Master's Thesis

Zero Knowledge Proofs in Hybrid Environments

Rittwik Hajra

ZERO KNOWLEDGE PROOFS IN HYBRID ENVIRONMENTS

Thesis submitted to the
Indian Statistical Institute, Kolkata
for partial fulfillment of requirements for the degree

of

Master of Technology in Cryptology & Security

by

Rittwik Hajra
Roll No. CrS2317

Under the guidance of

Dr. Sri Aravindakrishnan Thyagarajan
Lecturer
School of Computer Science
University of Sydney

Dr. Sabyasachi Karati
Assistant Professor
Cryptology & Security Research Unit (CSRU)
Indian Statistical Institute, Kolkata

INDIAN STATISTICAL INSTITUTE, KOLKATA

UNIVERSITY OF SYDNEY

11-th July, 2025

Zero Knowledge Proofs in Hybrid Environments

Rittwik Hajra
Sudipa Mandal
Dr. Sri Aravindakrishnan Thyagarajan

July, 2025

CERTIFICATE

This is to certify that the thesis entitled "**Zero Knowledge Proof in Hybrid Environments**", submitted by **Rittwik Hajra (CrS2317)** to **Indian Statistical Institute, Kolkata**, is a record of bonafide research work under my supervision and guidance, and I consider it worthy of consideration for the award of the degree of *Masters of Technology in Cryptology and Security* of the Institute.

Date: 11/07/25
Place: Sydney, Australia

Dr. Sri AravindaKrishnan Thyagarajan
Lecturer
School of Computer Science
University of Sydney

Date: 11/07/25
Place: Kolkata, India

Dr. Sabyasachi Karati
Assistant Professor
Cryptology and Security Research Unit
R. C. Bose Centre for Cryptology and Security
Indian Statistical Institute, Kolkata

DECLARATION

I hereby declare that the project entitled "**Zero Knowledge Proofs in Hybrid Environments**" submitted in partial fulfillment for the award of the degree of *Master of Technology in Cryptology and Security* completed under the supervision of Dr. Sri AravindaKrishnan Thyagarajan and Dr. Sabyasachi Karati, at Indian Statistical Institute is an authentic work. Further, I declare that I have not submitted this work for the award of any other degree elsewhere.

Date: 11/07/25
Place: Kolkata

Rittwik Hajra
Roll No.: CrS2317

Acknowledgement

I extend my heartfelt gratitude to my primary supervisor, Dr. Sri Aravindakrishnan Thyagarajan, whose exceptional guidance, unwavering encouragement, and intellectual generosity have been pivotal throughout the course of this research. His profound insights and thoughtful critiques have not only refined the core of this thesis but also profoundly influenced my academic growth.

I am equally indebted to my co-supervisor, Dr. Sabyasachi Karati, for their indispensable support, constructive feedback, and for fostering a research environment rich in academic rigor and openness. Their depth of knowledge and critical engagement significantly elevated the quality and direction of this work.

I would also like to express my sincere appreciation to Sudipa Mandal for her collaborative spirit, stimulating discussions, and sustained support during the course of this research. Her expertise and commitment have made a meaningful impact on the development and execution of this work. I am deeply grateful to my friends Tanushri, Arani and Subhradip for the countless moments of laughter, warmth, and camaraderie that made this journey all the more memorable. Their unwavering presence and shared joy brought light even to the most demanding days. I would like to express my sincere gratitude to my seniors, Soumit da and Subha da, for their unwavering support, patient guidance, and the wealth of knowledge they have generously shared with me throughout this journey. Their constant encouragement and insightful discussions have been instrumental in navigating the academic challenges I encountered, and their presence has been both reassuring and inspiring.

I am especially and profoundly indebted to Dr. Pratyay Mukherjee, whose depth of intellect, clarity of thought, and steady encouragement have profoundly shaped my academic journey. His insightful perspectives and generous guidance have not only illuminated complex ideas but also instilled in me a deeper appreciation for the rigor and elegance of research. His quiet yet impactful support has been a source of strength and inspiration throughout the course of this work, and for that, I remain deeply grateful.

Date: July 11, 2025

Place: Kolkata, India

Rittwik Hajra

Roll No.: CrS2317

Abstract

The impending advent of quantum computing poses a significant threat to classical cryptographic primitives, necessitating a robust migration toward post-quantum cryptographic (PQC) systems. However, a complete transition remains impractical in the short term, giving rise to hybrid environments where classical and PQC schemes coexist. This thesis addresses a fundamental challenge in such settings: the need for efficient and secure zero-knowledge proofs (ZKPs) that establish plaintext consistency across cryptographic primitives defined over distinct algebraic domains.

We present novel zero-knowledge protocols that bridge lattice-based schemes, specifically NTRU, with classical constructions like Pedersen vector commitments and ElGamal encryption. Our primary contributions include (1) a Σ-protocol for proving plaintext equality between an NTRU ciphertext and a Pedersen commitment, and (2) a ZKP of plaintext equality between NTRU and ElGamal ciphertexts. Both constructions ensure perfect honest-verifier zero-knowledge and computational soundness, while preserving efficiency and composability.

A central innovation of our work lies in constructing a common linear language across domains—leveraging homomorphic properties and inner product arguments—allowing the prover to demonstrate equivalence of messages without revealing their content. Our protocols integrate rejection sampling techniques to preserve privacy in the lattice setting and achieve $2n$-special soundness.

We further extend our constructions to support batch proofs, enabling scalable and bandwidth-efficient verification of multiple plaintext equalities. These protocols are, to the best of our knowledge, the first concrete and fully specified ZKPs achieving plaintext equality across NTRU and widely used classical primitives. Our work lays foundational tools for secure interoperability in hybrid systems and facilitates verifiable migration paths toward post-quantum secure infrastructures.

Table of Contents

Chapter 1

Introduction

As the threat of quantum computing becomes increasingly tangible, cryptographic systems must begin the transition from classical to post-quantum security models. Many standardized primitives that underpin today's cryptographic infrastructure—such as RSA, DSA, ElGamal, and discrete-log-based commitments—are rendered insecure in the presence of a quantum adversary. In response, post-quantum cryptography (PQC) aims to replace vulnerable primitives with constructions based on assumptions conjectured to be hard even for quantum computers. Among the most promising candidates are lattice-based encryption schemes, such as NTRU and those based on Learning With Errors (LWE), which offer practical efficiency and strong security foundations.

However, transitioning complex systems to PQC is nontrivial. In practice, the cryptographic landscape is increasingly characterized by *hybrid environments*, in which classical and post-quantum primitives coexist. For example, a system might use post-quantum encryption for long-term secrecy while continuing to rely on group-based commitments for compatibility with existing zero-knowledge infrastructures, or vice versa. This hybrid composition introduces the need for secure and privacy-preserving interoperability between primitives defined over fundamentally different algebraic domains.

A particularly important challenge in such settings is to establish *plaintext consistency* between two distinct representations of a secret—such as an encryption under a lattice-based scheme and a commitment or encryption under a classical group-based scheme. Without such guarantees, adversaries could equivocate by encrypting one value while committing to another. This undermines integrity and correctness, especially in applications involving voting, financial transactions, or outsourced computation. To address this challenge without compromising privacy, one requires efficient *zero-knowledge proofs of plaintext equality* between cryptographic objects defined in different worlds.

In this work, we have designed and analyzed zero-knowledge protocols for proving plain-text equality between representations in hybrid cryptographic environments. We focus on two key constructions:

1. A protocol to prove that a message encrypted under **NTRU** is equal to the message encrypted

under the **ElGamal encryption scheme**.

2. A protocol to prove that a message encrypted under the **NTRU encryption scheme** is included in a commitment, committed using a **Pedersen vector commitment** along with another bunch of messages.

Our protocols are built within the Sigma protocol framework and rely on the additive homomorphism shared by the involved schemes. The main technical challenge is to bridge the algebraic domains: lattice-based ciphertexts live in polynomial rings over integer moduli, while ElGamal and Pedersen constructions operate in discrete-logarithm groups. To resolve this, we construct a common language for linear relationships among plaintexts, allowing the prover to demonstrate equality of representations across these domains without revealing the plaintext itself.

For the NTRU–Pedersen case, we consider a setting in which the prover possesses a matrix $M \in \mathbb{Z}_p^{n \times N}$ and has committed to it via a compressed Pedersen vector commitment. The prover also publishes an NTRU ciphertext of one column m_k of the matrix. Using a carefully designed compression scheme via inner product with a public vector $\mathbf{x}$, and leveraging the additive homomorphism of both the commitment and encryption schemes, we construct a zero-knowledge protocol that proves the NTRU ciphertext decrypts to the same message committed in the corresponding position in M. To ensure privacy and soundness in the lattice setting, we incorporate rejection sampling and construct the protocol to satisfy $2n$*-special soundness*, where n is the dimension of the plaintext.

In the NTRU–ElGamal construction, we again exploit the linearity of the plaintext spaces to design a protocol that proves the equality of the two ciphertexts under the respective schemes. The key idea is to encode the same message in both encryption formats and prove, in zero knowledge, that the randomness used in the ElGamal encryption is consistent with a masked decryption of the NTRU ciphertext. Despite the domain mismatch, our protocol avoids revealing the message or the secret keys involved.

Both constructions are designed to be efficient, composable, and secure under standard assumptions. Our protocols satisfy perfect honest-verifier zero-knowledge and computational soundness. We also extend the first construction to support batching: proving equality of multiple ciphertexts simultaneously with a single commitment, achieving communication savings proportional to the number of encrypted messages.

To our knowledge, these are the first concrete and fully specified zero-knowledge proofs of plaintext equality between NTRU and widely-used classical primitives like Pedersen vector commitments and ElGamal ciphertexts. Our techniques contribute to the growing body of tools needed to support secure cryptographic migration and hybrid system design in the post-quantum era.

Chapter 2

Preliminaries

2.1 Notations

Let $R = \mathbb{Z}[X]/ < X^n + 1 >$ and $R_q = \mathbb{Z}_q[X]/ < X^n + 1 >$ for some prime q. Then $R_q = R/qR$. Any $a \in R$ is a polynomial of degree at-most $n-1$. We also can think of a as a vector $\mathbf{a} = (a_0, a_1, \ldots, a_{n-1})$ where the i^{th} component of $\mathbf{a}$ is the coefficient of i^{th} degree term the polynomial $\mathbf{a}$ i.e $\mathbf{a} = a_0 + a_1X + \ldots + a_{n-1}X^{n-1}$. Similarly any $\mathbf{b} \in R_q$ is a polynomial of degree at-most $n-1$ with coefficients in $\mathbb{Z}_q$ i.e $\|\mathbf{b}\|_\infty \leq q$.

Notice that $X^l \cdot \mathbf{a} = (-a_{n-l}, -a_{n-l+1}, \ldots, -a_{n-l}, a_0, \ldots, a_{n-l-1})$, as multiplying with X^l comes with modulo reduction by $X^n + 1$. We denote $X^l \cdot \mathbf{a}$ by $\mathbf{a}_{\ll l} = (-a_{n-l}, -a_{n-l+1}, \ldots, -a_{n-l}, a_0, \ldots, a_{n-l-1})$. We choose the *Discrete Gaussian Distribution* to sample a polynomial from R_q.

Definition 2.1.1. ***Discrete Gaussian Distribution.*** *The Discrete Gaussian Distribution over $\mathbb{Z}^n$ with mean v and standard deviation σ is defined by $D^n_{v,\sigma}(x) = \rho^n_{v,\sigma}(x)/\rho_\sigma(\mathbb{Z}^n)$, where $\rho^n_{v,\sigma}(x) = (\frac{1}{\sqrt{2\pi}\sigma})^n \, e^{-\frac{\|x-v\|^2}{2\sigma^2}}$ being the continuous normal distribution on $\mathbb{R}^n$ and $\rho_\sigma(\mathbb{Z}^n) = \sum_{z \in \mathbb{Z}^n} \rho^n_{0,\sigma}(z)$ is a scaling factor used to obtain a probability distribution.*

In case of $v = 0$ we write $D^n_{0,\sigma} = D^n_\sigma$. Note that we will write $u \xleftarrow{\$} D_{v,\sigma}$ or $D^n_{v,\sigma}$ to imply $u \in R_q$ is chosen over the discrete gaussian distribution. We will define our lattice based encryption scheme **NTRU** over R_q, where sampling from the distribution $D^n_{v,\sigma}$ plays a crucial role.

We define the inner product of two vectors $\mathbf{a}, \mathbf{b}$ by $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=0}^{n-1} a_i b_i$, where $\mathbf{a} = [a_i]_{i=0}^{n-1}$ and $\mathbf{b} = [b_i]_{i=0}^{n-1}$. In our context we denote $\langle \mathbf{a}, \mathbf{b} \rangle$ in short to express $\langle \mathbf{a}, \mathbf{b} \rangle$ mod p, where $p < q$ is a prime.

2.2 NTRU Encryption Scheme

In this work we aim to present zero-knowledge proof of plain-text equality between Additive Homomorphic Encryptions and *Pedersen's vector Commitment*. We start with a Lattice based encryption scheme

which is additive homomorphic for our proof of plain-text equality. We choose *NTRU* which is Lattice based Encryption scheme and is additive homomorphic. The scheme is describe below :

- **Set up :** Let p be a prime and q is another prime and $p < q$. Let $\lambda, \sigma \in \mathbb{R}$.
- **Plain-text Space :** The plain-text space or message space is $\mathbb{M} \subseteq \{\mathbf{m} \in R_q : \|\mathbf{m}\|_\infty < p\}$.
- **KeyGen :** $f', g \xleftarrow{\$} D_\sigma$ and set $f = pf' + 1$ and resample if f, g are not invertible. Set $h = \frac{pg}{f}$ and output $(pk, sk) = (h, f)$.
- **Encryption Algorithm :** To encrypt a message $m \in \mathcal{M}$, we sample $s, e \xleftarrow{\$} D_\sigma$ and set $y = hs + pe + m$. It is evident that $y \in R_q$ as the multiplications of the polynomials comes with modulo reduction by $X^n + 1$ and as f is invertible so $g/f \in R_q \implies h \in R_q$. Also $\|s\|, \|e\| \leq O(\sqrt{n}\lambda)$ i.e s, e have small coefficients thus $hs + pe \in R_q$.
- **Decryption Algorithm :** To decrypt we compute $m = fy \pmod p$.

For large $\sigma = O(n\sqrt{q})$, g/f becomes uniformly random in R_q [SS11]. The security of the encryption scheme is based on Ring-LWE problem. Although for small σ the scheme is more efficient but in that case we assume that $h = g/f$ is indistinguishable from uniformly random.
Let $R_q^\times$ be the collection of all invertible polynomials in R_q. Let $D_{z,\sigma}^\times$ be the discrete Gaussian distribution over $R_q^\times$. Then the following lemma from [SSTX09] explains the statistical closeness to uniformity of a quotient of two distributions:

Lemma 2.2.1. *Let $n \geq 8$ be a power of 2 such that $\Phi = x^n + 1$ splits into n linear factors modulo prime $q \geq 5$. Let $\epsilon > 0$ and $\sigma \geq 2n\sqrt{ln(8nq)} \cdot q^{\frac{1}{2}+2\epsilon}$. Let $f \in R_q^\times$ and $h_i \in R_q$ for $i \in \{1, 2\}$ and $z_i = -h_i f^{-1}$ mod q for $i \in \{1, 2\}$. Then*

$$\Delta\left[\frac{h_1 + f \cdot D_{z_1,\sigma}}{h_2 + f \cdot D_{z_2,\sigma}} \mod q \,;\, U(R_q^\times)\right] \leq 2^{3n} q^{\lfloor \epsilon n \rfloor}$$

This lemma includes the case of public key uniformity for *NTRU* i.e $h = pg/f$ is appears to be random when obtained from key **KeyGeneration** algorithm of *NTRU* for described sampling from D_σ with $\sigma \geq 2n\sqrt{ln(8nq)} \cdot q^{\frac{1}{2}+2\epsilon}$.

2.3 Rejection Sampling

A zero-knowledge protocol is considered to be *zero-knowledge* if it enables a prover to convince a verifier of the validity of a statement without revealing any additional information beyond the fact that the statement is true. For this property to hold, it is essential that the prover's responses throughout the protocol are statistically independent of the secret witness; that is, they must not leak any information that could be used to reconstruct or infer the witness. Any dependence between the responses and the witness could lead to a breach in privacy, violating the foundational principles of zero-knowledge. To

ensure such independence in our zero-knowledge proof construction, we adopt the technique of *rejection sampling*. This method allows the prover to generate candidate responses using some distribution and accept with certain probability, effectively filtering the outputs to achieve a distribution that is indistinguishable from one generated without knowledge of the witness. To maintain both practicality and security, we bound the number of sampling attempts to a constant value, ensuring that the process remains efficient while still satisfying the statistical properties necessary for zero-knowledge. By incorporating rejection sampling in this controlled manner, we ensure that the prover's messages reveal nothing about the underlying witness, thereby preserving the zero-knowledge nature of the protocol. To be concrete we state a theorem from a work of Lyubashevsky [Lyu12].

Theorem 2.3.1. *Let V be a subset of $\mathbb{Z}^l$ in which all elements have norm less than T , and let H be a probability distribution over V. Then for any constant C, there exists a $\sigma = \Theta(T)$ such that the output distributions of the following algorithms* $\mathtt{A}, \mathtt{F}$ *are statistically close :*

$$\mathtt{A} \ : \ v \xleftarrow{\$} H;\ z \xleftarrow{\$} D^l_{v,\sigma}\ ;\ \textit{output}\ (z, v)\ \textit{with probability}\ \ \mathtt{min}\left(\frac{D^l_\sigma(z)}{CD^l_{v,\sigma}(z)},\ 1\right)$$

$$\mathtt{F} \ : \ v \xleftarrow{\$} H;\ z \xleftarrow{\$} D^l_{0,\sigma}\ ;\ \textit{output}\ (z, v)\ \textit{with probability}\ \ \frac{1}{C}$$

The probability that $\mathtt{A}$ *outputs something is exponentially close to that of* $\mathtt{F}$*, i.e* $\frac{1}{C}$.

What it means in our context is that the secret witness of the zero-knowledge protocol, chosen from a distribution H and generating responses relying on a discrete normal distribution with a certain probability is statistically equivalent to generating responses independent of the secret.

2.4 A Technical Lemma

We now discuss a property of the ring $\mathbb{Z}[X]/ < X^n + 1 >$ which plays a crucial role in our zero-knowledge proof technique.

Lemma 2.4.1. *Let n be power of 2 and let $0 < i, j < 2n - 1$. Then $2(X^i - X^j)^{-1}$ mod $(X^n + 1)$ only has coefficients in $\{-1, 0, 1\}$.*

Proof. Without loss of generality, assume that $j > i$. Using that

$$X^n = -1 \mod (X^n + 1) \tag{2.1}$$

we have that

$$\begin{aligned}
& 2(X^i - X^j)^{-1} = 2X^{-i}(1 - X^{j-i})^{-1} \\
\implies & 2(X^i - X^j)^{-1} = -2(-1)X^{-i}(1 - X^{j-i})^{-1} \\
\implies & 2(X^i - X^j)^{-1} = -2(X^n)X^{-i}(1 - X^{j-i})^{-1} \quad (using\ 2.1) \\
\implies & 2(X^i - X^j)^{-1} = -2X^{n-i}(1 - X^{j-i})^{-1}
\end{aligned}$$

It is therefore sufficient to prove the claim for $i = 0$ only.
Now remark that, for every $k \geq 1$ it holds that:

$$(1 - X^j)(1 + X^j + X^{2j} + \cdots + X^{(k-1)j}) = 1 - X^{kj} \tag{2.2}$$

The key requirement of this proof is $X^{kj} = -1 \mod (X^n + 1)$.

Let us write $j = 2^{j'} j''$, with j'' a positive odd integer and $0 \leq j' \leq \log_2(n)$.
Now let us choose $k = 2^{\log_2(n) - j'}$ (recall that n is a power of 2).
Then we have,

$$\begin{aligned}
& jk = 2^{j'} j'' \cdot 2^{\log_2(n) - j'} \\
\implies & jk = 2^{\log_2(n) - j' + j'} j'' \\
\implies & jk = 2^{\log_2(n)} j'' \\
\implies & jk = nj''
\end{aligned}$$

So, $X^{kj} = X^{nj''} = (-1)^{j''} \mod (X^n + 1)$

$$\implies 1 - X^{kj} = 2 \mod (X^n + 1) \tag{2.3}$$

Therefore using 2.2 and 2.3 we have

$$\begin{aligned}
& 2 = (1 - X^j)(1 + X^j + X^{2j} + \cdots + X^{(k-1)j} \mod (X^n + 1)) \\
\implies & 2(1 - X^j)^{-1} = 1 + X^j + X^{2j} + \cdots + X^{(k-1)j} \mod (X^n + 1) \\
\implies & 2(1 - X^j)^{-1} = 1 + X^{j \mod n} + X^{2j \mod n} + \cdots + X^{(k-1)j \mod n} \mod (X^n + 1)
\end{aligned}$$

There is no smaller k that is no smaller power of 2, that would work because k needs to cancel out the $2^{j'}$ factor in j.

Finally, in this equation, no two exponents are equal, since otherwise that would mean that n divides jk' with $1 \leq k' < k$, which is impossible by definition of k.

□

2.5 Pedersen's Commitment

A cryptographic commitment scheme can be characterized by a tuple (`Setup, Commit, Open`). The `Setup` algorithm generates the public parameters that define the operational context of the scheme. Utilizing these parameters, the `Commit` algorithm allows a party to produce a commitment cmt to a message m, incorporating inherent randomness to ensure hiding and binding properties. To unveil the original message, the committer reveals both the message m and the randomness used. The `Open` algorithm then verifies the correctness of the commitment by checking the consistency between the disclosed data and the original commitment value.

The Pedersen commitment scheme, introduced by Pedersen [Ped92], is a well-established construction grounded in the hardness of the discrete logarithm problem. Consider a family of cyclic groups $\mathbb{G}(\lambda)_{\lambda\in\mathbb{N}}$ of prime order $\tilde{q} = \tilde{q}(\lambda)$, where λ denotes the security parameter. It is assumed that the discrete logarithm problem is computationally infeasible in each group $\mathbb{G}(\lambda)$. For clarity and consistency, we adopt additive notation throughout this discussion and denote elements of order $\tilde{q}$ with a tilde.

- `Setup`: The setup algorithm selects a generator $\tilde{g}$ of the group $\mathbb{G}$, and samples an element $\tilde{h} \xleftarrow{\$} \langle\tilde{g}\rangle$ uniformly at random. It outputs the public parameters $(\tilde{g}, \tilde{h})$.
- `Commit`: To commit to a message $m \in \mathbb{Z}_{\tilde{q}}$, the algorithm samples a random value $r \xleftarrow{\$} \mathbb{Z}_{\tilde{q}}$ and computes the commitment as the pair $(cmt, o) = (m\tilde{g} + r\tilde{h},\ r)$.
- `Open`: To open a commitment, the algorithm receives the commitment value cmt, opening value o, the public parameters $(\tilde{g}, \tilde{h})$, and a message m. It verifies correctness by checking whether $cmt \stackrel{?}{=} m\tilde{g} + o\tilde{h}$.

A commitment scheme is said to be perfectly hiding if, for any two messages $m_0, m_1 \in \mathcal{M}$, the distributions of their respective commitments are statistically indistinguishable—even to an unbounded adversary. In other words, the commitment reveals no information about the committed message, ensuring unconditional privacy.

Conversely, a scheme is computationally binding if it is infeasible for a polynomial-time adversary to find two distinct pairs (m, r) and (m', r') such that $m \neq m'$ but $m\tilde{g} + r\tilde{h} = m'\tilde{g} + r'\tilde{h}$. This property relies on the assumption that computing discrete logarithms in $\mathbb{G}$ is hard, thereby preventing equivocation by a dishonest committer.

Theorem 2.5.1. *Under the discrete logarithm assumption for $\mathbb{G}$, the given commitment scheme is perfectly hiding and computationally binding.*

2.5.1 Pedersen's Vector Commitment

Let $m = (m_0, m_1, \ldots, m_{N-1}) \in \mathbb{M}^N$, where $\mathbb{M} \subseteq \{x \in R_q : ||x||_\infty < p\}$ denotes the space of valid message vectors whose entries lie in $\mathbb{Z}_q$ and are bounded in absolute value by p. To commit such a plaintext,

one can employ the *Pedersen vector commitment* scheme. This method reduces space complexity while maintaining the core cryptographic properties.

In this construction, instead of sampling a vector of random values, we draw a single scalar $r \xleftarrow{\$} \mathbb{Z}_{\tilde{q}}$ to randomize the commitment. We randomly choose N generators $\tilde{g}_0, \tilde{g}_1, \ldots, \tilde{g}_{N-1} \xleftarrow{\$} \mathbb{G}$ and another random group element $\tilde{h} \xleftarrow{\$} \mathbb{G}$. We declare $\{\tilde{g}_0, \tilde{g}_1, \ldots, \tilde{g}_{N-1}\ ;\ \tilde{h}\}$ as the public parameters of the *Pedersen Vector Commitment*. The commitment is then computed as:

$$cmt = \sum_{i=0}^{N-1} m_i \cdot \tilde{g}_i \ + \ r \cdot \tilde{h}$$

To open the commitment, the sender reveals the message vector m and the scalar $o = r$. The verifier then confirms the validity of the opening by checking:

$$cmt \stackrel{?}{=} \sum_{i=0}^{N-1} m_i \cdot \tilde{g}_i \ + \ o \cdot \tilde{h}$$

Pedersen's vector commitment retains the fundamental security guarantees of the original scheme: it is *unconditionally hiding*, meaning that no adversary—regardless of computational power—can learn anything about the message from the commitment alone. At the same time, it is *computationally binding*, as it is computationally infeasible (under the discrete logarithm assumption) to find two distinct openings (m, r) and (m', r') that produce the same commitment.

2.6 ElGamal Encryption Scheme

The ElGamal encryption scheme is a classical public-key encryption system whose security relies on the computational hardness of the discrete logarithm problem in cyclic groups of prime order. It serves as one of the fundamental primitives in group-based cryptography and provides additive homomorphism under specific representations. Below, we describe the ElGamal encryption scheme within the context of a group $\mathbb{G}$ of prime order q.

Setup. Let $\mathbb{G}$ be a cyclic group of prime order q, where the discrete logarithm problem is hard. Let $\tilde{g} \in \mathbb{G}$ be a generator of the group. Let $\mathbb{A} \subseteq \mathbb{Z}_q$ and we define the message space by $\mathbb{M} = \{m \cdot \tilde{g} \ :\ m \in \mathbb{A}\}$.

Key Generation. The secret key is a random scalar $x \xleftarrow{\$} \mathbb{Z}_q$, and the corresponding public key is $\tilde{h} = \tilde{g}^x \in \mathbb{G}$. The key pair is $(\tilde{h}, x)$.

Encryption. To encrypt a message $m \in \mathbb{A}$ under public key $\tilde{h}$, the sender samples a random ephemeral key $r \xleftarrow{\$} \mathbb{Z}_q$ and computes the ciphertext as the pair:

$$\mathsf{Enc}(m; r) = (\tilde{c}_1, \tilde{c}_2) = (\tilde{g}^r, \tilde{h}^r + m \cdot \tilde{g}) \in \mathbb{G} \times \mathbb{G}$$

Note that the message m is encoded as an additive multiple of the generator $\tilde{g}$.

Decryption. Given the ciphertext $(\tilde{c}_1, \tilde{c}_2)$ and secret key x, the receiver computes:

$$\tilde{c}_2 - \tilde{c}_1^x = \tilde{h}^r + m \cdot \tilde{g} - (\tilde{g}^x)^r = m \cdot \tilde{g}$$

Homomorphic Property. The ElGamal encryption scheme is additively homomorphic when messages are encoded as scalar multiples of the group generator. Given two ciphertexts $\mathsf{Enc}(m_1; r_1) = (\tilde{c}_1^{(1)}, \tilde{c}_2^{(1)})$ and $\mathsf{Enc}(m_2; r_2) = (\tilde{c}_1^{(2)}, \tilde{c}_2^{(2)})$, we define:

$$(\tilde{c}_1^{(1)}, \tilde{c}_2^{(1)}) + (\tilde{c}_1^{(2)}, \tilde{c}_2^{(2)}) := (\tilde{c}_1^{(1)} \cdot \tilde{c}_1^{(2)}, \tilde{c}_2^{(1)} + \tilde{c}_2^{(2)})$$

The resulting ciphertext is a valid encryption of $m_1 + m_2$ under randomness $r_1 + r_2$.

Security. The security of ElGamal encryption in the chosen-plaintext setting relies on the Decisional Diffie-Hellman (DDH) assumption in $\mathbb{G}$. In particular, given $(\tilde{g}, \tilde{g}^a, \tilde{g}^b, \tilde{g}^{ab})$, it should be computationally hard to distinguish $\tilde{g}^{ab}$ from a random group element in $\mathbb{G}$.

Chapter 3

Polynomial Commitment Scheme

In the work of *Jonathan Bootle* and *Jens Groth* [BG18], the authors present a rigorous method for committing to a vector of plaintexts wherein each component itself is a vector, thereby enabling a structured and compositional commitment scheme. This approach of polynomial commitment scheme works as a Σ-protocol to verify the commitments. The intuition is that the prover does not need to send the entire message and the randomness as opening of the commitment rather it sends a computation dependent on the challenge sent by the verifier. This computation works as an opening and the verifier can verify without knowing the message. In this work we will use a variant of their approach, where we have made some necessary changes in their technique for our requirements regarding the proof.

Our problem statement evolves around proving the plaintext equality of a plaintext which is encrypted under *NTRU* and committed under *Pedersen Vector Commitment* along with another bunch of plaintexts. Let,

$$\mathcal{M} = [M_0 \;\; M_1 \;\; \dots \;\; M_{N-1}] = \begin{bmatrix} m_{0,0} & m_{0,1} & m_{0,2} & \dots & m_{0,(N-1)} \\ m_{1,0} & m_{1,1} & m_{1,2} & \dots & m_{1,(N-1)} \\ \vdots & & & & \\ m_{(n-1),0} & m_{(n-1),1} & m_{(n-1),2} & \dots & m_{(n-1),(N-1)} \end{bmatrix},$$

where $\forall\ j,\ M_j \in \mathbb{M},$ and is the j^{th} column vector of $\mathcal{M}$. Here $\mathcal{M}$ represents a polynomial

$$\mathcal{Q}(X) = \sum_{j=0}^{N-1} \left(\sum_{i=0}^{n-1} m_{i,j} \cdot X^i \right) X^{nj}$$

Let $\tilde{g}_0, \tilde{g}_1, \dots, \tilde{g}_{N-1} \in \mathbb{G}$ and $\tilde{h} \xleftarrow{\$} \mathbb{G}$ be the public parameters and $|\mathbb{G}| = p$. We choose $r_i \xleftarrow{\$} \mathbb{Z}_p$ for $i = 0, 1, \dots, n-1$ and commit each row of $\mathcal{M}$ with *Pedersen Vector Commitment* as follows,

$$L_i = \sum_{j=0}^{N-1} m_{i,j} \cdot \tilde{g}_j \; + r_i \cdot \tilde{h}, \quad \text{for } i = 0, 1, \dots, n-1$$

. Let $L = [L_0 \;\; L_1 \;\; \ldots \;\; L_{n-1}]$. We treat L as the commitment of $\mathcal{M}$. The prover sends L to the verifier and the verifier responds with a challenge vector $\mathbf{x} = [x_0 \;\; x_1 \;\; \ldots \;\; x_{n-1}]$. The prover computes $l_j = \langle M_j, \mathbf{x} \rangle$ for $j = 0, 1, \ldots, N-1$ and $\bar{r} = \langle \mathbf{r}, \mathbf{x} \rangle$, where $\mathbf{r} = [r_0 \;\; r_1 \;\; \ldots \;\; r_{n-1}]$. Prover sends $\mathbf{l} = [l_0 \;\; l_1 \;\; \ldots \;\; l_{N-1}]$ and $\bar{r}$ to the verifier as opening of the commitment L. The verifier checks if the the *Pedersen Vector Commitment* of $\mathbf{l}$ with randomizer $\bar{r}$ matches with $\langle L, \mathbf{x} \rangle$, i.e

$$\langle L, \mathbf{x} \rangle \stackrel{?}{=} \sum_{j=0}^{N-1} l_j \cdot \tilde{g}_j \; + \; \bar{r} \cdot \tilde{h}$$

. In the original work of *Jonathan Bootle* and *Jens Groth* [BG18], polynomial evaluation of a field element x is used over the polynomials M_j's, where $x \in \mathbb{F}$ was sent as a challenge by the verifier. For our convenience, we use inner product instead of polynomial evaluation as inner products are additively homomorphic i.e

$$\langle M_j, \mathbf{x} + \mathbf{x}' \rangle = \langle M_j, \mathbf{x} \rangle \; + \; \langle M_j, \mathbf{x}' \rangle \quad \text{but}$$

$$M_j(x + x') \neq M_j(x) + M_j(x')$$

Our construction of committing $\mathcal{M}$ follows this technique but in a different way. We choose a random vector $\mathbf{x}$ and compress the matrix $\mathcal{M}$ into the vector $\mathbf{l} = [\; l_j \;]_{j=0}^{N-1}$ and commit $\mathbf{l}$ with *Pedersen Vector Commitment* i.e $cmt = \sum_{j=0}^{N-1} l_j \cdot \tilde{g}_j \; + \; \left(\sum_{i=0}^{n-1} r_i \cdot x_i \right) \tilde{h}$, for randomly chosen $\mathbf{r} = [r_0 \;\; \ldots \;\; r_{n-1}] \xleftarrow{\$} \mathbb{Z}_p^n$. We call cmt as the commitment of $\mathcal{M}$. We keep the vectors $\mathbf{x}$ and L public for future use. To open the commitment one can send $\mathcal{M}$, $\mathbf{r}$ and verification is done by checking $cmt \stackrel{?}{=} \sum_{j=0}^{N-1} \langle M_j, \mathbf{x} \rangle \cdot \tilde{g}_j \; + \; \langle \mathbf{r}, \mathbf{x} \rangle \cdot \tilde{h}$. Our goal is to prove that the k^{th} column of $\mathcal{M}$ is encrypted to a publicly available *NTRU* cipher and also M_k is included in cmt. Since we commit the matrix $\mathcal{M}$ with a variant of **Pedersen Vector Commitment**, then the security guaranty follows from its *unconditionally hiding* and *computationally binding* property.

Chapter 4

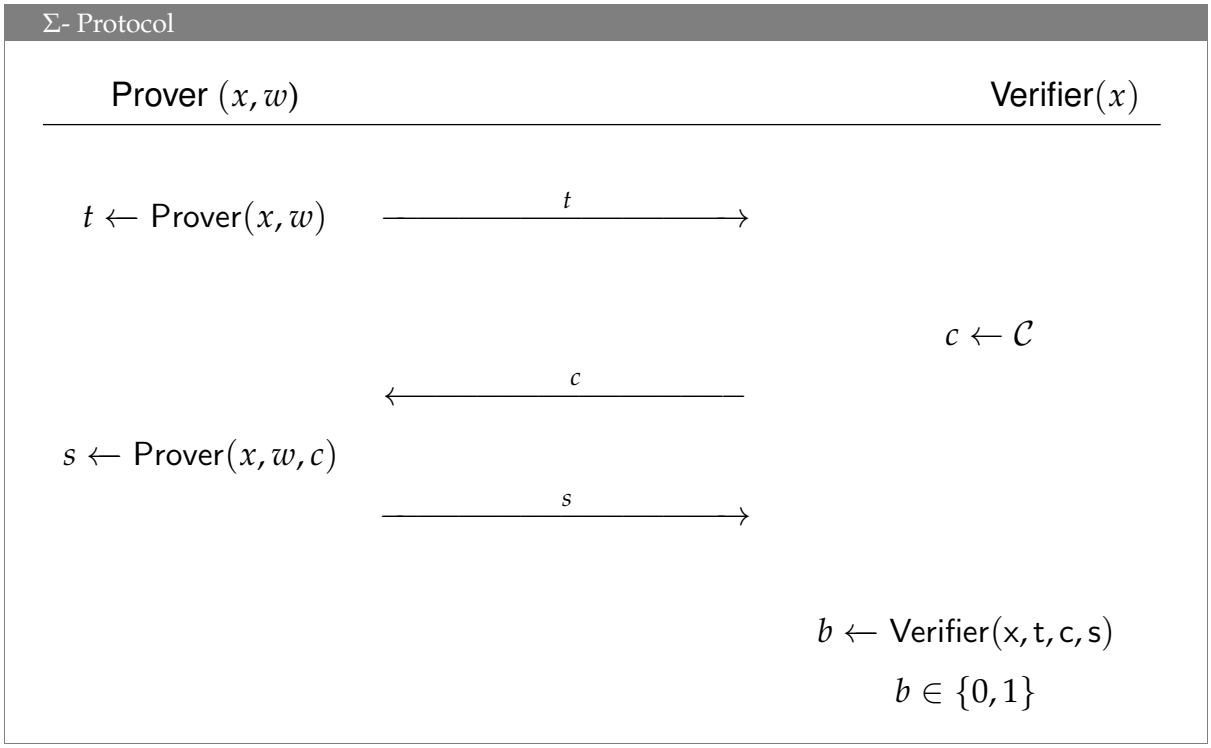
Σ-Protocols

Zero-knowledge proofs of knowledge (ZKPoKs) are cryptographic protocols where a prover convinces a verifier of knowledge of a secret (called a *witness*) for some public statement, without revealing anything beyond the validity of the claim. A formal treatment can be found in the work of Bellare and Goldreich [BG93].

Formally, let $\mathcal{L} \subseteq \{0,1\}^*$ be a language and $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ be the associated witness relation, where $(x, w) \in \mathcal{R}$ if w is a valid witness for $x \in \mathcal{L}$. In our context, the proofs we use are generalized Σ-protocols, referred to as Σ'-protocols. These protocols are three-move interactions between a prover and a verifier with some relaxed completeness conditions and an extended soundness model.

Definition 4.0.1 (Σ'-Protocol)**.** *Let (P, V) be a two-party interactive protocol where V is probabilistic polynomial-time (PPT). Let $\mathcal{L} \subseteq \{0,1\}^*$ be a language with a witness relation $\mathcal{R}$, and let $\mathcal{L}' \subseteq \{0,1\}^*$ with $\mathcal{R}' \supseteq \mathcal{R}$ be an extended relation. Then, (P, V) is a Σ'-protocol for $(\mathcal{L}, \mathcal{L}')$ with challenge space $\mathcal{C}$, completeness error α, and the following properties:*

- ***Three-move structure:*** *The prover sends a commitment t, the verifier replies with a challenge $c \in \mathcal{C}$ where $\mathcal{C}$ is the challenge space, and the prover responds with a value s. The verifier either accepts or rejects based on the transcript (t, c, s).*
- ***Completeness:*** *If $(x, w) \in \mathcal{R}$, then V accepts with probability at least $1 - \alpha$, for some negligible α.*
- ***Special soundness:*** *There exists a PPT extractor E that, given two accepting transcripts with the same first message and different challenges (t, c, s) and (t, c', s'), can extract a witness w' such that $(x, w') \in \mathcal{R}'$.*
- ***Special honest-verifier zero-knowledge (HVZK):*** *There exists a PPT simulator $\mathcal{S}$ that, on input $x \in \mathcal{L}$ and a challenge $c \in \mathcal{C}$, outputs (t, s) such that the transcript (t, c, s) is computationally indistinguishable from a real protocol execution.*



This variant of Σ-protocols differs from the classical notion in two primary aspects:

1. *Relaxed completeness:* We allow the honest prover to fail with probability α, to accommodate constructions such as rejection sampling that inherently include aborts.

2. *Extended soundness:* The relation $\mathcal{R}'$ may include more witness instances than $\mathcal{R}$. This accounts for protocols where the verifier only gains assurance about a weaker statement than the one held by the prover.

This formulation departs from the classical definition of Σ-protocols in two fundamental ways.

First, we permit the honest prover to fail in up to an α-fraction of the protocol executions, thereby relaxing the perfect completeness requirement ($\alpha = 0$) imposed in standard definitions. This flexibility is crucial in our construction due to the reliance on rejection sampling [Lyu09, Lyu12], where the prover may need to abort in order to preserve zero-knowledge.

Second, we generalize the notion of the witness relation. Specifically, we introduce an auxiliary language $\mathcal{L}'$ with a corresponding relation $\mathcal{R}' \subseteq \mathcal{R}$, where $\mathcal{R}$ defines the original relation. In this extended setting, it suffices that the prover holds a witness for $\mathcal{L}'$, while the verifier only ensures soundness with respect to the original language $\mathcal{L}$. This distinction allows for additional flexibility and has been utilized in prior works such as [AJL+12], and discussed informally in [DF02, FO97].

If the *soundness gap* between $\mathcal{R}$ and $\mathcal{R}'$ is suitably small, this relaxed approach still suffices for many higher-level cryptographic applications. It is worth noting that the classical Σ-protocol framework is recovered in the special case where $\alpha = 0$ and $\mathcal{R} = \mathcal{R}'$.

We emphasize that existing results establishing that a Σ-protocol constitutes an honest-verifier zero-knowledge proof of knowledge (ZKPoK) with knowledge error bounded by $1/|\mathcal{C}|$ continue to hold under our relaxed model, provided the condition $1 - \alpha > 1/|\mathcal{C}|$ is satisfied. Furthermore, achieving zero-knowledge against arbitrary (possibly malicious) verifiers remains possible by invoking well-known transformations, such as those by Damgård *et al.* [Dam00, DGOW95].

Finally, a well-known amplification technique ensures that both knowledge and completeness errors in α-relaxed settings can be reduced to negligible levels. Specifically, running the protocol in parallel λ times and accepting only if at least $\lambda(1-\alpha)/2$ transcripts are accepting, yields soundness provided there exists a constant c such that:

$$\frac{(1-\alpha)}{2} > \frac{1}{|\mathcal{C}|} + c.$$

Some Σ'-protocols used in our constructions further satisfy the following properties:

- **Quasi-unique responses:** It is computationally infeasible for an adversary to generate two distinct responses $s \neq s'$ such that both (t, c, s) and (t, c, s') are accepted by the verifier.
- **High-entropy commitments:** The initial message t generated by an honest prover is statistically unpredictable and has high min-entropy, ensuring resistance to offline guessing attacks.

These generalizations are essential for enabling efficient lattice-based zero-knowledge proofs that incorporate rejection sampling and extend to hybrid settings, where the witness may include elements from both lattice-based and classical primitives.

Now we define the properties of a Σ-protocol in a more concrete way,

Completeness The protocol is *perfectly complete* if an honest prover with a valid witness can always convince the verifier. That is, for all PPT adversaries A:

$$\Pr\left[\begin{array}{l} x \leftarrow \mathcal{L}; \\ t \leftarrow \mathsf{Prover}(x, w); \\ s \leftarrow \mathsf{Prover}(x, w, c) : \\ (x, w) \in \mathcal{R} \Rightarrow \mathsf{Verifier}(x, t, c, s) = 1 \end{array}\right] = 1$$

Special Soundness A Σ-protocol has n-special soundness if one can extract a valid witness from n accepting transcripts that share the same first message a, but use different challenges. That is, there exists a PPT algorithm χ such that:

$$\Pr\left[\begin{array}{l} (c_1, s_1, \ldots, c_n, s_n) \leftarrow A(\mathrm{x},\mathrm{t}); \\ w \leftarrow \chi(u, a, c_1, s_1, \ldots, c_n, s_n) : \\ \forall i, \mathsf{Verifier}(x, t, c_i, s_i) = 1 \Rightarrow (x, w) \in \mathcal{R} \end{array}\right] \approx 1$$

Special Honest-Verifier Zero-Knowledge (SHVZK) The protocol has SHVZK if the view of an honest verifier interacting with the prover can be simulated without access to the witness. That is, there exists a PPT simulator $\mathcal{S}$ such that:

$$\Pr\left[\begin{array}{l}(x,w,c) \leftarrow A(1^\lambda);\\(t,s) \leftarrow \mathcal{S}(x,c):\\ \mathsf{Verifier}(x,t,c,s)=1\end{array}\right] \approx \Pr\left[\begin{array}{l}(x,w,c) \leftarrow A(1^\lambda);\\ t \leftarrow \mathsf{Prover}(x,w);\\ s \leftarrow \mathsf{Prover}(x,w,c):\\ \mathsf{Verifier}(x,t,c,s)=1\end{array}\right]$$

Remarks While SHVZK is sufficient in the honest-verifier model, practical deployment often requires full zero-knowledge against arbitrary verifiers. This can be achieved via standard transformations, such as using the Fiat-Shamir heuristic in the random oracle model or applying compiler techniques to upgrade the SHVZK property to full zero-knowledge in the CRS model.

Chapter 5

Related Work

5.1 Zero-Knowledge Proofs for Ring-LWE Ciphertexts

Lattice-based encryption schemes (in particular Ring-LWE variants) admit Σ-protocols for proving plaintext knowledge or ciphertext correctness. Early constructions typically follow Stern-like protocols [Ste94], where the prover commits to random masks of the secret and responds to a two-way challenge, yielding soundness error $1/2$ per round. For example, Ling et al. [LNSW13] gave a Σ-protocol under SIS (extendable to LWE) but noted that handling the ring structure is non-trivial. Likewise, Damgård et al. [DPSZ12] devised an "amortized proof" to prove knowledge of $O(k)$ LWE plaintexts in the time for one, but this approach does not improve the single-instance case nor apply easily to Ring-LWE due to the required structured challenge matrix.

Subsequent work has developed specialized protocols exploiting the algebra of polynomial rings. Xie et al. [XXW13] adapt techniques from LPN-based proofs to the RLWE setting, showing a Σ-protocol that can certify any polynomial relation among ring elements. Their scheme proves committed ring elements $m, m_1, \ldots, m_t$ satisfy $m = f(m_1, \ldots, m_t)$ for a polynomial f, with amortized communication $\mathcal{O}(\lambda\,|f|)$ and exponentially small soundness error. In practice this means that one can prove complex relations on the plaintext (or error) components with soundness error negligible in the security parameter. More generally, Yang et al. (2019) observe that prior lattice ZK-arguments often had constant error (e.g. $2/3$) or only approximate extraction, and they construct linear-relation proofs with standard soundness and soundness error $1/\mathsf{poly}(n)$:contentReference[oaicite:3]index=3. These advances show that one can move beyond simple challenge spaces and achieve much smaller error in lattice proofs.

Building on these ideas, Benhamouda et al. [BCK+14] show a concrete soundness improvement for RLWE ciphertext proofs. They note that over the ring $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ one can exploit the identity $\frac{2}{X^i - X^j}$ has coefficients in $\{-1, 0, 1\}$. This algebraic trick allows the prover to show knowledge of a short

vector e such that $Ae = 2t$, where normally one would need $Ae = t$. Although $Ae = 2t$ is a weaker statement, it still suffices to recover the plaintext bit-string. Crucially, this expands the challenge space from size 2 to size $2n$, reducing the per-round soundness error from $1/2$ to roughly $1/(2n)$. In practical terms, this means roughly a $10\times$ reduction in the number of repetitions needed for a given security level. Their method applies to any ideal-lattice scheme (dual R-LWE, Lyubashevsky's two-element scheme, NTRU, etc.) that requires a proof of well-formedness.

More recently, specialized proofs have been proposed in application-specific settings. For instance, Bottazzi [Bot24] considers a voting scenario where each ballot ciphertext must be a valid RLWE encryption of a bit (0 or 1). His "Greco" protocol uses a ZK proof to ensure the submitted ciphertexts are well-formed Ring-LWE encryptions of a binary message. Similarly, Del Pino et al. [dLS19] improve proof efficiency by first forming a Pedersen commitment to the RLWE ciphertext's randomness and plaintext vector, then proving that the commitment and ciphertext match. By leveraging inner-product arguments for these commitments, they obtain very short proofs (on the order of 1–2 KB) and even handle multiple ciphertexts amortized in roughly $\log k$ overhead. These works illustrate the current trend: lattice ZK proofs are becoming much more efficient through clever algebraic techniques and integration with discrete-log commitments.

5.1.1 Proof Construction

Lets look at the primary work of Benhamouda et al. [BCK+14], where they have constructed a zero knowledge proof of plain-text knowledge for Ring-LWE secrets. Let $y = as + e$, where $a \in \mathbb{Z}_q^{n\times n}$ and $s, e \in R_q$. Their construction efficiently proves the knowledge of two short vectors $2s, 2e$ such that $2y = 2as + 2e$. For the construction of the proof let $s, e \xleftarrow{\$} D_\alpha$ be chosen from discrete Gaussian distribution with standard deviation α. The following protocol is equipped with capability of convincing the verifier that, the prover knows short vectors s' and e' such that $2y = 2as' + 2e'$. These vectors $2s', 2e'$ are evaluated under modulo q operations. The key intuition is to convince the verifier of prover knowing a secret witness for twice of the public input and hence prover might be aware of original witness. In this context, by *short* we mean the following: an honest prover can always successfully convince the verifier, provided that the norms satisfy $\|s\|, \|e\| \leq \mathcal{O}(\sqrt{n}\alpha)$. This condition holds with overwhelming probability when the vectors s and e are sampled according to the prescribed honest distribution. Conversely, the verifier is assured that any prover who succeeds must possess knowledge of LWE secrets whose norms do not exceed $\mathcal{O}(n^2\alpha)$. This discrepancy in the norm bounds between honest and arbitrary provers represents a typical soundness slack, as similarly encountered in prior works such as [AJL+12, DF02].

To enable simulation of aborts in the context of proving the zero-knowledge property of the proto-

col, the prover's initial message should not be transmitted in cleartext. Instead, it must be committed to and subsequently revealed during the final round of the Σ'-protocol. To achieve this, we employ an auxiliary commitment scheme denoted by $(\mathsf{aCSetup}, \mathsf{aCommit}, \mathsf{aCOpen})$, under the assumption that honestly generated parameters are publicly available to both parties.

We refrain from making any specific assumptions regarding the properties of the auxiliary commitment scheme. Nevertheless, if the scheme satisfies computational binding, then the overall protocol's soundness is upheld under that assumption. Analogously, if the scheme is computationally hiding, the zero-knowledge property holds under that condition. For simplicity, one may conceptualize the auxiliary scheme as behaving like a random oracle.

Zero-Knowledge Protocol for Knowledge of Ring-LWE Secrets

Prover (y, s, e)		Verifier (y)
$r_s, r_e \xleftarrow{\$} D_{O(\sqrt{n}\alpha)}$ $t = ar_s + r_e$		
Commit: $C_{aux} = \mathsf{aCommit}(t; d_{aux})$	$\xrightarrow{C_{aux}}$	
		$c \leftarrow \{0, \ldots, 2n-1\}$
	$\xleftarrow{c}$	
$S_s = r_s + X^c s$ $S_e = r_e + X^c e$ Accept with probability $\frac{D_{O(\sqrt{2n}\alpha)}((S_s, S_e))}{CD_{(X^c s, X^c e), O(\sqrt{2n}\alpha)}(S_s, S_e)}$		
	$\xrightarrow[S_s, S_e]{t\ ;\ d_{aux}}$	
		Check:
		$X^c y + t \stackrel{?}{=} aS_s + S_e$
		$\mathsf{aCOpen}(t, C_{aux}; d_{aux}) \stackrel{?}{=} accept$
		$\|S_s\|, \|S_e\| \leq \mathcal{O}(n\alpha)$

5.1.2 Security Analysis

Theorem 5.1.1. *The above protocol 5.1.1 is an* Honest Verifier Zero Knowledge Σ'*-protocol for the following relations:*

$$\mathcal{R} = \{((a,y),(s,e)) \ : \ y = as + e \ \wedge \ \|s\|, \|e\| \leq \mathcal{O}(\sqrt{n}\alpha)\}$$

$$\mathcal{R}' = \left\{((a,y),(s,e)) \ : \ 2y = 2as + 2e \ \wedge \ \|2s\|, \|2e\| \leq \mathcal{O}(n^2\alpha)\right\}$$

where $2s$ *and* $2e$ *are reduced modulo* q*. The protocol has a knowledge error of* $\frac{1}{2n}$*, a completeness error of* $1 - \frac{1}{C}$ *and high entropy commitments.*

It is worth observing that in 5.1.1, the rejection sampling technique is applied to the concatenated vector $(\mathbf{s}_e, \mathbf{s}_s)$ as a whole, rather than being invoked separately on each of $\mathbf{s}_e$ and $\mathbf{s}_s$. This unified approach leads to improved parameter efficiency—either in terms of C or the noise parameter $\sigma = \widetilde{\mathcal{O}}(T)$ as referenced in **Theorem 2.3.1**—achieving a reduction factor of approximately $\sqrt{2}$, attributed to the reliance on the Euclidean norm.

Proof. We prove the properties from Definition 4.0.1 one by one :
Completeness : Notice that by Theorem 2.3.1, the prover will respond with probability $\frac{1}{C}$. In case of no abort, we will have

$$aS_s + S_e = a(r_s + X^c s) + (r_e + X^c e) = X^c(as + e) + (ar_s + r_e) = X^c y + t$$

Regarding the norm bounds, we observe that

$$\|\mathbf{s}_s\| \leq \|\mathbf{r}_s\| + \|\mathbf{s}\| \leq \mathcal{O}(n\alpha)$$

holds with overwhelming probability. This follows from the fact that the standard deviation of $\mathbf{r}_s$ is $\widetilde{\mathcal{O}}(\sqrt{n}\alpha)$, and a similar argument applies to $\mathbf{s}_e$.

Honest Verifier Zero Knowledge : Consider a fixed challenge value c. The simulator $\mathcal{S}$ constructs a fake transcript in the following manner:

With probability $1 - \frac{1}{C}$, it outputs the tuple $(\mathsf{aCommit}(0), c, \perp)$ to simulate an abort.

With probability $\frac{1}{C}$, it proceeds as follows:

- It samples $S_s, S_e \xleftarrow{\$} D_{\mathcal{O}(\sqrt{n}\alpha)}$.

- It computes the commitment message:

$$t = aS_s + S_e - X^c y.$$

- It then computes $c_{aux} \xleftarrow{\$} \mathsf{aCommit}(t; d_{aux})$.
- Finally, the simulator outputs the tuple:

$$(c_{aux}, c, (t, d_{aux}, (S_s, S_e))).$$

From **Theorem** 2.3.1, it follows that if no abort occurs, the distribution of (S_s, S_e) is independent of the actual witness (s, e). Consequently, the simulator's output is computationally indistinguishable from an honest execution of the protocol.

In the event of an abort, indistinguishability is preserved due to the hiding property of the auxiliary commitment scheme aCommit, combined with the fact that the abort decision is made uniformly across all challenge values c.

Special Soundness : Assume we are given two accepting transcripts that share the same commitment:

$$(c_{aux}, c', (t', d'_{aux}, (S'_s, S'_e))) \quad \text{and} \quad (c_{aux}, c'', (t'', d''_{aux}, (S''_s, S''_e))),$$

both of which pass the verifier's checks.

Due to the binding property of the auxiliary commitment scheme aCommit, we can conclude that $t' = t'' =: t$.

Subtracting the corresponding verification equations:

$$(X^{c'} - X^{c''})y = a(S'_s - S''_s) + (S'_e - S''_e),$$

we isolate the secret using multiplication by the inverse of $(X^{c'} - X^{c''})$:

$$2y = a \cdot \frac{2(S'_s - S''_s)}{X^{c'} - X^{c''}} + \frac{2(S'_e - S''_e)}{X^{c'} - X^{c''}} =: 2a\hat{s} + 2\hat{e}.$$

Consequently, we derive a witness $(\hat{s}, \hat{e})$ for the relation $y = a\hat{s} + \hat{e}$.

Moreover, we can upper bound the norm of $2\hat{\mathbf{s}}$ as follows:

$$\|2\hat{s}\| \leq \|S'_s - S''_s\| \cdot \sqrt{n} \left\| \frac{2}{X^{c'} - X^{c''}} \right\| \leq \mathcal{O}(n^2\alpha),$$

where the second inequality leverages Lemma 2.4.1 . A symmetric bound holds for $\hat{e}$ as well.

High-entropy Commitments : This conclusion is an immediate consequence of the cryptographic security guarantees provided by the auxiliary commitment scheme used in the protocol. Specifically, the commitment's *binding* property ensures that it is computationally infeasible to open a given commitment to two different values. In the context of the protocol, this prevents a malicious prover from producing multiple accepting transcripts with the same commitment but different responses, thereby enforcing the uniqueness of the underlying message t. As such, the soundness of the extraction argument fundamentally relies on the assumption that the auxiliary commitment scheme resists forgery and equivocation, ensuring the integrity of the transcript structure.

□

As established in Section 4, both the *completeness* and the *knowledge error* of the protocol can be made negligible under appropriate parameter settings. In particular, when the dimension n exceeds the square of the number of possible challenges C, i.e., when $n > C^2$, the probability that an honest prover is rejected (completeness error) and the chance that a cheating prover succeeds without knowing the witness (knowledge error) both diminish to negligible levels. This follows from the structure of the challenge space and the statistical guarantees provided by the rejection sampling and soundness extraction techniques. As a result, the protocol remains both correct and robust, assuming this mild asymptotic condition on n.

5.2 Linking Lattice and Group-Based Primitives via Zero Knowledge

Another important line of work constructs proofs that a lattice-based object and a classical (discrete-log) object encode the same secret. In the hybrid group-signature context of Benhamouda et al., for example, one needs to show that a Pedersen commitment (in a prime-order group) and a Ring-LWE ciphertext hide the same message. They handle this by "running in parallel" in the two domains: the message m is parsed as a polynomial for the RLWE scheme and as individual scalar values for the commitments. By mimicking polynomial multiplication through exponent additions, their protocol uses the same plaintext-knowledge Σ-protocol to prove both that c_{RLWE} decrypts to m and that each Pedersen commitment is to the corresponding coefficient of m. In effect, the verifier learns that the two primitives commit to identical messages, even though one is in the ring and one is in the group.

Del Pino et al. [dLS19] take a related approach for verifiable encryption. They first open a Pedersen commitment to the plaintext and randomness of an FHE/RLWE ciphertext, and then prove consistency via an inner-product argument. Concretely, given a ciphertext c and a group commitment $C = g^m h^r$, they prove in ZK that c encrypts m with randomness r. Their proof links polynomial multiplications in

the ciphertext to dot-product computations in the commitment domain, enabling a very concise "same plaintext" proof. They further note that this hybrid proof can achieve classical soundness: breaking the proof implies solving a discrete-log problem, so the scheme is even "somewhat future-proof" despite relying on RLWE for confidentiality.

In our setting, the problem is analogous. We have a (lattice-based) ciphertext column and a (classical group-based) commitment matrix whose columns were committed entry-wise. To prove that a given ciphertext column is included (i.e. encrypts one of the committed columns), we can adapt the above approaches. Concretely, one commits each column entry with a discrete-log commitment, and then runs a Σ-protocol that simultaneously "decrypts" the RLWE column and verifies each entry matches the corresponding commitment. This is essentially the same cross-primitive linkage as in [29] and [40]: simulate the polynomial algebra in the exponent space of the group commitments, so that the verifier learns the plaintext is the same on both sides. Such protocols extend the classical trick of proving an ElGamal ciphertext and a Pedersen commitment share the same underlying value into the lattice regime, using the recent toolkit of lattice Σ-protocols:contentReference[oaicite:18]index=18:contentReference[oaicite:19]index=19.

5.2.1 Proof Construction

In their work, Benhamouda et al. [BCK+14] also showed how to prove the plain-text equality between a Ring-LWE based encryption *NTRU* and a classical primitive *Pedersen Commitment*. In what follows, we demonstrate how the foundational protocol introduced in Section 5.1 facilitates the linkage between number-theoretic constructions and lattice-based primitives through the lens of zero-knowledge proofs of knowledge. Specifically, we provide a mechanism to establish, in zero-knowledge, that the messages embedded in Pedersen commitments correspond precisely to those encrypted under a semantically secure instantiation of NTRU encryption.

It is important to emphasize that the cryptographic scheme employed for encryption is not fundamental to the protocol itself. The construction is agnostic to the choice of encryption primitive and can be adapted seamlessly to other schemes, such as classical NTRU [HPS98] or Ring-LWE-based encryption schemes [LPR10], by modifying appropriate parameters.

Let $m \in \{0,1\}^n$ denote the plaintext, and consider its NTRU encryption defined over the ring R_q as:

$$y = hs + pe + m,$$

where $h, s, e \in R_q$, and q is a sufficiently large modulus. Let $p > 2n^2$ be chosen such that it is co-prime to q. For commitment purposes, let $\tilde{g}$ and $\tilde{h}$ be group generators serving as Pedersen commitment param-

eters, chosen as described in Section 2.5. Then, for each $i \in \{0, \ldots, n-1\}$, we define the commitments:

$$cmt_i = m_i \cdot \tilde{g} + r_i \cdot \tilde{h},$$

where m_i is the i-th coefficient of m, and the order of both $\tilde{g}$ and $\tilde{h}$ is $q > 2n^2$. Let $cmt = [cmt]_{i=0}^{n-1}$ and $\mathbf{r} = [r_0\, r_1\, \ldots r_{n-1}]$.

Protocol 5.1.1 is then employed to prove, in a zero-knowledge fashion, that the ciphertext and the Pedersen commitments encode consistent plaintext data. That is, the protocol assures the verifier that the prover knows a message m whose encryption yields $2y$, and whose coefficients appear within the committed values $2cmt$. Importantly, the coefficients must be bounded in magnitude by p to ensure validity and consistency. The correctness of this correspondence implies that the encryption and commitment encodings are harmonized at the level of individual message components. This enables cross-primitive zero-knowledge proofs where encrypted and committed values can be verified for equality without revealing the underlying message.

Zero-Knowledge Protocol for Plaintext Equality between NTRU and Pedersen Commitment

Prover (y, cmt, m)		Verifier (y, cmt)
$r_s, r_e \xleftarrow{\$} D_{O(\sqrt{n}\alpha)}$		
$(m'_0, \ldots, m'_{n-1}) = m' \xleftarrow{\$} D_{O(\sqrt{n})}$		
$\mathbf{r}' = (r'_0, \ldots, r'_{n-1}) \xleftarrow{\$} \mathbb{Z}_q^n$		
$t = hr_s + pr_e + m'$		
$\tilde{t}_i = m_i \cdot \tilde{g} + r'_i \cdot \tilde{h}$		
$\tilde{t} \leftarrow (\tilde{t}_0, \ldots, \tilde{t}_{n-1})$		
Commit: $C_{aux} = \mathrm{Com}(t, \tilde{t}; d_{aux})$	$\xrightarrow{C_{aux}}$	
		$c \leftarrow \{0, \ldots, 2n-1\}$
	$\xleftarrow{c}$	
$S_s = r_s + X^c s$		
$S_e = r_e + X^c e$		
$S_m = m' + X^c m$		
$S_r = r' + X^c r$		
Accept with probability $\frac{D_{O(\sqrt{3n}\alpha)}((S_s, S_e, S_m))}{CD_{(X^c s, X^c e, X^c m), O(\sqrt{3n}\alpha)}((S_s, S_e, S_m))}$		
	$\xrightarrow[S_s, S_e, S_m, S_r]{t, \tilde{t}, d_{aux}}$	
		Check:
		$X^c y + t \stackrel{?}{=} hS_s + pS_e + S_m$
		$X^c cmt + \tilde{t} \stackrel{?}{=} S_m \cdot \tilde{g} + S_r \cdot \tilde{h}$
		$C_{open}(t, \tilde{t}, C_{aux}; d_{aux}) \stackrel{?}{=} accept$
		$\|S_s\|, \|S_e\| \leq \mathcal{O}(n\alpha)$
		$\|S_m\| \leq \mathcal{O}(n)$

5.2.2 Security Analysis

Theorem 5.2.1. *Protocol 5.2.1 is an HVZK Σ'-protocol for the following relations:*

$$\mathcal{R} = \left\{ \left((\tilde{g}, \tilde{h}(cmt_i)_{i=0}^{n-1}, h, p, y), (m, s, e, (r_i)_{i=0}^{n-1}) \right) : \begin{array}{l} y = hs + pe + m \\ \wedge \bigwedge_{i=0}^{n-1} cmt_i = m_i \tilde{g} + r_i \tilde{h} \\ \wedge \|m\|_\infty \leq 1 \wedge \|s\|, \|e\| \leq \mathcal{O}(\sqrt{n\alpha}) \end{array} \right\},$$

$$\mathcal{R}' = \left\{ \left((\tilde{g}, \tilde{h}, (cmt_i)_{i=0}^{n-1}, h, p, y), (m, s, e, (r_i)_{i=0}^{n-1}) \right) : \begin{array}{l} 2y = 2hs + 2pe + 2m \\ \wedge \bigwedge_{i=0}^{n-1} 2\widetilde{cmt}_i = (2m \bmod q)_i \tilde{g} + 2r_i \tilde{h} \\ \wedge \|2m\|_\infty \leq 2n^2 \wedge \|2s\|, \|2e\| \leq \mathcal{O}(n^2\alpha) \end{array} \right\}.$$

where $(2m \bmod q)_i$ *is the* i^{th} *coefficient of* $2m \in R_p$*. The protocol has a knowledge error of* $1/(2n)$ *and a completeness error of* $1 - 1/C$*.*

Furthermore, if for the auxiliary commitment a commitment does not only bind the user to a message , but also to the opening information, the protocol has quasi-unique responses and high entropy commitments.

Proof. Completeness :

$$hS_s + pS_e + S_m = hr_s + X^c \cdot hs + pr_e + X^c \cdot pe + m' + X^c \cdot m = X^c y + t$$

Also we have

$$(S_m \cdot \tilde{g} + S_r \cdot \tilde{h}) = (X^c m + m') \cdot \tilde{g} + (X^c \mathbf{r} + \mathbf{r}') \cdot \tilde{h} = X^c (m \cdot \tilde{g} + \mathbf{r} \cdot \tilde{h}) + (m' \cdot \tilde{g} + \mathbf{r}' \cdot \tilde{h}) = X^c cmt + \tilde{t}$$

For the norms we have that $\|S_s\| \leq \|r_s\| + \|s\| \leq \mathcal{O}(n\alpha)$ and similarly $\|S_m\| \leq \mathcal{O}(n)$.

Knowledge Soundness :

We design a knowledge extractor that rewinds the protocol with same (t, t'). Let $(t, \tilde{t}, d_{aux}, c, S_s, S_e, S_m, S_r)$ and $(t, \tilde{t}, d_{aux}, c', S'_s, S'_e, S'_m, S'_r)$ be two accepting transcripts. Then we have

$$X^c y + t = hS_s + pS_e + S_m \tag{5.1}$$

$$X^{c'} y + t = hS'_s + pS'_e + S'_m \tag{5.2}$$

By 5.1 − 5.2 we get:

$$(X^c - X^{c'})y = h(S_s - S'_s) + p(S_e - S'_e) + (S_m - S'_m)$$

$$\implies 2y = h \cdot 2(X^c - X^{c'})^{-1}(S_s - S'_s) + p \cdot 2(X^c - X^{c'})^{-1}(S_e - S'_e) + 2(X^c - X^{c'})^{-1}(S_m - S'_m)$$

$$\implies 2y = hs'' + pe'' + m''$$

Where:

$$s'' = 2(X^c - X^{c'})^{-1}(S_s - S'_s)$$
$$e'' = 2(X^c - X^{c'})^{-1}(S_e - S'_e)$$
$$m'' = 2(X^c - X^{c'})^{-1}(S_m - S'_m)$$

Again Let $R = 2(X^c - X^{c'})^{-1}(S_r - S'_r)$

Then $m'' \cdot \tilde{g} \ + \ R \cdot \tilde{h}$

$= 2(X^c - X^{c'})^{-1}(S_m - S_{m'}) \cdot \tilde{g} \ + \ 2(X^c - X^{c'})^{-1}(S_r - S_{r'}) \cdot \tilde{h}$

$= 2cmt$

As $(X^c \cdot cmt + \tilde{t}) - (X^{c'} \cdot cmt + \tilde{t}) = (S_m - S_{m'}) \cdot \tilde{g} \ + \ (S_r - S_{r'}) \cdot \tilde{h}$

$\implies 2cmt = 2\frac{S_m - S_{m'}}{X^c - X^{c'}} \cdot \tilde{g} \ + \ 2\frac{S_r - S_{r'}}{X^c - X^{c'}} \cdot \tilde{h}.$

Honest Verifier Zero-Knowledge : Property, we design the following simulator:

After receiving challenge c from the verifier with probability $\frac{1}{C}$ the simulator samples $S_s, S_e, S_r \xleftarrow{\$} D_{\mathcal{O}(\sqrt{n}\alpha)}$ and $S_m \xleftarrow{\$} D_{\mathcal{O}(\sqrt{n})}$ and computes the following

1. Set $t = hS_s + pS_e + S_m - X^c y$
2. Set $\tilde{t} = (S_r \cdot \tilde{g}, S_r \cdot PK + S_m \cdot \tilde{g}) - X^c ct$
3. Set $C_{aux} = Com(t, \tilde{t}; duax)$

Given a challenge value c, the simulator outputs the tuple $(Commit(0), c, \perp)$ with probability $1 - \frac{1}{C}$. With probability $\frac{1}{C}$, the simulator proceeds as above and outputs $(C_{aux}, (t, \tilde{t}, d_{aux}, S_s, S_e, S_m, S_r))$.

According to Theorem 2.3.1, when no abort occurs the distribution of $S_e,\ S_s,\ S_r,\ S_m$ is independent of the witness values s, e, which ensures that the simulation is computationally indistinguishable from a real execution. In the event of an abort, indistinguishability is still preserved by the hiding property of the **Commit** function and the uniform abort probability across all possible challenge values c.
Also note that $||2m|| \leq \mathcal{O}(n^2\alpha)$ and has similarity with the parameters with **Theorem 5.1.1**.

□

Chapter 6

Problem Statement and Our Contribution

6.1 Problem Statement

In this work, we focus on two primary problem statements, the first of which involves constructing a zero-knowledge proof of plain-text equality between ciphertexts produced by two fundamentally different cryptographic schemes: the *NTRU* encryption system, which is based on the presumed hardness of problems over ideal lattices and is considered a strong candidate for post-quantum security, and the classical *ElGamal* encryption scheme, which derives its security from the computational hardness of the discrete logarithm problem in cyclic groups. The core challenge lies in proving, in a zero-knowledge fashion, that two ciphertexts—originating from these structurally and algebraically distinct schemes—encrypt the same plain-text, without revealing any information about the message itself. This not only requires careful handling of the inherent differences in the underlying algebraic structures but also serves a broader goal: enabling secure and verifiable interoperability between pre-quantum and post-quantum systems, which is vital for facilitating smooth migration paths in hybrid cryptographic infrastructures.

The second problem we address pertains to the construction of a zero-knowledge proof of knowledge of a plain-text encrypted under a post-quantum cryptographic scheme, specifically the *NTRU* encryption system, while simultaneously demonstrating that this plain-text is committed to within a vector commitment. The vector commitment scheme employed here is grounded in the hardness of the discrete logarithm problem, a foundational assumption in classical cryptography. Consequently, this problem statement operates in a hybrid cryptographic environment, wherein primitives from both post-quantum and classical domains are employed in tandem. The crux of this challenge lies in reconciling the algebraic and structural disparities between the two frameworks, and in constructing a zero-

knowledge argument that certifies the consistency of the committed and encrypted messages without leaking any information about the underlying plain-text. This forms a critical component of cryptographic transition strategies, allowing for verifiable interoperability across diverse security assumptions during the shift toward post-quantum secure infrastructures.

The second problem statement constitutes the primary focus of this work and encapsulates the core technical challenge that we aim to address. In the following, we formally describe the cryptographic setting and the problem structure. Let $\mathcal{M} = [M_0\ M_1\ \ldots\ M_{N-1}]$ be a collection of N plaintext messages, where $N = \mathcal{O}(\lambda)$ and λ denotes the security parameter of the scheme. Each column vector M_j is defined over the ring $\mathbb{Z}_p$ and has dimension n, that is, $M_j \in \mathbb{Z}_p^n$. To maintain consistent notation, we write $M_j \in \mathbb{M}$, where

$$\mathbb{M} \subseteq \left\{\mathbf{x} \in R_q\ :\ \|\mathbf{x}\|_\infty < p\right\},$$

with parameters p, q satisfying $2p < q$, and both being prime. Since $n < N$, the matrix $\mathcal{M}$ is characterized as a *fat matrix* of dimension $n \times N$.

Let $k \in \{0, 1, \ldots, N-1\}$ be an index of interest. The encryption of the k^{th} column of $\mathcal{M}$ under the *NTRU* cryptosystem is given by:

$$y_k = hs + pe + M_k,$$

where h denotes the public encryption key, and s, e are short secret vectors known only to the prover. The ciphertext y_k is publicly known to both the prover and the verifier.

In addition to this ciphertext, both parties have access to a scalar *cmt* which serves as a commitment to the entire message matrix $\mathcal{M}$. The commitment is constructed using a classical *Pedersen Vector Commitment* scheme, relying on the hardness of the discrete logarithm problem in a cyclic group.

To generate this commitment, we begin by sampling a random vector $\mathbf{x} \xleftarrow{\$} \mathbb{Z}_p^n$, and compute the projection of each message column M_j along $\mathbf{x}$:

$$\left[\langle M_0, \mathbf{x}\rangle\ \ \langle M_1, \mathbf{x}\rangle\ \ \ldots\ \ \langle M_{N-1}, \mathbf{x}\rangle\right].$$

Note that while the vector $\mathbf{x}$ may be revealed, it must be freshly sampled for every new commitment instance to ensure security. Then, another vector $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_p^n$ is sampled uniformly at random, and the scalar $\langle \mathbf{r}, \mathbf{x}\rangle$ acts as the randomizer in the commitment.

We assume a cyclic group $\mathbb{G}$ of prime order p, and let

$$(\tilde{g}_0,\ \tilde{g}_1,\ \ldots\ ,\ \tilde{g}_{N-1}\ ;\ \tilde{h})$$

be the public parameters associated with the *Pedersen Vector Commitment*. The final commitment is then

computed as:

$$cmt = \sum_{j=0}^{N-1} \langle M_j, \mathbf{x} \rangle \cdot \tilde{g}_j + \langle \mathbf{r}, \mathbf{x} \rangle \cdot \tilde{h}.$$

Thus, both y_k and cmt are available to the verifier. The prover is now required to construct a zero-knowledge proof demonstrating two things: first, that they know the plaintext corresponding to the ciphertext y_k; and second, that this same plaintext is indeed included as the k^{th} component in the committed matrix $\mathcal{M}$. This problem elegantly combines a post-quantum encryption primitive (*NTRU*) with a classical commitment primitive (based on discrete logarithms), and its resolution serves as a significant step toward realizing hybrid zero-knowledge proof systems that enable secure coexistence of classical and quantum-resistant cryptographic techniques.

6.2 Our Contribution

In this work, we address fundamental challenges that arise in cryptographic hybrid environments, where classical and post-quantum primitives coexist. We design, analyze, and implement zero-knowledge protocols that establish *plaintext equality* between cryptographic objects defined over fundamentally different algebraic structures—namely, those based on lattice assumptions (like NTRU) and those grounded in discrete logarithm-based constructions (such as Pedersen commitments and ElGamal encryptions). Our key contributions are summarized as follows:

1. **Zero-Knowledge Proof of Plaintext Equality between NTRU and ElGamal Ciphertexts.**
   We construct a protocol that proves, in zero-knowledge, that a message encrypted under NTRU is equal to that encrypted under the classical ElGamal scheme. The core idea is to leverage the linear structure of the plaintext spaces and show consistency between the randomness used in ElGamal and the decryption of the NTRU ciphertext. Despite the disparity between the ring-based and group-based domains, our construction maintains privacy and achieves computational soundness without revealing the plaintext.

2. **Hybrid Zero-Knowledge Framework.**
   Our protocols demonstrate the feasibility of building composable zero-knowledge proofs that operate across heterogeneous cryptographic settings. By enabling zero-knowledge interoperability between lattice-based and discrete-log-based primitives, our work provides practical tools for the cryptographic migration process, where legacy and post-quantum systems must coexist securely.

3. **Batch Zero-Knowledge Proofs.**
   We extend our first construction to a batch setting, where the prover can simultaneously prove the equality of multiple ciphertexts to the columns of a committed matrix, using a single Pedersen vector commitment. This batching mechanism significantly improves communication efficiency and allows the protocol to scale to higher-dimensional data without linear growth in proof size.

4. **Rigorous Security Analysis and Formalization.**
   For all our constructions, we provide formal proofs of completeness, special soundness, and (special) honest-verifier zero-knowledge. Our analysis accounts for rejection sampling and hybrid soundness models, ensuring that the protocols adhere to the established definitions of Σ-protocols even in extended or relaxed settings.

5. **Novel Use of Compression via Inner Product Commitments.**
   To reduce communication overhead and maintain structural soundness, we introduce a compression technique that projects committed matrices into vectors via public inner products. This enables the prover to commit efficiently while still maintaining the binding and hiding properties necessary for security.

To the best of our knowledge, our work presents the first concrete and fully specified protocols for zero-knowledge proofs of plaintext equality between NTRU ciphertexts and both Pedersen commitments and ElGamal encryptions. The techniques developed here serve as important building blocks for realizing post-quantum secure systems that require hybrid cryptographic guarantees.

Chapter 7

Zero-Knowledge Proof of Plain-text Equality between NTRU and ElGamal

7.1 Proof Construction

In this section, we present a protocol for proving the equality of plaintexts encrypted under two fundamentally different cryptographic systems: one based on lattices and the other on cyclic groups. We choose the **NTRU** as our lattice-based encryption scheme, leveraging its additive homomorphic property, which allows for efficient operations on encrypted data. For the group-based encryption, we use the classic **ElGamal** scheme, interpreted in its additive homomorphic form for compatibility with our proof technique. Although **ElGamal** is traditionally multiplicative in nature, its additive variant can be realized by encoding messages appropriately within the group structure.

Our aim is to construct a non-interactive zero-knowledge proof (or a similar efficient argument system) that shows the plaintext encrypted under **ElGamal** is identical to the plaintext encrypted under **NTRU**, without revealing the underlying message. We note that the selection of **NTRU** is not essential to the protocol; it serves as a concrete example of a lattice-based scheme. The approach can be generalized to work with other lattice-based encryption mechanisms, such as the standard version of **NTRU** or schemes based on Ring-LWE, provided they support additive homomorphism. This flexibility makes the protocol broadly applicable in hybrid cryptographic systems where both lattice and group-based components coexist.

Let $y = hs + pe + m \in R_q$ represent an **NTRU** encryption of a message $m \in \mathbb{M}$, where $p > 2n^2$ is an integer coprime with q. Here, h is the public key, s and e are small random polynomials, and R_q denotes the ring of polynomials modulo q. Additionally, let $\tilde{g}$ denote the generator parameter for the **ElGamal** encryption scheme, where the underlying group has order q, satisfying $q > p$.

For each $i = 0, 1, \ldots, n-1$, define the ElGamal ciphertext components as

$$ct_i = (r_i \cdot \tilde{g},\ r_i \cdot \mathsf{PK} + m_i \cdot \tilde{g}),$$

where $r_i \xleftarrow{\$} \mathbb{Z}_q$ are random scalars, PK is the ElGamal public key, and m_i are the coefficients of the message m. The full ElGamal ciphertext is then given by

$$ct = (ct_0, ct_1, \ldots, ct_{n-1}).$$

The following protocol enables a prover to demonstrate, in zero knowledge, that the NTRU ciphertext y and the ElGamal ciphertext ct are consistent—that is, they encrypt the same underlying message. More precisely, the protocol assures the verifier of the following:

- The prover knows the plaintext corresponding to the scaled NTRU ciphertext $2y$.
- Each coefficient of the message is strictly less than p, ensuring proper encoding and bounds.
- The plaintext encrypted in the scaled ElGamal ciphertext $2ct$ is identical to that in $2y$, hence the ciphertexts are consistent.

This allows for secure verification of cross-scheme message consistency between lattice-based and group-based encryptions without revealing the plaintext.

Zero-Knowledge Protocol for Plaintext Equality between NTRU and ElGamal

Prover (y, ct, m)		Verifier (y, ct)
$r_s, r_e \xleftarrow{\$} D_{O(\sqrt{n}\alpha)}$		
$(m'_0, \ldots, m'_{n-1}) = m' \xleftarrow{\$} D_{O(\sqrt{n})}$		
$\mathbf{r}' = (r'_0, \ldots, r'_{n-1}) \xleftarrow{\$} \mathbb{Z}_q^n$		
$t = hr_s + pr_e + m'$		
$\tilde{t}_i = (r'_i \cdot \tilde{g}, r'_i \cdot PK + m'_i \cdot \tilde{g})$		
$\tilde{t} \leftarrow (\tilde{t}_0, \ldots, \tilde{t}_{n-1})$		
Commit: $C_{aux} = \mathrm{Com}(t, \tilde{t}; d_{aux})$	$\xrightarrow{C_{aux}}$	
	$\xleftarrow{c}$	$c \leftarrow \{0, \ldots, 2n-1\}$
$S_s = r_s + X^c s$		
$S_e = r_e + X^c e$		
$S_m = m' + X^c m$		
$S_r = \mathbf{r}' + X^c \mathbf{r}$		
Accept with probability $\frac{D_{O(\sqrt{3n}\alpha)}((S_s, S_e, S_m))}{CD_{(X^c s, X^c e, X^c m), O(\sqrt{3n}\alpha)}((S_s, S_e, S_m))}$		
	$\xrightarrow[S_s, S_e, S_m, S_r]{t, \tilde{t}; d_{aux}}$	
		Check:
		$X^c y + t \stackrel{?}{=} hS_s + pS_e + S_m$
		$X^c ct + \tilde{t} \stackrel{?}{=} (S_r \cdot \tilde{g}, S_r \cdot PK + S_m \tilde{g})$
		$C_{open}(t, \tilde{t}, C_{aux}; d_{aux}) \stackrel{?}{=} accept$
		$\|S_s\|, \|S_e\| \leq \tilde{\mathcal{O}}(n\alpha)$
		$\|S_m\| \leq \tilde{\mathcal{O}}(n\alpha)$

7.2 Security Analysis

The completeness is straight forward,

$hS_s + pS_e + S_m = hr_s + X^c \cdot hs + pr_e + X^c \cdot pe + m' + X^c \cdot m = X^c y + t$. Also we have $(S_r \cdot \tilde{g},\ S_r \cdot PK + S_m \tilde{g}) = (r'\tilde{g},\ r'PK + m'\tilde{g}) + X^c \cdot (r\tilde{g},\ rPK + m\tilde{g}) = X^c ct + \tilde{t}$. For the norms we have that $\|S_s\| \leq \|r_s\| + \|s\| \leq \tilde{\mathcal{O}}(n\alpha)$ and similarly $\|S_m\| \leq \tilde{\mathcal{O}}(n\alpha)$.

For *Knowledge Soundness* we design a knowledge extractor that rewinds the protocol with same (t, t'). Let $(t, \tilde{t}, d_{aux}, c, S_s, S_e, S_m, S_r)$ and $(t, \tilde{t}, d_{aux}, c', S'_s, S'_e, S'_m, S'_r)$ be two accepting transcripts. Then we have

$$X^c y + t = hS_s + pS_e + S_m \tag{7.1}$$

$$X^{c'} y + t = hS'_s + pS'_e + S'_m \tag{7.2}$$

By 7.1 − 7.2 we get:
$(X^c - X^{c'})y = h(S_s - S'_s) + p(S_e - S'_e) + (S_m - S'_m)$
$\implies 2y = h \cdot 2(X^c - X^{c'})^{-1}(S_s - S'_s) + p \cdot 2(X^c - X^{c'})^{-1}(S_e - S'_e) + 2(X^c - X^{c'})^{-1}(S_m - S'_m)$
$\implies 2y = hs'' + pe'' + m''$
Where:

$$s'' = 2(X^c - X^{c'})^{-1}(S_s - S'_s)$$
$$e'' = 2(X^c - X^{c'})^{-1}(S_e - S'_e)$$
$$m'' = 2(X^c - X^{c'})^{-1}(S_m - S'_m)$$

Again Let $R = 2(X^c - X^{c'})^{-1}(S_r - S'_r)$

Then $(R \cdot \tilde{g}, R \cdot PK + m''\tilde{g})$
$= (2(X^c - X^{c'})^{-1}(S_r - S'_r) \cdot \tilde{g}, 2(X^c - X^{c'})^{-1}(S_r - S'_r) \cdot PK + 2(X^c - X^{c'})^{-1}(S_m - S'_m) \cdot \tilde{g})$
$= 2ct$

As $(X^c \cdot ct + t) - (X^{c'} \cdot ct + t) = ((S_r - S'_r) \cdot \tilde{g}, (S_r - S'_r) \cdot PK + (S_m - S'_m) \cdot \tilde{g})$

$\implies 2ct = (2\frac{S_r - S'_r}{X^c - X^{c'}} \cdot \tilde{g}, 2\frac{S_r - S'_r}{X^c - X^{c'}} \cdot PK + 2\frac{S_m - S'_m}{X^c - X^{c'}} \cdot \tilde{g})$.

To show that the protocol satisfies *Honest Verifier Zero-Knowledge* property, we design the following simulator:

After receiving challenge c from the verifier with probability $\frac{1}{C}$ the simulator samples $S_s, S_e, S_r \xleftarrow{\$} D_{\mathcal{O}(\sqrt{n}\alpha)}$ and $S_m \xleftarrow{\$} D_{\mathcal{O}(\sqrt{n})}$ and computes the following

1. Set $t = hS_s + pS_e + S_m - X^c y$
2. Set $\tilde{t} = (S_r \cdot \tilde{g}, S_r \cdot PK + S_m \cdot \tilde{g}) - X^c ct$
3. Set $C_{aux} = Com(t, \tilde{t}; duax)$

Given a challenge value c, the simulator outputs the tuple $(Commit(0), c, \perp)$ with probability $1 - \frac{1}{C}$. With probability $\frac{1}{C}$, the simulator proceeds as above and outputs $(C_{aux}, (t, \tilde{t}, d_{aux}, S_s, S_e, S_m, S_r))$.

According to Theorem 2.3.1, when no abort occurs the distribution of $S_e,\ S_s,\ S_r,\ S_m$ is independent of the witness values s, e, which ensures that the simulation is computationally indistinguishable from a real execution. In the event of an abort, indistinguishability is still preserved by the hiding property of the **Commit** function and the uniform abort probability across all possible challenge values c.

Also note that $||2m|| \leq \tilde{\mathcal{O}}(n^2\alpha)$ and has similarity with the parameters with **Theorem 4.2** of [BCK+14]. Thus we arrive at the following,

Theorem 7.2.1. *The above protocol is a* ***Honest Verifier Zero Knowledge*** Σ*-protocol for the following relations:*

$$\mathcal{R} = \left\{ (\tilde{g},\ ct,\ h,\ p,\ y), (m,\ s,\ e,\ r) \ : \ \begin{array}{l} y = hs + pe + m \ \wedge \ \bigwedge_{i=0}^{n-1} ct_i = (r_i\tilde{g},\ r_iPK + m_i\tilde{g}) \\ \wedge \ \|m\|_\infty \leq p \ \wedge \ \|s\|, \|e\| \leq \tilde{\mathcal{O}}(\sqrt{n}\alpha) \end{array} \right\},$$

$$\mathcal{R}' = \left\{ (\tilde{g},\ ct,\ h,\ p,\ y), (m,\ s,\ e,\ r) \ : \ \begin{array}{l} 2y = 2hs + 2pe + 2m \ \wedge \ \bigwedge_{i=0}^{n-1} 2ct_i = (2r_i\tilde{g},\ 2r_iPK + 2m_i\tilde{g}) \ (mod\ q) \\ \wedge \ \|2m\|_\infty \leq 2n^2 \ \wedge \ \|2s\|, \|2e\| \leq \tilde{\mathcal{O}}(n^2\alpha) \end{array} \right\}.$$

Chapter 8

Proving the Knowledge of k^{th} Column and Its Inclusion in the Commitment

8.1 Proof Construction

In this section, we present a sigma protocol for proving the witness plaintext behind a *NTRU* cipher which is the encryption of the k^{th} column of a collection of N-plaintexts residing in matrix $\mathcal{M}$ and $n < N = \mathcal{O}(\lambda)$. Also we need to prove the involvement of the mentioned plain-text for a existing commitment cmt of $\mathcal{M}$, committed using the technique of section 3. In this case we must be careful about not leaking any information about the other columns of $\mathcal{M}$. On the other hand we keep the compression vector $\mathbf{x}$ and the vector L public as referred in 3.
We note that the selection of *NTRU* is not essential to the protocol; it serves as a concrete example of a lattice-based scheme. The approach can be generalized to work with other lattice-based encryption mechanisms, such as the standard version of *NTRU* or schemes based on Ring-LWE, provided they support additive homomorphism. This flexibility makes the protocol broadly applicable in hybrid cryptographic systems where both lattice and group-based components coexist.

Let our plain-text space be $\mathbb{M} = \{m \in R_q \ : \ \|m\|_\infty < p\}$, where q and p are primes and $2p < q$. Let $y_k = hs + pe + M_k \in R_q$ represent an *NTRU* encryption of a message M_k, which is the k^{th} column of $\mathcal{M}$. Here, h is the public key, s and e are small random polynomials, and R_q denotes the ring of polynomials modulo q.

Let $cmt = \sum_{j=0}^{N-1} l_j \cdot \tilde{g}_j \ + \ \langle \mathbf{r}, \mathbf{x} \rangle \tilde{h}$, where

- $l_j = \langle M_j, \mathbf{x} \rangle$ for $j = 0, \ldots, N-1$,
- vector $\mathbf{r} = [r_0 \ \ \ldots \ \ r_{n-1}] \xleftarrow{\$} \mathbb{Z}_p^n$,
- $(\tilde{g}_0, \ \ldots \ , \ \tilde{g}_{N-1}; \ \tilde{h})$ are public parameters of *Pedersen Vector Commitment*

- and the vector $\mathbf{x}$ is randomly chosen from $\mathbb{Z}_p^n$.

Then *cmt* is the commitment of $\mathcal{M}$. Further let $L = [L_0 \ \ldots \ L_{n-1}]$ be public vector, where $L_i = \sum_{j=0}^{N-1} m_{i,j} \cdot \tilde{g}_j \ + \ r_i \cdot \tilde{h}$ i.e L_i is the *Pedersen Vector Commitment* of the i^{th} row of $\mathcal{M}$. We need to prove the knowledge of M_k in both y_k and *cmt*.

The central intuition underpinning the proof lies in the random sampling of a matrix

$$\mathcal{M}' = [M'_0 \ M'_1 \ \ldots \ M'_{N-1}],$$

where each column M'_j is drawn uniformly at random from $\mathbb{Z}_p^n$, for all $j \in 0, 1, \ldots, N-1$. The prover then furnishes an *NTRU* encryption of M'_k alongside a commitment to the matrix $\mathcal{M}'$, compressed with respect to the same vector $\mathbf{x}$. Upon receiving a challenge vector $\mathbf{x}' \xleftarrow{\$} \mathbb{Z}_p^n$ from the verifier, the prover configures certain responses so that the verifier can validate whether the sum of the k^{th} compressed plaintexts—computed over the two vectors $\mathbf{x}$ and $\mathbf{x}'$—is equal to the sum of the corresponding compressed ciphertexts. In parallel, the verifier can also ascertain whether the combined commitments of $\mathcal{M}$ and $\mathcal{M}'$ equate to the sum of the commitment of one designated responses and that of another.

The following protocol enables a prover to demonstrate, in zero knowledge, that the*NTRU* ciphertext y_k and the commitment *cmt* are consistent—that is, the encryption contains the same underlying message which was included in the commitment. More precisely, the protocol assures the verifier of the following:

- The prover knows the plaintext corresponding to the scaled *NTRU* ciphertext y_k.
- The prover knows the secret vector $\mathbf{x}$, used to compress the matrix $\mathcal{M}$.
- Each coefficient of the message is strictly less than p, ensuring proper encoding and bounds.
- The plaintext included in the commitment *cmt* is identical to that in y_k.

Zero-Knowledge Protocol

$\mathcal{P}_{s,e,\mathbf{x},\mathbf{r}}(y_k, cmt, M_k)$ | $\mathcal{V}(y_k, cmt)$

$s', e' \xleftarrow{\$} D_{O(\sqrt{n}\alpha)}$

$M'_0, \ldots, M'_{N-1} \xleftarrow{\$} D_{O(\sqrt{n}p)}$

$\mathbf{r}' = (r'_0, \ldots, r'_{n-1}) \xleftarrow{\$} \mathbb{Z}_p^n$

$t = hs' + pe' + M'_k$

$l'_j = \langle M'_j, \mathbf{x} \rangle \; ; \; j = 0, \ldots, N-1$

$\tilde{t} \leftarrow \sum_{j=0}^{N-1} l'_j \cdot \tilde{g}_j \; + \; \langle \mathbf{r}', \mathbf{x} \rangle \tilde{h}$

SomeCommit: $C_{aux} = \mathrm{Com}(t, \tilde{t}; d_{aux})$ $\xrightarrow{C_{aux}}$

$\mathbf{x}' \leftarrow \mathbb{Z}_p^n$

$\xleftarrow{\mathbf{x}'}$

$S = \langle hs, \mathbf{x}' \rangle \; + \; \langle hs', \mathbf{x} \rangle$

$E = \langle e, \mathbf{x}' \rangle \; + \; \langle e', \mathbf{x} \rangle$

$\mathsf{M}_j = \left(\langle M_j, \mathbf{x}' \rangle \; + \; \langle M'_j, \mathbf{x} \rangle \right) \bmod \mathrm{p} \; ; \; j = 0, \ldots, N-1$

$\mathsf{M} \leftarrow [\mathsf{M}_0 \; \ldots \; \mathsf{M}_{N-1}]$

$R = (\langle \mathbf{r}, \mathbf{x}' \rangle \; + \; \langle \mathbf{r}', \mathbf{x} \rangle) \bmod \mathrm{p}$

Accept with probability
$\frac{D_{O(\sqrt{n}p)}(\mathsf{M})}{CD_{(\{\langle M_j, \mathbf{x}' \rangle\}_{j=0}^{N-1}), O(\sqrt{n}p)}(\mathsf{M})}$

$\xrightarrow[S,E,\mathsf{M},R]{t,\tilde{t};d_{aux}}$

Check:

$C_{open}(t, \tilde{t}, C_{aux}; d_{aux}) \stackrel{?}{=} accept$

$\langle y_k, \mathbf{x}' \rangle \; + \; \langle t, \mathbf{x} \rangle \stackrel{?}{=} S + pE + \mathsf{M}_k$

$cmt + \tilde{t} \stackrel{?}{=} \sum_{j=0}^{N-1} \mathsf{M}_j \cdot \tilde{g}_j \; + \; R\tilde{h} \; + \; \langle L, \mathbf{x} - \mathbf{x}' \rangle$

$\|\mathsf{M}\| < \tilde{\mathcal{O}}(\sqrt{N}p)$

8.2 Security Analysis

The completeness is straight forward,

1. $S + pE + \mathsf{M}_k$
   $= \langle hs, \mathbf{x}' \rangle \; + \; \langle hs', \mathbf{x} \rangle \; + \; p\langle e, \mathbf{x}' \rangle \; + \; p\langle e', \mathbf{x} \rangle \; + \; \langle M_k, \mathbf{x}' \rangle \; + \; \langle M'_k, \mathbf{x} \rangle$
   $= \langle hs \; + \; pe \; + \; M_k, \mathbf{x}' \rangle \; + \; \langle hs' \; + \; pe' \; + \; M'_k, \mathbf{x} \rangle$
   $= \langle y_k, \mathbf{x}' \rangle \; + \; \langle t, \mathbf{x} \rangle$
2. $\sum_{j=0}^{N-1} \mathsf{M}_j \cdot \tilde{g}_j \; + \; R\tilde{h} \; + \; \langle L, \mathbf{x} - \mathbf{x}' \rangle$

$$
\begin{aligned}
&= \sum_{j=0}^{N-1} \Big(\langle M_j, \mathbf{x}' \rangle \ + \ \langle M'_j, \mathbf{x} \rangle \Big) \cdot \tilde{g}_j \ + \ (\langle \mathbf{r}, \mathbf{x}' \rangle \ + \ \langle \mathbf{r}', \mathbf{x} \rangle)\, \tilde{h} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \\
&= \sum_{j=0}^{N-1} \langle M_j, \mathbf{x} + \mathbf{x}' - \mathbf{x} \rangle \cdot \tilde{g}_j \ + \ \langle \mathbf{r}, \mathbf{x} + \mathbf{x}' - \mathbf{x} \rangle \tilde{h} \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \\
&= \sum_{j=0}^{N-1} \langle M_j, \mathbf{x} \rangle \cdot \tilde{g}_j \ + \ \langle \mathbf{r}, \mathbf{x} \rangle \tilde{h} \ + \ \sum_{j=0}^{N-1} \langle M_j, \mathbf{x}' - \mathbf{x} \rangle \cdot \tilde{g}_j \ + \ \langle \mathbf{r}, \mathbf{x}' - \mathbf{x} \rangle \tilde{h} \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \\
&= cmt \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \ + \ \sum_{j=0}^{N-1} \langle M_j, \mathbf{x}' - \mathbf{x} \rangle \cdot \tilde{g}_j \ + \ \langle \mathbf{r}, \mathbf{x}' - \mathbf{x} \rangle \tilde{h} \\
&= cmt \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \ + \ \sum_{j=0}^{N-1} \left(\sum_{i=0}^{n-1} m_{i,j} \cdot (x'_i - x_i) \right) \tilde{g}_j \ + \ \sum_{i=0}^{n-1} r_i \cdot (x'_i - x_i) \tilde{h} \\
&= cmt \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \ + \ \sum_{i=0}^{n-1} \left(\sum_{j=0}^{N-1} m_{i,j} \cdot \tilde{g}_j \ + \ r_i \tilde{h} \right) (x'_i - x_i) \\
&= cmt \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \ + \ \sum_{i=0}^{n-1} L_i \cdot (x'_i - x_i) \\
&= cmt \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \ + \ \langle L, \mathbf{x}' - \mathbf{x} \rangle \\
&= cmt \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \ - \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \\
&= cmt \ + \ \tilde{t}
\end{aligned}
$$

3. Each $\mathsf{M}_j < p$ as $\mathsf{M}_j = \Big(\langle M_j, \mathbf{x}' \rangle \ + \ \langle M'_j, \mathbf{x} \rangle \Big) \bmod \mathrm{p}$. Thus $\|\mathsf{M}\| \ < \sqrt{Np^2} < \mathcal{O}(\sqrt{N}p)$.

Theorem 8.2.1. *The protocol above has computational* $2n$*-special soundness.*

Proof. Let $\left(t, \tilde{t}, \mathbf{x}^1, S^1, E^1, R^1, \mathsf{M}^1\right)$ and $\left(t, \tilde{t}, \mathbf{x}^{1'}, S^{1'};, E^{1'}, R^{1'}, \mathsf{M}^{1'}\right)$ be two accepting transcripts. Then from the check equation $\langle y_k, \mathbf{x}' \rangle \ + \ \langle t, \mathbf{x} \rangle \stackrel{?}{=} S + pE + \mathsf{M}_k$ we get,

$$\langle y_k, \mathbf{x}^1 \rangle \ + \ \langle t, \mathbf{x} \rangle = S^1 + pE^1 + \mathsf{M}_k^1 \quad \text{and}$$

$$\langle y_k, \mathbf{x}^{1'} \rangle \ + \ \langle t, \mathbf{x} \rangle = S^{1'} + pE^{1'} + \mathsf{M}_k^{1'}$$

Subtracting the above equations we get, $\langle y_k, \mathbf{x}^1 - \mathbf{x}^{1'} \rangle = (S^1 - S^{1'}) + p(E^1 - E^{1'}) + (\mathsf{M}_k^1 - \mathsf{M}_k^{1'})$. Now $\mathsf{M}_k^1 - \mathsf{M}_k^{1'} = \langle M_k, \mathbf{x}^1 - \mathbf{x}^{1'} \rangle$. Then we have one equation on $m_{i,k}$ for $i = 0, 1, \ldots, n-1$ i.e

$$m_{0,k}(x_0^1 - x_0^{1'}) \ + \ m_{1,k}(x_1^1 - x_1^{1'}) \ + \ \ldots \ + \ m_{n-1,\,k}(x_{n-1}^1 - x_{n-1}^{1'}) = \mathsf{M}_k^1 - \mathsf{M}_k^{1'} \tag{8.1}$$

Let $\mathbf{z}^1 \ = \ \mathbf{x}^1 - \mathbf{x}^{1'}$. Then the knowledge extractor can rewind over $n-1$ more challenge pairs $\{(\mathbf{x}^2, \mathbf{x}^{2'}), \ \ldots \ , \ (\mathbf{x}^n, \mathbf{x}^{n'})\}$ in such a way that $\{\mathbf{z}^1, \ \mathbf{z}^2, \ \ldots \ , \ \mathbf{z}^n\}$ are linearly independent. Thus we get a system of linear equations as follows

$$[\mathbf{z}^1 \ \ \mathbf{z}^2 \ \ \ldots \ \ \mathbf{z}^n]^T \cdot M_k = [\mathsf{M}_k^1 - \mathsf{M}_k^{1'} \ \ \ldots \ \ \mathsf{M}_k^n - \mathsf{M}_k^{n'}]^T$$

Since $\{\mathbf{z}^1, \ \mathbf{z}^2, \ \ldots \ , \ \mathbf{z}^n\}$ is a set of linearly independent vectors, the matrix $[\mathbf{z}^1 \ \ \mathbf{z}^2 \ \ \ldots \ \ \mathbf{z}^n]^T$ has n independent rows and is invertible. Thus the above system of linear equation is solvable i.e

$$M_k = \left([\mathbf{z}^1 \ \ \mathbf{z}^2 \ \ \ldots \ \ \mathbf{z}^n]^T \right)^{-1} \cdot [\mathsf{M}_k^1 - \mathsf{M}_k^{1'} \ \ \ldots \ \ \mathsf{M}_k^n - \mathsf{M}_k^{n'}]^T$$

.

Thus the knowledge extractor can extract M_k with $2n$ many queries to to the provers. Consequently this ensures that $\mathsf{M_k}$ is correctly computed and contains information of real M_k. Also the check $cmt + \tilde{t} \stackrel{?}{=} \sum_{j=0}^{N-1} \mathsf{M}_j \cdot \tilde{g}_j + R\tilde{h} + \langle L, \mathbf{x} - \mathbf{x}' \rangle$ ensures the involvement of M_k in cmt. Furthermore with these $2n$ many queries the knowledge extractor can extract M_j's for $j \in \{0, \ldots, N-1\} \setminus \{k\}$.

In similar way $\mathbf{r}$ can also be extracted from $2n$ many queries. After the extraction of $\mathcal{M}$ and $\mathbf{r}$, knowledge extractor can check $cmt \stackrel{?}{=} \sum_{j=0}^{N-1} \langle M_j, \mathbf{x} \rangle \cdot \tilde{g}_j + \langle \mathbf{r}, \mathbf{x} \rangle \cdot \tilde{h}$ in the end.

□

Theorem 8.2.2. *The above protocol has perfect special honest verifier zero-knowledge.*

Proof. For this we design a simulator $\mathcal{S}$, given a challenge value $\mathbf{x}'$ $\mathcal{S}$ outputs $(C_{aux}, \mathbf{x}', \perp)$ with probability $1 - 1/C$. With probability $1/C$, $\mathcal{S}$ proceeds as follows : $\mathcal{S}$ samples $S, E \xleftarrow{\$} \mathbb{Z}_q$ and $R \xleftarrow{\$} \mathbb{Z}_p$ and computes the following

- Let $\alpha = S + pE + \mathsf{M}_k - \langle y_k, \mathbf{x}' \rangle$. Sample $t_0, t_1, \ldots, t_{n-2} \xleftarrow{\$} \mathbb{Z}_q$ and set $t_{n-1} = \frac{1}{x_{n-1}} \cdot \left(\alpha - \sum_{i=0}^{n-2} t_i \cdot x_i \right)$. Finally set $t = [t_0\ t_1\ \ldots\ t_{n-1}]$.
- $\tilde{t} = \sum_{j=0}^{N-1} \mathsf{M}_j \cdot \tilde{g}_j + R\tilde{h} + \langle L, \mathbf{x} - \mathbf{x}' \rangle - cmt$,
- $C_{aux} \leftarrow \mathrm{SomeCommit}(t, \tilde{t}; d_{aux})$.

The simulator $\mathcal{S}$ then outputs $(C_{aux}, \mathbf{x}', (t, \tilde{t}, d_{aux}, S, E, R, \mathsf{M}))$.

Clearly the simulator outputs are correctly computed and they will pass the checks by the verifier as well. Owing to the rejection sampling technique, the distribution of M remains independent of $\mathcal{M}$, and furthermore, the variables S and E are statistically independent of s and e. Consequently, the simulated and real protocol transcripts are computationally indistinguishable. In scenarios involving aborts, indistinguishability is ensured by the hiding property of **SomeCommit** and the observation that aborts occur with equal probability for every $\mathbf{x}'$.

□

8.3 Proof Size

The proof size of a zero knowledge proof refers to the amount of data measured in bits generated by the prover to demonstrate a statement is true to the verifier, without revealing any additional information. A smaller proof size is desirable as it reduces communication and verification overhead. For our zero-knowledge protocol 5.1.1 the prover sends $(C_{aux}, t, \tilde{t}, d_{aux}, S, E, R, \mathsf{M})$. All of these except t and M can be expressed by $log\ p$ many bits each. Now t can be expressed using $nlog\ q$ many bits. Note that each M_j is in $\mathbb{Z}_p$ and can be expressed by $log\ p$ many bits and hence prover needs to send $Nlog\ p$ many bits

to express M. As $N \geq n$, thus the proof size is dominated by the size of M and finally it is bounded by $\mathcal{O}(Nlog\ p)$. Since p fixed for a $NTRU$, the proof size becomes linear.

Chapter 9

Batch Proof

9.1 Proof Construction

In this section we will prove the knowledge of multiple plaintexts along with their involvement in the commitment. We would show this for k *NTRU* plaintexts $\{y_{c_1}, y_{c_2}, \ldots, y_{c_k}\}$ for $c_1, c_2, \ldots, c_k \in \{0, \ldots, N-1\}$.

Let $\mathcal{M} = [M_0\, M_1\, \ldots\, M_{N-1}]$ be the collection of N plaintexts, where each $M_j \in \mathbb{M}$ for $j = 0, \ldots, N-1$. Let $y_{c_i} = h^{(c_j)}s^{(c_j)} + pe^{(c_j)} + M_{c_j}$ for $j = 1, 2, \ldots, k$ be k *NTRU* ciphers for $\{c_1, \ldots c_k\}$-th columns of $\mathcal{M}$ respectively and $s^{(c_j)}, e^{(c_j)}$ are short vectors. The reason that each column is encrypted under different public keys is if all the columns are encrypted under same public key, then selling decryption key for one column would give away the decryption key for other columns as well.

Let *cmt* be the commitment of $\mathcal{M}$ as described before and let $\mathbf{x}$ be the compression vector. Let $L = [L_0\, \ldots\, L_{n-1}]^T$ where L_i is the *Pedersen Vector Commitment* of the i^{th} row of $\mathcal{M}$ for $i = 0, \ldots, n-1$.

We use the additive homomorphic property of *NTRU* to send the knowledge of the k plaintexts behind $y_{c_1}, \ldots, y_{c_k}$ along with their involvement in the commitment *cmt*. The proof goes as follows

Zero-Knowledge Protocol for Batch Proving

$\mathcal{P}\left(\{y_{c_1},\ldots,y_{c_k}\}, cmt, \{M_{c_1},\ldots,M_{c_k}\}\right)$ $\qquad$ $\mathcal{V}(\{y_{c_1},\ldots,y_{c_k}\}, cmt)$

$s'^j, e'^j \xleftarrow{\$} D_{O(\sqrt{n}\alpha)}$, for $j = c_1, c_2, \ldots, c_k$

$M'_0, \ldots, M'_{N-1} \xleftarrow{\$} D_{O(\sqrt{n}p)}$

$\mathbf{r}' = (r'_0, \ldots, r'_{n-1}) \xleftarrow{\$} \mathbb{Z}_p^n$.

$t_{c_j} = h^{(c_j)} s'^{(c_j)} + p e'^{(c_j)} + M'_{c_j}$, for $j = 1, 2, \ldots, k$

$l'_j = \langle M'_j, \mathbf{x} \rangle \; ; \; j = 0, \ldots, N-1$

$\tilde{t} \leftarrow \sum_{j=0}^{N-1} l'_j \cdot \tilde{g}_j \; + \; \langle \mathbf{r}', \mathbf{x} \rangle \tilde{h}$

SomeCommit: $C_{aux} = \mathrm{Com}(t_{c_1}, \ldots, t_{c_k}, \tilde{t}; d_{aux})$ $\xrightarrow{C_{aux}}$

$\mathbf{x}' \leftarrow \mathbb{Z}_p^n$

$\xleftarrow{\mathbf{x}'}$

$S = \langle \sum_{j=1}^{k} h^{(c_j)} s^{(c_j)} \, , \, \mathbf{x}' \rangle \; + \; \langle \sum_{j=1}^{k} h^{(c_j)} s'^{(c_j)} \, , \, \mathbf{x} \rangle$

$E = \langle \sum_{j=1}^{k} e^{(c_j)} \, , \, \mathbf{x}' \rangle \; + \; \langle \sum_{j=1}^{k} e'^{(c_j)} \, , \, \mathbf{x} \rangle$

$\mathsf{M}_j = \left(\langle M_j, \mathbf{x}' \rangle \; + \; \langle M'_j, \mathbf{x} \rangle \right) \bmod \mathrm{p} \; ; \; j = 0, \ldots, N-1$

$\mathsf{M} \leftarrow [\mathsf{M}_0 \;\; \ldots \;\; \mathsf{M}_{N-1}]$

$R = (\langle \mathbf{r}, \mathbf{x}' \rangle \; + \; \langle \mathbf{r}', \mathbf{x} \rangle) \bmod \mathrm{p}$

Accept with probability
$\frac{D_{O(\sqrt{n}p)}(\mathsf{M})}{CD_{(|\langle M_j, \mathbf{x}' \rangle|_{j=0}^{N-1}), O(\sqrt{n}p)}(\mathsf{M})}$

$\xrightarrow[S,E,\mathsf{M},R]{t_{c_1},\ldots,t_{c_k},\tilde{t},d_{aux}}$

Check:

$C_{open}(t_{c_1}, \ldots, t_{c_k}, \tilde{t}, C_{aux}; d_{aux}) \stackrel{?}{=} accept$

$\langle \sum_{j=1}^{k} y_{c_j}, \mathbf{x}' \rangle + \langle \sum_{j=1}^{k} t_{c_j}, \mathbf{x} \rangle \stackrel{?}{=} S + pE + \sum_{j=1}^{k} \mathsf{M}_{c_j}$

$cmt + \tilde{t} \stackrel{?}{=} \sum_{j=0}^{N-1} \mathsf{M}_j \cdot \tilde{g}_j \; + \; R\tilde{h} \; + \; \langle L, \mathbf{x} - \mathbf{x}' \rangle$

$\|\mathsf{M}\| \leq \tilde{\mathcal{O}}(\sqrt{N}p)$

9.2 Security Analysis

The completeness is following :

1. $\langle \sum_{j=1}^{k} y_{c_j}, \mathbf{x}' \rangle \; + \; \langle \sum_{j=1}^{k} t_{c_j}, \mathbf{x} \rangle$

$= \sum_{j=1}^{k} h^{(c_j)} s^{c_j} + p e^{(c_j)} + M_{c_j} \, , \, \mathbf{x} \rangle \; + \; \langle \sum_{j=1}^{k} h^{c_j} s'^{c_j} + p e'^{c_j} + M'_{c_j} \, , \, \mathbf{x}' \rangle$

$= \left(\langle \sum_{j=1}^{k} h^{(c_j)} s^{(c_j)}, \mathbf{x}' \rangle + \langle \sum_{j=1}^{k} h^{(c_j)} s'^{(c_j)}, \mathbf{x} \rangle \right) \; + \; p \left(\langle \sum_{j=1}^{k} e^{(c_j)}, \mathbf{x}' \rangle + \langle \sum_{j=1}^{k} e'^{(c_j)}, \mathbf{x} \rangle \right) \; + \; \left(\langle \sum_{j=1}^{k} M_{c_j}, \mathbf{x}' \rangle + \langle \sum_{j=1}^{k} M'_{c_j}, \mathbf{x} \rangle \right)$

$= S \; + \; pE \; + \; \sum_{j=1}^{k} \mathsf{M}_{c_j}$

2. $\sum_{j=0}^{N-1} \mathsf{M}_j \cdot \tilde{g}_j \ + \ R\tilde{h} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle$

$= \sum_{j=0}^{N-1} \Big(\langle M_j, \mathbf{x}' \rangle \ + \ \langle M'_j, \mathbf{x} \rangle \Big) \cdot \tilde{g}_j \ + \ (\langle \mathbf{r}, \mathbf{x}' \rangle \ + \ \langle \mathbf{r}', \mathbf{x} \rangle)\, \tilde{h} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle$

$= \sum_{j=0}^{N-1} \langle M_j, \mathbf{x} + \mathbf{x}' - \mathbf{x} \rangle \cdot \tilde{g}_j \ + \ \langle \mathbf{r}, \mathbf{x} + \mathbf{x}' - \mathbf{x} \rangle \tilde{h} \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle$

$= \sum_{j=0}^{N-1} \langle M_j, \mathbf{x} \rangle \cdot \tilde{g}_j \ + \ \langle \mathbf{r}, \mathbf{x} \rangle \tilde{h} \ + \ \sum_{j=0}^{N-1} \langle M_j, \mathbf{x}' - \mathbf{x} \rangle \cdot \tilde{g}_j \ + \ \langle \mathbf{r}, \mathbf{x}' - \mathbf{x} \rangle \tilde{h} \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle$

$= cmt \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \ + \ \sum_{j=0}^{N-1} \langle M_j, \mathbf{x}' - \mathbf{x} \rangle \cdot \tilde{g}_j \ + \ \langle \mathbf{r}, \mathbf{x}' - \mathbf{x} \rangle \tilde{h}$

$= cmt \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \ + \ \sum_{j=0}^{N-1} \left(\sum_{i=0}^{n-1} m_{i,j} \cdot (x'_i - x_i) \right) \tilde{g}_j \ + \ \sum_{i=0}^{n-1} r_i \cdot (x'_i - x_i) \tilde{h}$

$= cmt \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \ + \ \sum_{i=0}^{n-1} \left(\sum_{j=0}^{N-1} m_{i,j} \cdot \tilde{g}_j \ + \ r_i \tilde{h} \right) (x'_i - x_i)$

$= cmt \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \ + \ \sum_{i=0}^{n-1} L_i \cdot (x'_i - x_i)$

$= cmt \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \ + \ \langle L, \mathbf{x}' - \mathbf{x} \rangle$

$= cmt \ + \ \tilde{t} \ + \ \langle L, \mathbf{x} - \mathbf{x}' \rangle \ - \ \langle L, \mathbf{x} - \mathbf{x}' \rangle$

$= cmt \ + \ \tilde{t}$

3. Each $\mathsf{M}_j < p$ as $\mathsf{M}_j = \Big(\langle M_j, \mathbf{x}' \rangle \ + \ \langle M'_j, \mathbf{x} \rangle \ \Big) \bmod \mathrm{p}$. Thus $\|\mathsf{M}\| \ < \sqrt{Np^2} < \mathcal{O}(\sqrt{N}p)$.

Theorem 9.2.1. *The protocol above has computational* $2n$*-special soundness.*

Proof. Let $\left(t_{c_1}, \ldots, t_{c_k}, \tilde{t}, \mathbf{x}^1, S^1, E^1, R^1, \mathsf{M}^1\right)$ and $\left(t_{c_1}, \ldots, t_{c_k}, \tilde{t}, \mathbf{x}^{1'}, S^{1'};, E^{1'}, R^{1'}, \mathsf{M}^{1'}\right)$ be two accepting transcripts. Then from the check equation $\langle \sum_{j=1}^{k} y_{c_j}, \mathbf{x}' \rangle \ + \ \langle \sum_{j=1}^{k} t_{c_j}, \mathbf{x} \rangle \stackrel{?}{=} S + pE + \sum_{j=1}^{k} \mathsf{M}_{c_j}$ we get,

$$\langle \sum_{j=1}^{k} y_{c_j}, \mathbf{x}^1 \rangle \ + \ \langle \sum_{j=1}^{k} t_{c_j}, \mathbf{x} \rangle = S^1 + pE^1 + \sum_{j=1}^{k} \mathsf{M}^1_{c_j} \quad \text{and}$$

$$\langle \sum_{j=1}^{k} y_{c_j}, \mathbf{x}^{1'} \rangle \ + \ \langle \sum_{j=1}^{k} t_{c_j}, \mathbf{x} \rangle = S^{1'} + pE^{1'} + \sum_{j=1}^{k} \mathsf{M}^{1'}_{c_j}$$

Subtracting the above equations we get, $\langle \sum_{j=1}^{k} y_{c_j}, \mathbf{x}^1 - \mathbf{x}^{1'} \rangle \ = \ (S^1 - S^{1'}) + p(E^1 - E^{1'}) + \sum_{j=1}^{k} (\mathsf{M}^1_{c_j} - \mathsf{M}^{1'}_{c_j})$.

Now $\mathsf{M}^1_{c_j} - \mathsf{M}^{1'}_{c_j} = \langle M_{c_j}, \mathbf{x}^1 - \mathbf{x}^{1'} \rangle$. Then we have one equation on m_{i,c_j} for $i = 0, 1, \ldots, n-1$ i.e

$$m_{0,c_j}(x^1_0 - x^{1'}_0) \ + \ m_{1,c_j}(x^1_1 - x^{1'}_1) \ + \ \ldots \ + \ m_{n-1,\,c_j}(x^1_{n-1} - x^{1'}_{n-1}) = \mathsf{M}^1_{c_j} - \mathsf{M}^{1'}_{c_j} \tag{9.1}$$

Let $\mathbf{z}^1 \ = \ \mathbf{x}^1 - \mathbf{x}^{1'}$. Then the knowledge extractor can rewind over $n-1$ more challenge pairs $\{(\mathbf{x}^2, \mathbf{x}^{2'}), \ \ldots \ , \ (\mathbf{x}^n, \mathbf{x}^{n'})\}$ in such a way that $\{\mathbf{z}^1, \ \mathbf{z}^2, \ \ldots \ , \ \mathbf{z}^n\}$ are linearly independent. Thus we get a system of linear equations as follows

$$[\mathbf{z}^1 \ \ \mathbf{z}^2 \ \ \ldots \ \ \mathbf{z}^n]^T \cdot M_{c_j} = [\mathsf{M}^1_{c_j} - \mathsf{M}^{1'}_{c_j} \ \ \ldots \ \ \mathsf{M}^n_{c_j} - \mathsf{M}^{n'}_{c_j}]^T$$

Since $\{\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^n\}$ is a set of linearly independent vectors, the matrix $[\mathbf{z}^1 \ \ \mathbf{z}^2 \ \ \ldots \ \ \mathbf{z}^n]^T$ has n independent rows and is invertible. Thus the above system of linear equation is solvable i.e

$$M_{c_j} = \left([\mathbf{z}^1 \ \ \mathbf{z}^2 \ \ \ldots \ \ \mathbf{z}^n]^T\right)^{-1} \cdot [\mathsf{M}^1_{c_j} - \mathsf{M}^{1'}_{c_j} \ \ \ldots \ \ \mathsf{M}^n_{c_j} - \mathsf{M}^{n'}_{c_j}]^T$$

.

Thus the knowledge extractor can extract M_{c_j} for $j = 1, 2, \ldots, k$ with $2n$ many queries to to the provers. Consequently this ensures that each M_{c_j} is correctly computed and contains information of real M_{c_j}. Also with these $2n$ many queries the knowledge extractor can extract M_j for $j \in \{0, \ldots, N-1\} \setminus \{c_1, \ldots, c_k\}$ like the above way with these same $2n$ many queries.

In similar manner $\mathbf{r}$ can also be extracted. After extracting $\mathcal{M}$ and $\mathbf{r}$, the knowledge extractor can check the legitimacy by $cmt \stackrel{?}{=} \sum_{j=0}^{N-1} \langle M_j, \mathbf{x} \rangle \cdot \tilde{g}_j + \langle \mathbf{r}, \mathbf{x} \rangle \cdot \tilde{h}$.

□

Theorem 9.2.2. *The above protocol has perfect special honest verifier zero-knowledge.*

Proof. With probability $1/C$, $\mathcal{S}$ proceeds as follows : $\mathcal{S}$ samples $S, E \xleftarrow{\$} \mathbb{Z}_q$ and $R \xleftarrow{\$} \mathbb{Z}_p$ and computes the following

- Let $\alpha = S + pE + \sum_{j=1}^{k} \mathsf{M}_{c_j} - \langle \sum_{j=1}^{k} y_{c_j}, \mathbf{x}' \rangle$. Sample $t_{c_1}, t_{c_2}, \ldots, t_{c_{k-1}} \xleftarrow{\$} \mathbb{Z}_q^n$ and $t_{c_k 0}, t_{c_k 1}, \ldots, t_{c_k (n-2)} \xleftarrow{\$} \mathbb{Z}_q$. Set $t_{c_k (n-1)} = \frac{1}{x_{n-1}} \cdot \left(\alpha - \sum_{j=1}^{k-1} \langle t_{c_j}, \mathbf{x} \rangle - \sum_{i=0}^{n-2} t_{c_k i} \cdot x_i \right)$. Finally set $t_{c_j} = [t_{c_j 0} \ t_{c_j 1} \ \ldots \ t_{c_j n-1}]$ for $j = 1, 2, \ldots k$.
- $\tilde{t} = \sum_{j=0}^{N-1} \mathsf{M}_j \cdot \tilde{g}_j + R\tilde{h} + \langle L, \mathbf{x} - \mathbf{x}' \rangle - cmt$,
- $C_{aux} \leftarrow \mathrm{SomeCommit}(t_{c_1}, \ldots, t_{c_k}, \tilde{t}; d_{aux})$.

The simulator $\mathcal{S}$ then outputs $\left(C_{aux}, \mathbf{x}', (t_{c_1}, \ldots, t_{c_k}, , \tilde{t}, d_{aux}, S, E, R, \mathsf{M})\right)$.

The simulator outputs are correctly computed and are indistinguishable from real protocol transcripts. □

9.3 Proof Size

Each t_{c_j} for $j = 1, 2, \ldots, k$ are in $\mathbb{Z}_q^n$ and can be represented with $\mathcal{O}(nloq\ q)$ many bits and hence $\mathcal{O}(k \cdot nlog\ q)$ many bits in total. On the other hand M requires $\mathcal{O}(N \cdot log\ q)$ many bits to represent. All the other responses $\tilde{t}, S, E, R$ can be represented by $log\ p$ many bits. Thus the proof size is dominated by the size of $t_{c_1}, \ldots, t_{c_k}$ and M. As p, q are fixed, hence the proof size for the batch protocol is

$\mathcal{O}(k \cdot n + N)$.

Note that to prove the knowledge and the involvement of k columns in $\mathcal{M}$ using the protocol in 8.1 we need to run the protocol independently for each column i.e total k independent protocol runs in total. Thus the total proof size becomes $\mathcal{O}(k \cdot N)$. But the batch protocol proves the same with proof size in $\mathcal{O}(k \cdot n + N)$. Clearly the batch proving becomes efficient if $k \cdot n + N < k \cdot N$ i.e $k > \frac{N}{N-n}$.

If $k \cdot n < N$ then the proof size of the batch protocol is dominated by $\mathcal{O}(N)$ i.e it becomes independent of k. Thus to have efficient batch protocol we must have $\frac{N}{N-n} < k < \frac{N}{n}$. This bound is possible when $n < N - n$ i.e $2n < N$.

Chapter 10

Applications and Future Work

10.1 Applications

The protocol proposed in this work has multiple real-world applications, particularly in hybrid cryptographic settings where both group-based and lattice-based primitives coexist. Such hybrid environments are becoming increasingly relevant as cryptographic infrastructures transition toward post-quantum security. Our zero-knowledge proof system provides a mechanism to prove that a message encrypted under a lattice-based encryption scheme (e.g., NTRU) is the same as the message committed using a classical Pedersen vector commitment. Below we explore several domains where this capability is crucial.

10.1.1 Post-Quantum Blockchain Commitments

In blockchain-based systems, users often rely on group-based commitments (such as Pedersen commitments) to hide transaction details while enabling verification of correctness through zero-knowledge proofs. However, existing commitments and protocols are vulnerable to quantum attacks. A natural direction is to adopt post-quantum encryption schemes, such as NTRU or schemes based on Ring-LWE, for future security.

Our protocol enables a user to prove, in zero knowledge, that the plaintext encrypted under a post-quantum encryption scheme is equal to the value committed using a legacy Pedersen vector commitment. This feature is essential for post-quantum migration paths where existing on-chain commitments remain unchanged, but future operations must be post-quantum secure. For example, a privacy coin protocol may commit transaction amounts using a Pedersen vector commitment, but later encrypt these values using NTRU for off-chain auditing or decryption. Our protocol ensures these two representations are consistent without revealing the committed values.

10.1.2 Verifiable Post-Quantum Voting Systems

End-to-end verifiable electronic voting systems require mechanisms that allow voters to prove that their encrypted vote is valid and matches the publicly committed value. Traditionally, this involves group-based encryption and commitments. In a post-quantum setting, votes may be encrypted using a lattice-based encryption scheme, while the commitment mechanism remains group-based for compatibility with existing verifiers.

In this setting, our protocol enables a voter to prove that their encrypted vote (under NTRU) is consistent with a Pedersen commitment published to the bulletin board. The zero-knowledge property ensures that no information about the vote is leaked, while the soundness guarantees that the encrypted and committed values match. This hybrid approach supports post-quantum verifiability while preserving compatibility with existing vote tallying and verification procedures.

10.1.3 Secure Multi-Party Computation and Smart Contracts

In secure multi-party computation (MPC) and zero-knowledge smart contract platforms (e.g., zkSNARK-based systems), parties may use lattice-based encryption for secure computation, while relying on group-based commitments for efficiency and interoperability. Ensuring consistency between encrypted and committed inputs is often required, especially in outsourced computation or zk-rollup settings.

Our protocol enables a party to commit to a dataset using a Pedersen vector commitment, encrypt selected parts using NTRU, and later prove that these encrypted components match the committed ones. This is particularly relevant in privacy-preserving decentralized applications (dApps), where smart contracts verify commitments while off-chain actors perform encrypted computations. The ability to bridge lattice encryption with classical commitments allows seamless integration of quantum-safe cryptography into current blockchain protocols.

10.1.4 Post-Quantum Anonymous Transactions

Privacy-preserving cryptocurrencies often use commitments and encrypted values to hide transaction details while proving correctness of transfers. For example, a user may commit to a coin value and use range proofs or other ZK arguments to prove properties of the transaction without revealing the amount. As these systems move to post-quantum security, encryption of coin values using schemes like NTRU becomes necessary.

Our protocol enables a user to prove that an NTRU-encrypted coin value matches the Pedersen commitment published on-chain. This ensures the encrypted value is authentic and consistent with previously committed data, allowing the system to support post-quantum encrypted transactions while retaining compatibility with existing commitment-based verification mechanisms.

10.1.5 Auditable Encrypted Storage and Integrity Verification

In cloud computing and outsourced data storage, sensitive data is often encrypted before being stored. To ensure data integrity and auditability, systems may require commitments to the encrypted data to be published or logged, e.g., on a blockchain or in a tamper-proof audit log.

Using our protocol, a client can prove that the encrypted data they submitted matches a public commitment without revealing the plaintext. This allows verifiers to confirm data consistency and integrity, even when the encryption is post-quantum secure. The protocol thus enables verifiable encrypted data storage systems where the encryption layer is post-quantum secure, while the audit layer remains group-based for efficiency.

10.1.6 General Utility in Hybrid Cryptography

More broadly, our construction demonstrates how to securely bridge cryptographic primitives from different worlds—namely, lattice-based and discrete-log-based systems—using zero-knowledge. This hybrid compatibility is crucial during the ongoing transition to post-quantum secure systems, where backward compatibility and interoperability are required. The modularity of the protocol allows it to be adapted to other commitment schemes or encryption systems, as long as they retain additive homomorphic properties.

In conclusion, our zero-knowledge protocol provides a critical tool for hybrid environments where lattice-based encryption coexists with group-based commitments. The ability to prove, in zero knowledge, that a post-quantum ciphertext contains the same value as a committed message unlocks a range of applications in privacy, integrity, verifiability, and secure computation. These features make the protocol highly relevant to the future of blockchain systems, voting platforms, secure computation frameworks, and beyond.

10.2 Future Work

While this thesis lays the groundwork for zero-knowledge interoperability in hybrid cryptographic environments, several important avenues remain open for further investigation. We outline below the most promising directions for future research.

1. **Reducing Proof Size for Column Membership Protocols.**
   The protocol presented in Section 8.1 allows a prover to demonstrate that an NTRU ciphertext encrypts a column committed within a Pedersen vector commitment. Although sound and complete, the current construction has proof size linear in the dimension of the message space. A major future direction is to optimize this protocol to achieve *sublinear proof size*, potentially via recursive argument techniques, polynomial commitments, or lattice-based succinct proof systems.

Such a development would greatly enhance the practicality of the protocol in high-dimensional applications like encrypted databases and outsourced computation.

2. **Plaintext Equality between Post-Quantum and Classical Vector Commitments.**
   A crucial next step for hybrid system design is to establish zero-knowledge proofs that a message committed under a *post-quantum vector commitment* (e.g., lattice- or hash-based) is equal to a message committed under a classical *discrete-logarithm-based vector commitment* (e.g., Pedersen). This problem represents a natural analogue of the cross-encryption proofs explored in this thesis and would form an essential cryptographic primitive for seamless and verifiable migration from classical commitments to post-quantum secure alternatives. Constructing such protocols with efficiency, soundness, and zero-knowledge guarantees remains an open and compelling challenge.

3. **Extending to Malicious Verifier Zero-Knowledge.**
   The protocols in this work are primarily designed under the honest-verifier model. A natural extension is to achieve full zero-knowledge against arbitrary (potentially malicious) verifiers. This could be realized through standard transformations such as the Fiat-Shamir heuristic (in the random oracle model) or the use of a common reference string (CRS) model.

4. **Proofs over Fully Homomorphic Ciphertexts.**
   Extending the notion of plaintext equality proofs to more expressive settings—such as fully homomorphic encryption (FHE)—would enable privacy-preserving auditing of outsourced computations across cryptographic domains. Constructing ZK protocols that relate classical and post-quantum FHE ciphertexts is a nontrivial and highly impactful open problem.

5. **Quantum-Resistant Non-Interactive Zero-Knowledge (NIZK) in Hybrid Settings.**
   An important direction is to instantiate our constructions in the non-interactive setting, particularly under quantum-safe assumptions. Developing efficient NIZKs for cross-primitive equality that remain secure in the quantum random oracle model (QROM) would further enhance the deployment feasibility of hybrid systems.

6. **Proof Composition and Universal Composability.**
   Another open problem is the composability of our protocols in larger systems such as secure multi-party computation, verifiable delay functions, or post-quantum voting. Proving that our protocols compose securely under the Universal Composability (UC) framework would strengthen their applicability in complex cryptographic protocols.

Pursuing these directions will not only improve the efficiency and flexibility of the protocols proposed in this work but also contribute to the broader goal of building secure and interoperable cryptographic infrastructures in the post-quantum era.

Bibliography

[AJL+12] Gilad Asharov, Abhishek Jain, Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, and Daniel Wichs. Multiparty computation with low communication, computation and interaction via threshold FHE. pages 483–501, 2012. doi:10.1007/978-3-642-29011-4_29.

[BCK+14] Fabrice Benhamouda, Jan Camenisch, Stephan Krenn, Vadim Lyubashevsky, and Gregory Neven. Better zero-knowledge proofs for lattice encryption and their application to group signatures. pages 551–572, 2014. doi:10.1007/978-3-662-45611-8_29.

[BG93] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. pages 390–420, 1993. doi:10.1007/3-540-48071-4_28.

[BG18] Jonathan Bootle and Jens Groth. Efficient batch zero-knowledge arguments for low degree polynomials. pages 561–588, 2018. doi:10.1007/978-3-319-76581-5_19.

[Bot24] Enrico Bottazzi. Greco: Fast zero-knowledge proofs for valid FHE RLWE ciphertexts formation. Cryptology ePrint Archive, Paper 2024/594, 2024. URL: https://eprint.iacr.org/2024/594.

[Dam00] Ivan Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. pages 418–430, 2000. doi:10.1007/3-540-45539-6_30.

[DF02] Ivan Damgård and Eiichiro Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. pages 125–142, 2002. doi:10.1007/3-540-36178-2_8.

[DGOW95] Ivan Damgård, Oded Goldreich, Tatsuaki Okamoto, and Avi Wigderson. Honest verifier vs dishonest verifier in public cain zero-knowledge proofs. pages 325–338, 1995. doi:10.1007/3-540-44750-4_26.

[dLS19] Rafa'el del Pino, Vadim Lyubashevsky, and Gregor Seiler. Short discrete log proofs for FHE and ring-LWE ciphertexts. pages 344–373, 2019. doi:10.1007/978-3-030-17253-4_12.

[DPSZ12] Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. pages 643–662, 2012. doi:10.1007/978-3-642-32009-5_38.

[FO97] Eiichiro Fujisaki and Tatsuaki Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. pages 16–30, 1997. doi:10.1007/BFb0052225.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. Ntru: A ring-based public key cryptosystem. In Joe P. Buhler, editor, *Algorithmic Number Theory*, pages 267–288, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

[LNSW13] San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. pages 107–124, 2013. doi:10.1007/978-3-642-36362-7_8.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. pages 1–23, 2010. doi:10.1007/978-3-642-13190-5_1.

[Lyu09] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. pages 598–616, 2009. doi:10.1007/978-3-642-10366-7_35.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. pages 738–755, 2012. doi:10.1007/978-3-642-29011-4_43.

[Ped92] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. pages 129–140, 1992. doi:10.1007/3-540-46766-1_9.

[SS11] Damien Stehlé and Ron Steinfeld. Making NTRU as secure as worst-case problems over ideal lattices. pages 27–47, 2011. doi:10.1007/978-3-642-20465-4_4.

[SSTX09] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. Efficient public key encryption based on ideal lattices. pages 617–635, 2009. doi:10.1007/978-3-642-10366-7_36.

[Ste94] Jacques Stern. A new identification scheme based on syndrome decoding. pages 13–21, 1994. doi:10.1007/3-540-48329-2_2.

[XXW13] Xiang Xie, Rui Xue, and Minqian Wang. Zero knowledge proofs from ring-lwe. In Michel Abdalla, Cristina Nita-Rotaru, and Ricardo Dahab, editors, *Cryptology and Network Security*, pages 57–73, Cham, 2013. Springer International Publishing.