Master's thesis

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

Sunniva Engan

Compressed Σ-Protocols from Different Underlying Assumptions

Master's thesis in Mathematical Sciences
Supervisor: Kristian Gjøsteen
Co-supervisor: Tjerand Silde
June 2025

NTNU
Norwegian University of
Science and Technology

Sunniva Engan

Compressed Σ-Protocols from Different Underlying Assumptions

Master's thesis in Mathematical Sciences
Supervisor: Kristian Gjøsteen
Co-supervisor: Tjerand Silde
June 2025

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

NTNU
Norwegian University of
Science and Technology

Abstract

A Σ-protocol is a well-established cryptographic primitive that allows a prover to convince a verifier in a 3-move protocol that it is in possession of some secret information (witness) that satisfies a given statement, without revealing anything beyond the claim being valid. This thesis gives an introduction to fundamental Σ-protocol theory and how to construct compressed Σ-protocols, both in the discrete logarithm (dlog) setting and from lattice assumptions.

We begin by proving in two different ways why every Σ-protocol is a proof of knowledge. We then present how Attema and Cramer (CRYPTO 2020) constructed compressed Σ-protocols in the dlog setting, something which allows us to reduce the communication complexity of certain Σ-protocols from linear to logarithmic in the size of the witness. By a motivating example of a Schnorr-like Σ-protocol in the lattice setting, we emphasize different challenges and pitfalls that arise when adapting dlog-based protocols to the lattice setting, and how to overcome them. We then show how Attema, Cramer and Kohl (CRYPTO 2021) made the appropriate changes to the given compression mechanism to make it compatible with lattice assumptions. Lastly, we study how it can be instantiated using the Module Short Integer Solution (MSIS) assumption to achieve poly-logarithmic communication complexity in the size of the witness.

Sammendrag

En Σ-protokoll er en veletablert kryptografisk primitiv som lar en beviser overtale en verifiserer i en 3-bevegelsesprotokoll at den besitter hemmelig informasjon (vitne) som tilfredsstiller et gitt utsagn, uten å røpe noe mer enn at påstanden er sann. Denne masteroppgaven gir en introduksjon til grunnleggende Σ-protokollteori og hvordan å konstruere komprimerte Σ-protokoller, både ved diskret logaritme (dlog)-antakelsen og fra gitterbaserte antakelser.

Vi begynner med å bevise på to forskjellige måter hvorfor enhver Σ-protokoll er et bevis av kunnskap. Deretter presenterer vi hvordan Attema og Cramer (CRYPTO 2020) konstruerte komprimerte Σ-protokoller fra dlog-antakelsen, noe som lar oss redusere kommunikasjonskompleksiteten i visse Σ-protokoller fra lineær til logaritmisk i størrelsen på vitnet. Gjennom et motiverende eksempel med en Schnorr-lignende Σ-protokoll bygget fra gitterbaserte antakelser, vektlegger vi forskjellige utfordringer og fallgruver som oppstår når man tilpasser dlog protokoller til å bli gitterbaserte, samt hvordan man kan overvinne dem. Deretter viser vi hvordan Attema, Cramer og Kohl (CRYPTO 2021) gjorde passende endringer til den dlog-baserte kompresjonsmekanismen for å gjøre den forenlig med gitterbaserte antakelser. Til slutt ser vi på hvordan den kan bli instansiert med den såkalte *Module Short Integer Solution* (MSIS) antakelsen for å oppnå polylogaritmisk kommunikasjonskompleksitet i størrelsen på vitnet.

United Nations Sustainable Development Goals

The 17 Sustainable Development Goals by the United Nations (UN) seek to make every country of UN take actions for necessary measures towards peace and prosperity for everyone [Uni25c]. In this thesis, one of the goals have been to make an accessible resource to learn about fundamental Σ-protocol theory, as well as present compressed Σ-protocol theory in such a way that the reader is equipped to learn even more about the research area later on. Providing accessible and high quality cryptography resources allow equal opportunities for learning about security measures, something which is in varying degrees important for everyone to learn more about.

In addition to this, guaranteeing that there are post-quantum cryptographically secure algorithms and systems ensures that crucial infrastructure will remain secure in the time to come. Therefore to some extent, this thesis promotes the fourth and tenth Sustainable Development Goals, namely Quality Education [Uni25a] and specifically goal 16.10 under Peace, Justice and Strong Institutions [Uni25b].

Acknowledgments

This thesis in cryptography concludes my Master's in Mathematical Sciences at the Norwegian University of Science and Technology (NTNU). First and foremost I would like to thank my supervisors Tjerand Silde and Kristian Gjøsteen for all the great guidance you have given me over the past year. Thank you Kristian for your helpful comments and advice. Thank you Tjerand for our weekly meetings, your explanations to all of the questions I have had have helped me immensely. I have learned a lot from our discussions, and I thank you for believing in me and encouraging my work, even when I felt like my progress was slow.

I am deeply grateful to Jiaxin Pan for convincing me to pursue cryptography from early on, and for all the interesting discussions we have had. I would also like to thank all the wonderful people in the Aarhus Crypto Group for welcoming me warmly during my exchange year. Thank you Marius for being an amazing office buddy (Norway is clearly a small place compared to Denmark).

Thank you "Mamma og Harald", for always supporting me and sending me lots of Eira pictures. Thank you Pernille, for helping me not take life too seriously all the time. Thank you Eivind, for always being there for me, whether it is supporting me academically or emotionally. Thank you all, for reminding me when I get stressed that there is more to life than the next upcoming deadline. Farfar, I wish you were here to see how far I have come.

Lastly, I would not be who I am today without the incredible people I have gotten to know through Matteland and the dancing community in Trondheim. I am grateful for all the invaluable friendships I have made over the past 6 years. I would especially like to thank all the amazing people I have met and danced with in NTNUI Swing og Folkedans. Every Sunday has been filled to the brim with endless swing dancing.

Sunniva Engan
Trondheim, May 2025

Contents

1 Introduction

Zero-knowledge proofs (ZKPs) have ever since they were introduced [GMR85] been of central significance in the construction of a wide variety of protocols. It allows a party often referred to as the prover to convince another party often referred to as the verifier that some publicly known statement is true, and doing so without revealing anything beyond the validity of the claim. There exist several types of ZKPs, where a proof of knowledge (PoK) [BG93] is a special type of such proofs, where the prover sets out to convince the verifier that it is in possession of some secret information referred to as a witness that satisfies a given statement, without revealing anything beyond the claim that is made is true. A Σ-protocol [Cra97] is a special type of ZKP that is also a PoK, where the interaction between the prover and the verifier only consists of three moves. Due to their simplicity and desirable properties, they are being deployed in a wide range of applications such as identification schemes [Dam10; Oka93; Sch90], anonymous credentials [BL13], electronic voting [Ara+23] and especially in constructing digital signatures [Sch91], where the Fiat-Shamir heuristic [FS87] gives us that any Σ-protocols can be turned into one.

1.1 Context

Due to the rise quantum computers, all systems based on the hardness of the discrete logarithm (dlog) assumption or factoring will be deemed insufficiently secure due to the famously known Shor's algorithm [Sho94]. As a result, all systems built on these assumptions have to be replaced with others built on assumptions that remain hard. The new post-quantum secure systems should ideally cover the same functionalities as the previous ones have, or at least similar ones. In particular, many of the constructions for the applications that have been mentioned so far will be deemed insecure, and would have to be replaced.

NIST One of the driving forces behind the transition to post-quantum secure cryptographic constructions are the several calls made by the National Institute of Science and Technology (NIST), with the goal of standardizing different kinds of post-quantum secure systems [Nat25a]. Σ-protocols play a central role in many of the constructions that have been proposed, where one in particular that has become a standard is the lattice-based digital signature algorithm CRYSTALS-Dilithium [Nat24]. Other candidates that also make use of Σ-protocols that are still under consideration [Nat25b], such as the isogeny-based digital signature SQIsign [Aar+25]. Furthermore, Σ-protocols can also be helpful when constructing protocols in threshold cryptography for distributing trust, to ensure that all participating parties in a given protocol behave honestly [Nat25c].

Some of the proposed systems are built on lattice-based cryptography, where two hardness assumption in particular are used to construct a wide range of protocols. These are referred to as the Learning With Errors (LWE) problem, proposed by Regev in 2005 [Reg05], and the Short Integer Solution (SIS) problem, proposed by Ajtai in 1996 [Ajt96]. There are different version of the assumptions that have emerged over the years, such as the ring LWE [LPR10] and ring SIS problems [PR06], as well as the module LWE and module SIS problems [LS15]. The latter has turned out to be the more flexible and effective assumptions, and are the ones we will mostly make use of in this thesis.

Σ-Protocol Regardless of the instantiation, there are some properties that any Σ-protocol has to satisfy. *Completeness* ensures that in an honest execution between a prover with a valid witness and a verifier, the verifier will always accept. In addition to this, there are two properties that provide guarantees against both cheating provers and verifiers. *Special soundness* ensures that a cheating prover cannot convince a verifier that a statement is true more than once, and if it would be able to do so it would possess a valid witness for the given statement. In addition to this, we say that a verifier should not learn anything beyond that the prover has a witness for the given statement, since it has a variant of zero-knowledge called *special honest-verifier zero-knowledge*. However to do so, there has to be made a choice of what it means to learn something.

Simulation Paradigm: One way to formalize what it means to learn something is through the simulation paradigm, where we consider that if information Y can be efficiently computed from information X, someone receiving information Y learns nothing more than information X [Gol01, Section 4.3]. In the context of ZKPs, this means that a verifier should not learn anything more by having interacted with the prover, than what it could have derived from the common statement and messages it according to the protocol sends to the prover. We refer to the work by Lindell for a more detailed discussion [Lin16].

1.2 Challenges

Reducing communication and round complexity of interactive protocols is ever relevant in cryptographic research. Doing so helps to continuously improve on existing ideas and systems, while it can also ensure a wider range of future applications. Over the past years, several attempts at constructing practical ZKPs have emerged from a variety of different underlying assumptions. In 1993, Stern proposed a ZKP based on syndrome-decoding for error-correcting codes [Ste94], however the number of rounds in the scheme depended on the desired security level. The reason for this was the high probability of a cheating prover succeeding in convincing the verifier of the

validity of the statement, so the proof had to be repeated a considerable amount of times to be deemed secure. More recent work have tried to build on Stern's ideas, by switching out underlying assumptions with lattice problems [KTX08] and further optimize the proof system [Lin+13].

Bulletproofs In 2018, Bünz et al. introduced Bulletproofs [Bün+18], which are ZKPs based on the dlog assumption, with proof size logarithmic in the size of the witness. The ideas used to achieve this were however tailor-made to their construction. For public parameters $\mathbf{g}, \mathbf{h} \in \mathbb{G}^n$, the relation they provided an efficient proof for can be described as

$$\begin{aligned} \mathcal{R}_{\text{bullet}} = \Big\{ \big(X = (P \in \mathbb{G}, u \in \mathbb{Z}_q),\ w = (\mathbf{a}, \mathbf{b} \in \mathbb{Z}_q^n)\big) \\ : P = \mathbf{g}^{\mathbf{a}}\mathbf{h}^{\mathbf{b}} \wedge u = \langle \mathbf{a}, \mathbf{b} \rangle \Big\}, \end{aligned} \tag{1}$$

where X denotes the common input that both a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ has access to, while w denotes the witness, namely the secret value that $\mathcal{P}$ want to convince $\mathcal{V}$ that it knows. The relation describes that a prover wants to convince a verifier that they possess an opening to a given commitment, that also satisfies a dot product constraint.

Because of the desirable proof size, extending the ideas of Bulletproofs to a wider range of statements encouraged researcher to attempt to construct a more general framework, specifying where the Bulletproofs compression techniques could be applied. This led to the rise of compressed Σ-protocol theory by Attema and Cramer [AC20], where they gave a theoretical framework for Σ-protocols in the dlog setting that could reduce the proof size of protocols that satisfy similar statements. In particular, they managed to give a framework where a prover could prove knowledge of an opening to a Pedersen vector commitment [Ped92], that also satisfies an extension of the dot product constraint, namely any linear constraint.

Their framework was however not compatible with assumptions other than the dlog problem, but studying how to construct systems for some underlying assumptions that may be easier to work with, can in some cases give an indication of how central ideas that can be extended to other underlying assumptions as well. Further research led to Attema, Cramer and Kohl to develop an even more generic framework that would be compatible with particularly the lattice setting [ACK21].

Due to size constraints of secret vectors in the lattice setting, protocols built from these assumptions can often be slower and more involved than those constructed in the dlog setting. A reason for this, is that leaking values dependent on the secret that have norm significantly higher that the secret can leak information, and we therefore sometimes would have to abort the protocol if this were to be the case. Furthermore, there may be a gap between values that can be extracted from the protocols and those we can prove knowledge of, which means we cannot achieve exact proofs without introducing more involved techniques.

1.3 Outline

The main goal of this thesis is to give an introduction to fundamental Σ-protocol theory and show how to construct compressed Σ-protocols, both in the dlog and lattice setting. In Section 2, we establish the necessary background material needed to cover the topics that are not specific to the techniques in a given section. We give an introduction to polynomial rings and the negative hypergeometric distribution, as well as central concepts in cryptography such as how to define security, hash functions, commitment schemes and protocols.

In Section 3, we define Σ-protocols, and study how to generically prove OR of two statements, given that the challenge space for the Σ-protocol we work with has a well-defined group operation. We show in two different ways why every Σ-protocol is a proof of knowledge, where we first look at the classical proof by Damgård [Dam10], and then an alternative version by Attema, Cramer and Kohl [AC20]. We end with an introduction of the Fiat-Shamir transformation [FS87], that allows us to turn any Σ-protocol into a non-interactive protocol or a digital signature.

In Section 4, we introduce the compressed Σ-protocol theory framework in the dlog setting [AC20]. It consists of several intermediate protocols we gradually compose to achieve the desired functionality. We introduce the Pedersen vector commitment scheme [Ped92], and then give the basic Σ-protocol used for composition. The last message in the Σ-protocol can be swapped out with a PoK for a statement that it satisfies, and we describe a protocol that it can be replaced with. We then present the compression mechanism, that we compose the previously mentioned protocols with until we have reduced the size of the last message the desired amount.

In Section 5, we define the challenge set necessary for adapting the Schnorr Σ-protocol [Sch90] to the lattice setting, as well as familiarize ourselves with the relevant underlying hardness assumptions in lattice-based cryptography, namely the LWE and SIS problems. We also give an intuition for how to choose the parameters involved in the assumptions to ensure that the problems are deemed sufficiently secure. Then in Section 6, we give an example of how the Schnorr Σ-protocol [Sch90] can be altered to be built from the MSIS assumption [Lyu24], and also show how we can instantiate the OR-proof when using the given protocol.

In Section 7, we present the framework created for extending compressed Σ-protocol to the lattice setting [ACK21]. As before, we introduce intermediate protocols that we gradually compose to obtain the final compressed Σ-protocol. After doing so, we look at how the framework can be instantiated when using the MSIS assumption, by using a compact lattice-based commitment scheme [Ajt96] that grants us poly-logarithmic communication complexity [ACK21]. We conclude with a discussion on possible directions for future work that builds on the theory we have covered.

2 Background

We begin by establishing some notation and basic definitions. For a set $\mathcal{S}$, let $s \leftarrow\!\!\$\ \mathcal{S}$ denote sampling an element s uniformly at random from it. For an algorithm $\mathcal{A}$, let $a \leftarrow\!\!\$\ \mathcal{A}(k)$ denote that $\mathcal{A}$ is a probabilistic algorithm that upon input k produces the result a. For any $\beta \in \mathbb{Z}^+$, we define the set

$$[\beta] := \{-\beta, \ldots, -1, 0, 1, \ldots, \beta\},$$

and we let $[\beta]^{n \times m}$ be the set of $n \times m$ matrices whose entries are all elements of $[\beta]$. Let $\mathbb{E}[X]$ denote the expectation of a random variable X.

Definition 2.1 (Dual Space)**.** Let V be a vector space over a field K. The dual space of V is the vector space consisting of all K-linear maps from V to K, which can be written as

$$\mathcal{L}(V) = \{(L : V \to K) : L \text{ is a } K\text{-linear map}\}. \tag{2}$$

2.1 Polynomial Rings

A generalization of the polynomial ring $\mathbb{Z}[X]$ with indeterminate X, is the ring $\mathcal{R}_f = \mathbb{Z}[X]/(f(x))$ where $f \in \mathbb{Z}[X]$ is a monic polynomial of degree d. Elements in $\mathcal{R}_f$ can be uniquely represented by polynomials of the form $a = \sum_{i=0}^{d-1} a_i X^i$, where $a_i \in \mathbb{Z}$. Addition of two polynomials in $\mathcal{R}_f$ is done by adding their corresponding coefficients, and multiplication is performed by multiplying the polynomials and then reducing the result modulo $f(X)$.

One can also define the polynomial ring

$$\mathcal{R}_{q,f} = \mathbb{Z}_q[X]/\left(f(X)\right), \tag{3}$$

where the coefficients are chosen from $\mathbb{Z}_q$ instead.

Matrix Notation Let $a, b \in \mathcal{R}_f$. Polynomial multiplication of the two polynomials mod f can be rewritten as matrix multiplication, by observing that

$$ab \mod f = a\left(\sum_{i=0}^{d-1} b_i X^i\right) \mod f = \sum_{i=0}^{d-1} \left(aX^i \mod f\right) b_i. \tag{4}$$

Coefficients of any polynomial $a = \sum_{i=0}^{d-1} a_i X^i \in \mathcal{R}_f$ can be represented as a vector in $\mathbb{Z}^d$ as

$$\mathcal{V}_a := \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{d-1} \end{bmatrix} \in \mathbb{Z}^d. \tag{5}$$

For a vector $\mathbf{a} \in \mathcal{R}_f^n$, we can similarly also define $\mathcal{V}_\mathbf{a}$ as

$$\mathcal{V}_\mathbf{a} := \begin{bmatrix} \mathcal{V}_{a_1} \\ \vdots \\ \mathcal{V}_{a_n} \end{bmatrix} \in \mathbb{Z}^{dn}. \tag{6}$$

Using the notation from Equation 5, we can express the coefficients of $\sum_{i=0}^{d-1}(aX^i \mod f)$ by

$$\mathcal{M}_a := \begin{bmatrix} \mathcal{V}_a & \mathcal{V}_{aX \mod f} & \dots & \mathcal{V}_{aX^{d-1} \mod f} \end{bmatrix} \in \mathbb{Z}_q^{d \times d}. \tag{7}$$

With this notation, we can rewrite $ab \mod f = c \mod f$ as $\mathcal{M}_a \cdot \mathcal{V}_b = \mathcal{V}_c$.

2.2 Hash Functions

Definition 2.2 (Hash Functions [Sho05, Chapter 8.7])**.** Let R, S and T be finite, non-empty sets, and suppose that for each $r \in R$ we have a function $\phi_r : S \to T$. Then we say that ϕ_r is a hash function from S to T, and we refer to $r \in \mathcal{R}$ as the hash key. ■

The output set T is generally much smaller than the input set S for a hash function ϕ_r, and one of the use cases for hash functions is to ensure that input to cryptographic algorithms are of the correct size. Ideally, the output of any hash function should look as close to uniformly distributed as possible, and we want the probability of finding hash collisions to be small.

Definition 2.3 (Universal Hash Function (UHF) [Sho05, Chapter 8.7])**.** Let $\{\phi_r\}_{r \in R}$ be a family of hash functions. Let H be a random variable that is uniformly distributed over R, and for each $s \in S$ let $\phi_H(s)$ be the random variable with the value $\phi_r(s)$ when $H = r$.

We say that $\{\phi_r\}_{r \in R}$ is a Universal Hash Function (UHF) if

$$\Pr[\phi_H(s) = \phi_H(s')] \leq \frac{1}{T}$$

for all $s, s' \in S$ with $s \neq s'$. ■

To prove that a commitment scheme later on is statistically hiding, we will make use of the Leftover Hash Lemma. It is a powerful tool that allows us to make arguments for when we have a sufficient amount of randomness in an output distribution.

Definition 2.4 (Collision probability [Sho05, Chapter 8.9])**.** Let X be a random variable that takes on values in a finite set S. Then we say that the collision probability of X equals

$$\sum_{s \in S} \Pr[X = s]^2.$$

■

Theorem 2.5 (Leftover Hash Lemma [Sho05, Theorem 8.37])**.** *Let $\{\phi_r\}_{r\in R}$ be a UHF from S to T, where $m := |T|$. Let H and X be independent random variables, where H is uniformly distributed over R, and X takes values in S. If β is the collision probability of X, and δ' is the statistical distance (Definition 2.9) of $(H, \phi_H(X))$ from uniform on $R \times T$, then*

$$\delta' \leq \frac{1}{2}\sqrt{m\beta}.$$

Random Oracle Model Since one of the desirable traits of a secure hash function is for its output to look uniformly and independently distributed, modeling hash functions as truly random functions can sometimes make security proofs for protocols simpler. In the Random Oracle Model (ROM), hash functions are replaced with random oracles that are publicly known, and algorithms that have been granted oracle access to them can send it values and only learns the evaluated result. The output of a random oracle is uniformly and independently distributed, where if the same value is requested multiple times, the oracle keeps tack of previous values and send the same one corresponding to the given input [BR93]. In practice, protocols that have been proven secure in the ROM are instantiated with hash functions that are considered to have a high enough security guarantee to parallel that of a truly random function.

2.3 Defining Security

Definition 2.6 (Negligible)**.** A function $f : \mathbb{N} \to \mathbb{R}$ is said to be negligible if for all $c \in \mathbb{N}$, there exists a $\lambda_0 \in \mathbb{N}$ such that $f(\lambda) \leq 1/\lambda^c$ for all $\lambda \geq \lambda_0$. ■

Definition 2.7 (Probabilistic Polynomial Time (PPT))**.** We say that an algorithm $\mathcal{A}$ runs in PPT, if it runs in polynomial time while potentially using some internal randomness. ■

Definition 2.8 (Security Parameter)**.** The security parameter determines how much computational power it takes for an underlying assumption to be deemed insecure, and can for example be chosen to be dependent of the length of some secret input we want to run the protocol on. ■

Definition 2.9 (Statistical Distance [Sho05, Chapter 8.8])**.** Let X and Y be random variables that both take on values from a finite set S. The statistical distance between X and Y is defined as

$$SD(X, Y) = \frac{1}{2}\sum_{s\in S}|\Pr[X = s] - \Pr[Y = s]|.$$

We also denote the statistical distance by δ' when X and Y are clear from the context. ■

Definition 2.10 (Indistinguishability [DN07, Chapter 2.2])**.** Let X and Y be two probabilistic algorithms and let X_s and Y_s denote the output distributions over T generated for each input $s \in S$, when running the algorithm on input s.

- We say that X and Y are perfectly indistinguishable, written $X \sim^p Y$, if $X_s = Y_s$ for all $s \in S$.
- We say that X and Y are statistically indistinguishable, written $X \sim^s Y$, if the statistical distance $SD(X_s, Y_s) \leq 2^{-\lambda}$ is negligible in the security parameter.
- We say that X and Y are computationally indistinguishable, written $X \sim^c Y$, if no PPT adversary $\mathcal{A}$ can distinguish the following with non-negligible advantage:

  0. Algorithm X run on input s and outputs t,
  1. Algorithm Y run on input s and outputs t.

  The advantage of adversary $\mathcal{A}$ that outputs a bit guess $b \in \{0, 1\}$ is defined as

$$\begin{aligned}\mathsf{Adv}(\mathcal{A}) = |\Pr[b = 1|\ t \leftarrow\!\$\ X(s), b \leftarrow\!\$\ \mathcal{A}(s, t)] \\ - \Pr[b = 1|\ t \leftarrow\!\$\ Y(s), b \leftarrow\!\$\ \mathcal{A}(s, t)]|.\end{aligned}$$

2.4 Commitment Schemes

Commitment schemes in our case are used between a sender and a recipient. The sender has some values it wants to commit to and that it wants to reveal at a later point to the recipient, while the recipient does not learn anything about the commited value until the sender chooses to reveal it.

Commitment schemes have two essential properties called hiding and binding. Hiding ensures that the recipient cannot tell what message the sender has commited to, while binding ensures that the sender cannot at a later point claim it has committed to a different value than it initially did.

Definition 2.11 (Commitment Scheme [DN07, Chapter 2.3])**.** A commitment scheme is defined by the following PPT algorithms.

- $\mathsf{Gen}(1^\lambda)$: return public parameters pp
- $\mathsf{Com}_{\mathsf{pp}}(x, \gamma) : \mathcal{M} \times \mathcal{R} \to \mathcal{N} \quad (x, \gamma) \mapsto c$

To open a commitment c, the sender forwards the message x together with the randomness γ used, such that the recipient can verify that it corresponds to the given commitment.

Before the public parameters are established by the sender and recipient of a given instance of a commitment scheme, its validity has to be checked by each participating party to ensure that the scheme works as intended. In some cases, it is assumed that all parties have already established a common and valid public parameter pp in advance for simplicity. However if it is the case that either the sender or recipient has to generate them, the one with the strongest computational power is the one to do it. That is, if the sender has unconditional computing power, it runs Gen and sends the public parameters pp to the recipient who checks its validity, and if the recipient is the one to have unconditional computing power, they have opposite roles.

Definition 2.12 (Hiding)**.** Let $(\mathsf{Gen}, \mathsf{Com})$ define a commitment scheme as in Definition 2.11. We define the hiding game in Figure 1, where $\mathcal{A}^{\mathsf{Com}(\cdot,\cdot)}$ denotes that adversary $\mathcal{A}$ can ask an oracle for a commitment of either message m_0 or m_1.

Hiding Game:	**Oracle** $\mathsf{Com}(m_0, m_1)$: // one query
$\mathsf{pp} \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$	$\gamma \leftarrow\!\!\$\ \mathcal{R}$
$b' \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{Com}(\cdot,\cdot)}(\mathsf{pp})$	$b \leftarrow\!\!\$\ \{0,1\}$
return b'	$c \leftarrow\!\!\$\ \mathsf{Com}_{\mathsf{pp}}(m_b, \gamma)$
	return c

Figure 1: Hiding Game

The advantage of adversary $\mathcal{A}$ is defined as

$$\begin{aligned}\mathsf{Adv}_{\text{hiding}}(\mathcal{A}) = |\Pr[b' = 1 |\ c \leftarrow\!\!\$\ \mathsf{Com}_{\mathsf{pp}}(m_0, \gamma), b' \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{Com}(\cdot,\cdot)}(\mathsf{pp})] \\ - \Pr[b' = 1 |\ c \leftarrow\!\!\$\ \mathsf{Com}_{\mathsf{pp}}(m_1, \gamma), b' \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{Com}(\cdot,\cdot)}(\mathsf{pp})]|.\end{aligned} \tag{8}$$

■

Definition 2.13 (Binding)**.** Let $(\mathsf{Gen}, \mathsf{Com})$ define a commitment scheme as in Definition 2.11. The advantage of adversary $\mathcal{A}$ breaking binding is defined as

$$\begin{aligned}\mathsf{Adv}_{\text{binding}}(\mathcal{A}) = \Pr[(m, \gamma, m', \gamma', c) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pp}) \mid m \neq m' \\ \wedge\ \mathsf{Com}_{\mathsf{pp}}(m, \gamma) = \mathsf{Com}_{\mathsf{pp}}(m', \gamma') = c].\end{aligned} \tag{9}$$

■

The computational capabilities of the recipient and sender determine the flavor of hiding and binding a commitment scheme should have to grant the desired functionality in a given use case. For hiding, we can either have perfect, statistical or computational security guarantees, as defined in Definition 2.10. This depends on the power of the adversary $\mathcal{A}$, where

if it has unconditional computing power, we can have perfect hiding if its advantage is 0, and statistical hiding if its advantage is negligible. If the distinguisher is computationally bounded and its advantage is negligible, we have computational hiding. For binding, we separate between guarantees against adversaries with unconditional computing power and those who are computationally bounded, as the security relies on them not being able to come up with a message and randomness that gives the same commitment as the one they are given.

2.5 Protocols and Composition

Let $(\mathcal{P}, \mathcal{V})$ denote an interactive protocol between a prover and a verifier. Running a protocol Π on common input X and witness w will also sometimes be denoted either by $\Pi(X; w)$ or $\mathsf{Input}(X; w)$. For two protocols Π_0 and Π_1, we say that the protocol obtained by replacing the last interaction in Π_0 with Π_1 is the protocol composition of Π_0 and Π_1, and is denoted by $\Pi_1 \diamond \Pi_0$.

Definition 2.14 (Public Coin Protocol)**.** We say that an interactive protocol $(\mathcal{P}, \mathcal{V})$ is public coin if the verifier $\mathcal{V}$ chooses all its messages uniformly at random, and that they are independent from the prover's messages. ■

2.6 Negative Hypergeometric Distribution

The following is based on "A Compressed Σ-Protocol Theory for Lattices" by Attema, Cramer and Kohl, and establishes the necessary background for the alternative proof of knowledge soundness for Σ-protocols [ACK21, Sections 1.2 and 2.2]. The proof makes use of the negative hypergeometric distribution when constructing a knowledge extractor.

The negative hypergeometric distribution can be described as follows: Given a bin of N balls where M are these are marked in a particular way, we look at the distribution of the number of attempt when we want to retrieve $k \leq M$ of the marked balls, when drawing without replacement. The expected number of draws before retrieving k of the M marked balls among the N balls in total equals

$$\frac{k(N+1)}{M+1}.$$

3 Foundations of Σ-Protocols

The following section is based on Damgård's notes on Σ-protocols [Dam10], as well as "Efficient Secure Two-Party Protocols" by Hazay and Lindell [HL10, Chapter 6].[1] Let $\mathcal{R}$ be a binary relation where $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ under the restriction that if $(x, w) \in \mathcal{R}$, the length of w is bounded by some polynomial $p(|x|)$.

3.1 Defining Σ-Protocols

Definition 3.1 (Σ-Protocol)**.** We say that a protocol Π is a Σ-protocol for relation $\mathcal{R}$ if it is a three-round protocol of the form presented in Figure 2 and that it satisfies the following requirements.

- **Completeness:** If both a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ follow the protocol specifications on common input x and $\mathcal{P}$'s private input w where $(x, w) \in \mathcal{R}$, then $\mathcal{V}$ always accepts any non-aborting transcripts. The prover's private input w is often referred to as a witness. The probability of an honest protocol executions being aborted is referred to as the completeness error, and is often denoted by δ.
- **Special Soundness:** There exists a PPT algorithm $\mathcal{E}$ that given the common input x and any two accepting transcripts (a, e, z) and (a, e', z') for x where $e \neq e'$, the extractor $\mathcal{E}$ outputs a witness w such that $(x, w) \in R$.
- **Special Honest Verifier Zero-Knowledge (Special HVZK):** There exists a simulator $\mathcal{S}$ that upon input x and e outputs a transcript of the form (a, e, z) with the same probability distribution as a transcripts honestly generated between $\mathcal{P}$ and $\mathcal{V}$ on common input x. More formally, for every x and w such that $(x, w) \in R$ and every $e \in \{0,1\}^\lambda$, it holds that
$$\{\mathcal{S}(x, e)\} \sim \{\langle \mathcal{P}(x, w), \mathcal{V}(x, e)\rangle\}.$$
$\mathcal{S}(x, e)$ denotes the output of simulator $\mathcal{S}$ on input (x, e), and $\langle \mathcal{P}(x, w), \mathcal{V}(x, e)\rangle$ denotes the output transcript of an execution between $\mathcal{P}$ on input (x, w) and $\mathcal{V}$ on input x with $\mathcal{V}$'s random tape equal to λ. The value λ determining the length of e is called the *challenge length*.

The three values sent during the interaction can be referred to as the commitment a, the challenge e and the response z. The triple $\pi = (a, e, z)$ consisting of all the values sent during the interaction is called a transcript.

[1] The definition as well as the OR-proof is a rewritten version of a technical essay written as a part of the course "cryptographic computing" (520172U008) at Aarhus university. The essay is available on request.

The reason we say that the last requirement above is *special* HVZK, is that the challenge e is assumed to be chosen uniformly at random from the challenge set.

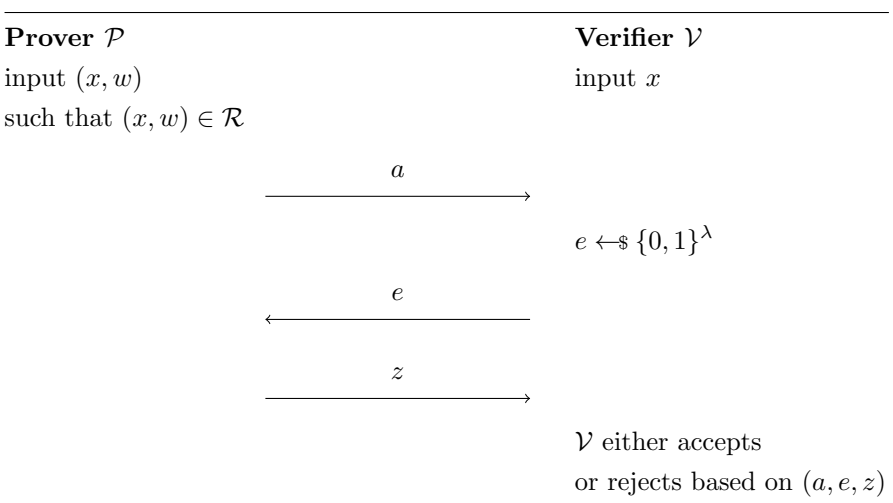


Figure 2: Σ-Protocol

Remark 3.2. Depending on the adversarial power taken into consideration, we can either have computational or unconditional special soundness, and computational, statistical or perfect special HVZK.

One can think of the requirements for a Σ-protocol as follows. Completeness ensures that when the protocol is run on valid input, it should always work as we expect it to. That is, the verifier should accept, because the prover indeed has a witness it tries to convince the verifier of.

Special soundness is a requirement protecting against cheating provers. That is, it guarantees that a malicious prover $\mathcal{P}^*$ can only answer at most one challenge correctly for a given commitment a, without actually being able to compute or be in possession of a witness. It captures the idea that $\mathcal{V}$ should only accept a false statement with a small probability. The property is also referred to as 2-special soundness, since if we have two valid transcripts for the same commitment, we can extract a witness. A generalization of this requirement, which is referred to as k-special soundness, then extends to having k accepting transcript on the same commitment to be able to extract a witness. The prover can then only answer at most $k-1$ challenges correctly without a witness.

For some protocols, there is a trade-off between what you want to prove in a Σ-protocol, and what you in practice can prove. An example of such a protocol will be given in Section 6, where we as a result of using lattice-based

hardness assumption have a norm bound on the size of the witness. Using a smaller witness in an honest execution of the protocol, but allowing for larger witnesses in order to obtain special soundness can be referred to as the protocol having a soundness gap. For Σ-protocols where we only have one repetition, this will not directly lead to any issues, however it may be undesirable for applications where protocol repetitions are necessary.

Special HVZK ensures that a verifier does not learn anything by following the protocol honestly. Everything the verifier receives, could also have been computed without a witness. This is however a weaker guarantee than zero-knowledge, as we are not considering cheating verifiers that may deviate from the protocol specifications. We note that for the Special HVZK requirement, it is the entire simulated transcripts that has to have the same probability distribution as the real transcripts obtained from an interaction between a prover and a verifier. Therefore, the order in which the elements of a simulated transcript are generated need not be chronological. this allows flexibility to generate simulated transcripts in "the wrong order", where one can adapt the commitment a to the challenge e, as it must not be decided before seeing what the challenge is.

3.2 Proving OR of Two Statements

We will now see one way that on common input (x_0, x_1), a prover $\mathcal{P}$ can convince a verifier $\mathcal{V}$ that it knows a witness w such that either $(x_0, w) \in R$ or $(x_1, w) \in \mathcal{R}$, without revealing to $\mathcal{V}$ which of the values x_0 and x_1 it knows a witness of. A relation for these requirements can be described as

$$\mathcal{R}_{\text{OR}} = \{(X = (x_0, x_1), w) |\ (x_0, w) \in \mathcal{R} \ \vee \ (x_1, w) \in \mathcal{R}\}. \tag{10}$$

The relation $\mathcal{R}_{\text{OR}}$ can be extended to cover the case where we have several common inputs $(x_0, \ldots, x_n)$ for the relations $\mathcal{R}_0, \ldots, \mathcal{R}_n$, and the prover $\mathcal{P}$ only knows witnesses for k of the common inputs.

If Π is a Σ-protocol for relation $\mathcal{R}$, the idea behind the OR-construction with relation $\mathcal{R}_{\text{OR}}$ is that prover $\mathcal{P}$ wants to make one instance of Π on common input x_0 and one on common input x_1. If $(x_b, w) \in \mathcal{R}$ for $b \in \{0, 1\}$, we can obtain an accepting transcript for any challenge directly, since we have the corresponding witness. However we cannot do this for x_{1-b}, as we only have a witness for one of the values x_0 and x_1. But since Π is a Σ-protocol, we know there exists a simulator $\mathcal{S}$ that outputs transcripts for Π that are indistinguishable from real ones, and we want to use this to complete an instance for x_{1-b}. Still, to use the simulator $\mathcal{S}$ directly to complete the instance, we would need to know the challenge e_{1-b} from $\mathcal{V}$ before sending the commitment a_{1-b}. We remedy this issue by running the simulator $\mathcal{S}$ on a random value e_{1-b} before receiving the challenge s from $\mathcal{V}$. Then, we use a combination of our random value e_{1-b} and the verifiers challenge s to create a second value e_b, which we then use as the challenge

for a_b. Assuming we have a witness w for x_b, we can always generate a valid response z_b for (a_b, e_b). The resulting protocol Π_{OR} is given in Figure 3.

Interaction in Π_{OR} for Relation $\mathcal{R}_{\text{OR}}$:

Prover $\mathcal{P}$		**Verifier** $\mathcal{V}$
input $((x_0, x_1), w) \in \mathcal{R}_{\text{OR}}$		input (x_0, x_1)
Sample $e_{1-b} \leftarrow\!\!\$\ \{0,1\}^\lambda$		
Run $(a_{1-b}, z_{1-b}) \leftarrow\!\!\$\ \mathcal{S}(x_{1-b}, e_{1-b})$		
Compute $a_b \leftarrow \Pi(x_b, w)$		
	$\xrightarrow{a_b, a_{1-b}}$	
		$s \leftarrow\!\!\$\ \{0,1\}^\lambda$
	$\xleftarrow{s}$	
Define $e_b = s \oplus e_{1-b}$		
Compute $z_b \leftarrow \Pi(a_b, e_b)$		
	$\xrightarrow{z_0, z_1, e_0, e_1}$	
		$\mathcal{V}$ accepts if $\Pi(a_0, e_0, z_0) = 1 \wedge \Pi(a_1, e_1, z_1) = 1 \wedge e_0 \oplus e_1 = s$

Figure 3: OR-Protocol

Theorem 3.3 (OR-proof)**.** *Let $\mathcal{R}_{OR}$ be defined as in Equation 10, and let Π be a Σ-protocol for relation $\mathcal{R}$. Then the protocol Π_{OR} given in Figure 3 is a Σ-protocol for relation $\mathcal{R}_{OR}$. Moreover, for any verifier $\mathcal{V}^*$, the probability distribution over transcripts between $\mathcal{P}$ and $\mathcal{V}^*$, where w is such that $(x_b, w) \in \mathcal{R}$, is independent of b.*

Proof. Let $b \in \{0,1\}$ be such that $(x_b, w) \in \mathcal{R}$.

Completeness Suppose that both $\mathcal{P}$ and $\mathcal{V}$ follow the protocol specifications of Π_{OR}. Then $e_b = s \oplus e_{1-b}$, so $e_b \oplus e_{1-b} = s \oplus e_{1-b} \oplus e_{1-b} = s$. We also have that $(a_{1-b}, e_{1-b}, z_{1-b})$ is the output of the simulator $\mathcal{S}$ for Π, and therefore has the same probability distribution as a real conversation between $\mathcal{P}$ and $\mathcal{V}$ behaving honestly. It is therefore an accepting transcript in Π on common input x_{1-b}. Since a_b and z_b are computed in Π using (x_b, w) and (x_b, a_b, e, w) as input respectively, (a_b, e_b, z_b) is an accepting transcript in Π on input x_b. Therefore $\mathcal{V}$ always accepts at the end of the protocol Π_{OR}, which shows protocol Π_{OR} has completeness.

Special Soundness Suppose that

$$\pi = ((a_0, a_1), s, (e_0, e_1, z_0, z_1)) \quad \text{and} \quad \pi' = ((a_0, a_1), s', (e'_0, e'_1, z'_0, z'_1))$$

are two accepting transcripts on common input (x_0, x_1) with $s \neq s'$. Since $s = e_0 \oplus e_1$ and $s' = e'_0 \oplus e'_1$, for some $c \in \{0, 1\}$ we have $e_c \neq e'_c$. By construction of Π_{OR}, (a_c, e_c, z_c) and (a'_c, e'_c, z'_c) are two accepting transcripts for protocol Π with common input x_c. Since we assume that Π is a Σ-protocol, special soundness gives us that we can efficiently compute a witness w such that $(x_c, w) \in \mathcal{R}$ from (a_c, e_c, z_c) and (a'_c, e'_c, z'_c), since $e_c \neq e'_c$. For the same witness, we also have $((x_0, x_1), w) \in \mathcal{R}_{\text{OR}}$ by definition. This shows special soundness for Π_{OR}.

Special HVZK We define a simulator $\mathcal{S}'$ as follows. On common input (x_0, x_1) and challenge s, define $e_0 \leftarrow\$ \{0, 1\}^\lambda$ and $e_1 = s \oplus e_0$. Since Π is a Σ-protocol, it is Special HVZK and we have the simulator $\mathcal{S}$. Therefore we run $\mathcal{S}$ on input (x_0, e_0) and (x_1, e_1) to output the accepting transcripts (a_0, e_0, z_0) and (a_1, e_1, z_1) respectively. From this, $(a_0, a_1, s, e_0, e_1, z_0, z_1)$ is an accepting transcripts for Π_{OR}. This shows Π_{OR} is Special HVZK and we can conclude that Π_{OR} is a Σ-protocol.

We will now show that the probability distribution of transcripts is independent of b. Suppose that $\mathcal{V}^*$ is an arbitrary verifier. Then any transcript between $\mathcal{P}$ and $\mathcal{V}^*$ is of the form $((a_0, a_1), s, (e_0, e_1, z_0, z_1))$. The commitments a_0 and a_1 are distributed as an honest prover in Π would choose them. This holds directly for a_b, and from the Special HVZK property of Π, a_{1-b} has the correct distribution as well. s is given by $\mathcal{V}^*$ and therefore has whatever distribution $\mathcal{V}^*$ outputs given the common input (x_0, x_1) and the values a_0, a_1. The values e_0 and e_1 are both random under the constraint that $e_0 \oplus e_1 = s$, since $e_{1-b} \leftarrow\$ \{0, 1\}^\lambda$. Also, z_0 and z_1 are distributed as an honest prover in Π would choose them, given common input x_0 and the first values a_0, e_0 and common input x_1 and the first values a_1, e_1 respectively. Similarly as the argument for a_0, a_1, since Π is Special HVZK, they are both distributed as an honest prover would choose them. Since the distribution of none of the values in the transcript are dependent of b, we conclude that the distribution over transcripts is independent of b. □

The last part of the proof shows that even for a malicious verifier $\mathcal{V}^*$ that deviates from the protocol, there is no way it can tell which of the possible witnesses that $\mathcal{P}$ knows, as there are several possible witnesses $\mathcal{P}$ can use to complete the protocol successfully.

3.3 Every Σ-Protocol is Knowledge Sound, Version I

In a proof of knowledge (PoK), a prover wants to convince a verifier that it is in possession of a valid witness for a given statement, something which is stronger than only being able to convince the verifier that it knows that such a value exists. The property that captures the notion of a prover "knowing" something is referred to as knowledge soundness. It requires that for a prover to be in possession of such a witness, there must exist an algorithm, referred to as the knowledge extractor, that can retrieve such a witness from the prover efficiently. A protocol is said to be a PoK if it has completeness and is knowledge sound.

A well-known result is that Σ-protocols are knowledge sound, which then again implies that they are PoKs. Slightly different definitions of knowledge soundness have been proposed over the years, leading to different knowledge extractors that serve the same purpose. We will look at two different ways we can show that Σ-protocols are knowledge sound, where we begin with the classical proof by Damgård [Dam10]. Then we cover another proof given by Attema, Cramer and Kohl a few years ago [ACK21], which in particular can be generalized to show that $(k_1, \ldots, k_\mu)$-special soundness tightly implies knowledge soundness.

For notation, let L_R be the set of common input x for which there exists at least one witness w such that $(x, w) \in \mathcal{R}$.

Definition 3.4 (Knowledge Soundness, Version I [Dam10; HL10])**.** Let $\kappa : \{0,1\}^* \to [0,1]$ be a function, and let $(\mathcal{P}, \mathcal{V})$ be an interactive protocol for the relation $\mathcal{R}$. Then we say that $(\mathcal{P}, \mathcal{V})$ is knowledge sound with knowledge error κ if it satisfies the following property.

- **Knowledge Soundness:** There exists a probabilistic (oracle) machine $\mathcal{K}$ called the *knowledge extractor*, as well as a constant $c > 0$ such that for every prover $\mathcal{P}^*$ and every $x \in L_R$, the knowledge extractor $\mathcal{K}$ satisfies the following condition.

  Let $\varepsilon(x)$ be the probability that $\mathcal{V}$ accepts on input x after interaction with $\mathcal{P}^*$. If $\varepsilon(x) > \kappa(x)$, then on input x and oracle access to $\mathcal{P}^*$, the knowledge extractor $\mathcal{K}$ outputs a string w such that $(x, w) \in \mathcal{R}$ within an expected number of steps bounded by

  $$\frac{|x|^c}{\varepsilon(x) - \kappa(x)}.$$

The knowledge error κ can be thought of as the probability that a prover that does not possess a witness w is able to convince the verifier to accept. Therefore if $\varepsilon(x) > \kappa(x)$ for a given prover $\mathcal{P}^*$, it means it has a higher probability of convincing the verifier than if it would not have a witness.

Furthermore, the efficiency of the knowledge extractor depends on how much better the given prover $\mathcal{P}^*$ is at convincing the verifier to accept, than a prover which does not have a witness. That is, the larger $\varepsilon(x)$ is compared to $\kappa(x)$, the less amount of steps are necessary for $\mathcal{K}$ to extract a witness.

Notice that the special soundness property of a Σ-protocol is connected to this witness extraction, in the following sense. If we have two accepting transcripts for the same commit message (first message) sent by the prover, the witness is efficiently computable. Therefore, if it is possible to construct an algorithm for arbitrary Σ-protocols that can find two such transcript in PPT, we would show the existence of a knowledge extractor and thus prove that every Σ-protocol has knowledge soundness.

Theorem 3.5 (Every Σ-Protocol is a Proof of Knowledge)**.** *Let Π be a Σ-protocol for a relation $\mathcal{R}$ with challenge length λ. Then Π is a proof of knowledge with knowledge error $2^{-\lambda}$.*

Proof Idea. Before we begin our proof, we will sketch the proof idea for knowledge soundness, highlighting the different cases we will consider to ensure that the constructed knowledge extractor always runs in a number of steps bounded as necessary. We first note that the knowledge error equals $\kappa(x) = 2^{-\lambda}$, since this is the probability that an arbitrary prover can guess the challenge sent by the verifier correctly before it receives it.

We want to find two accepting transcript for the same random choices ρ made by $\mathcal{P}^*$, i.e. the same first commit message sent by the prover. We assume that $\varepsilon(x) > 2^{-\lambda}$, and look into two cases. In both cases, if the extractor we construct finds an accepting transcript for some random choices ρ made by $\mathcal{P}^*$, it will try to find another for the same ρ. We first assume that $\varepsilon > 2^{-\lambda+2}$. As we will see, this ensures that for at least half of all accepting transcripts, there is at least one other accepting transcript for the same random choices made by $\mathcal{P}^*$.

We will first analyze the expected number of tries needed to find another accepting transcript for the same ρ, when assuming there does exist more than one accepting transcript for it. We will then look at how we can ensure the extractor still runs in PPT if it finds an accepting transcript for random choices ρ where there is not necessarily multiple accepting transcripts, or the algorithm is unlucky in its search. Using rejection sampling, we make sure the amount of steps used when checking a given ρ is not too high, while keeping the probability of finding a second accepting transcript in the case where it does exist not too low. The reason we do this, is that an extractor cannot check every possible challenge for ρ, as the challenge space has 2^λ elements, which is exponentially large.

Finally, we make a case distinction for when $2^{-\lambda} < \varepsilon < 2^{-\lambda+2}$. In this case, we can check every challenge for the random choices ρ before we proceed to the next one. Thus we only have to argue that the number of steps the algorithms must take in this case is still as required.

□

Proof. Let $\mathcal{P}^*$ be a prover that upon input x convinces a verifier $\mathcal{V}$ after interaction to accept with probability $\varepsilon := \varepsilon(x)$, and assume that $\varepsilon(x) > \kappa(x)$. We have that the knowledge error is $\kappa(x) = 2^{-\lambda}$, since this is the probability that an arbitrary prover can guess the challenge sent by the verifier correctly before it receives it.

Let H be a 0/1-matrix with dimensions such that we have a row for each of the possible random choices ρ made by the prover $\mathcal{P}^*$, and one column for each of the values that are in the challenge space. H is defined such that an entry is equal to 1 only if the verifier $\mathcal{V}$ accepts a transcript generated by the random choices and the challenge corresponding to the current row and column, respectively. The entry is defined as 0 otherwise.

We can probe random entries in H, by using $\mathcal{P}^*$ as a black-box while we are choosing its random coins at random as well as choosing random challenges for it. Rewinding $\mathcal{P}^*$ while reusing the same random choices ρ as before on different challenges allows us to probe new random entries in the same row. The goal for the knowledge extractor is therefore to find two 1's that are in the same row, such that we can use the special soundness property of the Σ-protocol to efficiently compute a witness for x.

By defining H as we have done, ε equals the fraction of 1-entries in H. Since we want to construct a knowledge extractor that can find two 1's in the same row, we want to understand how they are distributed in H. To do so, we define a row to be *heavy* if it contains a fraction of at least $\varepsilon/2$ 1-entries. That is, a heavy row contains at least $\varepsilon \cdot 2^{\lambda-1}$ ones. We claim that over half of the 1's in H are located in heavy rows. To see this, let H' be the sub-matrix of H consisting of all rows that are not heavy. Now let h be the number of entries in H and similarly let h' be the number of entries in H'. We then have that the number of 1's in H is $h \cdot \varepsilon$. Since every row in H' is not heavy, each row must contain less than $\varepsilon \cdot 2^{\lambda-1}$ ones. Therefore if we let r' be the number of rows in H', we have that the number of 1's in H' is smaller than $r' \cdot (\varepsilon \cdot 2^{\lambda-1}) = (r' \cdot 2^{\lambda}) \cdot \varepsilon/2 = h' \cdot \varepsilon/2$. It follows that the number of 1's that lie in heavy rows in H is greater than

$$h\varepsilon - h' \cdot \varepsilon/2 \geq h\varepsilon - h \cdot \varepsilon/2 = h \cdot \varepsilon/2,$$

which shows that over half of the rows in H are heavy.

We will now proceed with describing the behavior of the knowledge extractor when $\varepsilon \geq 2^{-\lambda+2}$. This is to ensure that every heavy row contains at least two 1's, since from above we know that each heavy row contains at least $\varepsilon \cdot 2^{\lambda-1} \geq 2^{-\lambda+2} \cdot 2^{\lambda-1} = 2$ ones. We will show that under this assumption, two 1's can be found in the same row in an expected number of steps bounded by $\mathcal{O}(1/\varepsilon)$.

The algorithm begins by first repeatedly probing H at random, until it finds a 1 entry as its "first hit". Since the fraction of 1's in H is ε, this

happens after an expected number of $1/\varepsilon$ tries. By the argument above, the first hit lies in a heavy row with a probability grater than $1/2$. We will now analyze the probability of finding a second hit in the same row, both when the first hit is in a heavy row and when it is not. Note that the algorithm cannot know if the first hit is in such a row or not. Assuming that the first hit does lie in a heavy row, if we probe the row at random the probability of finding a second 1 in one attempt is equal to

$$\frac{\varepsilon \cdot 2^{\lambda-1} - 1}{2^{\lambda}}.$$

This is because there are at least $\varepsilon \cdot 2^{\lambda-1}$ ones in heavy rows and we need to find a different 1 than our first hit. This implies that the expected number of tries T to find a second hit satisfies

$$T = \frac{2^{\lambda}}{\varepsilon \cdot 2^{\lambda-1} - 1} = \frac{2^{\lambda}}{\varepsilon/2 \cdot (2^{\lambda} - 2/\varepsilon)} = \frac{2}{\varepsilon} \cdot \frac{2^{\lambda}}{2^{\lambda} - 2/\varepsilon} \leq \frac{2}{\varepsilon} \cdot \frac{2^{\lambda}}{2^{\lambda} - 2^{\lambda-1}} = \frac{4}{\varepsilon},$$

where the inequality follows from our assumption that $\varepsilon \geq 2^{-\lambda+2}$. We can therefore conclude that when assuming the first hit was in a heavy row, the knowledge extractor succeeds within an expected number of $\mathcal{O}(1/\varepsilon)$ tries.

We will now look at how we can handle the case where the extractor's first hit was not in a heavy row, or it is unlucky when searching for the second hit. In this case, the extractor may spend too much time looking for the second hit. We remedy this by using rejection sampling to ensure that the extractor starts probing another row if it spends too much time looking for a second hit. We therefore end up with the following algorithm, which the knowledge extractor will repeat until it succeeds:

1. Probe random entries in H until the first hit is found. Name the row it is found in the *current row*.

2. Start the two following processes in parallel, the algorithm stops when either of these stop.

   - **Process *Pr1*:** Probe random entries in the current row until the second hit is found.
   - **Process *Pr2*:** Probe a random entry in H for a second hit and at the same time choose a random number out of $[d]$. This is equivalent to flipping a coin that comes out with heads with probability ε/d for some constant d repeatedly until you get heads. Output heads if both the entry in H is 1 and the number chosen at random from $[d]$ is 1.

We analyze the expected running time, as well as its probability of success. The expected run time of *Pr2* is d/ε, since the probability of the coin described being heads is ε/d. Therefore since d is a constant, the expected

running time is $\mathcal{O}(1/\varepsilon)$ as required. Note that the algorithm only has what it needs to extract a witness if *Pr1* finishes first. Therefore, we must choose d such that process 1 has enough time to finish (in reasonable time) if the row it is probing is heavy. We will now look at how to ensure that the described algorithm finds a second hit with probability 1/8.

The probability that *Pr2* finishes after exactly k attempts equals

$$\varepsilon/d \cdot (1 - \varepsilon/d)^{k-1},$$

since each coin toss in *Pr2* are independent. Using the estimate that $(1 - \varepsilon/d)^{k-1} \leq 1$ for any k, the probability of *Pr2* finishing within k attempts is less than or equal to $\sum_{i=1}^{k} \varepsilon/d = k \cdot \varepsilon/d$. So if we let $k\varepsilon/d = 1/2$ and $d = 16$, this ensures that the probability that process *Pr2* finishes within k steps is less than or equal to 1/2. By letting $d = 16$, we have that *Pr2* does not finish after

$$k = \frac{d}{2\varepsilon} = \frac{8}{\varepsilon}$$

tries with a probability greater than 1/2.

We will now show that process *Pr1* finishes before process *Pr2* with probability at least 1/8. Recall that the expected number of steps for *Pr1* to finish is $T = 4/\varepsilon$. We claim that the probability that process *Pr1* concludes in less than $8/\varepsilon$ steps given that the first hit is in a heavy row, is at least 1/2. This follows from Markov's inequality, for if we let X be the number of steps process *Pr1* takes, we have that

$$\Pr[X \geq 2T] \leq \frac{E(X)}{2T} = \frac{T}{2T} = \frac{1}{2}.$$

Therefore *Pr1* concludes in less than $8/\varepsilon$ steps with a probability of $1 - 1/2 = 1/2$ as well. Also, note that $2T = 8/\varepsilon = k$, where the right side is the expected number of steps process *Pr2* takes to finish with probability less than or equal to 1/2. That is, it does not finish with a probability greater than 1/2. Since process *Pr1* and *Pr2* are independent, we have that *Pr1* finishes before *Pr2* with a probability greater than $1/2 \cdot 1/2 = 1/4$. We have established that the probability of the first hit being heavy is greater than 1/2, and we can therefore conclude that the algorithm succeeds, meaning it finds a heavy row and process *Pr1* finishes first, with probability at least $1/2 \cdot 1/4 = 1/8$ as we wanted.

We can now analyze the knowledge extractor for the case $\varepsilon \geq 2^{-\lambda+2}$ as follows. As mentioned, it repeats the above algorithm until it succeeds, and we have seen, the expected number of repetitions in order for process *Pr1* to finish first is 8, so the number of repetitions is constant. As desired, the knowledge extractor therefore succeeds in finding a witness in an expected number of steps bounded by $\mathcal{O}(1/\varepsilon)$.

What now remains is to consider when $2^{-\lambda} < \varepsilon < 2^{-\lambda+2}$. In this case, ε is small enough for us to have time to probe an entire row. Therefore we

will construct another algorithm that the knowledge extractor $\mathcal{K}$ will run in parallel with the one we described above.

Define δ by $\varepsilon = (1+\delta)2^{-\lambda}$, which means that $0 < \delta < 3$ and let R be the number of rows in H. Then we have that the number of 1's in H can be rewritten as $(1+\delta)R$ since ε is the fraction of 1's in H. We are again interested in how the 1's in H are distributed. Since there are R rows, we first note that at most $R-1$ of the $(1+\delta)R$ ones can be alone in a row, and then all the other ones must be in the remaining row. Subtracting $R-1$ from the number of 1's in H, we see that at least $1+\delta R$ of the 1's in H are in rows that contain at least two 1's. This again implies that at least δR of the 1's are located in rows with at least two 1's. We call rows that contain at least two 1's *semi-heavy*. The algorithm can be described as follows.

1. Probe random entries in H until a 1 is found: call this the current row.

2. Search the entire current row for another 1 entry. If no such entry is found, go back to step 1.

It is clear that the described algorithm will successfully find two 1's in the same row. We argue that it does so in a number of steps bounded by $\mathcal{O}(1/(\varepsilon - 2^{-\lambda}))$. To analyze this, note that the fraction of 1's in semi-heavy rows among the 1's is at least $\delta R/((1+\delta)R) = \delta/(1+\delta)$, and the fraction of 1's in semi-heavy rows among all entries is at least $\delta R/(2^{\lambda}R) = \delta/2^{\lambda}$. Now, the expected number of probes to find a 1 in step 1 of the algorithm is $1/\varepsilon = 2^{\lambda}/(1+\delta)$, and the number of steps in step 2 is upper bounded by 2^{λ}. Furthermore, the expected number of 1's we must try in order to find one in a semi-heavy row is at least $\delta/(1+\delta)$. Therefore the expected number of times the algorithm runs both step 1 and step 2 is $\delta/(1+\delta)$. Putting this together, we obtain that the expected number of steps is upper-bounded by

$$\frac{1+\delta}{\delta} \cdot \left(\frac{2^{\lambda}}{1+\delta} + 2^{\lambda}\right) = 2^{\lambda} \cdot \left(\frac{1}{\delta} + \frac{1+\delta}{\delta}\right) = 2^{\lambda} \cdot \frac{2+\delta}{\delta} < 5 \cdot \frac{2^{\lambda}}{\delta},$$

where we obtain the last inequality from $0 < \delta < 3$. We can rewrite

$$\frac{1}{\varepsilon - 2^{\lambda}} = \frac{1}{(1+\delta)2^{-\lambda} - 2^{-\lambda}} = \frac{2^{\lambda}}{\delta},$$

and therefore conclude that the expected number of steps the algorithm takes is upper bounded by $\mathcal{O}(1/(\varepsilon - 2^{-\lambda}))$ as required. This completes the proof, showing that every Σ-protocol with challenge length λ is a proof of knowledge with knowledge error $2^{-\lambda}$. □

3.4 Every Σ-Protocol is Knowledge Sound, Version II

Definition 3.6 (Knowledge Soundness, Version II [ACK21])**.** Let $\Pi = (\mathcal{P}, \mathcal{V})$ be an interactive protocol for relation $\mathcal{R}$. Let $\kappa : \mathbb{N} \to [0, 1)$ be a function. Then Π is said to be knowledge sound with knowledge error κ, if there exists a polynomial $q : \mathbb{N} \to \mathbb{N}$ and an algorithm $\mathcal{K}$, called a knowledge extractor, with the following properties:

- The extractor $\mathcal{K}$, given input x and rewindable oracle access to a (potentially dishonest) prover $\mathcal{P}^*$, runs in an expected polynomial number of steps.
- Whenever $(\mathcal{P}^*, \mathcal{V})(x)$ outputs accept with probability $\varepsilon(x) \geq \kappa(|x|)$, the extractor $\mathcal{K}$ successfully outputs a witness w such that $(x, w) \in \mathcal{R}$ with probability at least

$$\frac{(\varepsilon(x) - \kappa(|x|))}{q(|x|)}.$$

■

Theorem 3.7 (Special Soundness Implies Knowledge Soundness, Version II [ACK21])**.** *Let $(\mathcal{P}, \mathcal{V})$ be a 2-special sound Σ-protocol for relation $\mathcal{R}$, where $\mathcal{V}$ samples each challenge uniformly at random from a challenge set of size $N = 2^\lambda \geq 2$. Then $(\mathcal{P}, \mathcal{V})$ is knowledge sound with soundness error $\kappa = 1/N$.*

Proof. Let $H \in \{0, 1\}^{R \times N}$ and prover $\mathcal{P}^*$ be defined as in the proof of Theorem 3.5. Assume that $\varepsilon(x) \geq \kappa(x)$, and note that the knowledge error equals $\kappa(x) = 1/N$, since this is the probability that an arbitrary prover can correctly guess the challenge sent by a verifier before it receives it.

The following algorithm describes a collision-game, that the knowledge extractor $\mathcal{K}$ performs in an attempt to retrieve two 1-entries from the same row of H. As before, the goal is to find two 1's that are in the same row, such that we can use the special soundness property of the Σ-protocol to efficiently compute a witness for the common input x.

Collision-Game

1. Sample an entry of H uniformly at random.
2. If the entry is 0, the algorithm aborts.
3. If the entry is a 1, continue to sample elements from the row it was found, without replacement. This in done until a second 1 is found, or the whole row has been checked.

In order to show that a knowledge extractor that runs the described algorithm satisfies Definition 3.6, we analyze the expected number of queries the algorithm will make, as well as the success probability when performing one execution of the collision-game.

Expected Number of Queries We want to upper bound the expected number of queries the collision-game will make before it aborts, to ensure the collision-game runs in expected polynomial time.

Let the random variable X denote the number of queries made by the collision-game. Let ε_i be the fraction of 1-entries in row i, which means that $\varepsilon_i N$ is the number of 1-entries in row i. We let the Q_i denote the event that the first query in the collision-game lies in row i. We also let A denote the event that the first query in the collision-game is a 0-entry, and let B denote the event that the first query in the collision-game is a 1-entry.

Using conditioned expectation, the expected number of queries made by the collision-game can be rewritten as

$$\begin{aligned}\mathbb{E}[X] &= \sum_{i=1}^{R} \Pr[Q_i] \cdot \mathbb{E}[X|\ Q_i] \\ &= \frac{1}{R} \sum_{i=1}^{R} \mathbb{E}[X|\ Q_i],\end{aligned}$$

where we have that $\Pr[Q_i] = 1/R$ for all $i = 1, \ldots, R$ since the collision-game always draws the first query uniformly at random from the entries in H. Our goal is to show that $\mathbb{E}[X|\ Q_i] \leq 2$ for all $i = 1, \ldots, R$, the expression above will then be at most 2 as well.

The expected number of queries made by the collision-game, given that that first query lies in row i can be computed as

$$\begin{aligned}\mathbb{E}[X|\ Q_i] &= \Pr[A|\ Q_i] \cdot \mathbb{E}[X|\ A \wedge Q_i] + \Pr[B|\ Q_i] \cdot \mathbb{E}[X|\ B \wedge Q_i] \\ &= (1 - \varepsilon_i) \cdot 1 + \varepsilon_i \cdot \mathbb{E}[X|\ B \wedge Q_i].\end{aligned}$$

Here, we have made the following observations. The probability that the first query made by the collision-game is 0-entry, given that the first query lies in row i (that is, $\Pr[A|\ Q_i]$) is $1 - \varepsilon_i$, since ε_i is the probability of choosing a 1-entry from row i uniformly at random. Similarly, we obtain that the probability that the first query made by the collision-game is a 1-entry, given that the first query lies in row i (that is, $\Pr[B|\ Q_i]$) is ε_i. We also know that the expected number of queries the collision-game makes, given that the first query is a 0-entry and lies in row i (that is, $\mathbb{E}[X|\ A \wedge Q_i]$) equals 1, since the algorithm is instructed to abort if the first query equals a 0-entry.

It remains to calculate $\mathbb{E}[X|\ B \wedge Q_i]$, namely the expected number of queries made by the collision-game, given that the first query is a 1-entry and it lies in row i. After obtaining the first query, which we know is a 1-entry, the algorithm will continue by sampling entries from H along row i without replacement, and it does so until it finds a second 1 or the whole row has been exhausted. This can be modeled by a negative hypergeometric

distribution with $N-1$ values to choose from, where $\varepsilon_i N - 1$ of these are favorable. We let the random variable Y denote the number of draws made according to this distribution, and can therefore rewrite the expression as

$$\mathbb{E}[X|\ B \wedge Q_i] = 1 + \mathbb{E}[Y]. \tag{11}$$

As mentioned, the 1 in the expression represent that the algorithm first obtains the 1-entry. By properties of the negative hypergeometric distribution distribution, this can be computed as

$$\mathbb{E}[Y] = \frac{1 \cdot ((N-1)+1)}{(\varepsilon_i N - 1) + 1} = \frac{1}{\varepsilon_i},$$

as long as $\varepsilon_i > 1/N$. In the case that row i contains only one 1-entry, namely the one we obtain on the first query, the number of draws is $N-1$, since this is the amount of entries left in row i. However in this case, we still have that $N - 1 < 1/\varepsilon_i = N$. Therefore, $\mathbb{E}[Y]$ can be upper-bounded by $1/\varepsilon_i$, and we obtain that Equation 11 is upper-bounded by $1 + 1/\varepsilon_i$.

Putting all of this together, we obtain that

$$\begin{aligned}
\mathbb{E}[X] &= \frac{1}{R}\sum_{i=1}^{R} \mathbb{E}[X|\ Q_i] \\
&\leq \frac{1}{R}\sum_{i=1}^{R} \left((1-\varepsilon_i)\cdot 1 + \varepsilon_i(1 + \frac{1}{\varepsilon_i}) \right) \\
&= 2.
\end{aligned}$$

This shows that the expected number of queries the collision-game makes is 2, which is certainly an expected polynomial number of steps.

Success Probability We say that the collision-game is successful if the first entry is a 1 that lies in a row with at least two 1-entries in it, since it is in this case we are able to obtain two 1-entries from the same row and thus extract a valid witness.

We let δ_k be the fraction of rows with exactly k 1s in it, and using this notation we have that

$$\Pr[\text{success}] = \sum_{k=2}^{N} \frac{k}{N}\delta_k = \left(\sum_{k=0}^{N} \frac{k}{N}\delta_k \right) - \frac{\delta_1}{N} = \varepsilon - \frac{\delta_1}{N} \geq \varepsilon - \frac{1}{N}.$$

Since $\kappa(x) = 1/N$, this shows that the success probability is of the desired form. Putting this together, we have shown that a knowledge extractor $\mathcal{K}$ that runs the described collision-game is expected to make at most 2 queries, and the success probability of the algorithm is of the correct form. □

Remark 3.8. The following are differences and similarities between the knowledge extractor in Theorem 3.7 and the knowledge extractor in Theorem 3.5. In Definition 3.4, the knowledge extractor does not necessarily run in expected polynomial time. We do however require a bound on the expected number of steps the algorithm takes, but the denominator $\varepsilon(x)-\kappa(x)$ may be negligible. The knowledge extractor that satisfies this definition is however guaranteed to always output a valid witness for the given common input. It repeats a sub-algorithm that succeeds with probability $1/8$ until it succeeds.

In Definition 3.6, we have a polynomial bound on the expected running time of the constructed knowledge extractor, however the algorithm outputs a witness with notably lower success probability, namely $\varepsilon - 1/N$. This trade-off in running time and success probability does however allow us to construct a simpler knowledge extractor that can abort when it gets unlucky. The expected running time only increases by a factor relative to the additional number of accepting transcripts needed to extract a witness if we extend the proof to k-special soundness.

3.5 The Fiat-Shamir Transformation

As we have seen, Σ-protocols are public-coin interactive protocols, since the verifier's challenge is chosen uniformly at random from a given challenge space. As interactive protocols can be expensive to use due to the communication that has to take place between a prover and a verifier, being able to turn any Σ-protocol into a non-interactive protocol is highly favorable. This can be achieved by hashing the first message by the prover together with the common input into the appropriate challenge space, and use the output as a replacement for the message that initially had to be sent by the verifier. If we in addition to the first message by the prover append an arbitrary message to it before hashing, transcripts of this form gives a digital signature scheme. This technique of making protocols non-interactive and constructing signature schemes is referred the Fiat-Shamir transformation [FS87], and is a reason for why Σ-protocols are of central significance when it comes to constructing digital signature schemes.

Assuming that we are in the ROM, we consider the output a hash function as uniformly distributed, so choosing the challenge to depend on the first message in this way still allows us to create accepting transcripts as one normally would do in the interactive setting [BR93]. Now the prover can generate the whole transcript without interaction with the verifier $\mathcal{V}$, and forward it to $\mathcal{V}$ who runs the verification check on it. Also, since the challenge now is a hash of the first message, the prover only needs to forward the challenge and response of the transcript for it to be possible to verify.

4 Compressed Σ-Protocols

In the following section, we look at a technique that allows us to lower the communication complexity of Σ-protocols when proving knowledge of a commited witness satisfying some linear constraint in the discrete log setting. It is referred to as a compression mechanism, and applying it recursively to appropriate Σ-protocols allow us to go from constant to logarithmic communication costs in the size of the witness. The technique was first used in Bulletproofs [Bün+18], but was adapted by Attema and Cramer to Σ-protocols in such a way that it can be applied to a wider selection of protocols [AC20]. In Section 7, we look at how Attema, Cramer and Kohl further extend these ideas to the lattice setting [ACK21].

For a publicly known commitment P and public parameters pp, a partial opening y and a linear form $L \in \mathcal{L}(V)$ for a vector space V over the field K, a prover $\mathcal{P}$ wants to convince a verifier $\mathcal{V}$ that they are in possession of a secret value $\mathbf{x}$ and randomness γ, such that P is a commitment of $\mathbf{x}$ using randomness γ, and that the partial opening y equals L evaluated in $\mathbf{x}$. A relation for these requirements can be described as

$$\begin{aligned}\mathcal{R}_{\mathsf{com}} := \big\{ (X = (P \in \mathbb{G},\ L \in \mathcal{L}(V),\ y \in K),\ w = (\mathbf{x} \in V, \gamma \in V')) \\ : P = \mathsf{Com}_{\mathsf{pp}}(\mathbf{x}, \gamma),\ y = L(\mathbf{x}) \big\}\,.\end{aligned} \tag{12}$$

The way we construct the compressed Σ-protocol in the dlog setting is by gradually composing protocols. We start with a basic Schnorr-like Σ-protocol, where one can see that the response message is significantly larger than the two previously sent. To lower communication costs, we observe that the last message is a PoK of a given relation, and that we can change it to another PoK of smaller size. Doing so recursively until the last message being sent only consists of two group elements, grants us a resulting protocol composition that has logarithmic communication costs in the size of the witness. This is however at the cost of increasing the number of messages being sent back and forth from constant to logarithmic.

4.1 Pedersen Vector Commitment Scheme

One of the building blocks for constructing compressed Σ-protocols in the discrete log setting, is a compact homomorphic vector commitment scheme. It allows a prover to commit to vectors of variable length in such a way that the commitments are all of the same size. The homomorphic property allows the prover and verifier to perform certain operations on the commited values locally to produce new commited values, while the values they perform the operations on remain hidden. An example of such a scheme is the Pedersen vector commitment scheme [Ped92], where a commitment will always consist of only one group element.

Definition 4.1 (Pedersen Vector Commitment [Ped92])**.** Let $\mathbb{G}$ be an abelian group of prime order q. The Pedersen vector commitment scheme is a compact homomorphic commitment scheme that is defined by the following PPT algorithms.

- $\mathsf{Gen}(1^\lambda) : \mathbf{g} = (g_1, \ldots, g_n) \leftarrow\!\!\$\ \mathbb{G}^n, h \leftarrow\!\!\$\ \mathbb{G}$, return $\mathsf{pp} := (\mathbf{g}, h)$.
- $\mathsf{Com}_{\mathsf{pp}}(\mathbf{x}, \gamma) : \mathbb{Z}_q^n \times \mathbb{Z}_q \to \mathbb{G} \quad (\mathbf{x}, \gamma) \mapsto h^\gamma \mathbf{g}^{\mathbf{x}} := h^\gamma \prod_{i=1}^n g_i^{x_i}$.

Notation wise, for $\mathbf{g} \in \mathbb{G}^n$, $\mathbf{x} \in \mathbb{Z}_q^n$ and $c \in \mathbb{Z}_q$ we let

$$\mathbf{g}^{\mathbf{x}} := \prod_{i=1}^{n} g_i^{x_i} \quad \text{as well as} \quad \mathbf{g}^c := (g_1^c, \ldots, g_n^c).$$

For $\mathbf{g}, \mathbf{h} \in \mathbb{G}^n$, we write the component-wise product as

$$\mathbf{g} * \mathbf{h}_2 := (g_1 h_1, \ldots, g_2, h_2).$$

Lemma 4.2. *The Pedersen Vector Commitment scheme is perfectly hiding and and computationally binding under the assumption that the prover does not know any non-trivial dlog solutions between any of the generators $g_1, \ldots, g_n, h$ in a given public parameter* pp*.*

Proof. We begin by proving that the commitment scheme is perfectly hiding. Given any commitment $c \in \mathbb{G}$, there is exactly one $\gamma \in \mathbb{Z}_q$ such that $\mathsf{Com}_{\mathsf{pp}}(\mathbf{x}, \gamma) = c$ for any $\mathbf{x} \in \mathbb{Z}_q^n$, as a consequence of h being a generator in group $\mathbb{G}$. Furthermore, every γ is chosen uniformly at random, and is independent of the message. Therefore the distribution of commitments is independent of the message choice, and each commitment have the same probability of occurring, and as a result the scheme is perfectly hiding.

To prove that the commitment scheme is computationally binding, we use a hybrid argument where we have the following sequence of games.

Game G_0: The first hybrid corresponds to the real version of the binding game for an adversary $\mathcal{A}$. An algorithm $\mathcal{B}$ samples $\mathsf{pp} = (g_1, \ldots, g_n, h)$ as the Gen algorithm in the commitment scheme would do, and forwards pp to $\mathcal{A}$. The adversary responds with the values $(c, \mathbf{x}, \gamma, \mathbf{x}', \gamma')$, and if it succeeds in breaking the binding property, the values satisfy $c = \mathsf{Com}_{\mathsf{pp}}(\mathbf{x}, \gamma) = \mathsf{Com}_{\mathsf{pp}}(\mathbf{x}', \gamma')$ with $\mathbf{x} \neq \mathbf{x}'$. We let $\mathsf{Adv}_{\text{binding}}(\mathcal{A})$ denote the probability that $\mathcal{A}$ successfully returns such values as in Definition 2.13, and we have that $\Pr[\mathbf{G}_0] = \mathsf{Adv}_{\text{binding}}(\mathcal{A})$.

Game G_1: In the next game, $\mathcal{B}$ changes the public parameters pp in the following way. It begins by sampling two elements $g_1, h \leftarrow\!\!\$\ \mathbb{G}$ uniformly at random. Then for $k = 2, \ldots, n$, it samples exponents $a_k \leftarrow\!\!\$\ \mathbb{Z}_q^*$ uniformly at random, and let $g_k := g_1^{a_k}$. For notation, we also define $a_1 = 1$. It then defines $\mathsf{pp} = (g_1, g_2, \ldots, g_n, h)$ and forwards it to adversary $\mathcal{A}$. Since each g^{a_k} is a generator in $\mathbb{G}$ and the exponents are chosen at random, the success probability in $\mathbf{G}_0$ equals the success probability in $\mathbf{G}_1$, as both public parameters are distributed correctly, and we have $\Pr[\mathbf{G}_0] = \Pr[\mathbf{G}_1]$.

Now, if algorithm $\mathcal{A}$ successfully breaks binding, algorithm $\mathcal{B}$ can use this as a subroutine to construct a solution to the dlog problem between two of the generators in a public parameters in $\mathbf{G}_1$ in the following way. Suppose that $(c, \mathbf{x}, \gamma, \mathbf{x}', \gamma')$ breaks binding, and say that $h = g_1^z$ implicitly. Then we have that

$$\mathsf{Com}_{\mathsf{pp}}(\mathbf{x}, \gamma) = \mathsf{Com}_{\mathsf{pp}}(\mathbf{x}', \gamma') \Leftrightarrow h^{\gamma} \prod_{i=1}^{n} g_i^{x_i} = h^{\gamma'} \prod_{i=1}^{n} g_i^{x_i'}$$

for $\mathbf{x} \neq \mathbf{x}'$ with non-negligible probability. Using g_1 as a base, we have in the exponents that

$$z \cdot \gamma + \sum_{i=1}^{n} x_i a_i = z \cdot \gamma' + \sum_{i=1}^{n} x_i' a_i,$$

and we can compute $z = (\gamma - \gamma')^{-1} \sum_{i=1}^{n} (x_i' - x_i) a_i$ since all the other values are known to us. This is a non-trivial dlog solution between generators g and h such that $g_1^z = h$, which contradicts our assumption that such pairs should be infeasible to compute for a given public parameter pp.

It follows that

$$\mathsf{Adv}_{\text{binding}}(\mathcal{A}) = \Pr[\mathbf{G}_0] = \Pr[\mathbf{G}_1] = \mathsf{Adv}_{\text{dlog}}(\mathcal{B}),$$

which proves that the Pedersen vector commitment scheme is computationally binding. □

Remark 4.3. Since the Pedersen vector commitment scheme is homomorphic, it follows that $\mathsf{Com}_{\mathsf{pp}}(a, b) \cdot \mathsf{Com}_{\mathsf{pp}}(c, d)^e = \mathsf{Com}_{\mathsf{pp}}(a + ce, b + de)$ for any public parameter pp. Also, if $\widehat{\mathbf{g}} = (\mathbf{g}, h) = (g_1, \ldots, g_n, h)$ and $\widehat{\mathbf{z}} = (\mathbf{z}, \phi) = (z_1, \ldots, z_n, \phi)$, then $\mathsf{Com}_{(\mathbf{g},h)}(\mathbf{z}, \phi) \cdot k^a = \mathsf{Com}_{(\widehat{\mathbf{g}},k)}(\widehat{\mathbf{z}}, a)$. Note that we let the length of the input to the commitment scheme be decided by the number of generators given in the public parameter pp.

4.2 Basic Σ-Protocol for Composition

We are now ready to introduce the basic Σ-protocol Π_0 for relation $\mathcal{R}_{\mathsf{com}}$ that we build the compressed protocol from, where we instantiate the relation as

$$\begin{aligned} \mathcal{R}_{\mathsf{com}} := \big\{ (X = (P \in \mathbb{G},\ L \in \mathcal{L}(\mathbb{Z}_q^n),\ y \in \mathbb{Z}_q),\ w = (\mathbf{x} \in \mathbb{Z}_q^n, \gamma \in \mathbb{Z}_q)) \\ : P = \mathsf{Com}_{(\mathbf{g},h)}(\mathbf{x}, \gamma) = \mathbf{g}^{\mathbf{x}} h^{\gamma},\ y = L(\mathbf{x}) \big\}. \end{aligned} \tag{13}$$

The Σ-protocol Π_0 for relation $\mathcal{R}_{\mathsf{com}}$ is given in Figure 4.

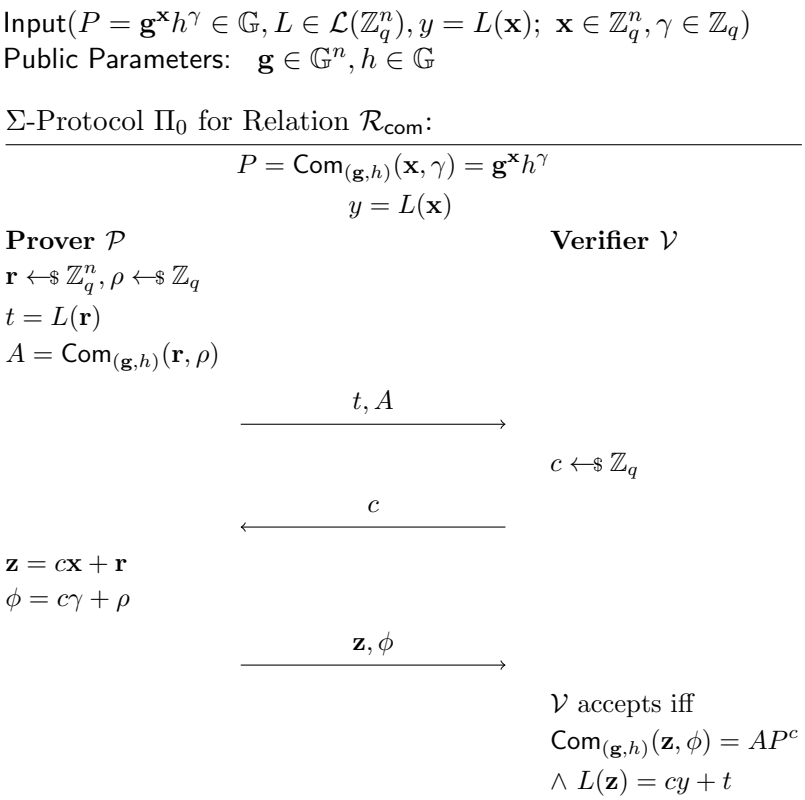


Figure 4: Σ-Protocol Π_0 for Relation $\mathcal{R}_{\mathsf{com}}$

Lemma 4.4 (Basic Pivot [AC20])**.** *The protocol Π_0 presented in* Figure 4 *is a Σ-protocol for relation $\mathcal{R}_{\mathsf{com}}$. In particular, it has perfect completeness and unconditional special soundness. The communication costs are as follows:*

- $\mathcal{P} \to \mathcal{V}$*:* 1 *elements of* $\mathbb{G}$ *and* $n + 2$ *elements of* $\mathbb{Z}_q$*.*
- $\mathcal{V} \to \mathcal{P}$*:* 1 *element of* $\mathbb{Z}_q$*.*

Proof. The following proof shares similarities with the proof of the Schnorr protocol [Sch90].

Completeness Suppose that a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ follow the protocol specifications of $\Pi_0(P, L, y; \mathbf{x}, \gamma)$ such that $((P, L, y), (\mathbf{x}, \gamma)) \in \mathcal{R}_{\mathsf{com}}$, and let $((t, A), c, (\mathbf{z}, \phi))$ be a non-aborting transcript. Then we have that

$$\mathsf{Com}_{(\mathbf{g},h)}(\mathbf{z}, \phi) = \mathsf{Com}_{(\mathbf{g},h)}(c\mathbf{x}+\mathbf{r}, c\gamma+\rho) = \mathsf{Com}_{(\mathbf{g},h)}(\mathbf{x}, \gamma)^c \cdot \mathsf{Com}_{(\mathbf{g},h)}(\mathbf{r}, \rho) = AP^c$$

as required. Also, it holds that

$$L(\mathbf{z}) = L(c\mathbf{x} + \mathbf{r}) = cL(\mathbf{x}) + L(\mathbf{r}) = cy + t,$$

which shows that protocol Π_0 has perfect completeness.

2-Special Soundness Let $\mathcal{E}$ be an extractor that is given common input $X = (P, L, y)$ and two accepting transcripts $((t, A), c, (\mathbf{z}, \phi))$ and $((t', A'), c', (\mathbf{z}', \phi'))$ with $c \neq c'$ for protocol Π_0. We want to construct a witness $(\tilde{\mathbf{x}}, \tilde{\gamma})$ such that $(X, (\tilde{\mathbf{x}}, \tilde{\gamma})) \in \mathcal{R}_{\mathsf{com}}$. Since both transcripts are accepting, we have that $L(\mathbf{z}) = cy + t$ as well as $L(\mathbf{z}') = c'y + t$. Subtracting the second equation from the first one, we obtain that $L(\mathbf{z} - \mathbf{z}') = (c - c')y$, and solving for y we get $y = L((c - c')^{-1}(\mathbf{z} - \mathbf{z}'))$. Therefore we define $\tilde{\mathbf{x}} := (c - c')^{-1}(\mathbf{z} - \mathbf{z}')$ as our candidate for the first part of the witness.

Furthermore, $\mathsf{Com}_{(\mathbf{g},h)}(\mathbf{z}, \phi) = AP^c$ and $\mathsf{Com}_{(\mathbf{g},h)}(\mathbf{z}', \phi') = AP^{c'}$. If we now divide the first equation by the second one and then solve for P, we can rewrite P as

$$P = \mathsf{Com}_{(\mathbf{g},h)}((c - c')^{-1}(\mathbf{z} - \mathbf{z}'), (c - c')^{-1}(\phi - \phi')).$$

Therefore we let $\tilde{\gamma} := (c - c')^{-1}(\phi - \phi')$. By how we defined $\tilde{\mathbf{x}}$ and $\tilde{\gamma}$, the pair $(\tilde{\mathbf{x}}, \tilde{\gamma})$ is a valid witness for (P, L, y). Therefore the extractor $\mathcal{E}$ is successful in extracting a valid witness from the given transcripts.

Special HVZK In Figure 5, we construct a simulator $\mathcal{S}$ for Π_0.

Simulator $\mathcal{S}\,(X = (P, L, y), c \in \mathbb{Z}_q)$:

1 : Sample $\mathbf{z} \leftarrow\!\$\ \mathbb{Z}_q^n$ and $\phi \leftarrow\!\$\ \mathbb{Z}_q$.
2 : Define $t := L(\mathbf{z}) - cy$ and $A := \mathsf{Com}_{(\mathbf{g},h)}(\mathbf{z}, \phi) \cdot P^{-c}$.
3 : **return** $\mathcal{T} = ((t, A), c, (\mathbf{z}, \phi))$

Figure 5: Simulator for Special HVZK of Protocol Π_0

Since $\mathbf{z}$ is chosen uniformly at random, t will be uniformly distributed as well. By the way we defined A, it can be written as $\mathsf{Com}_{(\mathbf{g},h)}(\mathbf{r}, \rho)$ for $\mathbf{r}$ and ρ implicitly defined as $\mathbf{r} = \mathbf{z} - c\mathbf{x}$ and $\rho = \phi - c\gamma$ by the expression above. Since $\mathbf{z}$ and ϕ are chosen uniformly at random, both $\mathbf{r}$ and ρ will be uniformly distributed as well, and we can argue that how we compute A in the simulated transcripts guarantees that it has the same distribution as a real one would. Therefore simulated transcripts and real ones have the same probability distributions, and we conclude that Π_0 is special HVZK. □

4.3 The Response Message as a PoK

The last message sent in protocol Π_0 consists of $n + 1$ elements of $\mathbb{Z}_q$. This accounts for almost all elements being sent by the prover, as it otherwise only sends one element from $\mathbb{G}$ and one from $\mathbb{Z}_q$. Therefore to lower communication complexity, we want to find a way to shorten it. We observe that the last message $\widehat{\mathbf{z}} = (\mathbf{z}, \phi) = (z_1, \ldots, z_n, \phi)$ is a PoK for $(P, L, y; \widehat{\mathbf{z}})$, where the relation it satisfies can be written as

$$\mathcal{R}_{\mathsf{com}_1} = \Big\{ (X = (\widehat{P} \in \mathbb{G},\ \widehat{L} \in \mathcal{L}(V),\ \widehat{y} \in K),\ w = \widehat{\mathbf{z}} \in \mathbb{Z}_q^{n+1}) \\ : \widehat{\mathbf{g}}^{\widehat{\mathbf{z}}} = \widehat{P} \wedge \widehat{y} = \widehat{L}(\widehat{\mathbf{z}}) \Big\}. \tag{14}$$

By use of protocol composition, we can compose protocol Π_0 with any protocol Π_1 for relation $\mathcal{R}_{\mathsf{com}_1}$. Then if messages sent in Π_1 are smaller in total than the last message sent in Π_0, we have effectively reduced communication costs to some extend, while we on the other hand have potentially increased the number of moves in the resulting protocol $\Pi_1 \diamond \Pi_0$. A protocol for relation $\mathcal{R}_{\mathsf{com}_1}$ is given in Figure 6.

In order to compose Π_0 with protocols that use the techniques from Bulletproofs [Bün+18], protocol Π_1 for relation $\mathcal{R}_{\mathsf{com}_1}$ is defined such that the last message is a PoK for an appropriate relation $\mathcal{R}_{\mathsf{com}_2}$. We define this relation by

$$\mathcal{R}_{\mathsf{com}_2} = \Big\{ (X = (Q \in \mathbb{G}, \widetilde{L} \in \mathcal{L}(\mathbb{Z}_q^{n+1}),\ w = \widehat{\mathbf{z}} \in \mathbb{Z}_q^{n+1}) \\ : Q = \mathsf{Com}_{(\widehat{\mathbf{g}},k)}(\widehat{\mathbf{z}}, \widetilde{L}(\widehat{\mathbf{z}})) \Big\}, \tag{15}$$

and note that difference between $\mathcal{R}_{\mathsf{com}_2}$ and the previous relation $\mathcal{R}_{\mathsf{com}_1}$ is that we have rewritten the constraints in such a way that there is only one condition that has to be satisfied.

$\mathsf{Input}(\widehat{P} = \widehat{\mathbf{g}}^{\widehat{\mathbf{z}}} \in \mathbb{G}, \widehat{L} \in \mathcal{L}(\mathbb{Z}_q^{n+1}), \widehat{y} = \widehat{L}(\widehat{\mathbf{z}});\ \widehat{\mathbf{z}} \in \mathbb{Z}_q^{n+1})$
$\mathsf{Public\ Parameters}:\quad \widehat{\mathbf{g}} \in \mathbb{G}^{n+1}, k \in \mathbb{G}$

Argument of Knowledge Π_1 for Relation $\mathcal{R}_{\mathsf{com}_1}$:

$\widehat{P} = \mathbf{g}^{\widehat{\mathbf{z}}} \in \mathbb{G}$
$\widehat{y} = \widehat{L}(\widehat{\mathbf{z}}) \in \mathbb{Z}_q$

Prover $\mathcal{P}$		**Verifier** $\mathcal{V}$
		$c \leftarrow\!\!\$\ \mathbb{Z}_q$
	$\xleftarrow{\quad c \quad}$	
	$\xrightarrow{\quad \widehat{\mathbf{z}} \quad}$	
		$\mathcal{V}$ accepts iff $\widehat{\mathbf{g}}^{\widehat{\mathbf{z}}} k^{c\widetilde{L}(\widehat{\mathbf{z}})} = \widehat{P} k^{c\widehat{y}}$

Figure 6: Argument of Knowledge Π_1 for $\mathcal{R}_{\mathsf{com}_1}$

Definition 4.5 (Discrete Logarithm Relation [Bün+18])**.** For all PPT adversaries $\mathcal{A}$ and $n \geq 2$, the discrete logarithm relation problem states that the probability that adversary $\mathcal{A}$ outputs values $(a_1, \ldots, a_n)$ that satisfies the given condition is negligible:

$$\Pr\left[\begin{array}{c} g_1, \ldots, g_n \leftarrow\!\!\$\ \mathbb{G}; \\ (a_1, \ldots, a_n) \leftarrow\!\!\$\ \mathcal{A}(\mathbb{G}, g_1, \ldots, g_n) \end{array} : \exists a_i \neq 0 \wedge \prod_{i=1}^{n} g_i^{a_i} = 1\right] \leq 2^{-\lambda}.$$

Lemma 4.6 (Protocol Π_1 [AC20])**.** *Protocol Π_1 in Figure 6 is a perfectly complete and computationally 2-special sound 2-move protocol under the discrete logarithm relation (Definition 4.5) assumption. The communication costs are as follows:*

- $\mathcal{P} \to \mathcal{V}$*: and $n+1$ elements of $\mathbb{Z}_q$.*
- $\mathcal{V} \to \mathcal{P}$*: 1 element of $\mathbb{Z}_q$.*

Proof. As completeness is apparent, we focus on the special soundness proof.

2-Special Soundness Let $\mathcal{E}$ be an extractor that is given common input $X = (\widehat{P}, \widehat{L}, \widehat{y})$ and two accepting transcripts $(c_0, \widehat{\mathbf{z}}_0)$ and $(c_1, \widehat{\mathbf{z}}_1)$, where $c_0 \neq c_1$ for protocol Π_1. Our goal is to construct a witness $\tilde{\mathbf{z}}$ such that $(X, \tilde{\mathbf{z}}) \in \mathcal{R}_{\mathsf{com}_1}$. Since both transcripts are accepting, we have that

$$\widehat{\mathbf{g}}^{\widehat{\mathbf{z}}_0} k^{c_0 \widetilde{L}(\widehat{\mathbf{z}}_0)} = \widehat{P} k^{c_0 \widehat{y}} \quad \text{and} \quad \widehat{\mathbf{g}}^{\widehat{\mathbf{z}}_1} k^{c_1 \widetilde{L}(\widehat{\mathbf{z}}_1)} = \widehat{P} k^{c_1 \widehat{y}}.$$

We divide the first expression by the second one, and obtain that

$$\widehat{\mathbf{g}}^{\widehat{\mathbf{z}}_0 - \widehat{\mathbf{z}}_1} k^{c_0 \widetilde{L}(\widehat{\mathbf{z}}_0) - c_1 \widetilde{L}(\widehat{\mathbf{z}}_1)} = k^{(c_0 - c_1)\widehat{y}}.$$

Here, we either have that $\widehat{\mathbf{z}}_0 = \widehat{\mathbf{z}}_1$ or not. If it is not the case, this would contradict the discrete log relation assumption, as

$$\widehat{\mathbf{g}}^{\widehat{\mathbf{z}}_0 - \widehat{\mathbf{z}}_1} k^{c_0(\widetilde{L}(\widehat{\mathbf{z}}_0) - \widehat{y}) - c_1(\widetilde{L}(\widehat{\mathbf{z}}_1) - \widehat{y})} = 1$$

would be a non-trivial solution. Therefore it must be the case that $\widehat{\mathbf{z}}_0 = \widehat{\mathbf{z}}_1$, from which it follows that

$$c_0 \widetilde{L}(\widehat{\mathbf{z}}_0) - c_1 \widetilde{L}(\widehat{\mathbf{z}}_1) = (c_0 - c_1)\widehat{y}.$$

Therefore we obtain that $\widetilde{L}(\widehat{\mathbf{z}}_0) = \widetilde{L}(\widehat{\mathbf{z}}_1) = \widehat{y}$, and $\mathcal{E}$ can return $\widehat{\mathbf{z}}_0$ as a valid witness such that $(X, \widehat{\mathbf{z}}_0) \in \mathcal{R}_{\mathsf{com}_1}$. □

4.4 The Compression Mechanism

Suppose that n is a power of 2. For $\mathbf{g} \in \mathbb{G}^n$ and $\mathbf{a} \in \mathbb{Z}_q^{(n+1)/2}$, define

$$\mathbf{g}_L^{\mathbf{a}} := \mathbf{g}^{(\mathbf{a}\ 0)^\top} \in \mathbb{G}^{(n+1)/2} \quad \text{and} \quad \mathbf{g}_R^{\mathbf{a}} := \mathbf{g}^{(0\ \mathbf{a})^\top} \in \mathbb{G}^{(n+1)/2}, \tag{16}$$

where $(\mathbf{a}\ 0)^\top \in \mathbb{Z}_q^n$ is the vector $\mathbf{a}$ appended with $(n-1)/2$ zeros in the end, such that it is a vector of dimension n. Similarly, $(0\ \mathbf{a})^\top \in \mathbb{Z}_q^n$ is the vector $\mathbf{a}$ appended with $(n-1)/2$ zeros in the front, such that it is a vector of dimension n. For $\mathbf{g} \in \mathbb{G}^n$, we define

$$\mathbf{g}_L = (g_1, \ldots, g_{n/2}) \quad \text{and} \quad \mathbf{g}_R = (g_{n/2+1}, \ldots, g_n), \tag{17}$$

and for a linear form $L : \mathbb{Z}_q^m \to \mathbb{Z}_q$, define

$$L_L : \mathbb{Z}_q^{(m-1)/2} \to \mathbb{Z}_q,\ \mathbf{x} \mapsto L(\begin{pmatrix}\mathbf{x}\\0\end{pmatrix}) \quad \text{and} \quad L_R : \mathbb{Z}_q^{(m-1)/2} \to \mathbb{Z}_q,\ \mathbf{x} \mapsto L(\begin{pmatrix}0\\\mathbf{x}\end{pmatrix}).$$

The compression mechanism Π_2 for relation $\mathcal{R}_{\mathsf{com}_2}$ (Equation 15) is given in Figure 7. The last step of the protocol halves the length of the last message being sent for each recursive call, and the number of times we repeat it depends on the length of the witness. Note that the verification constraint is stated in two different ways, where we have

$$\mathsf{Com}_{(\mathbf{g}',k)}(\mathbf{z}', L'(\mathbf{z}')) = \mathsf{Com}_{(\widehat{\mathbf{g}},k)}\left(\begin{pmatrix}c\mathbf{z}'\\\mathbf{z}'\end{pmatrix},\ \widetilde{L}(\begin{pmatrix}c\mathbf{z}'\\\mathbf{z}'\end{pmatrix})\right).$$

This equality can be obtained by combining that

$$L'(\mathbf{z}) = (c\widetilde{L}_L + \widetilde{L}_R)(\mathbf{z}') = cL(\begin{pmatrix}\mathbf{z}'\\0\end{pmatrix}) + \widetilde{L}_R(\begin{pmatrix}0\\\mathbf{z}'\end{pmatrix}) = \widetilde{L}(\begin{pmatrix}c\mathbf{z}'\\\mathbf{z}'\end{pmatrix})$$

and

$$\prod_{i=1}^{n+1} \widehat{\mathbf{g}}^{(c\mathbf{z}'\ \mathbf{z}')^\top} = \prod_{i=1}^{(n+1)/2} (\widehat{\mathbf{g}}_L^c * \widehat{\mathbf{g}}_R)^{\mathbf{z}'},$$

such that we get

$$\begin{aligned}
\mathsf{Com}_{(\widehat{\mathbf{g}},k)}\left(\begin{pmatrix}c\mathbf{z}'\\\mathbf{z}'\end{pmatrix},\ \widetilde{L}(\begin{pmatrix}c\mathbf{z}'\\\mathbf{z}'\end{pmatrix})\right) &= \mathsf{Com}_{(\widehat{\mathbf{g}},k)}\left(\begin{pmatrix}c\mathbf{z}'\\\mathbf{z}'\end{pmatrix},\ L'(\mathbf{z}')\right)\\
&= \prod_{i=1}^{n+1} \widehat{\mathbf{g}}^{(c\mathbf{z}'\ \mathbf{z}')^\top} \cdot k^{L'(\mathbf{z}')}\\
&= \prod_{i=1}^{(n+1)/2} (\widehat{\mathbf{g}}_L^c * \widehat{\mathbf{g}}_R)^{\mathbf{z}'} \cdot k^{L'(\mathbf{z}')}\\
&= \mathsf{Com}_{(\mathbf{g}',k)}(\mathbf{z}', L'(\mathbf{z}')).
\end{aligned}$$

$\mathsf{Input}(Q = \widehat{\mathbf{g}}^{\widehat{\mathbf{z}}} k^{\widetilde{L}(\widehat{\mathbf{z}})} \in \mathbb{G}, \widetilde{L} \in \mathcal{L}(\mathbb{Z}_q^{n+1});\ \widehat{\mathbf{z}} \in \mathbb{Z}_q^{n+1})$
$\mathsf{Public\ Parameters:}\ \ \widehat{\mathbf{g}}, k$

Compressed PoK Π_2 for Relation $\mathcal{R}_{\mathsf{com}_2}$:

$$Q = \mathsf{Com}_{(\widehat{\mathbf{g}},k)}(\widehat{\mathbf{z}}, \widetilde{L}(\widehat{\mathbf{z}}))$$
$$\widehat{\mathbf{g}} \in \mathbb{G}^{n+1}, k \in \mathbb{G}$$

Prover $\mathcal{P}$		**Verifier** $\mathcal{V}$
$A = \mathsf{Com}_{(\widehat{\mathbf{g}},k)}\left(\begin{pmatrix}0\\ \widehat{\mathbf{z}}_L\end{pmatrix}, \widetilde{L}(\begin{pmatrix}0\\ \widehat{\mathbf{z}}_L\end{pmatrix})\right)$		
$B = \mathsf{Com}_{(\widehat{\mathbf{g}},k)}\left(\begin{pmatrix}\widehat{\mathbf{z}}_R\\ 0\end{pmatrix}, \widetilde{L}(\begin{pmatrix}\widehat{\mathbf{z}}_R\\ 0\end{pmatrix})\right)$		
	$\xrightarrow{\quad A, B \quad}$	
		$c \leftarrow\!\!\$\ \mathbb{Z}_q$
	$\xleftarrow{\quad c \quad}$	
	$\mathbf{g}' := \widehat{\mathbf{g}}_L^c * \widehat{\mathbf{g}}_R \in \mathbb{G}^{(n+1)/2}$	
	$Q' := AQ^c B^{c^2}$	
	$L' := c\widetilde{L}_L + \widetilde{L}_R$	
$\mathbf{z}' = \widehat{\mathbf{z}}_L + c\widehat{\mathbf{z}}_R$		
if $\mathbf{z}' \in \mathbb{Z}_q^2$:		
	$\xrightarrow{\quad \mathbf{z}' \quad}$	
		$\mathcal{V}$ accepts iff $\mathsf{Com}_{(\mathbf{g}',k)}(\mathbf{z}', L'(\mathbf{z}')) = Q' \Leftrightarrow$ $\mathsf{Com}_{(\widehat{\mathbf{g}},k)}\left(\begin{pmatrix}c\mathbf{z}'\\ \mathbf{z}'\end{pmatrix},\ \widetilde{L}(\begin{pmatrix}c\mathbf{z}'\\ \mathbf{z}'\end{pmatrix})\right) = Q'$
else :		
	Run $\Pi_2(Q', L';\ \mathbf{z}')$ with $\mathsf{Public\ Parameters:}\ \mathbf{g}', k$	

Figure 7: Compressed Proof of Knowledge Π_2 for Relation $\mathcal{R}_{\mathsf{com}_2}$

Theorem 4.7 (Compression Mechanism [AC20]). *Assume that $n+1$ is a power of 2. The protocol Π_2 is a $(2\mu+1)$-move protocol for relation $\mathcal{R}_{\mathsf{com}_2}$, where $\mu = \lceil \log_2(n+1) \rceil - 1$. It is perfectly complete and unconditionally $(k_1 \ldots, k_\mu)$-special sound, where $k_i = 3$ for all $i = 1, \ldots, \mu$. Moreover, the communication costs are*

- $\mathcal{P} \to \mathcal{V}$: $2 \cdot \lceil \log_2(n+1) \rceil - 2$ *elements of* $\mathbb{G}$ *and* 2 *elements of* $\mathbb{Z}_q$.
- $\mathcal{V} \to \mathcal{P}$: $\lceil \log_2(n+1) \rceil - 1$ *elements of* $\mathbb{Z}_q$.

Proof. We let μ denote the number of recursive calls of Π_2, where we count the initial call of Π_2 as well. The witness $\widehat{\mathbf{z}}$ is an element of $\mathbb{Z}_q^{n+1}$, and since $n+1 = 2^m$ for some m, the length of the witness equals $\log_2(n+1)$.

The intermediate witness $\mathbf{z}'$ that we rerun Π_2 on in each recursive call is halved for each iteration, and Π_2 returns when $\mathbf{z}'$ has length 2, namely $\mathbf{z}' \in \mathbb{Z}_q^2$. Therefore we have that $\mu = \log_2(n+1) - 1$, since we stop before the resulting witness is of length 1. For each recursive call, two moves take place in the protocol, and the protocol finishes by the prover forwarding $\mathbf{z}'$ to the verifier. Therefore Π_2 is a $(2\mu+1)$-move protocol.

The number of $\mathbb{G}$ elements sent by the prover equals $2\mu = 2\log_2(n+1) - 2$, since the prover sends two elements of $\mathbb{G}$ for each time Π_2 calls itself. It also finishes the protocol by forwarding $\mathbf{z}'$, which as we argued above consists of two elements of $\mathbb{Z}_q$.

Furthermore, the number of messages sent by the verifier equals μ, as it forwards one element of $\mathbb{Z}_q$ for each time we call Π_2. Therefore the communication costs for the prover to the verifier is $2\log_2(n+1) - 2$ elements of $\mathbb{G}$ and 2 elements of $\mathbb{Z}_q$, and from the verifier to the prover is $\log_2(n+1) - 1$ elements of $\mathbb{Z}_q$.

Completeness Suppose that a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ follow the protocol specifications of $\Pi_2(Q, \widetilde{L}; \widehat{\mathbf{z}})$ such that $((Q, \widetilde{L}), \widehat{\mathbf{z}}) \in \mathcal{R}_{\mathsf{com}_2}$, and let $((A_1, B_1), c_1, (A_2, B_2), c_2, \ldots, (A_\mu, B_\mu), c_\mu, \mathbf{z}')$ be a non-aborting transcript. To keep track of notation, we also denote variables $\mathbf{g}_i, L_i, \mathbf{z}_i, Q_i$ with indices to show which recursion call it was defined in, where the values from the common input and public parameters $\widehat{\mathbf{g}}, \widetilde{L}, Q$ and $\widehat{\mathbf{z}}$ in the beginning of the protocol are indexed with 0 as $\mathbf{g}_0, L_0, \mathbf{z}_0$ and Q_0.

Note that at any recursion step i of the protocol, the verification check holds for the implicitly defined $\mathbf{z}_i$. Therefore we can prove completeness of the protocol with an induction argument.

We begin by proving the base case $k = 1$. That is, we want to prove that

$$\mathsf{Com}_{(\mathbf{g}_0, k)}\left(\begin{pmatrix} c_1\mathbf{z}_1 \\ \mathbf{z}_1 \end{pmatrix}, L_1(\begin{pmatrix} c_1\mathbf{z}_1 \\ \mathbf{z}_1 \end{pmatrix})\right) = Q_1 = A_1 Q_0^{c_1} B_1^{c_1^2}.$$

Using the definitions and properties of the homomorphic commitment scheme, we have that

$$\begin{aligned}
&\mathsf{Com}_{(\mathbf{g}_0,k)}\left(\begin{pmatrix} c_1\mathbf{z}_1 \\ \mathbf{z}_1 \end{pmatrix}, L_1(\begin{pmatrix} c_1\mathbf{z}_1 \\ \mathbf{z}_1 \end{pmatrix})\right) \\
&= \mathsf{Com}_{(\mathbf{g}_0,k)}\left(\begin{pmatrix} c_1\mathbf{z}_{0_L} + c_1^2\mathbf{z}_{0_R} \\ \mathbf{z}_{0_L} + c_1\mathbf{z}_{0_R} \end{pmatrix}, L_1(\begin{pmatrix} c_1\mathbf{z}_{0_L} + c_1^2\mathbf{z}_{0_R} \\ \mathbf{z}_{0_L} + c_1\mathbf{z}_{0_R} \end{pmatrix})\right) \\
&= A_1 \cdot Q_0^{c_1} \cdot B_1^{c_1^2} = Q_1,
\end{aligned}$$

where

$$\begin{aligned}
A_1 &= \mathsf{Com}_{(\mathbf{g}_0,k)}\left(\begin{pmatrix} 0 \\ \mathbf{z}_{0_L} \end{pmatrix}, L_0(\begin{pmatrix} 0 \\ \mathbf{z}_{0_L} \end{pmatrix})\right), \\
Q_0 &= \mathsf{Com}_{(\mathbf{g}_0,k)}\left(\mathbf{z}_0, L_0(\mathbf{z}_0)\right), \\
B_1 &= \mathsf{Com}_{(\mathbf{g}_0,k)}\left(\begin{pmatrix} \mathbf{z}_{0_L} \\ 0 \end{pmatrix}, L_0(\begin{pmatrix} \mathbf{z}_{0_L} \\ 0 \end{pmatrix})\right).
\end{aligned}$$

This proves the base case.

Now assume that the verification check

$$\mathsf{Com}_{(\mathbf{g}_{\mu-1},k)}(\mathbf{z}_{\mu-1}, L_{\mu-1}(\mathbf{z}_{\mu-1})) = Q_{\mu-1}$$

holds for $k = \mu - 1$. Then for $k + 1 = \mu$, we then have that

$$\mathsf{Com}_{(\mathbf{g}_\mu,k)}(\mathbf{z}_\mu, L_\mu(\mathbf{z}_\mu)) = \mathsf{Com}_{(\mathbf{g}_{\mu-1},k)}\left(\begin{pmatrix} c_\mu\mathbf{z}_\mu \\ \mathbf{z}_\mu \end{pmatrix}, L_\mu(\begin{pmatrix} c_\mu\mathbf{z}_\mu \\ \mathbf{z}_\mu \end{pmatrix})\right). \tag{18}$$

Since $\mathbf{z}_\mu = \mathbf{z}_{\mu-1_L} + c_\mu\mathbf{z}_{\mu-1_R}$ and we have

$$A_\mu = \mathsf{Com}_{(\mathbf{g}_{\mu-1},k)}\left(\begin{pmatrix} 0 \\ \mathbf{z}_{\mu-1_L} \end{pmatrix}, L_{\mu-1}(\begin{pmatrix} 0 \\ \mathbf{z}_{\mu-1_L} \end{pmatrix})\right)$$

as well as

$$B_\mu = \mathsf{Com}_{(\mathbf{g}_{\mu-1},k)}\left(\begin{pmatrix} \mathbf{z}_{\mu-1_R} \\ 0 \end{pmatrix}, L_{\mu-1}(\begin{pmatrix} \mathbf{z}_{\mu-1_R} \\ 0 \end{pmatrix})\right),$$

we can rewrite Equation 18 as

$$\mathsf{Com}_{(\mathbf{g}_\mu,k)}(\mathbf{z}_\mu, L_\mu(\mathbf{z}_\mu)) = A_\mu \cdot \mathsf{Com}_{(\mathbf{g}_{\mu-1},k)}(\mathbf{z}_{\mu-1}, L_{\mu-1}(\mathbf{z}_{\mu-1}))^{c_\mu} \cdot B_\mu^{c_\mu^2}.$$

By the assumption that the verification check holds for $k = \mu - 1$, we have that

$$\mathsf{Com}_{(\mathbf{g}_{\mu-1},k)}(\mathbf{z}_{\mu-1}, L_{\mu-1}(\mathbf{z}_{\mu-1})) = Q_{\mu-1},$$

which shows that the verification check

$$\mathsf{Com}_{(\mathbf{g}_\mu,k)}(\mathbf{z}_\mu, L_\mu(\mathbf{z}_\mu)) = A_\mu \cdot Q_{\mu-1}^{c_\mu} \cdot B_\mu^{c_\mu^2}$$

holds for $k+1 = \mu$ as well. Therefore we can conclude that protocol Π_2 has perfect completeness.

3-Special Soundness In the special soundness proof, we assume that protocol Π_2 only runs the recursive step once. Let $\mathcal{E}$ be an extractor that is given common input $(Q, \widetilde{L})$, generators $\widehat{\mathbf{g}}, k$ and three accepting transcripts $(A, B, c_1, \mathbf{z}'_1), (A, B, 2_2, \mathbf{z}'_2)$ and $(A, B, c_3, \mathbf{z}'_3)$, with $c_i \neq c_j$ for all i, j.

For the sake of clarity in this proof, we define

$$\mathsf{Com}\left(\mathbf{a}\right) := \mathsf{Com}_{(\widehat{\mathbf{g}},k)}\left(\mathbf{a}, \widetilde{L}(\mathbf{a})\right) = \widehat{\mathbf{g}}^{\mathbf{a}} k^{\widetilde{L}(\mathbf{a})}.$$

Using the values given to $\mathcal{E}$, we want to return a value $\bar{\mathbf{z}}$ such that $((Q, \widetilde{L}), \bar{\mathbf{z}}) \in \mathcal{R}_{\mathsf{com}_2}$. That is, we want to construct a $\bar{\mathbf{z}}$ such that

$$Q = \mathsf{Com}(\widehat{\mathbf{z}}) = \mathsf{Com}(\bar{\mathbf{z}}).$$

Since the verification checks hold, we know that

$$\begin{aligned}
\mathsf{Com}_{(\mathbf{g}',k)}\left(\mathbf{z}'_i, \widetilde{L}(\mathbf{z}'_i)\right) &= \mathsf{Com}(\begin{pmatrix} c_i \cdot \mathbf{z}'_i \\ \mathbf{z}'_i \end{pmatrix}) = Q' = AQ^{c_i}B^{c_i^2} \\
&= \mathsf{Com}(\begin{pmatrix} 0 \\ \widehat{\mathbf{z}}_L \end{pmatrix}) \cdot \mathsf{Com}(\widehat{\mathbf{z}})^{c_i} \cdot \mathsf{Com}(\begin{pmatrix} \widehat{\mathbf{z}}_R \\ 0 \end{pmatrix})^{c_i^2} \\
&= \mathsf{Com}(\begin{pmatrix} 0 \\ \widehat{\mathbf{z}}_L \end{pmatrix} + c_i \cdot \widehat{\mathbf{z}} + c_i^2 \cdot \begin{pmatrix} \widehat{\mathbf{z}}_R \\ 0 \end{pmatrix})
\end{aligned}$$

for $i = 1, 2, 3$.

Written differently, the equations can be combined such that the commitments of each row of

$$\begin{pmatrix} 1 & c_1 & c_1^2 \\ 1 & c_2 & c_2^2 \\ 1 & c_3 & c_3^2 \end{pmatrix} \begin{pmatrix} \begin{pmatrix} 0 \\ \widehat{\mathbf{z}}_L \end{pmatrix} \\ \widehat{\mathbf{z}} \\ \begin{pmatrix} \widehat{\mathbf{z}}_R \\ 0 \end{pmatrix} \end{pmatrix} \tag{19}$$

equals the commitments of each row of

$$\begin{pmatrix} \begin{pmatrix} c_1 \cdot \mathbf{z}'_1 \\ \mathbf{z}'_1 \end{pmatrix} \\ \begin{pmatrix} c_2 \cdot \mathbf{z}'_2 \\ \mathbf{z}'_2 \end{pmatrix} \\ \begin{pmatrix} c_3 \cdot \mathbf{z}'_3 \\ \mathbf{z}'_3 \end{pmatrix} \end{pmatrix} \in \mathbb{G}^3. \tag{20}$$

we define

$$V := \begin{pmatrix} 1 & c_1 & c_1^2 \\ 1 & c_2 & c_2^2 \\ 1 & c_3 & c_3^2 \end{pmatrix},$$

and note that this is a Vandermonde matrix. Therefore its determinant

$$\det(V) = (c_3 - c_1)(c_3 - c_2)(c_2 - c_1) \neq 0$$

is non-zero by the assumption that $c_i \neq c_j$ for all combinations of i, j, and V is invertible over $\mathbb{Z}_q$. Our goal is to compute $\bar{\mathbf{z}}$ from values that we know such that $\mathsf{Com}(\widehat{\mathbf{z}}) = \mathsf{Com}(\bar{\mathbf{z}})$. If we let

$$(a_1 \ a_2 \ a_3) := (0 \ 1 \ 0) \cdot V^{-1},$$

we can multiply both Equation 19 and Equation 20 by it to obtain that

$$\mathsf{Com}(\widehat{\mathbf{z}}) = \mathsf{Com}\begin{pmatrix} \sum_{i=1}^{3} a_i c_i \mathbf{z}'_i \\ \sum_{i=1}^{3} a_i \mathbf{z}'_i \end{pmatrix}.$$

All of the values inside the right side are known to us, and we therefore let

$$\bar{\mathbf{z}} := \begin{pmatrix} \sum_{i=1}^{3} a_i c_i \mathbf{z}'_i \\ \sum_{i=1}^{3} a_i \mathbf{z}'_i \end{pmatrix}.$$

This is a valid witness for $(Q, \widetilde{L})$, and the extractor has successfully computed valid witness for the relation $\mathcal{R}_{\mathsf{com}_2}$. □

4.5 Compressed Σ-Protocol from Dlog

Composing the protocols Π_0, Π_1 and Π_2 gives us the final interactive protocol $\Pi_c = \Pi_2 \diamond \Pi_1 \diamond \Pi_0$ for relation $\mathcal{R}_{\mathsf{com}}$ presented in Figure 8, which is the finalized compressed Σ-protocol.

Theorem 4.8 (Compressed Σ-protocol [AC20])**.** *Suppose that $n + 1$ is a power of 2. The protocol Π_c given in Figure 8 is a $(2\mu+3)$-move interactive protocol for relation $\mathcal{R}_{\mathsf{com}}$, where $\mu = \lceil \log_2(n+1) \rceil - 1$. It is perfectly complete, has computational $(2, 2, k_1, \ldots, k_\mu)$-special soundness with $k_i = 3$ for $i = 1, \ldots, \mu$ under the discrete log relation assumption, and is special HVZK. The communication costs are as follows:*

- $\mathcal{P} \to \mathcal{V}$*:* $2 \cdot \lceil \log_2(n+1) \rceil - 1$ *elements of* $\mathbb{G}$ *and* 3 *elements of* $\mathbb{Z}_q$*.*
- $\mathcal{V} \to \mathcal{P}$*:* $\lceil \log_2(n+1) \rceil + 1$ *elements of* $\mathbb{Z}_q$*.*

Proof. As we have previously seen, all the protocols Π_0, Π_1 and Π_2 have completeness, and therefore the composed protocol Π_c will also have the property. We have also shown that Π_0 and Π_1 have 2-special soundness (Lemma 4.2, Lemma 4.6), while Π_2 is $(k_1, \ldots, k_\mu)$-special sound with $k_i = 3$ where $i = 1, \ldots, \mu$ (Theorem 4.7). Combining the extractors then gives us one for the composed protocol as well. Lastly, we have Special HVZK since protocol Π_0 has the property (Lemma 4.2), so running the simulator to construct a valid transcript for Π_0 and then using the values from it to continue honest executions of Π_1 and Π_2 as a part of the composed protocol allows us to simulate transcripts for Π_c as well. □

$\mathsf{Input}(P = \mathsf{Com}_{(\mathbf{g},h)}(\mathbf{x},\gamma) = \mathbf{g}^{\mathbf{x}}h^{\gamma} \in \mathbb{G}, L \in \mathcal{L}(\mathbb{Z}_q^n), y = L(\mathbf{x});\ \mathbf{x} \in \mathbb{Z}_q^n, \gamma \in \mathbb{Z}_q)$
Public Parameters: $\mathbf{g}, h, k$

Compressed Σ-Protocol $\Pi_c := \Pi_2 \diamond \Pi_1 \diamond \Pi_0$ for Relation $\mathcal{R}_{\mathsf{com}}$:

$$P = \mathsf{Com}_{(\mathbf{g},h)}(\mathbf{x},\gamma) = \mathbf{g}^{\mathbf{x}}h^{\gamma} \in \mathbb{G}$$
$$y = L(\mathbf{x}) \in \mathbb{Z}_q$$

Prover $\mathcal{P}$		**Verifier** $\mathcal{V}$
$\mathbf{r} \leftarrow\!\!\$\ \mathbb{Z}_q^n, \rho \leftarrow\!\!\$\ \mathbb{Z}_q$		
$t = L(\mathbf{r}),\ \widehat{A} = \mathsf{Com}_{(\mathbf{g},h)}(\mathbf{r},\rho)$		
	$\xrightarrow{t, \widehat{A}}$	
		$c_0, c_1 \leftarrow\!\!\$\ \mathbb{Z}_q$
	$\xleftarrow{c_0, c_1}$	
$\mathbf{z} = c_0\mathbf{x} + \mathbf{r},\ \phi = c_0\gamma + \rho$		
$\widehat{\mathbf{z}} := (\mathbf{z}, \phi)$	$\widehat{\mathbf{g}} := (\mathbf{g}, h)$	
	$\widetilde{L}(\mathbf{z},\phi) := c_1 \cdot L(\mathbf{z})$	
	$Q := \widehat{A}P^{c_0}k^{c_1(c_o y + t)}$	
	$= \mathsf{Com}_{(\widehat{\mathbf{g}},k)}(\widehat{\mathbf{z}}, \widetilde{L}(\widehat{\mathbf{z}}))$	
$A = \mathsf{Com}_{(\widehat{\mathbf{g}},k)}\left(\begin{pmatrix}0\\ \widehat{\mathbf{z}}_L\end{pmatrix}, \widetilde{L}(\begin{pmatrix}0\\ \widehat{\mathbf{z}}_L\end{pmatrix})\right)$		
$B = \mathsf{Com}_{(\widehat{\mathbf{g}},k)}\left(\begin{pmatrix}\widehat{\mathbf{z}}_R\\ 0\end{pmatrix}, \widetilde{L}(\begin{pmatrix}\widehat{\mathbf{z}}_R\\ 0\end{pmatrix})\right)$	$\xrightarrow{A, B}$	
		$c \leftarrow\!\!\$\ \mathbb{Z}_q$
	$\xleftarrow{c}$	
	$\mathbf{g}' := \widehat{\mathbf{g}}_L^c * \widehat{\mathbf{g}}_R \in \mathbb{G}^{(n+1)/2}$	
	$Q' := AQ^cB^{c^2}$	
$\mathbf{z}' = \widehat{\mathbf{z}}_L + c\widehat{\mathbf{z}}_R$	$L' := c\widetilde{L}_L + \widetilde{L}_R$	
if $\mathbf{z}' \in \mathbb{Z}_q^2$:		
	$\xrightarrow{\mathbf{z}'}$	
		$\mathcal{V}$ accepts iff $\mathsf{Com}_{(\mathbf{g}',k)}(\mathbf{z}', L'(\mathbf{z}')) = Q' \Leftrightarrow$ $\mathsf{Com}_{(\widehat{\mathbf{g}},k)}\left(\begin{pmatrix}c\mathbf{z}'\\ \mathbf{z}'\end{pmatrix}, \widetilde{L}(\begin{pmatrix}c\mathbf{z}'\\ \mathbf{z}'\end{pmatrix})\right) = Q'$
else :	Run $\Pi_2(Q', L';\ \mathbf{z}')$ with Public Parameters: $\mathbf{g}', k$	

Figure 8: Compressed Σ-Protocol from Dlog Assumptions

5 Lattice Cryptography

In the following sections, our goal is to look at how to construct a Schnorr-like [Sch90] Σ-protocol from lattice assumptions, as presented by Lyubashevsky [Lyu24, Section 5.2]. Doing so reveals some of the challenges we meet when constructing lattice-based Σ-protocols and techniques we can use to overcome them. Afterwards, we look into how we can apply the OR-protocol from Figure 3 to the construction. To do so, it is necessary to be able to perform operations on elements of the challenge set, or bit strings that have a one-to-one correspondence to them. We will see how the function SampleInBall from the NIST Standard [Nat24, Algorithm 29] can be used to do obtain this desired property.

The Challenge Set When defining the challenge set we use in the Σ-protocol, we keep in mind that we want the values in it to have small norm, as we want to prove a statement that has a requirement on the size of the witness, and also that it is favorable to choose a set with low sample complexity. For $\mathcal{R}_{q,f}$ as defined in Section 2.1, we therefore define the challenge set $\mathcal{C} \subseteq \mathcal{R}_{q,f}$ as

$$\mathcal{C} = \{c \in [1], \|c\|_1 = \eta\}. \tag{21}$$

We define η to be the smallest integer such that $2^\eta \cdot \binom{d}{\eta} > 2^{256}$, and make sure that $\deg(f) = d$ is sufficiently large enough for this to be possible. The challenge set therefore consists of polynomials in $\mathcal{R}_{q,f}$ that have exactly η non-zero coefficients that are either -1 or 1.

It is worth noting that the challenge set $\mathcal{C}$ is not a group, and by adding two polynomials $c, c' \in \mathcal{C}$ together, we may end up with a new polynomial that is not contained in $\mathcal{C}$. This is the reason we have to make an adjustment to the initial protocol we present in Figure 9, so that we can apply the OR-proof to it. So even though we refer to $\mathcal{C}$ as the challenge set for the Σ-protocol, the verifier will send a challenge sampled from a set of bit strings $\{0,1\}^{256}$, that the prover and verifier then each maps into $\mathcal{C}$ using a publicly known hash function $\mathcal{H}$.

The difference between polynomials in $\mathcal{C}$ are however still of interest, since we must allow such polynomials to satisfy the altered Equation 28 we want to construct our Σ-protocol from. As we will see, doing so is necessary to obtain special soundness. We therefore define

$$\bar{\mathcal{C}} := \{\bar{c} = c - c' \quad \text{with} \quad c \neq c' \in \mathcal{C}\}. \tag{22}$$

5.1 The Learning With Errors Problem

The Learning With Errors (LWE) problem was first proposed by Regev [Reg05], and is one of the fundamental hardness assumptions in lattice-based cryptography. It is a decisional problem that relies on distinguishers

not being able to tell apart two different instances: one where a small error term has been added to a matrix multiplied with a vector, and another where a vector has been directly chosen at random from vectors of corresponding dimensions. The error term added in the first LWE instance makes Gaussian elimination an infeasible strategy for distinguishers trying to decide which instance of the problem it has been given.

Definition 5.1 (The Learning With Errors problem (The LWE problem) [Lyu24])**.** For positive integers $n, m, q, \beta \in \mathbb{Z}^+$, where $\beta < q$, the $\mathsf{LWE}_{n,m,q,\beta}$ problem asks an adversary $\mathcal{A}$ to distinguish between the two following distributions:

0. $(\mathbf{A}, \mathbf{As} + \mathbf{e})$, where $\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n\times m}, \mathbf{s} \leftarrow\!\$\ [\beta]^m, \mathbf{e} \leftarrow\!\$\ [\beta]^n$
1. $(\mathbf{A}, \mathbf{u})$, where $\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n\times m}$ and $\mathbf{u} \leftarrow\!\$\ \mathbb{Z}_q^n$.

The advantage of adversary $\mathcal{A}$ solving the $\mathsf{LWE}_{n,m,q,\beta}$ problem is defined as

$$\begin{aligned}\mathsf{Adv}_{\mathsf{LWE}_{n,m,q,\beta}}(\mathcal{A}) = |\Pr[b = 1|\ \mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n\times m}, \mathbf{u} \leftarrow\!\$\ \mathbb{Z}_q^n, b \leftarrow\!\$\ \mathcal{A}(\mathbf{A}, \mathbf{u})] \\ - \Pr[b = 1|\ \mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n\times m}, \mathbf{s} \leftarrow\!\$\ [\beta]^m, \mathbf{e} \leftarrow\!\$\ [\beta]^n, b \leftarrow\!\$\ \mathcal{A}(\mathbf{A}, \mathbf{As} + \mathbf{e})]|\end{aligned} \tag{23}$$

The LWE problem can be generalized by instead of choosing matrices over the integers, one can choose among matrices over the polynomial ring $\mathcal{R}_{q,f}$. This generalization, referred to as the Module LWE problem, is the version we will focus on the most.

Definition 5.2 (The Module Learning With Errors problem (The MLWE problem) [Lyu24])**.** For positive integers $n, m, q, \beta \in \mathbb{Z}^+$, where $\beta < q$ and ring $\mathcal{R}_{q,f}$, the $\mathcal{R}_{q,f}$-$\mathsf{LWE}_{n,m,\beta}$ problem asks to distinguish between the two following distributions:

0. $(\mathbf{A}, \mathbf{As} + \mathbf{e})$, where $\mathbf{A} \leftarrow\!\$\ \mathcal{R}_{q,f}^{n\times m}, \mathbf{s} \leftarrow\!\$\ [\beta]^m, \mathbf{e} \leftarrow\!\$\ [\beta]^n$
1. $(\mathbf{A}, \mathbf{u})$, where $\mathbf{A} \leftarrow\!\$\ \mathcal{R}_{q,f}^{n\times m}$ and $\mathbf{u} \leftarrow\!\$\ \mathcal{R}_{q,f}^n$.

The advantage of an adversary $\mathcal{A}$ solving the $\mathcal{R}_{q,f}$-$\mathsf{LWE}_{n,m,\beta}$ problem is defined as

$$\begin{aligned}\mathsf{Adv}_{\mathsf{MLWE}_{n,m,\beta}}(\mathcal{A}) = |\Pr[b = 1|\ \mathbf{A} \leftarrow\!\$\ \mathcal{R}_{q,f}^{n\times m}, \mathbf{u} \leftarrow\!\$\ \mathcal{R}_{q,f}^n, b \leftarrow\!\$\ \mathcal{A}(\mathbf{A}, \mathbf{u})] \\ - \Pr[b = 1|\ \mathbf{A} \leftarrow\!\$\ \mathcal{R}_{q,f}^{n\times m}, \mathbf{s} \leftarrow\!\$\ [\beta]^m, \mathbf{e} \leftarrow\!\$\ [\beta]^n, b \leftarrow\!\$\ \mathcal{A}(\mathbf{A}, \mathbf{As} + \mathbf{e})]|\end{aligned} \tag{24}$$

Unless otherwise stated, we let the $\mathsf{MLWE}_{n,m,\beta}$ problem refer to the $\mathcal{R}_{q,f}$-$\mathsf{LWE}_{n,m,\beta}$ problem, where the ring $\mathcal{R}_{q,f}$ is implicitly understood.

5.2 The Short Integer Solution Problem

The Short Integer Solution (SIS) problem was first proposed by Ajtai [Ajt96], and is a computational search problem that relies on the hardness of finding short vectors that satisfies an equation for a given matrix $\mathbf{A}$. That is, no adversary that is given a matrix $\mathbf{A}$ should be able to efficiently compute such a solution.

Definition 5.3 (The Short Integer Solution problem (The SIS problem) [Lyu24])**.** For positive integers $n, m, q, \beta \in \mathbb{Z}_q^+$ where $\beta < q$, the $\mathsf{SIS}_{n,m,q,\beta}$ problem asks to find, for a randomly chosen matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, vectors $\mathbf{s}_1 \in [\beta]^m$ and $\mathbf{s}_2 \in [\beta]^n$ (both not being $\mathbf{0}$), such that $\mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 = \mathbf{0} \mod q$. The advantage of adversary $\mathcal{A}$ solving the $\mathsf{SIS}_{n,m,q,\beta}$ problem is defined as

$$\begin{aligned}\mathsf{Adv}_{\mathsf{MSIS}_{n,m,\beta}}(\mathcal{A}) = &|\Pr[(\mathbf{s}_1, \mathbf{s}_2) \leftarrow\$ \mathcal{A}(\mathbf{A})| \ \mathbf{s}_1 \in [\beta]^m, \mathbf{s}_2 \in [\beta]^n, \\ &\mathbf{A} \leftarrow\$ \mathbb{Z}_q^{n \times m}, (\mathbf{s}_1, \mathbf{s}_2) \neq 0, \ \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 = \mathbf{0} \mod q]|\end{aligned} \tag{25}$$

■

Here as well, The SIS problem can be generalized by instead of choosing matrices over the integers, one can choose among matrices over the polynomial ring $\mathcal{R}_{q,f}$. This generalization, referred to as the Module SIS problem, is the version we will focus on the most.

Definition 5.4 (The Module Short Integer Solution problem (The MSIS problem) [Lyu24])**.** For positive integers $n, m, q, \beta \in \mathbb{Z}^+$ where $\beta < q$ and ring $\mathcal{R}_{q,f}$, the $\mathcal{R}_{q,f}$-$\mathsf{SIS}_{n,m,\beta}$ problem asks to find, for a randomly chosen matrix $\mathbf{A} \in \mathcal{R}_{q,f}^{n \times m}$, vectors $\mathbf{s}_1 \in [\beta]^m$ and $\mathbf{s}_2 \in [\beta]^n$ (both not being $\mathbf{0}$), such that $\mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 = \mathbf{0} \mod q$. The advantage of an adversary $\mathcal{A}$ solving the $\mathcal{R}_{q,f}$-$\mathsf{SIS}_{n,m,\beta}$ problem is defined as

$$\begin{aligned}\mathsf{Adv}_{\mathsf{MSIS}_{n,m,\beta}}(\mathcal{A}) = &|\Pr[(\mathbf{s}_1, \mathbf{s}_2) \leftarrow\$ \mathcal{A}(\mathbf{A})| \ \mathbf{s}_1 \in [\beta]^m, \mathbf{s}_2 \in [\beta]^n, \\ &\mathbf{A} \leftarrow\$ \mathcal{R}_{q,f}^{n \times m}, (\mathbf{s}_1, \mathbf{s}_2) \neq 0, \ \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 = \mathbf{0} \mod q]|\end{aligned} \tag{26}$$

■

Unless otherwise stated, we let the $\mathsf{MSIS}_{n,m,\beta}$ problem refer to the $\mathcal{R}_{q,f}$-$\mathsf{SIS}_{n,m,\beta}$ problem, where the ring $\mathcal{R}_{q,f}$ is implicitly understood.

5.3 Choice of Parameters

The following gives an intuition for how the parameters in the LWE problem and the SIS problem relate to the hardness of solving them. For the LWE problem, a larger norm bound β for the noise $\mathbf{e}$ adds more randomness to the public matrix multiplied with the secret vector, and therefore makes the problem harder to solve. For SIS, allowing solutions of larger norm β makes

the problem easier, since there can be more solutions that satisfies a given public matrix $\mathbf{A}$. Note that this is for a given modulus q and dimension n.

One of the difficulties when choosing parameters is that we often want both the LWE problem and the SIS problem to be hard simultaneously, but they have some opposite requirements on the parameters to offer the best security. If the LWE problem is sufficiently hard but the SIS problem is not, a solution could be to increase the modulus q. Conversely, if we have high SIS security but low LWE security, we can increase the norm bound β. Note that the SIS problem cannot be made harder by increasing m, for if we would find a solution to the SIS problem with all the other variables fixed, but for $m' < m$, we could extend the solution to m as well by letting the appropriate $m - m'$ remaining variables in the solution equal 0. For both assumptions, increasing the dimension n makes the problem harder, but a downside with this is that everything we work with then gets larger. If we would want to build a Σ-protocol from these assumptions, increasing the dimension would increase the proof size.

We also note that there is a difference between theoretically and practically chosen parameters for the hardness of the underlying assumptions. When doing a practical analysis for LWE, there are certain known attacks we have to check the system for, and then set the parameters such that these attacks become infeasible [APS15]. For SIS, we have a formula to choose the best parameters [MR09].

6 Σ-Protocols from Lattice Assumptions

Say that for an $\mathsf{MLWE}_{n,m,\beta}$ instance $(\mathbf{A}, \mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2)$, a prover $\mathcal{P}$ wants to prove that they have values $\mathbf{s}_1 \in [\beta]^m, \mathbf{s}_2 \in [\beta]^n$ that satisfies the equation for $\mathbf{t}$, while also being in the appropriate range. We can define a relation for these requirements as

$$\begin{aligned}\mathcal{R}_{\mathsf{SIS}} := \Big\{ &(X = (\mathbf{A} \in \mathcal{R}_{q,f}^{n\times m}, \mathbf{t} \in \mathcal{R}_{q,f}^{n}),\ w = (\mathbf{s}_1 \in [\beta]^m, \mathbf{s}_2 \in [\beta]^n)) \\ &: \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 = \mathbf{t} \Big\},\end{aligned} \tag{27}$$

where we refer to it as a relation for SIS, since the witness is a solution to the $\mathsf{MSIS}_{n,m,\beta}$ problem for matrix $\mathbf{A}$. As mentioned in Section 1, constructing a Σ-protocol directly can lead to inefficient protocols. The proof can be optimized by instead proving knowledge of $\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2$ with coefficients from a somewhat larger set $[2\bar{\beta}]$, that satisfy the altered equation

$$\mathbf{A}\bar{\mathbf{s}}_1 + \bar{\mathbf{s}}_2 = \bar{c}\mathbf{t}. \tag{28}$$

Here, $\bar{c}$ is either 1 or an element of $\bar{\mathcal{C}}$ as defined in Equation 22. The reason we allow the coefficients of $(\mathbf{s}_1, \mathbf{s}_2)$ to be in $[2\bar{\beta}]$, is that this will allow us to construct an extractor for special soundness that outputs a somewhat larger witness that still satisfies the equation.

As we will see in Lemma 6.2, solving Equation 28 with $\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c}$ as mentioned is equivalent to solving either the MLWE or the MSIS problem for matrix $\mathbf{A}$ [Lyu24, Section 5.1]. For this reason, constructing Σ-protocols where we prove possession of polynomials with slightly larger coefficients that satisfies the altered Equation 28 are still of interest.

6.1 Σ-Protocol from MSIS

In order to construct a Σ-protocol for the altered Equation 28, we define the altered relation

$$\begin{aligned}\bar{\mathcal{R}}_{\mathsf{SIS}} := \Big\{ &(X = (\mathbf{A} \in \mathcal{R}_{q,f}^{n\times m}, \mathbf{t} \in \mathcal{R}_{q,f}^{n}),\ w = (\bar{\mathbf{s}}_1 \in [2\bar{\beta}]^m, \bar{\mathbf{s}}_2 \in [2\bar{\beta}]^n)) \\ &: \mathbf{A}\bar{\mathbf{s}}_1 + \bar{\mathbf{s}}_2 = \bar{c}\mathbf{t},\ (\bar{c} \in \bar{\mathcal{C}} \ \vee \ \bar{c} = 1) \Big\}.\end{aligned} \tag{29}$$

We present an interactive protocol Π_{SIS} for relation $\bar{\mathcal{R}}_{\mathsf{SIS}}$ in Figure 9. There are multiple changes that can be made to protocol Π_{SIS}, depending on the desired use case. One of these is hashing the first message sent by the prover, and this change is shown in the white boxes.

Hashing Bit Strings into the Set $\mathcal{C}$ In protocol Π_{SIS}, a verifier instead of sending an element of $\mathcal{C}$ to the prover sends a bit string that the prover and verifier each maps into the challenge set $\mathcal{C}$. Mapping bit strings into $\mathcal{C}$ can be done by employing the hash function $\mathsf{HashInBall}$ described in the NIST Standard [Nat24, Section 7.3, Algorithm 29], which we will denote by $\mathcal{H}$. Since the hash function is collision-resistant, there is a negligible probability of collisions occurring, meaning that we have a one-to-one correspondence between bit strings from $\{0,1\}^{256}$ and elements of $\mathcal{C}$. Doing XOR operations on the bit strings before applying the hash function $\mathcal{H}$, will therefore allow us to apply the OR-construction to the protocol.

$\mathsf{Input}(\mathbf{A} \in \mathcal{R}_{q,f}^{n \times m}, \mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2 \in \mathcal{R}_{q,f}^{n};\ \mathbf{s}_1 \in [\beta]^m, \mathbf{s}_2 \in [\beta]^n)$

Interaction in Π_{SIS}, $\boxed{\Pi_{\mathsf{SIS}}}$ for Relation $\bar{\mathcal{R}}_{\mathsf{SIS}}$:

Prover $\mathcal{P}$		**Verifier $\mathcal{V}$**
$\mathbf{y}_1 \leftarrow\!\$\ [\gamma + \bar{\beta}]^m$		
$\mathbf{y}_2 \leftarrow\!\$\ [\gamma + \bar{\beta}]^n$		
$\mathbf{w} := \mathbf{A}\mathbf{y}_1 + \mathbf{y}_2$		
	$\mathbf{w},\ \boxed{H(\mathbf{w})} \longrightarrow$	
		$s \leftarrow\!\$\ \{0,1\}^{256}$
	$\longleftarrow s$	
$c := \mathcal{H}(s) \in \mathcal{C}$		
$\mathbf{z}_1 := c\mathbf{s}_1 + \mathbf{y}_1$		
$\mathbf{z}_2 := c\mathbf{s}_2 + \mathbf{y}_2$		
if $\mathbf{z}_1 \notin [\bar{\beta}]^m$ or $\mathbf{z}_2 \notin [\bar{\beta}]^n$:		
$(\mathbf{z}_1, \mathbf{z}_2) := \perp$		
	$(\mathbf{z}_1, \mathbf{z}_2) \longrightarrow$	
		$\mathcal{V}$ Computes $c := \mathcal{H}(s)$
		and accepts iff
		$(\mathbf{z}_1, \mathbf{z}_2) \neq \perp$
		$\wedge\ \mathbf{z}_1 \in [\bar{\beta}]^m \wedge\ \mathbf{z}_2 \in [\bar{\beta}]^n$
		$\wedge\ (\mathbf{w} = \mathbf{A}\mathbf{z}_1 + \mathbf{z}_2 - c\mathbf{t}$
		$\boxed{\vee\ H(\mathbf{w}) = H(\mathbf{A}\mathbf{z}_1 + \mathbf{z}_2 - c\mathbf{t})}\ )$

Figure 9: Schnorr-Like Σ-Protocol from Lattice Assumptions

Hashing the First Message Sent by the Prover In protocol $\boxed{\Pi_{\mathsf{SIS}}}$, instead of sending the commit value $\mathbf{w}$ in clear, we apply a hash function $H : \mathcal{R}_{q,f}^n \to \{0,1\}^*$ to it before sending it. The reason we do so, is that Π_{SIS} as an interactive protocol is not a Σ-protocol, as we are not guaranteed that simulated transcripts with aborts have the same probability distribution as honestly generated transcripts. The abort step in the protocol is dependent on the secret values $\mathbf{s}_1$ and $\mathbf{s}_2$, and may therefore leak information about the witness. By instead hashing the commit value $\mathbf{w}$, a verifier will not learn anything about the values $\mathbf{y}_1, \mathbf{y}_2$ used to compute it, and we can argue that $H(\mathbf{w})$ is uniformly distributed when proving Special HVZK in the ROM.

To prove that $\boxed{\Pi_{\mathsf{SIS}}}$ is a Σ-protocol, the following Lemma 6.1 grants us the necessary probabilities for proving that the protocol is Special HVZK.

Lemma 6.1 ([Lyu24]). *If $\gamma \in \mathbb{Z}^+$ is such that for all polynomials $\mathbf{s} \in [\beta]$, $c \in \mathcal{C}$ we have that $c\mathbf{s} \in [\gamma]$, then for all $\mathbf{s}_1, \mathbf{s}_2, c$ as in the protocol in Figure 9, we have*

$$\Pr_{\mathbf{y}_1,\mathbf{y}_2}[(\mathbf{z}_1, \mathbf{z}_2) \neq \perp] = \left(\frac{2\bar{\beta}+1}{2(\gamma+\bar{\beta})+1}\right)^{d(m+n)}, \tag{30}$$

and for all $\mathbf{z}_1' \in [\bar{\beta}]^m, \mathbf{z}_2' \in [\bar{\beta}]^n$, we have

$$\Pr_{\mathbf{y}_1,\mathbf{y}_2}[(\mathbf{z}_1, \mathbf{z}_2) = (\mathbf{z}_1', \mathbf{z}_2') | \ (\mathbf{z}_1, \mathbf{z}_2) \neq \perp] = \left(\frac{1}{2\bar{\beta}+1}\right)^{d(m+n)}. \tag{31}$$

Proof. Suppose that $\mathbf{z}_1 = c\mathbf{s}_1 + \mathbf{y}_1$ and $\mathbf{z}_2 = c\mathbf{s}_2 + \mathbf{y}_2$ for some $\mathbf{y}_1, \mathbf{y}_2 \in [\gamma + \bar{\beta}]$. Using vector notation, this can jointly be written as

$$\mathbf{z} := \begin{bmatrix}\mathbf{z}_1 \\ \mathbf{z}_2\end{bmatrix} = \begin{bmatrix}c\mathbf{s}_1 \\ c\mathbf{s}_2\end{bmatrix} + \begin{bmatrix}\mathbf{y}_1 \\ \mathbf{y}_2\end{bmatrix} \in \mathcal{R}_{q,f}^{n+m}, \tag{32}$$

since $\mathbf{z}_1 \in \mathcal{R}_{q,f}^m$ and $\mathbf{z}_2 \in \mathcal{R}_{q,f}^n$. Using the notation introduced in Section 2.1, we can also express $\mathbf{z}$ by the polynomial coefficients of $\mathbf{z}_1$ and $\mathbf{z}_2$ in an integer vector as

$$\mathcal{V}_{\mathbf{z}} := \begin{bmatrix}\mathcal{V}_{\mathbf{z}_1} \\ \mathcal{V}_{\mathbf{z}_2}\end{bmatrix} = \begin{bmatrix}\mathcal{V}_{c\mathbf{s}_1} \\ \mathcal{V}_{c\mathbf{s}_2}\end{bmatrix} + \begin{bmatrix}\mathcal{V}_{\mathbf{y}_1} \\ \mathcal{V}_{\mathbf{y}_2}\end{bmatrix} \in \mathbb{Z}_q^{d(n+m)}.$$

We will begin by obtaining Equation 30, which gives us the probability that the prover $\mathcal{P}$ does not abort. For $(\mathbf{z}_1, \mathbf{z}_2) \neq \perp$ to hold, we need that each coefficient $\nu_{\mathbf{z}} \in \mathcal{V}_{\mathbf{z}}$ satisfies $\nu_{\mathbf{z}} \in [\bar{\beta}]$, where $|[\bar{\beta}]| = 2\bar{\beta}+1$. This gives us the number of favorable outcomes for each of the $d(n+m)$ coefficients in $\mathcal{V}_{\mathbf{z}}$.

We are now interested in the number of all possible $(\mathbf{z}_1, \mathbf{z}_2)$ when we let $\mathbf{y}_1, \mathbf{y}_2$ vary. For any coefficient $\nu_{c\mathbf{s}}$ in $\begin{bmatrix}\mathcal{V}_{c\mathbf{s}_1} \\ \mathcal{V}_{c\mathbf{s}_2}\end{bmatrix}$ and $\nu_{\mathbf{y}}$ in $\begin{bmatrix}\mathcal{V}_{\mathbf{y}_1} \\ \mathcal{V}_{\mathbf{y}_2}\end{bmatrix}$, we have that $\nu_{c\mathbf{s}} \in [\gamma]$ by assumption and $\nu_{\mathbf{y}} \in [\gamma + \bar{\beta}]$ by how the values $\mathbf{y}_1, \mathbf{y}_2$ are sampled

in the protocol. Any coefficient $\nu_{\mathbf{z}} \in \mathcal{V}_{\mathbf{z}}$ is computed as $\nu_{\mathbf{z}} = \nu_{cs} + \nu_{\mathbf{y}}$, and must therefore be an element of $\{\nu_{cs} + [\gamma + \bar{\beta}]\}$. This set is of the same size as $[\gamma + \bar{\beta}]$, namely $2(\gamma + \bar{\beta}) + 1$, and gives us the number of possible outcomes for each of the $d(n+m)$ coefficients in $\mathcal{V}_{\mathbf{z}}$. Note also that $[\bar{\beta}] \subseteq \{\nu_{cs} + [\gamma + \bar{\beta}]\}$ as $\nu_{cs} \in [\gamma]$, which guarantees that all the favorable outcomes are contained in the possible ones.

Putting this together, we obtain the desired expression in Equation 30 from

$$\Pr_{\mathbf{y}_1,\mathbf{y}_2}[(\mathbf{z}_1, \mathbf{z}_2) \neq \perp] = \prod_{i=1}^{d(n+m)} \left(\frac{2\bar{\beta}+1}{2(\gamma+\bar{\beta})+1}\right) = \left(\frac{2\bar{\beta}+1}{2(\gamma+\bar{\beta})+1}\right)^{d(m+n)}.$$

For Equation 31, suppose that $\mathbf{z}'_1 \in [\bar{\beta}]^m$ and $\mathbf{z}'_2 \in [\bar{\beta}]^n$, and let $\mathbf{z}' := \begin{bmatrix}\mathbf{z}'_1 \\ \mathbf{z}'_2\end{bmatrix}$. We want to calculate the probability that a non-aborting $\mathbf{z}$ defined as in Equation 32 equals $\mathbf{z}'$, when we let $\mathbf{y}_1, \mathbf{y}_2$ vary.

We begin by proving the claim that

$$\forall \mathbf{z}'_1 \in [\bar{\beta}]^m, \mathbf{z}'_2 \in [\bar{\beta}]^n, \Pr_{\mathbf{y}_1,\mathbf{y}_2}[(\mathbf{z}_1, \mathbf{z}_2) = (\mathbf{z}'_1, \mathbf{z}'_2)] = \left(\frac{1}{2(\gamma+\bar{\beta})+1}\right)^{d(m+n)}. \quad (33)$$

To show this, let $\nu_{\mathbf{z}'} \in \mathcal{V}_{\mathbf{z}'}$ be the ith coefficient in $\mathbf{z}'$, and define $\nu_{cs}, \nu_{\mathbf{z}}$ and $\nu_{\mathbf{y}}$ similarly. For $\nu_{\mathbf{z}} = \nu_{\mathbf{z}'}$ to hold, we must have that $\nu_{\mathbf{y}} = \nu_{\mathbf{z}'} - \nu_{cs} \in [\gamma + \bar{\beta}]$. Since $\nu_{\mathbf{y}}$ is chosen uniformly at random from the same set $[\gamma + \bar{\beta}]$, the probability that it equals the specific value $\nu_{\mathbf{z}'} - \nu_{cs}$ is

$$\frac{1}{2(\gamma+\bar{\beta})+1}.$$

This proves Equation 33, as there are $d(m+n)$ coefficients in $\mathbf{z}$ this applies to.

We also have that

$$\forall \mathbf{z}'_1 \in [\bar{\beta}]^m, \mathbf{z}'_2 \in [\bar{\beta}]^n, \Pr_{\mathbf{y}_1,\mathbf{y}_2}[(\mathbf{z}_1, \mathbf{z}_2) \neq \perp | (\mathbf{z}_1, \mathbf{z}_2) = (\mathbf{z}'_1, \mathbf{z}'_2)] = 1, \quad (34)$$

since all the coefficients in $\mathbf{z}$ are in $[\bar{\beta}]$ when they equal coefficients in $\mathbf{z}'$ that by assumption are chosen from this set. Using Bayes' theorem, we obtain the desired Equation 31 from (31) = (34) · (33)/(30). □

The first part of the Lemma tells us the probability of obtaining aborting transcripts when running protocol Π_{SIS}. This tells us how to choose $\bar{\beta}$ to obtain a favorable number of protocol repetitions for obtaining an accepting transcript. It also tells us how often a simulator for the protocol should abort, in order to get the correct probability distribution for the simulated transcripts. We also note that one can, using the same arguments as above, prove that Lemma 6.1 also applies to the adjusted protocol $\boxed{\Pi_{\mathsf{SIS}}}$.

The second part of the Lemma states that non-aborting $(\mathbf{z}_1, \mathbf{z}_2)$ have coefficients with the same probability distribution as the ones in $\mathbf{y}_1, \mathbf{y}_2$ chosen as in the real protocol, namely the uniform distribution. This will be helpful when proving that the protocol satisfies Special HVZK, since we then want to choose the $\mathbf{z}$-values first and then use them to compute $\mathbf{w}$, while still arguing that the distribution of real and simulated transcripts are indistinguishable.

We now proceed with proving that the protocol $\boxed{\Pi_{\mathsf{SIS}}}$ is a Σ-protocol.

Completeness Since the protocol has a rejection sampling step, the protocol will not have perfect completeness, meaning that the protocol does not produce an accepting transcript for every honest execution of the protocol between a prover and a verifier. The rejection sampling step demands that the coefficients of $\mathbf{z}_1$ and $\mathbf{z}_2$ are in the range $[\bar{\beta}]$, and therefore this value affects the completeness of the protocol. We have that the probability of the prover not sending $\perp$ is approximately

$$\left(\frac{2\bar{\beta}+1}{2(\gamma+\bar{\beta})+1}\right)^{d(m+n)} > \left(\frac{\bar{\beta}}{\bar{\beta}+\gamma}\right)^{d(m+n)} = \left(1+\frac{\gamma}{\bar{\beta}}\right)^{-d(m+n)} \approx e^{-\gamma d(m+n)/\bar{\beta}},$$

so setting $\bar{\beta} = \gamma d(m+n)$ gives us a protocol that produces an accepting transcript after an expected number of e repetitions. The probability of getting an accepting transcript when doing one honest execution of the protocol is therefore $1/e$. Note that the expected number of repetitions can be lowered by choosing a larger $\bar{\beta}$, but doing so will however makes the protocol less efficient.

Special Soundness In order to prove that the protocol satisfies special soundness, we make use of the following Lemma.

Lemma 6.2 ([Lyu24])**.** *Suppose that an extractor $\mathcal{E}$ upon input $(\mathbf{A} \in \mathcal{R}_{q,f}^{n\times m}, \mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2)$ for $\mathbf{s}_1 \in [\beta], \mathbf{s}_2 \in [\beta]^n$ is able to produce values $\bar{\mathbf{s}}_1 \in [2\bar{\beta}]^m$, $\bar{\mathbf{s}}_2 \in [2\bar{\beta}]^n$ and $\bar{c} \in \bar{\mathcal{C}}$ such that the equation $\mathbf{A}\bar{\mathbf{s}}_1 + \bar{\mathbf{s}}_2 = \bar{c}\mathbf{t}$ is satisfied. Then it either breaks the $\mathsf{MLWE}_{n,m,2\bar{\beta}}$ or the $\mathsf{MSIS}_{n,m+1,2\bar{\beta}}$ problem.*

Proof. We prove the Lemma using a hybrid argument, where we have the following sequence of games.

Game G_0: The first game corresponds to the real version of the extraction procedure by $\mathcal{E}$. That is, we have some algorithm $\mathcal{B}$ that sends input $(\mathbf{A} \in \mathcal{R}_{q,f}^{n\times m}, \mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2)$ to an extractor $\mathcal{E}$, which tries to produce values $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c})$ that satisfy the winning condition. If we let $\mathsf{Adv}_{\mathrm{SS}}(\mathcal{E})$ denote the probability that extractor $\mathcal{E}$ successfully produces such values, we have that $\mathsf{Adv}_{\mathrm{SS}}(\mathcal{E}) = \Pr[\mathbf{G}_0]$.

Game G_1: In the next game, $\mathcal{B}$ changes the value it sends to extractor $\mathcal{E}$ to a matrix chosen uniformly at random. That is, it instead sends $\bar{\mathbf{A}} := [\mathbf{A}|\mathbf{t}] \leftarrow\$ \mathcal{R}_{q,f}^{n\times m+1}$ to extractor $\mathcal{E}$. By the $\mathsf{MLWE}_{n,m,2\bar{\beta}}$ assumption, it is computationally infeasible for an adversary $\mathcal{C}$ to distinguish the two games $\mathbf{G}_0$ and $\mathbf{G}_1$, since if there existed an algorithm distinguishing them with a non-negligible probability, $\mathcal{C}$ could run it as a subroutine and break the $\mathsf{MLWE}_{n,m,2\bar{\beta}}$ problem.

Furthermore, if extractor $\mathcal{E}$ is able to produce the correct values $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c})$, algorithm $\mathcal{B}$ can produce a solution to the $\mathsf{MSIS}_{n,m+1,2\bar{\beta}}$ problem for matrix $\bar{\mathbf{A}}$, namely $([\bar{\mathbf{s}}_1| - \bar{c}]^\top, \bar{\mathbf{s}}_2)$ such that $\bar{\mathbf{A}}\,[\bar{\mathbf{s}}_1| - \bar{c}]^\top + \bar{\mathbf{s}}_2 = \mathbf{A}\bar{\mathbf{s}}_1 - \bar{c}\mathbf{t} + \bar{\mathbf{s}}_2 = 0$.

Putting this together, we obtain that

$$\begin{aligned}\mathsf{Adv}_{\mathrm{SS}}(\mathcal{E}) &= \Pr[\mathbf{G}_0] \\ &\leq |\Pr[\mathbf{G}_0] - \Pr[\mathbf{G}_1]| + \Pr[\mathbf{G}_1] \\ &= \mathsf{Adv}_{\mathsf{MLWE}_{n,m,2\bar{\beta}}}(\mathcal{C}) + \mathsf{Adv}_{\mathsf{MSIS}_{n,m+1,2\bar{\beta}}}(\mathcal{B}).\end{aligned}$$

□

We now construct an extractor $\mathcal{E}$ for special soundness, given in Figure 10.

Extractor $\mathcal{E}(X, \pi, \pi')$:

1 : Upon input $X = (\mathbf{A} \in \mathcal{R}_{q,f}^{n\times m}, \mathbf{t} \in \mathcal{R}_{q,f}^{n})$ and two accepting transcripts $\pi = (\mathbf{w}, s, (\mathbf{z}_1, \mathbf{z}_2)), \pi' = (\mathbf{w}, s', (\mathbf{z}_1', \mathbf{z}_2'))$, compute $c = \mathcal{H}(s)$ and $c' = \mathcal{H}(s')$.

2 : Since π and π' are accepting transcripts, we have that $H(\mathbf{w}) = H(\mathbf{A}\mathbf{z}_1 + \mathbf{z}_2 - \mathbf{t}c)$ and $H(\mathbf{w}) = H(\mathbf{A}\mathbf{z}_1' + \mathbf{z}_2' - \mathbf{t}c')$.

3 : Because the hash function H is collision-resistant, $\mathbf{w} = \mathbf{A}\mathbf{z}_1 + \mathbf{z}_2 - \mathbf{t}c$ and $\mathbf{w} = \mathbf{A}\mathbf{z}_1' + \mathbf{z}_2' - \mathbf{t}c'$ with overwhelming probability.

4 : This implies that $\mathbf{A}\mathbf{z}_1 + \mathbf{z}_2 - \mathbf{t}c = \mathbf{A}\mathbf{z}_1' + \mathbf{z}_2' - \mathbf{t}c'$, which is equivalent to $\mathbf{A}(\mathbf{z}_1 - \mathbf{z}_1') + (\mathbf{z}_2 - \mathbf{z}_2') - \mathbf{t}(c - c') = 0$.

5 : We let $\bar{\mathbf{s}}_1 := (\mathbf{z}_1 - \mathbf{z}_1') \in [2\bar{\beta}]^m, \bar{\mathbf{s}}_2 := (\mathbf{z}_2 - \mathbf{z}_2') \in [2\bar{\beta}]^n$ and $\bar{c} := (c - c') \in \bar{\mathcal{C}}$.

6 : **return** $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c})$

Figure 10: Extractor for Special Soundness of Protocol $\boxed{\Pi_{\mathsf{SIS}}}$

The extractor $\mathcal{E}$ returns a triple $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c})$ with $\bar{\mathbf{s}}_1 \in [2\bar{\beta}]^m, \bar{\mathbf{s}}_2 \in [2\bar{\beta}]^n$ and $c \in \bar{\mathcal{C}}$, which by Lemma 6.2 either breaks the $\mathsf{MLWE}_{n,m,2\bar{\beta}}$ or the $\mathsf{MSIS}_{n,m+1,2\bar{\beta}}$ problem. Note that by how we defined the altered relation $\bar{\mathcal{R}}_{\mathsf{SIS}}$, the triple $(\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \bar{c})$ is a valid witness for common input X, even though the coefficients are from a larger set than what was initially allowed in relation $\mathcal{R}_{\mathsf{SIS}}$. We therefore in this protocol have a soundness gap between what we can prove knowledge of, and what $\mathcal{E}$ can extract.

Special HVZK To show that $\boxed{\Pi_{\mathsf{SIS}}}$ is Special HVZK, we construct a simulator $\mathcal{S}$, given in Figure 11.

Simulator $\mathcal{S}(X = (\mathbf{A}, \mathbf{t}), s \in \{0,1\}^{256})$:

1 : Flip a coin s.t. $\Pr[coin = 1] = \alpha := \left(\frac{2\bar{\beta}+1}{2(\gamma+\bar{\beta})+1}\right)^{d(m+n)}$ (30)
2 : **if** $coin = 1$: // non-aborting transcript
3 : $\quad \mathbf{z}_1 \leftarrow\$ [\bar{\beta}]^m, \mathbf{z}_2 \leftarrow\$ [\bar{\beta}]^n$
4 : $\quad c = \mathcal{H}(s) \in \mathcal{C}$
5 : $\quad \mathbf{w} := \mathbf{A}\mathbf{z}_1 + \mathbf{z}_2 - \mathbf{t}c$
6 : $\quad$ **return** $(H(\mathbf{w}), s, (\mathbf{z}_1, \mathbf{z}_2))$
7 : **else** : // aborting transcript
8 : $\quad h \leftarrow\$ \{0,1\}^*$ // sampled from the codomain of H
9 : $\quad$ **return** $(h, s, \perp)$

Figure 11: Simulator for Special HVZK of Protocol $\boxed{\Pi_{\mathsf{SIS}}}$

We proceed by arguing that transcripts produced by $\mathcal{S}$ are of the same distribution as those that are produced by an honest protocol run. We first note that $\mathcal{S}$ is a PPT algorithm, since sampling integers from $[\bar{\beta}]$ and performing ring operations are polynomial time operations. We split the analysis of the probability distributions into the two following cases.

- **Non-aborting transcripts:** In the real protocol, the values $\mathbf{z}_1$ and $\mathbf{z}_2$ will be uniformly distributed, since $\mathbf{y}_1$ and $\mathbf{y}_2$ are uniformly distributed. By Lemma 6.1, the $\mathbf{z}$-values produced by $\mathcal{S}$ will also be uniformly distributed, since $coin = 1$ is the equivalent condition as the one the probability is conditioned on in the Lemma. We have computed $\mathbf{w}$ to satisfy the exact equation needed for the transcript to be accepting, and $H(\mathbf{w})$ will therefore also be distributed correctly as well. Real transcripts and simulated transcripts with no aborts therefore have the same probability distribution.

- **Aborting transcripts:** For transcripts where we abort, h is uniformly distributed and therefore indistinguishable from a real execution of the protocol, when assuming that the output of the hash function H looks random. Therefore simulated transcripts with aborts and real ones have the same distribution.

The coin flip done in step 1 ensures that the expected number of simulated transcripts that abort and simulated transcripts that do not is the same as in the real protocol, as the coin equals 1 with the probability of not aborting as computed in Equation 30.

Putting all of this together, we have now shown that $\boxed{\Pi_{\mathsf{SIS}}}$ is a Σ-protocol.

6.2 Instantiating the OR-Construction

We are now ready to apply the OR-construction the protocol $\boxed{\Pi_{\mathsf{SIS}}}$. As we have seen, it is a Σ-protocol that has a group operation on the set of bit strings $\{0,1\}^{256}$ that the verifier chooses a challenge from. By using the hash function $\mathcal{H}$, we have a one-to-one correspondence to elements of $\mathcal{C}$, and can apply the OR-construction to it. We define the relation

$$\bar{\mathcal{R}}_{\mathsf{SIS\text{-}OR}} := \{((X, X'), w) \mid (X, w) \in \bar{\mathcal{R}}_{\mathsf{SIS}} \ \vee \ (X', w) \in \bar{\mathcal{R}}_{\mathsf{SIS}}\},$$

and assume without loss of generality that $(X, w) \in \mathcal{R}$. The resulting protocol $\Pi_{\text{SIS-OR}}$ is presented in Figure 12.

$\mathsf{Input}((\mathbf{A}, \mathbf{t}), (\mathbf{A}', \mathbf{t}') \in \mathcal{R}_{q,f}^{n \times m} \times \mathcal{R}_{q,f}^{n};\ \mathbf{s}_1 \in [\beta]^m, \mathbf{s}_2 \in [\beta]^n)$

Interaction in $\Pi_{\text{SIS-OR}}$ for Relation $\bar{\mathcal{R}}_{\mathsf{SIS\text{-}OR}}$:

Prover $\mathcal{P}$		**Verifier** $\mathcal{V}$
Sample $e' \leftarrow\!\!\$\ \{0,1\}^{256}$		
Run $(\mathbf{w}', (\mathbf{z}_1', \mathbf{z}_2')) \leftarrow\!\!\$\ \mathcal{S}(X', e')$		
$\mathbf{z}' := (\mathbf{z}_1', \mathbf{z}_2')$		
$\mathbf{y}_1 \leftarrow\!\!\$\ [\gamma + \bar{\beta}]^m$, $\mathbf{y}_2 \leftarrow\!\!\$\ [\gamma + \bar{\beta}]^n$		
$\mathbf{w} := \mathbf{A}\mathbf{y}_1 + \mathbf{y}_2$		
	$\xrightarrow{\mathbf{w},\ \mathbf{w}'}$	
		$s \leftarrow\!\!\$\ \{0,1\}^{256}$
	$\xleftarrow{s}$	
Define $e := s \oplus e'$		
Compute $c := \mathcal{H}(e) \in \mathcal{C}$		
$\mathbf{z}_1 := c\mathbf{s}_1 + \mathbf{y}_1$, $\mathbf{z}_2 := c\mathbf{s}_1 + \mathbf{y}_2$		
if $\mathbf{z}_1 \notin [\bar{\beta}]^m$ or $\mathbf{z}_2 \notin [\bar{\beta}]^n$:		
$(\mathbf{z}_1, \mathbf{z}_2) := \perp$		
$\mathbf{z} := (\mathbf{z}_1, \mathbf{z}_2)$		
	$\xrightarrow{\mathbf{z},\ \mathbf{z}',\ e,\ e'}$	
		$\mathcal{V}$ computes $c = \mathcal{H}(e)$
		and $c' = \mathcal{H}(e')$,
		and accepts if
		$e \oplus e' = s$
		$\wedge\ \mathbf{z}_1, \mathbf{z}_1' \in [\bar{\beta}]^m, \wedge\ \mathbf{z}_2, \mathbf{z}_2' \in [\bar{\beta}]^n$
		$\wedge\ \mathbf{A}\mathbf{z}_1 + \mathbf{z}_2 - c\mathbf{t} = \mathbf{w}$
		$\wedge\ \mathbf{A}'\mathbf{z}_1' + \mathbf{z}_2' - c'\mathbf{t}' = \mathbf{w}'$

Figure 12: OR-Proof for Schnorr-Like Σ-Protocol from Lattice Assumptions

7 Compressed Σ-Protocols from Lattices

The following section is based on "A General Framework for Compressed Σ-Protocols over Lattices" [ACK21, Section 5]. Applying the compression mechanism introduced in Section 4 to lattice-based Σ-protocols entails constructing a new framework that is compatible with different underlying assumptions. Ideally, we want a compression mechanism that can be composed recursively, in such a way that we reduce communication costs from linear to roughly logarithmic in the size of the witness. As we will see in Section 7.4, we are able to achieve poly-logarithmic communication costs when using a compact lattice-based commitment scheme (Definition 7.9).

The compression mechanism presented in Figure 7 is designed for the discrete log setting, and makes use of the fact that we build it from the Pedersen vector commitment scheme (Definition 4.1) when recursively redefining the generators in the public parameter to be halved for each iteration of the protocol. However when constructing Σ-protocols from a different underlying assumptions, we do not necessarily have generators in the common input that behave similarly to what we had in the dlog setting. Instead, we want to make use of what we will refer to as an extractable compression function (Definition 7.6), which is an abstraction that will allow us to halve the size of the response message sent by the prover, doing so while being compatible with the lattice setting. In addition to halve the size of the response message for each recursive call, we also want to be able to bound the norm of the response sent by the prover.

The setting is as follows. For some $\mathcal{R}$-modules M with norm $\|\cdot\|$ and N and $\mathcal{R}$-module homomorphism $\Psi : M \to N$, a prover want to convince the verifier that for a publicly known value $Y \in N$, it knows a value y of small norm in its preimage such that $\Psi(y) = Y$. This is captured by the relation

$$\mathcal{R}(\Psi, \alpha) = \{(Y \in N;\ y \in M) : Y = \Psi(y), \|y\| \leq \alpha\}\,. \tag{35}$$

Note that this is a generalization of how we defined the relation $\mathcal{R}_{\mathsf{com}}$ in Equation 12 for which we constructed the dlog compression mechanism, where we instead showed that a witness we commit to satisfies a linear constraint. If we let

$$\Psi : \mathbb{Z}_q^n \times \mathbb{Z}_q \to \mathbb{G} \times \mathbb{Z}_q, \quad (\mathbf{x}, \gamma) \mapsto \big(\mathsf{Com}_{(\mathbf{g},h)}(\mathbf{x}, \gamma), L(\mathbf{x})\big) \tag{36}$$

for $L \in \mathcal{L}(\mathbb{Z}_q^n)$, $y \in \mathbb{Z}_q$ and $Y \in \mathbb{G}$ where $\mathbb{G}$ is of order q, the relations $\mathcal{R}(\Psi, \alpha)$ and $\mathcal{R}_{\mathsf{com}}$ coincide given that we omit the norm bound.

One of the main difficulties when constructing lattice-based Σ-protocols, is that the witness is required to be small. As we saw in Figure 9 for protocol $\boxed{\Pi_{\mathsf{SIS}}}$, a rejection sampling step was included in order to not reveal values of too large norm to the verifier, since they may leak information about the small witness. Furthermore, in the special soundness proof for $\boxed{\Pi_{\mathsf{SIS}}}$ in

Theorem 7.4, there is a soundness gap between what we can prove and what the extractor given in Figure 10 can retrieve, leading to us to introduce an altered relation, as a relaxed version of the first one. We similarly define a relaxed version of $\mathcal{R}(\Psi, \alpha)$ as

$$\mathcal{R}(\Psi, \beta, \zeta) = \{(Y \in N; y \in M) : \zeta \cdot Y = \Psi(y),\ \|y\| \leq \beta\} \tag{37}$$

for some $\zeta \in \mathcal{R}$. Here, we have fixed the relaxed value ζ for the given relation, such that it will be easier to describe the soundness slack when composing protocols further. This is done instead of allowing each element of the relation to satisfy $\bar{c} \cdot Y = \Psi(y)$ for some arbitrary challenge difference, which is what we did for the relation $\bar{\mathcal{R}}_{\mathsf{SIS}}$ in Equation 28.

We begin by establishing some definitions and concepts that are necessary to construct the protocols we will use to obtain the final compressed Σ-protocol.

Definition 7.1 (V-Hiding and β-Bounded Sampling [ACK21])**.** Let $\mathcal{R}$ be a ring, M be an $\mathcal{R}$-module, and let V be a set such that $V \subseteq M$. Let $\mathcal{D}$ be an efficiently computable distribution of over M, and let $\mathcal{F}$ be a PPT algorithm. We say that $(\mathcal{D}, \mathcal{F})$ is V-hiding if there exists a PPT algorithm $\mathcal{F}'$, such that the output distributions of the following are statistically close:

0. $\mathcal{F}$ upon input $(r \in M, v \in V)$ outputs either $r + v$ or $\perp$,
1. $\mathcal{F}'$ upon input 1^λ outputs either $m \in M$ or $\perp$.

In other words, no adversary $\mathcal{A}$ should have non-negligible advantage

$$\begin{aligned}\mathsf{Adv}_{V\text{-hiding}}(\mathcal{A}) = |\Pr[b = 1|\ b \leftarrow\!\$\ \mathcal{A}(\mathcal{F}'(1^\lambda))] \\ - \Pr[b = 1|\ b \leftarrow\!\$\ \mathcal{A}(\mathcal{F}(r, v))]|.\end{aligned} \tag{38}$$

Furthermore, let $\beta \in \mathbb{N}$. We say that $(\mathcal{D}, \mathcal{F})$ is β-bounded if for all $v \in V$ and $r \in M$ output by $\mathcal{D}$ with non-zero probability such that $\mathcal{F}(r, v) \neq \perp$, it holds that $\|\mathcal{F}(r, v)\| \leq \beta$. ∎

Remark 7.2. As a consequence of V-hiding, the abort probability of $\mathcal{F}$ can be upper-bounded by

$$\delta := \Pr[\mathcal{F}'(1^\lambda) = \perp] + 2^{-\lambda}.$$

Definition 7.1 gives an abstraction of rejection sampling, and in Lemma 7.13 we show how it can be instantiated as uniform rejection sampling. The property β-bounded can help us ensure that the output of $\mathcal{F}$ is not too large compared to the secret witness, while V-hiding helps us guarantee that adding elements from V to elements of M adds sufficient randomness to hide them.

Definition 7.3 (ζ-Exceptional Subset [ACK21])**.** Let $\mathcal{R}$ be a ring, $\zeta \in \mathcal{R}$ and $\mathcal{C} \subseteq \mathcal{R}$. We say that $\mathcal{C}$ is a ζ-exceptional subset of $\mathcal{R}$ if for all distinct pairs $c, c' \in \mathcal{C}$, there exists a non-zero ring element $a \in \mathcal{R}$ such that

$$a(c - c') = \zeta.$$

For ζ-exceptional subsets $\mathcal{C}$, we define the upper bounds $w(\mathcal{C})$ and $\bar{w}(\mathcal{C}, \zeta)$:

$$\begin{aligned} w(\mathcal{C}) &:= \max_{c \in \mathcal{C}, x \in \mathcal{R} \setminus \{0\}} \frac{\|cx\|}{\|x\|}, \\ \bar{w}(\mathcal{C}, \zeta) &:= \max_{c \neq c' \in \mathcal{C}, x \in \mathcal{R} \setminus \{0\}} \max_{a \in \mathcal{R} : a(c-c') = \zeta} \frac{\|ax\|}{\|x\|}. \end{aligned} \tag{39}$$

Here, $w(\mathcal{C})$ is such that $\|cx\| \leq w(\mathcal{C})$ for all $c \in \mathcal{C}, r \in \mathcal{R}$, and $\bar{w}(\mathcal{C}, \zeta)$ gives an upper bound on how much the norm of an element of $\mathcal{R}$ increases, when it is multiplied with the inverse of some challenge differences.

For an $\mathcal{R}$-module M with norm $\|\cdot\|$ we also define

$$\begin{aligned} w_M(\mathcal{C}) &:= \max_{c \in \mathcal{C}, x \in M \setminus \{0\}} \frac{\|cx\|}{\|x\|}, \\ \bar{w}_M(\mathcal{C}, \zeta) &:= \max_{c \neq c' \in \mathcal{C}, x \in M \setminus \{0\}} \max_{a \in \mathcal{R} : a(c-c') = \zeta} \frac{\|ax\|}{\|x\|} \end{aligned} \tag{40}$$

in a similar manner, where we let $x \in M \setminus \{0\}$ instead. ∎

ζ-exceptional subsets can be seen as a tool to create an extension of the challenge set $\bar{\mathcal{C}}$ given in Equation 22, allowing us to bound challenge difference of a certain form. We will use ζ-exceptional subsets when defining the norm bound we allow in the relaxed relation.

7.1 Σ-Protocol for Composition

We present the standard Σ-protocol Π'_0 we use as our starting point in Figure 13. As pointed out by Attema, Cramer and Kohl [ACK21, Section 5.1], the protocol is very similar to the Σ-protocol presented by Schnorr [Sch90].

Theorem 7.4. *Let $\mathcal{R}$ be a ring, and let M with norm $\|\cdot\|$ and N be $\mathcal{R}$-modules. Let $\Psi : M \to N$ be an efficiently computable $\mathcal{R}$-module homomorphism, and for $\zeta \in \mathcal{R}$, let $\mathcal{C} \subseteq \mathcal{R}$ be a ζ-exceptional subset of $\mathcal{R}$. For $\alpha, \beta \in \mathbb{N}$, let*

$$V = \{cy |\ y \in M, \|y\| \leq \alpha, c \in \mathcal{C}\},$$

and let $(\mathcal{D}, \mathcal{F})$ be a β-bounded and V-hiding distribution with abort probability upper-bounded by δ.

The protocol Π'_0 given in Figure 13 is a Σ-protocol for the relations

$$(\mathcal{R}(\Psi, \alpha), \mathcal{R}(\Psi, 2\beta\sigma, \zeta)),$$

where we let $\sigma = \bar{w}_M(\mathcal{C}, \zeta)$. It has completeness with error δ, unconditional 2-special soundness and is statistical special HVZK without aborts.

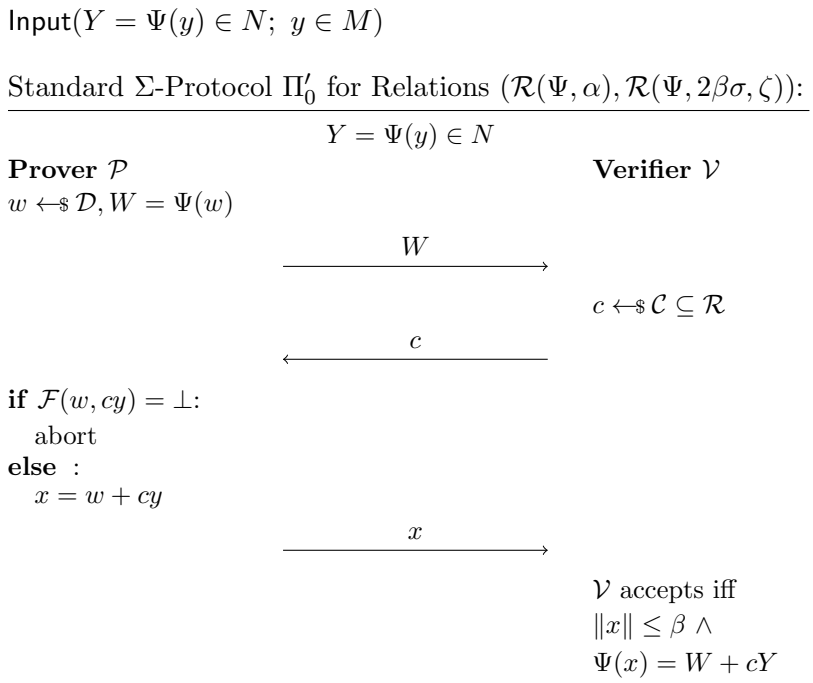


Figure 13: Standard Σ-Protocol Π_0'

Proof. We divide the proof into three parts, where we let each paragraph prove a property of being a Σ-protocol. The properties can be shown in a similar manner as the ones we have looked at previously.

Completeness Suppose that a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ follow the protocol specifications of $\Pi_0'(Y; y)$ such that $(Y, y) \in \mathcal{R}(\Psi, \alpha)$, and let (W, c, x) be a non-aborting transcript. Since $(\mathcal{D}, \mathcal{F})$ is β-bounded, the response x has to be of norm $\|x\| \leq \beta$. Since the transcript did not abort, we know that $x = w + cy$. Using that Ψ is an $\mathcal{R}$-module homomorphism,

$$\Psi(x) = \Psi(w + cy) = \Psi(w) + c\Psi(y) = W + cY.$$

This shows that the verifier always accepts non-aborting transcripts. Furthermore, since $(\mathcal{D}, \mathcal{F})$ is V-hiding, the abort probability of $\mathcal{F}$ upon input (w, cy) is upper-bounded by δ. As this is the only requirement that decides if the transcript aborts or not, the completeness error equals δ as well.

2-Special Soundness Let $\mathcal{E}$ be an extractor that is given common input Y and two accepting transcripts (W, c, x) and (W, c', x') with $c \neq c'$ for protocol Π_0'. We want to construct a witness $\tilde{y}$ such that $(Y; \tilde{y}) \in \mathcal{R}(\Psi, 2\beta\sigma, \zeta)$. Using that Ψ is a module homomorphism,

$$\Psi(x) - \Psi(x') = (W + cY) - (W + c'Y) = (c - c')Y.$$

Since $c \neq c' \in \mathcal{C}$ and we know that $\mathcal{C}$ is ζ-exceptional, there exists an $a \in \mathcal{R}$ such that $a(c - c') = \zeta$. Therefore if we let $\tilde{y} := a(x - x')$, we obtain

$$\Psi(\tilde{y}) = a(\Psi(x) - \Psi(x')) = a(c - c') \cdot Y = \zeta \cdot Y,$$

as required. Furthermore, it holds that

$$\|\tilde{y}\| = \|a(x - x')\| \leq \bar{w}_M(\mathcal{C}, \zeta)\|x - x'\| \leq \bar{w}_M(\mathcal{C}, \zeta)2\beta$$

as required, by the definition of $\bar{w}_M(\mathcal{C}, \zeta)$ and that $\|x\|, \|x'\| \leq \beta$. This shows that the extractor $\mathcal{E}$ manages to produce a valid witness $\tilde{y}$ for common input Y such that $(Y, \tilde{y})$ is an element of the relaxed relation $\mathcal{R}(\Psi, 2\beta\sigma, \zeta)$.

Special HVZK without aborts Let $\mathcal{F}'$ be a PPT algorithm that satisfies the V-hiding property of $(\mathcal{D}, \mathcal{F})$. We construct a simulator $\mathcal{S}$ for Π'_0, given in Figure 14.

Simulator $\mathcal{S}\,(Y = \Psi(y), c \in \mathcal{C})$:

1 : Run $x \leftarrow\!\!\$\ \mathcal{F}'(1^\lambda)$
2 : **if** $x \neq \perp$:
3 : Compute $W := \Psi(x) - c \cdot Y$
4 : **return** $\mathcal{T} = (W, c, x)$

Figure 14: Simulator for Special HVZK of Protocol Π'_0

Since $(\mathcal{D}, \mathcal{F})$ is V-hiding, the output distributions of $\mathcal{F}$ and $\mathcal{F}'$ are statistically close, which gives the distance between x-values in the real and simulated transcripts. The value W is decided by the choices of x, c and Y, and therefore the distribution of real and simulated transcripts are statistically close as well. This shows special HVZK in the case of no aborts. □

Remark 7.5. In order to achieve special HVZK with aborts, the prover can either replace the first message W that is sent in clear with a commitment of it using a statistically hiding commitment scheme, or it can send a hashed version of it as we did in protocol $\boxed{\Pi_{\mathsf{SIS}}}$ in Figure 9. If the prover chooses to commit to the first message, the response message has to include an opening of it when the protocol does not abort, and if the prover chooses to hash the first message, the value has to be chosen from a large set in such a way that it is not easily predictable by the verifier.

Applying either of these methods will however give us a weaker 2-special soundness guarantee given that we still want statistical special HVZK, as we will have to go from unconditional to computational special soundness. Similarly to how special soundness gives us guarantees against malicious provers and how special HVZK guarantees that the honest verifier does not

learn anything from talking to the prover, the binding property of a commitment scheme gives guarantees against cheating senders, and the hiding property gives guarantees against recipients that tries to learn what message the sender has commited to. As the prover is the one sending a commited value to the verifier, we must use a commitment scheme that is unconditionally hiding to continue to have statistical special HVZK. Any commitment scheme cannot be both unconditionally hiding and binding at once, so we must settle for computational binding which then implies we can only achieve computational 2-special soundness.

7.2 The Generic Compression Mechanism

Similarly to protocol Π_0 (Figure 4), the response message x in protocol Π_0' (Figure 13) is a trivial PoK for showing that $\Psi(x) = W + cY$. Following the same strategy as in Section 4.3 where we looked at the compression mechanism by Attema and Cramer [AC20], we want to take inspiration from the folding mechanism in Bulletproofs [Bün+18] to replace the last message with a smaller PoK for the same statement, in a way that can be repeated iteratively. The following definition give us a tool with the properties we need to construct the compression mechanism in the lattice setting.

Definition 7.6 (Extractable Compression Function [ACK21])**.** Let M and M' be $\mathcal{R}$-modules, where M has even rank n, and M' has rank $n/2$ over $\mathcal{R}$. Let $\mathcal{C} \subseteq \mathcal{R}$ be a ζ-exceptional subset. Let $\mathsf{Comp} = \{\mathsf{Comp}_c : M \to M' : c \in \mathcal{C}\}$ and $\Phi := \{\Phi_c : M' \to M : c \in \mathcal{C}\}$, where Φ_c is an $\mathcal{R}$-module homomorphism for each $c \in \mathcal{C}$. We say that (Comp, Φ) is an extractable compression function for $\mathcal{C}$ if there exists maps $\pi_L, \pi_R : M \to M'$ such that for all $c \in \mathcal{C}$,

$$\Phi_c(\mathsf{Comp}_c(x)) = \pi_L(x) + c \cdot x + c^2 \cdot \pi_R(x). \tag{41}$$

Furthermore, we say that (Comp, Φ) is (τ, τ')- preserving if for all $c \in \mathcal{C}$, $x \in M, z \in M'$:

$$\begin{aligned} \|\mathsf{Comp}_c(x)\| &\leq \tau \cdot \|x\|, \\ \|\Phi_c(z)\| &\leq \tau' \cdot \|z\|. \end{aligned}$$

Throughout this thesis, we refer to elements of Comp as compression functions, and elements of Φ as compression homomorphisms. ∎

Remark 7.7. By letting $M = \mathcal{R}^n$ and $M = \mathcal{R}^{n/2}$, the Bulletproof compression mechanism used in protocol Π_2 in Section 4.3 can be written as an extractable compression function by defining the compression functions and compression homomorphisms by

$$\mathsf{Comp}_c(\mathbf{z}) := \mathbf{z}_L + c \cdot \mathbf{z}_R \quad \text{and} \quad \Phi_c(\mathbf{z}) := \begin{pmatrix} c\mathbf{z} \\ \mathbf{z} \end{pmatrix}, \tag{42}$$

and also defining the projection maps as

$$\pi_L(\mathbf{z}) := \begin{pmatrix} 0 \\ \mathbf{z}_L \end{pmatrix} \quad \text{and} \quad \pi_R(\mathbf{z}) := \begin{pmatrix} \mathbf{z}_R \\ 0 \end{pmatrix}. \tag{43}$$

Moreover, it is $(1 + w(\mathcal{C}), w(\mathcal{C}))$-preserving, since for all $c \in \mathcal{C}$ and $\mathbf{z} \in M$, we have that

$$\|\mathbf{z}_L + c \cdot \mathbf{z}_R\| \leq \|\mathbf{z}\| + w(\mathcal{C})\|\mathbf{z}\| = (1 + w(\mathcal{C}))\|\mathbf{z}\| \quad \text{and}$$
$$\left\|\begin{pmatrix} c\mathbf{z} \\ \mathbf{z} \end{pmatrix}\right\| \leq w(\mathcal{C})\|\mathbf{z}\|.$$

For an extractable compression function (Comp, Φ) that is (τ, τ')-preserving, we present the generic compression mechanism in Figure 15. We want to apply a compression function Comp_c dependent on a given challenge c to the value x that we want to construct a PoK for. In order for the verifier to accept the conversation, we reveal partial evaluations of the secret value x using the projection maps composed with the compression homomorphism Φ_c, before sending $\mathsf{Comp}_c(x)$ as the response.

Note that if the extractable compression function is instantiated as in Equation 42 with projection maps as in Equation 43 while also defining the homomorphism as in Equation 36, the protocols Π_1' and Π_2 from Figure 7 coincide given that we omit the norm bound.

$\mathsf{Input}(X = \Psi(x) \in N;\ x \in M)$

Compression Mechanism Π_1' for Relations $(\mathcal{R}(\Psi, \beta), \mathcal{R}(\Psi, \beta\sigma, \zeta^3))$:

$X = \Psi(x) \in N$

Prover $\mathcal{P}$		**Verifier** $\mathcal{V}$
$A = \Psi(\pi_L(x))$		
$B = \Psi(\pi_R(x))$		
	$\xrightarrow{A, B}$	
		$c \leftarrow\!\$\ \mathcal{C} \subseteq \mathcal{R}$
	$\xleftarrow{c}$	
$z = \mathsf{Comp}_c(x)$		
	$\xrightarrow{z}$	
		$\mathcal{V}$ accepts iff
		$\|z\| \leq \tau \cdot \beta \wedge$
		$A + cX + c^2 B = \Psi(\Phi_c(z))$

Figure 15: Generic Compression Mechanism

Theorem 7.8 (Generic Compression Mechanism [ACK21])**.** *Let M, M' and N be $\mathcal{R}$-modules for a commutative ring $\mathcal{R}$, where M has even rank n, and M' has rank $n/2$ over $\mathcal{R}$, and let $\mathcal{C} \subseteq \mathcal{R}$ be ζ-exceptional for $\zeta \in \mathcal{R}$. Let $\Psi : M \to N$ be an $\mathcal{R}$-module homomorphism, and let (Comp, Φ) be a (τ, τ')-norm preserving extractable compression function with projection maps π_L and π_R, and let $\sigma := 6\bar{w}_M(\mathcal{C}, \zeta)^3 \cdot w_M(\mathcal{C})^2 \cdot \tau' \cdot \tau$. Then the protocol Π_1 given in Figure 15 for relations $(\mathcal{R}(\Psi, \beta), \mathcal{R}(\Psi, \beta\sigma, \zeta^3))$ has perfect completeness and 3-special soundness.*

Proof. The proof is divided into two parts, and we note that the 3-special soundness proof for Π_1' shares similarities with the 3-special soundness proof of Π_1 given in Theorem 4.7.

Completeness Suppose that a prover $\mathcal{P}$ and verifier $\mathcal{V}$ follows the protocol specifications of $\Pi_1'(X; x)$ for $(X, x) \in \mathcal{R}(\Psi, \beta)$, and let $((A, B), c, z)$ be a resulting transcript. Since Comp_c is τ-preserving for all $c \in \mathcal{C}$ and $\|x\| \leq \beta$ as $(X, x) \in \mathcal{R}(\Psi, \beta)$, we have that

$$\|z\| = \|\mathsf{Comp}_c(x)\| \leq \tau \cdot \|x\| \leq \tau \cdot \beta,$$

as required. Using that $\Phi_c \circ \mathsf{Comp}_c$ satisfies Equation 41 for all $c \in \mathcal{C}$ and that Ψ is an $\mathcal{R}$-module homomorphism, we also have that

$$\begin{aligned}
\Psi(\Phi_c(z)) &= \Psi(\Phi_c(\mathsf{Comp}_c(x))) \\
&= \Psi(\pi_L(x) + cx + c^2\pi_R(x)) \\
&= \Psi(\pi_L(x)) + c\Psi(x) + c^2\Psi(\pi_R(x)) \\
&= A + cX + C^2B,
\end{aligned}$$

which shows that the verifier always accepts.

3-Special Soundness Let $\mathcal{E}$ be an extractor that is given common input X and three accepting transcripts $((A, B), c_1, z_1)$, $((A, B), c_2, z_2)$ and $((A, B), c_3, z_3)$ for $\Pi_1'(X; x)$, with $c_i \neq c_j$ for all i, j. We want to construct an extractor $\mathcal{E}$ that returns a witness $\tilde{x}$ such that $(X, \tilde{x}) \in \mathcal{R}(\Psi, \beta\sigma, \zeta^3)$.

For each $i = 1, 2, 3$, we know that

$$A + c_iX + c_i^2B = \Psi(\Phi_{c_i}(z_i)).$$

This can be rewritten as

$$\begin{pmatrix} 1 & c_1 & c_1^2 \\ 1 & c_2 & c_2^2 \\ 1 & c_3 & c_3^2 \end{pmatrix} \begin{pmatrix} A \\ X \\ B \end{pmatrix} = \begin{pmatrix} \Psi(\Phi_{c_1}(z_1)) \\ \Psi(\Phi_{c_2}(z_2)) \\ \Psi(\Phi_{c_3}(z_3)) \end{pmatrix} \in N^3, \tag{44}$$

where

$$V := \begin{pmatrix} 1 & c_1 & c_1^2 \\ 1 & c_2 & c_2^2 \\ 1 & c_3 & c_3^2 \end{pmatrix}$$

is a Vandermonde matrix of the same form as the one we used in the proof of Theorem 4.7.

We want to retrieve an $\tilde{x}$ of sufficiently small norm such that $\Psi(\tilde{x}) = X$, and we therefore want to try and solve Equation 44 for X. Previously in the 3-special soundness proof of Theorem 4.7, we used that the determinant $\det(V) = (c_3 - c_1)(c_3 - c_2)(c_2 - c_1)$ is non-zero under the assumptions that $c_i \neq c_j$ for all i, j to find an inverse matrix V^{-1} to multiply with on each side. Now however, we are working over an arbitrary ring $\mathcal{R}$, and we are not guaranteed that an inverse V^{-1} exists. To circumvent this, we still want to find values a_1, a_2, a_3 such that

$$\begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} \cdot V = \begin{pmatrix} 0 & \tilde{c} & 0 \end{pmatrix}, \tag{45}$$

for some $\tilde{c} \in \mathcal{R}^*$. We do so to find an alternative expression for X multiplied with some constant, using a linear combination of the values we already know. By the expression above, we know that a_1, a_2 and a_3 has to satisfy the equations

$$\begin{aligned} a_1 + a_2 + a_3 &= 0 \quad \text{and} \\ a_1 c_1^2 + a_2 c_2^2 + a_3 c_3^2 &= 0. \end{aligned}$$

We let $a_1 = -a_2 - a_3$ and get that $(c_2^2 - c_1^2)a_2 + (c_3^2 - c_1^2)a_3 = 0$. Therefore we can choose $a_2 = c_1^2 - c_3^2$ and $a_3 = c_2^2 - c_1^2$, and when solving again for a_1 we obtain that

$$\begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} = \begin{pmatrix} c_3^2 - c_2^2 & c_1^2 - c_3^2 & c_2^2 - c_1^2 \end{pmatrix}.$$

If we plug this back into Equation 45, we obtain that

$$\begin{aligned} \tilde{c} &= (c_3^2 - c_2^2)c_1 + (c_1^2 - c_3^2)c_2 + (c_2^2 - c_1^2)c_3 \\ &= (c_1 - c_2)(c_1 - c_3)(c_2 - c_3) \end{aligned}$$

which is non-zero since $c_i \neq c_j$ for all i, j. The last equality is obtained by using that $\mathcal{R}$ is commutative. Since $\mathcal{C}$ is ζ-exceptional, there must exist a value $a \in \mathcal{R}$ such that $a \cdot \tilde{c} = \zeta^3$. We now let

$$\tilde{x} := a \cdot \sum_{i=1}^{3} a_i \Phi_{c_i}(z_i) \in M,$$

which satisfies

$$\begin{aligned} \Psi(\tilde{x}) = a \cdot \sum_{i=1}^{3} a_i \Psi(\Phi_{c_i}(z_i)) &= a \cdot \sum_{i=1}^{3} a_i (A + c_i X + c_i^2 B) \\ = a \cdot (0 \cdot A + \tilde{c} \cdot X + 0 \cdot B) &= \zeta^3 \cdot X \end{aligned}$$

as required.

Using that the extractable compression function (Comp, Φ) is (τ, τ')-preserving, as well as the definitions of $w_M(\bar{c})$ and $\bar{w}_M(\mathcal{C}, \zeta)$, we obtain

$$\begin{aligned}
\|\tilde{x}\| &\leq \bar{w}_M(\mathcal{C}, \zeta)^3 \cdot \|\sum_{i=1}^{3} a_i \Phi_{c_i}(z_i)\| \\
&\leq \bar{w}_M(\mathcal{C}, \zeta)^3 \cdot \sum_{i=1}^{3} 2w_M(\mathcal{C})^2 \cdot \|\Phi_{c_i}(z_i)\| \\
&\leq \bar{w}_M(\mathcal{C}, \zeta)^3 \cdot \sum_{i=1}^{3} 2w_M(\mathcal{C})^2 \cdot \tau' \cdot \|z_i\| \\
&\leq \bar{w}_M(\mathcal{C}, \zeta)^3 \cdot \sum_{i=1}^{3} 2w_M(\mathcal{C})^2 \cdot \tau' \cdot \tau \cdot \beta \\
&= 6\bar{w}_M(\mathcal{C}, \zeta)^3 \cdot w_M(\mathcal{C})^2 \cdot \tau' \cdot \tau \cdot \beta \\
&= \sigma \cdot \beta,
\end{aligned}$$

and note that we defined σ to equal $6\bar{w}_M(\mathcal{C}, \zeta)^3 \cdot w_M(\mathcal{C})^2 \cdot \tau' \cdot \tau$. This shows that the extractor $\mathcal{E}$ produces a witness $\tilde{x}$ such that $(X, \tilde{x}) \in \mathcal{R}(\Psi, \beta\sigma, \zeta^3)$, and we conclude that Π'_1 has 3-special soundness. □

7.3 General Framework for Compressed Σ-Protocols

In protocol Π'_1, the response message z is a PoK of $Q := A + cX + c^2B$ for the relation $\mathcal{R}(\Psi \circ \Phi_c, \tau \cdot \beta)$ from Equation 35. This allows us to compose protocol Π'_1 with itself iteratively, replacing the response message with a new run of Π'_1 until we have halved its size the desirable amount.

The final compressed Σ-protocol is obtained by combining the protocols Π'_0 from Figure 13 and Π'_1 from Figure 15. Here, we have not written Π'_1 as a recursive protocol, and therefore define the compressed Σ-protocol as

$$\Pi_{\mathsf{comp}} := \Pi'_1 \diamond \ldots \diamond \Pi'_1 \diamond \Pi'_0. \tag{46}$$

For each time we compose with protocol Π'_1, the size of the last message being sent is halved. If we let $M = \mathcal{R}^{2^\mu}$, we have to repeat the protocol Π'_1 μ times for the response message to be an element of $\mathcal{R}$.

In order to guarantee that the soundness gap that is introduced when composing the protocols is sufficiently small, a formal analysis has to be done. We refer to the work of Attema, Cramer and Kohl for this analysis [ACK21, Appendix B], where they introduce the notion of transforming protocols to guarantee that the composed protocol can have a witness extracted form it. As a result of their analysis, the soundness slack of Π_{comp} equals the product of the soundness slack of the protocols we have composed together to obtain it. If we instantiate the compression mechanism

with the $(1 + w(\mathcal{C}), w(\mathcal{C}))$-preserving extractable compression function described in Remark 7.7, we obtain the generic compressed Σ-protocol given in Corollary 7.8.1.

Corollary 7.8.1 (Generic Compressed Σ-Protocol [ACK21])**.** *Let $\mu \in \mathbb{N}$, and let $M = \mathcal{R}^{2^\mu}$ be an $\mathcal{R}$-module with infinity norm $\|\cdot\|_\infty$. Let $\Psi : M \to N$ be an $\mathcal{R}$-module homomorphism, let $\zeta \in \mathcal{R}$ and let $\mathcal{C}$ be a ζ-exceptional subset of $\mathcal{R}$. Let $\alpha, \beta \in \mathbb{N}$ and $\delta \in [0, 1)$, and define*

$$V := \{cy |\ y \in M, \|y\|_\infty \leq \alpha,\ c \in \mathcal{C}\}. \tag{47}$$

Let $(\mathcal{D}, \mathcal{F})$ be a β-bounded and V-hiding distribution with abort probability δ. Then there exists a $(2\mu + 3)$-move interactive public-coin protocol Π_{comp} for the relations

$$(\mathcal{R}(\Psi, \alpha), \mathcal{R}(\Psi, 2\beta \cdot \bar{w}(\mathcal{C}, \zeta) \cdot \sigma^\mu, \zeta^{3\mu+1})),$$

where

$$\sigma = 6 \cdot w(\mathcal{C})^3 \cdot (1 + w(\mathcal{C})) \cdot \bar{w}(\mathcal{C}, \zeta)^3.$$

It is unconditionally $(2, k_1, \ldots, k_\mu)$-special sound with $k_i = 3$ for $i = 1, \ldots, \mu$, non-abort special HVZK and has completeness with error δ.
The communication costs are:

- *$\mathcal{P} \to \mathcal{V}$: $2\mu + 1$ elements of N and 1 elements of $\mathcal{R}$.*
- *$\mathcal{V} \to \mathcal{P}$: $\mu + 1$ elements of $\mathcal{C}$.*

7.4 Compressed Σ-Protocol from MSIS

Suppose that we want to instantiate Π_{comp} from Equation 46 closely to protocol Π_c in Figure 8 by defining Ψ of a similar form as in Equation 36. That is, we want to define it as

$$\Psi(\mathbf{x}, \gamma) = (\mathsf{Com}_{\mathsf{pp}}(\mathbf{x}, \gamma), L(\mathbf{x}))$$

for some commitment scheme and linear form L. However this time, we want to make use of a lattice-based commitment scheme (Definition 7.9). The Pedersen vector commitment scheme (Definition 4.1) had the property of being compact, which allowed us to reduce the communication costs since each commitment only consisted of one group element. We similarly want the lattice-based commitment scheme to produce commitments of a size independent of the input we want to commit to.

The Lattice-based commitment scheme we will make use of, allows the prover to commit to vectors $\mathbf{x} \in \mathcal{R}_f^n$ for some n such that all the coefficients of $\mathbf{x}$ are elements in $[\eta]$ for some $\eta \in \mathbb{N}$, doing so while using randomness of some dimension r sampled with coefficients from the same set.

Definition 7.9 (Compact Lattice-Based Commitment Scheme [Ajt96; ACK21])**.** Let f be a monic and irreducible polynomial of degree d over $\mathbb{Z}$, and let $\mathcal{R}_f$ and $\mathcal{R}_{q,f}$ be defined as in Section 2.1 for a prime q. Let $\eta \in \mathbb{N}$ and define

$$S_\eta = \{x \in \mathcal{R}_f | : \|x\| \leq \eta\}. \tag{48}$$

Then the following PPT algorithms define a lattice-based commitment scheme.

- $\mathsf{Gen}(1^\lambda)$: $A_1 \leftarrow\!\$\, \mathcal{R}_{q,f}^{k\times r}, A_2 \leftarrow\!\$\, \mathcal{R}_{q,f}^{k\times n}$ return $\mathsf{pp} = (A_1, A_2)$.
- $\mathsf{Com}_{\mathsf{pp}}(\mathbf{x}, \gamma) : S_\eta^n \times S_\eta^r \to \mathcal{R}_{q,f}^k \quad (\mathbf{x}, \gamma) \mapsto A_1\gamma + A_2\mathbf{x} \mod q$.

When instantiating the commitment scheme, the sender samples γ uniformly at random from the set S_η^r. ■

Since the output of Com are elements of $\mathcal{R}_{q,f}^k$ and their size does not depend on the inputted messages that are elements of S_η^n, we say that the commitment scheme is compact. The size of the commitments is however polynomially bounded with respect to the security parameter we instantiate the system with. Furthermore, the commitment scheme satisfy the following hiding and binding guarantees.

Lemma 7.10 ([ACK21])**.** *The lattice-based commitment scheme given in Definition 7.9 is statistically hiding given that we choose r such that*

$$r \geq \frac{d \cdot k \cdot \log_2(q) + 2\lambda - 2}{d \cdot \log_2(2\eta + 1)}$$

for security parameter $\lambda \in \mathbb{N}$.

Furthermore, the commitment scheme is computationally binding under the assumption that the $\mathsf{MSIS}_{k,n+r,2\eta}$ problem is hard over $\mathcal{R}_{q,f}$.

Proof. We begin by proving that the commitment scheme is statistically hiding. That is, we want to prove that such that commited values are statistically close to uniformly chosen ones. In order to guarantee that the commitment scheme introduces a sufficient amount of randomness, we want to make use of the Leftover Hash Lemma given in Theorem 2.5.

Let $h_{A_1} : \mathcal{R}_{q,f}^r \to \mathcal{R}_{q,f}^k, \gamma \mapsto A_1\gamma$ be a family of functions indexed by $A_1 \in \mathcal{R}_{q,f}^{k\times r}$. We want to show that $\{h_{A_1}\}_{A_1 \in \mathcal{R}_{q,f}^{k\times r}}$ is a UHF (Definition 2.3), which means we want to show that for a uniformly chosen matrix A_1,

$$\Pr[A_1\gamma = A_1\gamma'] \leq \frac{1}{q^{dk}}$$

for $\gamma, \gamma' \in \mathcal{R}_{q,f}^r$ with $\gamma \neq \gamma$. Proving the statement above is equivalent to upper-bounding the probability of matrix A_1 sending a non-zero element of $\mathcal{R}_{q,f}^r$ to 0, namely showing that for $\mathbf{z} \neq 0 \in \mathcal{R}_{q,f}^r$,

$$\Pr[A_1\mathbf{z} = 0] \leq \frac{1}{q^{dk}}.$$

In general, if $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are independently chosen uniformly at random from $\mathcal{R}_{q,f}^r$, then for any non-zero $\mathbf{w} \in \mathcal{R}_{q,f}^r$, the dot product $\mathbf{v}_i \cdot \mathbf{w}$ will also for each $i = 1, \ldots, n$ be independently and uniformly distributed over $\mathcal{R}_{q,f}^r$. Here we are using that $\mathcal{R}_{q,f}$ is a field, and therefore any non-zero element is invertible. The row vectors of A_1 are sampled uniformly and independent of each other, and each entry of $A_1\mathbf{z}$ is given by a dot product of a row of A_1 with $\mathbf{z}$. Therefore $A_1\mathbf{z}$ is uniformly distributed over $\mathcal{R}_{q,f}^k$. It follows that $\Pr[A_1\mathbf{z} = 0]$ equals the probability of picking $0 \in \mathcal{R}_{q,f}^k$ at random, namely $1/q^{dk}$ as desired. This shows that $\{h_{A_1}\}_{A_1 \in \mathcal{R}_{q,f}^{k \times r}}$ is a UHF.

Since A_1 is chosen uniformly at random from $\mathcal{R}_{q,f}^{k \times r}$ and γ is sampled uniformly from S_η^r, they are independent of each other and we can apply the Leftover Hash Lemma given in Theorem 2.5. We have that $|S_\eta^r| = (2\eta+1)^{dr}$ and $|\mathcal{R}_{q,f}^k| = q^{dk}$, and note that the collision probability (Definition 2.4) β of γ equals $1/|S_\eta^r| = 1/(2\eta+1)^{dr}$ since γ is uniformly distributed. Therefore the statistical distance δ' (Definition 2.9) between $(A_1, h_{A_1}(\gamma))$ and the uniform distribution on $(\mathcal{R}_{q,f}^{k \times r}, \mathcal{R}_{q,f}^k)$ is upper-bounded by

$$\delta' \leq \frac{1}{2}\sqrt{\frac{q^{dk}}{(2\eta+1)^{dr}}}.$$

In order to obtain the desired level of security, we choose r such that

$$\frac{1}{2}\sqrt{\frac{q^{dk}}{(2\eta+1)^{dr}}} \leq 2^{-\lambda}.$$

Solving for r, this can be achieved by choosing r such that

$$r \geq \frac{d \cdot k \cdot \log_2(q) + 2\lambda - 2}{d \cdot \log_2(2\eta+1)}.$$

Therefore the commitment scheme is statistically hiding given that γ is chosen uniformly at random, and r satisfies the given constraint. Choosing the r value to satisfy the inequality above then ensures that the value γ used in the commitment scheme adds sufficient randomness to make the two distributions statistically close.

We proceed by showing that the commitment scheme is computationally binding when we assume that the $\mathsf{MSIS}_{k,n+r,2\eta}$ problem is hard over $\mathcal{R}_{q,f}$. Suppose that $(\mathbf{x}, \gamma)$ and $(\mathbf{x}', \gamma')$ with $\mathbf{x} \neq \mathbf{x}'$ are two distinct openings of a commitment P. Then if we let

$$\mathbf{s} := (\mathbf{x} - \mathbf{x}', \gamma - \gamma'),$$

we have that $\|\mathbf{s}\| \leq 2\eta$, since all the coefficients in each of $\mathbf{x}, \mathbf{x}', \gamma$ and γ' are bounded by η. In addition to this, $\mathbf{s}$ satisfies

$$\begin{aligned}(A_2, A_1) \cdot \mathbf{s} &= A_2(\gamma - \gamma') + A_1(\mathbf{x} - \mathbf{x}') \\ &= (A_2\gamma + A_1\mathbf{x}) - (A_2\gamma' + A_1\mathbf{x}') = P - P = 0,\end{aligned}$$

and is therefore a solution to the $\mathsf{MSIS}_{k,n+r,2\eta}$ problem. Therefore the commitment scheme is computationally binding, which completes our proof. □

Similarly to how we introduced relaxed relations to allow witnesses we can extract in special soundness to be of slightly larger norm, we introduce a relaxed version of commitment schemes.

Definition 7.11 ((β,ζ)-Relaxed Opening [ACK21])**.** Suppose $\beta \in \mathbb{N}$ and let $\zeta \in \mathcal{R}$. We say that $(\mathbf{x},\gamma)$ is a (β,ζ)-relaxed opening $(\mathbf{x},\gamma) \in \mathcal{R}^n \times \mathcal{R}^r$ of a commitment P if

$$\mathsf{Com}_{\mathsf{pp}}(\mathbf{x},\gamma) = \zeta \cdot P \quad \text{and} \quad \|(\mathbf{x},\gamma)\|_\infty \leq \beta.$$

Breaking binding of a (β,ζ)-relaxed version of the commitment scheme in Definition 7.9, can be shown to break MSIS with a norm bound only twice the size of the one with exact openings.

Lemma 7.12 (Binding of (β,ζ)-Relaxed Openings [ACK21])**.** *Suppose that $\beta \in \mathbb{N}$ and $\zeta \in \mathcal{R}_f$. The lattice-based commitment scheme described in Definition 7.9 is computationally binding with (β,ζ)-relaxed openings under the assumption that the $\mathsf{MSIS}_{k,n+r,2\beta}$ problem is hard over $\mathcal{R}_{q,f}$.*

Proof. Suppose that $(\mathbf{x},\gamma)$ and $(\mathbf{x}',\gamma')$ with $\mathbf{x} \neq \mathbf{x}'$ are two distinct (β,ζ)-relaxed openings of a commitment P. Then if we let

$$\mathbf{s} := (\mathbf{x}-\mathbf{x}', \gamma-\gamma'),$$

we have that $\|\mathbf{s}\| \leq 2\beta$, since all the coefficients in each of $\mathbf{x},\mathbf{x}',\gamma$ and γ' are bounded by β by assumption. In addition to this, $\mathbf{s}$ satisfies

$$\begin{aligned}(A_2, A_1)\cdot\mathbf{s} &= A_2(\gamma-\gamma') + A_1(\mathbf{x}-\mathbf{x}') \\ &= (A_1\gamma + A_2\mathbf{x}) - (A_1\gamma' + A_2\mathbf{x}') = \zeta\cdot P - \zeta\cdot P = 0,\end{aligned}$$

and is therefore a solution to the $\mathsf{MSIS}_{k,n+r,2\beta}$ problem. Hence, the commitment scheme is computationally (β,ζ)-binding, which completes our proof. □

In Lemma 7.13, we show how to instantiate the abstracted notion of rejection sampling from Definition 7.1 with uniform rejection sampling. Since our goal is to instantiate the compressed Σ-protocol given in Corollary 7.8.1, we want the distribution $(\mathcal{D},\mathcal{F})$ to be V-hiding on the set V defined in Equation 47. Furthermore, we want to make use of the lattice-based commitment scheme described in Definition 7.9, which takes in messages from the set S_η given in Equation 48. We let $\eta = \lceil (p-1)/2 \rceil$, and therefore state V-hiding on a set with that given bound as well.

Lemma 7.13 (Uniform Rejection Sampling [ACK21])**.** *Let f be a monic and irreducible polynomial of degree d over $\mathbb{Z}$, and let $\mathcal{R}_f$ be defined as in Section 2.1. Let $\mathcal{C} \in \mathcal{R}_f$, and suppose $m, \eta \in \mathbb{N}$. Let $\|\cdot\|_\infty$ be the infinity norm of the coefficients in $\mathcal{R}_f^m$. Let*

$$w(\mathcal{C}) = \max_{c \in \mathcal{C}, x \in \mathcal{R} \setminus \{0\}} \frac{\|cx\|_\infty}{\|x\|_\infty},$$

and define

$$V := \{cx \in \mathcal{R}_f^m : c \in \mathcal{C} \subseteq \mathcal{R}, \|x\|_\infty \leq \lceil (p-1)/2 \rceil\}.$$

Let $\mathcal{D}$ be the uniform distribution over $M := \{x \in \mathcal{R}_f^m : \|x\|_\infty \leq \eta\}$ and let

$$\mathcal{F}(r, v) := \begin{cases} \perp, & \text{if} \quad \|v + r\|_\infty > \eta - w(\mathcal{C})\lceil (p-1)/2 \rceil, \\ v + r, & \text{otherwise.} \end{cases}$$

Then $(\mathcal{D}, \mathcal{F})$ is perfectly V-hiding and β-bounded for

$$\beta = \eta - w(\mathcal{C})\lceil (p-1)/2 \rceil$$

with abort probability

$$\delta \leq 1 - e^{\frac{w(\mathcal{C})pmd}{2\eta+1}}.$$

Proof. Let δ denote the abort probability of algorithm $\mathcal{F}$, and let

$$\alpha := w(\mathcal{C})\lceil (p-1)/2 \rceil.$$

Then we have that

$$\delta = 1 - \Pr\left[\|v + r\|_\infty \leq \eta - \alpha : v \in V, r \leftarrow\!\!\$\, \mathcal{D}\right].$$

We note that how we calculate δ is similar to how we obtained Equation 30. Using the notation introduced in Section 2.1, we can express $v + r$ by its polynomial coefficients in an integer vector as

$$\mathcal{V}_{v+r} := \mathcal{V}_v + \mathcal{V}_r \in \mathbb{Z}^{dm}.$$

In order to have $v + r \neq \perp$, we must have that each coefficient $\nu_{v+r} \in \mathcal{V}_{v+r}$ satisfies $\nu_{v+r} \in [\eta - \alpha]$, where $|[\eta - \alpha]| = 2(\eta - \alpha) + 1$. Therefore the number of favorable outcomes equals $(2\eta + 1 - 2\alpha)^{dm}$.

We are now interested in how many possible $v + r$ there are, when we sample r from the distribution $\mathcal{D}$. For each $v \in V$, we have that

$$\|v\|_\infty = \|cx\|_\infty \leq w(\mathcal{C})\|x\|_\infty \leq w(\mathcal{C})\lceil (p-1)/2 \rceil = \alpha.$$

This holds for any coefficient $\nu_v \in \mathcal{V}_v$, and also we have that $\|r\|_\infty \leq \eta$ by sampling it uniformly from M. Therefore for any coefficient ν_v, we have

that $\nu_{v+r} \in \{\nu_v + [\eta]\}$. This set is of the same size as $[\eta]$, and therefore for each of the dm coefficients in $\mathcal{V}_{v+r}$, there are $2\eta + 1$ possibilities. Note that $[\eta - \alpha] \subseteq [\nu_v + [\eta]]$ since $\nu_v \in [\alpha]$, and therefore all the favorable outcomes are contained in the possible ones.

Putting this together, we obtain that

$$\begin{aligned}
\delta &= 1 - \Pr\left[\|v + r\|_\infty \leq \eta - \alpha : v \in V, r \leftarrow\!\!\$\ \mathcal{D}\right] \\
&= 1 - \left(\frac{2\eta + 1 - 2\alpha}{2\eta + 1}\right)^{dm} \\
&= 1 - \left(1 - \frac{2w(\mathcal{C})\lceil (p-1)/2 \rceil}{2\eta + 1}\right)^{dm} \\
&\leq 1 - \left(1 - \frac{w(\mathcal{C})p}{2\eta + 1}\right)^{dm} \\
&= 1 - e^{md \log\left(1 - \frac{w(\mathcal{C})p}{2\eta+1}\right)} \\
&\leq 1 - e^{-\frac{w(\mathcal{C})pmd}{2\eta+1}}.
\end{aligned}$$

We now continue by proving that $(\mathcal{D}, \mathcal{F})$ is perfectly V-hiding. Let $\mathcal{F}'$ be an algorithm that aborts with probability δ, and otherwise outputs

$$z \leftarrow\!\!\$\ \{x \in \mathcal{R}_f^m :\ \|z\|_\infty \leq \eta - w(\mathcal{C})\lceil (p-1)/2 \rceil\}$$

sampled uniformly at random. The output distributions $\{\mathcal{F}(r, v) |\ r \leftarrow\!\!\$\ \mathcal{D}\}$ and $\{\mathcal{F}'\}$ then both abort with exactly the same probability, and the non-aborting output has exactly the same distribution. Therefore $(\mathcal{D}, \mathcal{F})$ is V-hiding. Also, since $\mathcal{F}$ only outputs $v + r$ such that

$$\|v + r\|_\infty \leq \eta - \alpha = \eta - w(\mathcal{C})\lceil (p-1)/2 \rceil = \beta,$$

it follows directly that $(\mathcal{D}, \mathcal{F})$ is β-bounded. □

Remark 7.14. By choosing the parameters p, m, d, η such that the fraction

$$\frac{w(\mathcal{C})pmd}{2\eta + 1}$$

in the exponent is close to 0, we can decrease the abort probability of $(\mathcal{D}, \mathcal{F})$. The instantiation of the generic compressed Σ-protocol based on the MSIS assumption is given in Corollary 7.14.1. When defining the module homomorphism Ψ, we use the lattice-based commitment scheme given in Definition 7.9 with norm bound $\eta = \lceil (p-1)/2 \rceil$, as well as some arbitrary linear form L. Furthermore, we instantiate the compression mechanism with the $(1 + w(\mathcal{C}), w(\mathcal{C}))$-preserving extractable compression function from Remark 7.7, just as we did for the general Corollary 7.8.1. The resulting protocol uses a logarithmic amount of rounds, while the size of the messages being sent is polynomial in the security parameter. Therefore we achieve the poly-logarithmic communication complexity we desired.

Corollary 7.14.1 (Compressed Lattice-Based Σ-Protocol [ACK21])**.** *Let $n, r, \mu, \eta \in \mathbb{N}$ such that $n + r = 2^\mu$, and let p and q be primes. Let f be a monic and irreducible polynomial of degree d over $\mathbb{Z}$, and let $\mathcal{R}_f$, $\mathcal{R}_{q,f}$ and $\mathcal{R}_{p,f}$ be defined as in Section 2.1. Let $\zeta \in \mathcal{R}_f$ be non-zero when projected onto $\mathcal{R}_{p,f}$, and let $\mathcal{C}$ be a ζ-exceptional subset of $\mathcal{R}_f$. Let $(\mathcal{D}, \mathcal{F})$ with abort probability δ be defined as in Lemma 7.13. Let $A_1 \in \mathcal{R}_{q,f}^{k\times r}$ and $A_2 \in \mathcal{R}_{q,f}^{k\times n}$, and define*

$$\Psi : \mathcal{R}_f^n \times \mathcal{R}_f^r \to \mathcal{R}_{q,f}^k \times \mathcal{R}_{p,f}, \quad (\mathbf{x}, \gamma) \mapsto (A_1\gamma + A_2\mathbf{x} \mod q,\ L(\mathbf{x}) \mod p).$$

Then there exists a $(2\mu+3)$-move interactive public-coin protocol Π_Λ for the relation

$$\begin{aligned} \mathcal{R}_\Lambda := \Big\{ & (X = ((P, y) \in \mathcal{R}_{q,f}^k \times \mathcal{R}_{p,f}, w = (\mathbf{x}, \gamma) \in \mathcal{R}_f^n \times \mathcal{R}_f^r)) \\ & : \ \Psi(\mathbf{x}, \gamma) = (P, y),\ \|(\mathbf{x}, \gamma)\|_\infty \le \lceil (p-1)/2 \rceil \Big\} \end{aligned} \tag{49}$$

and relaxed relation

$$\begin{aligned} \mathcal{R}'_\Lambda := \Big\{ & (X' = (P', y') \in \mathcal{R}_{q,f}^k \times \mathcal{R}_{p,f}, w' = (\mathbf{x}', \gamma') \in \mathcal{R}_f^n \times \mathcal{R}_f^r) \\ & : \ \Psi(\mathbf{x}', \gamma') = \zeta^{3\mu+1} \cdot (P', y')\ , \\ & \|(\mathbf{x}', \gamma')\|_\infty \le 2(\eta - w(\mathcal{C}))\lceil (p-1)/2 \rceil \cdot \bar{w}(\mathcal{C}, \zeta) \cdot \sigma^\mu \Big\}, \end{aligned} \tag{50}$$

where

$$\sigma = 6 \cdot w(\mathcal{C})^3 \cdot (1 + w(\mathcal{C})) \cdot \bar{w}(\mathcal{C}, \zeta)^3.$$

It is unconditionally $(2, k_1, \ldots, k_\mu)$-special sound with $k_i = 3$ for $i = 1, \ldots, \mu$, non-abort special HVZK and has completeness with error

$$\delta \le 1 - e^{-\frac{w(\mathcal{C})p(n+r)d}{2\eta+1}}.$$

The communication costs are:

- *$\mathcal{P} \to \mathcal{V}$: $2\mu + 1$ elements of $\mathcal{R}_{q,f}^k$, $2\mu + 1$ elements of $\mathcal{R}_{p,f}$ and 1 elements of $\mathcal{R}_f$.*
- *$\mathcal{V} \to \mathcal{P}$: $\mu + 1$ elements of $\mathcal{C}$.*

8 Discussion

In the compression mechanisms, there is a trade-off between computational costs and communication costs. That is, even though we manage to reduce the communication costs from linear to roughly logarithmic, the prover and verifier have to do more computations when compressing the proof. For this reason, compressed Σ-protocols can be of great use in applications where there are limitations on the proof size rather than computing power.

Soundness Slack One of the topics that have not been covered in detail, is how to formally analyze the soundness slack that is introduced when composing the protocols Π_0' and Π_1' to obtain Π_{comp}, and ensure that it is low enough for the composed protocol to be of use. In order to guarantee that a witness extracted from a composed protocol is of some norm, we have to define *trees of transcripts*, which gets its name from the growth of possible transcripts one gets by including more rounds in a protocol, and the generalized notion of $(k_1, \ldots, k_n)$-special soundness. The aforementioned topic is covered by Attema, Cramer and Kohl [ACK21, Appendix B] through transforming protocols. There does however exist exact lattice-based proof systems, that eliminate the soundness slack between what we want to prove knowledge of and what we in reality can prove knowledge of. These constructions are however significantly more complex than Σ-protocols [LNP22].

Non-Linear Constraints Up until this point, we have seen how we can use compressed Σ-protocol theory to reduce the communication costs for interactive protocols where we want to prove knowledge of a partial opening of a commitment scheme that satisfies a linear constraint. These ideas can be extended to handle non-linear constraints as well, using techniques from MPC [ACK21, Section 7]. With this, a prover can, following the so-called commit-and-prove paradigm, commit to its witness and convince the verifier that it satisfies an arbitrary arithmetic circuit C such that $C(w) = 0$, where non-linear constraints include the use of multiplication gates.

QROM As mentioned in Section 3, the Fiat-Shamir transformation turns any Σ-protocol into a digital signature. However when we want to evaluate if systems are post-quantum secure, the attacker is also assumed to have access to a quantum computer. This result in its interaction with the random oracle having to be modeled slightly different than before, in order to handle an adversary that wants to evaluate superpositions of given inputs. The following setting is referred to as the Quantum Random Oracle Model (QROM). Because of the different modeling of the random oracle, security proofs in the ROM are not guaranteed to hold to in the QROM as well, and can be hard to formalize [Don+19].

Recently, an active research question has been to fill this gap between the classical ROM and the QROM. In 2021, Katsumata gave a semi-generic transform that ensures that several existing lattice-based Σ-protocols with security proofs in the ROM can be extended to the QROM as well [Kat21]. These constructions are also shown to be straight-line simulation extractable, which refers to the property of being able to extract a valid witness from a protocol without the need to rewind the prover. It is especially favorable in systems with many users where concurrent executions of a non-interactive Σ-protocol is demanded, as the running time of a rewindable extractor may increase exponentially. Other ways of turning Σ-protocols non-interactive have also emerged, such as the Fishlin transform [Fis05] that directly gives straight-line extractability. The proposed transform was not particularly efficient, however Lindell recently provided an optimized version that yields better run time and is of more practical use [CL24].

References

[Aar+25] Marius A. Aardal et al. *SQIsign*. Tech. rep. Accessed: 2025-05-27. National Institute of Standards and Technology, 2025. URL: https://sqisign.org.

[AC20] Thomas Attema and Ronald Cramer. "Compressed Σ-Protocol Theory and Practical Application to Plug & Play Secure Algorithmics". In: *Advances in Cryptology – CRYPTO 2020, Part III*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12172. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Cham, Switzerland, Aug. 2020, pp. 513–543. DOI: 10.1007/978-3-030-56877-1_18.

[ACK21] Thomas Attema, Ronald Cramer, and Lisa Kohl. "A Compressed Σ-Protocol Theory for Lattices". In: *Advances in Cryptology – CRYPTO 2021, Part II*. Ed. by Tal Malkin and Chris Peikert. Vol. 12826. Lecture Notes in Computer Science. Virtual Event: Springer, Cham, Switzerland, Aug. 2021, pp. 549–579. DOI: 10.1007/978-3-030-84245-1_19.

[Ajt96] Miklós Ajtai. "Generating Hard Instances of Lattice Problems (Extended Abstract)". In: *28th Annual ACM Symposium on Theory of Computing*. Philadephia, PA, USA: ACM Press, May 1996, pp. 99–108. DOI: 10.1145/237814.237838.

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. "On The Concrete Hardness Of Learning With Errors". In: *Journal of Mathematical Cryptology. Volume 9, Issue 3*. Oct. 2015, pp. 169–203. DOI: 10.1515/jmc-2015-0016.

[Ara+23] Diego F. Aranha, Carsten Baum, Kristian Gjøsteen, and Tjerand Silde. "Verifiable Mix-Nets and Distributed Decryption for Voting from Lattice-Based Assumptions". In: *ACM CCS 2023: 30th Conference on Computer and Communications Security*. Ed. by Weizhi Meng, Christian Damsgaard Jensen, Cas Cremers, and Engin Kirda. Copenhagen, Denmark: ACM Press, Nov. 2023, pp. 1467–1481. DOI: 10.1145/3576915.3616683.

[BG93] Mihir Bellare and Oded Goldreich. "On Defining Proofs of Knowledge". In: *Advances in Cryptology – CRYPTO'92*. Ed. by Ernest F. Brickell. Vol. 740. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer Berlin Heidelberg, Germany, Aug. 1993, pp. 390–420. DOI: 10.1007/3-540-48071-4_28.

[BL13] Foteini Baldimtsi and Anna Lysyanskaya. "Anonymous credentials light". In: *ACM CCS 2013: 20th Conference on Computer and Communications Security*. Ed. by Ahmad-Reza Sadeghi,

Virgil D. Gligor, and Moti Yung. Berlin, Germany: ACM Press, Nov. 2013, pp. 1087–1098. DOI: 10.1145/2508859.2516687.

[BR93] Mihir Bellare and Phillip Rogaway. "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols". In: *ACM CCS 93: 1st Conference on Computer and Communications Security*. Ed. by Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby. Fairfax, Virginia, USA: ACM Press, Nov. 1993, pp. 62–73. DOI: 10.1145/168588.168596.

[Bün+18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. "Bulletproofs: Short Proofs for Confidential Transactions and More". In: *2018 IEEE Symposium on Security and Privacy*. San Francisco, CA, USA: IEEE Computer Society Press, May 2018, pp. 315–334. DOI: 10.1109/SP.2018.00020.

[CL24] Yi-Hsiu Chen and Yehuda Lindell. "Optimizing and Implementing Fischlin's Transform for UC-Secure Zero Knowledge". In: *IACR Communications in Cryptology (CiC)* 1.2 (2024), p. 11. DOI: 10.62056/a66chey6b.

[Cra97] Ronald Cramer. *Modular Design of Secure yet Practical Cryptographic Protocols*. PhD Thesis. University of Amsterdam, 1997.

[Dam10] Ivan Damgård. *On Sigma-protocols*. 2010. URL: https://www.cs.au.dk/~ivan/Sigma.pdf.

[DN07] Ivan Damgård and Jesper Nielsen. "Commitment Schemes and Zero-Knowledge Protocols (2007)". In: *Lecture Notes in Computer Science - LNCS* (2007).

[Don+19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. "Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model". In: *Advances in Cryptology – CRYPTO 2019, Part II*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11693. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Cham, Switzerland, Aug. 2019, pp. 356–383. DOI: 10.1007/978-3-030-26951-7_13.

[Fis05] Marc Fischlin. "Communication-Efficient Non-interactive Proofs of Knowledge with Online Extractors". In: *Advances in Cryptology – CRYPTO 2005*. Ed. by Victor Shoup. Vol. 3621. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer Berlin Heidelberg, Germany, Aug. 2005, pp. 152–168. DOI: 10.1007/11535218_10.

[FS87] Amos Fiat and Adi Shamir. "How to Prove Yourself: Practical Solutions to Identification and Signature Problems". In: *Advances in Cryptology – CRYPTO'86*. Ed. by Andrew M. Odlyzko. Vol. 263. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer Berlin Heidelberg, Germany, Aug. 1987, pp. 186–194. DOI: 10.1007/3-540-47721-7_12.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract)". In: *17th Annual ACM Symposium on Theory of Computing*. Providence, RI, USA: ACM Press, May 1985, pp. 291–304. DOI: 10.1145/22145.22178.

[Gol01] Oded Goldreich. *Foundations of Cryptography*. Cambridge University Press, 2001.

[HL10] Carmit Hazay and Yehuda Lindell. *Efficient Secure Two-Party Protocols - Techniques and Constructions*. 1st ed. Springer Berlin, Heidelberg, 2010. ISBN: 978-3-642-14303-8. DOI: https://doi.org/10.1007/978-3-642-14303-8.

[Kat21] Shuichi Katsumata. "A New Simple Technique to Bootstrap Various Lattice Zero-Knowledge Proofs to QROM Secure NIZKs". In: *Advances in Cryptology – CRYPTO 2021, Part II*. Ed. by Tal Malkin and Chris Peikert. Vol. 12826. Lecture Notes in Computer Science. Virtual Event: Springer, Cham, Switzerland, Aug. 2021, pp. 580–610. DOI: 10.1007/978-3-030-84245-1_20.

[KTX08] Akinori Kawachi, Keisuke Tanaka, and Keita Xagawa. "Concurrently Secure Identification Schemes Based on the Worst-Case Hardness of Lattice Problems". In: *Advances in Cryptology – ASIACRYPT 2008*. Ed. by Josef Pieprzyk. Vol. 5350. Lecture Notes in Computer Science. Melbourne, Australia: Springer Berlin Heidelberg, Germany, Dec. 2008, pp. 372–389. DOI: 10.1007/978-3-540-89255-7_23.

[Lin+13] San Ling, Khoa Nguyen, Damien Stehlé, and Huaxiong Wang. "Improved Zero-Knowledge Proofs of Knowledge for the ISIS Problem, and Applications". In: *PKC 2013: 16th International Conference on Theory and Practice of Public Key Cryptography*. Ed. by Kaoru Kurosawa and Goichiro Hanaoka. Vol. 7778. Lecture Notes in Computer Science. Nara, Japan: Springer Berlin Heidelberg, Germany, Feb. 2013, pp. 107–124. DOI: 10.1007/978-3-642-36362-7_8.

[Lin16] Yehuda Lindell. *How To Simulate It - A Tutorial on the Simulation Proof Technique*. Cryptology ePrint Archive, Paper 2016/046. 2016. URL: https://eprint.iacr.org/2016/046.

[LNP22] Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. "Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General". In: *Advances in Cryptology – CRYPTO 2022, Part II*. Ed. by Yevgeniy Dodis and Thomas Shrimpton. Vol. 13508. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, Cham, Switzerland, Aug. 2022, pp. 71–101. DOI: 10.1007/978-3-031-15979-4_3.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *Advances in Cryptology – EUROCRYPT 2010*. Ed. by Henri Gilbert. Vol. 6110. Lecture Notes in Computer Science. French Riviera: Springer Berlin Heidelberg, Germany, May 2010, pp. 1–23. DOI: 10.1007/978-3-642-13190-5_1.

[LS15] Adeline Langlois and Damien Stehlé. "Worst-case to average-case reductions for module lattices". In: *Designs, Codes and Cryptography* 75.3 (2015), pp. 565–599. DOI: 10.1007/s10623-014-9938-4.

[Lyu24] Vadim Lyubashevsky. *Basic Lattice Cryptography: The concepts behind Kyber (ML-KEM) and Dilithium (ML-DSA)*. Cryptology ePrint Archive, Report 2024/1287. 2024. URL: https://eprint.iacr.org/2024/1287.

[MR09] Daniele Micciancio and Oded Regev. "Lattice-based Cryptography". In: *Post-Quantum Cryptography*. Ed. by Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 147–191. ISBN: 978-3-540-88702-7. DOI: 10.1007/978-3-540-88702-7_5. URL: https://doi.org/10.1007/978-3-540-88702-7_5.

[Nat24] National Institute of Standards and Technology. *Module-Lattice-Based Digital Signature Standard*. Department of Commerce, Washington, D.C., 2024. DOI: https://doi.org/10.6028/NIST.FIPS.204.

[Nat25a] National Institute of Standards and Technology (NIST). Accessed: 2025-05-27. 2025. URL: https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/Call-for-Proposals (visited on 05/12/2025).

[Nat25b] National Institute of Standards and Technology (NIST). Accessed: 2025-05-27. 2025. URL: https://csrc.nist.gov/projects/pqc-dig-sig/round-2-additional-signatures (visited on 03/13/2025).

[Nat25c] National Institute of Standards and Technology (NIST). Accessed: 2025-05-27. 2025. URL: https://csrc.nist.gov/projects/threshold-cryptography (visited on 05/13/2025).

[Oka93] Tatsuaki Okamoto. "Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes". In: *Advances in Cryptology – CRYPTO'92*. Ed. by Ernest F. Brickell. Vol. 740. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer Berlin Heidelberg, Germany, Aug. 1993, pp. 31–53. DOI: 10.1007/3-540-48071-4_3.

[Ped92] Torben P. Pedersen. "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing". In: *Advances in Cryptology – CRYPTO'91*. Ed. by Joan Feigenbaum. Vol. 576. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer Berlin Heidelberg, Germany, Aug. 1992, pp. 129–140. DOI: 10.1007/3-540-46766-1_9.

[PR06] Chris Peikert and Alon Rosen. "Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices". In: *TCC 2006: 3rd Theory of Cryptography Conference*. Ed. by Shai Halevi and Tal Rabin. Vol. 3876. Lecture Notes in Computer Science. New York, NY, USA: Springer Berlin Heidelberg, Germany, Mar. 2006, pp. 145–166. DOI: 10.1007/11681878_8.

[Reg05] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *37th Annual ACM Symposium on Theory of Computing*. Ed. by Harold N. Gabow and Ronald Fagin. Baltimore, MA, USA: ACM Press, May 2005, pp. 84–93. DOI: 10.1145/1060590.1060603.

[Sch90] Claus-Peter Schnorr. "Efficient Identification and Signatures for Smart Cards". In: *Advances in Cryptology – CRYPTO'89*. Ed. by Gilles Brassard. Vol. 435. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer, New York, USA, Aug. 1990, pp. 239–252. DOI: 10.1007/0-387-34805-0_22.

[Sch91] Claus-Peter Schnorr. "Efficient Signature Generation by Smart Cards". In: *Journal of Cryptology* 4.3 (Jan. 1991), pp. 161–174. DOI: 10.1007/BF00196725.

[Sho05] Victor Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, 2005.

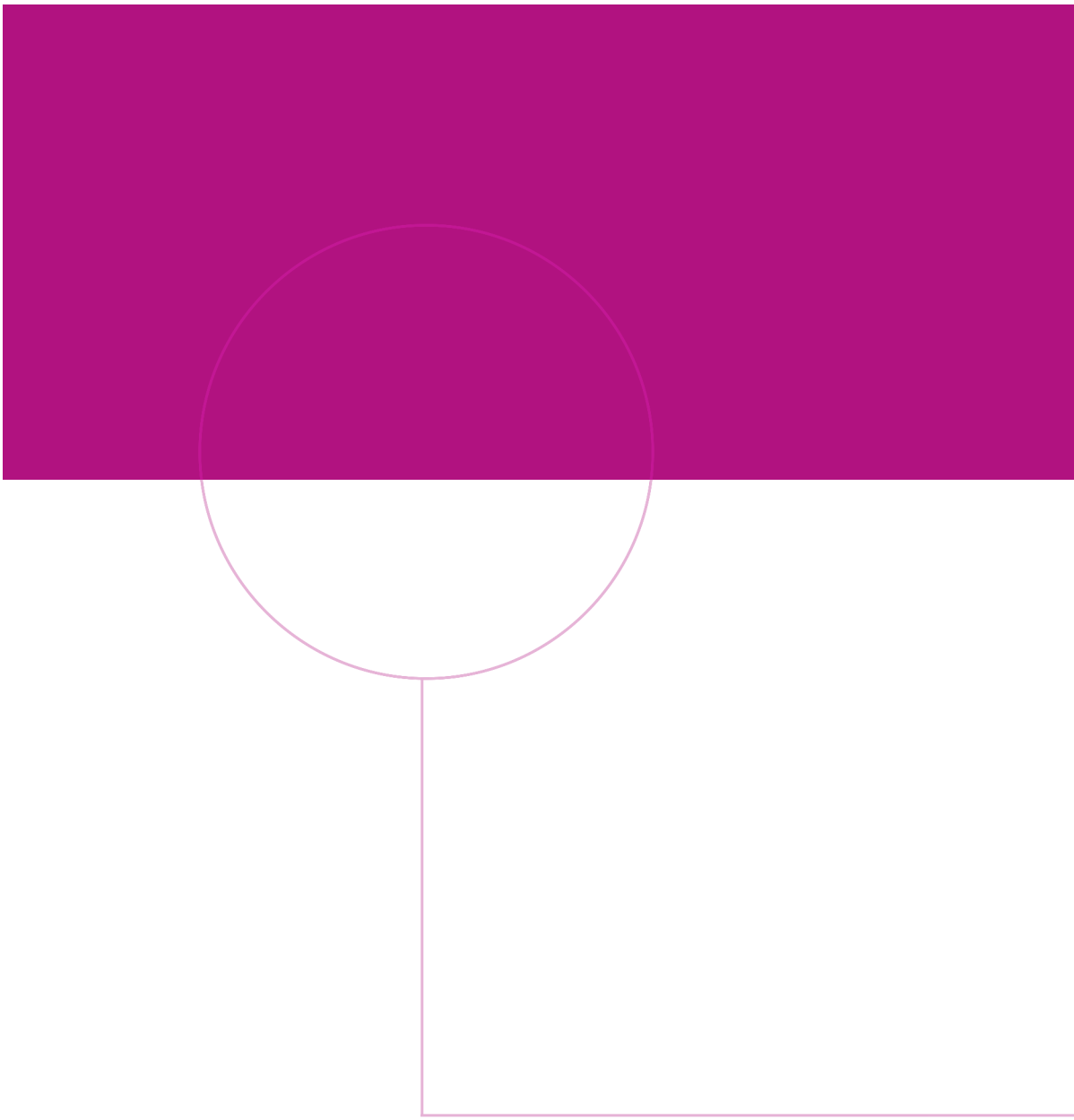
[Sho94] Peter W. Shor. "Algorithms for Quantum Computation: Discrete Logarithms and Factoring". In: *35th Annual Symposium on Foundations of Computer Science*. Santa Fe, NM, USA: IEEE Computer Society Press, Nov. 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[Ste94] Jacques Stern. "A New Identification Scheme Based on Syndrome Decoding". In: *Advances in Cryptology – CRYPTO'93*. Ed. by Douglas R. Stinson. Vol. 773. Lecture Notes in Computer Science. Santa Barbara, CA, USA: Springer Berlin Heidelberg, Germany, Aug. 1994, pp. 13–21. DOI: 10.1007/3-540-48329-2_2.

[Uni25a] United Nations. *Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all*. Accessed: 2025-05-11. 2025. URL: https://sdgs.un.org/goals/goal4.

[Uni25b] United Nations. *Promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all levels*. Accessed: 2025-05-17. 2025. URL: https://sdgs.un.org/goals/goal16#targets_and_indicators.

[Uni25c] United Nations. *The 17 Goals*. Accessed: 2025-05-11. 2025. URL: https://sdgs.un.org/goals.



NTNU

Norwegian University of
Science and Technology