\boldsymbol{r}_{\mathsf{e}})_i \equiv \boldsymbol{B}_0\frac{\boldsymbol{z}_i - \boldsymbol{z}'_i}{\overline{c}_i} \equiv \boldsymbol{t}_0 \pmod{X^{N/k} - \zeta^i}, \text{ and}$$
$$\boldsymbol{B}_0\boldsymbol{y}_{\mathsf{e}} \equiv \boldsymbol{B}_0(\boldsymbol{y}_{\mathsf{e}})_i \equiv \boldsymbol{B}_0\frac{c_i\boldsymbol{z}'_i - c'_i\boldsymbol{z}_i}{\overline{c}_i} \equiv \boldsymbol{w} \pmod{X^{N/k} - \zeta^i}.$$

Hence the lemma holds by the Chinese remainder theorem. □

The extracted $\boldsymbol{r}_{\mathsf{e}}$ can be used to define $m_{\mathsf{e}}$ such that $\boldsymbol{t}_1 = \langle \boldsymbol{b}_1, \boldsymbol{r}_{\mathsf{e}} \rangle + m_{\mathsf{e}}$. Then we have found a weak opening, which is defined as below, for the commitment $\boldsymbol{t} = \boldsymbol{t}_0 \| \boldsymbol{t}_1$.

**Definition 25.** A *weak opening* for the commitment $\boldsymbol{t} = \boldsymbol{t}_0 \| \boldsymbol{t}_1$ consists of $k$ polynomials $\overline{c}_i \in R_q$, a randomness vector $\boldsymbol{r}$ over $R_q$, and a message $m \in R_q$ such that

$$\begin{aligned}
&\|\overline{c}_i\|_1 \leq 2\kappa \text{ and } \overline{c}_i \mod (X^{N/k} - \zeta^i) \neq 0 \text{ for all } 1 \leq i \leq k,\\
&\|\overline{c}_i\boldsymbol{r}\| \leq 2B \text{ for all } 1 \leq i \leq k,\\
&\boldsymbol{B}_0\boldsymbol{r} = \boldsymbol{t}_0,\\
&\langle \boldsymbol{b}_1, \boldsymbol{r} \rangle + m = \boldsymbol{t}_1.
\end{aligned}$$

We now prove that the scheme is still binding with respect to weak openings.

**Theorem 13.** If there is an algorithm $\mathcal{A}$ who can break the binding property of the BDLOP commitment scheme with respect to weak openings with probability $\varepsilon$, then there is an algorithm $\mathcal{A}'$ with advantage $\varepsilon$ in solving the $\mathsf{MSIS}_{\mu,\lambda+\mu+1,8\kappa B}$ problem.

*Proof.* Suppose that $\mathcal{A}'$ is given a $\mathsf{MSIS}_{\mu,\lambda+\mu+1,8\kappa B}$-instance $\boldsymbol{B}_0 \in R_q^{\mu\times(\lambda+\mu+1)}$, for which it is supposed to find a vector $\boldsymbol{z} \in R_q^{\lambda+\mu+1}$ such that $\boldsymbol{B}_0\boldsymbol{z} = 0$ and $\|\boldsymbol{z}\| \leq 8\kappa B$. $\mathcal{A}'$ creates a random $\boldsymbol{b}_1 \in R_q^{\lambda+\mu+1}$, and outputs $\boldsymbol{B}_0, \boldsymbol{b}_1$ as public parameters. With probability $\varepsilon$, $\mathcal{A}$ is able to come up with two weak openings $((\overline{c}_i), \boldsymbol{r}_{\mathsf{e}}, m_{\mathsf{e}})$ and $((\overline{c}'_i), \boldsymbol{r}'_{\mathsf{e}}, m'_{\mathsf{e}})$ such that $m_{\mathsf{e}} \neq m'_{\mathsf{e}}$. This implies that

$$\langle \boldsymbol{b}_1, \boldsymbol{r}_{\mathsf{e}} \rangle + m_{\mathsf{e}} = \boldsymbol{t}_1 = \langle \boldsymbol{b}_1, \boldsymbol{r}'_{\mathsf{e}} \rangle + m'_{\mathsf{e}},$$

which then implies that $\boldsymbol{r}_{\mathsf{e}} \neq \boldsymbol{r}'_{\mathsf{e}}$. Hence there must exists an $i \in \{1, \ldots, k\}$ for which $\boldsymbol{r}_{\mathsf{e}} \not\equiv \boldsymbol{r}'_{\mathsf{e}} \pmod{X^{N/k} - \zeta^i}$. Since the polynomials $\overline{c}_i$ and $\overline{c}'_i$ are non-zero modulo $X^{N/k} - \zeta^i$, this implies that

$$\overline{c}_i\overline{c}'_i(\boldsymbol{r}_{\mathsf{e}} - \boldsymbol{r}'_{\mathsf{e}}) = \overline{c}'_i\overline{c}_i\boldsymbol{r}_{\mathsf{e}} - \overline{c}_i\overline{c}'_i\boldsymbol{r}'_{\mathsf{e}} \neq 0.$$

This in turn implies that

$$\boldsymbol{B}_0\overline{c}_i\overline{c}'_i(\boldsymbol{r}_{\mathsf{e}} - \boldsymbol{r}'_{\mathsf{e}}) = 0$$

for a non-zero $\overline{c}_i\overline{c}'_i(\boldsymbol{r}_{\mathsf{e}} - \boldsymbol{r}'_{\mathsf{e}})$. Since we have that $\|\overline{c}_i\|_1 \leq 2\kappa$ and $\|\overline{c}_i\boldsymbol{r}_{\mathsf{e}}\| \leq 2B$ for weak openings, we get that $\|\overline{c}_i\overline{c}'_i(\boldsymbol{r}_{\mathsf{e}} - \boldsymbol{r}'_{\mathsf{e}})\| \leq 8\kappa B$. Hence $\mathcal{A}'$ has advantage $\varepsilon$ in solving the $\mathsf{MSIS}_{\mu,\lambda+\mu+1,8\kappa B}$ problem. □

We choose parameters such that the maximum probability over $\mathbb{Z}_q$ of each of the $N/k$ coefficients of $c \mod X^{N/k} - \zeta^i$ is not much bigger than $1/q$, by the upper bound in Lemma 14. Then the protocol has a cheating probability of about $q^{-N/k}$. If this is not negligible, we can run $l$ copies of the protocol in parallel to reduce the cheating probability to $q^{-lN/k}$.

$\Pi^{\sigma}_{\mathsf{open}}$

Public Information: $\boldsymbol{B}_0 \in R_q^{\mu\times(\lambda+\mu+1)}, \boldsymbol{b}_1 \in R_q^{\lambda+\mu+1}, \boldsymbol{t}_0 = \boldsymbol{B}_0\boldsymbol{r}, \boldsymbol{t}_1 = \langle \boldsymbol{b}_1, \boldsymbol{r}\rangle + m$
Prover's Information: $\boldsymbol{r} \in \{-1,0,1\}^{(\lambda+\mu+1)N} \subset R_q^{\lambda+\mu+1}, m \in R_q$

| Prover | | Verifier |
|---|---|---|
| For $i = 0,\dots,l-1$ : | | |
| $\boldsymbol{y}_i \xleftarrow{\$} D_{R^{\lambda+\mu+1},\mathfrak{s}}$ | | |
| $\boldsymbol{w}_i = \boldsymbol{B}_0\boldsymbol{y}_i$ | | |
| | $\xrightarrow{\boldsymbol{w}_i}$ | |
| | $\xleftarrow{c}$ | $c \xleftarrow{\$} C$ |
| For $i = 0,\dots,l-1$ : | | |
| $\boldsymbol{z}_i = \boldsymbol{y}_i + \sigma^i(c)\boldsymbol{r}$ | | |
| If $\mathsf{Rej}_1((\boldsymbol{z}_i),(\sigma^i(c)\boldsymbol{r}),\mathfrak{s}) = 1$, abort | $\xrightarrow{\boldsymbol{z}_i}$ | |
| | | Accept iff: |
| | | For $i = 0,\dots,l-1$ : |
| | | $\|\boldsymbol{z}_i\| \overset{?}{\leq} B = \mathfrak{s}\sqrt{2(\lambda+\mu+1)N}$ |
| | | $\boldsymbol{B}_0\boldsymbol{z}_i \overset{?}{=} \boldsymbol{w}_i + \sigma^i(c)\boldsymbol{t}_0$ |

Figure 5.2: Proof of knowledge $\Pi^{\sigma}_{\mathsf{open}}$ of $(m, \boldsymbol{r}, \sigma^i(\overline{c})) \in R_q \times R_q^{\lambda+\mu+1} \times R_q^k$ such that $\boldsymbol{B}_0\boldsymbol{r} = \boldsymbol{t}_0$, $\langle \boldsymbol{b}_1, \boldsymbol{r}\rangle + m = \boldsymbol{t}_1$ and $\|\sigma^i(\overline{c})\boldsymbol{r}\| \leq 2B$ for all $i \in [k]$.

Instead of sampling the challenges independently, we let the challenges in the $l$ parallel executions be the images $\sigma^j(c)$, $j = 0,\dots,l-1$, of a single polynomial $c \in \mathcal{C}$. We use the automorphism of order $lN/k$, $\sigma = \sigma_{2k/l+1} \in \mathsf{Aut}(R_q)$, that stabilizes the ideals

$$\left(X^{lN/k} - \zeta^{il}\right) = \prod_{j=0,\dots,l-1} \sigma^j\left(X^{N/k} - \zeta^i\right) = \prod_{j\in\langle 2k/l+1\rangle/\langle 2k+1\rangle} \left(X^{N/k} - \zeta^{ij}\right)$$

for $i \in \langle -1, 5\rangle/\langle 2k/l+1\rangle \cong \mathbb{Z}^{\times}_{2k/l}$.

Now the maximum probability of $c \mod (X^{lN/k} - \zeta^{il})$ is essentially $q^{-lN/k}$, which is negligible. This way we can assure that the prover must answer two challenges $c, c'$ that differ modulo $X^{lN/k} - \zeta^{il}$, and $\overline{c} = c - c'$ must be non-zero modulo at least one of the divisors, say $X^{N/k} - \zeta^i$. For every other divisor $\sigma^i(X^{N/k} - \zeta^i)$, we then have

$$\sigma^j(\overline{c}) \mod \sigma^j\left(X^{N/k} - \zeta^i\right) = \sigma^j\left(\overline{c} \mod (X^{N/k} - \zeta^i)\right) \neq 0.$$

This means that we have an accepting transcript pair with non-zero $\overline{c}$ modulo every prime divisor of $(X^{lN/k} - \zeta^{il})$. By repeating the argument for every $i \in \mathbb{Z}^{\times}_{2k/l}$, we get that we can get an accepting transcript pair with non-zero $\overline{c}$ modulo every prime divisor of $X^N + 1$.

The resulting opening proof looks like $\Pi_{\mathsf{open}}$, only that it is run in $l$ parallels and uses Galois automorphisms. We set the standard deviation, in accordance with the commitment scheme, as $\mathfrak{s} = \gamma\kappa\sqrt{(\lambda+\mu+1)N}$, since $\beta = 1$.

The protocol is given as $\Pi^{\sigma}_{\mathsf{open}}$ in Figure 5.2. We give the full security analysis of the protocol in Appendix B.1.

### 5.2.2 Product proof

We now explain how we can use Galois automorphisms to construct a more efficient protocol for proving multiplicative relations among committed values. Suppose that the prover knows an opening of a commitment $\boldsymbol{t}$ to the secret polynomials $m_1, m_2, m_3 \in R_q$

$$\begin{aligned} \boldsymbol{t}_0 &= \boldsymbol{B}_0\boldsymbol{r}, \\ \boldsymbol{t}_1 &= \langle \boldsymbol{b}_1, \boldsymbol{r}\rangle + m_1, \\ \boldsymbol{t}_2 &= \langle \boldsymbol{b}_2, \boldsymbol{r}\rangle + m_2, \\ \boldsymbol{t}_3 &= \langle \boldsymbol{b}_3, \boldsymbol{r}\rangle + m_3, \end{aligned} \tag{5.8}$$

and want to prove that $m_1 m_2 = m_3$ in $R_q$. If we were to follow the idea from Section 5.1, we would commit to random masking polynomials $a_1, a_2, a_3 \in R_q$, and upon receiving a challenge $c$ from the verifier, compute the masked openings $f_i = a_i + cm_i$ for $i = 1, 2, 3$ and prove their well-formedness.

But now we instead let the verifier compute the masked openings as $f_i = \langle \boldsymbol{b}_i, \boldsymbol{z}\rangle - c\boldsymbol{t}_i$, where $\boldsymbol{z} = \boldsymbol{y} + c\boldsymbol{r}$ is from the commitment opening proof. This is made possible by the results we discussed in the previous section, namely that the verifier will be convinced that the $\boldsymbol{z}$ in the opening proof is of the form $\boldsymbol{z} = \boldsymbol{y}_{\mathsf{e}} + c\boldsymbol{r}_{\mathsf{e}}$, where $\boldsymbol{y}_{\mathsf{e}}, \boldsymbol{r}_{\mathsf{e}}$ are independent of $c$, and that $\boldsymbol{t}_i = \langle \boldsymbol{b}_i, \boldsymbol{r}_{\mathsf{e}}\rangle + (m_i)_{\mathsf{e}}$. Thus the verifier will be convinced that

$$f_i = \langle \boldsymbol{b}_i, \boldsymbol{z}\rangle - c\boldsymbol{t}_i = \langle \boldsymbol{b}_i, \boldsymbol{y}_{\mathsf{e}}\rangle - c(m_i)_{\mathsf{e}},$$

which is exactly a masked opening of $(m_i)_{\mathsf{e}}$ with challenge $c$ and masking polynomial $(a_i)_{\mathsf{e}} = \langle \boldsymbol{b}_i, \boldsymbol{y}_{\mathsf{e}}\rangle$.

We observe that

$$f_1 f_2 - cf_3 = c^2(m_1 m_2 - m_3) + c(a_1 m_2 + a_2 m_1 - a_3) + a_1 a_2.$$

In order to get rid of garbage terms in this equation, we make the commitment $\boldsymbol{t}_4 = \langle \boldsymbol{b}_4, \boldsymbol{r}\rangle + a_3 - a_1 m_2 - a_2 m_1$, take the masked opening $f_4 = \langle \boldsymbol{b}_4, \boldsymbol{z}\rangle - c\boldsymbol{t}_4$, and instead use the relation $f_1 f_2 + cf_3 + f_4$. In this expression the coefficient for $c$ vanishes, and so this polynomial is constant in $c$, with constant coefficient $v = \langle \boldsymbol{b}_4, \boldsymbol{y}\rangle + a_1 a_2$. Thus the prover can send this polynomial before seeing $c$. Since we use the masking polynomials $a_i = \langle \boldsymbol{b}_i, \boldsymbol{y}\rangle$, we get that the commitment to the garbage term is

$$\boldsymbol{t}_4 = \langle \boldsymbol{b}_4, \boldsymbol{r}\rangle + \langle \boldsymbol{b}_3, \boldsymbol{y}\rangle - m_1\langle \boldsymbol{b}_2, \boldsymbol{y}\rangle - m_2\langle \boldsymbol{b}_1, \boldsymbol{y}\rangle.$$

This means that the verifier can just check that $f_1 f_2 + cf_3 + f_4 = v$ for $v = \langle \boldsymbol{b}_4, \boldsymbol{y}\rangle + \langle \boldsymbol{b}_1, \boldsymbol{y}\rangle\langle \boldsymbol{b}_2, \boldsymbol{y}\rangle$.

We also for this protocol need to be able to run several copies in parallel to reduce the soundness error, and will then use the automorphism opening proof from Section 5.2.1. So instead of

proving $m_1 m_2 \equiv m_3 \pmod{\sigma^i(X^{N/k} - \zeta^j)}$, we linear combine all the permutations $\sigma^i(m_1 m_2 - m_3)$ with independently uniformly random challenge polynomials $\alpha_i$, and set out to prove that

$$\sum_{i=0}^{l-1} \alpha_i \sigma^i(m_1 m_2 - m_3) \equiv 0 \pmod{\sigma^{i'}(X^{N/k} - \zeta^j)}$$

for $i' = 0, \ldots, l-1$. For each $i'$ the prover has independent cheating probability, and the above equation proves that

$$\begin{aligned} &\sigma^i(m_1 m_2 - m_3) \equiv 0 \pmod{\sigma^{i'}(X^{N/k} - \zeta^j)} \\ &\Rightarrow m_1 m_2 - m_3 \equiv 0 \pmod{\sigma^{i'-i}(X^{N/k} - \zeta^j)} \end{aligned}$$

for all $i = 0, \ldots, l-1$. This way the probability that a cheating prover succeeds is significantly reduced. Further, the $l$ masked openings are now computed as

$$\boldsymbol{f}_j^{(i)} = \langle \boldsymbol{b}_j, \boldsymbol{z}_i \rangle - \sigma^i(c)\boldsymbol{t}_j \text{ for } j = 1, 2, 3,$$

where $\boldsymbol{z}_i = \boldsymbol{y}_i + \sigma^i(c)\boldsymbol{r}$ are from the automorphism opening proof. This gives the extracted expressions

$$\boldsymbol{f}_j^{(i)} = \langle \boldsymbol{b}_j, (\boldsymbol{y}_i)_{\mathsf{e}} \rangle - \sigma^i(c)(m_j)_{\mathsf{e}} \text{ for } j = 1, 2, 3.$$

In order to apply the previously explained method in this situation, we observe that

$$\begin{aligned} \sum_{i=0}^{l-1} \alpha_i \sigma^{-i}\Big(\boldsymbol{f}_1^{(i)} \boldsymbol{f}_2^{(i)} + \sigma^i(c)\boldsymbol{f}_3^{(i)}\Big) &= \sum_{i=0}^{l-1} \alpha_i \sigma^{-i}\Big(\langle \boldsymbol{b}_1, \boldsymbol{y}_i^* \rangle \langle \boldsymbol{b}_2, \boldsymbol{y}_i^* \rangle\Big) + c \sum_{i=0}^{l-1} \alpha_i \sigma^{-i}\Big(\langle \boldsymbol{b}_3, (\boldsymbol{y}_i)_{\mathsf{e}} \rangle \\ &\quad - (m_1)_{\mathsf{e}} \langle \boldsymbol{b}_2, (\boldsymbol{y}_i)_{\mathsf{e}} \rangle - (m_2)_{\mathsf{e}} \langle \boldsymbol{b}_1, (\boldsymbol{y}_i)_{\mathsf{e}} \rangle\Big) + c^2 \left( \sum_{i=0}^{l-1} \alpha_i \sigma^{-i}\big((m_1)_{\mathsf{e}}(m_2)_{\mathsf{e}} - (m_3)_{\mathsf{e}}\big) \right). \end{aligned}$$

In order to make this polynomial constant in $c$ we add the term $\boldsymbol{f}_4 = \langle \boldsymbol{b}_4, \boldsymbol{z}_0 \rangle - c\boldsymbol{t}_4$ stemming from the garbage commitment

$$\boldsymbol{t}_4 = \langle \boldsymbol{b}_4, \boldsymbol{r} \rangle + \sum_{i=0}^{l-1} \alpha_i \sigma^{-i}\Big(\langle \boldsymbol{b}_3, \boldsymbol{y}_i \rangle - m_1 \langle \boldsymbol{b}_2, \boldsymbol{y}_i \rangle - m_2 \langle \boldsymbol{b}_1, \boldsymbol{y}_i \rangle\Big). \tag{5.9}$$

The verifier can now check that the new polynomial equals

$$\boldsymbol{v} = \langle \boldsymbol{b}_4, \boldsymbol{y}_0 \rangle + \sum_{i=0}^{l-1} \alpha_i \sigma^{-i}\Big(\langle \boldsymbol{b}_1, \boldsymbol{y}_i \rangle \langle \boldsymbol{b}_2, \boldsymbol{y}_i \rangle\Big). \tag{5.10}$$

We note that the distribution $\chi$ on $R$ that we use to sample randomness, can be any of the standard choices. We set the standard deviation, in accordance with the commitment scheme, as $\mathfrak{s} = \gamma\kappa\beta\sqrt{(\lambda + \mu + 4)N}$.

The protocol is given as $\Pi^{\sigma}_{\mathsf{prod}}$ in Figure 5.3. We give the full security analysis of the protocol in Appendix B.2.

$\Pi^{\sigma}_{\mathsf{prod}}$

Public Information: $\boldsymbol{B}_0 \in R_q^{\mu \times (\lambda+\mu+4)}, \boldsymbol{b}_1, \ldots, \boldsymbol{b}_4 \in R_q^{\lambda+\mu+4}$
Prover's Information: $m_1, m_2, m_3 \in R_q$

| Prover | | Verifier |
|---|---|---|
| $\boldsymbol{r} \xleftarrow{\$} \chi^{(\lambda+\mu+4)N}$ | | |
| $\boldsymbol{t}_0 = \boldsymbol{B}_0\boldsymbol{r}$ | | |
| $\boldsymbol{t}_1 = \langle \boldsymbol{b}_1, \boldsymbol{r} \rangle + m_1$ | | |
| $\boldsymbol{t}_2 = \langle \boldsymbol{b}_2, \boldsymbol{r} \rangle + m_2$ | | |
| $\boldsymbol{t}_3 = \langle \boldsymbol{b}_3, \boldsymbol{r} \rangle + m_3$ | | |
| $\boldsymbol{t} = \boldsymbol{t}_0 \Vert \boldsymbol{t}_1 \Vert \boldsymbol{t}_2 \Vert \boldsymbol{t}_3$ | | |
| For $i = 0, \ldots, l-1$ : | | |
| $\boldsymbol{y}_i \xleftarrow{\$} D_{R_q^{\lambda+\mu+4}, \mathfrak{s}}$ | | |
| $\boldsymbol{w}_i = \boldsymbol{B}_0\boldsymbol{y}_i$ | | |
| | $\xrightarrow{\boldsymbol{t}, \boldsymbol{w}_i}$ | |
| | $\xleftarrow{\alpha_0, \ldots, \alpha_{l-1}}$ | $\alpha_0, \ldots, \alpha_{l-1} \xleftarrow{\$} R_q$ |
| Define $\boldsymbol{t}_4$ as in (5.9) | | |
| Define $\boldsymbol{v}$ as in (5.10) | | |
| | $\xrightarrow{\boldsymbol{t}_4, \boldsymbol{v}}$ | |
| | $\xleftarrow{c}$ | $c \xleftarrow{\$} C$ |
| For $i = 0, \ldots, l-1$ : | | |
| $\boldsymbol{z}_i = \boldsymbol{y}_i + \sigma^i(c)\boldsymbol{r}$ | | |
| If $\mathsf{Rej}_1((\boldsymbol{z}_i), (\sigma^i(c)\boldsymbol{r}), \mathfrak{s}) = 1$, abort | $\xrightarrow{\boldsymbol{z}_i}$ | $\boldsymbol{f}_4 = \langle \boldsymbol{b}_4, \boldsymbol{z}_0 \rangle - c\boldsymbol{t}_4$ |
| | | Accept iff: |
| | | For $i = 0, \ldots, l-1$ : |
| | | $\Vert \boldsymbol{z}_i \Vert \overset{?}{\leq} B = \mathfrak{s}\sqrt{2(\lambda+\mu+4)N}$ |
| | | $\boldsymbol{B}_0\boldsymbol{z}_i \overset{?}{=} \boldsymbol{w}_i + \sigma^i(c)\boldsymbol{t}_0$ |
| | | $\boldsymbol{f}_1^{(i)} = \langle \boldsymbol{b}_1, \boldsymbol{z}_i \rangle - \sigma^i(c)\boldsymbol{t}_1$ |
| | | $\boldsymbol{f}_2^{(i)} = \langle \boldsymbol{b}_2, \boldsymbol{z}_i \rangle - \sigma^i(c)\boldsymbol{t}_2$ |
| | | $\boldsymbol{f}_3^{(i)} = \langle \boldsymbol{b}_3, \boldsymbol{z}_i \rangle - \sigma^i(c)\boldsymbol{t}_3$ |
| | | $\sum_{i=0}^{k-1} \alpha_i \sigma^{-i}\Big(\boldsymbol{f}_1^{(i)}\boldsymbol{f}_2^{(i)} + \sigma^i(c)\boldsymbol{f}_3^{(i)}\Big) + \boldsymbol{f}_4 \overset{?}{=} \boldsymbol{v}$ |

Figure 5.3: Proof of knowledge $\Pi^{\sigma}_{\mathsf{prod}}$ of $m_1, m_2, m_3 \in R_q$ such that $m_1 m_2 = m_3$.

## 5.3 Proof of linear relations using inner products

The scheme from Section 5.1 uses a challenge space of size $q$ for proving knowledge of $\vec{s}$, and is thus restricted to a soundness error of $1/q$. Esgin et al. introduced in [10] a new approach for proving knowledge of $\vec{s}$ with much lower soundness error. We assume in this section that $q \equiv 1 \pmod{2N}$, such that

$$X^N + 1 \equiv \prod_{j \in \mathbb{Z}_{2N}^\times} (X - \zeta^j) \mod q$$

for a primitive $2N$-th root of unity $\zeta$, according to Lemma 2.

The idea behind the protocol is that if $A\vec{s} = \vec{u}$, then for all $\vec{\gamma} \in \mathbb{Z}_q^m$ we have that $\langle A\vec{s} - \vec{u}, \vec{\gamma} \rangle = 0$, whilst if $A\vec{s} \neq \vec{u}$ then this holds true only with probability $1/q$. Suppose that $\vec{s} = \mathsf{NTT}(\check{s})$ for some $\check{s} \in R_q$. We then use Lemma 9 and the inverse $\mathsf{NTT}$ to derive that

$$\begin{aligned}
\langle A\vec{s} - \vec{u}, \vec{\gamma} \rangle &= \langle A\vec{s}, \vec{\gamma} \rangle - \langle \vec{u}, \vec{\gamma} \rangle = \langle \vec{s}, A^T\vec{\gamma} \rangle - \langle \vec{u}, \vec{\gamma} \rangle \\
&= \sum_{j \in \mathbb{Z}_{2N}^\times} \check{s}(\zeta^j)(\mathsf{NTT}^{-1}(A^T\vec{\gamma}))(\zeta^j) - \langle \vec{u}, \vec{\gamma} \rangle \\
&= \frac{1}{N} \sum_{j \in \mathbb{Z}_{2N}^\times} f(\zeta^j) = f_0,
\end{aligned}$$

where we have defined $f := \mathsf{NTT}^{-1}(NA^T\vec{\gamma})\check{s} - \langle \vec{u}, \vec{\gamma} \rangle \in R_q$, and $f_0 \in \mathbb{Z}_q$ is the constant coefficient of $f$. So, in order to show that $\langle A\vec{s} - \vec{u}, \vec{\gamma} \rangle = 0$, one can show that the constant coefficient of $f$ is zero. Suppose that the prover knows an opening of a commitment to $\check{s}$

$$\begin{aligned}
\boldsymbol{t}_0 &= \boldsymbol{B}_0\boldsymbol{r}, \\
\boldsymbol{t}_1 &= \langle \boldsymbol{b}_1, \boldsymbol{r} \rangle + \check{s}.
\end{aligned}$$

In order to prove that the constant coefficient of $f$ is zero, the prover samples a masking polynomial $g$ with a zero constant coefficient. Upon receiving $\vec{\gamma}$ from the verifier, it sends $h := f + g$. The verifier can now just check if $h_0 = 0$.

We must also prove that $h$ was constructed correctly. In order to do this, the prover sends a commitment $\boldsymbol{t}_2 = \langle \boldsymbol{b}_1, \boldsymbol{r} \rangle + g$ to $g$. We notice that

$$\mathsf{NTT}^{-1}(NA^T\vec{\gamma})\boldsymbol{t}_1 - \langle \vec{u}, \vec{\gamma} \rangle = \langle \mathsf{NTT}^{-1}(NA^T\vec{\gamma})\boldsymbol{b}_1, \boldsymbol{r} \rangle + f$$

is a commitment to $f$. Hence the verifier can compute the commitment $\boldsymbol{\tau} = \mathsf{NTT}^{-1}(NA^T\vec{\gamma})\boldsymbol{t}_1 - \langle \vec{u}, \vec{\gamma} \rangle$ to $f$. For well-formedness, the prover can now prove that $\boldsymbol{\tau} + \boldsymbol{t}_2 - h$ is a commitment to 0.

In order to do this, we use the masking polynomial $\boldsymbol{y}$ and corresponding $\boldsymbol{z} = \boldsymbol{y} + c\boldsymbol{r}$ from the opening proof for the commitments. If we let $\boldsymbol{v} = \langle \mathsf{NTT}^{-1}(NA^T\gamma)\boldsymbol{b}_1 + \boldsymbol{b}_2, \boldsymbol{y} \rangle$, it suffices that verifier can check that

$$\langle \mathsf{NTT}^{-1}(NA^T\gamma)\boldsymbol{b}_1 + \boldsymbol{b}_2, \boldsymbol{z} \rangle = \boldsymbol{v} + c(\boldsymbol{\tau} + \boldsymbol{t}_2 - h).$$

This protocol has a cheating probability of $1/q$, but it is possible to make the soundness error negligible with little additional cost. We now explain how.

Suppose we were to have the $l$ functions $\mathsf{L}_0, \ldots, \mathsf{L}_{l-1}$, such that for any $0 \leq \mu < l$ and $\vec{\gamma}_\mu \in \mathbb{Z}_q^m$,

we have that all coefficients of $\mathsf{L}_\mu(\vec{\gamma}_\mu) \in R_q$ are zero, except for the $\mu$-th one which is equal to $\langle A\vec{s} - \vec{u}, \vec{\gamma}_\mu \rangle$. Then we could instead define the polynomial

$$f = \mathsf{L}_0(\vec{\gamma}_0) + \cdots + \mathsf{L}_{l-1}(\vec{\gamma}_{l-1}),$$

which has the property that for all $0 \leq \mu < l$, the coefficient of $X^\mu$ is equal to $\langle A\vec{s} - \vec{u}, \vec{\gamma}_\mu \rangle$. This implies that if $A\vec{s} = \vec{u}$, then $f_0 = f_1 = \cdots = f_{l-1} = 0$, and if $A\vec{s} \neq \vec{u}$ this holds true only with probability $1/q^l$. Thus we can let the verifier send $l$ independently uniform vectors $\vec{\gamma}_0, \ldots, \vec{\gamma}_{l-1}$, and then prove that the $l$ first coefficients of $f$ are zero in order to reduce the soundness error. In accordance with this new approach, the prover will also use the automorphism opening proof with soundness error $1/q^l$ from Section 5.2.1, to prove that the commitments are valid.

We leave the details of how to construct these functions $\mathsf{L}_\mu$ to [10], but

$$\mathsf{L}_\mu(\vec{\gamma}) = \frac{1}{l} X^\mu \sum_{\nu=0}^{l-1} \sigma^\nu(\mathsf{NTT}^{-1}(NA^T\vec{\gamma})\check{s} - \langle \vec{u}, \vec{\gamma} \rangle)$$

has this property when $\sigma = \sigma_{2N/l+1} \in \mathsf{Aut}(R_q)$. This is the same automorphism that was used in the automorphism opening proof. The new function $f$ is now

$$f = \sum_{\mu=0}^{l-1} \frac{1}{l} X^\mu \sum_{\nu=0}^{l-1} \sigma^\nu(\mathsf{NTT}^{-1}(NA^T\vec{\gamma})\check{s} - \langle \vec{u}, \vec{\gamma} \rangle). \tag{5.11}$$

The prover will now have to draw a masking polynomial $g$ with $g_0 = \cdots = g_{l-1} = 0$ instead, set $h := g + f$, and let the verifier check if $h_0 = \cdots = h_{l-1} = 0$.

In order to prove that $h$ was constructed correctly, the prover sends the commitment $\boldsymbol{t}_2 = \langle \boldsymbol{b}_2, \boldsymbol{r} \rangle + g$ to $g$. We also note that the verifier will have to compute a commitment to this new $f$ from $\boldsymbol{t}_1$ and $\vec{\gamma}_0, \ldots, \vec{\gamma}_{l-1}$, and this is achieved via

$$\begin{aligned}\boldsymbol{\tau} &= \sum_{\mu=0}^{l-1} \frac{1}{l} X^\mu \sum_{\nu=0}^{l-1} \sigma^\nu(\mathsf{NTT}^{-1}(NA^T\vec{\gamma}_\mu)\boldsymbol{t}_1 - \langle \vec{u}, \vec{\gamma}_\mu \rangle) \\ &= \sum_{\mu=0}^{l-1} \frac{1}{l} X^\mu \sum_{\nu=0}^{l-1} \sigma^\nu\Big(\langle \mathsf{NTT}^{-1}(NA^T\vec{\gamma}_\mu)\boldsymbol{b}_1, \boldsymbol{r} \rangle\Big) + f.\end{aligned}$$

In order to prove that $\boldsymbol{\tau} + \boldsymbol{t}_2 - h$ is a commitment to 0, we use the $\boldsymbol{y}_i$ and corresponding $\boldsymbol{z}_i = \boldsymbol{y}_i + \sigma^i(c)\boldsymbol{r}$ for $i = 0, \ldots, l-1$ from the automorphism opening proof. We now let

$$\boldsymbol{v}_i = \sum_{\mu=0}^{l-1} \frac{1}{l} X^\mu \sum_{\nu=0}^{l-1} \sigma^\nu\Big(\langle \mathsf{NTT}^{-1}(NA^T\vec{\gamma}_\mu)\boldsymbol{b}_1, \boldsymbol{y}_{i-\nu \mod l} \rangle\Big) + \langle \boldsymbol{b}_2, \boldsymbol{y}_i \rangle \tag{5.12}$$

for $i = 0, \ldots, l-1$. To prove well-formedness of $h$, it now suffices that the verifier can check if

$$\sum_{\mu=0}^{l-1} \frac{1}{l} X^\mu \sum_{\nu=0}^{l-1} \sigma^\nu\Big(\langle \mathsf{NTT}^{-1}(NA^T\vec{\gamma}_\mu)\boldsymbol{b}_1, \boldsymbol{z}_{i-\nu \mod l} \rangle\Big) + \langle \boldsymbol{b}_2, \boldsymbol{z}_i \rangle = \boldsymbol{v}_i + \sigma^i(c)(\boldsymbol{\tau} + \boldsymbol{t}_2 - h),$$

which implies that $\boldsymbol{\tau} + \boldsymbol{t}_2 - h$ is a commitment to zero.

We set the standard deviation, in accordance with the commitment scheme, as $\mathfrak{s} = \gamma\kappa\beta\sqrt{(\lambda + \mu + 2)N}$. The complete protocol is given as $\Pi_{\mathsf{inner}}$ in Figure 5.4, and all the verification equations are gathered in Algorithm 4. We give the full security analysis of the protocol in Appendix C.

$\underline{\Pi_{\mathsf{inner}}}$

Public Information: $A \in \mathbb{Z}_q^{m \times n}, \vec{u} = A\vec{s} \in \mathbb{Z}_q^m$ $\boldsymbol{B}_0 \in R_q^{\mu \times (\lambda+\mu+2)}, \boldsymbol{b}_1, \boldsymbol{b}_2 \in R_q^{\lambda+\mu+2}$
Prover's Information: $\check{s} \in R_q, \vec{s} = \mathsf{NTT}(\check{s})$

| Prover | | Verifier |
|---|---|---|
| $g \xleftarrow{\$} \{g \in R_q \mid g_0 = \cdots = g_{l-1} = 0\}$ | | |
| $\boldsymbol{r} \xleftarrow{\$} \chi^{(\lambda+\mu+2)N}$ | | |
| $\boldsymbol{t}_0 = \boldsymbol{B}_0 \boldsymbol{r}$ | | |
| $\boldsymbol{t}_1 = \langle \boldsymbol{b}_1, \boldsymbol{r} \rangle + \check{s}$ | | |
| $\boldsymbol{t}_2 = \langle \boldsymbol{b}_2, \boldsymbol{r} \rangle + g$ | | |
| $\boldsymbol{t} = \boldsymbol{t}_0 \| \boldsymbol{t}_1 \| \boldsymbol{t}_2$ | | |
| For $i = 0, \ldots, l-1$ : | | |
| $\boldsymbol{y}_i \xleftarrow{\$} D_{R_q^{\lambda+\mu+2}, \mathfrak{s}}$ | | |
| $\boldsymbol{w}_i = \boldsymbol{B}_0 \boldsymbol{y}_i$ | | |
| | $\xrightarrow{\boldsymbol{t}, \boldsymbol{w}_i}$ | |
| | $\xleftarrow{\vec{\gamma}_0, \ldots, \vec{\gamma}_{l-1}}$ | $\vec{\gamma}_0, \ldots, \vec{\gamma}_{l-1} \xleftarrow{\$} \mathbb{Z}_q^m$ |
| $h = g + f,$ where $f$ is defined as in (5.11) | | |
| For $i = 0, \ldots, l-1$ : | | |
| Define $\boldsymbol{v}_i$ as in (5.12) | | |
| | $\xrightarrow{h, \boldsymbol{v}_i}$ | |
| | $\xleftarrow{c}$ | $c \xleftarrow{\$} C$ |
| For $i = 0, \ldots, l-1$ : | | |
| $\boldsymbol{z}_i = \boldsymbol{y}_i + \sigma^i(c)\boldsymbol{r}$ | | |
| If $\mathsf{Rej}_1((\boldsymbol{z}_i), (\sigma^i(c)\boldsymbol{r}), \mathfrak{s}) = 1,$ abort | | |
| | $\xrightarrow{\boldsymbol{z}_i}$ | $\mathsf{Verify}(\boldsymbol{t}, \boldsymbol{w}_i, \vec{\gamma}_i, h, \boldsymbol{v}_i, c, \boldsymbol{z}_i)$ |

Figure 5.4: Proof of knowledge $\Pi_{\mathsf{inner}}$ of $\vec{s} \in \mathbb{Z}_q^N$ satisfying $A\vec{s} = \vec{u}$ over $\mathbb{Z}_q$.

**Algorithm 4** $\mathsf{Verify}(\boldsymbol{t}, \boldsymbol{w}_i, \vec{\gamma}_i, h, \boldsymbol{v}_i, c, \boldsymbol{z}_i)$

1: For $i = 0, \ldots, l-1$ :
2: $\quad \|\boldsymbol{z}_i\|_\infty \overset{?}{\leq} B = \mathfrak{s}\sqrt{2(\lambda+\mu+2)N}$
3: $\quad \boldsymbol{B}_0 \boldsymbol{z}_i \overset{?}{=} \boldsymbol{w}_i + \sigma^i(c)\boldsymbol{t}_0$
4: $h_0 \overset{?}{=} \ldots \overset{?}{=} h_{l-1} \overset{?}{=} 0$
5: $\boldsymbol{\tau} = \sum_{\mu=0}^{l-1} \frac{1}{l} X^\mu \sum_{\nu=0}^{l-1} \sigma^\nu(\mathsf{NTT}^{-1}(NA^T\vec{\gamma}_\mu)\boldsymbol{t}_1 - \langle \vec{u}, \vec{\gamma}_\mu \rangle)$
6: For $i = 0, \ldots, l-1$ :
7: $\quad \sum_{\mu=0}^{l-1} \frac{1}{l} X^\mu \sum_{\nu=0}^{l-1} \sigma^\nu\Big(\langle \mathsf{NTT}^{-1}(NA^T\vec{\gamma}_\mu)\boldsymbol{b}_1, \boldsymbol{z}_{i-\nu \mod l} \rangle\Big) + \langle \boldsymbol{b}_2, \boldsymbol{z}_i \rangle \overset{?}{=} \boldsymbol{v}_i + \sigma^i(c)(\boldsymbol{\tau} + \boldsymbol{t}_2 - h)$

## 5.4 Putting everything together

We can now construct a zero-knowledge protocol for proving knowledge of a small $\vec{s}$ satisfying $A\vec{s} = \vec{u}$ over $\mathbb{Z}_q$ by proving knowledge of $\vec{s}$ using the protocol from Section 5.3, and proving that $\vec{s}$ is short by using the linear proof from Section 5.1 with the product proof from Section 5.2.2. This protocol produces proofs that are shorter than the ones by the protocol from Section 5.1 alone, by a factor 8. The biggest obstacle in reducing the proof size even more at this point, is the size of the commitments and commitment opening proofs.

# Chapter 6

# Proof of quadratic relations based on ABDLOP commitments

Lyubashevsky et al. presented in [16] a new proof technique for proving quadratic relations between committed values, that no longer uses the NTT coefficients. Hence there is no need to commit to a large polynomial whose NTT coefficients are the coefficients of the secret $\boldsymbol{s}$, and committing to the small secret can now be done with the Ajtai commitment scheme. There is still the need to commit to polynomials with larger coefficients, for instance the garbage polynomials, and so the ABDLOP commitment scheme is used in order to optimize the total commitment size.

Since the NTT technique is no longer used, the requirement to choose $q$ such that $X^N + 1$ splits into linear, or almost linear, factors modulo $q$ no longer applies. Hence we can rather choose $q$ such that the challenge space is larger, which reduces the soundness error. For the rest of this paper we will actually assume that $q$ is a product of $n$ odd primes, $q = q_1 \cdots q_n$ such that $q_1 < \ldots < q_n$. Usually we choose $n = 1$, which would be the same ring as previously, or $n = 2$. Further, we also assume that each $q_i$ is such that $q_i \equiv 5 \pmod 8$. Then $\mathbb{Z}_q$ contains a primitive 4-th root of unity $\zeta_i$, and no elements with order a higher power of two. By Lemma 2 we then have that $X^N + 1$ factors into two irreducible factors modulo each $q_i$, namely

$$X^N + 1 \equiv (X^{N/2} - \zeta_i)(X^{N/2} - \zeta_i^3) \pmod{q_i}.$$

The goal of this chapter is to give a protocol that proves many quadratic relations in $\boldsymbol{s}$ and proves that many polynomial evaluations in $\boldsymbol{s}$ have zero constant coefficients, for an $\boldsymbol{s}$ that we define below. Such a protocol can be used for instance to prove norms bounds on $\boldsymbol{s}$, which we will see in Chapter 7. Suppose that we have the message vectors $\boldsymbol{s}_1 \in R_q^{m_1}$ and $\boldsymbol{m} \in R_q^{\ell}$ with $\|\boldsymbol{s}_1\| \leq \alpha$, for which the prover knows a ABDLOP commitment $(\boldsymbol{t}_A, \boldsymbol{t}_B)$ to under randomness $\boldsymbol{s}_2$.

For an automorphism $\sigma \in \mathsf{Aut}(R_q)$ of degree $k$ over $R$, we for notation define

$$(\sigma^i(\boldsymbol{x}))_{i\in[k]} := (\boldsymbol{x}, \sigma(\boldsymbol{x}), \ldots, \sigma^{k-1}(\boldsymbol{x})) \in R_q^{ka}, \text{ for an arbitrary } \boldsymbol{x} \in R_q^a.$$

We can now define $\boldsymbol{s}$ to be

$$\boldsymbol{s} := \begin{bmatrix} (\sigma^i(\boldsymbol{s}_1))_{i\in[k]} \\ (\sigma^i(\boldsymbol{m}))_{i\in[k]} \end{bmatrix} \in R_q^{k(m_1+\ell)}.$$

The reason that we want to construct protocols for this $\boldsymbol{s}$ with automorphisms, is that it allows for a greater variety of applications. For instance we can set $\sigma = \sigma_{-1}$, and prove inner products

by using Lemma 8. The resulting protocol is quite complex, and so we break it down into smaller building blocks in order to understand it properly. We have three types of statements that we want to construct proofs of knowledge of $\boldsymbol{s}$ satisfying.

- *Single quadratic equation with automorphisms:* for a public $k(m_1+\ell)$-variate quadratic function $f$ over $R_q$:
$$f(\boldsymbol{s}) = 0.$$
- *Many quadratic equations with automorphisms:* for $d$ public $k(m_1+\ell)$-variate quadratic functions $f_1, \ldots, f_d$ over $R_q$:
$$f_j(\boldsymbol{s}) = 0 \text{ for } j = 1, \ldots, d.$$
- *Many quadratic equations with automorphisms and that polynomial evaluations have no constant coefficients:* for $d + D$ public $k(m_1 + \ell)$-variate quadratic functions $f_1, \ldots, f_d$ and $F_1, \ldots, F_D$ over $R_q$:
  - $f_j(\boldsymbol{s}) = 0$ for $j = 1, \ldots, d$,
  - let $x_j := F_j(\boldsymbol{s}) \in R_q$ for $j = 1, \ldots, D$. Then $\tilde{x}_1 = \cdots = \tilde{x}_D = 0$.

In Section 6.1 we explain how to construct a proof of knowledge of $\boldsymbol{s}$ satisfying a single quadratic relation. Some of the ideas from Sections 5.1 and 5.2 are used here. We also give the full security analysis of the scheme. In the next Section 6.2 we explain how the first protocol can be used to efficiently prove knowledge of $\boldsymbol{s}$ satisfying many quadratic relations. For this protocol we only prove knowledge soundness, as the remaining security properties are implicitly included in the final protocol. Lastly we in Section 6.3 show how we can extend the previous protocol to also prove that many polynomials evaluated in $\boldsymbol{s}$ have constant coefficient equal to zero, using ideas from Section 5.3. We give the full security analysis of this scheme.

## 6.1 Single quadratic equation

We now want to prove a quadratic relation in $\boldsymbol{s}$, namely that $f(\boldsymbol{s}) = 0$ for a $k(m_1 + \ell)$-variate quadratic function $f$ over $R_q$. We want to do this using the same idea as for the product proof in Section 5.2.2, by constructing a polynomial in $c$ with $f(\boldsymbol{s})$ as the quadratic term, and then prove that the quadratic term vanishes.

We start with the observation that each such quadratic function $f$ evaluated in $\boldsymbol{s}$ can be written explicitly as
$$f(\boldsymbol{s}) = \boldsymbol{s}^T \boldsymbol{R}_2 \boldsymbol{s} + \boldsymbol{r}_1^T \boldsymbol{s} + r_0 = 0,$$
where $r_0 \in R_q$, $\boldsymbol{r}_1 \in R_q^{k(m_1+\ell)}$ and $\boldsymbol{R}_2 \in R_q^{k(m_1+\ell)\times k(m_1+\ell)}$. Before we start, we recall that the prover knows a commitment $(\boldsymbol{t}_A, \boldsymbol{t}_B)$ to $(\boldsymbol{s}_1, \boldsymbol{m})$, and that the protocol $\Pi'_{\mathsf{open}}$ is used to prove that the commitments are valid.

We recall that we use the masked openings to construct the desired function in $c$. In the opening proof for the commitment, the prover sends the masked openings $\boldsymbol{z}_i = c\boldsymbol{s}_i + \boldsymbol{y}_i$ of $\boldsymbol{s}_i$ for $i = 1, 2$. This is though not the case for $\boldsymbol{m}$, so we construct a masked opening of $\boldsymbol{m}$ as
$$\boldsymbol{z}_m := c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2 = c\boldsymbol{m} - \boldsymbol{B}\boldsymbol{y}_2,$$

which is of the desired form and can be computed by the verifier. By the definition of the challenge space we for $c \in \mathcal{C}$ have that $\sigma(c) = c$. We can then define the following two vectors $\boldsymbol{y}$ and $\boldsymbol{z}$

$$\boldsymbol{y} := \begin{bmatrix} (\sigma^i(\boldsymbol{y}_1))_{i\in[k]} \\ -(\sigma^i(\boldsymbol{B}\boldsymbol{y}_2))_{i\in[k]} \end{bmatrix}, \quad \boldsymbol{z} := \begin{bmatrix} (\sigma^i(\boldsymbol{z}_1))_{i\in[k]} \\ (\sigma^i(\boldsymbol{z}_m)_{i\in[k]} \end{bmatrix} \in R_q^{k(m_1+\ell)}.$$

These vectors satisfies $\boldsymbol{z} = c\boldsymbol{s} + \boldsymbol{y}$, since

$$\begin{aligned} \boldsymbol{z} = \begin{bmatrix} (\sigma^i(\boldsymbol{z}_1))_{i\in[k]} \\ (\sigma^i(\boldsymbol{z}_m)_{i\in[k]} \end{bmatrix} &= \begin{bmatrix} (\sigma^i(c\boldsymbol{s}_1))_{i\in[k]} \\ (\sigma^i(c\boldsymbol{m}))_{i\in[k]} \end{bmatrix} + \begin{bmatrix} (\sigma^i(\boldsymbol{y}_1))_{i\in[k]} \\ -(\sigma^i(\boldsymbol{B}\boldsymbol{y}_2))_{i\in[k]} \end{bmatrix} \\ &= c \begin{bmatrix} (\sigma^i(\boldsymbol{s}_1))_{i\in[k]} \\ (\sigma^i(\boldsymbol{m}))_{i\in[k]} \end{bmatrix} + \begin{bmatrix} (\sigma^i(\boldsymbol{y}_1))_{i\in[k]} \\ -(\sigma^i(\boldsymbol{B}\boldsymbol{y}_2))_{i\in[k]} \end{bmatrix} = c\boldsymbol{s} + \boldsymbol{y}, \end{aligned}$$

by the homomorphic properties of $\sigma$. We can now use this to obtain the expressions

$$\begin{aligned} &\boldsymbol{z}^T\boldsymbol{R}_2\boldsymbol{z} = c^2\boldsymbol{s}^T\boldsymbol{R}_2\boldsymbol{s} + c(\boldsymbol{s}^T\boldsymbol{R}_2\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{R}_2\boldsymbol{s}) + \boldsymbol{y}^T\boldsymbol{R}_2\boldsymbol{y}, \\ &c\boldsymbol{r}_1^T\boldsymbol{z} = c^2\boldsymbol{r}_1^T\boldsymbol{s} + c\boldsymbol{r}_1^T\boldsymbol{y}. \end{aligned}$$

Using this, we can now derive the desired polynomial in $c$ with $f(\boldsymbol{s})$ as the quadratic term, in the following manner:

$$\boldsymbol{z}^T\boldsymbol{R}_2\boldsymbol{z} + c\boldsymbol{r}_1^T\boldsymbol{z} + c^2 r_0 = c^2(\boldsymbol{s}^T\boldsymbol{R}_2\boldsymbol{s} + \boldsymbol{r}_1^T\boldsymbol{s} + r_0) + cg_1 + g_0, \tag{6.1}$$

where the polynomials $g_1$ and $g_0$ are defined as

$$g_1 = \boldsymbol{s}^T\boldsymbol{R}_2\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{R}_2\boldsymbol{s} + \boldsymbol{r}_1^T\boldsymbol{y}, \quad g_0 = \boldsymbol{y}^T\boldsymbol{R}_2\boldsymbol{y}. \tag{6.2}$$

We recall from Section 5.2.2 that the idea for proving that the quadratic term of this polynomial vanishes, is to commit to the polynomial $g_1$, remove the masked opening of this commitment from the polynomial, and let the verifier check if the resulting polynomial equals only its constant term. Concretely, the prover commits to $g_1$ by $t = \boldsymbol{B}_{\mathsf{ext}}^T\boldsymbol{s}_2 + g_1$, and we let the verifier check if for $f = ct - \boldsymbol{B}_{\mathsf{ext}}^T\boldsymbol{z}_2 = cg_1 + \boldsymbol{B}_{\mathsf{ext}}^T\boldsymbol{y}_2$, we have

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

where $v := g_0 + \boldsymbol{B}_{\mathsf{ext}}^T\boldsymbol{y}_2$ is the resulting constant term of the polynomial. This would then prove that $f(\boldsymbol{s}) = \boldsymbol{s}^T\boldsymbol{R}_2\boldsymbol{s} + \boldsymbol{r}_1^T\boldsymbol{s} + r_0 = 0$. The complete proof of knowledge of $\boldsymbol{s}$ satisfying a single quadratic equation, with commitment opening proof, is given as $\Pi^{(2)}$ in Figure 6.1.

We note here that the commitment to $g_1$ is constructed as an additional commitment under the same randomness using $\boldsymbol{B}_{\mathsf{ext}}$. This means that the protocol is not zero-knowledge, but better modeled as a commit-and-prove protocol. We now give the complete security analysis.

**Theorem 14.** Suppose that $\mathfrak{s}_1 = \gamma_1\alpha\eta$ and $\mathfrak{s}_2 = \gamma_2\nu\eta\sqrt{m_2 N}$ for some $\gamma_1, \gamma_2 > 0$. Then the protocol $\Pi^{(2)}$ is complete, in the sense that if $m_1, m_2 \geq 640/N$ then the honest prover convinces the honest verifier with probability

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

*Proof.* This Theorem follows directly from Theorem 10, and that the final verification equation will hold for an honest prover, by the discussion above. □

$\underline{\Pi^{(2)}}$

Public Information: $\boldsymbol{A}_1 \in R_q^{n\times m_1}$, $\boldsymbol{A}_2 \in R_q^{n\times m_2}$, $\boldsymbol{B} \in R_q^{\ell\times m_2}, \boldsymbol{B}_{\mathsf{ext}} \in R_q^{m_2}$.
$r_0 \in R_q, \boldsymbol{r}_1 \in R_q^{k(m_1+\ell)}, \boldsymbol{R}_2 \in R_q^{k(m_1+\ell)\times k(m_1+\ell)}$ such that $f(\boldsymbol{s}) = \boldsymbol{s}^T\boldsymbol{R}_2\boldsymbol{s}+\boldsymbol{r}_1^T\boldsymbol{s}+r_0 = 0, \sigma \in \mathsf{Aut}(R_q)$
Prover's Information: $(\boldsymbol{s}_1, \boldsymbol{m}) \in R_q^{m_1+\ell}$ so that $\|\boldsymbol{s}_1\| \le \alpha$, $\boldsymbol{s}_2 \xleftarrow{\$} \chi^{m_2}$
Commitment: $\boldsymbol{t} = \begin{bmatrix}\boldsymbol{t}_A\\ \boldsymbol{t}_B\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{s}_1 + \begin{bmatrix}\boldsymbol{A}_2\\ \boldsymbol{B}\end{bmatrix}\boldsymbol{s}_2 + \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{m}\end{bmatrix}$

| Prover | | Verifier |
|---|---|---|
| $\boldsymbol{s} := \begin{bmatrix}(\sigma^i(\boldsymbol{s}_1))_{i\in[k]}\\ (\sigma^i(\boldsymbol{m}))_{i\in[k]}\end{bmatrix}$<br>$\boldsymbol{y}_1 \xleftarrow{\$} D_{R^{m_1},\mathfrak{s}_1}$<br>$\boldsymbol{y}_2 \xleftarrow{\$} D_{R^{m_2},\mathfrak{s}_2}$<br>$\boldsymbol{w} := \boldsymbol{A}_1\boldsymbol{y}_1 + \boldsymbol{A}_2\boldsymbol{y}_2$<br>$\boldsymbol{y} := \begin{bmatrix}(\sigma^i(\boldsymbol{y}_1))_{i\in[k]}\\ -(\sigma^i(\boldsymbol{B}\boldsymbol{y}_2))_{i\in[k]}\end{bmatrix}$<br>$g_1 := \boldsymbol{s}^T\boldsymbol{R}_2\boldsymbol{y} + \boldsymbol{y}^T\boldsymbol{R}_2\boldsymbol{s} + \boldsymbol{r}_1^T\boldsymbol{y}$<br>$t := \boldsymbol{B}_{\mathsf{ext}}^T\boldsymbol{s}_2 + g_1$<br>$v := \boldsymbol{y}^T\boldsymbol{R}_2\boldsymbol{y} + \boldsymbol{B}_{\mathsf{ext}}^T\boldsymbol{y}_2$ | | |
| | $\xrightarrow{\boldsymbol{w}, t, v}$ | |
| | | $c \xleftarrow{\$} \mathcal{C}$ |
| | $\xleftarrow{c}$ | |
| $\boldsymbol{z}_1 = c\boldsymbol{s}_1 + \boldsymbol{y}_1$<br>$\boldsymbol{z}_2 = c\boldsymbol{s}_2 + \boldsymbol{y}_2$<br>for $i = 1, 2$ :<br>if $\mathsf{Rej}_i(\boldsymbol{z}_i, c\boldsymbol{s}_i, \mathfrak{s}_i) = 1$<br>then abort | | |
| | $\xrightarrow{\boldsymbol{z}_1, \boldsymbol{z}_2}$ | |
| | | $\boldsymbol{z} := \begin{bmatrix}(\sigma^i(\boldsymbol{z}_1))_{i\in[k]}\\ (\sigma^i(c\boldsymbol{t}_B - \boldsymbol{B}\boldsymbol{z}_2))_{i\in[k]}\end{bmatrix}$<br>$f := ct - \boldsymbol{B}_{\mathsf{ext}}^T\boldsymbol{z}_2$<br>Accept iff:<br>1. $\|\boldsymbol{z}_1\| \le \mathfrak{s}_1\sqrt{2m_1N}$<br>2. $\|\boldsymbol{z}_2\| \le \mathfrak{s}_2\sqrt{2m_2N}$<br>3. $\boldsymbol{A}_1\boldsymbol{z}_1 + \boldsymbol{A}_2\boldsymbol{z}_2 - c\boldsymbol{t}_A = \boldsymbol{w}$<br>4. $\boldsymbol{z}^T\boldsymbol{R}_2\boldsymbol{z} + c\boldsymbol{r}_1^T\boldsymbol{z} + c^2\boldsymbol{r}_0 - f = v$ |

Figure 6.1: Proof of knowledge $\Pi^{(2)}((\boldsymbol{s}_2, \boldsymbol{s}_1, \boldsymbol{m}), \sigma, f)$ of $((\boldsymbol{s}_1, \boldsymbol{m}), \boldsymbol{s}_2, \bar{c}) \in R_q^{m_1+\ell} \times R_q^{m_2} \times \overline{\mathcal{C}}$ that satisfy $\boldsymbol{A}_1\boldsymbol{s}_1 + \boldsymbol{A}_2\boldsymbol{s}_2 = \boldsymbol{t}_A$, $\boldsymbol{B}\boldsymbol{s}_2 + \boldsymbol{m} = \boldsymbol{t}_B$, $\|\bar{c}\boldsymbol{s}_i\| \le 2\mathfrak{s}_i\sqrt{2m_iN}$ for $i = 1, 2$ and $f\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$.

**Theorem 15.** Let $\mathfrak{s}_1 = \gamma_1 \alpha \eta$ and $\mathfrak{s}_2 = \gamma_2 \nu \eta \sqrt{m_2 N}$ for some $\gamma_1, \gamma_2 > 0$. Then the protocol $\Pi^{(2)}$ is commit-and-prove simulatable, meaning that there exists a simulator $\mathcal{S}$ that, without access to private information $(\boldsymbol{s}_1, \boldsymbol{m})$, outputs a simulation of a commitment $(\boldsymbol{t}_A, \boldsymbol{t}_B)$ along with a non-aborting transcript of the protocol between the prover and verifier such that for every algorithm $\mathcal{A}$ that has advantage $\varepsilon$ in distinguishing the simulated commitment and transcript from the real commitment and transcript, whenever the prover does not abort, there is an algorithm $\mathcal{A}'$ with the same running time that has advantage $\varepsilon/2 - 2^{-128}$ in solving the $\mathsf{Extended\text{-}MLWE}_{n+\ell+1, m_2-n-\ell-1, \chi, \mathcal{C}, \mathfrak{s}_2}$ problem.

*Proof.* In order to construct such a simulator $\mathcal{S}$, we start by defining a simulator $\mathcal{S}_0$ that knows the secret information $\boldsymbol{s}_1, \boldsymbol{m}$. It operates in the following manner. When given a challenge $c \xleftarrow{\$} \mathcal{C}$, it honestly computes the commitment $(\boldsymbol{t}_A, \boldsymbol{t}_B, t)$ under randomness $\boldsymbol{s}_2 \xleftarrow{\$} \chi^{m_2}$. In order to simulate the rest of the transcript, it samples masked openings $\boldsymbol{z}_1 \xleftarrow{\$} D_{R^{m_1 N}, \mathfrak{s}_1}, \boldsymbol{z}_2 \xleftarrow{\$} D_{R^N, \mathfrak{s}_2}$ conditioned on $\langle \boldsymbol{s}_2, \boldsymbol{z}_2 \rangle$, otherwise $\mathsf{Rej}_2$ would abort, and computes $\boldsymbol{w} := \boldsymbol{A}_1 \boldsymbol{z}_1 + \boldsymbol{A}_2 \boldsymbol{z}_2 - c\boldsymbol{t}_A$ and $v := \boldsymbol{z}^T \boldsymbol{R}_2 \boldsymbol{z} + c\boldsymbol{r}_1^T \boldsymbol{z} + c^2 \boldsymbol{r}_0 - ct + \boldsymbol{B}_{\mathsf{ext}}^T \boldsymbol{z}_2$. Hence the verification equations will hold, and according to Lemma 4 the distribution of the commitment and a transcript output by $\mathcal{S}_0$ is within a statistical distance of $2^{-128}$ to the one in the actual non-aborting protocol.

We also define the simulator $\mathcal{S}_1$ that knows the secret information $\boldsymbol{s}_1, \boldsymbol{m}$ as follows. It runs identically as $\mathcal{S}_0$, but samples $\boldsymbol{u} \xleftarrow{\$} R_q^{n+\ell+1}$ and sets

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \\ t \end{bmatrix} = \boldsymbol{u} + \begin{bmatrix} \boldsymbol{A}_1 \boldsymbol{s}_1 \\ \boldsymbol{m} \\ g_1 \end{bmatrix},$$

instead of computing the commitment honestly. We can now show that if there is an adversary $\mathcal{A}$ that can distinguish between the outputs of $\mathcal{S}_0$ and $\mathcal{S}_1$ with probability $\varepsilon$, then we can construct an adversary $\mathcal{B}$ with advantage at least $\varepsilon/2$ in solving the $\mathsf{Extended\text{-}MLWE}_{n+\ell+1, m_2-n-\ell-1, \chi, \mathcal{C}, \mathfrak{s}_2}$ problem.

$\mathcal{B}$ is constructed as follows. Given an $\mathsf{Extended\text{-}MLWE}$ instance $(\boldsymbol{C}, \boldsymbol{u}, \boldsymbol{z}_2, s)$, where

$$\boldsymbol{C} := \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \\ \boldsymbol{B}_{\mathsf{ext}} \end{bmatrix},$$

it runs identically as $\mathcal{S}_1$, and outputs the commitment and transcript to $\mathcal{A}$. We notice that if $\boldsymbol{u} = \boldsymbol{C}\boldsymbol{s}_2$, then the output of $\mathcal{B}$ is the same as the one from $\mathcal{S}_0$, but if $\boldsymbol{u}$ was uniformly random, then the output of $\mathcal{B}$ is the same as the one from $\mathcal{S}_1$. So conditioned on $s = 1$, which has probability at least $1/2$, $\mathcal{B}$ solves the $\mathsf{Extended\text{-}MLWE}$ problem with probability $\varepsilon$.

The commit-and-prove simulator $\mathcal{S}$, without access to private information, can now be defined to run identically to $\mathcal{S}_1$, but by directly sampling $(\boldsymbol{t}_A, \boldsymbol{t}_B, t) \xleftarrow{\$} R_q^{n+\ell+1}$ instead. Since the commitment computed by $\mathcal{S}_1$ looks uniformly random, the output distributions of $\mathcal{S}$ and $\mathcal{S}_1$ are identical. Hence if there an adversary $\mathcal{A}$ that can distinguish between the outputs of $\mathcal{S}$ and the real protocol, there is an adversary $\mathcal{A}'$ that that has advantage $\varepsilon/2 - 2^{-128}$ in solving the $\mathsf{Extended\text{-}MLWE}_{n+\ell+1, m_2-n-\ell-1, \chi, \mathcal{C}, \mathfrak{s}_2}$ problem. □

**Theorem 16.** The protocol $\Pi^{(2)}$ is knowledge sound, meaning that there is an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a probabilistic prover $\mathcal{P}^*$, which

convinces $\mathcal{V}$ with probability $\varepsilon \geq 2/|\mathcal{C}|$, the extractor $\mathcal{E}$ with probability at least $\varepsilon - 2/|\mathcal{C}|$ either outputs $(\overline{\boldsymbol{s}}_2, \overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}) \in R_q^{m_1+m_2+\ell}$ and $\overline{c} \in R_q^{\times}$ such that

- $\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \overline{\boldsymbol{s}}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \overline{\boldsymbol{s}}_2 + \begin{bmatrix} \boldsymbol{0} \\ \overline{\boldsymbol{m}} \end{bmatrix}$
- $\|\overline{c}\|_\infty \leq 2\omega$
- $\|\overline{c}\overline{\boldsymbol{s}}_1\| \leq 2\mathfrak{s}_1\sqrt{2m_1N}$ and $\|\overline{c}\overline{\boldsymbol{s}}_2\| \leq 2\mathfrak{s}_2\sqrt{2m_2N}$
- $f\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$

or an $\mathsf{MSIS}_{n,m_1+m_2,B}$ solution for $\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{bmatrix}$ in expected time at most $3T$ where running $\mathcal{P}^*$ once is assumed to take at most $T$ time and $B = 8\eta\sqrt{(\mathfrak{s}_1\sqrt{2m_1N})^2 + (\mathfrak{s}_2\sqrt{2m_2N})^2}$.

*Proof.* In order to prove knowledge soundness we use the collision game strategy introduced in Section 2.3.4. We let $H \in \{0,1\}^{R\times|\mathcal{C}|}$ be the binary matrix where the $R$ rows correspond to the prover's randomness and $|\mathcal{C}|$ columns correspond to the different choices for the challenge $c$, and let $H(r,c)$ denote the entry corresponding to randomness $r$ and challenge $c \in \mathcal{C}$. By assumption there is a fraction of $\varepsilon$ 1-entries in $H$. We define the following extractor.

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

If we assume that $\mathcal{E}$ can check values of each entry in $H$ in time most $T$, then Lemma 1 states that the expected time of $\mathcal{E}$ is at most $3T$ and that $\mathcal{E}$ extracts three valid transcripts with probability at least $\varepsilon - 2/|\mathcal{C}|$. We denote the valid transcripts as

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

We start by focusing on $\mathsf{tr}^{(0)}$ and $\mathsf{tr}^{(1)}$, and define

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

By definition of the challenge space we have $\|\overline{c}\|_\infty \leq 2\omega$, and by construction we then have $\|\overline{c}\overline{\boldsymbol{s}}_1\| \leq 2\mathfrak{s}_1\sqrt{2m_1N}$, $\|\overline{c}\overline{\boldsymbol{s}}_2\| \leq 2\mathfrak{s}_2\sqrt{2m_2N}$, since $\|\boldsymbol{z}_i\| \leq \mathfrak{s}_i\sqrt{2m_iN}$ for accepting transcripts. From the verification equations we get that $\boldsymbol{A}_1\boldsymbol{z}_1^{(i)} + \boldsymbol{A}_2\boldsymbol{z}_2^{(i)} = \boldsymbol{w} + c^{(i)}\boldsymbol{t}_A$ for $i = 0, 1$, and thus we get

$$\begin{aligned} \boldsymbol{A}_1\overline{\boldsymbol{s}}_1 + \boldsymbol{A}_2\overline{\boldsymbol{s}}_2 &= \boldsymbol{A}_1(\boldsymbol{z}_1^{(1)}/\overline{c} - \boldsymbol{z}_1^{(0)}/\overline{c}) + \boldsymbol{A}_2(\boldsymbol{z}_2^{(1)}/\overline{c} - \boldsymbol{z}_2^{(0)}/\overline{c}) \\ &= c^{(1)}\boldsymbol{t}_A/\overline{c} - c^{(0)}\boldsymbol{t}_A/\overline{c} \\ &= \boldsymbol{t}_A. \end{aligned}$$

We also define the extracted values $\overline{\boldsymbol{m}} := \boldsymbol{t}_B - \boldsymbol{B}\overline{\boldsymbol{s}}_2$ and $\overline{g}_1 := t - \boldsymbol{B}_{\mathsf{ext}}^T\overline{\boldsymbol{s}}_2$, so that we get

$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \\ t \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \\ 0 \end{bmatrix} \overline{\boldsymbol{s}}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \\ \boldsymbol{B}_{\mathsf{ext}}^T \end{bmatrix} \overline{\boldsymbol{s}}_2 + \begin{bmatrix} \boldsymbol{0} \\ \overline{\boldsymbol{m}} \\ \overline{g}_1 \end{bmatrix}.$$

We can now let $\overline{\boldsymbol{y}}_i := \boldsymbol{z}_i^{(1)} - c^{(1)}\overline{\boldsymbol{s}}_i = \boldsymbol{z}_i^{(0)} - c^{(0)}\overline{\boldsymbol{s}}_i$ for $i = 1, 2$. Considering the third transcript $\mathsf{tr}^{(2)}$ we can define $\boldsymbol{y}_i^{(2)} := \boldsymbol{z}_i^{(2)} - c^{(2)}\overline{\boldsymbol{s}}_i$ for $i = 1, 2$. We then have $(\boldsymbol{z}_i^{(1)} - \overline{\boldsymbol{y}}_i)/c^{(1)} = \overline{\boldsymbol{s}}_i = (\boldsymbol{z}_i^{(2)} - \overline{\boldsymbol{y}}_i^{(2)})/c^{(2)}$.

We assume that $(\overline{\boldsymbol{y}}_1, \overline{\boldsymbol{y}}_2) \neq (\boldsymbol{y}_1^{(2)}, \boldsymbol{y}_2^{(2)})$, and see that we must then have

$$\boldsymbol{A}_1(\boldsymbol{z}_1^{(1)} - \overline{\boldsymbol{y}}_1)/c^{(1)} + \boldsymbol{A}_2(\boldsymbol{z}_2^{(1)} - \overline{\boldsymbol{y}}_2)/c^{(1)} = \boldsymbol{A}_1(\boldsymbol{z}_1^{(2)} - \overline{\boldsymbol{y}}_1^{(2)})/c^{(2)} + \boldsymbol{A}_2(\boldsymbol{z}_2^{(2)} - \overline{\boldsymbol{y}}_2^{(2)})/c^{(2)}.$$

We multiply this equation by $c^{(1)}c^{(2)}$ and get

$$\boldsymbol{A}_1((\boldsymbol{z}_1^{(1)} - \overline{\boldsymbol{y}}_1) + \boldsymbol{A}_2(\boldsymbol{z}_2^{(1)} - \overline{\boldsymbol{y}}_2))c^{(2)} = \boldsymbol{A}_1((\boldsymbol{z}_1^{(2)} - \overline{\boldsymbol{y}}_1^{(2)}) + \boldsymbol{A}_2(\boldsymbol{z}_2^{(2)} - \overline{\boldsymbol{y}}_2^{(2)}))c^{(1)},$$

which can now be rearranged into

$$\boldsymbol{A}_1\big((\boldsymbol{z}_1^{(1)} - \overline{\boldsymbol{y}}_1)c^{(2)} - (\boldsymbol{z}_1^{(2)} - \overline{\boldsymbol{y}}_1^{(2)})c^{(1)}\big) + \boldsymbol{A}_2\big((\boldsymbol{z}_2^{(1)} - \overline{\boldsymbol{y}}_2)c^{(2)} - (\boldsymbol{z}_2^{(2)} - \overline{\boldsymbol{y}}_2^{(2)})c^{(1)}\big) = 0.$$

We know that $\|(\boldsymbol{z}_1^{(1)} - \overline{\boldsymbol{y}}_1)c^{(2)} - (\boldsymbol{z}_1^{(2)} - \overline{\boldsymbol{y}}_1^{(2)})c^{(1)}\| \leq 8\eta\mathfrak{s}_1\sqrt{2m_1N}$ and $\|(\boldsymbol{z}_2^{(1)} - \overline{\boldsymbol{y}}_2)c^{(2)} - (\boldsymbol{z}_2^{(2)} - \overline{\boldsymbol{y}}_2^{(2)})c^{(1)}\| \leq 8\eta\mathfrak{s}_2\sqrt{2m_2N}$, since $\|\boldsymbol{z}_i^{(j)}\| \leq 2\mathfrak{s}_i\sqrt{2m_iN}$ and multiplication by $c$ increases the norm by at most a factor $2\eta$, by Lemma 12. Hence we either have an $\mathsf{MSIS}_{n,m_1+m_1,B}$ solution for $\begin{bmatrix}\boldsymbol{A}_1 & \boldsymbol{A}_2\end{bmatrix}$ with $B = 8\eta\sqrt{(\mathfrak{s}_1\sqrt{2m_1N})^2 + (\mathfrak{s}_2\sqrt{2m_2N})^2}$, or we have that $(\overline{\boldsymbol{y}}_1, \overline{\boldsymbol{y}}_2) = (\boldsymbol{y}_1^{(2)}, \boldsymbol{y}_2^{(2)})$. The former case is as in the statement of the Theorem, and so we focus on the latter case.

We define, as in the protocol

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

Since $\mathsf{tr}^{(0)}, \mathsf{tr}^{(1)}$ and $\mathsf{tr}^{(2)}$ are valid transcripts, we get from the verification equations that

$$\boldsymbol{z}^{(i)^T}\boldsymbol{R}_2\boldsymbol{z}^{(i)} + c^{(i)}\boldsymbol{r}_1^T\boldsymbol{z}^{(i)} + c^{(i)^2}\boldsymbol{r}_0 - (c^{(i)}t - \boldsymbol{B}_{\mathsf{ext}}\boldsymbol{z}_2^{(i)}) = v \text{ for } i = 0, 1, 2, \tag{6.3}$$

where

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

We can now expand Equation (6.3) in the equivalent manner as we did in (6.1) and (6.2), and obtain

$$c^{(i)^2}(\overline{\boldsymbol{s}}^T\boldsymbol{R}_2\overline{\boldsymbol{s}} + \boldsymbol{r}_1^T\overline{\boldsymbol{s}} + \boldsymbol{r}_0) + c^{(i)}g_1' + g_0' = 0 \text{ for } i = 0, 1, 2,$$

where the polynomials $g_1'$ and $g_0'$ are defined as

$$\begin{aligned} g_1' &= \overline{\boldsymbol{s}}^T\boldsymbol{R}_2\overline{\boldsymbol{y}} + \overline{\boldsymbol{y}}^T\boldsymbol{R}_2\overline{\boldsymbol{s}} + \boldsymbol{r}_1^T\overline{\boldsymbol{y}} - \overline{g}_1, \\ g_0' &= \overline{\boldsymbol{y}}^T\boldsymbol{R}_2\overline{\boldsymbol{y}} + \boldsymbol{B}_{\mathsf{ext}}\overline{\boldsymbol{y}}_2 - v. \end{aligned}$$

The system of these three equations can be written in matrix form as follows

$$\begin{bmatrix}1 & c^{(0)} & c^{(0)^2} \\ 1 & c^{(1)} & c^{(1)^2} \\ 1 & c^{(2)} & c^{(2)^2}\end{bmatrix}\begin{bmatrix}g_0' \\ g_1' \\ \overline{\boldsymbol{s}}^T\boldsymbol{R}_2\overline{\boldsymbol{s}} + \boldsymbol{r}_1^T\overline{\boldsymbol{s}} + \boldsymbol{r}_0\end{bmatrix} = \begin{bmatrix}0 \\ 0 \\ 0\end{bmatrix}.$$

Since the difference of each two of the challenges $c^{(0)}, c^{(1)}, c^{(2)}$ is invertible over $R_q$, the matrix of challenges is invertible. This implies that $f(\overline{\boldsymbol{s}}) = \overline{\boldsymbol{s}}^T\boldsymbol{R}_2\overline{\boldsymbol{s}} + \boldsymbol{r}_1^T\overline{\boldsymbol{s}} + \boldsymbol{r}_0 = 0$. Hence we have proved the Theorem. □

## 6.2 Many quadratic equations

We can use the protocol $\Pi^{(2)}$ to prove that $f_j(\boldsymbol{s}) = 0$ for $d$ public $k(m_1 + \ell)$-variate quadratic functions $f_1, \ldots, f_d$ over $R_q$. The obvious approach would be to use the protocol $\Pi^{(2)}$ directly on each individual function. Then one would end up committing to $d$ garbage polynomials, which makes the total proof size large. We instead present a method that allows us to commit to only one garbage polynomial, thus significantly reducing the total proof size. The idea is that the linear combination of the functions is with high probability only zero in $\boldsymbol{s}$ if each of the functions $f_j$ are zero in $\boldsymbol{s}$.

We start the protocol by letting the verifier send challenges $\mu_1, \ldots, \mu_d \xleftarrow{\$} R_q$. From this we can construct the linear combination

$$f = \sum_{j=1}^{d} \mu_j f_j,$$

which has the property that if one of the $f_j$ is such that $f_j(\boldsymbol{s}) \neq 0$, then $f(\boldsymbol{s}) = 0$ only with probability $q_1^{-N/2}$, since $X^N + 1$ splits into two irreducible factors modulo $q_1$. So we can now run $\Pi^{(2)}$ with the function $f$, and prove that $f_j(\boldsymbol{s}) = 0$ for all $j = 1, \ldots, d$ with great certainty.

The protocol is given as $\Pi^{(2)}_{\mathsf{many}}$ in Figure 6.2. It is mainly used as a building block for the more general protocol in the next section, and so we do not present the full security analysis here, since it will be implicitly included in the next chapter. We do however consider the knowledge soundness of the protocol.

$\underline{\Pi^{(2)}_{\mathsf{many}}}$

Public Information: $\boldsymbol{A}_1 \in R_q^{n \times m_1}$, $\boldsymbol{A}_2 \in R_q^{n \times m_2}$, $\boldsymbol{B} \in R_q^{\ell \times m_2}$, $\boldsymbol{B}_{\mathsf{ext}} \in R_q^{m_2}$.
$f_1, \ldots, f_d : R_q^{k(m_1+\ell)} \to R_q, \sigma \in \mathsf{Aut}(R_q)$
Prover's Information: $(\boldsymbol{s}_1, \boldsymbol{m}) \in R_q^{m_1+\ell}$ so that $\|\boldsymbol{s}_1\| \leq \alpha$, $\boldsymbol{s}_2 \xleftarrow{\$} \chi^{m_2}$
Commitment: $\boldsymbol{t} = \begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \boldsymbol{s}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \boldsymbol{s}_2 + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{m} \end{bmatrix}$

| Prover | | Verifier |
|---|---|---|
| | | $\mu_1, \ldots, \mu_d \xleftarrow{\$} R_q$ |
| | $\xleftarrow{\mu_1, \ldots, \mu_d}$ | |
| $f := \sum_{j=1}^{d} \mu_j f_j$ | | |
| Run $\Pi^{(2)}((\boldsymbol{s}_2, \boldsymbol{s}_1, \boldsymbol{m}), \sigma, f)$ | | |

Figure 6.2: Proof of knowledge $\Pi^{(2)}_{\mathsf{many}}((\boldsymbol{s}_2, \boldsymbol{s}_1, \boldsymbol{m}), \sigma, (f_1, \ldots, f_d))$ of $((\boldsymbol{s}_1, \boldsymbol{m}), \boldsymbol{s}_2, \bar{c}) \in R_q^{m_1+\ell} \times R_q^{m_2} \times \overline{\mathcal{C}}$ that satisfy $\boldsymbol{A}_1\boldsymbol{s}_1 + \boldsymbol{A}_2\boldsymbol{s}_2 = \boldsymbol{t}_A$, $\boldsymbol{B}\boldsymbol{s}_2 + \boldsymbol{m} = \boldsymbol{t}_B$, $\|\bar{c}\boldsymbol{s}_i\| \leq 2\mathfrak{s}_i\sqrt{2m_iN}$ for $i = 1, 2$ and $f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$ for $j \in [d]$.

**Theorem 17.** The protocol $\Pi^{(2)}_{\mathsf{many}}$ is knowledge sound, meaning that there exists an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a probabilistic prover $\mathcal{P}^*$, which convinces $\mathcal{V}$ with probability $\varepsilon \geq 2/|\mathcal{C}| + q_1^{-N/2}$, the extractor $\mathcal{E}$ with probability at least $\varepsilon - 2/|\mathcal{C}| - q_1^{-N/2}$ either outputs $(\overline{\boldsymbol{s}}_2, \overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}) \in R_q^{m_1+m_2+\ell}$ and $\overline{c} \in R_q^{\times}$ such that

- $\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \overline{\boldsymbol{s}}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \overline{\boldsymbol{s}}_2 + \begin{bmatrix} \boldsymbol{0} \\ \overline{\boldsymbol{m}} \end{bmatrix}$
- $\|\overline{c}\|_\infty \leq 2\omega$
- $\|\overline{c}\overline{\boldsymbol{s}}_1\| \leq 2\mathfrak{s}_1\sqrt{2m_1N}$ and $\|\overline{c}\overline{\boldsymbol{s}}_2\| \leq 2\mathfrak{s}_2\sqrt{2m_2N}$
- for all $j \in [d]$, $f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$

or an $\mathsf{MSIS}_{n,m_1+m_2,B}$ solution for $\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{bmatrix}$ in expected time at most $6T$ where running $\mathcal{P}^*$ once is assumed to take at most $T$ time and $B = 8\eta\sqrt{(\mathfrak{s}_1\sqrt{2m_1N})^2 + (\mathfrak{s}_2\sqrt{2m_2N})^2}$.

*Proof.* We let $\mathcal{P}^*$ be such a probabilistic prover as described in the Theorem, that convinces the verifier with probability $\varepsilon \geq 2/|\mathcal{C}| + q_1^{-N/2}$ and runs in time at most $T$. In order to construct the desired extractor $\mathcal{E}$ we first define a deterministic algorithm $\mathcal{A}$ that utilize the $\Pi^{(2)}$-extractor from Theorem 16. It is defined as follows. Given randomness $\rho \in \mathfrak{R}$ and challenge $\boldsymbol{\mu} \in R_q^d$, $\mathcal{A}(\rho, \boldsymbol{\mu})$ runs the extractor $\mathcal{E}^*(\rho)$ from Theorem 16 with randomness $\rho$. $\mathcal{E}^*(\rho)$ will then call $\mathcal{P}^*(\boldsymbol{\mu})$ in a black-box manner, and will according to Theorem 16 with a certain probability either extract a valid $\mathsf{MSIS}$ solution or output $(\overline{\boldsymbol{s}}_2, \overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{c})$ such that

- $\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \overline{\boldsymbol{s}}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \overline{\boldsymbol{s}}_2 + \begin{bmatrix} \boldsymbol{0} \\ \overline{\boldsymbol{m}} \end{bmatrix}$
- $\|\overline{c}\|_\infty \leq 2\omega$
- $\|\overline{c}\overline{\boldsymbol{s}}_1\| \leq 2\mathfrak{s}_1\sqrt{2m_1N}$ and $\|\overline{c}\overline{\boldsymbol{s}}_2\| \leq 2\mathfrak{s}_2\sqrt{2m_2N}$
- $\sum_{j=1}^{d} \mu_j f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$

We say that $\mathcal{A}$ succeeds if the latter holds true. We also know from Theorem 16 that the expected run time of $\mathcal{A}$ for any $\boldsymbol{\mu}$ and $\rho \xleftarrow{\$} \mathfrak{R}$ is at most $3T$ and the probability that $\mathcal{A}$ succeeds is at least $\varepsilon - 2/|\mathcal{C}|$, when we assume that extraction of a valid $\mathsf{MSIS}$ solution never occurs.

In order to define the extractor $\mathcal{E}$ we need some definitions and results. We define $H \subseteq \mathfrak{R} \times R_q^d$ to be the set of pairs $(\rho, \boldsymbol{\mu})$ such that $\mathcal{A}(\rho, \boldsymbol{\mu})$ succeeds. Then we can define $H(\rho)$ to be the set of all $\boldsymbol{\mu}$ for which $(\rho, \boldsymbol{\mu}) \in H$. When $\mathcal{A}(\rho, \boldsymbol{\mu})$ succeeds for a fixed $(\rho, \boldsymbol{\mu}) \in H$, we denote the output as $(\overline{\boldsymbol{s}}_2^{(\rho,\boldsymbol{\mu})}, \overline{\boldsymbol{s}}_1^{(\rho,\boldsymbol{\mu})}, \overline{\boldsymbol{m}}^{(\rho,\boldsymbol{\mu})})$. We can then define, in the same manner as previous,

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

Finally, we can define the set of $(\rho, \boldsymbol{\mu}) \in H$ for which at least one of the $f_j$ are non-zero in $\overline{\boldsymbol{s}}^{(\rho,\boldsymbol{\mu})}$,

$$H' := \Big\{(\rho, \boldsymbol{\mu}) \in H \mid \exists\ j \in [d] \text{ s.t. } f_j(\overline{\boldsymbol{s}}^{(\rho,\boldsymbol{\mu})}) \neq 0\Big\}.$$

Before we define $\mathcal{E}$, we present a Lemma on the probability that $f(\overline{\boldsymbol{s}}) = 0$ if one of the $f_j$ are non-zero in $\overline{\boldsymbol{s}}$:

**Lemma 16.** If $(\rho, \boldsymbol{\mu}) \in H$ then $\Pr_{\boldsymbol{\mu}' \xleftarrow{\$} R_q^d}[(\rho, \boldsymbol{\mu}') \in H] > 0$. Moreover, if $(\rho, \boldsymbol{\mu}) \in H'$ then

$$\Pr_{\boldsymbol{\mu}' \xleftarrow{\$} R_q^d}\left[\sum_{j=1}^{d} \mu'_j f_j\left(\overline{\boldsymbol{s}}^{(\rho,\boldsymbol{\mu})}\right) = 0\right] \leq q_1^{-N/2}.$$

*Proof.* For the first part, we observe that if $(\rho, \boldsymbol{\mu}) \in H$, then we have

$$\Pr_{\boldsymbol{\mu}' \xleftarrow{\$} R_q^d}[(\rho, \boldsymbol{\mu}') \in H] \geq \Pr_{\boldsymbol{\mu}' \xleftarrow{\$} R_q^d}[\boldsymbol{\mu}' = \boldsymbol{\mu}] > 0.$$

For the second part, if we assume that $f_j(\overline{\boldsymbol{s}}^{(\rho,\boldsymbol{\mu})}) \neq 0$ for some $j$, then the probability over $\mu'_j \xleftarrow{\$} R_q$ that $\mu'_j f_j(\overline{\boldsymbol{s}}^{(\rho,\boldsymbol{\mu})}) = a$ for any fixed $a \in R_q$ is at most $q_1^{-N/2}$, since $X^N + 1$ splits into two irreducible factors modulo $q_1$. Hence the claim follows. □

We are now ready to define the extractor $\mathcal{E}$, and it is constructed in the following manner.

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

When $\mathcal{E}$ does not abort, this procedure will yield two extracted tuples $x = (\overline{\boldsymbol{s}}_2, \overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{c})$ and $x' = (\overline{\boldsymbol{s}}'_2, \overline{\boldsymbol{s}}'_1, \overline{\boldsymbol{m}}', \overline{c}')$ since it runs until $\mathcal{A}$ succeeds, twice. We say that $\mathcal{E}$ succeeds if it extracts two such tuples such that one of the below conditions holds.

- $(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{s}}_2) \neq (\overline{\boldsymbol{s}}'_1, \overline{\boldsymbol{s}}'_2)$, $\max(\|\overline{c}\|_\infty, \|\overline{c}'\|_\infty) \leq 2\omega$ and $\max(\|\overline{c}\overline{\boldsymbol{s}}_i\|, \|\overline{c}'\overline{\boldsymbol{s}}'_i\|) \leq 2\mathfrak{s}_i\sqrt{2m_iN}$ for $i = 1, 2$, and
$$\begin{bmatrix}\boldsymbol{A}_1 \\ \boldsymbol{0}\end{bmatrix}\overline{\boldsymbol{s}}_1 + \begin{bmatrix}\boldsymbol{A}_2 \\ \boldsymbol{B}\end{bmatrix}\overline{\boldsymbol{s}}_2 + \begin{bmatrix}\boldsymbol{0} \\ \overline{\boldsymbol{m}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{t}_A \\ \boldsymbol{t}_B\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_1 \\ \boldsymbol{0}\end{bmatrix}\overline{\boldsymbol{s}}'_1 + \begin{bmatrix}\boldsymbol{A}_2 \\ \boldsymbol{B}\end{bmatrix}\overline{\boldsymbol{s}}'_2 + \begin{bmatrix}\boldsymbol{0} \\ \overline{\boldsymbol{m}}'\end{bmatrix}.$$
- For all $j \in [d]$, $f_j\left((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\right) = 0$ and $\|\overline{c}\|_\infty \leq 2\omega$ and $\|\overline{c}\overline{\boldsymbol{s}}_1\| \leq 2\mathfrak{s}_1\sqrt{2m_1N}$ and $\|\overline{c}\overline{\boldsymbol{s}}_2\| \leq 2\mathfrak{s}_2\sqrt{2m_2N}$ and
$$\begin{bmatrix}\boldsymbol{t}_A \\ \boldsymbol{t}_B\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_1 \\ \boldsymbol{0}\end{bmatrix}\overline{\boldsymbol{s}}_1 + \begin{bmatrix}\boldsymbol{A}_2 \\ \boldsymbol{B}\end{bmatrix}\overline{\boldsymbol{s}}_2 + \begin{bmatrix}\boldsymbol{0} \\ \overline{\boldsymbol{m}}\end{bmatrix}.$$

If the first case holds true, we break the binding property of the commitment scheme, which gives us the relevant $\mathsf{MSIS}$ solution. If the second case holds true, we have extracted the desired $(\overline{\boldsymbol{s}}_2, \overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{c})$. We will now present and prove two claims about $\mathcal{E}$, which completes the proof.

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

*Proof.* Let $X$ be the expected number of calls to $\mathcal{A}$, and let $\varepsilon$ be the probability that $\mathcal{A}(\rho, \boldsymbol{\mu})$ succeeds for random $\rho$ and $\boldsymbol{\mu}$. We define the event $E$ that $\mathcal{A}$ succeeds in the first step. We then get, by the law of total expectation, that

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

Hence the claim holds. □

So the expected run time of $\mathcal{E}$ is thus at most $6T$, twice the expected run time of $\mathcal{A}$.

*Claim.* The probability that $\mathcal{E}$ succeeds is at least $\varepsilon - 2/|\mathcal{C}| - q_1^{-N/2}$.

*Proof.* As we discussed, $\mathcal{E}$ terminates with probability at least $\varepsilon - 2/|\mathcal{C}|$. So, we assume $\mathcal{E}$ terminates and let $(\boldsymbol{\mu}, \overline{\boldsymbol{s}}_2, \overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{c})$ and $(\boldsymbol{\mu}', \overline{\boldsymbol{s}}'_2, \overline{\boldsymbol{s}}'_1, \overline{\boldsymbol{m}}', \overline{c}')$ be the respective outputs of $\mathcal{A}$ in the first and second step of $\mathcal{E}$. We can divide the possible such outputs into three disjoint cases. The first case stems from the first of the success conditions for $\mathcal{E}$, as described above. The two final cases stems from the second success condition for $\mathcal{E}$, one of them such that all $f_j$ are zero in $\overline{\boldsymbol{s}}$ and the other one such that at least one of the $f_j$ are non-zero in $\overline{\boldsymbol{s}}$.

*Case 1*:

- $(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{\boldsymbol{s}}_2) \neq (\overline{\boldsymbol{s}}'_1, \overline{\boldsymbol{m}}', \overline{\boldsymbol{s}}'_2)$
- $\sum_{j=1}^{d} \mu_j f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$ and $\sum_{j=1}^{d} \mu'_j f_j\big((\sigma^i(\overline{\boldsymbol{s}}'_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}'))_{i\in[k]}\big) = 0$
- $\max(\|\overline{c}\|_\infty, \|\overline{c}'\|_\infty) \leq 2\omega$ and $\max(\|\overline{c}\overline{\boldsymbol{s}}_i\|, \|\overline{c}'\overline{\boldsymbol{s}}'_i\|) \leq 2\mathfrak{s}_i\sqrt{2m_iN}$ for $i = 1, 2$
- $\begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{0}\end{bmatrix}\overline{\boldsymbol{s}}_1 + \begin{bmatrix}\boldsymbol{A}_2\\ \boldsymbol{B}\end{bmatrix}\overline{\boldsymbol{s}}_2 + \begin{bmatrix}\boldsymbol{0}\\ \overline{\boldsymbol{m}}\end{bmatrix} = \begin{bmatrix}\boldsymbol{t}_A\\ \boldsymbol{t}_B\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{0}\end{bmatrix}\overline{\boldsymbol{s}}'_1 + \begin{bmatrix}\boldsymbol{A}_2\\ \boldsymbol{B}\end{bmatrix}\overline{\boldsymbol{s}}'_2 + \begin{bmatrix}\boldsymbol{0}\\ \overline{\boldsymbol{m}}'\end{bmatrix}$

*Case 2*:

- $(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{\boldsymbol{s}}_2) = (\overline{\boldsymbol{s}}'_1, \overline{\boldsymbol{m}}', \overline{\boldsymbol{s}}'_2)$
- $\sum_{j=1}^{d} \mu_j f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$ and $\sum_{j=1}^{d} \mu'_j f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$
- $\|\overline{c}\|_\infty \leq 2\omega$ and $\|\overline{c}\overline{\boldsymbol{s}}_1\| \leq 2\mathfrak{s}_1\sqrt{2m_1N}$ and $\|\overline{c}\overline{\boldsymbol{s}}_2\| \leq 2\mathfrak{s}_2\sqrt{2m_2N}$
- $\begin{bmatrix}\boldsymbol{t}_A\\ \boldsymbol{t}_B\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{0}\end{bmatrix}\overline{\boldsymbol{s}}_1 + \begin{bmatrix}\boldsymbol{A}_2\\ \boldsymbol{B}\end{bmatrix}\overline{\boldsymbol{s}}_2 + \begin{bmatrix}\boldsymbol{0}\\ \overline{\boldsymbol{m}}\end{bmatrix}$
- for all $j \in [d]$, $f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$

*Case 3:*

- $(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{\boldsymbol{s}}_2) = (\overline{\boldsymbol{s}}'_1, \overline{\boldsymbol{m}}', \overline{\boldsymbol{s}}'_2)$
- $\sum_{j=1}^{d} \mu_j f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$ and $\sum_{j=1}^{d} \mu'_j f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$
- $\|\overline{c}\|_\infty \leq 2\omega$ and $\|\overline{c}\overline{\boldsymbol{s}}_1\| \leq 2\mathfrak{s}_1\sqrt{2m_1N}$ and $\|\overline{c}\overline{\boldsymbol{s}}_2\| \leq 2\mathfrak{s}_2\sqrt{2m_2N}$
- $\begin{bmatrix}\boldsymbol{t}_A\\ \boldsymbol{t}_B\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{0}\end{bmatrix}\overline{\boldsymbol{s}}_1 + \begin{bmatrix}\boldsymbol{A}_2\\ \boldsymbol{B}\end{bmatrix}\overline{\boldsymbol{s}}_2 + \begin{bmatrix}\boldsymbol{0}\\ \overline{\boldsymbol{m}}\end{bmatrix}$
- there exists $j \in [d]$ so that $f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) \neq 0$

We now define the event $E_i$ that $\mathcal{E}$ terminates and the output satisfies case $i$. We can then express the probability that $\mathcal{E}$ terminates as

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

We also have

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

Hence, we need to upper bound the probability of $E_3$. We do this by counting the number of tuples in $\mathfrak{R} \times R_q^N$ that can give outputs belonging to case 3, and by using Lemma 16, as follows.

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

We used that the total size of $\mathfrak{R} \times R_q^N$ is $|\mathfrak{R}| \cdot q^{dN}$. Hence the probability that $\mathcal{E}$ succeeds is $\Pr[\mathcal{E} \text{ terminates}] - \Pr[E_3] \geq \varepsilon - 2/|\mathcal{C}| - q_1^{-N/2}$. □

From the construction and the above claims, we see that $\mathcal{E}$ has the desired properties. □

## 6.3 Many quadratic equations and that polynomial evaluations have no constant coefficients

We now want to expand the protocol $\Pi^{(2)}_{\mathsf{many}}$ to also prove that for $D$ quadratic $k(m_1 + \ell)$-variate polynomials $F_1, \ldots, F_D$, the evaluations $F_j(\boldsymbol{s})$ have constant coefficients equal to zero. In order to achieve this, we use ideas from Section 5.3.

The basic idea is to let the prover draw a masking polynomial $g$ with constant coefficient equal to zero, and upon receiving challenges $\gamma_1, \ldots, \gamma_D$ from the verifier, let the verifier check that

$$h := g + \sum_{j=1}^{D} \gamma_j F_j(\boldsymbol{s})$$

has constant coefficient equal to zero. This would imply that $F_j(\boldsymbol{s}) = 0$ for all $j$, except with probability $1/q_1$. We will decrease the soundness error by doing this in parallel repetitions.

In order to obtain a soundness error of $q_1^{-\lambda}$, we instead start by drawing $\lambda$ masking polynomials with constant coefficients equal to zero, $\boldsymbol{g} = (g_1, \dots, g_\lambda) \xleftarrow{\$} \{x \in R_q \mid \tilde{x} = 0\}^\lambda$. We then make an additional commitment to $\boldsymbol{g}$ under the same randomness,

$$\boldsymbol{t}_g := \boldsymbol{B}_g \boldsymbol{s}_2 + \boldsymbol{g},$$

for a matrix $\boldsymbol{B}_g \in R_q^{\lambda \times m_2}$, and send $\boldsymbol{t}_g$ to the verifier. The verifier then sends a challenge matrix $(\gamma_{i,j})_{i \in [\lambda], j \in [D]} \xleftarrow{\$} \mathbb{Z}_q^{\lambda \times D}$, where the $\lambda$ rows corresponds to the challenges for which we can compute the polynomial

$$h_i := g_i + \sum_{j=1}^{D} \gamma_{i,j} F_j(\boldsymbol{s}).$$

The goal is to prove that all the polynomials $h_i$ have constant coefficients equal to zero, and this is simply done by sending them to the verifier, and the verifier can check if $\tilde{h}_i = 0$ for all $i \in [\lambda]$. We also need to prove that $\boldsymbol{h} = (h_1, \dots, h_\lambda)$ was constructed correctly, which can be reduced to proving quadratic relations as follows. If we let $\boldsymbol{x}_1 \in R_q^{km_1}$, $\boldsymbol{x}_2 = (\boldsymbol{x}_{2,1}, \dots, \boldsymbol{x}_{2,k}) \in R_q^{k(\ell+\lambda)}$ and

$$\begin{aligned}
&\boldsymbol{x}_{2,j} := \big(\boldsymbol{x}_{2,j}^{(m)}, \boldsymbol{x}_{2,j}^{(g)}\big) \in R_q^{\ell+\lambda} \text{ for } j \in [k], \\
&\boldsymbol{x}_2^{(m)} := \big(\boldsymbol{x}_{2,1}^{(m)}, \dots, \boldsymbol{x}_{2,k}^{(m)}\big), \quad \boldsymbol{x}_{2,1}^{(g)} := \big(x_{2,1,1}^{(g)}, \dots, x_{2,1,\lambda}^{(g)}\big),
\end{aligned}$$

then we can define polynomials $f_{d+1}, \dots, f_{d+\lambda} : R_q^{k(m_1+\ell+\lambda)} \to R_q$ as follows.

$$f_{d+i}(\boldsymbol{x}_1, \boldsymbol{x}_2) := x_{2,1,i}^{(g)} + \sum_{j=1}^{D} \gamma_{i,j} F_j\big(\boldsymbol{x}_1, \boldsymbol{x}_2^{(m)}\big) - h_i \text{ for } i \in [\lambda]. \tag{6.4}$$

We do this because if we now were to set $(\boldsymbol{x}_1, \boldsymbol{x}_2) = \Big((\sigma^i(\boldsymbol{s}_1))_{i \in [k]}, (\sigma^i(\boldsymbol{m} \| \boldsymbol{g}))_{i \in [k]}\Big)$, we would get

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

This means that $h_i = g_i + \sum_{j=1}^{D} \gamma_{i,j} F_j\big((\sigma^i(\boldsymbol{s}_1))_{i \in [k]}, (\sigma^i(\boldsymbol{m}))_{i \in [k]}\big)$ if and only if

$$f_{d+i}\big((\sigma^i(\boldsymbol{s}_1))_{i \in [k]}, (\sigma^i(\boldsymbol{m} \| \boldsymbol{g}))_{i \in [k]}\big) = 0.$$

So since proving that $\boldsymbol{h}$ was correctly constructed is now reduced to proving many quadratic equations in $\boldsymbol{s}$, we for convenience also define the polynomials $f_1, \dots, f_d : R_q^{k(m_1+\ell+\lambda)} \to R_q$ as

$$f_j(\boldsymbol{x}_1, \boldsymbol{x}_2) := f_j\big(\boldsymbol{x}_2, \boldsymbol{x}_2^{(m)}\big) \text{ for } j \in [d]. \tag{6.5}$$

This is so that we can simply run $\Pi_{\mathsf{many}}^{(2)}$ on all the polynomials $f_1, \dots, f_d, f_{d+1}, \dots, f_{d+\lambda}$ at the same time. The protocol is given as $\Pi_{\mathsf{eval}}^{(2)}$ in Figure 6.3, and we now give the full security analysis of the protocol.

**Theorem 18.** Suppose that $\mathfrak{s}_1 = \gamma_1 \alpha \eta$ and $\mathfrak{s}_2 = \gamma_2 \nu \eta \sqrt{m_2 N}$ for some $\gamma_1, \gamma_2 > 0$. Then the protocol $\Pi_{\mathsf{eval}}^{(2)}$ is complete, in the sense that if $m_1, m_2 \geq 640/N$ then the honest prover convinces the honest verifier with probability

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

$\underline{\Pi^{(2)}_{\mathsf{eval}}}$

Public Information: $\boldsymbol{A}_1 \in R_q^{n\times m_1}$, $\boldsymbol{A}_2 \in R_q^{n\times m_2}$, $\boldsymbol{B} \in R_q^{\ell\times m_2}, \boldsymbol{B}_{\mathsf{ext}} \in R_q^{m_2}, \boldsymbol{B}_g \in R_q^{\lambda\times m_2}$.
$f_1,\dots,f_d,F_1,\dots,F_D : R_q^{k(m_1+\ell)} \to R_q, \sigma \in \mathsf{Aut}(R_q)$
Prover's Information: $(\boldsymbol{s}_1,\boldsymbol{m}) \in R_q^{m_1+\ell}$ so that $\|\boldsymbol{s}_1\| \le \alpha$, $\boldsymbol{s}_2 \xleftarrow{\$} \chi^{m_2}$
Commitment: $\begin{bmatrix}\boldsymbol{t}_A\\ \boldsymbol{t}_B\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{0}\end{bmatrix}\boldsymbol{s}_1 + \begin{bmatrix}\boldsymbol{A}_2\\ \boldsymbol{B}\end{bmatrix}\boldsymbol{s}_2 + \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{m}\end{bmatrix}$

| Prover | | Verifier |
|---|---|---|
| $\boldsymbol{s} := \begin{bmatrix}(\sigma^i(\boldsymbol{s}_1))_{i\in[k]}\\ (\sigma^i(\boldsymbol{m}))_{i\in[k]}\end{bmatrix}$<br>$\boldsymbol{g} := (g_1,\dots,g_\lambda) \xleftarrow{\$} \{x \in R_q \mid \tilde{x} = 0\}^\lambda$<br>$\boldsymbol{t}_g := \boldsymbol{B}_g\boldsymbol{s}_2 + \boldsymbol{g}$ | | |
| | $\xrightarrow{\boldsymbol{t}_g}$ | |
| | | $(\gamma_{i,j}) \xleftarrow{\$} \mathbb{Z}_q^{\lambda\times D}$ |
| | $\xleftarrow{(\gamma_{i,j})_{i\in[\lambda],j\in[D]}}$ | |
| For $i \in [\lambda]$ :<br>$h_i := g_i + \sum_{j=1}^{D}\gamma_{i,j}F_j(\boldsymbol{s})$ | | |
| | $\xrightarrow{h_1,\dots,h_\lambda}$ | |
| Define $f_1,\dots,f_{d+\lambda}$ as in (6.4) and (6.5)<br>run $\Pi^{(2)}_{\mathsf{many}}((\boldsymbol{s}_2,\boldsymbol{s}_1,\boldsymbol{m}\|\boldsymbol{g}),\sigma,(f_i)_{i\in[d+\lambda]})$ | | |
| | | Accept iff:<br>1. $\Pi^{(2)}_{\mathsf{many}}$ verifies<br>2. $\tilde{h}_1 = \cdots = \tilde{h}_\lambda = 0$ |

Figure 6.3: Proof of knowledge $\Pi^{(2)}_{\mathsf{eval}}((\boldsymbol{s}_2,\boldsymbol{s}_1,\boldsymbol{m}),\sigma,(f_1,\dots,f_d),(F_1,\dots,F_D))$ of $((\boldsymbol{s}_1,\boldsymbol{m}),\boldsymbol{s}_2,\overline{c}) \in R_q^{m_1+\ell}\times R_q^{m_2}\times\overline{\mathcal{C}}$ that satisfy $\boldsymbol{A}_1\boldsymbol{s}_1+\boldsymbol{A}_2\boldsymbol{s}_2=\boldsymbol{t}_A$, $\boldsymbol{B}\boldsymbol{s}_2+\boldsymbol{m}=\boldsymbol{t}_B$, $\|\overline{c}\boldsymbol{s}_i\| \le 2\mathfrak{s}_i\sqrt{2m_iN}$ for $i=1,2$, $f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]},(\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big)=0$ for $j\in[d]$ and all the evaluations $F_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]},(\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big)$ for $j\in[D]$ have constant coefficient equal to zero .

*Proof.* This follows directly from Theorem 14 and that for an honest prover, the remaining verification equations holds by construction. □

**Theorem 19.** Let $\mathfrak{s}_1 = \gamma_1 \alpha \eta$ and $\mathfrak{s}_2 = \gamma_2 \nu \eta \sqrt{m_2 N}$ for some $\gamma_1, \gamma_2 > 0$. Then the protocol $\Pi^{(2)}_{\mathsf{eval}}$ is commit-and-prove simulatable, meaning that there exists a simulator $\mathcal{S}$ that, without access to private information $(\boldsymbol{s}_1, \boldsymbol{m})$, outputs a simulation of a commitment $(\boldsymbol{t}_A, \boldsymbol{t}_B)$ along with a non-aborting transcript of the protocol between the prover and verifier such that for every algorithm $\mathcal{A}$ that has advantage $\varepsilon$ in distinguishing the simulated commitment and transcript from the real commitment and transcript, whenever the prover does not abort, there is an algorithm $\mathcal{A}'$ with the same running time that has advantage $\varepsilon/2 - 2^{-128}$ in solving the $\mathsf{Extended\text{-}MLWE}_{n+\ell+\lambda+1, m_2-n-\ell-\lambda-1, \chi, \mathcal{C}, \mathfrak{s}_2}$ problem.

*Proof.* We simulate the commitment and the transcript identically as in the proof of Theorem 15 with two additional steps.

1. We simulate the commitment $\boldsymbol{t}_g$ to $\boldsymbol{g}$ by setting $\boldsymbol{t}_g \xleftarrow{\$} R_q^\lambda$ to be a uniformly random vector.
2. We simulate the polynomials $h_1, \ldots, h_\lambda$ by choosing then uniformly random from $X := \{x \in R_q \mid \tilde{x} = 0\}$.

Each $h_i$ is simulated perfectly since the $g_i$'s are also sampled uniformly from $X$ in the real execution and $\sum_{j=1}^{D} \gamma_{i,j} F_j(\boldsymbol{s}) \in X$. The total additional commitment is now of dimension $\lambda + 1$, and hence by the same argument as in the proof of Theorem 15, if there is an algorithm $\mathcal{A}$ with advantage $\varepsilon$ in distinguishing this transcript from a real one, there is an algorithm $\mathcal{A}'$ with advantage $\varepsilon/2 - 2^{-128}$ in solving the $\mathsf{Extended\text{-}MLWE}_{n+\ell+\lambda+1, m_2-n-\ell-\lambda-1, \chi, \mathcal{C}, \mathfrak{s}_2}$ problem. □

**Theorem 20.** The protocol $\Pi^{(2)}_{\mathsf{eval}}$ is knowledge sound, meaning that there is an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a probabilistic prover $\mathcal{P}^*$, which convinces $\mathcal{V}$ with probability $\varepsilon \geq 2/|\mathcal{C}| + q_1^{-N/2} + q_1^{-\lambda}$, the extractor $\mathcal{E}$ with probability at least $\varepsilon - 2/|\mathcal{C}| - q_1^{-N/2} - q_1^{-\lambda}$ either outputs $(\overline{\boldsymbol{s}}_2, \overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}) \in R_q^{m_1+m_2+\ell}$ and $\overline{c} \in R_q^\times$ such that

- $\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \overline{\boldsymbol{s}}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \overline{\boldsymbol{s}}_2 + \begin{bmatrix} \boldsymbol{0} \\ \overline{\boldsymbol{m}} \end{bmatrix}$
- $f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$ for $j \in [d]$
- each $F_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) \in R_q$ where $j \in [D]$, has constant coefficient equal to zero
- $\|\overline{c}\|_\infty \leq 2\omega$
- $\|\overline{c}\overline{\boldsymbol{s}}_1\| \leq 2\mathfrak{s}_1\sqrt{2m_1N}$ and $\|\overline{c}\overline{\boldsymbol{s}}_2\| \leq 2\mathfrak{s}_2\sqrt{2m_2N}$

or an $\mathsf{MSIS}_{n,m_1+m_2,B}$ solution for $\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{bmatrix}$ in expected time at most $12T$ where running $\mathcal{P}^*$ once is assumed to take at most $T$ time and $B = 8\eta\sqrt{(\mathfrak{s}_1\sqrt{2m_1N})^2 + (\mathfrak{s}_2\sqrt{2m_2N})^2}$.

*Proof.* We let $\mathcal{P}^*$ be such a probabilistic prover as described in the Theorem, that convinces the verifier with probability $\varepsilon \geq 2/|\mathcal{C}| + q_1^{-N/2} + q_1^{-\lambda}$ and runs in time at most $T$. In order to construct the desired extractor $\mathcal{E}$ we first define a deterministic algorithm $\mathcal{A}$ that uses the $\Pi^{(2)}_{\mathsf{many}}$-extractor $\mathcal{E}^*$ from Theorem 17. It is defined as follows. Given randomness $\rho = (\rho_P, \rho_E) \in \mathfrak{R}_P \times \mathfrak{R}_E$ and challenge matrix $\Gamma \in \mathbb{Z}_q^{\lambda \times D}$, $\mathcal{A}(\rho_P, \rho_E, \Gamma)$ runs $\mathcal{P}^*(\rho_P)$ on randomness $\rho_P$ with challenge $\Gamma$ and stops after

the third round. We let $\boldsymbol{t}_g$ and $\boldsymbol{h}$ be the output of $\mathcal{P}^*(\rho_P)$ in the first and third round, respectively. Then it runs the extractor $\mathcal{E}^*(\rho_E)$ from Theorem 17 with randomness $\rho_E$. Hence $\mathcal{P}^*(\rho_P, \Gamma)$ is run in a black-box manner, and according to Theorem 17 we know that $\mathcal{E}^*(\rho_E)$ will with a certain probability either extract a valid $\mathsf{MSIS}$ solution or output $(\overline{\boldsymbol{s}}_2, \overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{c})$. We say that $\mathcal{A}$ succeeds if it outputs $(\boldsymbol{t}_g, \Gamma, \boldsymbol{h}, \overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{\boldsymbol{g}}, \overline{\boldsymbol{s}}_2, \overline{c})$ such that

- $\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \\ \boldsymbol{t}_g \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \cdot \overline{\boldsymbol{s}}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \\ \boldsymbol{B}_g \end{bmatrix} \cdot \overline{\boldsymbol{s}}_2 + \begin{bmatrix} \boldsymbol{0} \\ \overline{\boldsymbol{m}} \\ \overline{\boldsymbol{g}} \end{bmatrix}$
- $\tilde{h}_1 = \cdots = \tilde{h}_\lambda = 0$
- $f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$ for $j \in [d]$
- for all $i \in [\lambda]$, $h_i = \overline{g}_i + \sum_{j=1}^{D} \gamma_{i,j} F_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big)$
- $\|\overline{c}\|_\infty \leq 2\omega$
- $\|\overline{c}\overline{\boldsymbol{s}}_1\| \leq 2\mathfrak{s}_1\sqrt{2m_1N}$ and $\|\overline{c}\overline{\boldsymbol{s}}_2\| \leq 2\mathfrak{s}_2\sqrt{2m_2N}$

If we assume that $\mathcal{E}^*$ does not extract a valid $\mathsf{MSIS}$ solution, then Theorem 17 gives that the probability that $\mathcal{A}$ succeeds for a random $\rho$ and $\Gamma$ is at least $\varepsilon - 2/|\mathcal{C}| - q_1^{-N/2}$, and that the expected run time of $\mathcal{A}$ for any fixed $\rho_P, \Gamma$ and $\rho_E \xleftarrow{\$} \mathfrak{R}_E$ is at most $6T$.

In order to define the extractor $\mathcal{E}$ we need some definitions and results. We define $H \subseteq \mathfrak{R}_P \times \mathfrak{R}_E \times \mathbb{Z}_q^{\lambda\times D}$ to be the set of triples $(\rho, \Gamma) = (\rho_P, \rho_E, \Gamma)$ such that $\mathcal{A}(\rho, \Gamma)$ succeeds. Then we can define $H(\rho_P)$ to be the set of all $(\rho_E, \Gamma)$ for which $(\rho_P, \rho_E, \Gamma) \in H$. When $\mathcal{A}(\rho, \Gamma)$ succeeds for a fixed $(\rho, \Gamma) \in H$, we denote the output as $(\overline{\boldsymbol{s}}_2^{(\rho,\Gamma)}, \overline{\boldsymbol{s}}_1^{(\rho,\Gamma)}, \overline{\boldsymbol{m}}^{(\rho,\Gamma)})$. We can then define, in the same manner as previous,

$$\overline{\boldsymbol{s}}^{(\rho,\Gamma)} := \begin{bmatrix} (\sigma^i(\overline{\boldsymbol{s}}_1^{(\rho,\Gamma)}))_{i\in[k]} \\ (\sigma^i(\overline{\boldsymbol{m}}^{(\rho,\Gamma)}))_{i\in[k]} \end{bmatrix} \in R_q^{k(m_1+\ell)}.$$

Finally, we can define the set of $(\rho, \Gamma) \in H$ for which at least one of the $F_j(\overline{\boldsymbol{s}}^{(\rho,\Gamma)})$ has non-zero constant coefficient,

$$H' := \big\{(\rho, \Gamma) \in H \mid \exists j \in [D] : x_j = F_j(\overline{\boldsymbol{s}}^{(\rho,\Gamma)}), \tilde{x}_j \neq 0\big\}.$$

Before we define $\mathcal{E}$, we present a Lemma on the probability that all the $F_j(\overline{\boldsymbol{s}}^{(\rho,\Gamma)})$'s have zero constant coefficients when $(\rho, \Gamma) \in H'$.

**Lemma 17.** If $(\rho_P, \rho_E, \Gamma) \in H$ then $\Pr_{(\rho'_E,\Gamma')\xleftarrow{\$}\mathfrak{R}_E\times\mathbb{Z}_q^{\lambda\times D}}[(\rho_P, \rho'_E, \Gamma') \in H] > 0$. Moreover, if $(\rho_p, \rho_E, \Gamma) \in H'$ then

$$\Pr_{\Gamma'\xleftarrow{\$}\mathbb{Z}_q^{\lambda\times D}}\left[\forall i \in [\lambda], \tilde{x}_i = 0 \;\middle|\; x_i := \overline{g}_i^{(\rho,\Gamma)} + \sum_{j=1}^{D} \gamma'_{i,j} F_j\big(\overline{\boldsymbol{s}}^{(\rho,\Gamma)}\big)\right] \leq q_1^{-\lambda}.$$

*Proof.* For the first part, we observe that if $(\rho_P, \rho_E, \Gamma) \in H$, then we have

$$\Pr_{(\rho'_E,\Gamma')\xleftarrow{\$}\mathfrak{R}_E\times\mathbb{Z}_q^{\lambda\times D}}[(\rho_P, \rho'_E, \Gamma') \in H] \geq \Pr_{(\rho'_E,\Gamma')\xleftarrow{\$}\mathfrak{R}_E\times\mathbb{Z}_q^{\lambda\times D}}[(\rho'_E, \Gamma') = (\rho_E, \Gamma)] > 0.$$

For the second part, if we assume that for some $j$, $\tilde{x}_j \neq 0$ when $x_j = F_j(\overline{\boldsymbol{s}}^{(\rho,\Gamma)})$, then the probability that $\tilde{x}_i = 0$ for $x_i := \overline{g}_i^{(\rho,\Gamma)} + \sum_{j=1}^{D} \gamma'_{i,j} F_j\big(\overline{\boldsymbol{s}}^{(\rho,\Gamma)}\big)$, i.e. the probability over $\gamma'_{i,j} \xleftarrow{\$} \mathbb{Z}_q$ that $\gamma'_{i,j} F_j\big(\overline{\boldsymbol{s}}^{(\rho,\Gamma)}\big) = 0$, is at most $q_1^{-1}$. Hence the probability that $\tilde{x}_i = 0$ for all $i \in [\lambda]$ is at most $q_1^{-\lambda}$. □

We are now ready to define the extractor $\mathcal{E}$, and it is constructed in the following manner:

1. Sample $\rho = (\rho_R, \rho_E) \xleftarrow{\$} \mathfrak{R}_P \times \mathfrak{R}_E$ and $\Gamma \in \mathbb{Z}^{\lambda \times D}$, run $\mathcal{A}(\rho, \Gamma)$, and abort if $\mathcal{A}(\rho, \Gamma)$ does not succeed.
2. If $\mathcal{A}(\rho, \Gamma)$ does succeed, run $\mathcal{A}(\rho_P, \rho'_E, \Gamma')$ with fresh $\rho'_E \xleftarrow{\$} \mathfrak{R}_E$ and $\Gamma' \xleftarrow{\$} \mathbb{Z}_q^{\lambda \times D}$ until $\mathcal{A}$ succeeds.

When $\mathcal{E}$ does not abort, this procedure will yield two extracted tuples $x = (\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{\boldsymbol{s}}_2, \overline{c})$ and $x' = (\overline{\boldsymbol{s}}'_1, \overline{\boldsymbol{m}}', \overline{\boldsymbol{s}}'_2, \overline{c}')$ since it runs until $\mathcal{A}$ succeeds twice. We say that $\mathcal{E}$ succeeds if it extracts two such tuples such that one of the below conditions holds.

- $(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{s}}_2) \neq (\overline{\boldsymbol{s}}'_1, \overline{\boldsymbol{s}}'_2)$, $\max(\|\overline{c}\|_\infty, \|\overline{c}'\|_\infty) \leq 2\omega$ and $\max(\|\overline{c}\overline{\boldsymbol{s}}_i\|, \|\overline{c}'\overline{\boldsymbol{s}}'_i\|) \leq 2\mathfrak{s}_i\sqrt{2m_iN}$ for $i = 1, 2$, and
$$\begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \overline{\boldsymbol{s}}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \overline{\boldsymbol{s}}_2 + \begin{bmatrix} \boldsymbol{0} \\ \overline{\boldsymbol{m}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \overline{\boldsymbol{s}}'_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \overline{\boldsymbol{s}}'_2 + \begin{bmatrix} \boldsymbol{0} \\ \overline{\boldsymbol{m}}' \end{bmatrix}.$$
- For all $j \in [d]$, $f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$ and for all $j \in [D]$, the constant coefficient of $F_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big)$ equals zero, and $\|\overline{c}\|_\infty \leq 2\omega$ and $\|\overline{c}\overline{\boldsymbol{s}}_1\| \leq 2\mathfrak{s}_1\sqrt{2m_1N}$ and $\|\overline{c}\overline{\boldsymbol{s}}_2\| \leq 2\mathfrak{s}_2\sqrt{2m_2N}$ and
$$\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \end{bmatrix} \overline{\boldsymbol{s}}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \end{bmatrix} \overline{\boldsymbol{s}}_2 + \begin{bmatrix} \boldsymbol{0} \\ \overline{\boldsymbol{m}} \end{bmatrix}.$$

If the first case holds true, we break the binding property of the commitment scheme, which gives us the relevant MSIS solution. If the second case holds true, we have extracted the desired $(\overline{\boldsymbol{s}}_2, \overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{c})$. We will now present and prove two claims about $\mathcal{E}$, which completes the proof.

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

The proof of this claim is identical to the proof of the equivalent claim in Theorem 17. We can therefore conclude that the expected run time of $\mathcal{E}$ is at most $12T$, twice the expected run time of $\mathcal{A}$.

*Claim.* The probability that $\mathcal{E}$ succeeds is at least $\varepsilon - 2/|\mathcal{C}| - q_1^{-N/2} - q_1^{-\lambda}$.

*Proof.* As we discussed, $\mathcal{E}$ terminates with probability at least $\varepsilon - 2/|\mathcal{C}| - q_1^{-N/2}$. So, we assume $\mathcal{E}$ terminates and let $(\boldsymbol{t}_g, \Gamma, \boldsymbol{h}, \overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{\boldsymbol{g}}, \overline{\boldsymbol{s}}_2, \overline{c})$ and $(\boldsymbol{t}_g, \Gamma', \boldsymbol{h}', \overline{\boldsymbol{s}}'_1, \overline{\boldsymbol{m}}', \overline{\boldsymbol{g}}', \overline{\boldsymbol{s}}'_2, \overline{c}')$ be the respective outputs of $\mathcal{A}$ in the first and second step of $\mathcal{E}$. We can divide the possible such outputs into three disjoint cases. The first case stems from the first of the success conditions for $\mathcal{E}$, as described above. The two final cases stems from the second success condition for $\mathcal{E}$, one of them such that all the $F_j(\overline{\boldsymbol{s}})$'s have constant coefficient zero, and the other one such that at least one of the $F_j(\overline{\boldsymbol{s}})$ have non-zero constant coefficient.

*Case 1*:

- $(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{\boldsymbol{g}}, \overline{\boldsymbol{s}}_2) \neq (\overline{\boldsymbol{s}}'_1, \overline{\boldsymbol{m}}', \overline{\boldsymbol{g}}', \overline{\boldsymbol{s}}'_2)$
- for $i \in [\lambda]$, $\tilde{h}_i = \tilde{h}'_i = 0$
- for $i \in [\lambda]$, $h_i = \overline{g}_i + \sum_{j=1}^{D} \gamma_{i,j} F_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big)$
- for $i \in [\lambda]$, $h'_i = \overline{g}'_i + \sum_{j=1}^{D} \gamma'_{i,j} F_j\big((\sigma^i(\overline{\boldsymbol{s}}'_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}'))_{i\in[k]}\big)$
- for $j \in [d]$, $f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$ and $f_j\big((\sigma^i(\overline{\boldsymbol{s}}'_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}'))_{i\in[k]}\big) = 0$
- $\max(\|\overline{c}\|_\infty, \|\overline{c}'\|_\infty) \leq 2\omega$ and $\max(\|\overline{c}\overline{\boldsymbol{s}}_i\|, \|\overline{c}'\overline{\boldsymbol{s}}'_i\|) \leq 2\mathfrak{s}_i\sqrt{2m_iN}$ for $i = 1, 2$
- $\begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \cdot \overline{\boldsymbol{s}}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \\ \boldsymbol{B}_g \end{bmatrix} \cdot \overline{\boldsymbol{s}}_2 + \begin{bmatrix} \boldsymbol{0} \\ \overline{\boldsymbol{m}} \\ \overline{\boldsymbol{g}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \\ \boldsymbol{t}_g \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \cdot \overline{\boldsymbol{s}}'_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \\ \boldsymbol{B}_g \end{bmatrix} \cdot \overline{\boldsymbol{s}}'_2 + \begin{bmatrix} \boldsymbol{0} \\ \overline{\boldsymbol{m}}' \\ \overline{\boldsymbol{g}}' \end{bmatrix}$

*Case 2*:

- $(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{\boldsymbol{g}}, \overline{\boldsymbol{s}}_2) = (\overline{\boldsymbol{s}}'_1, \overline{\boldsymbol{m}}', \overline{\boldsymbol{g}}', \overline{\boldsymbol{s}}'_2)$
- for $i \in [\lambda]$, $\tilde{h}_i = \tilde{h}'_i = 0$
- for $i \in [\lambda]$, $h_i = \overline{g}_i + \sum_{j=1}^{D} \gamma_{i,j} F_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big)$
- for $i \in [\lambda]$, $h'_i = \overline{g}_i + \sum_{j=1}^{D} \gamma'_{i,j} F_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big)$
- for $j \in [d]$, $f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$
- $\|\overline{c}\|_\infty \leq 2\omega$ and $\|\overline{c}\overline{\boldsymbol{s}}_1\| \leq 2\mathfrak{s}_1\sqrt{2m_1N}$ and $\|\overline{c}\overline{\boldsymbol{s}}_2\| \leq 2\mathfrak{s}_2\sqrt{2m_2N}$
- $\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \\ \boldsymbol{t}_g \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \cdot \overline{\boldsymbol{s}}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \\ \boldsymbol{B}_g \end{bmatrix} \cdot \overline{\boldsymbol{s}}_2 + \begin{bmatrix} \boldsymbol{0} \\ \overline{\boldsymbol{m}} \\ \overline{\boldsymbol{g}} \end{bmatrix}$
- for $j \in [D]$, the constant coefficient of $F_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big)$ is zero.

*Case 3:*

- $(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}}, \overline{\boldsymbol{g}}, \overline{\boldsymbol{s}}_2) = (\overline{\boldsymbol{s}}'_1, \overline{\boldsymbol{m}}', \overline{\boldsymbol{g}}', \overline{\boldsymbol{s}}'_2)$
- for $i \in [\lambda]$, $\tilde{h}_i = \tilde{h}'_i = 0$
- for $i \in [\lambda]$, $h_i = \overline{g}_i + \sum_{j=1}^{D} \gamma_{i,j} F_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big)$
- for $i \in [\lambda]$, $h'_i = \overline{g}_i + \sum_{j=1}^{D} \gamma'_{i,j} F_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big)$
- for $j \in [d]$, $f_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) = 0$

– $\|\overline{c}\|_\infty \leq 2\omega$ and $\|\overline{c}\overline{\boldsymbol{s}}_1\| \leq 2s_1\sqrt{2m_1N}$ and $\|\overline{c}\overline{\boldsymbol{s}}_2\| \leq 2s_2\sqrt{2m_2N}$

– $\begin{bmatrix} \boldsymbol{t}_A \\ \boldsymbol{t}_B \\ \boldsymbol{t}_g \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \cdot \overline{\boldsymbol{s}}_1 + \begin{bmatrix} \boldsymbol{A}_2 \\ \boldsymbol{B} \\ \boldsymbol{B}_g \end{bmatrix} \cdot \overline{\boldsymbol{s}}_2 + \begin{bmatrix} \boldsymbol{0} \\ \overline{\boldsymbol{m}} \\ \overline{\boldsymbol{g}} \end{bmatrix}$

– there exists $j \in [D]$, so that the constant coefficient of $F_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big)$ is non-zero.

We now define the event $E_i$ that $\mathcal{E}$ terminates and the output satisfies case $i$. We can then express the probability that $\mathcal{E}$ terminates as

$$\Pr[\mathcal{E} \text{ terminates}] = \Pr[E_1 \vee E_2 \vee E_3] \geq \varepsilon - 2/|\mathcal{C}| - q_1^{-N/2}.$$

We also have

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

Hence, we need to upper bound the probability of $E_3$. By using Lemma 17, and the same idea as for the corresponding claim in the proof of Theorem 17, we get the bound

$$\Pr[E_3] \leq \Pr\left[\begin{array}{l} (\mathcal{A}(\rho,\Gamma) \text{ succeeds}) \wedge (\exists j \in [D] : \text{ the const. coeff. of } F_j\big((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) \text{ is non-zero}) \\ \wedge (\forall i \in [\lambda] : \text{ const. coeff. of } \overline{g}_i^{(\rho,\Gamma)} + \sum_{j=1}^{D} \gamma'_{i,j} F_j((\sigma^i(\overline{\boldsymbol{s}}_1))_{i\in[k]}, (\sigma^i(\overline{\boldsymbol{m}}))_{i\in[k]}\big) \text{ is zero}) \end{array}\right]$$

$$\leq \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho,\Gamma)\in H'} \Pr_{(\rho'_E,\Gamma') \xleftarrow{\$} H(\rho_P)}\Big[\forall i \in [\lambda] : \text{ const. coeff. of } \overline{g}_i^{(\rho,\Gamma)} + \sum_{j=1}^{D} \gamma'_{i,j} F_j\big(\overline{\boldsymbol{s}}^{(\rho,\Gamma)}\big) \text{ is zero}\Big]$$

$$\leq \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho,\Gamma)\in H'} \frac{\Pr_{\Gamma' \xleftarrow{\$} \mathbb{Z}_q^{\lambda\times M}}\Big[\forall i \in [\lambda] : \text{ const. coeff. of } \overline{g}_i^{(\rho,\Gamma)} + \sum_{j=1}^{D} \gamma'_{i,j} F_j\big(\overline{\boldsymbol{s}}^{(\rho,\Gamma)}\big) \text{ is zero}\Big]}{\Pr_{(\rho'_E,\Gamma') \xleftarrow{\$} \mathfrak{R}_E \times \mathbb{Z}_q^{\lambda\times M}}[(\rho'_E,\Gamma') \in H(\rho_P)]}$$

$$\leq \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho,\Gamma)\in H'} \frac{q_1^{-\lambda} \cdot q^{\lambda M} \cdot |\mathfrak{R}_E|}{|H(\rho_P)|}$$

$$\leq \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{(\rho,\Gamma)\in H} \frac{q_1^{-\lambda} \cdot q^{\lambda M} \cdot |\mathfrak{R}_E|}{|H(\rho_P)|}$$

$$\leq \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{\rho_P\in\mathfrak{R}_P} \sum_{(\rho_E,\Gamma)\in H(\rho_P)} \frac{q_1^{-\lambda} \cdot q^{\lambda M} \cdot |\mathfrak{R}_E|}{|H(\rho_P)|}$$

$$\leq \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \sum_{\rho_P\in\mathfrak{R}_P} |H(\rho_P)| \cdot \frac{q_1^{-\lambda} \cdot q^{\lambda M} \cdot |\mathfrak{R}_E|}{|H(\rho_P)|}$$

$$\leq \frac{1}{|\mathfrak{R}_P| \cdot |\mathfrak{R}_E| \cdot q^{\lambda M}} \cdot (|\mathfrak{R}_P| \cdot q_1^{-\lambda} \cdot q^{\lambda M} \cdot |\mathfrak{R}_E|)$$

$$\leq q_1^{-\lambda}.$$

Hence the probability that $\mathcal{E}$ succeeds is $\Pr[\mathcal{E} \text{ terminates}] - \Pr[E_3] \geq \varepsilon - 2/|\mathcal{C}| - q_1^{-N/2} - q_1^{-\lambda}$. □

From the construction and the above claims, we see that $\mathcal{E}$ has the desired properties. □

# Chapter 7

# General protocol

Lyubashevsky et al. explains in [16] how the protocol $\Pi^{(2)}_{\mathsf{eval}}$ can be used to construct a more general protocol that can be used to prove a variety of lattice statements, amongst others how to prove norm bounds on lattice elements. This final protocol is constructed such that applications to cryptographic primitives result in a single instantiation of the protocol. In this chapter we use $\sigma := \sigma_{-1} \in \mathsf{Aut}(R_q)$, which is of order 1, such that $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{m}, \sigma(\boldsymbol{s}_1), \sigma(\boldsymbol{m}))$. The reason for this choice of $\sigma$ is that we then can, according to Lemma 8, use the function $\mathsf{T}$ to prove inner products modulo $q$.

In addition to proving quadratic relations on $\boldsymbol{s}$ and proving that quadratic polynomials evaluated in $\boldsymbol{s}$ have zero constant coefficients, this general protocol will also allow one to prove approximate $\ell_\infty$ norm bounds and exact $\ell_2$ norm bounds on linear relations in $\boldsymbol{s}$, and that linear combinations of $\boldsymbol{s}$ have binary coefficients. Proving exact $\ell_2$ norm bounds is important in many lattice based protocols. Approximate $\ell_\infty$ norm proofs can be used to assure that there is no overflow modulo $q$, and for instance lift equations from $\mathbb{Z}_q$ to $\mathbb{Z}$.

Concretely, this general protocol is a proof of knowledge of $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{m}, \sigma(\boldsymbol{s}_1), \sigma(\boldsymbol{m})) \in R_q^{2m_1} \times R_q^{2\ell}$, when the prover knows a $\mathsf{ABDLOP}$ commitment to $(\boldsymbol{s}_1, \boldsymbol{m})$, that satisfies

$$\forall i \in [\rho],\ f_i(\boldsymbol{s}) = 0, \tag{7.1}$$

$$\forall i \in [\rho_{\mathsf{eval}}],\ \tilde{F}_i(\boldsymbol{s}) = 0, \tag{7.2}$$

$$\forall i \in [v_d],\ \|\boldsymbol{D}_i\boldsymbol{s} - \boldsymbol{u}_i\|_\infty \leq \beta_i^{(d)}, \tag{7.3}$$

$$\forall i \in [v_e],\ \|\boldsymbol{E}_i\boldsymbol{s} - \boldsymbol{v}_i\| \leq \beta_i^{(e)}, \tag{7.4}$$

$$\boldsymbol{E}_{\mathsf{bin}}\boldsymbol{s} - \boldsymbol{v}_{\mathsf{bin}} \in \{0,1\}^{Nk_{\mathsf{bin}}}, \tag{7.5}$$

for the following public parameters.

- Quadratic functions for $i \in [\rho]$, $f_i : R_q^{2(m_1+\ell)} \to R_q$.
- Evaluation functions for $i \in [\rho_{\mathsf{eval}}]$, $F_i : R_q^{2(m_1+\ell)} \to R_q$.
- For $i \in [v_d]$, $\boldsymbol{D}_i \in R_q^{k_i \times 2(m_1+\ell)}$, $\boldsymbol{u}_i \in R_q^{k_i}$.
- For $i \in [v_e]$, $\boldsymbol{E}_i \in R_q^{p_i \times 2(m_1+\ell)}$, $\boldsymbol{v}_i \in R_q^{p_i}$.

– Bounds $(\beta_i^{(d)})_{i\in[v_d]}$, $(\beta_i^{(e)})_{i\in[v_e]}$.

– $\boldsymbol{E}_{\mathsf{bin}} \in R_q^{k_{\mathsf{bin}}\times 2(m_1+\ell)}$, $\boldsymbol{v}_{\mathsf{bin}} \in R_q^{k_{\mathsf{bin}}}$.

Here $\beta_i^{(d)}$ is an approximate bound on $\|\boldsymbol{D}_i\boldsymbol{s} - \boldsymbol{u}_i\|_\infty$, $\beta_i^{(e)}$ is a tight bound on $\|\boldsymbol{E}_i\boldsymbol{s} - \boldsymbol{v}_i\|$, and the matrix $\boldsymbol{E}_{\mathsf{bin}}$ and vector $\boldsymbol{v}_{\mathsf{bin}}$ are such that $\boldsymbol{E}_{\mathsf{bin}}\boldsymbol{s} - \boldsymbol{v}_{\mathsf{bin}}$ is binary.

In order to prove (7.1) and (7.2) we can use $\Pi_{\mathsf{eval}}^{(2)}$ in a straight forward manner. The remaining equations (7.3) to (7.5) can also be proven using this protocol, but in a less obvious manner. In the following subsections we will explain how, before we put it all together in the final protocol that is presented in Section 7.3. Section 7.1 explains how we can construct approximate norm proofs, in both $\ell_2$ and $\ell_\infty$ norms, before we explain how to prove that a vector is binary and how to prove exact norm bounds in Section 7.2.

## 7.1 Proving approximate norms

Suppose that for $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{m})$ such that $\|\boldsymbol{s}\| \leq B$, we want to prove that $\|\boldsymbol{s}\|_p \leq \psi B$, for $p \in \{2, \infty\}$ and some public constant $\psi > 1$. In this section we will provide a method for proving such a looser bound. The idea behind this proof is to use Lemmas 5 and 6, which states that the norm of $R\vec{s}+\vec{y}$ is approximately the same as the norm of $\boldsymbol{s}$, in the $\ell_\infty$ and $\ell_2$ norms respectively, when $R$ is drawn from a binomial distribution. Then we can shrink a possibly very long vector $\boldsymbol{s}$ into the much shorter one $R\vec{s}+\vec{y}$ with approximately the same norm, where $R \xleftarrow{\$} \mathsf{Bin}_1^{k\times(m_1+\ell)N}$ and $\vec{y} \in \mathbb{Z}_q^k$ for a $k$ such that $N|k$.

So the verifier has to check the norm of $R\vec{s}+\vec{y}$. In order for the prover to be able to send this vector to the verifier without revealing anything about $\vec{s}$, $\vec{y}$ is drawn as a masking vector, and rejection sampling is performed on $\vec{z} = R\vec{s}+\vec{y}$. The protocol is now constructed by letting the prover sample a masking vector $\boldsymbol{y} \xleftarrow{\$} D_{R^{k/N},\mathfrak{s}_3}$ and committing to it. The verifier sends a challenge matrix $R \xleftarrow{\$} \mathsf{Bin}_1^{k\times(m_1+\ell)N}$, for which the prover computes $\vec{z} = R\vec{s}+\vec{y}$. It then uses rejection sampling on $\vec{z}$, and the verifier accepts if the norm of $\vec{z}$ is small enough according to Lemma 5 or 6, depending on which norm we use. What remains now, is to prove the well-formedness of $\vec{z}$.

We note that this $\vec{z}$ is well suited for bimodal rejection sampling, since when $b$ is a sign, $bR$ is still from the binomial distribution. This reduces the standard deviation $\mathfrak{s}_3$, which also reduces the norm bound we are able to prove. In order to do this, we sample a sign $b \xleftarrow{\$} \{-1, 1\}$, compute the masking $\vec{z} = bR\vec{s}+\vec{y}$ and run the rejection sampling algorithm $\mathsf{Rej}_0(\vec{z}, bR\vec{s}, \mathfrak{s}_3)$ instead. In addition to this, we must now also commit to $b$ and prove that $b \in \{-1, 1\}$. The commitment to $b$ is added in the $\mathsf{BDLOP}$ part of the commitment to $\boldsymbol{s}$, and the zero-knowledge proof that $b$ is a sign can be added as a part of the well-formedness proof of $\vec{z}$. Hence this is not expensive.

We now explain how to prove that $b$ is a sign. This is done in two steps. We first prove that $b$ is an integer, and then we prove that $(b-1)(b+1) = 0$. This is sufficient, since $\mathbb{Z}_q$ is a field, and thus $(b-1)(b+1) = 0$ implies that $b \in \{-1, 1\}$. In order to prove that $b$ is an integer, we define $\delta_i := X^i \in R_q$, and prove that the inner products $\langle \delta_i, b\rangle = 0$ for all $i = 1, \ldots, N-1$. Since $\langle \delta_i, b\rangle$ maps $b$ to its $i$-th coefficient, this proves that $b$ is zero in every coefficient except the constant coefficient. Finally, we can prove that $(b-1)(b+1) = 0$ by using the protocol $\Pi_{\mathsf{eval}}^{(2)}$, where we input

$f(b) := (b-1)(b+1)$ as a quadratic function.

We now explain how we can instantiate $\Pi^{(2)}_{\mathsf{eval}}$ to prove well-formedness of $\vec{z}$, that $b$ is a sign and that $f(b) = 0$, in a single instance of the protocol. For each of the $k$ rows of $\vec{z}$, we define the function $F_i$,

$$F_i(\boldsymbol{s}, \boldsymbol{y}, b) = z_i - \mathsf{T}(b\vec{r_i}, \vec{s}) - y_i, \forall i \in [k],$$

where $\vec{r}_i \in \mathbb{Z}_q^{N(m_1+\ell)}$ is the $i$-th row of $R$, and $\mathsf{T}$ is the function defined in Section 3.5. We can now use Lemma 8 to see that if we are able to prove that $\tilde{F}_i = 0$, then we have proved $z_i = b\langle \vec{r}_i, \vec{s}\rangle + y_i$. So if we prove that $\tilde{F}_i = 0$ for all $1 \leq i \leq k$, then we must have that $\vec{z} = bR\vec{s} + \vec{y}$, and thus that $\vec{z}$ was formed correctly. Using the same idea, we can define the functions $G_j$,

$$G_j(b) := \mathsf{T}(\delta_j, b), \forall j \in [N].$$

If we now are able to prove that $\tilde{G}_j = 0$, this implies that $\langle \delta_j, b\rangle = 0$. We therefore define $\Psi := \{F_1, \ldots, F_k, G_1, \ldots, G_{N-1}\}$, which will be the evaluation functions. Thus we can run $\Pi^{(2)}_{\mathsf{many}}((\boldsymbol{s}_2, \boldsymbol{s}_1, (\boldsymbol{m}, b)), \sigma, f, \Psi)$ to achieve the desired proof.

We now set $k = 256$. As in Lemma 4 we let $\mathfrak{s}_3 = \gamma_3 T$ for a $\gamma_3 > 0$, where $T$ is an upper bound on $\|bR\vec{s}\|$. By Lemma 7 we get that $\|Rs\| \leq \sqrt{337}B$, except with probability $2^{-128}$, since we have $\|\boldsymbol{s}\| \leq B$. We thus set $\mathfrak{s}_3 := \gamma_3\sqrt{337}B$.

If we want to do the proof in the $\ell_\infty$ norm, we can use Lemma 3 to see that for each $1 \leq i \leq 256$ we have $|\vec{z}_i| < \delta\mathfrak{s}_3$ with probability at least $1 - 2e^{-\delta^2/2}$. Thus if we set $\kappa = \delta^2/2$, the probability that $\|\vec{z}\|_\infty \leq \sqrt{2\kappa}\mathfrak{s}_3$ is at least $1 - 256 \cdot 2e^{-\kappa}$. We now use Lemma 5 and see that if $\|bR\vec{s} + \vec{y}\|_\infty \leq \sqrt{2\kappa}\mathfrak{s}_3$, then we have $\|\boldsymbol{s}\|_\infty \leq 2\sqrt{2\kappa}\mathfrak{s}_3$, except with negligible probability $2^{-256}$. So if we let the verifier check that $\|\vec{z}\|_\infty \leq \sqrt{2\kappa}\mathfrak{s}_3$, this protocol convinces the verifier that $\|\boldsymbol{s}\|_\infty \leq 2\gamma_3\sqrt{2\kappa \cdot 337}B$. We present the protocol for the $\ell_\infty$ norm below, with $\kappa = 128$.

If we instead want to do the proof in the $\ell_2$ norm, we also have by Lemma 3 that the probability that $\|\vec{z}\| \leq \delta\mathfrak{s}_3\sqrt{256}$ is at least $1 - \delta^{256}\exp(128(1-\delta^2))$ when $\boldsymbol{z} \xleftarrow{\$} D_{R^{256/N}, \mathfrak{s}_3}$. We now use Lemma 6 and see that if $\|bR\vec{s} + \vec{y} \mod q\| \leq \delta\mathfrak{s}_3\sqrt{256}$, then we have $\|\boldsymbol{s}\| \leq \frac{2}{\sqrt{26}}\delta\mathfrak{s}_3\sqrt{256}$, except with negligible probability $2^{-128}$ if there is no overflow modulo $q$. We get $\delta^{256}\exp(128(1-\delta^2)) = 2^{-128}$ by choosing $\delta = 1.64$. So if we instead let the verifier check if $\|\vec{z}\| \leq 1.64\mathfrak{s}_3\sqrt{256}$, we have a protocol $\Pi^2_{\mathsf{arp}}$ that convinces the verifier that $\|\boldsymbol{s}\| \leq 2\sqrt{\frac{337\cdot 256}{26}}1.64\gamma_3 B$. We note that in order to achieve no overflow modulo $q$, we must require that

$$q \geq 41(m_1+\ell)N\frac{2}{\sqrt{26}}1.64\mathfrak{s}_3\sqrt{256},$$

as explained in Lemma 6, since in our case $m = (m_1+\ell)N$ and $b = \frac{2}{\sqrt{26}}1.64\mathfrak{s}_3\sqrt{256}$.

The protocol for $p \in \{2, \infty\}$ is given as $\Pi^p_{\mathsf{arp}}$ in Figure 7.1. We omit the commit-and-prove simulatability of this protocol, since it will be implicit in the security analysis of the complete protocol. We do however consider the correctness and knowledge soundness of the protocol.

**Theorem 21.** Suppose that $\mathfrak{s}_3 = \gamma_3\sqrt{337}B$ for some $\gamma_3 > 0$. Then the protocol $\Pi^p_{\mathsf{arp}}$ is complete,

$\underline{\Pi^p_{\mathsf{arp}}}$

Public Information: $\boldsymbol{A}_1 \in R_q^{n\times m_1}$, $\boldsymbol{A}_2 \in R_q^{n\times m_2}$, $\boldsymbol{B} \in R_q^{\ell\times m_2}, \boldsymbol{B}_2 \in R_q^{256\times m_2}, \boldsymbol{b}_2 \in R_q^{m_2}$.
$\sigma \in \mathsf{Aut}(R_q)$.
Prover's Information: $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{m}) \in R_q^{m_1+\ell}$ so that $\|\boldsymbol{s}\| \le B$, $\boldsymbol{s}_2 \in R_q^{m_2}$
Commitment: $\begin{bmatrix}\boldsymbol{t}_A \\ \boldsymbol{t}_B\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}_1 \\ \boldsymbol{0}\end{bmatrix}\boldsymbol{s}_1 + \begin{bmatrix}\boldsymbol{A}_2 \\ \boldsymbol{B}\end{bmatrix}\boldsymbol{s}_2 + \begin{bmatrix}\boldsymbol{0} \\ \boldsymbol{m}\end{bmatrix}$

| Prover | | Verifier |
|---|---|---|
| $b \xleftarrow{\$} \{-1,1\} \subset R_q$<br>$\boldsymbol{y}_3 \xleftarrow{\$} D_{R^{256/N},\mathfrak{s}_3}$<br>$t_b = \boldsymbol{b}_1\boldsymbol{s}_2 + b$<br>$\boldsymbol{t}_y = \boldsymbol{B}_2\boldsymbol{s}_2 + \boldsymbol{y}_3$ | | |
| | $\xrightarrow{t_b, \boldsymbol{t}_y}$ | |
| | | $R \xleftarrow{\$} \mathsf{Bin}_1^{256\times(m_1+\ell)N}$ |
| | $\xleftarrow{R}$ | |
| $\vec{z} := bR\vec{s} + \vec{y}_3$<br>If $\mathsf{Rej}_0(\vec{z}, bR\vec{s}, \mathfrak{s}_3) = 1$, abort | | |
| | $\xrightarrow{\vec{z}}$ | |
| Run $\Pi := \Pi^{(2)}_{\mathsf{eval}}((\boldsymbol{s}_2, \boldsymbol{s}_1, (\boldsymbol{m}, b)), \sigma, f, \Psi)$ | | |
| | | Accept iff:<br>1. $\Pi$ verifies<br>2. If $p = 2$ :<br>$\|\vec{z}\| \le 1.64\sqrt{256}\mathfrak{s}_3$<br>If $p = \infty$ :<br>$\|\vec{z}\|_\infty \le \sqrt{256}\mathfrak{s}_3$ |

Figure 7.1: Proof of knowledge $\Pi^p_{\mathsf{arp}}((\boldsymbol{s}_2, \boldsymbol{s}_1, \boldsymbol{m}), \sigma)$ of $((\boldsymbol{s}_1, \boldsymbol{m}), \boldsymbol{s}_2, \overline{c}) \in R_q^{m_1+\ell} \times R_q^{m_2} \times \overline{\mathcal{C}}$ that satisfy $\boldsymbol{A}_1\boldsymbol{s}_1 + \boldsymbol{A}_2\boldsymbol{s}_2 = \boldsymbol{t}_A$, $\boldsymbol{B}\boldsymbol{s}_2 + \boldsymbol{m} = \boldsymbol{t}_B$, $\|\overline{c}\boldsymbol{s}_i\| \le 2\mathfrak{s}_i\sqrt{2m_iN}$ for $i = 1, 2$, and $\|\boldsymbol{s}\| \le 2\sqrt{\frac{337\cdot256}{26}}1.64\gamma_3 B$ if we use $p = 2$, or $\|\boldsymbol{s}\|_\infty \le 2\gamma_3\sqrt{256\cdot337}B$ if we use $p = \infty$.

meaning that an honest prover will convince an honest verifier with probability

$$\approx \frac{P_{\mathsf{eval}}}{\exp\left(\frac{1}{2\gamma_3^2}\right)},$$

where $P_{\mathsf{eval}}$ is the success probability of $\Pi$.

*Proof.* By Lemma 8, we have for an honest prover that

$$\tilde{F}_i(\boldsymbol{s}, \boldsymbol{y}, b) = z_i - b\langle \vec{r}_i, \vec{s}\rangle - y_1 = b\langle \vec{r}_i, \vec{s}\rangle - b\langle \vec{r}_i, \vec{s}\rangle = 0,$$

for all $i \in [256]$ since $\vec{z} = R\vec{s} + \vec{y}$. $f(b) = 0$ will clearly hold since $b \in \{-1, 1\}$. Also by using Lemma 8, we get that $\tilde{G}_j(b) = 0$ for all $j \in [N]$, since $b$ is a sign. By the discussion above, $\Pi$ is correctly instantiated, and has success probability $P_{\mathsf{eval}}$. According to Lemma 4, the probability that the protocol does not reject is $1/\exp(1/(2\gamma_3^2))$.

If we now consider $p = 2$, we get by Lemma 3 that $\|\boldsymbol{z}\| \leq 1.64\sqrt{256}\mathfrak{s}_3$, except with negligible probability $2^{-128}$. If we consider $p = \infty$, we get by Lemma 3 that $\|\boldsymbol{z}\|_\infty \leq \sqrt{256}\mathfrak{s}_3$, except with negligible probability $256 \cdot 2 \cdot 2^{-128}$. Hence the theorem holds. □

**Theorem 22.** Let $\mathfrak{s}_3 = \gamma_3\sqrt{337}B$ for some $\gamma_3 > 0$, and assume that $q \geq 41(m_1+\ell)N\frac{2}{\sqrt{26}}1.64\mathfrak{s}_3\sqrt{256}$. Let $P_{\mathcal{E}'}$ and $T_{\mathcal{E}'}$ be the success probability and run time of the extractor $\mathcal{E}'$ from Theorem 20. Then the protocol $\Pi^p_{\mathsf{arp}}$ is knowledge sound, meaning that there exists an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a probabilistic prover $\mathcal{P}^*$, which convinces the verifier with probability $\varepsilon \geq 2/|\mathcal{C}| + q^{-N/2} + q^{-\lambda} + 2^{-128}$, the extractor $\mathcal{E}$ either breaks the binding of the commitment or recovers a valid opening $(\overline{\boldsymbol{s}}_2, \overline{\boldsymbol{s}}_1, \overline{\boldsymbol{y}}, \overline{\boldsymbol{m}}, \bar{b}, \bar{c})$ to the commitment $(\boldsymbol{t}, t_b, \boldsymbol{t}_y)$ with either $\|(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}})\| \leq 2\sqrt{\frac{337\cdot 256}{26}}1.64\gamma_3 B$ for $p = 2$ or $\|(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}})\|_\infty \leq 2\gamma_3\sqrt{256 \cdot 337}B$ for $p = \infty$, with probability $P_{\mathcal{E}}(1 - 2^{-128})$ in expected time $2T_{\mathcal{E}}$.

*Proof.* We let $\mathcal{E}'$ be the extractor from Theorem 20, and define $\mathcal{E}$ as follows.

1. We let $\mathcal{E}$ run until the third round on an honestly generated challenge $R$, and then let it run $\mathcal{E}'$. It aborts if $\mathcal{E}'$ does not obtain a valid opening $(\overline{\boldsymbol{s}}, \overline{\boldsymbol{y}}, \bar{b})$, where $\overline{\boldsymbol{s}} = (\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{m}})$, that satisfies the relations in $\Psi$.

2. $\mathcal{E}$ rewinds the prover until the third round, and new honestly generated challenge $R'$ and then run $\mathcal{E}'$ until it obtains a valid opening $(\overline{\boldsymbol{s}}', \overline{\boldsymbol{y}}', \bar{b}')$, where $\overline{\boldsymbol{s}}' = (\overline{\boldsymbol{s}}'_1, \overline{\boldsymbol{m}}')$, that satisfies the relations in $\Psi$.

By the same argumentation as in Theorem 17, the expected run time of $\mathcal{E}$ is twice the expected run time of $\mathcal{E}'$, and $\mathcal{E}$ has the same success probability as $\mathcal{E}'$. Since $\varepsilon \geq 2/|\mathcal{C}| + q^{-N/2} + q^{-\lambda} + 2^{-128}$ and the success probability of the prover at producing a valid $\Pi$ is at least $2/|\mathcal{C}| + q^{-N/2} + q^{-\lambda}$, we therefore have $P_{\mathcal{E}'} \geq \varepsilon - 2/|\mathcal{C}| - q^{-N/2} - q^{-\lambda} \geq 2^{-128}$.

If the extracted openings are such that $(\overline{\boldsymbol{s}}, \overline{\boldsymbol{y}}) \neq (\overline{\boldsymbol{s}}', \overline{\boldsymbol{y}}')$, then $\mathcal{E}$ breaks the binding property of the commitment scheme. So we assume that $(\overline{\boldsymbol{s}}, \overline{\boldsymbol{y}}) = (\overline{\boldsymbol{s}}', \overline{\boldsymbol{y}}')$ and that $\mathcal{E}$ finds $(\overline{\boldsymbol{s}}_2, \overline{\boldsymbol{s}}_1, \overline{\boldsymbol{y}}, \overline{\boldsymbol{m}}, \bar{b}) \in R_q^{m_2+m_1+256/N+\ell+1}$ and $\bar{c} \in R_q^\times$ such that $(\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{y}}, \overline{\boldsymbol{m}}, \bar{b})$ are valid $\mathsf{ABDLOP}$ messages for the randomness $\overline{\boldsymbol{s}}_2$ and

1. $f(\bar{b}) = 0$ and $f(\bar{b}') = 0$,

2. For $i \in [256]$, $\tilde{F}_i(\overline{\boldsymbol{s}}, \overline{\boldsymbol{y}}, \overline{b}) = 0$ and $\tilde{F}_i(\overline{\boldsymbol{s}}, \overline{\boldsymbol{y}}, \overline{b}') = 0$

3. For $j \in [N]$, $\tilde{G}_j(\overline{b}) = 0$ and $\tilde{G}_j(\overline{b}') = 0$.

Using Lemma 8 we get that 3) implies that every coefficient of $\overline{b}$ is zero, except the constant one. Hence $\overline{b} \in \mathbb{Z}_q$, and then 1) implies that $\overline{b}$ is a sign. The same holds for $\overline{b}'$. 2) implies, by using Lemma 8, that $\vec{z} = \overline{b}R\vec{s} + \vec{\overline{y}}$ has correct form. It also implies that $\vec{z} = \overline{b}'R'\vec{\overline{s}}' + \vec{\overline{y}}'$, where $\overline{b}'$, $R'$ are independent of $(\overline{\boldsymbol{s}}, \overline{\boldsymbol{y}}) = (\overline{\boldsymbol{s}}', \overline{\boldsymbol{y}}')$. Since $\overline{b}'$ and $R'$ are independent of $\vec{\overline{s}}$, the verification equations now give

– For $p = 2$:

$$\begin{aligned}\|\vec{z}\| = \|\overline{b}R'\vec{\overline{s}} + \vec{\overline{y}} \mod q\| &\leq 1.64\sqrt{256}\mathfrak{s}_3 = 1.64\sqrt{256}\gamma_3\sqrt{337}B \\ &= \frac{1}{2}\sqrt{26}\big(2\sqrt{\frac{337 \cdot 256}{26}}1.64\gamma_3 B\big).\end{aligned}$$

Lemma 6 now states that if $\|\overline{\boldsymbol{s}}\| \geq 2\sqrt{\frac{337 \cdot 256}{26}}1.64\gamma_3 B$, then this is true only with probability $2^{-128}$, because of the assumption on $q$. Hence we have that $\|\overline{\boldsymbol{s}}\| \leq 2\sqrt{\frac{337 \cdot 256}{26}}1.64\gamma_3 B$ with probability $1 - 2^{-128}$.

– For $p = \infty$:

$$\|\vec{z}\|_\infty = \|\overline{b}R'\vec{\overline{s}} + \vec{\overline{y}}\|_\infty \leq \sqrt{256}\mathfrak{s}_3 = \sqrt{256}\gamma_3\sqrt{337}B,$$

which by Lemma 5 implies that $\|\overline{\boldsymbol{s}}\|_\infty \leq 2\gamma_3\sqrt{2 \cdot 256 \cdot 337}B$ with probability $1 - 2^{-128}$.

Hence the theorem holds. □

## 7.2 Proving exact norm bounds and that a vector is binary

Both for proving Equation (7.5), and as we soon will see, for proving Equation (7.4), we need to be able to prove that a vector is binary. So, suppose that we want to prove that a vector $\vec{x} \in \mathbb{Z}^n$ has binary coefficients, meaning that $\vec{x} \in \{0,1\}^n$. Since we have efficient schemes for proving inner product relations, efficient proof schemes for proving that a vector is binary can be constructed by using the following result.

**Lemma 18.** Let $n \in \mathbb{N}$ and $\vec{x} \in \mathbb{Z}^n$. If $\langle \vec{x}, \vec{x} - \vec{1}_n \rangle = 0$, then $\vec{x} \in \{0,1\}^n$.

*Proof.* If we let $x = (x_1 \ \ldots \ x_n)$, then every term in the inner product $\langle \vec{x}, \vec{x} - \vec{1}_n \rangle$ is of the form $x_i(x_i - 1)$. We can now use the fact that

$$\forall a \in \mathbb{Z},\ a(a-1) \geq 0$$

and $a(a-1) = 0$ only if $a \in \{0,1\}$. Hence if $\langle \vec{x}, \vec{x} - \vec{1}_n \rangle = 0$, then $x_i \in \{0,1\}$ for all $i = 1, \ldots, n$, and we thus have $\vec{x} \in \{0,1\}^n$. □

So in order to prove that a vector $\vec{x} \in \mathbb{Z}^n$ has binary coefficients, we can just prove that $\langle \vec{x}, \vec{x} - \vec{1}_n \rangle = 0$. This can be achieved by first proving that $\langle \vec{x}, \vec{x} - \vec{1}_n \rangle = 0 \mod q$, and then by proving

that $\|\vec{x}\| \leq B$ for some bound $B$. The first part can effectively be done by using the protocol $\Pi^{(2)}_{\mathsf{eval}}$, with the function

$$\mathsf{T}(\vec{x}, \vec{x} - \vec{1}_n)$$

as evaluation function, since proving that the constant coefficient of this function is equal to zero, implies that $\langle \vec{x}, \vec{x} - \vec{1}_n \rangle = 0$. The second part can be proved by using the protocol $\Pi^{2}_{\mathsf{arp}}$. This will suffice if $B$ is such that $B^2 + \sqrt{n}B < q$, since

$$\left| \sum_{i=1}^{n} x_i(x_i - 1) \right| \leq \sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} |x_i| \leq \sum_{i=1}^{n} x_i^2 + n\sqrt{\frac{\sum_{i=1}^{n} x_i^2}{n}} \leq B^2 + B\sqrt{n},$$

and this would then imply that $\langle \vec{x}, \vec{x} - \vec{1}_n \rangle = 0$ also over the integers, since there will be no overflow modulo $q$. Thus we have proved that $\vec{x}$ is binary.

### 7.2.1 Proving exact norm bounds

We are now ready to explain how we can construct proofs of exact $\ell_2$ norm, which we need in order to prove Equation (7.4). So, suppose that we want to prove that $\|\boldsymbol{s}\| \leq B$ for the true bound $B$. We remember that we can use the protocol $\Pi^{(2)}_{\mathsf{eval}}$ to effectively prove that the inner product of two commitments modulo $q$ is some constant, and that $\|\boldsymbol{s}\|^2 = \langle \vec{s}, \vec{s} \rangle$. Hence in the case that $\|\boldsymbol{s}\| = B$, we could prove the norm exactly by using $\Pi^{(2)}_{\mathsf{eval}}$ to prove that $\|\boldsymbol{s}\|^2 = B^2 \mod q$, and by using $\Pi^{2}_{\mathsf{arp}}$ to show that there is no overflow modulo $q$.

The problem with this method is that we then give away the exact norm of $\boldsymbol{s}$, thus revealing information about $\boldsymbol{s}$. So we instead set out to prove that the difference between the bound and the norm is a positive integer. In order to do this, we prove that $B^2 - \langle \vec{s}, \vec{s} \rangle$ can be written with a binary representation $\vec{x}$ of length $2\log(B) \leq N$. More precisely, we define $\vec{p} := \begin{pmatrix} 1 & 2 & \dots & 2^{2\log B} & 0 & \dots & 0 \end{pmatrix}$, and let $\vec{x}$ be the binary vector such that

$$\langle \vec{p}, \vec{x} \rangle = B^2 - \|\vec{s}\|^2.$$

We then commit to $\vec{x}$ and prove that it is binary with the technique from the previous section, and prove that the above equation holds over the integers by using $\Pi^{2}_{\mathsf{arp}}$.

## 7.3 The complete protocol

We are now ready explain how we can construct a protocol for proving knowledge of $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{m}, \sigma(\boldsymbol{s}_1), \sigma(\boldsymbol{m}))$ satisfying equations (7.1) to (7.5) using a single instantiation of $\Pi^{(2)}_{\mathsf{eval}}$. As we have seen, the first two equations are straight forward, and we can just pass the functions $f_i, F_i$ to $\Pi^{(2)}_{\mathsf{eval}}$ as quadratic functions and evaluation functions, respectively. Equation (7.3) is now a direct application of $\Pi^{\infty}_{\mathsf{arp}}$ with vector input

$$\boldsymbol{e}^{(d)} := \begin{bmatrix} \boldsymbol{D}_1 \boldsymbol{s} - \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{D}_{v_d} \boldsymbol{s} - \boldsymbol{u}_{v_d} \end{bmatrix}. \tag{7.6}$$

This means that we, as per Section 7.1, let the prover draw the masking vector $\boldsymbol{y}^{(d)} \xleftarrow{\$} D_{R^{256/N}, \mathfrak{s}^{(d)}}$ and a sign $b^{(d)} \xleftarrow{\$} \{-1, 1\}$, and append commitments to them in the BDLOP part. We set the

challenge dimension as $c^{(d)} = N\sum_{i=1}^{v_d} k_i$, which is the dimension of $\vec{e}^{(d)}$. Upon receiving the challenge matrix $R^{(d)} \xleftarrow{\$} \mathsf{Bin}_1^{256\times c^{(d)}}$ from the verifier, the prover will perform bimodal rejection sampling on $\vec{z}^{(d)} := b^{(d)}R^{(d)}\vec{e}^{(d)} + \vec{y}^{(d)}$. The verifier can now check that $\|\vec{z}^{(d)}\|_\infty \leq \sqrt{256}\mathfrak{s}^{(d)}$. In order to prove that $\vec{z}^{(d)}$ was formed correctly, and that $b^{(d)}$ indeed is a sign, we define the functions

$$
\begin{aligned}
&H_j^{(d)}(\boldsymbol{s}, \boldsymbol{y}^{(d)}, b^{(d)}) := z_j^{(d)} - \mathsf{T}(b^{(d)}\vec{r}_j^{(d)}, \vec{e}^{(d)}) - y_j^{(d)}, \forall j \in [256],\\
&g^{(d)}(b^{(d)}) := (b^{(d)} - 1)(b^{(d)} + 1),\\
&J_j^{(d)}(b^{(d)}) := \mathsf{T}(\delta_j, b^{(d)}), \forall j \in [N],
\end{aligned}
$$

which we will pass to $\Pi_{\mathsf{eval}}^{(2)}$ as quadratic and evaluation functions.

As we saw in the previous section, in order to prove equation (7.4), we prove that $(\beta_i^{(e)})^2 - \|\boldsymbol{E}_i\boldsymbol{s} - \boldsymbol{v}_i\|^2$ can be written with a binary representation $\vec{x}_i$ of length $2\log(\beta_i^{(e)})$ for all $1 \leq i \leq v_e$. So at the start of the protocol, the prover appends a commitment to the binary representation vector $\boldsymbol{x} = (x_1\| \ldots \|x_{v_e})$ in the Ajtai part of the commitment, since it is small. We can now define $\boldsymbol{x}'$ to be the concatenation of $\boldsymbol{x}$ and $\boldsymbol{E}_{\mathsf{bin}}\boldsymbol{s} - \boldsymbol{v}_{\mathsf{bin}}$, such that $\boldsymbol{x}'$ contains everything we want to prove is binary. Hence in order to prove Equations (7.4) and (7.5), we define

$$
\boldsymbol{e}^{(e)} := \begin{bmatrix} \boldsymbol{E}_1\boldsymbol{s} - \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{E}_{v_e}\boldsymbol{s} - \boldsymbol{v}_{v_e} \\ \boldsymbol{x}' \end{bmatrix} \text{ and } \vec{p}_i := \begin{pmatrix} 1 & 2 & \ldots & 2^{2\log(\beta_i^{(e)})} & 0 & \ldots & 0 \end{pmatrix} \ \forall i \in [v_e], \tag{7.7}
$$

and set out to prove the following.

$$
\begin{aligned}
&\langle \boldsymbol{x}', \boldsymbol{x}' - \mathbf{1}_{(v_e+k_{\mathsf{bin}})N}\rangle = 0 \mod q,\\
&\langle \boldsymbol{E}_i\boldsymbol{s} - \boldsymbol{v}_i, \boldsymbol{E}_i\boldsymbol{s} - \boldsymbol{v}_i\rangle + \langle \vec{p}_i, \vec{x}_i\rangle = (\beta_i^{(e)})^2 \mod q, \forall i \in [v_e],\\
&\|\boldsymbol{e}^{(e)}\| \text{ is small enough so that the above equations hold over } \mathbb{Z}.
\end{aligned}
$$

According to Section 7.2 this is sufficient for proving both (7.4) and (7.5).

We now explain how we can construct functions to initialize $\Pi_{\mathsf{eval}}^{(2)}$ to prove the above. We recall that the constant coefficient of $\mathsf{T}$ equals the inner product of its inputs, and we can therefore define the following functions

$$
\begin{aligned}
&G(\boldsymbol{x}') := \mathsf{T}(\boldsymbol{x}', \boldsymbol{x}' - \mathbf{1}_{(v_e+k_{\mathsf{bin}})N}),\\
&I_i(\boldsymbol{s}, \boldsymbol{x}) := \mathsf{T}(\boldsymbol{E}_i\boldsymbol{s} - \boldsymbol{v}_i, \boldsymbol{E}_i\boldsymbol{s} - \boldsymbol{v}_i) + \mathsf{T}(\vec{p}_i, \vec{x}_i) - (\beta_i^{(e)})^2, \forall i \in [v_e],
\end{aligned}
$$

to pass to $\Pi_{\mathsf{eval}}^{(2)}$ as evaluation function in order to prove the above inner product relations.

In order to prove that the $\ell_2$ norm of $\boldsymbol{e}^{(e)}$ is small, we run $\Pi_{\mathsf{arp}}^2$ on $\boldsymbol{e}^{(e)}$. So, the prover will draw the masking vector $\boldsymbol{y}^{(e)} \xleftarrow{\$} D_{R^{256/N}, \mathfrak{s}^{(e)}}$ and a sign $b^{(e)} \xleftarrow{\$} \{-1, 1\}$, and append commitments to them in the $\mathsf{BDLOP}$ part. We set the challenge dimension as $c^{(e)} = N(k_{\mathsf{bin}} + \sum_{i=1}^{v_e}(p_i + 1))$, which is the dimension of $\vec{e}^{(e)}$. Upon receiving the challenge matrix $R^{(e)} \xleftarrow{\$} \mathsf{Bin}_1^{256\times c^{(e)}}$ from the verifier, the prover will perform bimodal rejection sampling on $\vec{z}^{(e)} := b^{(e)}R^{(e)}\vec{e}^{(e)} + \vec{y}^{(e)}$. The verifier can now

check that $\|\vec{z}^{(e)}\| \leq 1.64\sqrt{256}\mathfrak{s}^{(e)}$. In order to prove that $\vec{z}^{(e)}$ was formed correctly, and that $b^{(e)}$ indeed is a sign, we define the functions

$$\begin{aligned}
&H_j^{(e)}(\boldsymbol{x}', \boldsymbol{s}, \boldsymbol{y}^{(e)}, b^{(e)}) := z_j^{(e)} - \mathsf{T}(b^{(e)}\vec{r}_j^{(e)}, \vec{e}^{(e)}) - y_j^{(e)}, \forall j \in [256],\\
&g^{(e)}(b^{(e)}) := (b^{(e)} - 1)(b^{(e)} + 1),\\
&J_j^{(e)}(b^{(e)}) := \mathsf{T}(\delta_j, b^{(e)}), \forall j \in [N],
\end{aligned}$$

which we will pass to $\Pi_{\mathsf{eval}}^{(2)}$ as quadratic and evaluation functions. We now define the set of all the quadratic functions and all the evaluation functions that we will pass to $\Pi_{\mathsf{eval}}^{(2)}$, respectively, as

$$\phi := (f_1, \ldots, f_\rho, g^{(d)}, g^{(e)}), \tag{7.8}$$

$$\Psi := \big(F_i, \ldots, F_{\rho_{\mathsf{eval}}}, G, (H_j^{(d)})_{j\in[256]}, (H_j^{(e)})_{j\in[256]}, (I_i)_{i\in v_e}, (J_j^{(d)})_{j\in[N]}, (J_j^{(e)})_{j\in[N]}\big). \tag{7.9}$$

The complete protocol is given as $\Pi$ in Figure 7.2.

**Theorem 23.** Suppose that $\mathfrak{s}^{(d)} = \gamma^{(d)}\sqrt{337}\alpha^{(d)}$ and $\mathfrak{s}^{(e)} = \gamma^{(e)}\sqrt{337}\alpha^{(e)}$ for some $\gamma^{(d)}, \gamma^{(e)} > 0$. Let $B^{(e)} := 2\sqrt{\frac{256}{26}}1.64\gamma^{(e)}\sqrt{337}\alpha^{(e)}$, the bound on $\boldsymbol{b}^{(e)}$ that we are able to prove with $\Pi_{\mathsf{arp}}^2$, and assume that

$$2(\max_{i\in[v_e]}\beta_i^{(e)})^2 + (B^{(e)})^2 - 1 < q$$

Then the protocol $\Pi$ is complete, meaning that an honest prover will convince an honest verifier with probability

$$\approx \frac{P_{\mathsf{eval}}}{\exp\left(\frac{1}{2(\gamma^{(d)})^2}\right)\exp\left(\frac{1}{2(\gamma^{(e)})^2}\right)},$$

where $P_{\mathsf{eval}}$ is the success probability of $\Pi^*$.

*Proof.* The probability that the honest prover succeeds is at least the probability that

1. Both rejection sampling steps on $\vec{z}^{(d)}$ and $\vec{z}^{(e)}$ do not abort. According to Lemma 4 each rejection sampling have independent probability of respectively $\exp\left(-\frac{1}{2(\gamma^{(d)})^2}\right)$ and $\exp\left(-\frac{1}{2(\gamma^{(e)})^2}\right)$ to not abort.
2. Both norm checks are satisfied. As we discussed in Theorem 21, the probability that the $\ell_\infty$ norm check is verified is $1 - 512 \cdot 2^{-128}$, and the probability that the $\ell_2$ norm check is verified is $1 - 2^{-128}$.
3. The protocol $\Pi^*$ successfully convinces the verifier.

We must therefore show that $\Pi^*$ is a valid instantiation of $\Pi_{\mathsf{eval}}^{(2)}$, and that it therefore convinces the verifier with probability $P_{\mathsf{eval}}$. We assume that $\boldsymbol{s}$ satisfies Equations (7.1) to (7.5), since the prover is honest. We start by considering the quadratic functions $\phi$, and see that for all $1 \leq i \leq \rho$, Equation (7.1) then implies that $f_i(\boldsymbol{s}) = 0$. Since both $b^{(d)}$ are signs, we also have that $g^{(d)}(b^{(d)}) = g^{(e)}(b^{(e)}) = 0$.

We then consider the evaluation functions $\Psi$, and see that Equation (7.2) implies that $\tilde{F}_i(\boldsymbol{s}) = 0$ for all $1 \leq i \leq \rho_{\mathsf{eval}}$. By construction and Equation (7.5), the vector $\boldsymbol{x}'$ is binary. Hence by Lemma 8 $G(\boldsymbol{x}') = \langle \boldsymbol{x}', \boldsymbol{x}' - \mathbf{1}\rangle \mod q$, since all the non-constant coefficients of $\mathsf{T}(\boldsymbol{x}', \boldsymbol{x}' - \mathbf{1})$ will then vanish,

$\Pi$

Public Information: $\boldsymbol{A}_1 \in R_q^{n\times(m_1+v_e)}$, $\boldsymbol{A}_2 \in R_q^{n\times m_2}$, $\boldsymbol{B} \in R_q^{\ell\times m_2}, \boldsymbol{B}^{(d)}, \boldsymbol{B}^{(e)} \in R_q^{256/N\times m_2}$, $\boldsymbol{b}^{(d)}, \boldsymbol{b}^{(e)} \in R_q^{m_2}$. $\sigma := \sigma_{-1} \in \mathsf{Aut}(R_q)$
Public parameters as defined in the start of Chapter 7.
Bounds $\alpha^{(d)}, \alpha^{(e)}$ such that $\|\boldsymbol{e}^{(d)}\| \le \alpha^{(d)}$, $\|\boldsymbol{e}^{(e)}\| \le \alpha^{(e)}$
$\phi$ and $\Psi$ as defined in (7.8) and (7.9).
Prover's Information: $\boldsymbol{s} = (\boldsymbol{s}_1, \boldsymbol{m}) \in R_q^{m_1+\ell}$ and randomness $\boldsymbol{s}_2 \in R_q^{m_2}$ so that Equations (7.1) to (7.5) holds. Binary decomposition $x_i \in R_q$ of $(\beta_i^{(e)})^2 - \|\boldsymbol{E}_i\boldsymbol{s} - \boldsymbol{v}_i\|^2$. Vectors $\boldsymbol{e}^{(d)}$, $\boldsymbol{e}^{(e)}$ as defined in (7.6) and (7.7).
Commitment: $\boldsymbol{t} = \begin{bmatrix}\boldsymbol{A}_1\\ \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{s}_1\\ \boldsymbol{x}\end{bmatrix} + \begin{bmatrix}\boldsymbol{A}_2\\ \boldsymbol{B}\end{bmatrix}\boldsymbol{s}_2 + \begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{m}\end{bmatrix}$

| Prover | | Verifier |
|---|---|---|
| $b^{(d)}, b^{(e)} \xleftarrow{\$} \{-1, 1\} \subset R_q$<br>$\boldsymbol{y}^{(d)} \xleftarrow{\$} D_{R^{256/N}, \mathfrak{s}^{(d)}}$<br>$\boldsymbol{y}^{(e)} \xleftarrow{\$} D_{R^{256/N}, \mathfrak{s}^{(e)}}$<br>$\boldsymbol{t}^{(d)} = \boldsymbol{B}^{(d)}\boldsymbol{s}_2 + \boldsymbol{y}^{(d)}$<br>$\boldsymbol{t}^{(e)} = \boldsymbol{B}^{(e)}\boldsymbol{s}_2 + \boldsymbol{y}^{(e)}$<br>$t^{(d)} = \boldsymbol{b}^{(d)}\boldsymbol{s}_2 + b^{(d)}$<br>$t^{(e)} = \boldsymbol{b}^{(e)}\boldsymbol{s}_2 + b^{(e)}$ | | |
| | $\xrightarrow{\boldsymbol{t}^{(d)}, t^{(d)}, \boldsymbol{t}^{(e)}, t^{(e)}}$ | |
| | | $R^{(d)} \xleftarrow{\$} \mathsf{Bin}_1^{256\times c^{(d)}}$<br>$R^{(e)} \xleftarrow{\$} \mathsf{Bin}_1^{256\times c^{(e)}}$ |
| | $\xleftarrow{R^{(d)}, R^{(e)}}$ | |
| $\vec{z}^{(d)} := b^{(d)}R^{(d)}\vec{e}^{(d)} + \vec{y}^{(d)}$<br>$\vec{z}^{(e)} := b^{(e)}R^{(e)}\vec{e}^{(e)} + \vec{y}^{(e)}$<br>If $\mathsf{Rej}_0(\vec{z}^{(d)}, b^{(d)}R^{(d)}\vec{e}^{(d)}, \mathfrak{s}^{(d)}) = 1$ or<br>$\mathsf{Rej}_0(\vec{z}^{(e)}, b^{(e)}R^{(e)}\vec{e}^{(e)}, \mathfrak{s}^{(e)}) = 1$, abort<br>$\boldsymbol{s}^* := (\boldsymbol{s}_2, (\boldsymbol{s}_1, \boldsymbol{x}), (\boldsymbol{m}, \boldsymbol{y}^{(d)}, \boldsymbol{y}^{(e)}, b^{(d)}, b^{(e)}))$ | | |
| | $\xrightarrow{\vec{z}^{(d)}, \vec{z}^{(e)}}$ | |
| Run $\Pi^* := \Pi^{(2)}_{\mathsf{eval}}(\boldsymbol{s}^*, \sigma, \phi, \Psi)$ | | |
| | | Accept iff:<br>1. $\Pi^*$ verifies<br>2. $\|\vec{z}^{(d)}\|_\infty \le \sqrt{256}\mathfrak{s}^{(d)}$<br>3. $\|\vec{z}^{(e)}\| \le 1.64\sqrt{256}\mathfrak{s}^{(e)}$ |

Figure 7.2: Proof of knowledge $\Pi((\boldsymbol{s}_2, \boldsymbol{s}_1, \boldsymbol{m}), \sigma)$ of $((\boldsymbol{s}_1, \boldsymbol{m}), \boldsymbol{s}_2, \overline{c}) \in R_q^{m_1+\ell} \times R_q^{m_2} \times \overline{\mathcal{C}}$ that satisfy $\boldsymbol{A}_1\boldsymbol{s}_1 + \boldsymbol{A}_2\boldsymbol{s}_2 = \boldsymbol{t}_A$, $\boldsymbol{B}\boldsymbol{s}_2 + \boldsymbol{m} = \boldsymbol{t}_B$, $\|\overline{c}\boldsymbol{s}_i\| \le 2\mathfrak{s}_i\sqrt{2m_iN}$ for $i = 1, 2$, and $\boldsymbol{s} := (\boldsymbol{s}_1, \boldsymbol{m}, \sigma(\boldsymbol{s}_1), \sigma(\boldsymbol{m}))$ verifies Equations (7.1) to (7.5).

by the way $\mathsf{T}$ is constructed. Then by Lemma 18 we have $\tilde{G}(\boldsymbol{x}') = 0$.

We also have by Lemma 8 that if $\vec{z}^{(d)}$ and $\vec{z}^{(e)}$ are constructed honestly, then $\tilde{H}_j^{(d)} = z_j^{(d)} - y_j^{(d)} - \langle b^{(d)}\vec{r}_j^{(d)}, \vec{e}^{(d)}\rangle = z_j^{(d)} - z_j^{(d)} = 0$, and similarly $\tilde{H}_j^{(e)} = 0$. Since $\vec{x}_i$ is the binary decomposition of $(\beta_i^{(e)})^2 - \|\boldsymbol{E}_i\boldsymbol{s} - \boldsymbol{v}_i\|^2$, it has support at most $2\log\beta_i^{(e)} \leq N$. Since $\vec{p}_i$ is defined defined as a vector of powers of two until $2\log\beta_i^{(e)}$, followed by zeros, we have

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

for all $i \in [v_e]$. We can again use Lemma 8 combined with this, to see that $\tilde{I}_i(\boldsymbol{s}, \boldsymbol{x}) = \langle \boldsymbol{E}_i\boldsymbol{s} - \boldsymbol{v}_i, \boldsymbol{E}_i\boldsymbol{s} - \boldsymbol{v}_i\rangle + \langle \vec{p}_i, \vec{x}_i\rangle - (\beta_i^{(e)})^2 = 0$ for all $i \in [v_e]$ when the inner products are taken modulo $q$. But this will also hold over the integers, since we assumed that $2(\max_{i\in[v_e]}\beta_i^{(e)})^2 + (B^{(e)})^2 - 1 < q$. Lastly, Lemma 8 implies that $\tilde{J}_j^{(d)} = \langle \delta_j, b^{(d)}\rangle = 0$ for $1 \leq j \leq N-1$, since $b^{(d)}$ is a constant. Similarly, $\tilde{J}_j^{(e)} = 0$.

Hence we have showed that $\Pi^*$ indeed is a valid instantiation of $\Pi_{\mathsf{eval}}^{(2)}$, and it therefore convinces the verifier with probability $P_{\mathsf{eval}}$. So the success probability of $\Pi$ is then at least $P_{\mathsf{eval}}(1 - 2^{-128})(1 - 512 \cdot 2^{-128})\exp\left(-\frac{1}{2(\gamma^{(d)})^2}\right)\exp\left(-\frac{1}{2(\gamma^{(e)})^2}\right)$, and the Lemma follows. □

**Theorem 24.** The protocol $\Pi$ is commit-and-prove simulatable, meaning that there exists a simulator $\mathcal{S}$ that, without access to private information $(\boldsymbol{s}_1, \boldsymbol{m})$, outputs a simulation of a commitment $(\boldsymbol{t}, \boldsymbol{t}^{(d)}, t^{(d)}, \boldsymbol{t}^{(e)}, t^{(e)})$ along with a non-aborting transcript of the protocol between the prover and the verifier such that for every algorithm $\mathcal{A}$ that has advantage $\varepsilon$ in distinguishing the simulated commitment and transcript from the real commitment and transcript, whenever the prover does not abort, there is an algorithm $\mathcal{B}$ with the same running time that has advantage $\varepsilon/2 - 2^{-128}$ in solving the $\mathsf{Extended\text{-}MLWE}_{n+\ell+\lambda+(256/N+1)+(256/N+1),m_2-n-\ell-\lambda-(256/N+1)-(256/N+1),\chi,\mathcal{C},\mathfrak{s}_2}$.

*Proof.* According to Lemma 4, $\boldsymbol{z}^{(d)}$ and $\boldsymbol{z}^{(e)}$ are within statistical distance of $2^{-128}$ from $D_{R^{256/N},\mathfrak{s}^{(d)}}$ and $D_{R^{256/N},\mathfrak{s}^{(e)}}$, and are independent of $R^{(d)}$ and $R^{(e)}$. Hence the simulator can just sample $\boldsymbol{z}^{(d)} \xleftarrow{\$} D_{R^{256/N},\mathfrak{s}^{(d)}}$, $\boldsymbol{z}^{(e)} \xleftarrow{\$} D_{R^{256/N},\mathfrak{s}^{(e)}}$, $R^{(d)} \xleftarrow{\$} \mathsf{Bin}_1^{256\times c^{(d)}}$ and $R^{(e)} \xleftarrow{\$} \mathsf{Bin}_1^{256\times c^{(e)}}$.

We simulate the appended commitments and the protocol $\Pi^*$ with the commit-and-prove simulator from the proof of in Theorem 19. Since the appended commitments have total dimension $\lambda + (256/N+1) + (256/N+1)$, we get that if there is an algorithm $\mathcal{A}$ with advantage $\varepsilon$ in distinguishing this transcript from a real one, there is an algorithm $\mathcal{A}'$ with advantage $\varepsilon/2 - 2^{-128}$ in solving the $\mathsf{Extended\text{-}MLWE}_{n+\ell+\lambda+(256/N+1)+(256/N+1),m_2-n-\ell-\lambda-(256/N+1)-(256/N+1),\chi,\mathcal{C},\mathfrak{s}_2}$ problem. □

**Theorem 25.** Let $B^{(d)} := 2\sqrt{256}\sqrt{337}\gamma^{(d)}\alpha^{(d)}$, $B^{(e)} := 2\sqrt{\frac{256}{26}}1.64\gamma^{(e)}\sqrt{337}\alpha^{(e)}$, the norm bounds we are able to prove with $\Pi_{\mathsf{arp}}^p$ for $\boldsymbol{e}^{(d)}$ and $\boldsymbol{e}^{(e)}$, and assume that

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

Then the protocol $\Pi$ is knowledge sound, meaning that there exists an extractor $\mathcal{E}$ with the following properties. When given rewindable black-box access to a probabilistic prover $\mathcal{P}$, which convinces

the verifier $\mathcal{V}$ with probability $\varepsilon \geq 2/|\mathcal{C}| + q^{-N/2} + q^{-\lambda} + 2^{-128}$, the extractor $\mathcal{E}$ either breaks the binding of the commitment or recovers a valid opening

$$(\overline{\boldsymbol{s}}_2, (\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{x}}), (\overline{\boldsymbol{m}}, \overline{\boldsymbol{y}}^{(d)}, \overline{\boldsymbol{y}}^{(e)}, \overline{b}^{(d)}, \overline{b}^{(e)}), \overline{c}) \in R_q^{m_1+m_2+\ell+v_d\cdot 256/N+v_e\cdot 256/N+1+1} \times R_q^{\times}$$

for the commitment $(\boldsymbol{t}, \boldsymbol{t}^{(d)}, t^{(d)}, \boldsymbol{t}^{(e)}, t^{(e)})$ with probability $P_{\mathsf{eval}}(1-2^{-256})(1-2^{-128})$ and in expected time $2T_{\mathsf{eval}}$, where $P_{\mathsf{eval}}$ and $T_{\mathsf{eval}}$ are success probability and the expected run time of the extractor from Theorem 19, satisfying Equations (7.1) to (7.5).

*Proof.* We let $\mathcal{E}'$ be the extractor from $\Pi^{(2)}_{\mathsf{eval}}$, and define the extractor $\mathcal{E}$ for this soundness proof in the following manner.

1. Run the prover until the third round on honestly generated challenges $R^{(d)}, R^{(e)}$, and then run $\mathcal{E}'$. Abort if $\mathcal{E}'$ does not obtain a valid opening $(\overline{\boldsymbol{s}}_2, (\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{x}}), (\overline{\boldsymbol{m}}, \overline{\boldsymbol{y}}^{(d)}, \overline{\boldsymbol{y}}^{(e)}, \overline{b}^{(d)}, \overline{b}^{(e)}), \overline{c}) \in R_q^{m_1+m_2+\ell+v_d\cdot 256/N+v_e\cdot 256/N+1+1} \times R_q^{\times}$ satisfying Equations (7.1) to (7.5).
2. Rewind the prover until the third round, send new honestly generated challenges $R^{(d)\prime}, R^{(e)\prime}$ and run $\mathcal{E}'$ until it obtains a valid opening $(\overline{\boldsymbol{s}}'_2, (\overline{\boldsymbol{s}}'_1, \overline{\boldsymbol{x}}'), (\overline{\boldsymbol{m}}', \overline{\boldsymbol{y}}^{(d)\prime}, \overline{\boldsymbol{y}}^{(e)\prime}, \overline{b}^{(d)\prime}, \overline{b}^{(e)\prime}), \overline{c}')$ satisfying Equations (7.1) to (7.5).

By the same argumentation as in Theorem 20, the expected run time of $\mathcal{E}$ it twice the expected run time of $\mathcal{E}'$, and $\mathcal{E}$ has the same success probability as $\mathcal{E}'$. Since $\varepsilon \geq 2/|\mathcal{C}| + q^{-N/2} + q^{-\lambda} + 2^{-128}$ and the success probability of the prover at producing a valid $\Pi^*$ is at least $2/|\mathcal{C}| + q^{-N/2} + q^{-\lambda}$, we therefore have $P_{\mathcal{E}'} \geq \varepsilon - 2/|\mathcal{C}| - q^{-N/2} - q^{-\lambda} \geq 2^{-128}$.

By the same argumentation as in the proof of Theorem 22, $\mathcal{E}$ will either find two valid openings to different messages and break the binding property of the commitment scheme, or the messages in both transcripts are the same. We focus on the latter case, which would imply that the challenge matrices $R^{(d)}$ and $R^{(e)}$ are independent of those messages. We must now show that this common message satisfies Equations (7.1) to (7.5).

Since $(\overline{\boldsymbol{s}}_2, (\overline{\boldsymbol{s}}_1, \overline{\boldsymbol{x}}), (\overline{\boldsymbol{m}}, \overline{\boldsymbol{y}}^{(d)}, \overline{\boldsymbol{y}}^{(e)}, \overline{b}^{(d)}, \overline{b}^{(e)}), \overline{c})$ is a valid opening, given how we initialized $\Pi^{(2)}_{\mathsf{eval}}$, we have that it satisfies the following:

1. For $i \in [\rho]$, $f_i(\overline{\boldsymbol{s}}, \sigma(\overline{\boldsymbol{s}})) = 0$
2. For $i \in [\rho_{\mathsf{eval}}]$, $\tilde{F}_i(\overline{\boldsymbol{s}}, \sigma(\overline{\boldsymbol{s}})) = 0$
3. $g^{(d)}(\overline{b}^{(d)}) = 0$, $g^{(e)}(\overline{b}^{(e)}) = 0$
4. $G(\overline{\boldsymbol{x}}') = 0$
5. For $j \in [256]$, $\tilde{H}^{(d)}_j(\overline{\boldsymbol{s}}, \overline{\boldsymbol{y}}^{(d)}, \overline{b}^{(d)}) = 0$
6. For $j \in [256]$, $\tilde{H}^{(e)}_j(\overline{\boldsymbol{s}}, \overline{\boldsymbol{y}}^{(e)}, \overline{b}^{(e)}) = 0$
7. For $i \in [v_e]$, $\tilde{I}_i(\overline{\boldsymbol{s}}, \overline{\boldsymbol{x}}) = 0$
8. For $j \in [N]$, $\tilde{J}^{(d)}_j(b^{(d)}) = 0$, $\tilde{J}^{(e)}_j(b^{(e)}) = 0$

It is obvious that 1) and 2) implies that this opening satisfies Equations (7.1) and (7.2). Now, 3) implies that both $\overline{b}^{(d)}$ and $\overline{b}^{(e)}$ are roots of $(X+1)(X-1)$, and 8) implies that all coefficients of $\overline{b}^{(d)}$ and $\overline{b}^{(e)}$ are zero, except for the constant coefficients. Since we are working over $\mathbb{Z}_q$, this implies that $\overline{b}^{(d)}$ and $\overline{b}^{(e)}$ are signs.

5) implies that $\vec{z}^{(d)} = \overline{b}^{(d)} R^{(d)} \vec{\overline{e}}^{(d)} + \vec{\overline{y}}^{(d)}$ is well-formed, where $\overline{\boldsymbol{e}}^{(d)}$ is defined as previous but with the extracted messages. We also have that $\|\vec{z}^{(d)}\|_\infty = \|\overline{b}^{(d)} R^{(d)} \vec{\overline{e}}^{(d)} + \vec{\overline{y}}^{(d)}\|_\infty \leq \sqrt{256}\mathfrak{s}^{(d)}$, from the norm verification on $\vec{z}^{(d)}$. We notice that since $\overline{b}^{(d)}$ is a sign, the distribution of $\overline{b}^{(d)} R^{(d)}$ is also $\mathsf{Bin}_1^{256 \times c^{(d)}}$, and therefore, since $\vec{\overline{e}}^{(d)}$ is fixed, we can use Lemma 5 to derive that the probability over $R^{(d)}$ that $\|\vec{z}^{(d)}\|_\infty \leq \frac{1}{2}\|\vec{\overline{e}}^{(d)}\|_\infty$ is less than $2^{-256}$. Hence we have that $\|\vec{e}^{(d)}\|_\infty \leq 32\mathfrak{s}^{(d)}$ with probability at least $1 - 2^{-256}$, and this opening satisfies equation (7.3).

Similarly, 6) implies that $\vec{z}^{(e)} = \overline{b}^{(e)} R^{(e)} \vec{\overline{e}}^{(e)} + \vec{\overline{y}}^{(e)}$ is well-formed, where $\overline{\boldsymbol{e}}^{(e)}$ is defined as previous but with the extracted messages. Since $\overline{b}^{(e)}$ is a sign, the distribution of $\overline{b}^{(e)} R^{(e)}$ is also $\mathsf{Bin}_1^{256 \times c^{(e)}}$, and is independent of $\vec{\overline{e}}^{(e)}$. Since we assumed $B^{(e)} < \frac{q}{41c^{(e)}}$, we can use Lemma 6 to derive that if $\|\vec{\overline{e}}^{(e)}\| \geq B^{(e)}$, then the probability that $\|\vec{\overline{z}}^{(e)}\| \leq \frac{1}{2} B^{(e)} \sqrt{26}$ is less than $2^{-128}$. Hence the probability that $\|\vec{\overline{e}}^{(e)}\| \leq B^{(e)}$ is at least $1 - 2^{-128}$.

4) implies that $\overline{\boldsymbol{x}}'$, which is defined as previous but with the extracted messages, satisfies $\langle \overline{\boldsymbol{x}}', \overline{\boldsymbol{x}}' - \mathbf{1} \rangle = 0 \mod q$. This also holds over the integers, since we assumed that $(B^{(e)})^2 + \sqrt{(v_e + k_{\mathsf{bin}})N} B^{(e)} < q$. Lemma 18 then implies that $\overline{\boldsymbol{x}}'$ is binary. Hence the opening satisfies Equation (7.5).

7) implies that $\langle \vec{p}_i, \vec{\overline{x}}_i \rangle = (\beta_i^{(e)})^2 - \|\vec{\overline{e}}^{(e)}\|^2 \mod q$. This also holds over the integers, since we assumed that $2(\max_{i \in [v_e]} \beta_i^{(e)})^2 + (B^{(e)})^2 - 1 < q$. This then implies that $(\beta_i^{(e)})^2 - \|\vec{\overline{e}}^{(e)}\|^2$ is a positive integer. Hence the opening also satisfies Equation (7.4).

So, either $\mathcal{E}$ will break the hiding property of the commitment, or it finds a valid opening to messages satisfying Equations (7.1) to (7.5) with probability $P_{\mathcal{E}'}(1 - 2^{-256})(1 - 2^{-128})$ in time $2T_{\mathcal{E}'}$. □

### 7.3.1 Example instantiation

We now have a protocol that can be used to prove various lattice statement. We will give the overview of one simple example. Suppose that we want to prove knowledge of a $\mathsf{MSIS}$ secret with error, specifically that we want to prove knowledge of $(\boldsymbol{s}, \boldsymbol{e}) \in R_q^{m+n}$ such that $\|(\boldsymbol{s}, \boldsymbol{e})\| \leq B$ and

$$\boldsymbol{A}\boldsymbol{s} + \boldsymbol{e} = \boldsymbol{u} \text{ over } R_q,$$

for public $\boldsymbol{A} \in R_q^{n \times m}$ and $\boldsymbol{u} \in R_q^n$. We can then commit to $\boldsymbol{s}$ in the Ajtai part, and prove with $\Pi$ that

$$\left\| \begin{bmatrix} \boldsymbol{s} \\ \boldsymbol{A}\boldsymbol{s} - \boldsymbol{u} \end{bmatrix} \right\| = \left\| \begin{bmatrix} \boldsymbol{I}_m \\ \boldsymbol{A} \end{bmatrix} \boldsymbol{s} - \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u} \end{bmatrix} \right\| \leq B.$$

This is achieved by setting public parameters $\boldsymbol{E}_1 = \begin{bmatrix} \boldsymbol{I}_m \\ \boldsymbol{A} \end{bmatrix}$, $\boldsymbol{v}_1 = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u} \end{bmatrix}$ and $\beta_1 = B$, and running $\Pi$ to only prove (7.4). This proof will be approximately 2.5 times smaller than in the previous works [16].

# Bibliography


[1] Miklós Ajtai. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 99–108, 1996.

[2] Thomas Attema, Ronald Cramer, and Lisa Kohl. A compressed $\sigma$-protocol theory for lattices. In *Advances in Cryptology–CRYPTO 2021: 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16–20, 2021, Proceedings, Part II*, pages 549–579. Springer, 2021.

[3] Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler. Practical product proofs for lattice commitments. In *Annual International Cryptology Conference*, pages 470–499. Springer, 2020.

[4] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. More efficient commitments from structured lattice assumptions. In *International Conference on Security and Cryptography for Networks*, pages 368–385. Springer, 2018.

[5] Manuel Blum. Coin flipping by telephone a protocol for solving impossible problems. *ACM SIGACT News*, 15(1):23–27, 1983.

[6] Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. Algebraic techniques for short (er) exact lattice-based zero-knowledge proofs. In *Annual International Cryptology Conference*, pages 176–202. Springer, 2019.

[7] Ivan Damgård. On $\sigma$-protocols. *Lecture Notes, University of Aarhus, Department for Computer Science*, page 84, 2002.

[8] Ivan Damgård and Jesper Nielsen. Commitment schemes and zero-knowledge protocols (2011), 2008.

[9] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In *Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 40–56. Springer, 2013.

[10] Muhammed F Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. Practical exact proofs from lattices: New techniques to exploit fully-splitting rings. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 259–288. Springer, 2020.

[11] Shafi Goldwasser, Silvio Micali, and Chales Rackoff. The knowledge complexity of interactive proof-systems. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 203–225. 2019.

[12] Thijs Laarhoven, Joop van de Pol, and Benne de Weger. Solving hard lattice problems and the security of lattice-based cryptosystems. *Cryptology ePrint Archive*, 2012.

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

[14] Vadim Lyubashevsky. Fiat-shamir with aborts: Applications to lattice and factoring-based signatures. In *Advances in Cryptology–ASIACRYPT 2009: 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings 15*, pages 598–616. Springer, 2009.

[15] Vadim Lyubashevsky. Lattice signatures without trapdoors. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 738–755. Springer, 2012.

[16] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plancon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. *Cryptology ePrint Archive*, 2022.

[17] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. In *IACR International Conference on Public-Key Cryptography*, pages 215–241. Springer, 2021.

[18] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 1–23. Springer, 2010.

[19] Vadim Lyubashevsky and Gregor Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 204–224. Springer, 2018.

[20] Ueli Maurer. Unifying zero-knowledge proofs of knowledge. In *International Conference on Cryptology in Africa*, pages 272–286. Springer, 2009.

[21] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

[22] Ngoc Khanh Nguyen. *Lattice-Based Zero-Knowledge Proofs Under a Few Dozen Kilobytes*. PhD thesis, ETH Zurich, 2022.

[23] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):1–40, 2009.

[24] Oded Regev. The learning with errors problem. *Invited survey in CCC*, 7(30):11, 2010.

[25] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[26] Jacques Stern. A new identification scheme based on syndrome decoding. In *Annual International Cryptology Conference*, pages 13–21. Springer, 1993.

[27] Tiril Stjernberg. Verifiable encryption from lattices. Specialization project in TMA4500, Department of Mathematical Sciences, NTNU – Norwegian University of Science and Technology, Dec. 2022.

[28] 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 *Annual International Cryptology Conference*, pages 147–175. Springer, 2019.

# Appendices

# Appendix A

# Security analysis of $\Pi_{\mathsf{NTT}}$

**Theorem 26.** Let $\mathfrak{s} = \gamma\sqrt{6N}$ for some $\gamma > 0$. Then the protocol $\Pi_{\mathsf{NTT}}$ is complete, meaning that the honest prover $\mathcal{P}$ convinces the honest verifier $\mathcal{V}$ with probability

$$\approx \frac{1}{\exp(14/\gamma + 1/(2\gamma^2))}.$$

*Proof.* According to Lemma 4, the probability that $\mathcal{P}$ does not abort is at least

$$\frac{1}{\exp(14/\gamma + 1/(2\gamma^2))}.$$

The tail-bound in Lemma 3 with $\delta = \sqrt{2}$ implies that, $\|\boldsymbol{z}\| \leq \mathfrak{s}\sqrt{12N}$, except with negligible probability $\sqrt{2}^{6N}\exp(-3N)2^{-128}$, since $\boldsymbol{z}$ has statistical distance at most $2^{-128}$ from $D_{R^6}$. For an honest prover, the rest of the verification equations follow from construction. Hence the theorem holds. □

**Theorem 27.** Let $\mathfrak{s} = \gamma\sqrt{6N}$ for some $\gamma > 0$. Then the protocol $\Pi_{\mathsf{NTT}}$ is zero-knowledge, meaning that there exists a simulator $\mathcal{S}$, that without access to secret information outputs a simulation of a non-aborting transcript of the protocol, such that for every algorithm $\mathcal{A}$ that has advantage $\varepsilon$ in distinguishing the simulated transcript from the actual transcript, there is an algorithm $\mathcal{A}'$ with the same running time that has advantage $\varepsilon - 2^{-128}$ in solving the $\mathsf{MLWE}_{1,5,\chi}$ problem.

*Proof.* We start by noting that $z$ looks uniformly random in $R_q$ and that $\boldsymbol{z}'$ is within statistical distance of $2^{-128}$ from $D_{R^6,\mathfrak{s}}$, according to Lemma 4. Thus the simulator can simply draw $z \xleftarrow{\$} R_q$ and $\boldsymbol{z}' \xleftarrow{\$} D_{R^6,\mathfrak{s}}$. We also know that since $y$ was drawn uniformly at random and from the rejection sampling that the challenges $c$ and $f$ are independent of $z$ and $\boldsymbol{z}'$, respectively, and so these can also just be drawn at random, $c \xleftarrow{\$} \mathbb{Z}_q$, $f \xleftarrow{\$} \mathcal{C}$.

Since the commitment $\boldsymbol{t}$ is computationally indistinguishable from a dummy commitment if the $\mathsf{MLWE}_{1,5,\chi}$ problem is hard, by the hiding property, the simulator can also just draw a uniformly

random $\boldsymbol{t} \xleftarrow{\$} R_q^5$. The remaining messages can now be computed as

$$\begin{aligned}
\vec{w} &= A\hat{z} - c\vec{u} \\
\boldsymbol{w}' &= \boldsymbol{b}_0 \boldsymbol{z}' - f\boldsymbol{t}_0 \\
\boldsymbol{x}_1 &= (\boldsymbol{b}_1 + c\boldsymbol{b}_2)\boldsymbol{z}' + fz - f(\boldsymbol{t}_1 + c\boldsymbol{t}_2) \\
\boldsymbol{x}_2 &= ((z-c)(z-2c)\boldsymbol{b}_2 - z\boldsymbol{b}_3 + \boldsymbol{b}_4)\boldsymbol{z}' - f((z-c)(z-2c)\boldsymbol{t}_2 - z\boldsymbol{t}_3 + \boldsymbol{t}_4)
\end{aligned}$$

Now all the verification equations will hold, and hence the simulated transcript has a statistical distance of at most $2^{-128}$ from the honest one. Since the transcripts only differ in that $\boldsymbol{t}$ is distributed differently, any algorithm $\mathcal{A}$ that has advantage $\varepsilon$ in distinguishing these transcripts, must have advantage $\varepsilon - 2^{-128}$ in solving the $\mathsf{RLWE}_5$ problem. □

**Theorem 28.** The protocol $\Pi_{\mathsf{NTT}}$ is knowledge sound, meaning that there is an extractor $\mathcal{E}$ with the following properties. When given rewindable access to a deterministic prover $\mathcal{P}^*$ that convinces $\mathcal{V}$ with probability $\varepsilon > 2/q + 1/N$, $\mathcal{E}$ either outputs a solution $\vec{s}^* \in \{0,1,2\}^N$ to $A\vec{s}^* = \vec{u}$, or a $\mathsf{MSIS}_{1,6,8B}$ solution for $\boldsymbol{b}_0$ in expected time at most $144/(\varepsilon - 2/q - 1/N)$ when running $\mathcal{P}^*$ once is assumed to take unit time.

*Proof.* We construct the extractor $\mathcal{E}$ by letting it run $\mathcal{P}^*$ until it obtains six accepting transcripts, for three different first challenges $c_1, c_2$ and $c_3$ such that there are two valid transcripts for different second challenges $f_{i,1} \neq f_{i,2}$, for each of the $c_i$'s. The expected time it takes in order to obtain the first accepting transcript with challenges $c_1$ and $f_{1,1}$ is clearly $1/\varepsilon$. We can now use the heavy rows argument, to see that conditioned on the first challenge $c_1$, $\mathcal{E}$ will with probability $1/2$ be able obtain a valid transcript for a uniformly random second challenge $f_{1,2}$ with probability $\varepsilon/2$. In this case $\mathcal{E}$ obtains the second accepting transcript for first challenge $c_1$ and second challenge $f_{1,2} \neq f_{1,1}$ with probability at least $\varepsilon/2 - 1/(2N)$, since the challenge space for the second challenges is of size $2N$. This has expected time $(\varepsilon/2 - 1/(2N))^{-1}$.

For the third transcript with challenges $c_2 \neq c_1$ and $f_{2,1}$, the extractor now succeeds with probability at least $\varepsilon - 1/q$, and hence in expected time at most $(\varepsilon - 1/q)^{-1}$, since the size of the first challenge space is $q$. Again by the heavy rows argument, with probability $1/2$, $\mathcal{E}$ obtains another valid transcript conditioned on $c_2$ with a uniformly random second challenge with probability $\varepsilon/2 - 1/(2q)$. In this case $\mathcal{E}$ obtains the fourth accepting transcript for $c_2$ and $f_{2,2} \neq f_{2,1}$ in conditioned expected time at most $(\varepsilon/2 - 1/(2q) - 1/(2N))^{-1}$.

For the fifth transcript with first challenge $c_3 \neq c_2$ and uniformly random $f_{3,1}$, $\mathcal{E}$ succeeds in expected time at most $(\varepsilon - 2/q)^{-1}$. For the sixth transcript with $f_{3,2} \neq f_{3,1}$, $\mathcal{E}$ with probability $1/2$ succeeds in conditioned expected time at most $(\varepsilon/2 - 1/q - 1/(2N))^{-1}$.

Thus we can say that with probability $1/2$, $\mathcal{E}$ is able to obtain two valid transcripts for each of the $c_i$'s in expected time at most

$$\frac{1}{\varepsilon - 2/q} + \frac{1}{\varepsilon/2 - 1/(2q) - 1/(2N)} \leq \frac{3}{\varepsilon - 2/q - 1/N}.$$

So in total, with probability $1/8$, $\mathcal{E}$ obtains the six accepting transcripts in expected time at most

$$T = \frac{9}{\varepsilon - 2/q - 1/N}.$$

We limit the run time of $\mathcal{E}$ to $2T$, so that if the extractor does not obtain these six transcripts within reasonable time, which happens with probability 7/8, it will just terminate. With this new condition, we can use Markov's inequality to get that the extractor now obtains the six valid transcripts in expected time at most $2T$ with probability 1/16. So in total, when we account for restarting in case of failure, the extractor will obtain the valid transcripts in expected time at most $16T$.

We now denote the last messages by $\mathcal{P}^*$ in the accepting protocols by $\boldsymbol{z}'_{i,j}$ for $i = 1, 2, 3$ and $j = 1, 2$. For the pairs of transcripts with the same first challenge, we define the differences $\boldsymbol{z}'_i = \boldsymbol{z}'_{i,1} - \boldsymbol{z}'_{i,2}$ and $\overline{f}_i = f_{i,1} - f_{i,2}$. From the verification equations we get that $\boldsymbol{b}_0\boldsymbol{z}'_{i,j} = \boldsymbol{w}' + f_{i,j}\boldsymbol{t}_0$, which implies that

$$\boldsymbol{b}_0\boldsymbol{z}'_i = \overline{f}_i\boldsymbol{t}_0.$$

For each $i = 1, 2, 3$ we can now define the openings of $\boldsymbol{t}$ as

$$m_k = \boldsymbol{t}_k - \boldsymbol{b}_k\frac{\boldsymbol{z}'_i}{\overline{f}_i}, \text{ for } k = 1, \ldots, 4$$

All of these are valid relaxed openings, and we have that $\|\boldsymbol{z}'_i\| \leq 2B$, since the $\boldsymbol{z}'_{i,j}$ are such that $\|\boldsymbol{z}'_{i,j}\| \leq B$. If we now were to obtain different openings $m_k \neq m'_k$ for two different $i$, we would get, for instance, that

$$\boldsymbol{t}_k - \boldsymbol{b}_k\frac{\boldsymbol{z}'_1}{\overline{f}_1} \neq \boldsymbol{t}_k - \boldsymbol{b}_k\frac{\boldsymbol{z}'_2}{\overline{f}_2}.$$

This implies that $\boldsymbol{b}_0(\overline{f}_2\boldsymbol{z}'_1 - \overline{f}_1\boldsymbol{z}'_2) = 0$, which since $\overline{f}_2\boldsymbol{z}'_1 - \overline{f}_1\boldsymbol{z}'_2 \neq 0$ and $\|\overline{f}_i\boldsymbol{z}'_i\| \leq 2\|\boldsymbol{z}'_i\| \leq 4B$, gives us an $\mathsf{MSIS}_{1,6,8B}$ solution.

So we assume that the openings are the same for $i = 1, 2, 3$, and denote by $z_i$ the message $z$ that is sent by the prover in the two transcripts for challenge $c_i$. The verification equations now implies that

$$(\boldsymbol{b}_1 + c_i\boldsymbol{b}_2)\boldsymbol{z}'_{i,j} + f_{i,j}z_i = \boldsymbol{x}_{1,i} + f_{i,j}(\boldsymbol{t}_1 + c_i\boldsymbol{t}_2).$$

By subtracting for $j = 2$ from $j = 1$ for the same $i$, we get that

$$(\boldsymbol{b}_1 + c_i\boldsymbol{b}_2)\frac{\boldsymbol{z}'_i}{\overline{f}_i} + z_i = \boldsymbol{t}_1 + c_i\boldsymbol{t}_2.$$

If we define the opening $m_1 = y^*$, $m_2 = s^*$, we get the following opening corresponding to $\boldsymbol{t}_1 + c_i\boldsymbol{t}_2$:

$$y^* + c_is^* = m_1 + c_im_2 = \boldsymbol{t}_1 + c_i\boldsymbol{t}_2 - (\boldsymbol{b}_1 + c_i\boldsymbol{b}_2)\frac{\boldsymbol{z}'_i}{\overline{f}_i}.$$

Hence we have that $z_i = y^* + c_is^*$, and the messages $z_i$ are of the expected form. From the verification equations we further also get that

$$((z_i - c_i)(z_i - 2c_i)\boldsymbol{b}_2 - z_i\boldsymbol{b}_3 + \boldsymbol{b}_4)\boldsymbol{z}'_{i,j} = \boldsymbol{x}_{2,i} + f_{i,j}((z_i - c_i)(z_i - 2c_i)\boldsymbol{t}_2 - z_i\boldsymbol{t}_3 + \boldsymbol{t}_4).$$

We can now use the opening $(z_i - c_i)(z_i - 2c_i)s^* - z_im_3 + m_4$ corresponding to $(z_i - c_i)(z_i - 2c_i)\boldsymbol{t}_2 - z_i\boldsymbol{t}_3 + \boldsymbol{t}_4$ and that $z_i = y^* + c_is^*$, to get the following:

$$\begin{aligned}
&(z_i - c_i)(z_i - 2c_i)s^* - z_im_3 + m_4 \\
&= (y^* + c_i(s^* - 1))(y^* + c_i(s^* - 2))s^* - y^*m_3 - c_is^*m_3 + m_4 \\
&= ((y^*)^2s^* - y^*m_3 + m_4) + (y^*(2s^* - 3) - m_3)s^*c_i + (s^* - 1)(s^* - 2)s^*c_i^2 \\
&= 0.
\end{aligned}$$

So we have a quadratic polynomial that is zero in $c_1, c_2$ and $c_3$, which can be expressed in the following matrix-equation over $R_q$:

$$\begin{bmatrix} 1 & c_1 & c_1^2 \\ 1 & c_2 & c_2^2 \\ 1 & c_3 & c_3^2 \end{bmatrix} \begin{bmatrix} ((y^*)^2 s^* - y^* m_3 + m_4) \\ (y^*(2s^* - 3) - m_3)s^* \\ (s^* - 1)(s^* - 2)s^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Since the difference of each of two of the challenges $c_1, c_2, c_3$ is invertible over $R_q$, the first matrix is invertible. Hence this implies that $(s^* - 1)(s^* - 2)s^* = 0$. We can take the $\mathsf{NTT}$ transformation of this equation, to get that

$$\hat{s}^* \circ (\hat{s}^* - \vec{1}) \circ (\hat{s}^* - \vec{2}) = \vec{0}$$

in $\mathbb{Z}_q^N$. This implies that the coefficients if $\hat{s}^*$ are in $\{0, 1, 2\}$.

Finally, we can use the second verification equation, $A\hat{z}_i = \vec{w} + c_i\vec{u}$, to obtain $A(\hat{z}_1 - \hat{z}_2) = (c_1 - c_2)\vec{u}$. But since

$$\frac{\hat{z}_1 - \hat{z}_2}{c_1 - c_2} = \hat{s}^*,$$

and $\hat{s}^*$ has coefficients in $\{0, 1, 2\}$, this is the desired solution to the linear equation with $A$. □

# Appendix B

# Security analysis of $\Pi^{\sigma}_{\mathsf{open}}$ and $\Pi^{\sigma}_{\mathsf{prod}}$

## B.1 Opening proof

**Theorem 29.** Suppose that $\mathfrak{s} = \gamma\kappa\sqrt{(\lambda+\mu+1)N}$ for some $\gamma > 0$. Then the protocol $\Pi^{\sigma}_{\mathsf{open}}$ is complete, meaning that the honest prover $\mathcal{P}$ convinces the honest verifier $\mathcal{V}$ with probability

$$\approx \frac{1}{\exp(14/\gamma + 1/(2\gamma^2))}.$$

*Proof.* According to Lemma 4, the probability that $\mathcal{P}$ does not abort is at least

$$\frac{1}{\exp(14/\gamma + 1/(2\gamma^2))},$$

and the $\boldsymbol{z}_i$ has statistical distance at most $2^{-128}$ from $D_{R^{\lambda+\mu+1},\mathfrak{s}}$. According to Lemma 3 with $\delta = \sqrt{2}$ we have $\|\boldsymbol{z}_i\| \leq B = \mathfrak{s}\sqrt{2(\lambda+\mu+1)N}$, except with negligible probability $\sqrt{2}^{(\lambda+\mu+1)N}\exp(-(\lambda+\mu+1)N/2)$. For an honest prover, the remaining verification equation holds by construction, and hence the theorem holds. □

**Theorem 30.** Suppose that $\mathfrak{s} = \gamma\kappa\sqrt{(\lambda+\mu+1)N}$ for some $\gamma > 0$. Then the protocol $\Pi^{\sigma}_{\mathsf{open}}$ is zero-knowledge, meaning that there exists a simulator $\mathcal{S}$, that without access to secret information outputs a simulation of a non-aborting transcript of the protocol which has statistical distance at most $2^{-128}$ to the actual transcript.

*Proof.* By Lemma 4 the $\boldsymbol{z}_i$ are within a statistical distance of $2^{-128}$ from $D_{R^{\lambda+\mu+1},\mathfrak{s}}$ in non-aborting honest transcripts. Hence the simulator can just sample $\boldsymbol{z}_i \xleftarrow{\$} D_{R^{\lambda+\mu+1},\mathfrak{s}}$. Also from the rejection sampling, we get that the challenge $c$ is independent of the $\boldsymbol{z}_i$, and so the simulator can simply draw a random $c \xleftarrow{\$} C$. We can now define $\boldsymbol{w}_i = \boldsymbol{B}_0\boldsymbol{z}_i - \sigma^i(c)\boldsymbol{t}_0$, and thus the verification equations will hold. Hence the simulated transcript has statistical distance at most $2^{-128}$ from the honest one. □

**Theorem 31.** Let $p$ be the maximum probability over $\mathbb{Z}_q$ of the coefficients of $c \mod X^{lN/k} - \zeta^l$ as in Lemma 14. Then the protocol $\Pi^{\sigma}_{\mathsf{open}}$ is knowledge sound, meaning that there is an extractor $\mathcal{E}$ with the following properties. When given rewindable access to a deterministic prover $\mathcal{P}^*$ that convinces $\mathcal{V}$ with probability $\varepsilon > p^{lN/k}$, $\mathcal{E}$ either outputs a weak opening for the commitment $\boldsymbol{t}$ or an $\mathsf{MSIS}_{\mu,\lambda+\mu+1,8\kappa B}$ solution for $\boldsymbol{B}_0$ in expected time at most $1/\varepsilon + (k/l)(\varepsilon - p^{Nl/k})^{-1}$ when running

$\mathcal{P}^*$ is assumed to take one unit time.

Moreover, the weak opening can be extended to also include $k$ vectors $(\boldsymbol{y}_{\mathsf{e}})_i \in R_q^{\lambda+\mu+1}$ such that $\boldsymbol{B}_0(\boldsymbol{y}_{\mathsf{e}})_i = \boldsymbol{w}_i$, where $\boldsymbol{w}_i$ are the prover commitments sent by $\mathcal{P}^*$ in the first round. Furthermore, for every accepting transcript of an interaction with $\mathcal{P}^*$, the prover replies are given by $\boldsymbol{z}_i = (\boldsymbol{y}_{\mathsf{e}})_i + \sigma^i(c)\boldsymbol{r}_{\mathsf{e}}$.

*Proof.* We construct the extractor $\mathcal{E}$ by letting it repeatedly run $\mathcal{P}^*$ with freshly sampled challenges until it obtains an accepting transcript, $(\boldsymbol{w}_i, c, \boldsymbol{z}_i)$. Next, for each $j \in \mathbb{Z}_{2k/l}^{\times}$, $\mathcal{E}$ rewinds the prover to just after the first round and sends a random challenge that differ from $c \mod (X^{lN/k} - \zeta^{jl})$, until it obtains an accepting transcript $(\boldsymbol{w}_i, c_j, \boldsymbol{z}_{ij})$. In this manner, $\mathcal{E}$ obtains $k/l$ more accepting transcripts such that for each of the $k/l$ ideals $(X^{lN/k} - \zeta^{jl})$, there is a transcript whose challenge differ modulo $(X^{lN/k} - \zeta^{jl})$. We can now write $\overline{c}_j = c - c_j$, and by construction $\overline{c}_j \mod (X^{lN/k} - \zeta^{jl}) \neq 0$.

We can now fix $e \in \{0, \ldots, l-1\}$ and $f \in \mathbb{Z}_{2k/l}^{\times}$, and focus on the prime ideal $\mathfrak{s}_{ef} = \sigma^e(X^{N/k} - \zeta^f)$, that is a divisor of $(X^{lN/k} - \zeta^{fl})$. We know that there must exist an $e' \in \{0, \ldots, l-1\}$ such that $\sigma^{e'}(\overline{c}_f) \mod \mathfrak{p}_{ef} \neq 0$. Hence we can define

$$(\boldsymbol{r}_{ef})_{\mathsf{e}} = \frac{\boldsymbol{z}_{e'} - \boldsymbol{z}_{e'f}}{\sigma^{e'}(\overline{c}_f)} \mod \mathfrak{p}_{ef}.$$

Next, we define $\boldsymbol{r}_{\mathsf{e}} \in R_q^{\lambda+\mu+1}$ to be the vector for which $\boldsymbol{r}_{\mathsf{e}} \equiv (\boldsymbol{r}_{ef})_{\mathsf{e}} \pmod{\mathfrak{p}_{ef}}$ for all $e \in \{0, \ldots, l-1\}$ and $f \in \mathbb{Z}_{2k/l}^{\times}$. We now show that either $\sigma^i(\overline{c}_j)\boldsymbol{r}_{\mathsf{e}} = \boldsymbol{z}_i - \boldsymbol{z}_{ij}$ for all $i \in \{0, \ldots, l-1\}$ and $j \in \mathbb{Z}_{2k/l}^{\times}$, or we find a $\mathsf{MSIS}$ solution for $\boldsymbol{B}_0$. From the verification equations we have that

$$\boldsymbol{B}_0(\boldsymbol{z}_i - \boldsymbol{z}_{ij}) = \sigma^i(\overline{c}_j)\boldsymbol{t}_0 \tag{B.1}$$

for all $i \in \{0, \ldots, l-1\}$ and $j \in \mathbb{Z}_{2k/l}^{\times}$. This implies that we either have a $\mathsf{MSIS}_{\mu,\lambda+\mu+1,8\kappa B}$ solution for $\boldsymbol{B}_0$, or that

$$\sigma^{e'}(\overline{c}_f)(\boldsymbol{z}_i - \boldsymbol{z}_{ij}) = \sigma^i(\overline{c}_j)(\boldsymbol{z}_{e'} - \boldsymbol{z}_{e'f}).$$

We assume that the latter is true, for which we get that

$$\begin{aligned}
\sigma^i(\overline{c}_j)\boldsymbol{r}_{\mathsf{e}} &\equiv \sigma^i(\overline{c}_j)(\boldsymbol{r}_{ef})_{\mathsf{e}} \\
&\equiv \sigma^i(\overline{c}_j)\frac{\boldsymbol{z}_{e'} - \boldsymbol{z}_{e'f}}{\sigma^{e'}(\overline{c}_f)} \\
&\equiv \boldsymbol{z}_i - \boldsymbol{z}_{ij} \pmod{\mathfrak{p}_{ef}}.
\end{aligned}$$

Our claim now follows from the Chinese remainder theorem. We can plug this into (B.1), and get that for all $i \in \{0, \ldots, l-1\}$ and $j \in \mathbb{Z}_{2k/l}^{\times}$, it must hold that $\boldsymbol{B}_0\sigma^i(\overline{c}_j)\boldsymbol{r}_{\mathsf{e}} = \sigma^i(\overline{c}_j)\boldsymbol{t}_0$. This in turn implies that

$$\boldsymbol{B}_0\boldsymbol{r}_{\mathsf{e}} = \boldsymbol{t}_0.$$

We can now define the extracted message $\boldsymbol{m}_{\mathsf{e}}$ to be such that

$$\boldsymbol{t}_1 = \langle \boldsymbol{b}_1, \boldsymbol{r}_{\mathsf{e}} \rangle + \boldsymbol{m}_{\mathsf{e}}.$$

Hence the extractor has obtained a weak opening $(\sigma^i(\overline{c}_j), \boldsymbol{r}_{\mathsf{e}}, \boldsymbol{m}_{\mathsf{e}})$ for the commitment $\boldsymbol{t}$, for which $\|\sigma^i(\overline{c}_j)\boldsymbol{r}_{\mathsf{e}}\| \leq 2B$.

We now investigate the run time for the extractor. Obtaining the first transcript is expected to take time $1/\varepsilon$. When we put our restriction on the challenges, the success probability is reduced to at least $\varepsilon - p^{lN/k}$. Thus the total expected time for the extractor to obtain the $1 + k/l$ accepting transcripts is at most

$$\frac{1}{\varepsilon} + \frac{k}{l}\frac{1}{\varepsilon - p^{lN/k}}.$$

We now consider the $(\boldsymbol{y}_{\mathsf{e}})_i$, and set them to be the vectors defined by

$$\boldsymbol{z}_i = (\boldsymbol{y}_i)_{\mathsf{e}} + \sigma^i(c)\boldsymbol{r}_{\mathsf{e}}.$$

Clearly it must hold that $\boldsymbol{B}_0(\boldsymbol{y}_{\mathsf{e}})_i = \boldsymbol{B}_0(\boldsymbol{z}_i - \sigma^i(c)\boldsymbol{r}_{\mathsf{e}}) = \boldsymbol{w}_i$. Suppose now that there is another accepting transcript with the same $\boldsymbol{w}_i$, $(\boldsymbol{w}_i, c', \boldsymbol{z}_i')$, and write $\boldsymbol{z}_i' = (\boldsymbol{y}_{\mathsf{e}}')_i + \sigma^i(c')\boldsymbol{r}_{\mathsf{e}}$. From the verification equations for $\boldsymbol{z}_i$ and $\boldsymbol{z}_i'$ we get

$$\boldsymbol{B}_0(\boldsymbol{z}_i - \boldsymbol{z}_i') = \sigma^i(\overline{c})\boldsymbol{t}_0$$

for all $i \in \{0, \ldots, l-1\}$, where $\overline{c} = c - c'$. Using the same argumentation as for (B.1), we get that we either have a $\mathsf{MSIS}$ solution or that

$$\sigma^{e'}(\overline{c}_f)(\boldsymbol{z}_i - \boldsymbol{z}_i') = \sigma^i(\overline{c})(\boldsymbol{z}_{e'} - \boldsymbol{z}_{e'f}).$$

Since $\boldsymbol{z}_{e'} - \boldsymbol{z}_{e'f} = \sigma^{e'}(\overline{c}_f)\boldsymbol{r}_{\mathsf{e}}$, this implies that

$$\sigma^{e'}(\overline{c}_f)((\boldsymbol{y}_{\mathsf{e}})_i - (\boldsymbol{y}_{\mathsf{e}}')_i) = 0.$$

But since $\sigma^{e'}(\overline{c}_f) \not\equiv 0 \pmod{\mathfrak{p}_{ef}}$, this implies that $(\boldsymbol{y}_{\mathsf{e}})_i \equiv (\boldsymbol{y}_{\mathsf{e}}')_i \pmod{\mathfrak{p}_{ef}}$. Thus $(\boldsymbol{y}_{\mathsf{e}})_i = (\boldsymbol{y}_{\mathsf{e}}')_i$, and hence the theorem holds. □

## B.2 Product proof

**Theorem 32.** Suppose that $\mathfrak{s} = \gamma\kappa\beta\sqrt{(\lambda+\mu+4)N}$ for some $\gamma > 0$. Then the protocol $\Pi^{\sigma}_{\mathsf{prod}}$ is complete, meaning that the honest prover $\mathcal{P}$ convinces the honest verifier $\mathcal{V}$ with with probability

$$\approx \frac{1}{\exp(14/\gamma + 1/(2\gamma^2))}.$$

*Proof.* According to Lemma 4, the probability that $\mathcal{P}$ does not abort is at least

$$\frac{1}{\exp(14/\gamma + 1/(2\gamma^2))},$$

and the $\boldsymbol{z}_i$ has statistical distance at most $2^{-128}$ from $D_{R^{\lambda+\mu+4},\mathfrak{s}}$. According to Lemma 3 with $\delta = \sqrt{2}$ we have $\|\boldsymbol{z}_i\| \leq \beta = \mathfrak{s}\sqrt{2(\lambda+\mu+4)N}$, except with negligible probability $\sqrt{2}^{(\lambda+\mu+4)N}\exp(-(\lambda+\mu+4)N/2)$. For an honest prover, the remaining verification equation holds by construction, and hence the theorem holds. □

**Theorem 33.** Suppose that $\mathfrak{s} = \gamma\kappa\beta\sqrt{(\lambda+\mu+4)N}$ for some $\gamma > 0$. Then the protocol $\Pi^{\sigma}_{\mathsf{prod}}$ is zero-knowledge, meaning that there exists a simulator $\mathcal{S}$, that without access to secret information outputs a simulation of a non-aborting transcript of the protocol, such that for every algorithm $\mathcal{A}$ that has advantage $\varepsilon$ in distinguishing the simulated transcript from the actual transcript, there is an algorithm $\mathcal{A}'$ with the same running time that has advantage $\varepsilon - 2^{-128}$ in solving the $\mathsf{MLWE}_{\lambda+\mu+4,\mu+4,\chi}$ problem.

*Proof.* By Lemma 4, the $\boldsymbol{z}_i$ are within a statistically distance of $2^{-128}$ to $D_{R^{\lambda+\mu+4},\mathfrak{s}}$ in non-aborting transcripts. Hence the simulator can just sample $\boldsymbol{z}_i \xleftarrow{\$} D_{R^{\lambda+\mu+4},\mathfrak{s}}$. Also from the rejection sampling, we get that the challenge $c$ is independent of the $\boldsymbol{z}_i$, and so the simulator can simply draw a random $c \xleftarrow{\$} C$.

By the hiding property of the commitment scheme, the commitments are indistinguishable from truly random if the $\mathsf{MLWE}_{\lambda+\mu+4,\mu+4,\chi}$ problem is hard. Hence the simulator can also just sample random $\boldsymbol{t}_0 \xleftarrow{\$} R_q^\mu$ and $\boldsymbol{t}_1, \boldsymbol{t}_2, \boldsymbol{t}_3, \boldsymbol{t}_4 \xleftarrow{\$} R_q$. The remaining messages can then be defined as

$$\begin{aligned}
\boldsymbol{w}_i &= \boldsymbol{B}_0 \boldsymbol{z}_i - \sigma^i(c)\boldsymbol{t}_0 \\
\boldsymbol{f}_4 &= \langle \boldsymbol{b}_4, \boldsymbol{z}_0 \rangle - c\boldsymbol{t}_4 \\
\boldsymbol{v} &= \sum_{i=0}^{k-1} \alpha_i \sigma^{-i}\Big(\boldsymbol{f}_1^{(i)}\boldsymbol{f}_2^{(i)} + \sigma^i(c)\boldsymbol{f}_3^{(i)}\Big) + \boldsymbol{f}_4
\end{aligned}$$

Thus the verification equations will hold. Hence an adversary that has has advantage $\varepsilon$ in distinguishing the simulated transcript from the real one, must have advantage $\varepsilon - 2^{-128}$ in distinguishing the $\mathsf{MLWE}_{\lambda+\mu+4,\mu+4,\chi}$ samples in the commitments from uniform. □

**Theorem 34.** Let $p$ be the maximum probability over $\mathbb{Z}_q$ of the coefficients of $c \mod X^{lN/k} - \zeta^l$ as in Lemma 14. Then the protocol $\Pi^\sigma_{\mathsf{prod}}$ is knowledge sound, meaning that there is an extractor $\mathcal{E}$ with the following properties. When given rewindable access to a deterministic prover $\mathcal{P}^*$ that convinces $\mathcal{V}$ with probability $\varepsilon > (3p^{N/k})^l$, $\mathcal{E}$ either outputs a weak opening for the commitment $\boldsymbol{t}$ with messages $(\boldsymbol{m}_1)_\mathsf{e}, (\boldsymbol{m}_2)_\mathsf{e}$ and $(\boldsymbol{m}_3)_\mathsf{e}$ such that $(\boldsymbol{m}_1)_\mathsf{e}(\boldsymbol{m}_2)_\mathsf{e} = (\boldsymbol{m}_3)_\mathsf{e}$, or a $\mathsf{MSIS}_{\mu,\lambda+\mu+4,8\kappa B}$ solution for $\boldsymbol{B}_0$ in expected time at most $1/\varepsilon + (k/l)(\varepsilon - p^{Nl/k})^{-1}$ when running $\mathcal{P}^*$ is assumed to take one unit time.

*Proof.* The extractor $\mathcal{E}$ is constructed by first letting it open the commitments $\boldsymbol{t}_1, \ldots, \boldsymbol{t}_4$. We can then use Theorem 31 to see that either $\mathcal{E}$ finds a $\mathsf{MSIS}_{\mu,\lambda+\mu+4,8\kappa B}$ solution for $\boldsymbol{B}_0$, or that it can compute vectors $\boldsymbol{y}_\mathsf{e}$ and $\boldsymbol{r}_\mathsf{e}$ such that for ever accepting transcript with first messages $\boldsymbol{t}$ and $\boldsymbol{w}_i$, we have that

$$\boldsymbol{z}_i = (\boldsymbol{y}_\mathsf{e})_i + \sigma^i(c)\boldsymbol{r}_\mathsf{e}.$$

From this we can then define the extracted messages $(m_1)_\mathsf{e}, \ldots, (m_4)_\mathsf{e}$ to satisfy

$$\begin{aligned}
\boldsymbol{t}_1 &= \langle \boldsymbol{b}_1, \boldsymbol{r}_\mathsf{e} \rangle + (m_1)_\mathsf{e} \\
\boldsymbol{t}_2 &= \langle \boldsymbol{b}_2, \boldsymbol{r} \rangle + (m_2)_\mathsf{e} \\
\boldsymbol{t}_3 &= \langle \boldsymbol{b}_3, \boldsymbol{r}_\mathsf{e} \rangle + (m_3)_\mathsf{e} \\
\boldsymbol{t}_4 &= \langle \boldsymbol{b}_4, \boldsymbol{r}_\mathsf{e} \rangle + (m_4)_\mathsf{e}
\end{aligned}$$

The first three messages will be independent of the challenges $\alpha_i$, since their corresponding commitment was sent before the challenges were chosen, but $(m_4)_\mathsf{e}$ can depend on the $\alpha_i$'s. We can now use the expression for $\boldsymbol{z}_i$ and input the commitments into the expressions for the $\boldsymbol{f}_j$'s to obtain

$$\begin{aligned}
\boldsymbol{f}_j^{(i)} &= \langle \boldsymbol{b}_j, (\boldsymbol{y}_\mathsf{e})_i \rangle - \sigma^i(c)(m_j)_\mathsf{e} \text{ for } j = 1, 2, 3 \\
\boldsymbol{f}_4 &= \langle \boldsymbol{b}_4, (\boldsymbol{y}_\mathsf{e})_0 \rangle - c(m_4)_\mathsf{e}
\end{aligned}$$

We can now substitute these expressions into the last verification and get that

$$\begin{aligned}
&\langle \boldsymbol{b}_4, (\boldsymbol{y}_{\mathsf{e}})_0 \rangle + \sum_{i=0}^{l-1} \alpha_i \sigma^{-i} \Big( \langle \boldsymbol{b}_1, (\boldsymbol{y}_{\mathsf{e}})_i \rangle \langle \boldsymbol{b}_2, (\boldsymbol{y}_{\mathsf{e}})_i \rangle \Big) \\
&+ c \sum_{i=0}^{l-1} \alpha_i \sigma^{-i} \Big( \langle \boldsymbol{b}_3, (\boldsymbol{y}_{\mathsf{e}})_i \rangle - (m_1)_{\mathsf{e}} \langle \boldsymbol{b}_2, (\boldsymbol{y}_{\mathsf{e}})_i \rangle - (m_2)_{\mathsf{e}} \langle \boldsymbol{b}_1, (\boldsymbol{y}_{\mathsf{e}})_i \rangle - (m_4)_{\mathsf{e}} \Big) \\
&+ c^2 \left( \sum_{i=0}^{l-1} \alpha_i \sigma^{-i} \big( (m_1)_{\mathsf{e}} (m_2)_{\mathsf{e}} - (m_3)_{\mathsf{e}} \big) \right) - \boldsymbol{v} = 0.
\end{aligned}$$

For any accepting transcripts it is important that this holds, and that this polynomial in $c$ has coefficients that are independent from $c$. As we recall, $(m_4)_{\mathsf{e}}$ and $\boldsymbol{v}$ are the only terms that can depend on the $\alpha_i$'s.

We now assume that $(m_1)_{\mathsf{e}}(m_2)_{\mathsf{e}} \neq (m_3)_{\mathsf{e}}$, and bound the success probability conditioned on this assumption. We must in this case have that $(m_1)_{\mathsf{e}}(m_2)_{\mathsf{e}} - (m_3)_{\mathsf{e}}$ is non-zero modulo at least one of the prime ideals:

$$(m_1)_{\mathsf{e}}(m_2)_{\mathsf{e}} - (m_3)_{\mathsf{e}} \not\equiv 0 \pmod{\sigma^i(X^{N/k} - \zeta^j)}$$

for some $i \in \{0, \ldots, l-1\}$ and $j \in \mathbb{Z}_{2k/l}^{\times}$. Then we have that the following polynomial

$$\boldsymbol{p} = \sum_{i=0}^{l-1} \alpha_i \sigma^{-i} ((m_1)_{\mathsf{e}}(m_2)_{\mathsf{e}} - (m_3)_{\mathsf{e}}) \mod (X^{lN/k} - \zeta^j l)$$

is uniformly random for uniformly random $\alpha_i$. The probability then that it is non-zero modulo all $l$ prime ideals that divide $(X^{lN/k} - \zeta^j l)$, is $(1 - \frac{1}{q^{N/k}})^l$. Also, modulo each prime ideal, there can be at most two points that make the evaluation of the previous verification polynomial zero. Hence there are only $2^l$ possible elements modulo $X^{lN/k} - \zeta^j l$. So, if we assume that the probability of $c$ mod $(X^{lN/k} - \zeta^j l)$ hitting an element is at most $p^{lN/k}$, the success probability of the prover must be bounded by $2^l p^{lN/k}$.

But if $\boldsymbol{p}$ is zero in one of the $l$ prime ideals, which has probability $\frac{l}{q^{N/k}}(1 - \frac{1}{q^{N/k}})^{l-1}$ of happening, then there are at most $2^{l-1} q^{N/k}$ possible values for $c \mod (X^{lN/k} - \zeta^j l)$. Hence the success probability in this case is bounded by $2^{l-1} q^{N/k} p^{lN/k}$. We can continue this argument for each of the cases, and we can see that the total success probability must be bounded by

$$\varepsilon \leq \sum_{i=0}^{l} \binom{l}{i} \Big( \frac{1}{q^{N/k}} \Big)^i \Big( 1 - \frac{1}{q^{N/k}} \Big)^{l-i} 2^{l-i} q^{iN/k} p^{lN/k} < \big( 3p^{N/k} \big)^l.$$

This is a contradiction to the bound in the theorem, and thus we must have that $(m_1)_{\mathsf{e}}(m_2)_{\mathsf{e}} = (m_3)_{\mathsf{e}}$. We notice that the running time for the extractor is the same as the running time for the extractor in Theorem 31, and hence the theorem holds. □

# Appendix C

# Security analysis of $\Pi_{\mathsf{inner}}$

**Theorem 35.** Suppose that $\mathfrak{s} = \gamma\kappa\beta\sqrt{(\lambda+\mu+2)N}$ for some $\gamma > 0$. Then the protocol $\Pi_{\mathsf{inner}}$ is complete, meaning that the honest prover $\mathcal{P}$ convinces the honest verifier $\mathcal{V}$ with with probability

$$\approx \frac{1}{\exp(14/\gamma + 1/(2\gamma^2))}.$$

*Proof.* According to Lemma 4, the probability that $\mathcal{P}$ does not abort is at least

$$\frac{1}{\exp(14/\gamma + 1/(2\gamma^2))},$$

and the $\boldsymbol{z}_i$ has statistical distance at most $2^{-128}$ from $D_{R^{\lambda+\mu+2},\mathfrak{s}}$. According to Lemma 3 with $\delta = \sqrt{2}$ we have $\|\boldsymbol{z}_i\| \leq \beta = \mathfrak{s}\sqrt{2(\lambda+\mu+2)N}$, except with negligible probability $\sqrt{2}^{(\lambda+\mu+2)N}\exp(-(\lambda+\mu+2)N/2)$. For an honest prover, the remaining verification equation holds by construction, and hence the theorem holds. □

**Theorem 36.** Suppose that $\mathfrak{s} = \gamma\kappa\beta\sqrt{(\lambda+\mu+2)N}$ for some $\gamma > 0$. Then the protocol $\Pi_{\mathsf{inner}}$ is zero-knowledge, meaning that there exists a simulator $\mathcal{S}$, that without access to secret information outputs a simulation of a non-aborting transcript of the protocol, such that for every algorithm $\mathcal{A}$ that has advantage $\varepsilon$ in distinguishing the simulated transcript from the actual transcript, there is an algorithm $\mathcal{A}'$ with the same running time that has advantage $\varepsilon - 2^{-128}$ in solving the $\mathsf{MLWE}_{\lambda+\mu+2,\mu+2,\chi}$ problem.

*Proof.* By Lemma 4, the $\boldsymbol{z}_i$ are within a statistically distance of $2^{-128}$ to $D_{R^{\lambda+\mu+2},\mathfrak{s}}$ in non-aborting transcripts. Hence the simulator can just sample $\boldsymbol{z}_i \xleftarrow{\$} D_{R^{\lambda+\mu+2},\mathfrak{s}}$. Also from the rejection sampling, we get that the challenge $c$ is independent of the $\boldsymbol{z}_i$, and so the simulator can simply draw a random $c \xleftarrow{\$} C$. Since the polynomial $h$ is such that $h_0 = \cdots = h_{\ell-1} = 0$ and the other coefficients are uniformly random, the simulator can also just sample $h \xleftarrow{\$} \{h \in R_q : h_0 = \cdots = h_{\ell-1} = 0\}$. The simulator also just draws the independently uniformly random challenges $\vec{\gamma}_\mu \xleftarrow{\$} \mathbb{Z}_q^m$.

In honest protocols, $\boldsymbol{r}$ is statistically independent of the $\boldsymbol{z}_i$'s, $c, h$, the $\vec{\gamma}_\mu$'s, $\check{s}$ and $g$, and thus the commitment $\boldsymbol{t}$ is indistinguishable from uniform if the $\mathsf{MLWE}_{\lambda+\mu+2,\mu+2,\chi}$ problem is hard, by the

hiding property of the commitment scheme. Hence the simulator can also simply sample $\boldsymbol{t} \xleftarrow{\$} R_q^{\mu+2}$. The remaining messages are now constructed as

$$\begin{aligned}
\boldsymbol{w}_i &= \boldsymbol{B}_0 \boldsymbol{z}_i - \sigma^i(c)\boldsymbol{t}_0, \\
\boldsymbol{v}_i &= \sum_{\mu=0}^{l-1} \frac{1}{l} X^\mu \sum_{\nu=0}^{l-1} \sigma^\nu \Big( \langle \mathsf{NTT}^{-1}(NA^T \vec{\gamma}_\mu)\boldsymbol{b}_1, \boldsymbol{z}_{i-\nu \mod l} \rangle \Big) + \langle \boldsymbol{b}_2, \boldsymbol{z}_i \rangle - \sigma^i(c)(\boldsymbol{\tau} + \boldsymbol{t}_2 - h),
\end{aligned}$$

so that the verification equations will hold. Now the simulated transcript is indistinguishable from the real one, in the sense that if there is an algorithm that can distinguish between these transcripts with advantage $\varepsilon$, then it has advantage $\varepsilon - 2^{-128}$ in distinguishing the $\mathsf{MLWE}_{\lambda+\mu+2,\mu+2,\chi}$ samples in $\boldsymbol{t}$ from uniform. □

**Theorem 37.** Let $p$ be the maximum probability over $\mathbb{Z}_q$ of the coefficients of $c \mod X^{lN/k} - \zeta^l$ as in Lemma 14. Then the protocol $\Pi_{\mathsf{inner}}$ is knowledge sound, meaning that there is an extractor $\mathcal{E}$ with the following properties. When given rewindable access to a deterministic prover $\mathcal{P}^*$ that sends the commitment $\boldsymbol{t}$ in the first round and convinces $\mathcal{V}$ with probability $\varepsilon > q^{-\ell} + p^\ell$, $\mathcal{E}$ either outputs a weak opening for the commitment $\boldsymbol{t}$ with message $s_{\mathsf{e}}$ such that $A\mathsf{NTT}(\check{s}_{\mathsf{e}}) = \vec{u}$, or an $\mathsf{MSIS}_{\mu,\lambda+\mu+2,8\kappa B}$ solution for $\boldsymbol{B}_0$ in expected time at most $1/\varepsilon + (N/l)(\varepsilon - p^l)^{-1}$ when running $\mathcal{P}^*$ is assumed to take one unit time.

*Proof.* The extractor starts by opening the commitments $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$. Since the $\Pi^\sigma_{\mathsf{open}}$ protocol was used for the commitment opening proof, we can use Theorem 31 to see that unless $\mathcal{E}$ finds a $\mathsf{MSIS}_{\mu,\lambda+\mu+2,8\kappa B}$ solution, it can compute $\boldsymbol{y}_{\mathsf{e}}$ and $\boldsymbol{r}_{\mathsf{e}}$ such that for every accepting transcript we have

$$\boldsymbol{z}_i = \boldsymbol{y}_{\mathsf{e}} + \sigma^i(c)\boldsymbol{r}_{\mathsf{e}}.$$

Next, we let $\check{s}_{\mathsf{e}} \in R_q$ and $g_{\mathsf{e}} \in R_q$ be the extracted messages defined by

$$\boldsymbol{t}_1 = \langle \boldsymbol{b}_1, \boldsymbol{r}_{\mathsf{e}} \rangle + \check{s}_{\mathsf{e}} \text{ and } \boldsymbol{t}_2 = \langle \boldsymbol{b}_2, \boldsymbol{r}_{\mathsf{e}} \rangle + g_{\mathsf{e}}.$$

If we substitute these expression into the commitment $\boldsymbol{\tau}$ to $f$, we get

$$\boldsymbol{\tau} = \sum_{\mu=0}^{l-1} \frac{1}{l} X^\mu \sum_{\nu=0}^{l-1} \sigma^\nu \Big( \langle \mathsf{NTT}^{-1}(NA^T \vec{\gamma}_\mu)\boldsymbol{b}_1, \boldsymbol{r} \rangle \Big) + f_{\mathsf{e}}$$

where

$$f_{\mathsf{e}} = \sum_{\mu=0}^{l-1} \frac{1}{l} X^\mu \sum_{\nu=0}^{l-1} \sigma^\nu (\mathsf{NTT}^{-1}(NA^T \vec{\gamma}_\mu)\check{s}_{\mathsf{e}} - \langle \vec{u}, \vec{\gamma}_\mu \rangle).$$

From construction we know that for all $\mu = 0, \ldots, l-1$ we have $(f_{\mathsf{e}})_\mu = \langle A\vec{s}_{\mathsf{e}} - \vec{u}, \vec{\gamma}_\mu \rangle$, where $\vec{s}_{\mathsf{e}} = \mathsf{NTT}(\check{s}_{\mathsf{e}})$. We also get from the last verification equation that

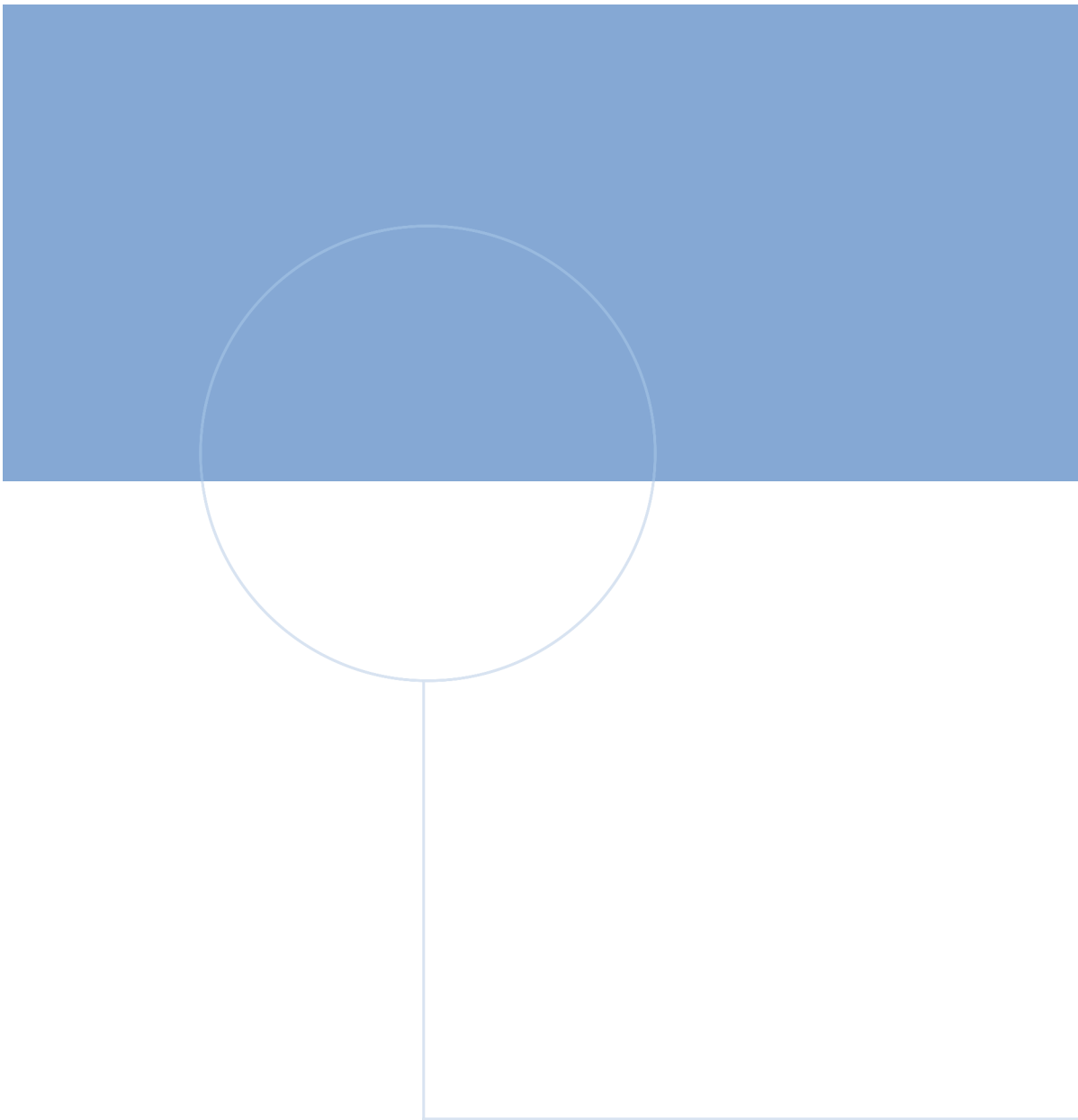
$$\sum_{\mu=0}^{l-1} \frac{1}{l} X^\mu \sum_{\nu=0}^{l-1} \sigma^\nu \Big( \langle \mathsf{NTT}^{-1}(NA^T \vec{\gamma}_\mu)\boldsymbol{b}_1, (\boldsymbol{y}_{\mathsf{e}})_{i-\nu \mod l} \rangle \Big) + \langle \boldsymbol{b}_2, \boldsymbol{y}_{\mathsf{e}} \rangle - \boldsymbol{v}_i = \sigma^i(c)(f_{\mathsf{e}} + g_{\mathsf{e}} - h) \quad \text{(C.1)}$$

for all $i = 0, \ldots, l-1$. All coefficients of these linear polynomials in $\sigma^i(c)$ are independent of $c$ in an accepting transcript.

We now investigate the success probability $\varepsilon$ of a prover, conditioned on $A\vec{s}_{\mathsf{e}} \neq \vec{u}$. In this case, for all $\mu = 0, \ldots, l-1$ we have that $(f_{\mathsf{e}})_\mu$ are uniformly random elements in $\mathbb{Z}_q$, and hence also $(f_{\mathsf{e}})_\mu + (g_{\mathsf{e}})_\mu$ is uniformly random. But since $h_\mu = 0$ for all $\mu = 0, \ldots, l-1$ in accepting transcripts, we can only get that $(f_{\mathsf{e}})_\mu + (g_{\mathsf{e}})_\mu - (h_{\mathsf{e}})_\mu = (f_{\mathsf{e}})_\mu + (g_{\mathsf{e}})_\mu \neq 0$ if there exists some $j \in \mathbb{Z}_{2N}^\times$ with $f_{\mathsf{e}} + g_{\mathsf{e}} - h \mod (X - \zeta^j) \neq 0$. This means the in order for (C.1) to hold for all $i$, there is only one possible value modulo $(X^l - \zeta^{jl})$ for the challenge $c$. And since we bounded the probability of each coefficient of $c \mod (X^l - \zeta^{jl})$ by $p$, we thus get

$$\begin{aligned}\varepsilon = \Pr[\text{accepting}] &< \left(\frac{1}{q}\right)^l + \Pr[\text{accepting } |(f_{\mathsf{e}})_\mu + (g_{\mathsf{e}})_\mu \neq 0 \text{ for some } \mu] \\ &\leq \left(\frac{1}{q}\right)^l + p^l.\end{aligned}$$

This is a contradiction to the bound in the theorem, and hence it must hold that $A\vec{s}_{\mathsf{e}} = \vec{u}$. □



NTNU

Norwegian University of
Science and Technology