Master's thesis

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

Ole Martin Edstrøm

Zero-Knowledge Protocols for proof of Correct Shuffle using Lattices

Master's thesis in Mathematical Sciences
Supervisor: Kristian Gjøsteen
June 2022

NTNU
Kunnskap for en bedre verden

Ole Martin Edstrøm

Zero-Knowledge Protocols for proof of Correct Shuffle using Lattices

Master's thesis in Mathematical Sciences
Supervisor: Kristian Gjøsteen
June 2022

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

Abstract. The main purpose of this thesis is to study zero-knowledge protocols for proof of correct shuffle using lattices. To understand the computations done in the protocols and why the protocols are secure, one chapter is dedicated to studying the ring structure of lattices. We then go over to study two existing protocols to prove a correct shuffle before we give a try to construct a new zero-knowledge protocol for shuffle using permutation matrices.

Sammendrag. Hovedformålet med denne oppgaven er å studere protokoller for kunskapsløse bevis av korrekt ommstokning ved bruk av gitter. For å forstå utregningene gjort i protkollene og hvorfor protokollene er sikre, har vi dedikert et kapittel til å studere ringstrukturen til gittere. Deretter går vi videre til å studere to eksisterende protocoller for bevis av korekt omstokning før vi gir et forsøk på å konstruere et nytt kunskapsløst beves for omstokning ved bruk av permutasjonsmatriser.

Preface

This thesis was written from September 2021 to June 2022 under the supervision of Kristian Gjøsteen and marks the end of a master's in mathematical sciences and my five years as a student at NTNU.

I will, first of all, give a big thanks to Kristian for helping me choose this topic and for meeting with me almost every week for guidance. These meetings have helped me a lot in my intuition of the different securities and where to go next.

I also want to give a big thanks to Linjeforeningen Delta, the student association for my field of study. I want to thank them for giving me a place for social gatherings, a place to grow as a person, and the best five years of my life. I want to thank Eivind, Erling, Håkon, Maxim, Sturla, and Trygve as my mathematics student partners. Finally, I want to thank Magnus Sigurd Lie for proofreading and feedback.

Even tho I did not have enough time to develop any practical new protocols, the subject of zero-knowledge has been very interesting to work on throughout this year. I also hope that I see some further study on protocols of correct shuffle using permutation matrices in the future.

Ole Martin Edstrøm
Trondheim, June 2022

Contents

CHAPTER 1

Introduction

A zero-knowledge protocol is a proof of something you have done or something you know. These protocols are some of the most central things in anonymous electronic voting. One of these zero-knowledge proofs is to give a proof of correct shuffle, and this is essential because of anonymity. We want one party to receive the votes and know who sent them but not be able to see what the vote is. Then we want a second party to be able to see what the votes are and count them, but not be able to know whom each vote is from. However, the second party must know that each vote is from someone, so it knows the first party did not cheat by excluding votes or adding extra. Therefore we let the first party perform a zero-knowledge proof of correct shuffle to prove that the votes the second party receives are the same as the first party received, just in a different order.

When working with cyclic groups and the ElGamal encryption scheme, there are multiple ways to give a zero-knowledge proof of correct shuffle. Furukawa and Sako [**5**], and Furukawa [**6**] give two proofs of shuffle using permutation matrices and that the messages committed to are a multi commitment of the known messages with the permutation matrix applied to them. While Neff [**11**] gives a protocol that proves a shuffle by creating two polynomials, one which has the known messages as roots and another that has the messages committed to as its roots. Then since the permutation of roots in a polynomial does not matter, it holds to show that these polynomials have the same evaluations for a single element.

We will be studying an adaptation of Neff's protocol which works over lattices instead of cyclic groups, which are given and proven secure by Aranha et al. [**1**]. We will also give a small try to adapt the proof of Furukawa and Sako to a protocol over lattices and discuss what theory we would need to do it.

The reason to adopt these protocols to lattices instead of cyclic groups is because of the developments of quantum computers. It has been proven that it is easy to compute the discreet logarithm and factor numbers with a quantum computer, which cyclic group encryption relies on being hard for its security. However, for post-quantum cryptography, encryption schemes over lattices are some of the most promising of being secure. And it is, therefore, wanted to adapt most cryptography schemes to rely on lattice problems.

We start in Chapter 3 to define commitment schemes as done by Baum et al. [**2**] and define the different types of securities we want these schemes to have. Then we define zero-knowledge proofs and Σ-protocols the same way as it is done by Damgård [**3**].

In Chapter 4, we continue by introducing what a lattice is and state some results of the ring structures to certain types of lattices. We also introduce how to define the Gaussian distribution over lattices and give two results showing that the protocols we study will be complete and that one would not leak any information. We will also introduce two knapsack problems over lattices which the different types of securities will rely on. We will also give scenarios where both of these problems can not be solved by any all-powerful adversary.

Chapter 5 is where we will introduce the commitment scheme over lattices that we will be using and prove the correlation between its security and the hardness of the two knapsack problems. We will also give a few protocols and show that they are zero-knowledge proofs.

In Chapter 6, we will study two protocols for a correct shuffle. The first one was given by Aranha et al. [**1**], and then the adaptation of it [**4**] to prove a correct shuffle of encrypted messages given as vectors of lattice points.

Finally, in Chapter 7, we will give an extremely slow protocol to prove that a matrix is a permutation matrix. We will also discuss how we could make a more useful protocol if we had the necessary theory.

CHAPTER 2

Notation

We start by introducing some notations that will be used throughout this thesis.

First of all, we will write vectors and matrices with bold letters where small letters like $\mathbf{b}$ will denote vectors and capital letters like $\mathbf{A}$ will denote matrices, where $\mathbf{0}^{n\times k}$ denote the $n\times k$ zero-matrix and $\mathbf{I}_n$ the n times n identity matrix. If we write $\mathbf{x}\cdot\mathbf{y}$ it will mean the usual inner product of two vectors unless it is stated otherwise.

If we let S be a set, then we denote $x \xleftarrow{\$} S$ as picking a value uniformly random from S and giving it to x. If $\mathcal{S}$ is a probability distribution we denote $x \xleftarrow{\$} \mathcal{S}$ as picking x according to $\mathcal{S}$. Sometimes we will also write $x \leftarrow \mathcal{A}(\text{input})$ to give x the value of the output from an algorithm $\mathcal{A}$, or we write $x \leftarrow 3+4$ to give x the value $3+4$.

We will also have a notation for writing probabilities. To show how we write this we give an example of when have a key generating algorithm KeyGen, an algorithm $\mathcal{A}$ and a set X, then we denote

$$\Pr\left[\mathcal{A}(pk,x)=1 \middle| pk \leftarrow \mathsf{KeyGen}, x \xleftarrow{\$} X\right]$$

the probability that $\mathcal{A}$ in input pk and x outputs 1 given that pk is computed from KeyGen and x chosen uniformly at random from X. If we are doing computations with probabilities it would be nice with a more compact notation and write

$$\Pr_{x\xleftarrow{\$}X}\left[\mathcal{A}(pk,x)=1 | pk \leftarrow \mathsf{KeyGen}\right].$$

Which means exactly the same as above.

CHAPTER 3

Commitments and Zero-Knowledge Proof

We will, in this chapter, introduce what a commitment scheme is and what kind of properties we want from this. After this, we will define what a zero-knowledge proof (which we will shorten to ZK proofs) is. We will also define what it means for a ZK proof to be an honest-verifier zero-knowledge proof (which we write as HVZK proof). Then we will define a special type of HVZK proof which are Σ-protocols. These definitions are what we will be using for the protocols we will study in Chapter 5, 6, and 7.

1. What is a Commitment scheme

The intuition of a commitment scheme is to be able to bind yourself to a message without letting anyone know what the message is. The properties of the commitment scheme will later, in the protocols for ZK proofs, be the reason the protocols' security properties hold.

Definition 3.1. *A commitment scheme for a message space $\mathcal{M}$ is a set of tree algorithms* $\mathsf{KeyGen}, \mathsf{Commit}$ *and* Open *which are defined as follows:*

- KeyGen *is a PPT algorithm that take the security parameter λ as input and returns a public parameter $pk \in \{0,1\}^{poly(\lambda)}$.*
- Commit *is a PPT algorithm that takes as input a public parameter $pk \in \{0,1\}^{poly(\lambda)}$ and a message $m \in \mathcal{M}$. and returns a commitment $c \in \{0,1\}^{poly(\lambda)}$ and $r \in \{0,1\}^{poly(\lambda)}$.*
- Open *is a deterministic polynomial-timed algorithm that takes as input a public parameter $pk \in \{0,1\}^{poly(\lambda)}$, a commitment $c \in \{0,1\}^{poly(\lambda)}$, $r \in \{0,1\}^{poly(\lambda)}$ and a message $m \in \mathcal{M}$ and returns a bit $b \in \{0,1\}$.*

If $pk = \mathsf{KeyGen}()$, then $\mathsf{Open}(pk, \mathsf{Commit}(pk, m), m) = 1$.

A commitment scheme in itself would not be very useful. We also want some additional properties. The first one of these is the hiding property which can be described as if we have two messages in the message space and a commitment to one of them, an adversary would not be able to determine which message the commitment is related to.

Definition 3.2. *We say that a commitment scheme is ϵ-hiding if for any algorithm $\mathcal{A}$ that is given $pk \leftarrow \mathsf{KeyGen}()$, and for any two messages $m_0 \neq m_1$ chosen by $\mathcal{A}$*

$$\Pr\left[\mathcal{A}(pk, m_0, m_1, c) = b \;\middle|\; b \xleftarrow{\$} \{0,1\},\ c, r \leftarrow \mathsf{Commit}(pk, m_b)\right] \leq \epsilon + \frac{1}{2}.$$

If we let $\mathcal{A}$ be any all-powerful algorithm, we say that the commitment scheme is statistically hiding. However, if we restrict $\mathcal{A}$ to be PPT, we say it is computationally hiding.

The second property we want a commitment scheme to have is the binding property which is that given a public parameter pk, it is hard to find two messages which can have the same commitment c.

Definition 3.3. *We say a commitment scheme is ϵ-binding if given any algorithm the following holds*

$$\Pr\left[\begin{array}{c} \mathcal{A}(pk) = (m, m', r, r', c) \text{ s.t. } m \neq m', \\ \mathsf{Open}(pk, c, r, m) = \mathsf{Open}(pk, c, r', m') = 1 \end{array} \;\middle|\; pk \leftarrow \mathsf{KeyGen}()\right] \leq \epsilon.$$

Here we also say that the commitment scheme is statistically binding if we let $\mathcal{A}$ be any all-powerful algorithm and computational binding if we restrict it to be PPT.

We now give two trivial examples of commitment schemes. One that will be statistically binding and not at all hiding and one that will be the opposite.

Example 3.4. *We define* $\mathsf{KeyGen}()$ *just to return a random bit string pk. and* $\mathsf{Comit}(pk, m) = (c, r)$ *where $c = m$ and r just a random bit string. Then* $\mathsf{Open}(pk, c, r, m)$ *returns* 1 *if and only if $c = m$.*

We can easily see that this is not at all hiding since the commitment is the message, but it is binding since we can not find another message equal to the commitment.

Example 3.5. *We define the commitment scheme for a message space $\mathcal{M} = \{0,1\}^\lambda$ with* $\mathsf{KeyGen}()$ *just the same as last time. It returns a random bit string to pk. But now we define* $\mathsf{Comit}(pk, m) = (c, r)$ *where $r \xleftarrow{\$} \mathcal{M}$ and $c = m \otimes r$, where $\otimes$ is the xor operation. Then* $\mathsf{Open}(pk, c, r, m)$ *returns* 1 *if and only if $m = c \otimes r$.*

In this commitment scheme, we see that it is impossible to break the hiding property since any message can be related to a commitment. But the scheme will not at all be binding, since the algorithm can given a commitment c just pick two different messages m and m' and compute $r = c \otimes m$ and $r' = c \otimes m'$.

These two examples of commitment schemes are useless, but they show that a commitment scheme can be either hiding or binding without it being the other at all. When we introduce the commitment scheme in Chapter 5, we will see that the scheme will have a stronger or weaker hiding or binding property depending on the size we let the set where we pick our r to have.

2. Zero-Knowledge proofs and Σ-protocols

A ZK proof of something is a conversation between a prover and a verifier, where the prover wants to prove to the verifier that it has some knowledge, and the

verifier wants to be sure that it can not be convinced by the prover, if the prover does not have the knowledge. One example we will have of this in Chapter 5 is a proof of having a commitment to an unknown or known message. But now we start by defining what a ZK proof is.

Definition 3.6. *A zero-knowledge proof with soundness error ϵ of a relation $R \subseteq X \times W$ for sets X and W, is an interactive protocol Π between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. Both $\mathcal{P}$ and $\mathcal{V}$ are given public input x, and at the end of Π, $\mathcal{V}$ will output accept or reject. We also want a ZK proof to have the following properties.*

- *Completeness: If $\mathcal{P}$ knows a witness $w \in W$ such that $(x, w) \in R$ and follows Π honestly, $\mathcal{V}$ will output accept with overwhelming probability.*
- *Soundness: For any prover $\mathcal{P}^*$ that can get $\mathcal{V}$ to accept for a given input x with probability $\epsilon' > \epsilon$, there exists a constant c and an algorithm $\mathcal{E}$ called the extractor that can with black-box access to $\mathcal{P}^*$ produce a witness w' such that $(x, w') \in R$ with at most*

$$\frac{|x|^c}{\epsilon' - \epsilon}$$

steps. Access to $\mathcal{P}^*$ is counted as one step.

We take some time to discuss the two properties we want the ZK proof to have. The completeness property is pretty straightforward, we want an honest conversation to work. The not so obvious property is the soundness property, but the intuition is that it should be easier to find a witness w' such that $(x, w') \in R$ than to convince the verifier without knowing a witness. Notice also that the witness w' extracted by $\mathcal{E}$ does not need to be the same as the one $\mathcal{P}^*$ knows, but the soundness shows that $\mathcal{P}^*$ knows one.

Sometimes it is enough to have a ZK proof, but we would like the proofs to have an additional property. We want the verifier to learn nothing about the witness that the prover knows.

Definition 3.7. *We say that a zero-knowledge proof Π is honest-verifier zero-knowledge if there exists a PPT algorithm $\mathcal{A}$ that given $x \in X$ can simulate an accepting transcript of Π that is statistically indistinguishable from an honest transcript of Π.*

We can also define Σ-protocols, which is another form of an interactive protocol between two algorithms. We define Σ-protocols because we later will see that Σ-protocols are HVZK proofs, and it is usually easier to prove that something is a Σ-protocol than to prove it is an HVZK proof.

Definition 3.8. *Let $R \subseteq X \times W$ be a relation for sets X and W. If we have an interactive protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ who gets $x \in X$ as input and $\mathcal{P}$ additionally gets $(x, w) \in R$ as input and the interaction goes in the following way.*

1. *$\mathcal{P}$ sends a message α to $\mathcal{V}$.*
2. *$\mathcal{V}$ sends a challange $\beta \in \{0,1\}^t$ to $\mathcal{P}$.*

3. *$\mathcal{P}$ sends an answer γ to $\mathcal{V}$.*
4. *$\mathcal{V}$ accepts or rejects depending on x, α, β and γ.*

Then we call this protocol a Σ-protocol if the following tree properties holds.

- *Completeness: If $\mathcal{P}$ and $\mathcal{V}$ follow the protocol for input $x \in X$ and privet input w for $\mathcal{P}$ where $(x, w) \in R$, $\mathcal{V}$ will output accept with overwhelming probability.*
- *Special soundness: For any input x with a pair of accepting conversations (α, β, γ) and $(\alpha', \beta', \gamma')$, where $\beta \neq \beta'$, one can efficiently compute a witness w' such that $(x, w') \in R$.*
- *Honest-Verifier Zero Knowledge: There exist a PPT simulator $\mathcal{A}$ that on input $x \in X$ and and a $\beta \in \{0,1\}^t$ can output an accepting conversation (α, β, γ) that is statistically indistinguishable from a honest conversation between $\mathcal{P}$ and $\mathcal{V}$.*

The name Σ-protocol comes from the way you can draw up the conversation if we include one step at the start when the prover sends x over to the verifier. Then the protocol looks like a Σ.

We notice that in the security sense, the only difference between a Σ-protocol and an HVZK proof of knowledge is the special soundness and soundness property. Furthermore, what we will see in the following theorem is that a Σ-protocol is an HVZK proof of knowledge.

Theorem 3.9. *Let Π be a Σ-protocol for a relation $R \subseteq X \times W$, where the length of the challenge bit-string β is $t > 2$. Then Π is an HVZK proof of knowledge for R with soundness error 2^{-t+2}.*

The soundness error of this theorem is proven by Damgård [**3**] that can be decreased to 2^{-t}, but since we will work with $t \gg 2$, we only prove it for 2^{-t+2}.

Proof. We see that the completeness and the honest-verifier zero-knowledge follow since it is the same defined properties.

Let $\mathcal{P}^*$ be a prover that can get $\mathcal{V}$ to accept for a given x with probability $\epsilon' > 2^{-t+2}$. Let $\mathbf{H}$ be the binary matrix where we have a row for each α that $\mathcal{P}^*$ can send, and a column for each challenge β. An element of $\mathbf{H}$ is 1 if $\mathcal{P}^*$ is able to send an accepting answer and 0 otherwise. Now we see that because of the special soundness property, if we find two 1's in a single row, then we will be able to extract a witness w'. The idea of the extractor $\mathcal{E}$ we construct is that it will ask $\mathcal{P}^*$ for a message α and give it a random challenge β until it hits a 1 in $\mathbf{H}$, then $\mathcal{E}$ will continue with the same α, but different challenges until it hits a second 1.

We know that $\mathbf{H}$ contains $1/\epsilon'$ number of 1's and therefore $\mathcal{E}$ will hit a 1 with $\mathcal{O}(1/\epsilon')$ expected calls, but we do not know if a row with a 1 will contain a second 1. So if $\mathcal{E}$ hits a row with a single 1 in it, it will never finish. We, therefore, define a row to be heavy if more than $\epsilon'/2$ of the elements in it are 1. Since $\epsilon' > 2^{-t+2}$, it is easy to see that a heavy row will contain at least two 1's. Let $\mathbf{H}'$ be the submatrix

of $\mathbf{H}$ of all rows that are not heavy, h' be the number of elements in $\mathbf{H}'$ and h the number of elements of $\mathbf{H}$. Then the number of 1's in $\mathbf{H}$ will be $h\epsilon'$ by assumption and the number of 1's in $\mathbf{H}'$ will be less then $h'\epsilon'/2$. If we now let g be the number of 1's in heavy rows, we get that

$$g \geq h\epsilon' - h'\epsilon'/2 > h\epsilon' - h\epsilon'/2 = h\epsilon'/2,$$

which means that more than half of the 1's is contained in heavy rows. Therefore $\mathcal{E}$ will, with more than $1/2$ in probability, hit a heavy row when it hits its first 1.

Because of the size of the challenge space $\mathbf{H}$ will have 2^t columns and $\mathcal{E}$ will for each β it sends after hitting a heavy row have provability $\frac{\epsilon' 2^{t-1}-1}{2^t}$ hitting another 1. We then have that in a heavy row, $\mathcal{E}$ will hit a second 1 in expected T callings, where

$$T = \frac{2^t}{\epsilon' 2^{t-1} - 1} \leq \frac{4}{\epsilon'}.$$

So to find the second 1, $\mathcal{E}$ will also be expected to need $\mathcal{O}(1/\epsilon')$ steps.

We now define how $\mathcal{E}$ will work:

1. Probe random elements of $\mathbf{H}$ until a 1 is found.
2. In parallel, continues with the two following steps and stop if either of the two stops:
   i. Try a different element of the row where the first 1 was found until a second 1 is found.
   ii. Pick a random element of $\mathbf{H}$ and a random element from $\{1, ..., d\}$. Stop if both were 1.
3. If 2. stopped because of i. extract w', if 2. stopped because of ii. then return to step 1.

We see that the chance of 2. stopping because of ii. is ϵ'/d. This means that we want to pick a d such that if $\mathcal{E}$ finds the first 1 in a heavy row, it will with sufficient probability find the second 1 before 2. stops because of ii.

The probability of ii. finishing after k steps is $\frac{\epsilon'}{d}\left(1 - \epsilon'/d\right)^{k-1}$. Then since $\left(1 - \epsilon'/d\right)^{k-1} \leq 1$, we get that the probability for ii. to finish after k or fewer steps is at most $k\epsilon'/d$. For $k = d/2\epsilon'$ this bound is equal to $1/2$. If we set $d = 16$ we get that ii. will stop after more then $8/\epsilon'$ iterations with probability at least $1/2$.

We now see that both Step 1 and 2 will finish in the expected time $\mathcal{O}(1/\epsilon')$ and Step 3 in constant time. We also have in Step 1 that $\mathcal{E}$ hits a heavy row with a probability of $1/2$. From Markov's inequality, we have that i. in Step 2 will finish with probability $1/2$ in $8/\epsilon'$ steps or fewer. Furthermore, we have shown that ii. in Step 2 will stop after more than $8/\epsilon'$ steps with a probability greater than $1/2$. Then we have that $\mathcal{E}$ will extract a witness in Step 3 with a probability of at least $1/8$. So each iteration of $\mathcal{E}$ to get to Step 3 is with expected $\mathcal{O}(1/\epsilon')$ steps with constant probability to terminate after each iteration, which means that

$\mathcal{E}$ is expected to run $\mathcal{O}(1/\epsilon')$ steps. This is better then what is needed which is $\mathcal{O}\left(\frac{1}{\epsilon'-2^{-t+2}}\right)$, which means we are done. □

This type of proof is called a heavy-row argument and is a standard way to prove that an extraction algorithm terminates in polynomial time and therefore proves the soundness property of a protocol. However, suppose a protocol has the special soundness property. In that case, it is usually easier to prove this property instead, and because of Theorem 3.9, this will be sufficient to prove that a protocol is a ZK proof.

In the ZK proofs we will study, we will allow the prover to abort after receiving the challenge as introduced by Lyubashevsky [**8**] so it does not reveal unnecessary information when the verifier would not accept. However, the prover will not necessarily abort whenever it can not answer the challenge sent by the verifier, it will with a higher probability abort, when the verifier would not accept with a high probability, as done by Baum et al. [**2**]. We will talk more about this in Section 2 of Chapter 4. Whenever the prover aborts, they start over with a new message and challenge.

CHAPTER 4

Lattices

Here we will introduce the necessary theory about lattices to construct and prove the security of the commitment scheme and ZK proofs we will be studying in Chapter 5, 6, and 7. We will first look at what a lattice is, then look at some nice results before defining the discrete Gaussian distribution over lattice points. Finally, we will define two knapsack problems connected to lattices and prove their unconditional hardness under some parameters.

1. What are Lattices

An integer lattice Λ is usually looked at as a subgroup of $\mathbb{Z}^N$. And we define a full rank lattice Λ with basis $\mathbf{B} \in \mathbb{Z}^{N\times N}$ to be

$$\Lambda = \left\{ \mathbf{v} \in \mathbb{Z}^N \;\middle|\; \exists \mathbf{z} \in \mathbb{Z}^N \text{ s.t. } \mathbf{B}\cdot\mathbf{z} = \mathbf{v} \right\}.$$

Then we can at once define one usual problem for a lattice Λ, which is to find the shortest non-zero vector of Λ. Or more general, the approximate Shortest Vector Problem ($\mathrm{SVP}_\gamma(\Lambda)$), which asks to find a non-zero element of Λ of a smaller norm than γ with the usual ℓ_2-norm.

But we will be studying lattices with an additionally property. We want Λ to be $\big(X^N+1\big)$-cyclic, which means that if we have an element $(v_1,...,v_N) \in \Lambda$ then we want $(-v_N, v_1, ..., v_{N-1}) \in \Lambda$. One type of lattices that has this property is ideals of the ring $R = \mathbb{Z}[X]/\langle X^N+1\rangle$. We can see that this property holds by multiplying the element by X. Then we can define the tree ℓ-norms of an element $f = f_{N-1}X^{N-1} + ... + f_1X + f_0 \in R$:

$$\begin{aligned} ||f||_1 &= \sum_{i=0}^{N-1} |f_i|, \\ ||f||_2 &= \left(\sum_{i=0}^{N-1} |f_i|^2 \right)^{1/2}, \\ ||f||_\infty &= \max_{i\in[N-1]} |f_i|. \end{aligned}$$

Further more since we want to do cryptography, we define the finite ring $R_p = \mathbb{Z}_p[X]/\langle X^N+1\rangle$ for a prime p. Then if $f = \sum_{i=0}^{N-1} \bar{f}_i X^i \in R_p$ we define the norms of f to be the norm of an element in R with coefficients $f_i \in \left[-\frac{q-1}{2}, \frac{q-1}{2}\right]$ such that $f_i \equiv \bar{f}_i \pmod p$ for each i. And just as usual for this norm we also get for a $f \in R_p$ the inequalities

$$||f||_1 \leq \sqrt{N}||f||_2 \leq N||f||_\infty \text{ and } ||f||_\infty \leq ||f||_1.$$

We also get two more nice bounds that we state in the following lemma.

Lemma 4.1. *Let $f, g \in R_p$, then we get the following two properties:*

1: If $||f||_\infty \leq \beta$ and $||g||_1 \leq \gamma$, then $||f \cdot g||_\infty \leq \beta \cdot \gamma$.
2: If $||f||_2 \leq \beta$ and $||g||_2 \leq \gamma$, then $||f \cdot g||_\infty \leq \beta \cdot \gamma$.

Next, we would like to know when a lattice point is invertible in R_p. We can not in general guarantee that all lattice points are invertible, but it can be proven [**9**] that when either the 2-norm or ∞-norm is not too large, we will know that they are invertible.

Lemma 4.2. *Let $N \geq \delta > 1$ be powers of 2, p a prime such that $p \equiv 2\delta + 1 \pmod{4\delta}$. Then $X^N + 1$ factors into δ irreducible polynomials $X^{N/\delta} + r_j$ modulo p, and any $y \in R_p \backslash \{0\}$ such that*

$$||y||_\infty < \frac{1}{\sqrt{\delta}} \cdot p^{1/\delta} \quad or \quad ||y||_2 < p^{1/\delta}$$

is invertible in R_p

We can also look at polynomials where the coefficients are elements in R_p. Then we can ask if there is a bound on how many zeros such a polynomial will have. And there is, which we give in the following lemma which we give a proof for.

Lemma 4.3. *Let $N \geq \delta \geq 1$ be powers of 2, p a prime such that $p \equiv 2\delta + 1 \pmod{4\delta}$ and $T \subseteq R_p$. Let $g(X) \in R_p[X]$ be a polynomial of degree τ. Then, g has at most τ^δ roots in T, and $\Pr[g(\rho) = 0 | \rho \xleftarrow{\$} T] \leq \tau^\delta / |T|$.*

Proof. By Lemma 4.2 $X^N + 1$ factors into δ different irreducible factors $X^{N/\delta} + r_j$. The factor ring of each irreducible polynomial will contribute to at most τ roots of $g(X) \in R_p[X]$. Then by the Chinese Remainder Theorem, we get that there will be at most τ^δ roots of $g(X)$. Then if $\rho \xleftarrow{\$} R_p$, the chance of ρ being a root must be less than the maximum number of roots divided by the number of elements to choose from. The same for T, since T can not have more roots of $g(X)$ then R_p itself. □

We continue by defining a few subsets of R_p which we will use in the next chapters. The first one is the Challenge Space

$$\mathcal{C} = \{ f \in R_p \mid \ ||f||_\infty = 1, ||f||_1 \leq \kappa \}, \tag{1}$$

for a chosen κ. This is the challenge space we will use in our ZK proofs in the next chapters. If we want $|\mathcal{C}| > 2^\lambda$ then we find a κ such that $\binom{N}{\kappa} \cdot 2^\kappa > 2^\lambda$. We also define the space of differences of things in the challenge space

$$\bar{\mathcal{C}} = \{ f_1 - f_2 \mid \ f_1, f_2 \in \mathcal{C}, f_1 \neq f_2 \}. \tag{2}$$

Finally we define the following set for a positive integer α

$$S_\alpha = \{ f \in R_p \backslash \{ 0 \} \mid \ ||f||_\infty \leq \alpha \}. \tag{3}$$

We will always pick an α such that all elements of S_α will be invertible.

2. The Gaussian Distribution over R^k

We continue by defining the Gaussian distribution over lattices. We do this since later, when we define the different HVZK proofs, it will be a result that will give the honest verifier property, which says that a vector sampled from a Gaussian distribution around $\mathbf{0}$ is statistically indistinguishable from a vector sampled from a Gaussian distribution around a small vector. But we first recall that the usual Gaussian distribution around a vector $v \in \mathbb{R}^N$ with standard deviation σ is given by the following density function.

$$\rho^N_{\mathbf{v},\sigma}(\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma} \cdot \exp\left(\frac{-||\mathbf{x}-\mathbf{v}||_2^2}{2\sigma^2}\right).$$

We will denote it $\rho^N_\sigma(\mathbf{x})$ when the distribution is centered around $\mathbf{0}$, and we continue by defining the Gaussian distribution around a vector from R^k.

Definition 4.4. *Let $\mathbf{v} \in R^k$ be a vector of lattice points. We then define the Gaussian distribution over R^k centred in $\mathbf{v}$ with standard deviation σ to be*

$$\mathcal{N}^k_{\mathbf{v},\sigma}(\mathbf{x}) = \frac{\rho^{k\cdot N}_{\mathbf{v},\sigma}(\mathbf{x})}{\rho^{k\cdot N}_\sigma\left(R^k\right)},$$

where $\rho^{k\cdot N}_\sigma\left(R^k\right) = \sum_{\mathbf{x}\in R^k} \rho^{k\cdot N}_\sigma(\mathbf{x})$.

If $\mathbf{v} = \mathbf{0}$, then we write $\mathcal{N}^k_\sigma(\mathbf{x})$. Now we state two results about sampling from a normal distribution, both of which are proven by Lyubashevsky [**10**]. The first one is a tail bound which says we can expect the size of the vectors we sample from the Gaussian distribution not to be too big.

Lemma 4.5. *For any $d > 0$*

$$\Pr\left[||\mathbf{z}||_2 > d\sigma\sqrt{kN} | \mathbf{z} \xleftarrow{\$} \mathcal{N}^k_\sigma\right] < d^{kN} \cdot \exp\left(\frac{kN}{2}\left(1-d^2\right)\right).$$

This lemma will be used later to know that when we first sample a vector from a Gaussian distribution and then add a small vector, we can expect with an overwhelming probability that the norm still will not get too big. This will be one of the verifier's criteria to accept in the ZK proofs.

Next, we state another result regarding the statistical difference of the Gaussian distribution around $\mathbf{0}$ and another small vector.

Lemma 4.6. *Let $V \subseteq R^k$ such that all elements have 2-norm less then T, let $\sigma \in \mathbb{R}$ such that $\sigma = \omega\left(T\sqrt{\log(kN)}\right)$ and let h be a probability distribution on V. Then there exists a $M = O(1)$ such that the following algorithms $\mathcal{A}$ and $\mathcal{S}$ are within statistical distance $2^{-\omega(\log(kN))}/M$.*

$\mathcal{A}$:

1. $\mathbf{v} \xleftarrow{\$} h$
2. $\mathbf{z} \xleftarrow{\$} \mathcal{N}^k_{\mathbf{v},\sigma}$
3. *Output* $(\mathbf{z}, \mathbf{v})$ *with probability* $\min\left(1, \dfrac{\mathcal{N}^k_\sigma(\mathbf{z})}{M \cdot \mathcal{N}^k_{\mathbf{v},\sigma}(\mathbf{z})}\right)$

$\mathcal{S}$:

1. $\mathbf{v} \xleftarrow{\$} h$
2. $\mathbf{z} \xleftarrow{\$} \mathcal{N}^k_\sigma$
3. *Output* $(\mathbf{z}, \mathbf{v})$ *with probability* $1/M$

The probability of $\mathcal{A}$ *outputting something is at least* $\frac{1-2^{-\omega(\log(kN))}}{M}$.

Baum et al. [**2**] mentions that by setting $\sigma = \alpha T$, we get that

$$M = \exp\left(12/\alpha + 1/\left(2\alpha^2\right)\right)$$

such that the statistical distance between $\mathcal{A}$ and $\mathcal{S}$ will be $2^{-100}/M$ and $\mathcal{A}$ will have probability at least $\left(1 - 2^{-100}\right)/M$ to output something. We will always chose k and N such that $kN \gg 128$, but already with $kN = 128$, we have that $M \approx 4.5$, and it decreases as k and N increases.

As we mentioned in Chapter 3 in the ZK proofs we will be studying, the prover will be able to abort when the probability for rejection is higher. This probability will come from the distribution of $\mathcal{A}$ in Lemma 4.6. The prover will have some private input $\mathbf{r} \in S^k_\alpha$ for an α and get a challenge $d \in \mathcal{C}$, then $\mathbf{v} = d \cdot \mathbf{r}$.

3. Knapsack Problems

Now we introduce the two knapsack problems for lattices, that will give the hiding and binding properties to the commitment scheme we will introduce in Chapter 5. Both the knapsack problems are basically the same as the usual Learning With Error (LWE) problem and the Short Integer Solution (SIS) problem for lattices. After we have defined these problems, we will give proof of their unconditional hardness for specific parameters. Finally, we discuss vaguely some of the ways to solve the two knapsack problems. The reason we use the two knapsack problems instead of LWE and SIS is that it will be easier to correlate the commitment scheme to these problems.

We first define the Search Knapsack problem, which is just like the ring-SIS problem but with a matrix in Hermite Normal Form.

Definition 4.7. *The Search Knapsack problem denoted* $\mathsf{SKS}^2_{n,k,\beta}$ *asks to find a short vector* $\mathbf{y}$ *satisfying* $\begin{bmatrix}\mathbf{I}_n & \mathbf{A}'\end{bmatrix} \cdot \mathbf{y} = \mathbf{0}^n$ *when given a random* $\mathbf{A}'$. *We say*

an algorithm $\mathcal{A}$ has advantage ϵ in solving the $\mathsf{SKS}^2_{n,k,\beta}$ problem if the following probability

$$\Pr\left[\left|\left|y_i\right|\right|_2 \leq \beta \wedge \begin{bmatrix}\mathbf{I}_n & \mathbf{A}'\end{bmatrix} \cdot \mathbf{y} = \mathbf{0}^n \;\middle|\; \mathbf{A}' \xleftarrow{\$} R_p^{n\times(k-n)}; \mathbf{0}^k \neq \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} \leftarrow \mathcal{A}\left(\mathbf{A}'\right)\right]$$

is at most ϵ.

We now define the Decisional Knapsack problem, which is the same as the ring-LWE problem, where the error vector is the first n entries of the short vector we multiply by our matrix.

Definition 4.8. *The Decisional Knapsack problem denoted $\mathsf{DKS}^{\infty}_{n,k,\beta}$ asks to distinguish between $\begin{bmatrix}\mathbf{I}_n & \mathbf{A}'\end{bmatrix} \cdot \mathbf{y}$ given a short vector $\mathbf{y}$ and a uniformly chosen $\mathbf{u}$ from R_p^n given $\mathbf{A}'$. We say a algorithm $\mathcal{A}$ has advantage ϵ in solving the $\mathsf{DKS}^{\infty}_{n,k,\beta}$ problem if*

$$\begin{aligned}&\left|\Pr\left[b = 1 \;\middle|\; \mathbf{A}' \xleftarrow{\$} R_p^{n\times(k-n)}, \mathbf{y} \xleftarrow{\$} S_\beta^k, b \leftarrow \mathcal{A}\left(\mathbf{A'}, \begin{bmatrix}\mathbf{I}_n & \mathbf{A}'\end{bmatrix} \cdot \mathbf{y}\right)\right]\right. \\ &\left. - \Pr\left[b = 1 \;\middle|\; \mathbf{A}' \xleftarrow{\$} R_p^{n\times(k-n)}, \mathbf{u} \xleftarrow{\$} R_p^n, b \leftarrow \mathcal{A}\left(\mathbf{A}', \mathbf{u}\right)\right]\right| \leq \epsilon.\end{aligned}$$

Since the hiding and binding properties for the commitment scheme we use will depend on the difficulty of solving the two knapsack problems, we continue by proving their unconditional hardness under specific parameters. To do this, we will need the following lemma.

Lemma 4.9. *Let $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} \in R_p^k \backslash \{\,\mathbf{0}\,\}$ be such that all nonzero entries are invertible. Then*

$$\Pr_{\mathbf{A}' \xleftarrow{\$} R_p^{n\times(k-n)}}\left[\begin{bmatrix}\mathbf{I}_n & \mathbf{A}'\end{bmatrix} \cdot \mathbf{y} = \mathbf{0}^n\right] \leq p^{-n\cdot N}.$$

Proof. First notice that at least one of $y_{n+1}, ..., y_k$ must be none-zero. Without loss of generality we can assume that $y_k \neq 0$. If we write the columns of $\begin{bmatrix}\mathbf{I}_n & \mathbf{A}'\end{bmatrix}$ as $\mathbf{a}_1, ..., \mathbf{a}_k$, we then get that the the wanted probability can be written as follows

$$\Pr_{\mathbf{a}_k \xleftarrow{\$} R_p^n}\left[\mathbf{a}_k \cdot y_k = -\sum_{i=1}^{k-1} \mathbf{a}_i \cdot y_i\right] = \Pr_{\mathbf{a}_k \xleftarrow{\$} R_p^n}\left[\mathbf{a}_k = -y_k^{-1} \cdot \sum_{i=1}^{k-1} \mathbf{a}_i \cdot y_i\right] = p^{-n\cdot N}.$$

Which is what we wanted. □

Theorem 4.10. *Let $N \geq \delta \geq 1$ be powers of 2, p a prime such that $p \equiv 2\delta + 1 (mod\ 4\delta)$ and*

$$p^{n/k} \cdot 2^{256/(k\cdot N)} \leq 2\beta < \frac{1}{\sqrt{\delta}} p^{1/\delta},$$

then any algorithm $\mathcal{A}$ will have advantage at most 2^{-128} in solving $\mathsf{DKS}^{\infty}_{n,k,\beta}$.

PROOF. First of all we notice that because of the upper bound of 2β, all non-zero elements of $S_{2\beta}$ will be invertible by Lemma 4.2. We first show that

$$\mathcal{H} = \left\{ h_{\mathbf{A}'} : S_\beta^k \to R_p^n \right\}, \text{ where } h_{\mathbf{A}'}(\mathbf{y}) = \begin{bmatrix}\mathbf{I}_n & \mathbf{A}'\end{bmatrix} \cdot \mathbf{y}$$

is a universal set of functions. In other words, we want to show that given two $\mathbf{y}_1, \mathbf{y}_2 \in S_\beta^k$, the probability of finding an $h_{\mathbf{A}'}$ with a collision is small. This we get from Lemma 4.9 by the following

$$\begin{aligned}&\Pr_{\mathbf{A}' \xleftarrow{\$} R_p^{(k-n)\times n}} \left[\begin{bmatrix}\mathbf{I}_n & \mathbf{A}'\end{bmatrix} \cdot \mathbf{y}_1 = \begin{bmatrix}\mathbf{I}_n & \mathbf{A}'\end{bmatrix} \cdot \mathbf{y}_2 \right] \\ &= \Pr_{\mathbf{A}' \xleftarrow{\$} R_p^{(k-n)\times n}} \left[\begin{bmatrix}\mathbf{I}_n & \mathbf{A}'\end{bmatrix} \cdot \mathbf{y}' = \mathbf{0}^n \right] \leq p^{-n\cdot N},\end{aligned}$$

where $\mathbf{y}' = \mathbf{y}_1 - \mathbf{y}_2 \in S_{2\beta}$. Then we know that $\mathcal{H}$ is a universal hash family onto R_p^n. We also get a lower bound on the min-entropy (the negative logarithm of the probability of the most likely outcome) of $\mathbf{y} \xleftarrow{\$} S_{2\beta}^k$ by

$$\begin{aligned}-\log\left(\frac{1}{|S_{2\beta}^k|}\right) \geq \log\left((2\beta)^{k\cdot N}\right) &= k \cdot N \cdot \log(2\beta) \\ &\geq k \cdot N \cdot \log\left(p^{n/k} \cdot 2^{256/(k\cdot N)}\right) \\ &= k \cdot N \cdot \left(\frac{n}{k}\log(p) + \frac{256}{k\cdot N}\log(2)\right) \\ &= \log|R_p^n| + 256.\end{aligned}$$

Then by the Leftover Hash Lemma [7] the distributions $\left(\mathbf{A}', h_{\mathbf{A}'}(\mathbf{y})\right)$ and $\left(\mathbf{A}', \mathbf{u}\right)$ where $\mathbf{y} \xleftarrow{\$} S_\beta^k$ and $\mathbf{u} \xleftarrow{\$} R_p^n$ will have statistical difference at most 2^{-128}. Then we see that the advantage of solving $\mathsf{DKS}_{n,k,\beta}^\infty$ will also be at most 2^{-128}. □

Notice that in the proof, we have the upper bound of β only such that Lemma 4.9 holds, so we can prove that $\mathcal{H}$ is universal. However, intuitively it should be harder to solve $\mathsf{DKS}_{n,k,\beta}^\infty$ when β increases and therefore be harder for β greater than the upper bound in Theorem 4.10. There exists reductions from $\mathsf{DKS}_{n,k,\beta}^\infty$ to $\mathsf{DKS}_{n,k,\beta'}^\infty$ when β divides β' but this does not hold in general. But as we will see from the next lemma, we do not want β to be too big since the harness of $\mathsf{SKS}_{n,k,\beta}^2$ will increase as β decreases, which means that we want to use a β as small as possible that satisfy the lower bound in Theorem 4.10.

Theorem 4.11. *Let $N \geq \delta \geq 1$ be powers of 2, p a prime such that $p \equiv 2\delta + 1 (mod\ 4\delta)$ and*

$$\begin{aligned}&\beta < p^{1/\delta}, \text{ and} \\ &\beta < \sqrt{\frac{N}{2\pi e}} \cdot p^{n/k} 2^{-128/(k\cdot N)} - \frac{\sqrt{N}}{2},\end{aligned}$$

then any algorithm $\mathcal{A}$ will have advantage at most 2^{-128} in solving $\mathsf{SKS}_{n,k,\beta}^2$.

PROOF. First of all we see because the first bound of β, we can use Lemma 4.9. Now if we let $V_N(r)$ be the volume of a N-dimensional ball of radius r, then there are fewer then $V_N\left(\beta + \sqrt{N}/2\right)$ elements $y \in R$ such that $||y||_2 \leq \beta$. Then by the union bound we get

$$\begin{aligned}
&\Pr_{\mathbf{A}' \xleftarrow{\$} R_p^{(k-n)\times n}} \left[\exists \mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} \text{ s.t. } ||y_i||_2 \leq \beta \text{ and } \begin{bmatrix} \mathbf{I}_n & \mathbf{A}' \end{bmatrix} \right] \\
&\leq V_N\left(\beta + \sqrt{N}/2\right)^k \cdot p^{-n\cdot N} \\
&< \left(\sqrt{\frac{2\pi e}{N}} \cdot \left(\beta + \sqrt{N}/2\right)\right)^{k\cdot N} \cdot p^{-n\cdot N} < 2^{-128},
\end{aligned}$$

where the last inequality comes from the second bound of β. This bound says that the probability of the existence of a solution $\mathbf{y}$ of $\mathsf{SKS}^2_{n,k,\beta}$ is small and we therefor get the wanted bound. □

So we have proven for which bounds on β we have unconditional hardness of the two knapsack problems, but we will also discuss the scenarios when these bounds do not hold. For $\mathsf{SKS}^2_{n,k,\beta}$ we can define the set

$$\Lambda = \left\{ \mathbf{y} \in R^k \mid \begin{bmatrix} \mathbf{I}_n & \mathbf{A}' \end{bmatrix} \cdot \mathbf{y} = \mathbf{0} \mod p \right\}.$$

We can see that this is a group under addition over R^k. We can also see that finding a solution $\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix}$ such that $||y_i||_2 \leq \beta$ is at least as hard as finding a $\mathbf{y}$ such that $||\mathbf{y}||_2 \leq \beta\sqrt{k}$. And since Λ is also a group over $\mathbb{Z}^{kN}$, this problem is the same as finding a vector of norm $\beta\sqrt{k}$ in a random lattice of dimension kN. From Theorem 4.11 such a vector does not exist if β is small enough, so even an all-powerful adversary will not be able to find such a vector. But we know it will be easier to find as β increases.

For $\mathsf{DKS}^\infty_{n,k,\beta}$ the best current method of solving it, is to find a close vector to a target in Λ. In the case $\mathbf{t}$ is picked uniformly at random, then the target vector will be uniformly distributed, but if $\mathbf{t} = \begin{bmatrix} \mathbf{I}_n & \mathbf{A}' \end{bmatrix}$ for a $\mathbf{y}$ with small coefficients, then the target vector will be close to Λ. From Theorem 4.10 we have that when β is large enough, then $\mathbf{t} = \begin{bmatrix} \mathbf{I}_n & \mathbf{A}' \end{bmatrix}$ will have the same distribution as a uniform $\mathbf{t}$ making the problem unsolvable, but as β gets smaller it becomes easier to solve.

CHAPTER 5

Commitments and Zero-Knowledge Proofs using Lattices

This chapter will use what we have introduced of lattices to construct the commitment scheme we will be using. We will then prove that the hiding and binding properties of this commitment scheme depend on the hardness of the knapsack problems introduced in Chapter 4. We will then use this commitment scheme to construct multiple ZK proofs of different relations. Two of these ZK proofs will be proofs of a correct shuffle, the first of messages $m_i \in R_p$, and the second of messages $\mathbf{m} \in R_p^\ell$.

In Table 1, we see a list of different sets and parameters and the notation we use for them. We will use these parameters in the commitment scheme and the ZK proofs in the following chapters.

Parameters	**Explanation**
$R = \mathbb{Z}[X]/\langle X^N + 1\rangle$	The ring over which we define norms of vectors
$R_p = \mathbb{Z}_p[X]/\langle X^N + 1\rangle$	The ring over which we do most of the computation
p	The prime modulus defining R_p
k	With (over R_p) of the commitment matrix
n	Height (over R_p) of the commitment matrix $\mathbf{A}_1$
ℓ	Dimention (over R_p) of the message space
β	Norm bound for honest prover's randomness in ℓ_∞-norm
S_β	Set of all $x \in R_p$ with ℓ_∞-norm at most β
$\mathcal{C}$	Subset of S_1 which is the challenge space (see (1))
$\bar{\mathcal{C}}$	The set of differences $\mathcal{C} - \mathcal{C}$ excluding 0
κ	The maximum ℓ_1-norm of ellements in $\mathcal{C}$
$\sigma = 11 \cdot \kappa \cdot \beta \cdot \sqrt{kN}$	Standard deviation used in the zero-knowledge proofs

TABLE 1. An overview of parameters and notation

1. The Commitment Scheme

We will start by defining the commitment scheme we are going to use in Chapter 5, 6, and 7. We do this by describing the three algorithms KeyGen, Commit,

and Open. The message space for the commitment scheme will be R_p^ℓ.

KeyGen: Generate $\mathbf{B}_1 \in R_p^{n\times k}$ and $\mathbf{B}_2 \in R_p^{\ell\times k}$ as

$$\mathbf{B}_1 = \begin{bmatrix}\mathbf{I}_n & \mathbf{B}_1'\end{bmatrix}, \quad \text{where } \mathbf{B}_1' \xleftarrow{\$} R_p^{n\times(k-n)} \tag{4}$$

$$\mathbf{B}_2 = \begin{bmatrix}\mathbf{0}^{\ell\times n} & \mathbf{I}_\ell & \mathbf{B}_2'\end{bmatrix}, \quad \text{where } \mathbf{B}_2' \xleftarrow{\$} R_p^{\ell\times(k-n-\ell)} \tag{5}$$

Then output $\mathbf{B} = \begin{bmatrix}\mathbf{B}_1\\ \mathbf{B}_2\end{bmatrix} \in R_p^{(n+\ell)\times k}$.

Commit: The commit algorithm take as input public parameter $\mathbf{B} = \begin{bmatrix}\mathbf{B}_1\\ \mathbf{B}_2\end{bmatrix} \in R_p^{(n+\ell)\times k}$ and a message $\mathbf{m} \in R_p^\ell$, then it generates a $\mathbf{r} \xleftarrow{\$} S_\beta^k$ and compute

$$Com(\mathbf{m};\mathbf{r}) := \mathbf{c} = \begin{bmatrix}\mathbf{c}_1\\ \mathbf{c}_2\end{bmatrix} = \begin{bmatrix}\mathbf{B}_1\\ \mathbf{B}_2\end{bmatrix}\cdot\mathbf{r} + \begin{bmatrix}\mathbf{0}^n\\ \mathbf{m}\end{bmatrix} \tag{6}$$

and outputs $\mathbf{c}$ and $(\mathbf{r};1)$.

Open: The opening algorithm take as input public parameter $\mathbf{B} = \begin{bmatrix}\mathbf{B}_1\\ \mathbf{B}_2\end{bmatrix} \in R_p^{(n+\ell)\times k}$, a commitment $\mathbf{c} = \begin{bmatrix}\mathbf{c}_1\\ \mathbf{c}_2\end{bmatrix}$, a message $\mathbf{m} \in R_p$, a $\mathbf{r} = \begin{bmatrix}r_1\\ \vdots\\ r_k\end{bmatrix} \in R_p^k$ and a challenge $f \in \bar{\mathcal{C}}$. It outputs 1 if

$$f\cdot\begin{bmatrix}\mathbf{c}_1\\ \mathbf{c}_2\end{bmatrix} = \begin{bmatrix}\mathbf{B}_1\\ \mathbf{B}_2\end{bmatrix}\cdot\mathbf{r} + f\cdot\begin{bmatrix}\mathbf{0}^n\\ \mathbf{m}\end{bmatrix}, \tag{7}$$

and $||r_i||_2 \leq 4\sigma\sqrt{N}$ for all i, and outputs 0 otherwise. σ is defined as in Table 1.

We start by commenting on the challenge parameter f. Why do we have this when the Commit algorithm just sets it to 1? And when Open checks that (7) holds, if $f \neq 1$ then it is a sign of an attack or someone who try to cheat. The reason for f is that in the ZK proofs we will study, the witnesses are always openings of commitments. Furthermore, when extracting witnesses in the proofs of the soundness property, we can not guarantee that (6) holds. However, this will not cause any security problems, as we will prove that the hiding and binding properties will follow from the two knapsack problems for lattices.

Before we prove the hiding and binding properties, we show that the commitment scheme is complete, which means that a commitment with associated $\mathbf{r}$ and f outputted from Commit opens to 1.

Theorem 5.1. *Given a message $\mathbf{m} \in R_p^\ell$ and publick parameter $\mathbf{B}$ generated from* KeyGen. *Then* $\mathsf{Open}(\mathbf{B}, \mathsf{Commit}(\mathbf{B},\mathbf{m}),\mathbf{m}) = 1$.

PROOF. Let $\mathbf{c}$, $\mathbf{r} = \begin{bmatrix} r_1 \\ \vdots \\ r_k \end{bmatrix} \in S_\beta^k$ and $f = 1$ be the output of $\mathsf{Commit}(\mathbf{B}, \mathbf{m})$. First we observe that since $r_i \in S_\beta$ for each i we get that

$$||r_i||_2 \leq \sqrt{N}||r_i||_\infty \leq \beta\sqrt{N} \leq 4\sigma\sqrt{N},$$

since $\sigma = 11 \cdot \kappa \cdot \beta \cdot \sqrt{kN}$. We also see that (7) is the same as (6) since $f = 1$, which is how $\mathbf{c}$ is computed and therefor holds. □

Now we will start to prove the hiding and binding property of the commitment scheme. We will do this by showing if there exists an algorithm that can break the hiding or binding property, then there exists an algorithm that can break the Search Knapsack problem or the Decisional Knapsack problem.

Theorem 5.2. *If there exists an algorithm $\mathcal{A}$ which can brake the ϵ-hiding property of the commitment scheme, then there exists another algorithm $\mathcal{A}'$ that runs at the same time and has an advantage at least ϵ in solving $\mathsf{DSK}^\infty_{n+\ell,k,\beta}$.*

PROOF. Given an instance $\mathbf{A} = \begin{bmatrix} \mathbf{I}_{n+\ell} & \mathbf{A}' \end{bmatrix}, \mathbf{t}$ of $\mathsf{DSK}^\infty_{n+\ell,k,\beta}$. $\mathcal{A}'$ starts by choosing the public parameter $\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}$ as

$$\begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n & \mathbf{R} \\ \mathbf{0}^{\ell \times n} & \mathbf{I}_\ell \end{bmatrix} \cdot \mathbf{A},$$

where $R \xleftarrow{\$} R_p^{n \times \ell}$. Then sends $\mathbf{B}$ to the algorithm $\mathcal{A}$ and obtains $\mathbf{m}_0, \mathbf{m}_1 \in R_p^\ell$. $\mathcal{A}'$ generates a bit $b \xleftarrow{\$} \{0, 1\}$ and commits to $\mathbf{m}_b$ as

$$\mathbf{c} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n & \mathbf{R} \\ \mathbf{0}^{\ell \times n} & \mathbf{I}_\ell \end{bmatrix} \cdot \mathbf{t} + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m}_b \end{bmatrix},$$

and sends it to $\mathcal{A}$. If $\mathcal{A}$ returns $b' = b$, then $\mathcal{A}'$ will output 1, and 0 if $\mathcal{A}$ outputs $b' \neq b$.

First suppose the public parameters $\begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}$ are correctly distributed for the commitment scheme. We can then see that if $\mathbf{t} = \mathbf{A} \cdot \mathbf{y}$ for some $\mathbf{y} \in S_\beta^k$, then

$$\begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n & \mathbf{R} \\ \mathbf{0}^{\ell \times n} & \mathbf{I}_\ell \end{bmatrix} \cdot \mathbf{A} \cdot \mathbf{y} + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m}_b \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} \cdot \mathbf{y} + \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m}_b \end{bmatrix} = Com(\mathbf{m}_b; \mathbf{y}),$$

and $\mathcal{A}$ should then have probability $1/2 + \epsilon$ to output $b' = b$, which means $\mathcal{A}'$ will in this case output 1 with prbability $1/2 + \epsilon$.

In the case that $\mathbf{t} \xleftarrow{\$} R_p^k$, then we can see that the commitment of $\mathbf{m}_b$ will be independent of the message, so $\mathcal{A}$ will output $b' = b$ with probability $1/2$. We can then see that $\mathcal{A}'$ will have an advantage of ϵ of solving $\mathsf{DSK}^\infty_{n+\ell,k,\beta}$.

Now we just have to see that the public parameters are correctly distributed. We start by writing

$$A = \begin{bmatrix} \mathbf{I}_n & \mathbf{0}^n & \mathbf{A}_1' \\ \mathbf{0}^{\ell \times n} & \mathbf{I}_\ell & \mathbf{A}_2' \end{bmatrix}.$$

Then we get the following

$$\begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n & \mathbf{R} & \mathbf{A}_1' + \mathbf{R} \cdot \mathbf{A}_2' \\ \mathbf{0}^{\ell \times n} & \mathbf{I}_\ell & \mathbf{A}_2' \end{bmatrix}.$$

We then see that since $\mathbf{A}_1', \mathbf{A}_2'$ and $\mathbf{R}$ are just uniformly picked matrices, that our public parameters will have the same distributioan as in (4) and (5). □

Next, we prove that if one can break the binding property of the commitment scheme, then one will also be able to solve SKS^2.

Theorem 5.3. *Suppose there exist an algorithm $\mathcal{A}$ that can brake the ϵ-binding property of the commitment scheme, then there exists an algorithm $\mathcal{A}'$ that has an advantage at least ϵ in solving $\mathsf{SKS}^2_{n,k,16\sigma\sqrt{\kappa N}}$.*

Proof. Let $\mathcal{A}'$ be given $\mathbf{B}_1 = \begin{bmatrix} \mathbf{I}_n & \mathbf{B}_1' \end{bmatrix}$ as an instance of $\mathsf{SKS}_{n,k,\gamma}$. $\mathcal{A}'$ then generates $\mathbf{B}_2' \xleftarrow{\$} R_p^{\ell \times (k-n-1)}$ and sets $\mathbf{B}_2 = \begin{bmatrix} \mathbf{0}^{\ell \times n} & \mathbf{I}_\ell & \mathbf{B}_2' \end{bmatrix}$. $\mathcal{A}'$ then sends $\begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix}$ to $\mathcal{A}$ as public parameters for the commitment scheme. Suppose $\mathcal{A}$ is able to come up with a commitment $\begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix}$ and two valid openings $(\mathbf{m}; \mathbf{r}; f)$ and $(\mathbf{m}'; \mathbf{r}'; f')$, where $\mathbf{m} \neq \mathbf{m}'$, then we get that

$$f \cdot \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} \cdot \mathbf{r} + f \cdot \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m} \end{bmatrix}, \tag{8}$$

$$f' \cdot \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} \cdot \mathbf{r}' + f' \cdot \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m}' \end{bmatrix}. \tag{9}$$

Then $\mathcal{A}'$ multiply (8) by f' and (9) by f and subtract them from another and get.

$$0 = (f'\mathbf{r} - f\mathbf{r}') \cdot \mathbf{B}_1,$$
$$0 = (f'\mathbf{r} - f\mathbf{r}') \cdot \mathbf{B}_2 + f \cdot f' \cdot (\mathbf{m} - \mathbf{m}')$$

Since $\mathbf{m} \neq \mathbf{m}'$ and f and f' are invertible, $f \cdot f' \cdot (\mathbf{m} - \mathbf{m}')$ is not zero, which impies that $f'\mathbf{r} - f\mathbf{r}'$ can not be zero either. Since thise are valid openings $f, f' \in \bar{\mathcal{C}}$, which implies that $||f||_2, ||f'||_2 \leq 2\sqrt{\kappa}$. We also know then that $||r_i||_2, ||r_i'||_2 \leq 4\sigma\sqrt{N}$ for each i. Form this we get

$$||f'\mathbf{r} - f\mathbf{r}'||_2 \leq 2||f'\mathbf{r}||_2 \leq 16\sigma\sqrt{\kappa N}.$$

So we can conclude that $\mathcal{A}'$ will with advantage at least ϵ at producing a solution $f'\mathbf{r} - f\mathbf{r}'$ of $\mathsf{SKS}_{n,k,16\sigma\sqrt{\kappa N}}$. □

We have now seen that the binding property depends on SKS and the hiding property depends on DKS, and both of these knapsack problems depend on the parameter β. We saw in Chapter 4 that the hardness of SKS decreases as β increases, but then the hardness of DKS increases. This means that we must prioritize either the binding or hiding property of the commitment scheme. We can decide that the

commitment scheme shall be statistically binding or statistical hiding, but then the other property will be much weaker. But we can also choose a β in between such that we get strong binding and hiding properties, but not statistical binding or statistical hiding.

For the rest of the thesis, we will assume that we have chosen parameters such that the hiding and binding properties are satisfactory for our purposes.

2. Zero-Knowledge proof of Opening

Now that we have a commitment scheme that is both sufficiently hiding and binding for existing parameters, we are ready to introduce our two first ZK proofs. These will be two proofs where the prover convinces the verifier that it knows a valid opening of a known commitment $\mathbf{c}$. One of them, the prover will prove that the commitment opens to a specific known message, and the other, the prover only shows that it knows an opening without revealing the message.

We will start with Π_{Open} as given in Figure 5.1, which is the one where the message is not revealed. Then we go on to prove the completeness, special soundness, and honest-verifier zero-knowledge properties of the protocol Π_{Open}, which will show that Π_{Open} is a Σ-protocol.

Before we prove the properties of Π_{Open}, we notice that in the protocol, we pick $\mathbf{y} \xleftarrow{\$} \mathcal{N}_\sigma^k$, which means $\mathbf{y}$ will be an element of R^k. We will however still do computations with $\mathbf{y}$ as it was an element of R_p^k instead. In practice, when sampling a vector from $\mathcal{N}_\sigma^k$, it will not be picked from it truly, since a computer is not able to compute the infinite sum $\rho_\sigma^{k\cdot N}\left(R^k\right)$. However, this will not be a problem in practice.

Lemma 5.4. *The verifier in Π_{Open} given in Figure 5.1 will output 1 with overwhelming probability when not aborted. The probability of abort is at most $1 - \frac{1-2^{-100}}{M}$.*

Proof. By Lemma 4.6 we get that the probability of $\mathcal{P}$ aborting is $1-\frac{1-2^{-100}}{M}$.

If we look at the equality $\mathbf{B}_1 \cdot \mathbf{z} = \mathbf{t} + d \cdot \mathbf{c}_1$, when Π_{Open} is followed honestly, we get on the left hand side,

$$\mathbf{B}_1 \cdot \mathbf{z} = \mathbf{B}_1 \cdot \mathbf{y} + d \cdot \mathbf{B}_1 \cdot \mathbf{r},$$

and on the right hand side, where $\mathbf{c}_1$ is as in (6),

$$\mathbf{t} + d \cdot \mathbf{c}_1 = \mathbf{B}_1 \cdot \mathbf{y} + d \cdot \mathbf{B}_1 \cdot \mathbf{r}.$$

Finely we get from Lemma 4.5 that since $\mathbf{y} \xleftarrow{\$} \mathcal{N}_\sigma^k$ that $||z_i||_2 \leq 2\sigma\sqrt{N}$ for each i with exception of a negligible probability. □

Then we go over to prove the special soundness property of Π_{Open}.

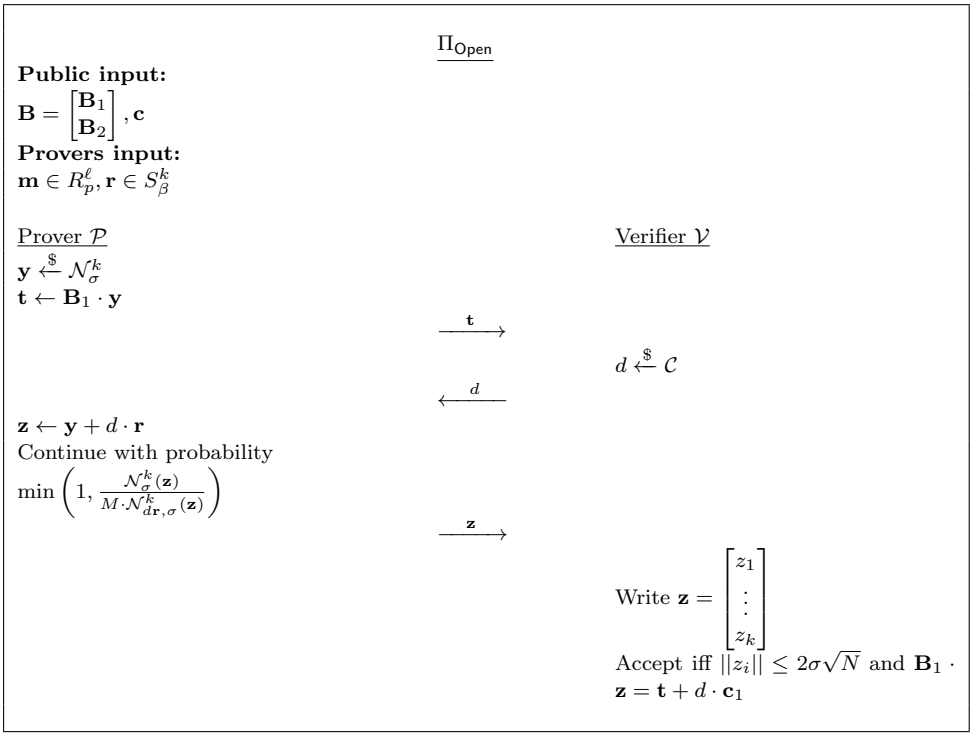


FIGURE 5.1. Zero-Knowledge Proof of opening.

Lemma 5.5. *Given a commitment* $\mathbf{c} = \begin{bmatrix}\mathbf{c}_1 \\ \mathbf{c}_2\end{bmatrix}$ *and a pair of accepted transcripts* $(\mathbf{t}, d, \mathbf{z}), (\mathbf{t}, d', \mathbf{z}')$ *of* Π_{Open}*, then we can find a valid opening* $(\mathbf{m}, \mathbf{r}', f)$ *of the commitment scheme.*

PROOF. We start by setting $f = d - d' \in \bar{\mathcal{C}}$, where we can see that f is non-zero since $d \neq d'$. Hence f is invertible by Lemma 4.2. We then set $\mathbf{r}' = \mathbf{z} - \mathbf{z}'$ and we set $\mathbf{m}' = \mathbf{c}_2 - f^{-1} \cdot \mathbf{B}_2 \cdot \mathbf{r}'$. Then we see that

$$\begin{aligned}
\begin{bmatrix}\mathbf{B}_1 \\ \mathbf{B}_2\end{bmatrix} \cdot \mathbf{r}' + f \cdot \begin{bmatrix}\mathbf{0}^n \\ \mathbf{m}'\end{bmatrix} &= \begin{bmatrix}\mathbf{B}_1 \cdot (\mathbf{z} - \mathbf{z}') \\ \mathbf{B}_2 \cdot \mathbf{r}' + f \cdot \left(\mathbf{c}_2 - f^{-1}\mathbf{B}_2 \cdot \mathbf{r}'\right)\end{bmatrix} \\
&= \begin{bmatrix}\mathbf{B}_1 \cdot (\mathbf{t} + d \cdot \mathbf{c}_1 - \mathbf{t} - d' \cdot \mathbf{c}_1) \\ f \cdot \mathbf{c}_2\end{bmatrix} \\
&= \begin{bmatrix}(d - d') \cdot \mathbf{c}_1 \\ f \cdot \mathbf{c}_2\end{bmatrix} \\
&= f \cdot \begin{bmatrix}\mathbf{c}_1 \\ \mathbf{c}_2\end{bmatrix}.
\end{aligned}$$

We also get that $||r_i'||_2 \leq ||z_i||_2 + ||z_i'||_2 \leq 4\sigma\sqrt{N}$, which shows that $(\mathbf{m}', \mathbf{r}', f)$ is a valid opening of $\mathbf{c}$. □

Theorem 5.6. *Given* $\mathbf{c} = Com(\mathbf{m}; \mathbf{r})$*, the protocol* Π_{Open} *given in Figure 5.1 is an HVZK proof of the relation*

$$R_{\mathsf{Open}} = \left\{ (s, w) \;\middle|\; \begin{array}{c} s = (\mathbf{c}, \mathbf{B}_1, \mathbf{B}_2), w = (\tilde{\mathbf{m}}, \tilde{\mathbf{r}}, f), \\ \mathsf{Open}(\mathbf{c}, \tilde{\mathbf{m}}, \tilde{\mathbf{r}}, f) = 1 \end{array} \right\}.$$

Proof. From Lemma 5.4 and Lemma 5.5, we get the completeness and special soundness properties of Π_{Open}.

To show that Π_{Open} is HVZK we let $\mathcal{S}$ be an algorithm that chooses $\mathbf{z} \xleftarrow{\$} \mathcal{N}_\sigma^k$ and $d \xleftarrow{\$} \mathcal{C}$. If we then set $\mathbf{t} = \mathbf{B}_1\mathbf{z} - d\mathbf{c}_1$ we get from Lemma 4.6 that this will be statistically indistinguishable from a real non-aborting transcript of Π_{Open}.

This shows that Π_{Open} is a Σ-protocol and is therefore by Theorem 3.9 an HVZK proof. □

Π_{Open} is an excellent example of an HVZK proof. We will not use it in any way later, but the ideas in the proof of it being an HVZK proof will be used in relevant to understanding the security proofs of the ZK proofs we will introduce later.

We continue by introducing the following ZK proof Π_{Commit} as given in Figure 5.2 for the opening of a known message. This protocol will be almost the same as Π_{Open}, but in this proof, the message $\mathbf{m}$ will be part of the public input, and the prover will convince the prover that $\mathbf{c}$ is indeed a commitment of the message $\mathbf{m}$.

Theorem 5.7. *Let* $\mathbf{c} = Com(\mathbf{m}; \mathbf{r})$*, then* Π_{Commit} *is an HVZK proof of the relation*

$$R_{\mathsf{Commit}} = \left\{ (s, w) \;\middle|\; \begin{array}{c} s = (\mathbf{c}, \mathbf{m}, \mathbf{B}_1, \mathbf{B}_2), w = (\tilde{\mathbf{r}}, f), \\ \mathsf{Open}(\mathbf{c}, \mathbf{m}, \tilde{\mathbf{r}}, f) = 1 \end{array} \right\}.$$

Proof. The completeness of Π_{Commit} follows exactly as for Π_{Open}. So does almost the special soundness property.
Given two valid transcripts $\{\mathbf{t}, d_1, \mathbf{z}_1\}$ and $\{\mathbf{t}, d_2, \mathbf{z}_2\}$ of Π_{Commit} we define $\mathbf{r}' = \mathbf{z}_1 - \mathbf{z}_2$ and $f = d_1 - d_2$ and get that

$$\begin{aligned} \mathbf{B} \cdot \mathbf{r}' + f \cdot \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m} \end{bmatrix} &= \mathbf{t} + d_1\mathbf{c} - d_1 \cdot \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m} \end{bmatrix} - \mathbf{t} + d_1\mathbf{c} - d_2 \cdot \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m} \end{bmatrix} + (d_1 - d_2) \cdot \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m} \end{bmatrix} \\ &= (d_1 - d_2) \cdot \mathbf{c} = f \cdot \mathbf{c}. \end{aligned}$$

And just as in the proof of Theorem 5.6 $||r_i'|| \leq 4\sigma\sqrt{N}$ for all i. So then we have that $(\mathbf{r}', f)$ is a witness for $(\mathbf{c}, \mathbf{m}, \mathbf{B}_1, \mathbf{B}_2)$, and we have special soundness.

We simulate Π_{Commit} almost the same way as Π_{Open} by choosing $\mathbf{z} \xleftarrow{\$} \mathcal{N}_\sigma^k$ and $d \xleftarrow{\$} \mathcal{C}$, and then we sett $\mathbf{t} = \mathbf{B} \cdot \mathbf{z} - d \cdot \mathbf{c} - d \cdot \begin{bmatrix} \mathbf{0}^n \\ \mathbf{m} \end{bmatrix}$, which we see that is statistically indistinguishable from a real transcript of Π_{Commit}.

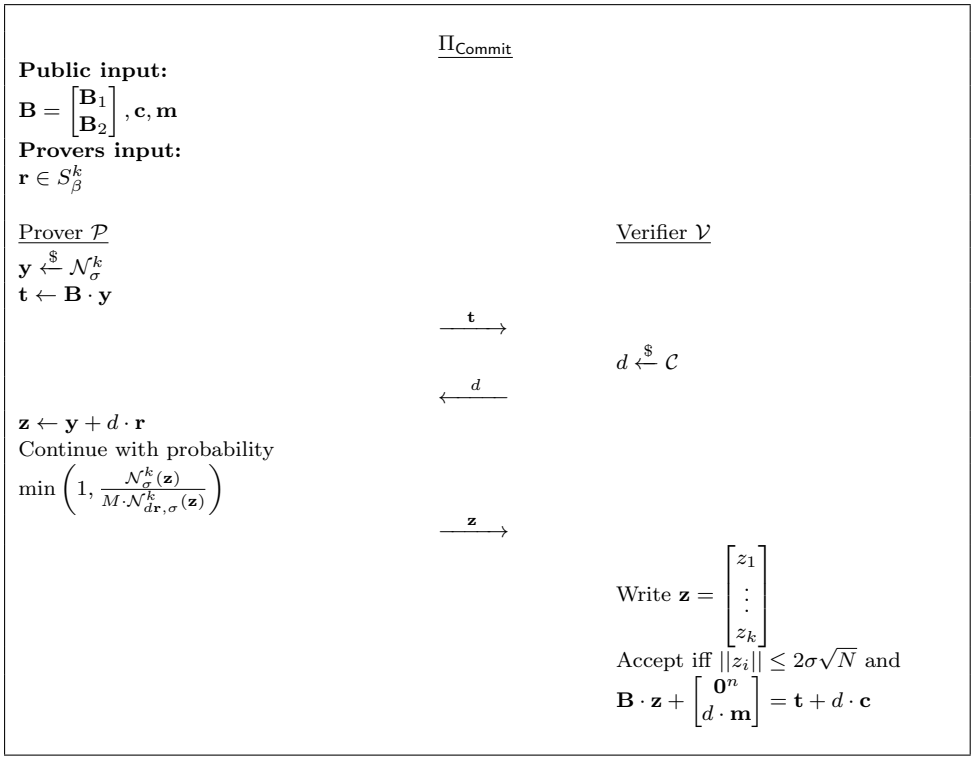


FIGURE 5.2. Zero-Knowledge Proof of committing to a specific message $\mathbf{m}$.

Then we have shown that Π_{Commit} is a Σ-protocol, and it is, therefore, an HVZK proof. □

We notice that when simulating Π_{Commit}, we could simulate it with a message $\mathbf{m}$, without $\mathbf{c}$ being the commitment to $\mathbf{m}$. This will be relevant in a ZK proof we will introduce in Chapter 7.

3. Zero-Knowledge proof of Linear Relations

We have given a ZK proof of having the opening of a commitment $\mathbf{c}$, but we can extend this protocol to be a ZK proof of this with two commitments as well as a linear relation between them. Given two commitments $\mathbf{c} = Com(\mathbf{m}; \mathbf{r})$ and $\mathbf{c}' = Com(\mathbf{m}'; \mathbf{r}')$, we want to give a proof that $\mathbf{m}' = \alpha\mathbf{m} + \beta$ given public $\alpha, \beta \in R_p$, and that $\mathbf{c}'$ and $\mathbf{c}$ opens to $\mathbf{m}'$ and $\mathbf{m}$.

Theorem 5.8. *Let* $\alpha, \beta \in R_p$ *and* $\mathbf{c} = Com(\mathbf{m}; \mathbf{r}) = \begin{bmatrix}\mathbf{c}_1\\ \mathbf{c}_2\end{bmatrix}$, $\mathbf{c}' = Com(\mathbf{m}'; \mathbf{r}') = \begin{bmatrix}\mathbf{c}_1'\\ \mathbf{c}_2'\end{bmatrix}$ *be commitments, then* Π_{Lin} *given in Figure 5.3 is an HVZK proof of the*

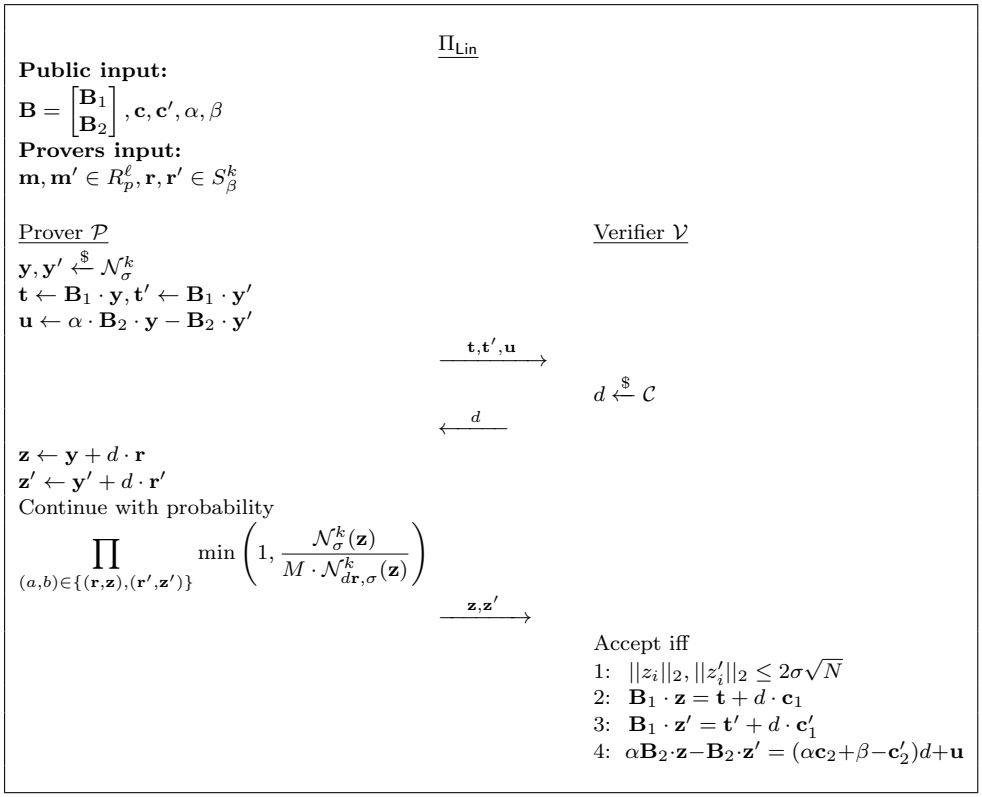


FIGURE 5.3. Zero-Knowledge Proof of linear relation of two messages.

relation

$$R_{\mathsf{Lin}} = \left\{ (s, w) \;\middle|\; \begin{array}{c} s = (\alpha, \beta, \mathbf{c}, \mathbf{c}'\mathbf{B}_1, \mathbf{B}_2), w = (\tilde{\mathbf{m}}, \tilde{\mathbf{r}}, \tilde{\mathbf{r}}', f), \\ \mathsf{Open}(\mathbf{c}, \tilde{\mathbf{m}}, \tilde{\mathbf{r}}, f) = \mathsf{Open}(\mathbf{c}', \alpha\tilde{\mathbf{m}} + \beta, \tilde{\mathbf{r}}', f) = 1 \end{array} \right\}.$$

This result is proven by Baum et al. [**2**] and Aranha et al. [**1**] state that it is easy to extend it to a proof of this for a linear relation of multiple messages. We will denote this ZK proof Π_{Lin^+} which is given in Figure 5.4, which will be a proof of the following relation,

$$R_{\mathsf{Lin}^+} = \left\{ (s, w) \;\middle|\; \begin{array}{c} s = (\alpha_1, ...\alpha_\tau, \beta, \mathbf{c}_1, ..., \mathbf{c}_\tau, \mathbf{c}', \mathbf{B}_1, \mathbf{B}_2), \\ w = (\tilde{\mathbf{m}}_1, ..., \tilde{\mathbf{m}}_\tau, \tilde{\mathbf{r}}_1, ..., \tilde{\mathbf{r}}_\tau, \tilde{\mathbf{r}}', f_1, ..., f_\tau, f), \\ \mathsf{Open}(\mathbf{c}', \sum_{i=1}^{\tau} \alpha_i\tilde{\mathbf{m}}_i + \beta, \tilde{\mathbf{r}}', f) = 1, \\ \mathsf{Open}(\mathbf{c}_i, \tilde{\mathbf{m}}_i, \tilde{\mathbf{r}}_i, f_i) = 1 \forall i \in [\tau] \end{array} \right\}.$$

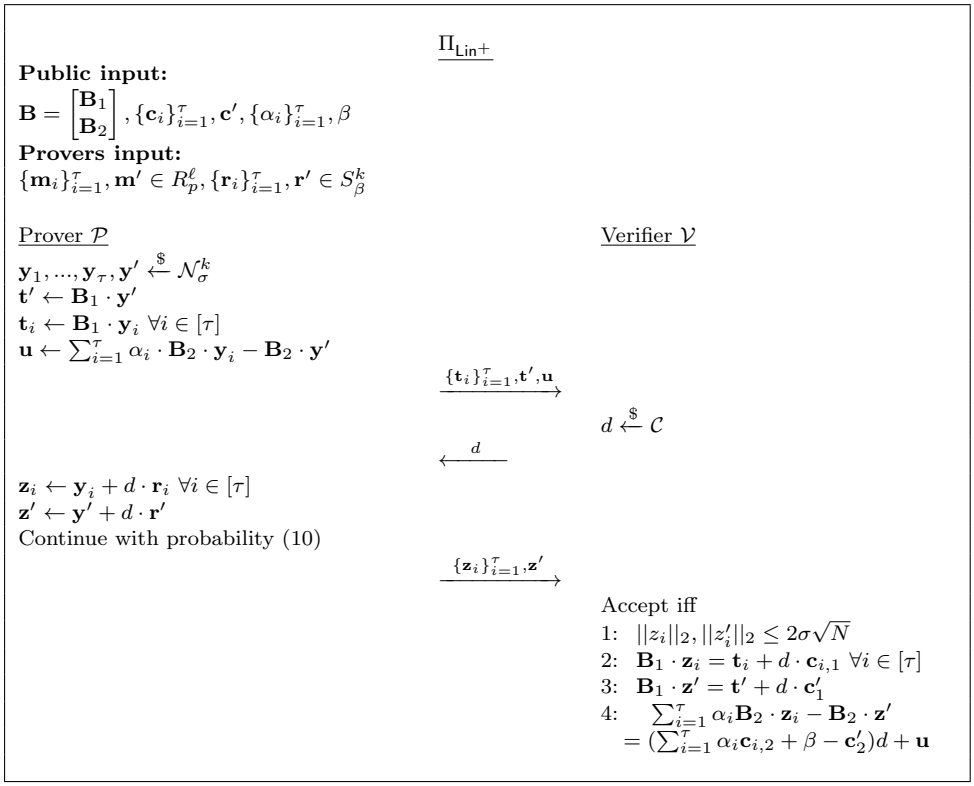


FIGURE 5.4. Zero-Knowledge Proof of linear relation between multiple messages.

First of all we see from Lemma 4.6 that the probability of the prover to not aborting in Π_{Lin^+} will be

$$\prod_{(a,b)\in\{(\mathbf{r}',\mathbf{z}'),\{(\mathbf{r}_i,\mathbf{z}_i)\}_{i=1}^{\tau}\}} \min\left(1, \frac{\mathcal{N}_\sigma^k(\mathbf{z})}{M \cdot \mathcal{N}_{d\mathbf{r},\sigma}^k(\mathbf{z})}\right), \tag{10}$$

which will approximately the same as $(1/M)^{\tau+1}$. If we do not let M be to big and only use the protocol for small τ's, the protocol will be sufficient.

We now give proof that Π_{Lin^+} is an HVZK proof.

Theorem 5.9. *Given scalars $\alpha_1, ..., \alpha_\tau, \beta$ and commitments $\mathbf{c}_1, ..., \mathbf{c}_\tau, \mathbf{c}'$, the protocol Π_{Lin^+} given in Figure 5.4 is an HVZK proof of the relation R_{Lin^+}.*

PROOF. We start by proving the completeness of Π_{Lin^+} when the prover does not abort. So assume $\mathbf{c}_i = Com(\mathbf{m}_i; \mathbf{r}_i)$ for each $i \in [\tau]$ and $\mathbf{c}' = Com(\mathbf{m}'; \mathbf{r}')$ where $\mathbf{m}' = \sum_{i=1}^{\tau} \alpha_i \mathbf{m}_i + \beta$, and assume the prover $\mathcal{P}$ does not abort. Then the first three points of accepting follows exactly as in the proof of Lemma 5.4. Then

we check the the forth requirement and see that

$$\begin{aligned}\left(\sum_{i=1}^{\tau}\alpha_i\mathbf{c}_{i,2}+\beta-\mathbf{c}'_2\right)d+\mathbf{u} &= \left(\sum_{i=1}^{\tau}(\alpha_i\mathbf{B}_2\cdot\mathbf{r}_i+\alpha_i\mathbf{m}_i)+\beta-\mathbf{B}_2\cdot\mathbf{r}'-\sum_{i=1}^{\tau}\alpha_i\mathbf{m}_i-\beta\right)\cdot d\\ &\quad+\sum_{i=1}^{\tau}\alpha_i\cdot\mathbf{B}_2\cdot\mathbf{y}_i-\mathbf{B}_2\cdot\mathbf{y}'\\ &=\sum_{i=1}^{\tau}\alpha_i\mathbf{B}_2\cdot(\mathbf{y}_i+d\mathbf{r}_i)-\mathbf{B}_2\cdot(\mathbf{y}'+d\mathbf{r}')\\ &=\sum_{i=1}^{\tau}\alpha_i\mathbf{B}_2\cdot\mathbf{z}_i-\mathbf{B}_2\cdot\mathbf{z}'.\end{aligned}$$

Now we prove the special soundness of Π_{Lin^+}. Assume that $(\{\mathbf{z}_i\}_{i=1}^{\tau},\mathbf{z}',d)$ and $(\{\bar{\mathbf{z}}_i\}_{i=1}^{\tau},\bar{\mathbf{z}}',\bar{d})$ are both accepting transcripts of $\{\mathbf{t}_i\}_{i=1}^{\tau},\mathbf{t}',\mathbf{u}$, where $d\neq\bar{d}$. Then we define $f=d-\bar{d}$, $\tilde{\mathbf{r}}_i=\mathbf{z}_i-\bar{\mathbf{z}}_i$ for each i and $\tilde{\mathbf{r}}'=\mathbf{z}'-\bar{\mathbf{z}}'$. We then set

$$\begin{aligned}\tilde{\mathbf{m}}_i &= \mathbf{c}_{i,2}-f^{-1}\mathbf{B}_2\cdot\tilde{\mathbf{r}}_i \text{ for } i\in[\tau],\\ \tilde{\mathbf{m}}' &= \mathbf{c}'_2-f^{-1}\mathbf{B}_2\cdot\tilde{\mathbf{r}}',\end{aligned}$$

and just as in the proof of Lemma 5.5 we get that $(\mathbf{c}_i,\tilde{\mathbf{m}}_i,\tilde{r}_i,f)$ and $(\mathbf{c},\tilde{\mathbf{m}}',\tilde{r}',f)$ will all be valid openings, which is what we want. But we also need to show that $\mathbf{m}'=\sum_{i=1}^{\tau}\alpha_i\mathbf{m}_i+\beta$. This we do directly as follows

$$\begin{aligned}\sum_{i=1}^{\tau}\alpha_i\tilde{\mathbf{m}}_i+\beta &= \sum_{i=1}^{\tau}\alpha_i\left(\mathbf{c}_{i,2}-\left(d-\bar{d}\right)^{-1}\mathbf{B}_2\left(\mathbf{z}_i-\bar{\mathbf{z}}_i\right)\right)+\beta\\ &=\sum_{i=1}^{\tau}\alpha_i\mathbf{c}_{i,2}-\left(d-\bar{d}\right)^{-1}\left(\sum_{i=1}^{\tau}\alpha_i\mathbf{B}_2\mathbf{z}_i-\sum_{i=1}^{\tau}\alpha_i\mathbf{B}_2\bar{\mathbf{z}}_i\right)+\beta\\ &=\sum_{i=1}^{\tau}\alpha_i\mathbf{c}_{i,2}-\left(d-\bar{d}\right)^{-1}\left(d-\bar{d}\right)\left(\sum_{i=1}^{\tau}\alpha_i\mathbf{c}_{i,2}+\beta-\mathbf{c}'_2\right)\\ &\quad-\left(d-\bar{d}\right)^{-1}\left((\mathbf{u}-\mathbf{u})+\mathbf{B}_2\right)(\mathbf{z}'-\bar{\mathbf{z}}')+\beta\\ &=\mathbf{c}'_2-\left(d-\bar{d}\right)^{-1}\mathbf{B}_2\left(\mathbf{z}'-\bar{\mathbf{z}}'\right)=\tilde{\mathbf{m}}'.\end{aligned}$$

Where the third equality comes from the fourth accepting requirement. Then we have proven that $(\tilde{\mathbf{m}}_1,...,\tilde{\mathbf{m}}_\tau,\tilde{\mathbf{r}}_1,...,\tilde{\mathbf{r}}_\tau,\{f\}_{i=1}^{\tau},f)$ is a witness for s. Therefor we have that Π_{Lin^+} has the special soundness property.

To prove the HVZK property, we let $\mathcal{S}$ be an algorithm that start by choosing $\{\mathbf{z}_i\}_{i=1}^{\tau},\mathbf{z}'\xleftarrow{\$}\mathcal{N}_\sigma^k$ and $d\xleftarrow{\$}\mathcal{C}$, then we set

$$\begin{aligned}\mathbf{t}_i &= \mathbf{B}_1\cdot\mathbf{z}_i-d\mathbf{c}_{i,1} \text{ for each } i\in[\tau],\\ \mathbf{t}' &= \mathbf{B}_1\cdot\mathbf{z}'-d\mathbf{c}'_1,\\ \mathbf{u} &= \sum_{i=1}^{\tau}\alpha_i\mathbf{B}_2\cdot\mathbf{z}_i-\mathbf{B}_2\cdot\mathbf{z}'-\left(\sum_{i=1}^{\tau}\alpha_i\mathbf{c}_{i,2}+\beta-\mathbf{c}'_2\right)d\end{aligned}$$

We see that u will just look like a random value, and form Lemma 4.6 the $\mathbf{z}_i$'s and $\mathbf{z}'$ chosen by $\mathcal{S}$ will be statistically indistinguishable from those computed in Π_{Lin^+}.

Then we have proven that Π_{Lin^+} is a Σ-protocol, and it is, therefore, an HVZK proof. $\square$

CHAPTER 6

Zero-Knowledge proof of Correct Shuffle

This chapter is where we will study the two existing protocols for a correct shuffle. In the first Section, we will start by proving a proposition regarding the invertibility of multiple messages chosen uniformly. This will be used in the first protocol for shuffle. In the second Section we will make a change in the commitment scheme, so we can use the first protocol on messages that are vectors of lattice elements.

1. Correct shuffle of Messages in R_p

Onwards we will denote $[\![\mathbf{x}]\!]$ as the commitment $Com(\mathbf{x}; \mathbf{r_x})$ of an element $\mathbf{x} \in R_p^\ell$. In this section we will use $\ell = 1$, since the protocol Π_{Shuffle} given in Figure 6.1 only works then. This is because Π_{Shuffle} works with messages that are invertible, and this only make sense when we talk about lattice elements, not vectors.

A ZK proof of correct shuffle is where the public input is a set of commitments $\{ \mathbf{c}_i \}_{i\in[\tau]}$ and messages $\{ \hat{m}_i \}_{i\in[\tau]}$, where the prover wants to convince the verifier that the set of commitments opens to the set of messages in some order, and that it knows the opening of all of them. The ZK proof will be of the relation R_{Shuffle}, which is given as

$$R_{\mathsf{Shuffle}} = \left\{ (s, w) \;\middle|\; \begin{array}{c} s = (\mathbf{c}_1, ..., \mathbf{c}_\tau, \hat{m}_1, ..., \hat{m}_\tau, \mathbf{B}_1, \mathbf{b}_2), \\ w = (\pi, \mathbf{r}_i, ..., \mathbf{r}_\tau, f_1, ..., f_\tau), \pi \in S_\tau, \\ \mathsf{Open}(\mathbf{c}_i, \hat{m}_{\pi(i)}, \mathbf{r}_i, f_i) = 1 \ \forall i \in [\tau] \end{array} \right\}.$$

We first state the following property of the commitment scheme and give an easy proof of it.

Proposition 6.1. *Let $\mathbf{c} = Com(m; \mathbf{r})$ be a commitment with opening $(m; \mathbf{r}, f)$ and let $\mathbf{c}_\rho = Com(\rho; 0)$. Then $\mathbf{c} - \mathbf{c}_\rho$ is a commitment with opening $(m - \rho; \mathbf{r}, f)$.*

Proof. First we note that $||r_i||_2 \leq 4\sigma\sqrt{N}$ for each i, since $\mathbf{r}$ is part of a valid opening of $\mathbf{c}$. We also compute

$$f \cdot (\mathbf{c} - \mathbf{c}_\rho) = f\mathbf{c} - f\mathbf{c}_\rho = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{b}_2 \end{bmatrix} \cdot \mathbf{r} + f \cdot \begin{bmatrix} \mathbf{0}^n \\ m \end{bmatrix} + f \cdot \begin{bmatrix} \mathbf{0}^n \\ \rho \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{b}_2 \end{bmatrix} \cdot \mathbf{r} + f \cdot \begin{bmatrix} \mathbf{0}^n \\ m - \rho \end{bmatrix}.$$

And by this we have that $(m - \rho; \mathbf{r}, f)$ is a valid opening of $\mathbf{c} - \mathbf{c}_\rho$. □

We will denote $[\![m]\!] - \rho$ to be the commitment $Com(m - \rho; \mathbf{r}_m)$.

In the protocol, we need to have the property that given a $\rho \xleftarrow{\$} R_p$, then $m_i - \rho$ will be invertible for all the messages m_i. This will be true with a huge probability by the following proposition.

Proposition 6.2. *Let $N \geq \delta \geq 1$ be powers of 2, and p a prime such that $p \equiv 2\delta + 1 \pmod{4\delta}$. Then*

$$\Pr_{x_1,\ldots,x_\tau,\rho \xleftarrow{\$} R_p}[x_1 - \rho, ..., x_\tau - \rho \text{ invertible in } R_p] \geq 1 - \min(1, \tau \cdot (1 - e^{-\delta/p})).$$

Proof. Start by letting S_i be the set of $\rho \in R_p$ such that $m_i - \rho$ is non-invertible. From Lemma 4.2 we know that $X^N + 1$ factors into δ irreducible polynomials, which means that there will be

$$\left(p^{N/\delta - 1} \cdot (p-1)\right)^\delta = \left(p^{N/\delta} - p^{N/\delta - 1}\right)^\delta$$

invertible elements in R_p. Then we get that $|S_i| = p^N - \left(p^{0N/\delta} - p^{N/\delta-1}\right)^\delta$, and by treating the $S_i's$ as disjoint subsets we get a bound of there union as $|S_1 \cup ... \cup S_\tau| \leq \tau \cdot p^N - \tau \cdot \left(p^{N/\delta} - p^{N/\delta-1}\right)^\delta$. Now by dividing by the size of R_p we get a bound of the probability of picking a ρ such that at least one of $m_i - \rho$ is non-invertible which is

$$\begin{aligned}
\tau \cdot \frac{p^N - \left(p^{N/\delta} - p^{N/\delta-1}\right)^\delta}{p^N} &= \tau - \tau \cdot \left(\frac{p^{N/\delta} - p^{N/\delta-1}}{p^{N/\delta}}\right)^\delta \\
&= \tau \cdot \left(1 - \left(1 - \frac{1}{p}\right)^\delta\right) \\
&\leq \tau \cdot \left(1 - e^{-\delta/p}\right),
\end{aligned}$$

where the last inequality follows from that $1 + x \leq e^x$. We take the minimum of this and 1 to not get larger probability then 1, and subtracting it form 1 to get the complement which is all of them being invertible. □

Usual parameters for the protocol will be where $\tau \approx 1000000$, $\delta = 2$ and $p \approx 2^{32}$. Then we can se that the probability of all of the $x_i - \rho$ being invertible is almost equal to 1.

In the protocol Π_{Shuffle} given in Figure 6.1 the verifier will pick a $\rho \xleftarrow{\$} R_p$ and send it to the prover, then both computes $\hat{M}_i = \hat{m}_i - \rho$ and the prover computes $M_i = m_i - \rho$, then the prover will convince the verifier that $\prod_{i=1}^\tau M_i = \prod_{i=1}^\tau \hat{M}_i$. We will see that this is sufficient to prove the relation R_{Shuffle}. To prove this $\mathcal{P}$ will start by picking $\theta \xleftarrow{\$} R_p$ and then computing $[\![D_i]\!]$ as follows

$$\begin{aligned}
&[\![D_1]\!] = [\![\theta_1 \hat{M}_1]\!], \\
&[\![D_i]\!] = [\![\theta_{i-1} M_i + \theta_i \hat{M}_i]\!] \quad \text{for } i \in [\tau - 1] \backslash \{1\}, \\
&[\![D_\tau]\!] = [\![\theta_{\tau-1} M_\tau]\!].
\end{aligned} \tag{11}$$

Π_{Shuffle}

Public input:

$\mathbf{B} = \begin{bmatrix}\mathbf{B}_1\\ \mathbf{b}_2\end{bmatrix}, \{[\![m_i]\!]\}_{i=1}^{\tau} \in R_p, \{\hat{m}_i\}_{i=1}^{\tau} \in S_\beta^k$

Provers input:

$\{m_i\}_{i=1}^{\tau}, \{\mathbf{r}_i\}_{i=1}^{\tau}$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
		$\rho \xleftarrow{\$} R_p \backslash \{\hat{m}_i\}_{i=1}^{\tau}$
	$\xleftarrow{\rho}$	
$\hat{M}_i = \hat{m}_i - \rho$		$\hat{M}_i = \hat{m}_i - \rho$
$M_i = m_i - \rho$		$[\![M_i]\!] = [\![m_i]\!] - \rho$
$\theta_i \xleftarrow{\$} R_p, \forall i \in [\tau - 1]$		
$\forall i \in [\tau - 1]$ commpute $[\![D_i]\!]$ as in (11)		
	$\xrightarrow{\{[\![D_i]\!]\}_{i=1}^{\tau}}$	
		$\beta \xleftarrow{\$} R_p$
	$\xleftarrow{\beta}$	
$\forall i \in [\tau - 1]$ commpute s_i as in (12)		
	$\xrightarrow{\{s_i\}_{i=1}^{\tau-1}}$	
		Use Π_{Lin} to prove that 1: $\beta[\![M_1]\!] + s_1\hat{M}_1 = [\![D_1]\!]$ 2: $\forall i \in [\tau - 1]\backslash\{1\}$, $s_{i-1}[\![M_i]\!] + s_i\hat{M}_i = [\![D_i]\!]$ 3: $s_{\tau-1}[\![M_\tau]\!] + (-1)^\tau\beta\hat{M}_\tau = [\![D_\tau]\!]$ Accept iff all the instances of Π_{Lin} are accepted

FIGURE 6.1. Zero-Knowledge Proof of Shuffle

The verifier then chose a challenge $\beta \xleftarrow{\$} R_p$, sends it to the prover, then the prover computes s_i in the following way

$$s_i = (-1)^i \cdot \beta \prod_{j=1}^{i} \frac{M_j}{\hat{M}_j} + \theta_i \tag{12}$$

Finely the prover uses Π_{Lin} to prove that D_i, M_i and $\hat{M}_i$ are of the forms in (13), (14) and (15). So to prove the completeness of Π_{Shuffle} comes down to proving the following lemma.

Lemma 6.3. *If $m_1, ..., m_\tau$ is a permutation of $\hat{m}_1, ..., \hat{m}_\tau$ and $M_i - \rho$ is ivertible for all $i \in [\tau]$, then the s_i's computed as in (12) satisfy the following equations*

$$\beta M_1 + s_1\hat{M}_1 = \theta_1\hat{M}_1, \tag{13}$$

$$s_{i-1}M_i + s_i\hat{M}_i = \theta_{i-1}M_i + \theta_i\hat{M}_i \quad \text{for } i \in [\tau - 1]\backslash\{1\}, \tag{14}$$

$$s_{\tau-1}M_\tau + (-1)^\tau\beta\hat{M}_\tau = \theta_{\tau-1}M_\tau. \tag{15}$$

PROOF. We prove this by checking one equation at the time. We start with (13).

$$\beta M_1 + s_1 \hat{M}_1 = \beta M_1 + \left(\theta_1 - \beta \frac{M_1}{\hat{M}_1}\right) \cdot \hat{M}_1 = \theta_1 \hat{M}_1.$$

Now we continue with (14),

$$\begin{aligned} s_{i-1} M_i + s_i \hat{M}_i &= \left((-1)^{i-1} \cdot \beta \prod_{j=1}^{i-1} \frac{M_j}{\hat{M}_j} + \theta_{i-1}\right) \cdot M_i \\ &\quad + \left((-1)^i \cdot \beta \prod_{j=1}^{i} \frac{M_j}{\hat{M}_j} + \theta_i\right) \cdot \hat{M}_i \\ &= \theta_{i-1} M_i + \theta_i \hat{M}_i. \end{aligned}$$

We observe that when multiplying in the M_i and $\hat{M}_i$ in the equation above, we get the extra M_i in the first product and cancel $\hat{M}_i^{-1}$ in the second, which makes the products differ by sign since the second product has one higher exponent of -1.

Finally we verify (15),

$$s_{\tau-1} M_\tau + (-1)^\tau \beta \hat{M}_\tau = \left((-1)^{\tau-1} \cdot \beta \prod_{j=1}^{\tau-1} \frac{M_j}{\hat{M}_j} + \theta_{\tau-1}\right) \cdot M_\tau + (-1)^\tau \beta \hat{M}_\tau$$

Since the M_i's are a permutation of the $\hat{M}_i$'s, the product will include the inverse of all M_i except the one that is equal to $\hat{M}_\tau$, which means that $M_\tau \cdot \prod_{j=1}^{\tau-1} \frac{M_j}{\hat{M}_j} = \hat{M}_\tau$. So we get that

$$s_{\tau-1} M_\tau + (-1)^\tau \beta \hat{M}_\tau = (-1)^{\tau-1} \beta \hat{M}_\tau + \theta_{\tau-1} M_\tau + (-1)^\tau \beta \hat{M}_\tau = \theta_{\tau-1} M_\tau.$$

□

Lemma 6.4. *Assume the commitment scheme is binding and* Π_{Lin} *given in Figure 5.3 is a sound proof of knowledge of* R_{Lin} *except with probability* t*. Then* Π_{Shuffle} *given in Figure 6.1 is a sound proof of knowledge of* R_{Shuffle} *except with probability* $\epsilon \leq \frac{\tau^\delta + 1}{|R_p|} + 4\tau t$*.*

PROOF. We first create an extractor $\mathcal{E}$ that given an instance s of Π_{Shuffle} and black-box access to $\mathcal{P}^*$ will create a witness w such that $(s, w) \in R_{\mathsf{Shuffle}}$. We devide $\mathcal{E}$ in τ subextractors $\mathcal{E}_i$ which will work as follows.

1. Run instances with arbitrary randomness tape for $\mathcal{P}^*$ as well as arbitrary challenges until an accepted transcript is collected.
2. Rewind $\mathcal{P}^*$ until after the first message in the ith instance of Π_{Lin} and search for a second challenge that gives an accepted transcript of Π_{Lin}.

$\mathcal{E}$ then set $m_i = M_i - \rho_i$, where ρ_i is the value used by $\mathcal{E}_i$ and M_i the message extracted. If $\{m_i\}_{i=1}^\tau$ is a permutation of $\{\hat{m}_i\}_{i=1}^\tau$ $\mathcal{E}$ outputs $w = (\tau, \mathbf{r}_1, ..., \mathbf{r}_\tau, f_1, ..., f_\tau)$.

The proof that $\mathcal{E}$ works in polynomial time is stated by Aranha et al. [**1**] that it can be done by a usual heavy-row argument, which we exclude.

Now assume we have two accepted transcripts of $\mathcal{P}^*$ with the same ρ and different challenges β and β', then we also have values s_i and s'_i such that the following equalities hold by the soundness of Π_{Lin}.

1. $\beta M_1 + s_1\hat{M}_1 = D_1 = \beta' M_1 + s_1\hat{M}_1$.
2. $s_{i-1}M_i + s_i\hat{M}_i = D_i = s'_{i-1}M_i + s'_i\hat{M}_i$ for $i \in [\tau - 1]\backslash\{1\}$.
3. $s_{\tau-1} + (-1)^\tau\beta\hat{M}_i = D_\tau = s'_{\tau-1} + (-1)^\tau\beta'\hat{M}_i$.

By subtracting the right side from each equations we can express this in the following form:

$$\begin{bmatrix} M_1 & \hat{M}_1 & 0 & \dots & 0 & 0 \\ 0 & M_2 & \hat{M}_2 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & M_{\tau-1} & \hat{M}_{\tau-1} \\ (-1)^\tau\hat{M}_\tau & 0 & 0 & \dots & 0 & M_\tau \end{bmatrix} \cdot \begin{bmatrix} \beta - \beta' \\ s_1 - s'_1 \\ \vdots \\ s_{\tau-2} - s'_{\tau-2} \\ s_{\tau-1} - s'_{\tau-1} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}.$$

Inductively since $M_i, \hat{M}_i$ and $\beta - \beta'$ are non-zero all the $s_i - s'_i$ will also be non-zero. If we now call the matrix to the left for $\mathbf{M}$ and the non-zero vector $\mathbf{c}$ we get that $\mathbf{M} \cdot \mathbf{c} = \mathbf{0}$. Since $\mathbf{c}$ is a non-zero vector we know from linear algebra that $\det(\mathbf{M}) = 0$. The determinant of M can easily be computed to

$$\det(\mathbf{M}) = \prod_{i=1}^{\tau} M_i + (-1)^{2\tau-1}\prod_{i=1}^{\tau}\hat{M}_i = \prod_{i=1}^{\tau} M_i - \prod_{i=1}^{\tau}\hat{M}_i.$$

We can now define the two polynomials $g(X) = \prod_{i=1}^{\tau}(m_i - X)$ and $\hat{g}(X) = \prod_{i=1}^{\tau}(\hat{m}_i - X)$. Then we see that $\det(\mathbf{M}) = 0$ if and only if $g(\rho) = \hat{g}(\rho)$. If the m_i's are a permutation of $\hat{m}_i$'s then $g(X) = \hat{g}(X)$ so we are done. If not, $g(X) - \hat{g}(X)$ will be a polynomial of degree at least τ so by Lemma 4.3 it will have at most τ^δ zeros. Therefor we can see that with less then $\frac{\tau^\delta+1}{|R_p|}$ probability the $\hat{m}_i$'s will be a permutation of the m_i's. □

Theorem 6.5. *Assuming that $\tau \ll |R_p|$. If the commitment scheme is hiding, and Π_{Lin} is an HVZK proof, then Π_{Shuffle} given in Figure 6.1 is an HVZK proof of the relation R_{Shuffle}.*

Proof. First since $\tau \ll |R_p|$ we get form Proposition 6.2 that it is high probability to pick a ρ which makes each M_i invertible. Then the completeness property of Π_{Shuffle} follows from Lemma 6.3 and the completeness of Π_{Lin}.

The soundness property follows from Lemma 6.4

We will prove the HVZK property by looking at the series of games given in Figure 6.2:

Π_{Shuffle}	Game 1	Game 2	Simulator
1: $\theta_i \xleftarrow{\$} R_p$	1: $\theta_i \xleftarrow{\$} R_p$	1: $s_i \xleftarrow{\$} R_p$	1: $[\![D_i]\!] \leftarrow [\![0]\!]$
2: $D_i \leftarrow (11)$	2: $D_i \leftarrow (11)$	2: $\theta_i \xleftarrow{\$} R_p$	2: $s_i \xleftarrow{\$} R_p$
3: Send $[\![D_i]\!]$	3: Send $[\![D_i]\!]$	3: $D_i \leftarrow (16)$	3: 3: Send $[\![D_i]\!]$
4: $s_i \leftarrow (12)$	4: $s_i \leftarrow (12)$	4: Send $[\![D_i]\!]$	4: Send s_i
5: Send s_i	5: Send s_i	5: Send s_i	5: Sim Π_{Lin}
6: Π_{Lin}	6: Sim Π_{Lin}	6: Sim Π_{Lin}	

FIGURE 6.2. Games used to prove the hovest-verifier zero-knowledge property of Π_{Shuffle}. We use the notation $x \leftarrow (i)$ to denote that x is computed as i equation (i).

First we note that the only difference between Π_{Shuffle} and Game 1 is that in Game 1 we simulate Π_{Lin} instead of ruining it. So from the HVZK property of Π_{Lin} we get that Π_{Shuffle} and Game 1 are statistical indistinguishable.

When proving that Game 1 and Game 2 are statistical indistinguishable we note that the way s_i are computed in (12) are statistical indistinguishable from uniformly random picked elements of R_p since when computing s_i we add θ_i which are picked uniformly at random. We also see that in Game 1 we compute D_i as in (11) which will be a random element. And so will also he D_i witch we compute as

$$\begin{aligned} D_1 &= \beta M_1 + s_1 \hat{M}_1, \\ D_i &= s_{i-1} M_i + s_i \hat{M}_i \ i \in [\tau - 1] \backslash \{1\}, \\ D_\tau &= s_{\tau-1} M_\tau + (-1)^\tau \beta \hat{M}_\tau. \end{aligned} \tag{16}$$

Then since the s_i's and D_i's are the only things that are different in the two games, which both just look like random elements, the two games are statistically indistinguishable.

Finely we show that given an adversary $\mathcal{A}$ that can distinguish Game 2 and Simulator, then we can construct an algorithm $\mathcal{A}'$ that can break the hiding property of our commitment scheme.

First we let a commitment oracle for each $i \in [\tau]$ pick a random message $m_i \xleftarrow{\$} R_p$ and $\mathbf{r}_i \xleftarrow{\$} S_\beta^k$ then compute $\mathbf{c}_{0,i} \leftarrow Com(0, \mathbf{r_i})$ and $\mathbf{c}_{1,i} \leftarrow Com(m_i, \mathbf{r_i})$. The oracle then picks a bit $b \xleftarrow{\$} \{0,1\}$ and sends $\{\mathbf{c}_{b,i}\}_{i=1}^\tau$ to $\mathcal{A}'$. $\mathcal{A}'$ then pick $s_i \xleftarrow{\$} R_p$ and $\beta \xleftarrow{\$} R_p$ and sends $\{\mathbf{c}_{b,i}\}_{i=1}^\tau, \{s_i\}_{i=1}^\tau$ and to $\mathcal{A}$ and receives a bit b' which $\mathcal{A}'$ outputs.

We note that $\mathcal{A}$ receiving $\mathbf{c}_{1,i}$ will be indistinguishable from receiving $[\![D_i]\!]$ computed as in (16) since the D_i's just looks like randomly picked elements.

From these three arguments we get that the Simulator and Π_{Shuffle} are statistical indistinguishable and Π_{Shuffle} is therefor HVZK proof of R_{Shuffle}. □

2. Correct Shuffle of Encrypted Messages

So we have a nice protocol for proving a shuffle of messages, but usually, we want to have a shuffle of encrypted messages. The problem with Π_{Shuffle} is that it only works for messages in R_p, but ciphertexts of lattice points are usually in R_p^2. So in this section, we will introduce a protocol that uses Π_{Shuffle} but works for messages in R_p^ℓ.

We will do this by using Π_{Shuffle} on $\boldsymbol{\rho}\cdot\mathbf{m}_i$ for all i. Where $\boldsymbol{\rho} = (1, h, h^2, ..., h^{\ell-1})$ for a $h \xleftarrow{\$} R_p$ picked by the verifier $\mathcal{V}$. We alsa change the comitmets $[\![\mathbf{m}_i]\!] = \begin{bmatrix}\mathbf{c}_{i,1}\\ \mathbf{c}_{i,2}\end{bmatrix}$ to $[\![\boldsymbol{\rho}\cdot\mathbf{m_i}]\!] = \begin{bmatrix}\mathbf{c}_{i,1}\\ \boldsymbol{\rho}\cdot\mathbf{c}_{i,2}\end{bmatrix}$ by changeing the public key $pk = \mathbf{B} = \begin{bmatrix}\mathbf{B}_1\\ \mathbf{B}_2\end{bmatrix}$ to

$$pk' = B' = \begin{bmatrix}\mathbf{B}_1\\ \boldsymbol{\rho}\cdot\mathbf{B}_2\end{bmatrix} = \begin{bmatrix}\mathbf{B}_1\\ \mathbf{b}_2'\end{bmatrix} \tag{17}$$

We clearly see that this does not make the hiding property of the commitment weaker since the attacker could have done this in the attack. However, we prove that this new pk does not change anything about the binding property.

Proposition 6.6. *If a PPT attacker $\mathcal{A}$ can brake the binding property of the commitment scheme using pk' as in (17), then there exists a PPT attacker $\mathcal{A}'$ that can brake the binding property of our original commitment scheme.*

Proof. Assume $\mathcal{A}$ outputs $(m_1, \mathbf{r}_1, f_1)$ and $(m_2, \mathbf{r}_2, f_2)$ which both are valid openings for a commitment $\begin{bmatrix}\mathbf{c}_1\\ \boldsymbol{\rho}\cdot\mathbf{c}_2\end{bmatrix} = \begin{bmatrix}\mathbf{c}_1\\ c_2'\end{bmatrix}$ where $m_1 \neq m_2$. Then we have that $f_1 c_2' = \mathbf{b}_2'\cdot\mathbf{r}_1 + f_1\cdot m_1$ and $f_2 c_2' = \mathbf{b}_2'\cdot\mathbf{r}_2 + f_2\cdot m_2$. Then by multiplying the first one with f_2 and the second with f_1 and subtracting we get

$$0 = \mathbf{b}_2'\cdot(f_2\mathbf{r}_1 - f_1\mathbf{r}_2) + f_1 f_2\,(m_1 - m_2)$$

Since $m_1 \neq m_2$ we then know that $\mathbf{b}_2'\cdot(f_2\mathbf{r}_1 - f_1\mathbf{r}_2) \neq 0$. We then set $\mathbf{m_1} = \mathbf{c_2} - f_1^{-1}\mathbf{B}_2\mathbf{r}_1$ and $\mathbf{m_2} = \mathbf{c_2} - f_2^{-1}\mathbf{B}_2\mathbf{r}_2$. Then we can see that $(\mathbf{m}_1, \mathbf{r}_1, f_1)$ and $(\mathbf{m}_2, \mathbf{r}_2, f_2)$ are valid openings by

$$\mathbf{B}_2\cdot\mathbf{r}_i + f_i\mathbf{m}_i = \mathbf{B}_2\cdot\mathbf{r}_i + f_i\left(\mathbf{c_2} - f_i^{-1}\mathbf{B}_2\mathbf{r}_i\right) = f_i\cdot\mathbf{c}_i$$

for $i \in \{1, 2\}$. Now the only thing left is to prove that $\mathbf{m}_1$ and $\mathbf{m}_2$ are distinct. This we do by assuming they are equal and multiply the expression of them by $f_1 f_2$ and subtracting so we get

$$\mathbf{0} = \mathbf{B}_2\,(f_2\mathbf{r}_1 - f_1\mathbf{r}_2)\,.$$

$\Pi^{\ell}_{\mathsf{Shuffle}}$

Public input:

$\mathbf{B} = \begin{bmatrix}\mathbf{B}_1\\ \mathbf{B}_2\end{bmatrix}$, $\{[\![\mathbf{m}_i]\!]\}_{i=1}^{\tau}$, $\{\hat{\mathbf{m}}_i\}_{i=1}^{\tau}$

Provers input:

$\{\mathbf{m}_i\}_{i=1}^{\tau} \in R_p^{\ell}, \{\mathbf{r}_i\}_{i=1}^{\tau} \in S_\beta^k, \pi \in S_\tau$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
		$h \xleftarrow{\$} R_p$
		$\boldsymbol{\rho} \leftarrow (1, h, h^2, ..., h^{\ell-1})$
	$\xleftarrow{\boldsymbol{\rho}}$	
Run Π_{Shuffle} with $\mathcal{V}$ to prove that $\{[\![\boldsymbol{\rho} \cdot \mathbf{m}_i]\!]\}_{i=1}^{\tau}$ is a permutation of $\{\boldsymbol{\rho} \cdot \hat{\mathbf{m}}_i\}_{i=1}^{\tau}$ with the same π and $\mathbf{B}' = \begin{bmatrix}\mathbf{B}_1\\ \boldsymbol{\rho} \cdot \mathbf{B}_2\end{bmatrix}$.		
		Accept iff Π_{Shuffle} returns accept.

FIGURE 6.3. Zero-Knowledge Proof of Shuffle of messages $\mathbf{m}_i \in R_p^{\ell}$

If then multiply from left with $\boldsymbol{\rho}$ we get

$$0 = \mathbf{b}_2' \left(f_2 \mathbf{r}_1 - f_1 \mathbf{r}_2 \right),$$

which is a contradiction, so $\mathbf{m}_1 \neq \mathbf{m}_2$. □

We are then ready to introduce the protocol that will be an HVZK proof of the following relation

$$R^{\ell}_{\mathsf{Shuffle}} = \left\{ (s, w) \;\middle|\; \begin{array}{c} s = (\mathbf{c}_1, ..., \mathbf{c}_\tau, \hat{\mathbf{m}}_1, ..., \hat{\mathbf{m}}_\tau, \mathbf{B}_1, \mathbf{B}_2), \\ w = (\pi, \mathbf{r}_i, ..., \mathbf{r}_\tau, f_1, ..., f_\tau), \pi \in S_\tau, \\ \mathsf{Open}(\mathbf{c}_i, \hat{\mathbf{m}}_{\pi(i)}, \mathbf{r}_i, f_i) = 1 \; \forall i \in [\tau] \end{array} \right\},$$

and then prove that it is an HVZK proof.

Theorem 6.7. *If Π_{Shuffle} given in Figure 6.3 is an HVZK proof of the relation R_{Shuffle} with soundness error ϵ', then $\Pi^{\ell}_{\mathsf{Shuffle}}$ is an HVZK proof of the relation $R^{\ell}_{\mathsf{Shuffle}}$ with soundness error $\epsilon = 2\epsilon' + 3\left(\frac{\ell-1}{p}\right)^N$.*

PROOF. First, we see that both the completeness and HVZK properties follow from these properties of Π_{Shuffle}.

To show the soundness property, we will assume there exist a prover $\mathcal{P}^*$ that can given an input x, with probability $\nu > \epsilon$, convince the verifier to output accept. We construct an extractor $\mathcal{E}$ that by calling on $\mathcal{P}^*$ will construct a witness w for x.

When constructing this $\mathcal{E}$, we remember that we have an extractor $\mathcal{E}'$ of Π_{Shuffle} and use this to construct $\mathcal{E}$, which will go as follows and restart every time it aborts.

1. Run random instances of $\Pi^{\ell}_{\mathsf{Shuffle}}$ until a valid instance with challenge h is generated. Do this at most $2/\epsilon$ times, otherwise, abort.
2. Run $\mathcal{E}'$ with h fixed by $\mathcal{P}^*$ until it outputs $\pi, \{ \mathbf{r}_i, f_i \}_{i\in[\tau]}$. If $\mathcal{E}'$ aborts then abort and in parallel start a new loop until $\mathcal{E}'$ finishes.
3. Let $\tilde{\mathbf{m}}_i = (f_i \mathbf{c}_{2,i} - \mathbf{B}_2 \mathbf{r}_i) \cdot f_i^{-1}$. If $\tilde{\mathbf{m}}_{\pi(i)} = \hat{\mathbf{m}}_i$ for all $i \in [\tau]$ then output $\pi, \{ \mathbf{r}_i, f_i \}_{i\in[\tau]}$ otherwise abort.

First, we notice that since the f_i's are invertible, the $\tilde{\mathbf{m}}_i$'s are well-defined. And if $\mathcal{E}$ outputs something, then it is a valid witness. We continue to prove that each loop is run in polynomial time with constant probability for $\mathcal{E}$ to output something. We do this with a heavy-row argument.

In the first step, we expect to find an accepting transcript after less than $1/\epsilon$ iterations. Since we run it $2/\epsilon$ iterations, the probability of not aborting is more than $1/2$. Let $\mathbf{H}$ be the binary matrix with a row for each h and columns for choices in Π_{Shuffle}. An entry of $\mathbf{H}$ is 1 if $\mathcal{P}^*$ would convince $\mathcal{V}$ and 0 otherwise. We say a row of $\mathbf{H}$ is heavy if it contains more than $\epsilon/2 > \epsilon'$ 1's. Then by a counting argument, more than half of the 1's will be in a heavy row, and the probability of hitting a h that is a heavy row is then greater than $1/2$. If we hit a heavy row, then we know $\mathcal{E}'$ will output a valid witness for Π_{Shuffle} in polynomial time by assumption, and we are then done. Once $\mathcal{E}'$ outputs something, then Step 3 is inexpensive and is omitted.

We can also assume that $\mathcal{E}$ outputs the same opening with a probability of $1/2$ in at least $2/3$ of the heavy rows. If not, it can easily be proven that we can break the binding property of the commitment scheme by Proposition 6.6.

We also show that there must be more then $\frac{3}{2}\left(\frac{\ell-1}{p}\right)^N$ heavy rows by assuming otherwise. Assume there are exactly $\frac{3}{2}\left(\frac{\ell-1}{p}\right)^N$ where each of these contains only 1's and all other rows contains $\epsilon/2$ 1's such that we have the maximum of 1's possible without more then $\frac{3}{2}\left(\frac{\ell-1}{p}\right)^N$ heavy rows. Then we get that the probability for $\mathcal{P}^*$ to convince $\mathcal{V}$ will be

$$\frac{3}{2}\left(\frac{\ell-1}{p}\right)^N + \left(1 - \frac{3}{2}\left(\frac{\ell-1}{p}\right)^N\right) \cdot \frac{\epsilon}{2} < \frac{3}{2}\left(\frac{\ell-1}{p}\right)^N + \frac{\epsilon}{2} < \epsilon.$$

This is a contradiction, so the number of heavy rows must be more than $\frac{3}{2}\left(\frac{\ell-1}{p}\right)^N$.

Now assume that $\mathcal{E}'$ extract a valid witness $\pi, \{ \mathbf{r}_i, f_i \}_{i\in[\tau]}$ for $\{ [\![\boldsymbol{\rho} \cdot \mathbf{m}_i]\!] \}_{i\in[\tau]}$ and messages $\{ \boldsymbol{\rho} \cdot \hat{\mathbf{m}}_i \}_{i\in[\tau]}$, but the extracted $\tilde{\mathbf{m}}_i$'s is not a permutation of the $\hat{\mathbf{m}}_i$'s. Then there exists a $i \in [\tau]$ such that

$$f_i \cdot \left(\boldsymbol{\rho} \cdot \mathbf{c}_{2,\pi(i)}\right) = \boldsymbol{\rho} \cdot \mathbf{B}_2 \cdot \mathbf{r}_i + f_i \cdot \left(\boldsymbol{\rho} \cdot \hat{\mathbf{m}}_i\right),$$

but

$$f_i \cdot \mathbf{c}_{2,\pi(i)} = \mathbf{B}_2 \cdot \mathbf{r}_i + f_i \cdot (\hat{\mathbf{m}}_i + \boldsymbol{\delta})$$

where $\tilde{\mathbf{m}}_i = \hat{\mathbf{m}}_i + \boldsymbol{\delta}$ for a non-zero vector $\boldsymbol{\delta} = \begin{bmatrix} \delta_0 \\ \vdots \\ \delta_{\ell-1} \end{bmatrix}$. Combining both equations we get that $\mathbf{0} = \boldsymbol{\rho} \cdot \boldsymbol{\delta}$ which is the same as saying that the polynomial $\delta(X) = \sum_{i=0}^{\ell-1} \delta_i X^i$ is zero evaluated at h. From Lemma 4.3 $\delta(X)$ can have a maximum of $(\ell-1)^d < (\ell-1)^N$ number of roots, but the transcript is extraditable and thus acceptable for strictly more then $(\ell-1)^N$ different choices of h. Therefor $\boldsymbol{\delta}$ must be zero, which means that Step 3 only aborts if $\mathcal{E}'$ extracts a witness $\pi, \{ \mathbf{r}_i, f_i \}_{i \in [\tau]}$ for R_{Shuffle} that is not the "default one", which only happens with constant probability. □

CHAPTER 7

Zero-Knowledge proofs using Permutation Matrices

In the last chapter, we looked at a protocol for an HVZK proof of shuffle of messages in R_p and then used it to create a shuffle for messages in R_p^ℓ. In this chapter, we will start on a try to prove the shuffle of messages using permutation matrices. We will also discuss ways we could go forward in creating a proof of correct shuffle using permutation matrices if we had the necessary theory.

1. A Naive Proof of Permutation Matrix

We start by introducing what a permutation matrix is and give a really bad proof of a matrix being a permutation matrix.

Definition 7.1. *A square matrix $\mathbf{A} = (a_{i,j})$ is a permutation matrix if all the rows and columns sum up to 1 and $a_{i,j} \in \{0, 1\}$ for each i and j.*

Our strategy in this section will be to prove that a matrix $\mathbf{A}$ is a permutation matrix by directly following the definition. We will first prove that each matrix element is either 0 or 1 and then prove that each row and column sums up to 1. But first, we introduce an HVZK proof of a message being one of two values $\mathbf{m}_0$ and $\mathbf{m}_1$.

Theorem 7.2. *Assume the binding property of our commitment scheme and that Π_{Commit} is an HVZK proof. Then Π_{Or} given in Figure 7.1 is an HVZK proof of the relation*

$$R_{\mathsf{Or}} = \left\{ (s, w) \;\middle|\; \begin{array}{c} s = (\mathbf{c}, \{\mathbf{m}_0, \mathbf{m}_1\}, \mathbf{B}_1, \mathbf{B}_2), \\ w = (b, \tilde{\mathbf{r}}, f), b \in \{0, 1\}, \\ \mathsf{Open}(\mathbf{c}, \mathbf{m}_b, \tilde{\mathbf{r}}, f) = 1 \end{array} \right\}.$$

Proof. Completeness and HVZK both follow directly from the completeness and HVZK property of Π_{Commit}. So we only need to prove the special soundness property.

Let $(\mathbf{t}_0, \mathbf{t}_1, \beta, \beta_0, \beta_1, \mathbf{z}_0, \mathbf{z}_1)$ and $(\mathbf{t}_0, \mathbf{t}_1, \beta', \beta'_0, \beta'_1, \mathbf{z}'_0, \mathbf{z}'_1)$ be two valid transcripts of Π_{Or} where $\beta \neq \beta'$. Then just as in the proof of Theorem 5.7 we get two valid openings of $\mathsf{Open}(\mathbf{c}, \mathbf{m}_0, \mathbf{z}_0 \tilde{-} \mathbf{z}'_0, \beta_0 - \beta'_0)$ and $\mathsf{Open}(\mathbf{c}, \mathbf{m}_1, \mathbf{z}_1 \tilde{-} \mathbf{z}'_1, \beta_1 - \beta'_1)$. But this breaks the binding property of the commitment scheme unless $\mathbf{z}_{1-b} = \mathbf{z}'_{1-b}$ and $\beta_{1-b} = \beta'_{1-b}$ for a bit b. So by assumption we will have such a bit and set the

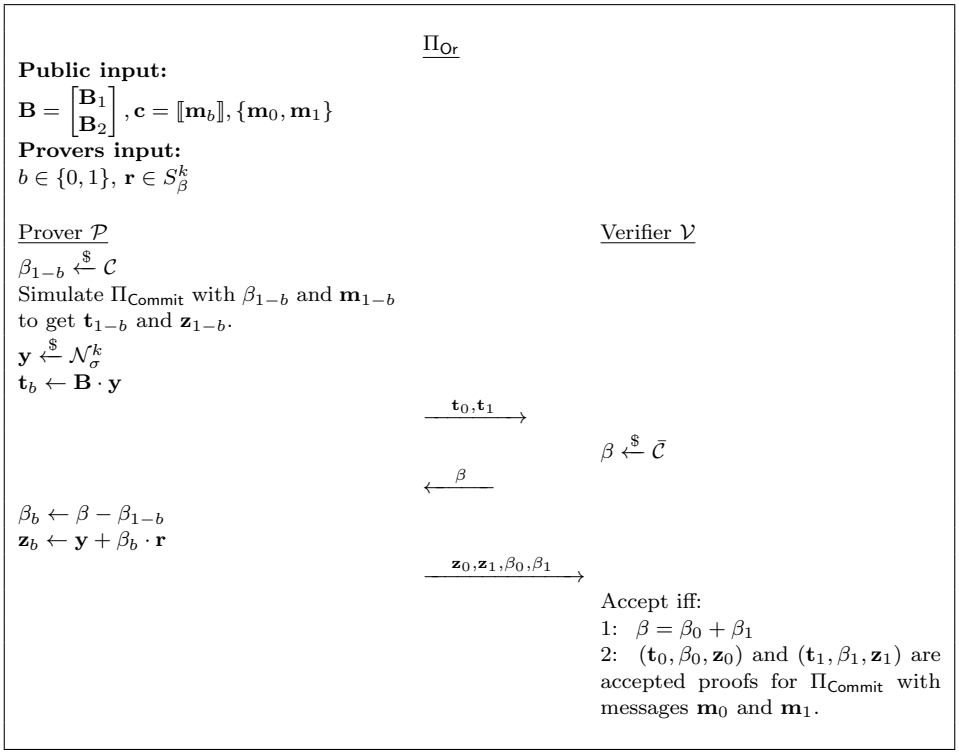


FIGURE 7.1. Zero-Knowledge Proof of committing to one of two messages $\mathbf{m}_0$ or $\mathbf{m}_1$.

witness to be $w = (b, \mathbf{z}_b - \mathbf{z}'_b, \beta_b - \beta'_b)$.

Π_{Or} is therefor a Σ-protocol, and therefor an HVZK proof of R_{Or} by Theorem 3.9. □

Then we are ready to give an HVZK proof of a matrix A being a permutation matrix by proving the followin relation.

$$R_{\mathsf{PermMat}} = \left\{ (s,w) \;\middle|\; \begin{array}{c} s = (\mathbf{C} = (\mathbf{c}_{i,j}), \mathbf{B}_1, \mathbf{B}_2),\ i,j \in [\tau], \\ w = (\mathbf{A} = (a_{i,j}), \mathbf{R} = (\tilde{\mathbf{r}}_{i,j}), \mathbf{F} = (f_{i,j})), \\ a_{i,j} \in \{0,1\}, \sum_{k=1}^{\tau} a_{k,j} = \sum_{k=1}^{\tau} a_{i,k} = 1\ \forall i,j \in [\tau], \\ \mathsf{Open}(\mathbf{c}_{i,j}, a_{i,j}, \tilde{\mathbf{r}}_{i,j}, f_{i,j}) = 1\ \forall i,j \in [\tau] \end{array} \right\}.$$

Permutation protocol. Here, we give the steps in the protocol Π_{PermMat} to prove that A is a permutation matrix:

1. The prover $\mathcal{P}$ and the verifier $\mathcal{V}$ are given a $\tau \times \tau$-matrix of commitments $\mathbf{C} = (\mathbf{c}_{i,j})$ and the public key to the commitment scheme $\mathbf{B} = \begin{bmatrix}\mathbf{B}_1\\ \mathbf{B}_2\end{bmatrix}$. $\mathcal{P}$ is also given $\mathbf{A} = (a_{i,j}) \in \{0,1\}^{\tau \times \tau}$ and $\mathbf{R} = (\mathbf{r}_{i,j}) \in S_\beta^{\tau \times \tau}$.

2. $\mathcal{P}$ uses Π_{Or} to prove that each $\mathbf{c}_{i,j}$ opens up to either 0 or 1.
3. In a spaning tree fashion $\mathcal{P}$ uses Π_{Lin^+} $\tau \log \tau$ times on each row and column to prove that the commitment of there sums are $\mathbf{c}_1, ..., \mathbf{c}_{2\tau}$. Then $\mathcal{P}$ use Π_{Commit} to prove that each of these commitments opens up to 1.
4. $\mathcal{V}$ accepts if and only if all of the proofs returns accept.

We notice that in Step 3 of Π_{PermMat}, the prover does not use Π_{Lin^+} to prove the linear relation directly of all the τ elements in each row and column. This is because the probability of not aborting is, as we see by (10) exponential decreasing with τ, and since we usually want to use $\tau \approx 1000000$, $\mathcal{P}$ in Π_{Lin^+} would almost never not abort.

Theorem 7.3. *Assuming that* $\Pi_{\mathsf{Or}}, \Pi_{\mathsf{Commit}}$ *and* Π_{Lin^+} *are HVZK proofs, then* Π_{PermMat} *is an HVZK proof of the relation* R_{PermMat}.

Proof. We start by noticing that both the completeness and HVZK properties follow directly from these properties in $\Pi_{\mathsf{Or}}, \Pi_{\mathsf{Commit}}$ and Π_{Lin^+}.

The soundness property will with an extractor that rewinds on each of the protocols, and a heavy-row argument also follows from the soundness of the same tree ZK-proofs. □

2. Further work

Now we have a ZK proof for proving that a matrix A is a permutation matrix, the problem with this is that it is in no way usable since it is way too slow as well as it does not prove the shuffle of messages. Furukawa and Sako [**5**] give another protocol for proving this property by using an equivalent definition of permutation matrices and multi commitments. However, this protocol is working over cyclic groups, where schemes for multi commitments exist.

They talk about a proof that the ElGamal cipher texts $E_1, ..., E_\tau$ are a permutation of $E'_1, ..., E'_\tau$. These cipher texts are of the form $E_i = (g_i, m_i)$, where $g_i, m_i \in \mathbb{Z}_p^*$ are elements of order q. Where q and p primes such that $p = q \cdot k + 1$ for an integer k. The E'_i are the commitments of the E_j of the form $E_i = (g'_i, m'_i) = \left(g^{r_i} \cdot g_{\pi(i)}, y^{r_i} \cdot m_{\pi(i)}\right)$ for publicly known g and $y = g^X$, a secret $X \in \mathbb{Z}_p$ and a set $\{r_i\}$. But since this commitment scheme works for multi commitments and they work with permutation matrices they can prove that

$$E'_i = (g'_i, m'_i) = \left(g^{r_i} \prod_{i=1}^{\tau} g_i^{A_{ij}}, y^{r_i} \prod_{i=1}^{\tau} m_i^{A_{ij}} \right), \tag{18}$$

where $\mathbf{A} = (A_{ij})$ is a permutation matrix. In this ZK proof they use the following theorem, which gives an equivalent definition of permutation matrices.

Theorem 7.4. *A matrix* $\mathbf{A} = (A_{ij})_{(i,j=1,\dots,\tau)}$ *is a permutation matrix if and only if for all* i, j *and* k *both*

$$\sum_{h=1}^{\tau} A_{hi}A_{h,j} = \begin{cases} 1, & \text{if } i = j \\ 0, & \text{if } i \neq j \end{cases} \tag{19}$$

$$\sum_{h=1}^{\tau} A_{hi}A_{hj}A_{hk} = \begin{cases} 1, & \text{if } i = j = k \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

hold.

Note that Furukawa and Sako [**5**] state the Theorem for permutation matrices over $\mathbb{Z}_p$, however, the Theorem does hold in general.

The ZK proof is divided into four parts. So they do it by proving the following.

1. Given $\{g_i\}$ and $\{g'_i\}$. The $\{g'_i\}$ can be expressed as in (18) with integers $\{r_i\}$ and a matrix $\mathbf{A}$ such that 19 holds.
2. Given $\{g_i\}$ and $\{g'_i\}$. The $\{g'_i\}$ can be expressed as in (18) with integers $\{r_i\}$ and a matrix $\mathbf{A}$ such that 20 holds.
3. The $\{r_i\}$ and $\mathbf{A}$ used in the two above proofs are the same.
4. For each pair (g'_i, m'_i) the same $\{r_i\}$ and $\mathbf{A}$ are used.

They combine these fore proofs into one single protocol such that they do not have multiple conversations.

If we had a multi commitment scheme for lattice commitments, we could maybe do something similar to give a ZK proof of shuffle. If we assume the messages $\mathbf{m}_i = \begin{bmatrix} m_{i,1} \\ m_{i,2} \end{bmatrix}$ still got put into one commitment $\mathbf{c}_i$ then we could omit the fourth proof, such that the proof would go by proving the following tree statements.

1. Given $\{\mathbf{m}_i\}$ and $\{\mathbf{c}_i\}$. The $\{\mathbf{c}_i\}$ are multi commitments of the $A_{ij} \cdot \mathbf{m}_{\mathbf{j}\in[\tau]}$ with vectors $\{\mathbf{r}_i\}$ and a matrix $\mathbf{A}$ such that 19 holds.
2. Given $\{\mathbf{m}_i\}$ and $\{\mathbf{c}_i\}$. The $\{\mathbf{c}_i\}$ are multi commitments of the $A_{ij} \cdot \mathbf{m}_{\mathbf{j}\in[\tau]}$ with vectors $\{\mathbf{r}_i\}$ and a matrix $\mathbf{A}$ such that 20 holds.
3. The $\{\mathbf{r}_i\}$ and $\mathbf{A}$ used in the two above proofs are the same.

Where we again would combine these three proofs into one conversation.

Unfortunately, we do not have a notion of multi commitments using lattices of this scale. One could first think of using multi commitments by just adding up messages so that we commit to $\mathbf{m}_1 + \mathbf{m}_2$, but then the commitment scheme would either be not at all binding since we could just find two different messages adding up to the same sum as $\mathbf{m}_1$ and $\mathbf{m}_2$. Or we would need to give a proof like Π_{Lin^+} from Figure 5.4, which we have seen dos not work with large values for τ.

A second multi commitment scheme one could think of is just by increasing ℓ, but then both the size of the public matrix and the commitments would be too big. The size of these could decrease by doing as we did for $\Pi^{\ell}_{\mathsf{Shuffle}}$, by taking the inner product of $\mathbf{m}$ with $\boldsymbol{\rho}$. However, we have not had time to look at this and therefore do not know if this would be sufficient.

Bibliography

[1] Diego F. Aranha, Carsten Baum, Kristian Gjøsteen, Tjerand Silde, Thor Tunge: Lattice-Based Proof of Shuffle and Applications to Electronic Voting. CT-RSA 2021: 227-251

[2] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, Chris Peikert: More Efficient Commitments from Structured Lattice Assumptions. SCN 2018: 368-385

[3] Damgård, I.: On σ-protocols, `https://cs.au.dk/~ivan/Sigma.pdf`

[4] Diego F. Aranha, Xavier Boyen, Thomas Haines and Johannes Mueller: A Verifiable and Practical Lattice-Based Decryption Mix Net with External Auditing. Cryptology ePrint Archive, Report 2020/115, `https://ia.cr/2020/115`

[5] Jun Furukawa and Kazue Sako. An efficient scheme for proving a shuffle. Inproceedings of CRYPTO '01, LNCS series, volume 2139, pages 368–387, 2001.

[6] Jun Furukawa. Efficient and verifiable shuffling and shuffle-decryption. IEICE Transactions, 88-A(1):172–188, 2005.

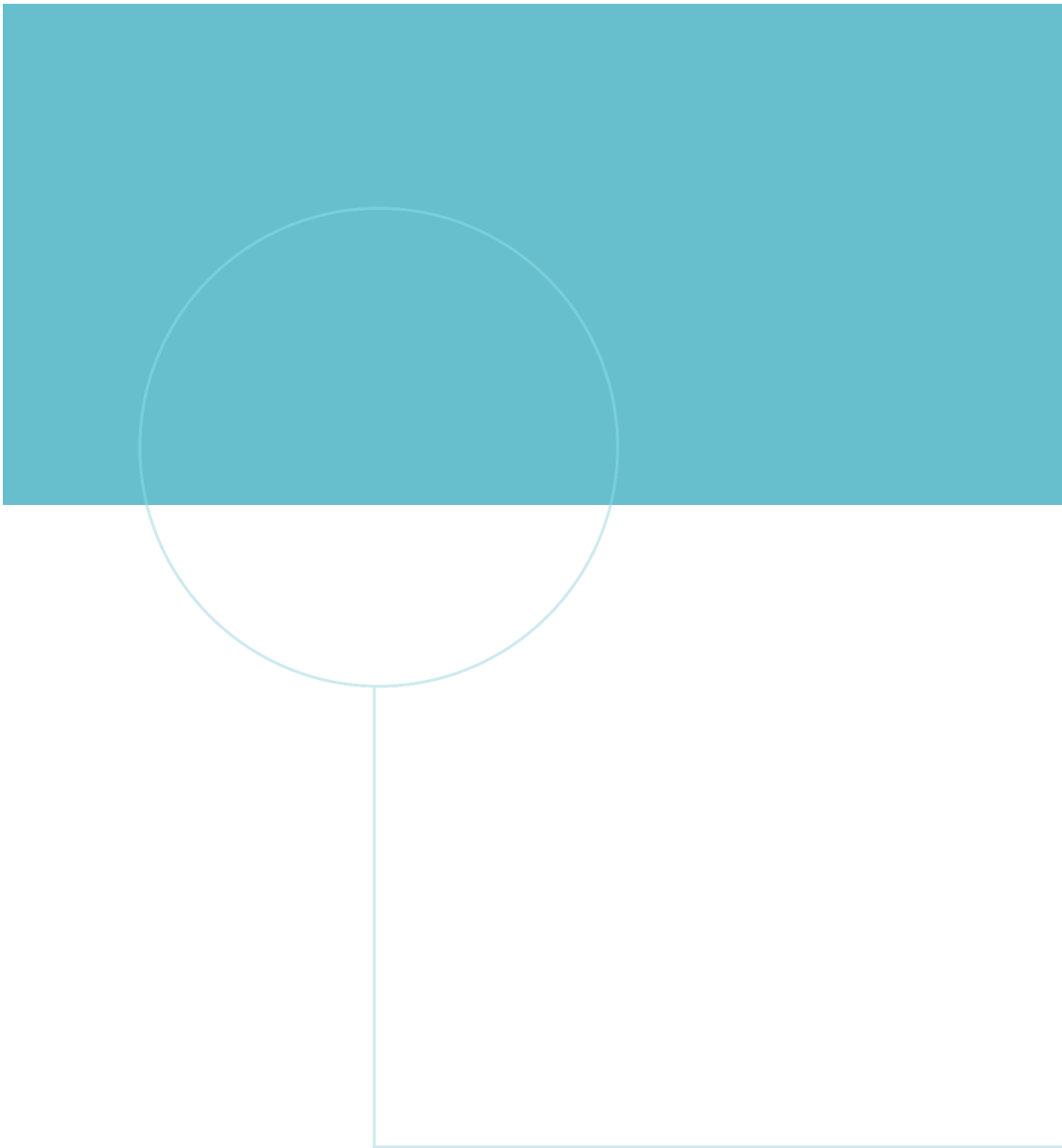
[7] R. Impagliazzo, L. A. Levin, and M. Luby Pseudo-random generation from one-way functions. In Proceedings of the twenty-first annual ACM symposium on Theory of computing (STOC '89) pages 12-24, 1989

[8] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, ASIACRYPT 2009, volume 5912 of LNCS, pages 598–616. Springer, Heidelberg, December 2009.

[9] Vadim Lyubashevsky and Gregor Seiler. Short, invertible elements in partially splitting cyclotomic rings and applications to lattice-based zero-knowledge proofs. In Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, TelAviv, Israel, April 29 - May 3, 2018 Proceedings, Part I, pages 204–224, 2018.

[10] Vadim Lyubashevsky. Lattice signatures without trapdoors. In Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings, pages 738–755, 2012.

[11] Neff, C.A.: A verifiable secret shuffle and its application to e-voting. In: ACM CCS 2001. ACM Press, November 2001



NTNU

Kunnskap for en bedre verden