Compressed Σ-protocol theory

Attema, T.

COMPRESSED Σ-PROTOCOL THEORY

THOMAS ATTEMA

Compressed Σ-Protocol Theory

Thomas Attema

This research has been supported by the Netherlands Organisation for Applied Scientific Research (TNO) and carried out at the Cryptology Group of Centrum Wiskunde & Informatica (CWI).

ISBN: 978-94-6469-236-5
Printed by: ProefschriftMaken || www.proefschriftmaken.nl
Cover design by: Univorm

Compressed Σ-Protocol Theory

Proefschrift

ter verkrijging van
de graad van doctor aan de Universiteit Leiden,
op gezag van rector magnificus prof.dr.ir. H. Bijl,
volgens besluit van het college voor promoties
te verdedigen op donderdag 1 juni 2023
klokke 13:45 uur

door

Thomas Attema
geboren te Amersfoort, Nederland
in 1990

Promotores:

Prof.dr. R. Cramer	(CWI Amsterdam & Leiden University)
Prof.dr. S.O. Fehr	(CWI Amsterdam & Leiden University)

Promotiecommissie:

Prof.dr. H.J. Hupkes	
Prof.dr. F.M. Spieksma	
Prof.dr. J. Groth	(DFINITY, Switzerland)
Prof.dr. A. Lysyanskaya	(Brown University, United States)
Prof.dr. J.B. Nielsen	(Aarhus University, Denmark)

Contents

CHAPTER 1

1 Introduction

1.1 Background

1.1.1 Mathematical Proofs

Proofs and arguments are natural concepts; they provide evidence attesting the correctness of a claim. They are a cornerstone in every scientific discipline, systematizing and structuring our understanding of the universe. However, the exact form of a proof may differ between disciplines. In natural sciences, such as biology, chemistry and physics, a "proof" may consist of a set of experimental observations confirming a hypothesis. In mathematics, a proof consists of logical arguments inferring a statement from agreed upon ground rules, referred to as axioms. Some of the first mathematical proofs date back to the ancient Greeks, with Euclid introducing the axiomatic approach.

Mathematical proofs are not unique; typically, true statements admit many different proofs. The quality or elegance of proofs can be considered a matter of taste, but their formality and rigor certainly has an aesthetic appeal. Hardy explicitly argued the importance of elegance, and Erdős often referred to aesthetically pleasing proofs as *"proofs from the book."*

> *"The mathematician's patterns, like the painter's or the poet's must be beautiful; the ideas, like the colours or the words must fit together in a harmonious way. Beauty is the first test: there is no permanent place in this world for ugly mathematics."*
>
> — G.H. Hardy, A Mathematician's Apology (1940)

The elegance of a proof is strongly related to its verifiability. In an elegant proof it is much harder to hide a mistake in obscurity, intentionally or unintentionally. Elegance thus simplifies the verification of a proof.

Hilbert, one of the greatest mathematicians of the 19th and 20th centuries, envisioned that every mathematical truth admits a proof. More precisely, he conjectured the existence of a consistent system of axioms in which every mathematical truth can be proven. In fact, he even hoped for the existence of an automated procedure for finding or "computing" these proofs. However, his hope was in vain. First, Gödel's *Incompleteness Theorem* [Göd31] shows that any formal system

of axioms (sufficiently powerful to define certain elementary arithmetic) admits mathematical statements that are unprovable, i.e., statements that can neither be proved nor disproved. Second, Church and Turing independently showed that there does not exist a finite procedure to decide on the validity of arbitrary mathematical statements [Chu36; Tur36]. In particular, there cannot exist a procedure that yields a proof for every provable statement.

The work of Hilbert, Church, Gödel, Turing and others formalized notions such as computability and algorithms, marking the birth of computer science. However, it soon became clear that many problems that are theoretically computable remain intractable in practice, simply because of the overwhelming resources that appear to be required to compute a solution. Therefore, computer science extended its scope and, besides mere computability, it started to study the efficiency of computations, giving rise to the subfield of computational complexity theory. One could argue that the elegance and beauty of mathematical proofs, referred to by Hardy and Erdős, not only imply verifiability, but also an informal notion of *efficient* verifiability.

1.1.2 Cryptography

Also in cryptography, when aiming to realize certain functionalities in the presence of adversarial entities, proofs play an essential role; they allow provers to convince verifiers of their truthfulness and honesty. However, in many cryptographic scenarios, proofs contain secret information that needs to remain hidden from the adversary. This poses the question whether it is possible to prove a claim without revealing any information about the proof beyond its existence. To some extend this is indeed possible, and proofs with this property are referred to as *zero-knowledge proofs*. Not only the theory of proofs, but also computational complexity theory has a strong connection to cryptography. Let us first discuss the latter connection before returning to the theory of (cryptographic) proofs.

Traditionally, cryptography dealt with protecting communication channels against unwanted eavesdroppers. For instance, Julius Caesar encrypted his messages using a secret key such that only his generals, with knowledge of this secret key, would be able to decrypt the encrypted messages. The security of the established communication channel held under the assumption that messages can only be decrypted *efficiently* when given the secret key; without this key decrypting should be infeasible. In other words, the security depends on the *computational complexity* of decrypting a message without a key. Unfortunately, Caesar's cipher turned out to be broken; there exist efficient procedures for decrypting even without knowledge of the secret key. Still many modern encryption schemes follow the same principle; they rely on the computational complexity, or hardness, of decrypting a message without the secret key.

Caesar's cipher and its downfall present one of the first events in an everlasting arms race between cryptographers and cryptanalysts. Cryptographers aiming to develop new encryption schemes, and cryptanalysts aiming to break these schemes. The field of research containing both cryptography and cryptanalysis is referred to as *cryptology*. A notable example in this arms race is the Vigenère cipher [Bel53], designed in 1553 and dubbed "*le chiffrage indéchiffrable*" (the unbreakable cipher). After more than 300 years, in 1863, also this "unbreakable" cipher was

broken [Kas63]. Another famous example is the Enigma code, used by the German military during the Second World War. Turing's successful efforts in breaking this code are believed to have shortened the Second World War.[1] More generally, his foundational contributions to the field of computing forced cryptographers to design ciphers capable of withstanding attacks aided by electronic computing. Currently, the next chapter of this arms race has commenced; protecting communication channels against the looming threat of quantum computers. It is known that, once available, powerful enough quantum computers will be able to break some of the most commonly used encryption schemes [Sho94]. For this reason, cryptographers worldwide are developing novel schemes capable of withstanding attacks from both classical and quantum computers. This relatively young field of research is referred to as *post-quantum cryptography*.

Previously the security of encryption schemes mainly relied on heuristics; as long as it was unknown how to break a cipher it could be considered secure. However, the developments of the 20th century, such as the birth of computational complexity theory, turned cryptology into an exact science. Additionally, Shannon's information theory rigorously defined what it means for an encryption scheme to be perfectly secure [Sha48a; Sha48b; Sha49]. He proved that one-time-pad encryption admits this level of security. More precisely, even adversaries with unlimited resources will not be able to break a one-time-pad encryption. But he also showed that perfect security requires secret keys that are at least as long as the underlying message, deeming perfect security impractical for many application scenarios.

The Vigenère cipher, Enigma code and one-time-pad are *symmetric* encryption schemes; the same secret key is used for both encryption and decryption. An important limitation of these schemes is that they can only be used after the secret key has been distributed amongst the sender and recipient of the communication channel. Moreover, before distributing the secret key, the communication channel remains unprotected, i.e., the channel cannot be used to distribute the secret key. In the late 1960s, while working at the Government Communication Headquarters (GCHQ) of the United Kingdom, Ellis started working on a solution for this key distribution problem. He managed to prove that, in principle, it should be possible to secure a communication channel without pre-shared secret keys, but he did not find a cryptographic primitive for this task. In 1973, Cocks joined GCHQ and learned about Ellis' efforts. He soon realized that the integer factorization problem possesses the asymmetry required to secure a communication channel without pre-shared keys; it is easy to compute the product of two primes but it is (or at least appears to be) hard to find the prime factors of a composite integer.

Cocks' solution was the first public-key (or asymmetric) encryption scheme. A public-key encryption scheme uses two keys; a public key pk for encrypting messages and a secret key sk for decrypting encrypted messages. Because the public key can only be used for encryption, it does not need to remain secret. For this reason, a public-key encryption scheme is not subject to the key distribution problem; the public key can be distributed over insecure communication channels. Note that, to prevent an adversary from impersonating honest users, an authentication

[1] Prior to the outbreak of the Second World War, the Polish mathematicians Marian Rejewski, Jerzy Rózycki and Henryk Zygalski broke earlier versions of the Enigma code, thereby laying the foundation for ultimately breaking the Enigma code.

mechanism is still required.

Also at GCHQ, Williamson learned about Cocks' breakthroughs. The somewhat counterintuitive notion of public-key encryption led him to believe that Cocks' solution must contain a flaw. Williamson did not manage to find a flaw, but in 1974, while trying to find one, he invented an alternative solution for the key distribution problem.

The results of GCHQ remained classified until the late 1990s, but nowadays Cocks, Ellis and Williamson are broadly recognized for their breakthroughs in cryptography. For instance, in 2010 the Institute of Electrical and Electronics Engineers (IEEE) awarded them the 100th IEEE Milestone Award.

Fortunately, in 1976, the revolutionizing notion of public-key encryption was independently put forward by Diffie and Hellman [DH76]. Without knowledge of the work done in secrecy at GCHQ, Diffie and Hellman reinvented Williamson's protocol, currently well known as the Diffie-Hellman (DH) key exchange protocol. In 1978, also Cocks' approach, i.e., basing public-key encryption on the hardness of factoring integers, was rediscovered by Rivest, Shamir and Adleman [RSA78]. Their protocol is now known as the RSA encryption scheme. Eventually, these solutions to the key distribution problem brought cryptography to the masses; nowadays encryption schemes are omnipresent in society.

In the 1970s, Diffie and Hellman not only invented public-key cryptography,[2] they also described how public-key encryption schemes rely on the existence of *trapdoor one-way functions*. These are functions that can be evaluated efficiently, but are hard to invert without knowledge of a secret trapdoor. In other words, the computational complexity of inverting certain functions underlies the security of public-key encryption schemes, again exemplifying the strong relation between cryptology and computational complexity theory.

Almost 50 years after their introduction, it still has not been proven that the functions underlying the Diffie-Hellman and RSA schemes indeed possess the required one-way property. Therefore, the security of these schemes relies on the *computational assumption* that inverting these functions is indeed intractable. Hence, this security notion still has a somewhat heuristic nature; security holds as long as no one finds an efficient procedure for solving the underlying computational problem. The confidence in a computationally secure scheme grows with the amount of research that has gone into solving the underlying problem. It is common practice to reduce breaking a cryptographic primitive to solving a well-studied computational problem. For instance, the security of RSA encryption scheme is related to the integer factorization problem; a problem that has been studied for at least 300 years. Hence, already at the time of its introduction, it had withstood cryptanalytic efforts. Based on this, and accounting for future developments, Rivest, Shamir and Adleman suggested the use of 200 digit (or 664 bit) public keys. However, the publication of this cryptographic primitive further incentivized the study of the integer factorization problem. Notably, in 1988, Pollard proposed a new algorithm for factoring integers. His approach, later improved and generalized, has become known as the *number field sieve* (NFS) [LL93]. The NFS is currently the most efficient (classical) approach known for factoring integers. In

[2] Only in 1997, GCHQ revealed that Ellis, Clifford and Williamson had already invented public-key cryptography, although in secrecy.

particular, it shows that 664 bit public keys do not offer a reasonable amount of security anymore. For this reason, it is recommended to use public RSA keys of at least 2048 bits. Similar progress has been made into solving the discrete logarithm problem underlying the Diffie-Hellman key exchange protocol. However, both the integer factorization and the discrete logarithm problem have remained classically intractable; there still does not exist an efficient, i.e., polynomial time, algorithm for solving these problems on classical computers. By contrast, Shor has shown how to solve both problems efficiently on a quantum computer [Sho94]. Hence, once powerful enough quantum computers become available, the security of the Diffie-Hellman and RSA schemes can no longer be guaranteed; post-quantum cryptography must be deployed well before this happens.

1.1.3 Multilateral Cryptography

Besides a solution for the key distribution problem, Diffie and Hellman also proposed a novel cryptographic functionality: *digital signature schemes*. A digital signature allows anyone to verify the authenticity of the sender, i.e., to verify its identity. It can also be used to show that a message has not been altered during transmission, i.e., guarantee its integrity. Diffie and Hellman therefore broadened the scope of cryptology beyond the confidentiality of communication channels. Today cryptology deals not only with confidentiality, but also with authenticity, integrity and non-repudiation.[3]

The broadened scope of cryptology inspired the development of many more advanced cryptographic functionalities. For instance, already in 1978 the concept of a *privacy homomorphism* was formulated [RAD78]. A privacy homomorphism, now known as a homomorphic encryption scheme, allows computations to be performed on encrypted data. This way the party performing the computations does not need to have access to the input data, but only to their encryption. While this concept has existed for decades, it took until 2009 before the first fully homomorphic encryption scheme, allowing *arbitrary* computations to be performed on encrypted data, was constructed [Gen09]. Further, Blum showed how two mutually distrustfully and physically separated parties can flip a coin without using a trusted third party [Blu81]. A protocol for playing a "mental" game of poker over the telephone was designed [SRA81; GM82]. And, more generally, it was shown how multiple parties can collaboratively evaluate arbitrary functions on their private inputs without revealing these inputs to each other, giving rise to the flourishing field of *multiparty computation* (MPC) [Yao82; Yao86; GMW87; CDG87; BGW88; CCD88].

A common denominator in these more advanced cryptographic primitives is that they aim to protect parties not only against external adversaries, but also against each other. For instance, guaranteeing that players are not cheating in a game of mental poker. This type of security is also referred to as *multilateral* security, whereas security against merely external adversaries is referred to as *unilateral* security. When aiming for multilateral security, it is desirable that parties *prove* that they behave honestly or, more generally, prove that the claims they make are

[3] Non-repudiation requires the identity of a sender to be verifiable not only by the sender but also by a third party. In this case, the sender cannot deny having sent the message.

valid.

1.1.4 Probabilistic Proof Systems

In cryptography, it is typically sufficient for provers to be able to prove the validity of certain subclasses or families of claims; they do not need to be able to prove the validity of all possible claims. For instance, the family of all integers composed of two prime factors; each integer in this family corresponds to the claim that it is indeed the product of two primes, and the prime factors constitute a proof for such a claim. By multiplying these factors the proof can be verified efficiently. This example describes a *proof system* for the family of all integers composed of two prime factors. More formally, a proof system for a family of valid claims L is defined by an efficiently computable and deterministic verification function V_L that, on input a claim x and a purported proof w, outputs either accept or reject. A family L and a claim x are also referred to as a *language* and a *statement*, respectively. Thus, in this formalization, a prover claims that a statement x is in the language L, i.e., $x \in L$. A proof w such that $V_L(x; w) = \mathsf{accept}$ is also called a *witness* for statement x. This formalization is due to Cook and Reckhow [CR79]. They required a proof system to be *complete* and *sound*. A proof system is complete if every valid statement $x \in L$ admits a witness w. It is sound if for every invalid statement $x \notin L$ and every w it holds that $V_L(x; w) = \mathsf{reject}$.

The class of languages that admit a proof system as above is denoted by NP. Moreover, P denotes the class of languages for which claims can be efficiently verified without knowledge of a witness, i.e., even without a witness one can efficiently verify that $x \in L \subseteq \mathsf{P}$. For this reason, proof systems for languages that are not (known to be) in P are typically more interesting. However, since verifying a proof may require less resources than computing a proof from scratch, also proof systems for languages in P can be of interest. Clearly $\mathsf{P} \subseteq \mathsf{NP}$, however it is unknown whether $\mathsf{P} = \mathsf{NP}$, i.e., whether every problem that admits an efficiently verifiable solution can also be solved efficiently. The P versus NP problem [Coo71; Lev73] of computational complexity theory is one of the biggest open problems in mathematics and computer science.

In 1985, two seminal works independently generalized the notion of a proof system, by allowing randomness, interaction and errors [Bab85; GMR85]. In this generalization, called an *interactive* or *probabilistic* proof, two parties, a prover and a verifier, interact before the verifier decides whether to accept the prover's claim. In other words, the verifier is allowed to ask the prover a number of questions before making its decision. The verifier still has to be efficient, but is no longer required to be deterministic. In fact, since the prover can predict the questions asked by a deterministic verifier, any interactive proof with a deterministic verifier can be made non-interactive, i.e., by predicting the verifier's questions the prover can output all its answers without interacting with the verifier. Therefore, interaction can only give something new for *probabilistic* verifiers. Further, the verifier of an interactive proof is allowed to make errors, i.e., it might reject valid claims (completeness error) or accept false claims (soundness error). In many occasions, by deploying certain amplification techniques, the error probabilities of interactive proofs can be made arbitrarily small. Interestingly, these relaxations have opened a whole new world of possibilities.

First, interactive proofs can be constructed for certain languages that are not known to be in NP. For instance, while it is unknown whether there exists an efficiently verifiable proof attesting that two graphs are not isomorphic, there do exist interactive proofs for the graph non-isomorphism problem [GS86; GMW86].

Second, and perhaps more surprisingly, many interactive proofs can be made *zero-knowledge*. A zero-knowledge proof is an interactive proof in which the verifier learns nothing beyond the correctness of the prover's claim. For instance, it allows a prover to convince a verifier that an integer is the product of two primes without revealing the prime factors. The notion zero-knowledge was introduced by Goldwasser, Micali and Rackoff [GMR85]. They further gave the first zero-knowledge proof system. Zero-knowledge proofs have proven to be extremely powerful cryptographic primitives. They can for instance be used to prove knowledge of a secret password without revealing the password, or to prove that votes have been tallied honestly without revealing the individual votes. The existence of zero-knowledge proofs is related to the (conjectured) existence of one-way functions. More precisely, one-way functions exist if and only if all languages in NP admit a zero-knowledge proof system [GMW91; OW93].

Third, every claim in NP admits a proof that can be verified by checking only a small part of the proof, i.e., when given a statement x and its purported witness $w \in \{0,1\}^*$, represented as a bitstring, the verifier only needs to choose, at random, a small number of w's bits to verify. A proof or witness that can be verified in this manner is called a *Probabilistically Checkable Proof* (PCP) [AS92]. One of the most influential theorems in computational complexity theory, the PCP theorem, states that every statement in NP admits a PCP [ALM+98; Din07]. Unfortunately, even with a PCP, it is impossible to construct an interactive proof system for arbitrary NP-languages with *succinct* communication [GH98], i.e., an interactive proof with communication costs that grow only sublinearly in the size of the statement x. By contrast, interactive *arguments* do not suffer from this restriction. Interactive arguments relax the soundness property of interactive proofs; instead of requiring soundness against computationally unbounded provers, interactive arguments are only required to be sound against *computationally bounded* provers. By using a certain class of one-way functions, Kilian showed how to compile any PCP into an interactive argument with succinct communication [Kil92].

1.1.5 Proofs and Arguments of Knowledge

Interactive proofs and arguments only consider provers claiming that a public statement x is in a language L. If L is an NP-language, $x \in L$ implies that the statement x admits an efficiently verifiable witness w. An interactive proof for such a language merely allows a prover to convince a verifier of the *existence* of a witness, it does not necessarily allow proving *knowledge* of such a witness. In some cases the existence of a witness is trivially satisfied and therefore a void statement, whereas knowledge of a witness is a completely different story. For instance, consider a prover claiming that an integer has a prime factorization; clearly every integer has a prime factorization, but finding or knowing such a factorization can be highly nontrivial. This example demonstrates the need for a stronger functionality, allowing a prover to prove knowledge of a witness. While early interactive proofs seemed to satisfy this requirement intuitively, it took several years before

satisfactory definitions of *knowledge soundness* and *proofs of knowledge* (PoKs), as strengthenings of ordinary soundness and interactive proofs, were derived [GMR85; TW87; FFS88; BG92].

Informally, a prover is said to know a witness w, if there exists an efficient algorithm, also referred as the extractor, capable of extracting w from the prover. To this end, the extractor may invoke the prover and reply with arbitrary messages, playing the role of the verifier. Further, the extractor is allowed to rewind the prover to previous states. Hence, a dishonest prover knows a witness w if, by running the extractor, it can efficiently compute w.

As before, an *argument of knowledge* (AoK) is a relaxation of a PoK, in which knowledge soundness only holds against computationally bounded provers.

1.1.6 Σ-Protocol Theory

In the late 1980s and the early 1990s, various zero-knowledge proof systems were introduced [FS86; FFS88; GQ88; Sch91; Oka92]. Due to its efficiency, especially Schnorr's protocol [Sch91] is still broadly used today, e.g., as the main building block for many digital signature schemes. He proposed an elegant and practical interactive proof for proving knowledge of a discrete logarithm without revealing any information about the discrete logarithm itself. In his solution, the prover first sends a message to the verifier, who replies with a challenge sampled uniformly at random from some finite set, and after receiving the prover's response, the verifier decides whether to accept or reject the prover's claim. Nowadays interactive proofs that follow the same 3-round structure and design principle as Schnorr's protocol are referred to as Σ-*protocols* [Cra96].

Over the past decades, Σ-protocol theory has developed into a well-established and versatile theory for secure algorithmics. Loosely speaking, with secure algorithmics we refer to the design of cryptographic realizations of standard algorithmic tasks. In other words, this entails porting algorithms for standard tasks to cryptographic scenarios. For instance, in MPC where mutually distrustfully parties wish to collaboratively evaluate an algorithm without revealing their input values, or in zero-knowledge where a prover aims to convince a verifier that an algorithm has been evaluated honestly, again without revealing the input.

More generally, Schnorr's interactive proof is for proving knowledge of a homomorphism preimage [Cra96; CD98], i.e., it reveals a *linear* relation between a prover's secret witness w and a public statement x. The theory of Σ-protocols has been extended towards realizing a much broader class of (not necessarily linear) functionalities. For instance, there exist Σ-protocols for proving *partial* knowledge of a subset of discrete logarithms [CDS94]. Further, it is known how to prove the satisfiability of an arithmetic circuit by using Σ-protocols [CD98], i.e., for proving the existence of an input for which the arithmetic circuit evaluates to 0. The arithmetic circuit satisfiability problem is NP-complete, i.e., every problem in the complexity class NP can be written as a circuit satisfiability problem, demonstrating the power of Σ-protocols. Moreover, Σ-protocols have been instantiated based on various cryptographic hardness assumptions beyond the discrete logarithm assumption, e.g., based on lattice assumptions, plausibly providing post-quantum security [MV03].

The versatility of Σ-protocol theory comes largely due to its *modularity*; ad-

vanced cryptographic primitives are composed of smaller abstract building blocks. These abstract building blocks are easy to analyze and can be instantiated from a wide variety of cryptographic hardness assumptions. By generic composition results, the (security) properties of cryptographic protocols are easily derived from the properties of their abstract building blocks.

1.1.7 Recent Efficiency Improvements in Proof Systems

The introduction of interactive proofs ignited a rich field of research. Notably, Wigderson, who played an influential role in the development of computational complexity theory and (interactive) proof theory, was awarded the 2021 Abel prize (along with Lovász) for his contributions to theoretical computer science and discrete mathematics. For instance, together with Goldreich and Micali, Wigderson showed that the validity of any NP-statement can be proven in zero-knowledge, assuming the existence of one-way functions [GMW86]. For an elaborate history of this field of research, we refer to his book [Wig19].

Additionally, the growing adoption of cloud and decentralized computing platforms has caused an increased interest in efficient (zero-knowledge) proof systems. Namely, in many scenarios, outsourcing computations to (untrusted) computing platforms requires verification. Verifiable computation deals with the integrity of computations outsourced to untrusted parties, i.e., it guarantees that computations have been executed correctly. The naive method for establishing *computational integrity* consists in redoing the computation and verifying its output. However, this approach has two major disadvantages. First, it is *inefficient*, i.e., it often completely beats the purpose of outsourcing computations to a party with more computational resources. Second, verifying a computation in this manner requires (private) input values to be revealed. Kilian's interactive proof for arbitrary NP-statements [Kil92] already demonstrated that zero-knowledge proof systems might offer a solution. His solution, although impractical due to a significant computational overhead, has succinct communication and is zero-knowledge. Alternatively, Σ-protocols offer concretely efficient zero-knowledge proofs for many languages [CD98]. However, their communication complexity scales linearly with the size of the statement, and the verification part of a Σ-protocol typically requires more computational resources than the computation that is to be verified. Hence, Σ-protocols only offer a partial solution for the computational integrity problem.

Recently, Bulletproofs [BCC+16; BBB+18] have been introduced as a "drop-in replacement" for Σ-Protocols in several important applications. Notably, this includes proving the satisfiability of an arithmetic circuit; protocols for this task are also referred to as *circuit zero-knowledge* protocols. The communication complexity of standard Σ-protocols is linear in the size of the circuit, whereas Bulletproofs reduce the communication complexity down to logarithmic. At the heart of Bulletproofs is an interactive proof of knowledge between a prover and verifier showing that a Pedersen commitment to a vector of large length n satisfies a multivariate polynomial equation of degree 2, defined with an inner product. This pivotal protocol stands out in that, by means of a split-and-fold technique, it ingeniously *compresses* the communication costs down to $\mathcal{O}(\log n)$ elements from $\mathcal{O}(n)$ via traditional Σ-protocols. Although this is at the expense of introducing a logarith-

mic number of communication rounds between the prover and verifier (instead of constant), its public-coin[4] nature ensures that it can be rendered non-interactive using the Fiat-Shamir heuristic [FS86]. However, applications following this novel paradigm meet a number of technical difficulties. First, this inner-product protocol is not zero-knowledge, and second, cryptographic protocol theory has to be reinvented with the quadratic constraint proved as its pivot. This leads to a deviation from the natural and well-established linearization strategy adopted by Σ-protocol theory.

Besides Bulletproofs, many novel interactive proof and, more generally, argument systems have recently been proposed. These systems offered practical computational integrity, even for lengthy and complicated computations. The current wealth of argument systems is partially due to the large number of distinctive features they possess. There does not exist a single argument system that outperforms its competitors on all terrains; the optimal solution depends largely on the application scenario. There are different performance metrics quantifying the efficiency of arguments, e.g., the computational complexities of the prover and the verifier, and the communication complexity or proof size. Moreover, most arguments require some set of public parameters known to all parties involved. Preferably this set of parameters, referred to as the *common reference string* (CRS), is as small as possible. Additionally, some argument systems enable efficiency improvements at the cost of requiring a *trusted* setup, i.e., a setup phase that is guaranteed to be executed honestly. When considering mutually distrustful parties, a trusted setup is challenging to realize. Argument systems that do not require a trusted setup are called *transparent*. Further, for their zero-knowledge and (knowledge) soundness properties, proofs and arguments may rely on different cryptographic assumptions. Some assumptions are more conservative and are even assumed to hold against quantum adversaries, e.g., the existence of one-way functions. While other assumptions, such as the knowledge of exponent (KEA) assumption, are unfalsifiable and could be considered more controversial.

1.2 Contributions

In this dissertation, we enhance Σ-protocol theory with a compression mechanism, allowing the communication complexity to be reduced from linear down to (poly)logarithmic. More precisely, we show how to combine compact commitments, arithmetic secret-sharing and an adaptation of Bulletproofs' split-and-fold technique to develop a versatile theory for the modular design of communication-efficient zero-knowledge proof systems: *Compressed Σ-Protocol Theory*. Further, we provide a number of applications and show that our approach is supported by various cryptographic platforms, including one plausibly offering post-quantum security.

A key design principle in our theory is *linearization*; we solve the linear problem instances first and then show how to linearize nonlinear ones. More precisely, our basic compressed Σ-protocols prove knowledge of homomorphism preimages,

[4] An interactive proof is *public-coin* if all of the verifiers random choices are made public, i.e., they are sent to the prover.

i.e., they prove a linear relation between a public element and its secret preimage. By a novel variation of an arithmetic secret-sharing based technique for Σ-protocols [CDP12], we then show how to linearize nonlinear problem instances, i.e., where the relation between the public statement and the secret witness is not captured by a linear mapping. Mathematically, solving the linear instances first and then linearizing the nonlinear ones is perhaps among the most natural problem solving strategies.

Additionally, we identify and close three gaps in the general theory of multi-round interactive proofs. First, we provide the first tight knowledge soundness analysis for the class of *special-sound* multi-round interactive proofs, containing Bulletproofs and compressed Σ-protocols. Second, we prove that the t-fold parallel repetition of special-sound multi-round interactive proofs optimally reduces the success probability of dishonest provers, or more precisely the knowledge error, from κ down to κ^t. Third, for special-sound interactive proofs, we show that the security loss of the Fiat-Shamir heuristic, rendering (public-coin) interactive proofs non-interactive, is independent of the number of rounds.

Below these contributions are described in more detail.

1.2.1 Compressed Σ-Protocols

We start, in Chapter 3, by combining two essential components. First, as an abstract building block, or *pivot*, we consider a basic Σ-protocol for proving knowledge of the preimage of a group homomorphism $\Psi\colon \mathbb{G}^n \to \mathbb{H}$, where $n \in \mathbb{N}$. Hence, our pivot is a Σ-protocol for proving knowledge of an n-dimensional vector. The zero-knowledge property states that evaluating the Σ-protocol does not reveal any information about the preimage. The communication complexity of this pivot grows linearly in the input dimension n. More precisely, the final message of the Σ-protocol, sent from the prover to the verifier, is a vector of dimension n. Second, this Σ-protocol is *compressed* by replacing the final (long) prover-message with an appropriate adaptation of Bulletproofs' inner-product argument; instead of sending its final message to the verifier, the prover shows it knows it. For many homomorphisms of interest, namely if the size of the codomain $\mathbb{H}$ is constant or logarithmic in n, this *compression mechanism* has a communication complexity that is logarithmic or polylogarithmic in the dimension n. Note that the compression mechanism does not need to be zero-knowledge; it replaces a message that the prover would have revealed otherwise. As a result, the required soundness and zero-knowledge properties of the Σ-protocol are preserved, but the overall communication drops from linear down to (poly)logarithmic.

1.2.1.1 Opening Linear Forms on Compact Commitments

Compressed Σ-protocols can be instantiated for a broad class of homomorphisms. A notable example is given by homomorphisms of the form

$$\psi(\mathbf{x};\gamma) = \big(\textsc{com}(\mathbf{x};\gamma), L(\mathbf{x})\big)\,,$$

where $\mathbf{x} \in \mathbb{Z}_q^n$ is the prover's secret input vector, COM is a (homomorphic) commitment scheme, γ is the commitment randomness and $L\colon \mathbb{Z}_q^n \to \mathbb{Z}_q$ is a linear form. If the commitment scheme is *compact*, i.e., the size of a commitment is constant in

the dimension n of the committed vector $\mathbf{x}$, the compression mechanism reduces the communication complexity from linear down to (poly)logarithmic in n. This instantiation allows a prover to prove knowledge of a commitment opening $(\mathbf{x}; \gamma)$ that satisfies a linear constraint captured by the linear form L. Evaluating this (compressed) Σ-protocol reveals nothing beyond the value $y = L(\mathbf{x})$, and is therefore also referred to as *opening* linear form L on committed vector $\mathbf{x}$.

1.2.1.2 Functionality Enhancements

Many techniques known from Σ-protocol theory directly apply to compressed Σ-protocols. For instance, standard amortization techniques allow many linear forms $L_1, \ldots, L_s$ to be opened, instead of just one, without increasing the overall communication complexity. Similarly, a prover can open a single linear form on many different committed vectors for the price of one. Further, using this and by plug-and-play with our basic theory, we show how to handle the application scenario where the linear form takes as secret input a long vector that is initially dispersed across several commitments. We handle this scenario by *compactifying* these dispersed components into a single commitment first. This is useful in important applications, such as commit-and-prove zero-knowledge proofs for arithmetic circuit satisfiability, where the prover has committed to the input vector *before* the arithmetic circuit is provided. More precisely, in many relevant practical scenarios, we must assume that the commitment to the prover's secret input vector, about which something is to be proved in zero-knowledge, has already been produced before the zero-knowledge protocol is run. In these scenarios commit-and-prove functionality is required. Moreover, to prepare for Strong-RSA and lattice instantiations, we further extend the compressed Σ-protocols to provers additionally claiming that the preimage is *short*.

1.2.1.3 Higher Level Functionalities

In Chapter 4, the significance of opening linear forms surfaces. First, we integrate this basic functionality with a novel variation on arithmetic secret-sharing based techniques for Σ-Protocols [CDP12], inspired by MPC. These techniques allow for linearization of nonlinear relations. More precisely, we show how to prove the correctness of large sets of committed multiplication triples $(\alpha_i, \beta_i, \gamma_i := \alpha_i \beta_i)$. It will turn out that, combined with an appropriate adaptation of [CDP12], we only need black-box access to our basic functionality of opening linear forms. The (poly)logarithmic communication complexity of the compressed Σ-protocols is directly inherited by our protocol for proving the correctness of multiplication triples.

Second, we consider another scenario that cannot be handled directly with a basic compressed Σ-protocol. Namely, a prover claiming to know k-out-of-n homomorphism preimages. More precisely, for a fixed homomorphism ψ, the prover claims to know k preimages out of n public elements $P_1, \ldots, P_n$ in the codomain of ψ. As before, the prover wishes to convince a verifier of the veracity of this claim without revealing any additional information. In particular, it should remain a secret for which k elements the prover knows the preimages. *Proofs of*

partial knowledge were introduced in [CDS94]. In [CDS94], a k-out-of-n proof of partial knowledge Σ-protocol with *linear* (in n) communication complexity was presented. Unfortunately, their Σ-protocol cannot be compressed. For this reason, we construct a novel Σ-protocol for proving k-out-of-n partial knowledge. More precisely, we deploy a linear secret-sharing scheme to reduce the k-out-of-n scenario to the n-out-of-n scenario. For the n-out-of-n scenario, standard amortization techniques, together with our compression mechanism, apply. Altogether, this results in a k-out-of-n proof of partial knowledge with *logarithmic* (in k and n) communication complexity. Again we only need black-box access to basic compressed Σ-protocols.

These functionality enhancements explain why our basic compressed Σ-protocols do not need any *direct* provision to handle nonlinearity. In both cases, it is the combination of proving knowledge of homomorphism preimages and (arithmetic) secret-sharing that allows for linearizing nonlinear relations.

1.2.1.4 Suitable Cryptographic Platforms

In Chapter 5, we show that compressed Σ-protocols can be instantiated in a variety of cryptographic platforms. First, we consider a discrete logarithm based instantiation that starts from the Pedersen vector commitment scheme. This instantiation allows a prover to open linear forms on committed vectors with a logarithmic communication complexity. Further, we show that this instantiation can be extended to pairing based platforms. In addition, compressed Σ-protocols can be based on a Knowledge-of-Exponent Assumption (KEA), further reducing the communication complexity down to *constant* instead of logarithmic. Note that the KEA is unfalsifiable and its application is not completely without controversy [Nao03; BCP+14]. Moreover, this approach introduces a trusted set-up, which might be undesirable. Finally, we show how to base compressed Σ-protocols on the Strong-RSA and certain lattice assumptions. However, these instantiations are subject to a so called *soundness slack*. An interactive proof is said to have soundness slack if a prover can only convince the verifier of the correctness of a related, but somewhat relaxed, claim. More precisely, in these instantiations the prover claims to know not an arbitrary but a *short* ψ-preimage $\mathbf{x}$ of an element P, i.e., $\psi(\mathbf{x}) = P$ and $\|\mathbf{x}\| \leq \beta$ for some homomorphism ψ and some $\beta \in \mathbb{R}_{\geq 0}$. While such a witness $\mathbf{x}$ is required to convince the verifier, i.e., for completeness, knowledge soundness only guarantees the verifier that the prover knows an input $\tilde{\mathbf{x}}$ such that $\psi(\tilde{\mathbf{x}}) = \zeta \cdot P$ and $\|\tilde{\mathbf{x}}\| \leq \tau \cdot \beta$. The element ζ is referred to as the approximation factor and τ is referred to as the soundness slack. The source of the soundness slack is twofold. First, during the execution of the compressed Σ-protocol, while its dimension decreases, the norm of the preimage increases. Second, the protocol is proven to be knowledge sound by constructing an efficient algorithm capable of extracting a witness from any prover that convinces the verifier with large enough probability. The extraction algorithm contributes to the soundness slack and additionally introduces an approximation factor. In many application scenarios this relaxation is acceptable. However, selection of larger implementation parameters is warranted, causing the communication complexity to be *poly*-logarithmic instead of logarithmic or constant.

1.2.2 Knowledge Extractor Analysis

In Chapter 6, we continue with the security analysis or, more precisely, the knowledge soundness analysis of compressed Σ-protocols. The goal of a compressed Σ-protocol is for a prover to convince a verifier that it knows some secret witness; a prover without knowledge of a witness should not be able to convince the verifier. This security property is formalized by the notion of knowledge soundness. Informally, knowledge soundness states that any prover, that succeeds in convincing the verifier with large enough probability, should be able to efficiently compute a witness satisfying the claimed properties. For this reason, to prove that an interactive proof or argument is knowledge sound, an efficient algorithm capable of extracting a witness from a prover must be constructed. The extractor may invoke the prover arbitrarily many times and also *rewind* the prover to previous states. In this process, the extractor plays the role of the verifier and provides the challenges to the prover. As such the extractor obtains different protocol transcripts, which it uses to compute a witness. The success probability and runtime of the extractor may, and typically do, depend on the success probability of the prover.

It is generally nontrivial to show that an interactive proof admits an extractor and, thus, is knowledge sound. By contrast, the weaker ordinary soundness notion does not require the existence of an extractor. More precisely, soundness only states that the existence of a prover with large enough success probability implies the existence of a witness; it does not require the witness to be efficiently computable. For this reason, it is typically much easier to prove ordinary soundness than knowledge soundness.

In the context of Σ-protocols, the more convenient notion *special-soundness* was introduced [Cra96]. A Σ-protocol is said to be k-special-sound if there exists an efficient algorithm that, on input k accepting protocol transcripts $(a, c_1, z_1), \ldots, (a, c_k, z_k)$ with common first message a and pairwise distinct challenges c_i, outputs a witness. Recall that a Σ-protocol transcript (a, c, z) contains three messages; the first message a is sent from the prover to the verifier, the verifier sends a challenge c sampled uniformly at random from some finite challenge set, and the prover sends the final response z. Subsequently, the verifier decides whether to accept or reject the transcript and thus the prover's claim. We also refer to k-special-soundness as k-out-of-N special-soundness, where N is the size of the verifier's challenge set.

In a k-out-of-N special-sound Σ-protocol, no matter what a dishonest prover does for the first message, if the statement does not admit a witness, there are at most $k-1$ challenges that the dishonest prover can possibly answer. Hence, since the challenges are sampled uniformly at random, a dishonest prover succeeds, on invalid statements without a witness, with probability at most $(k-1)/N$. This already shows that k-out-of-N special-soundness implies ordinary soundness with soundness error $(k-1)/N$. However, in the case of knowledge soundness, this line of reasoning does not apply, since in principle it is possible to answer all the challenges – and indeed the prover can do so if he knows a witness. The challenge is to show that the prover necessarily needs to know a witness to be able answer many challenges; formally, to show the existence of a knowledge extractor.

Although nontrivial to show, it is well known k-out-of-N special-sound

Σ-protocols admit a knowledge extractor. More precisely, k-out-of-N special-soundness implies knowledge soundness with knowledge error $(k-1)/N$, where the knowledge error is the optimal success probability of a dishonest prover. To prove knowledge soundness, it is thus sufficient to show that a Σ-protocol is special-sound, which is usually much easier than proving knowledge soundness directly. Namely, the special-soundness algorithm is given a set of accepting transcripts, whereas the knowledge extractor is only given access to a prover attacking the interactive proof.

Recently, and particularly for the aforementioned compression techniques, natural *multi-round* generalizations of special-soundness have become relevant. For instance, in Chapter 3, we show that compressed Σ-protocols satisfy a multi-round special-soundness notion. In fact, many recently introduced multi-round interactive proofs are special-sound, e.g., [BCC+16; BBB+18; MBK+19; BFS20; BLN+20]. However, known proof techniques, proving that special-soundness implies knowledge soundness, are no longer directly applicable. Namely, the nature of the compression mechanism significantly reduces the efficiency of the corresponding knowledge extractors. More precisely, the efficiencies of naive generalizations of known knowledge extractors scale *exponentially* in the number of rounds of the interactive proof. Several works have attempted to close this gap in the theory of multi-round interactive proofs [BCC+16; HKR19; PLS19; JT20; AL21]. However, their extractors either only provide an asymptotical analysis, requiring for instance exponentially large challenge sets, or their concrete security bounds are non-tight.

1.2.2.1 Special-Sound Multi-Round Interactive Proofs

We provide the first *tight* knowledge soundness analysis for multi-round special-sound interactive proofs and arguments. First, we construct a knowledge extractor that runs in *strict* polynomial time. Unfortunately, this extractor is only applicable to a portion of the full parameter space relevant to our applications. More precisely, it only applies to interactive proofs with a constant number of rounds, whereas Bulletproofs and compressed Σ-protocols have a logarithmic number of rounds. For this reason, we construct a second extractor for special-sound multi-round interactive proofs. In contrast to our first extractor, it runs in *expected* polynomial time. However, it is applicable to the full parameter space and therefore provides a complete solution to the aforementioned knowledge soundness problem. Along the way, we significantly simplify the knowledge soundness analysis of 3-round special-sound interactive proofs.

1.2.2.2 Parallel Repetition

In many occasions, the knowledge error κ, or the success probability of a dishonest prover, is not small enough, and thus needs to be reduced. This can be done generically by repeating the interactive proof in parallel. Naively, one expects that if a prover can cheat in a single instance with probability at most ϵ, then he can cheat at most with probability ϵ^t in a t-fold repetition. However, it is not immediately clear how to prove this – and in general it is actually not true [BIN97; PW07]. The issue is that the prover may potentially make the t runs dependent, and, for example, achieve that with probability ϵ he wins all of them (and thus he wins the parallel repetition) and with probability $1-\epsilon$ he loses all of them. This

situation does not contradict the security of a single run, because in each individual run he only wins with probability ϵ, and so it's not clear how to conclude security of the parallel repetition from the security of a single run (only).

In the case of k-out-of-N special-sound Σ-protocols, the t-fold parallel repetition is easily seen to be ℓ-out-of-N^t special-sound, with $\ell = (k-1)^t + 1$. This immediately implies that the soundness error is $(k-1)^t/N^t$, i.e., the t-fold parallel repetition reduces the soundness error from $\sigma = (k-1)/N$ down to σ^t. However, as before, this line of reasoning does not extend to the stronger notion of knowledge soundness. Namely, the expected runtime of the knowledge extractor for k-out-of-N special-sound interactive proofs is linear in k. Therefore, applying this knowledge extractor to the t-fold parallel repetition results in a runtime that is linear in $\ell = (k-1)^t + 1$, i.e., for $k > 2$ it is exponential in t which is too large. Therefore, to show that t-fold parallel repetition reduces the knowledge error from κ down to κ^t, one cannot merely rely on the special-soundness property. The situation becomes even more complicated when considering multi-round interactive proofs.

Parallel repetition is a fundamental technique in the theory of probabilistic proofs, and its effect on the ordinary soundness error has been studied extensively in many contexts [BIN97; PV07; Hai09; HPW+10; CL10; PV12; CP15]. However, somewhat surprisingly, the effect of parallel repetition on the knowledge error has largely remained unstudied. In this dissertation, we show that t-fold parallel repetition reduces the knowledge error of special-sound multi-round interactive proofs at an optimal rate; from κ down to κ^t. At the core of our results is an alternative, in some sense more fine-grained, measure of quality of a dishonest prover than its success probability, for which we show that it characterizes when knowledge extraction is possible. This new measure then turns out to be very convenient when it comes to analyzing the parallel repetition of such interactive proofs.

Additionally, we provide a novel knowledge extractor that is not only applicable to special-sound interactive proofs, but to the larger class of public-coin interactive proofs. This generality comes at a cost; for public-coin interactive proofs, we show that t-fold parallel repetition reduces the knowledge error from κ down to $\kappa^t + \nu$, for any arbitrary non-negligible ν.

1.2.2.3 The Fiat-Shamir Transformation

Public-coin interactive proofs are typically made non-interactive before being deployed in practice. This can be done by applying the widely used Fiat-Shamir transformation [FS86]. The general idea is to compute the verifier's i-th challenge c_i as a hash of the i-th prover message a_i and (some part of) the previous communication transcript. Recall that, since the interactive proof is public-coin, the i-th challenge c_i is sampled uniformly at random from some finite set. The security of the Fiat-Shamir transformation is usually proven in the idealized *random oracle model* (ROM), where it is assumed that the hash function behaves as a random function. More precisely, in this model the only way to compute the evaluation $H(x)$, of hash function H on input x, is by querying a "random oracle" that has sampled the function table of H uniformly at random. The security of the Fiat-Shamir transformation thus relies on the assumption that the hash

function H behaves as a random oracle in the context of the considered scheme. There exist contrived counterexamples of protocols that are secure in the ROM, but insecure when the random oracle is instantiated with any concrete hash function [CGH04]. However, this transformation is broadly used and, in practice, it appears to withstand all known attacks.

Unfortunately, the Fiat-Shamir transformation introduces a security loss. Namely, in the interactive setting, a dishonest prover must succeed on the challenges it receives from the verifier. By contrast, in the non-interactive setting, a dishonest honest prover may invoke the hash function several times and try multiple sets of challenges when forging a proof. Clearly, the security loss depends on the number of queries Q the prover is allowed to make to the hash function, which is thus modeled as a random oracle.

This also makes the security or extractor analysis of non-interactive Fiat-Shamir transformations significantly more complicated than the analysis of interactive proofs. In the interactive setting, the extractor determines which challenges to provide to the prover. In the non-interactive setting, the extractor does not know for which challenges the prover will output a proof.

The Fiat-Shamir transformation of Σ-protocols has been well-studied. In particular, it is known that the Fiat-Shamir transformation preserves the relevant security properties of a Σ-protocol (in the ROM), with a security loss that is linear in the prover's query complexity Q. However, in general, the security loss of the Fiat-Shamir transformation is *exponential* in the number of rounds of the interactive proof. In fact, it is easy to find interactive proofs that are indeed subject to this exponential security loss.

For multi-round interactive proofs, such as Bulletproofs and compressed Σ-protocols, this is a very unfortunate situation when it comes to choosing concrete security parameters. If one wants to rely on the proven security reduction, one needs to choose a large security parameter for the interactive proof, in order to compensate for exponential security loss, affecting its efficiency; alternatively, one has to give up on proven security and simply assume that the security loss is much milder than what the general bound suggests – indeed, for many interactive proofs, the known attacks do not feature such a large security loss. The latter, of simply assuming the loss to be milder, has become common practice.

This raises the question whether certain (natural) classes of interactive proofs feature a milder security loss. Ideally, the exponential loss appears for contrived examples only. So far, the only positive results in that direction are [CCH+19; GT21]. They show that, in some restricted settings and for certain specific interactive proofs, the Fiat-Shamir security loss is independent of the number of rounds. These results require additional cryptographic assumptions and only apply to a subclass of compressed Σ-protocols.

In this work, we resolve the state-of-affairs by giving both positive and negative answers to the above question. On the positive side, we show that for special-sound interactive proofs the security loss is *independent* of the number of rounds. One can now rely on proven security without choosing overly conservative, and hence inefficient, protocol parameters. On the negative side, we show that for t-fold parallel repetitions of typical special-sound interactive proofs the security loss is *exponential* in the number of rounds. This shows that the exponential security loss

is not only exhibited by contrived interactive proofs.

The extractor analyses of Chapter 6 immediately generalizes from interactive proofs to interactive arguments.

1.2.3 Applications

Finally, in Chapter 7, we discuss two applications of compressed Σ-protocol. First, we consider the circuit satisfiability problem. An interactive proof for circuit satisfiability allows a prover to prove, for any arithmetic circuit $C\colon \mathbb{Z}_q^n \to \mathbb{Z}_q^s$, that it knows a satisfiable input $\mathbf{x} \in \mathbb{Z}_q^n$, i.e., an input $\mathbf{x}$ such that $C(\mathbf{x}) = 0$. With a specialized reduction, we reduce proving the satisfiability of an arithmetic circuit to proving the correctness of a list of multiplication triples. For the latter task, the linearization strategy of Section 4.2 suffices.

Recall that the circuit satisfiability problem is NP-complete, i.e., every problem in NP can be written as a circuit satisfiability problem. However, oftentimes a significant overhead can be avoided by solving a specific problem *directly*, i.e., without reducing it to the standard circuit satisfiability scenario. For instance, by plug-and-play with our basic theory, we construct a *commit-and-prove* zero-knowledge protocol for circuit satisfiability directly. In a commit-and-prove protocol, the prover has already committed to the input vector $\mathbf{x} \in \mathbb{Z}_q^n$ before the start of the protocol, and claims that the committed vector satisfies the constraint $C(\mathbf{x}) = 0$ for some arithmetic circuit C. The naive solution reduces the commit-and-prove scenario to the standard circuit satisfiability scenario. This solution requires the commitment function to be described by a (typically large) arithmetic circuit, and therefore introduces an overhead. We avoid this reduction, and the corresponding overhead, and handle the commit-and-prove scenario directly via plug-and-play with compressed Σ-protocol theory.

Second, we construct a novel transparent and succinct threshold signature scheme (TSS). A k-out-of-n TSS allows any subset of at least k players to sign a message. Our TSS is transparent, because it does not require a trusted setup, and it is succinct, because the size of a threshold signature grows only logarithmically in the total number of players n. A TSS can be constructed immediately, by translating the TSS problem to a circuit satisfiability problem and applying a circuit zero-knowledge protocol. However, we again follow a direct approach and combine a carefully chosen signature scheme with the proofs of partial knowledge of Section 4.3. In contrast to the naive circuit zero-knowledge approach, this direct solution avoids a significant concrete overhead.

These applications demonstrate the advantage of a modular theory for secure algorithmics. There is a set of standard and abstract scenarios that can be handled with basic (compressed) Σ-protocols directly. The basic theory is appended with certain functionality enhancements, increasing its versatility. Application scenarios are handled via a plug-and-play with the abstract building blocks, and by appropriate instantiations thereof. Further, the (security) properties of compound protocols, handling (complex) application scenarios, follow directly from the properties of the basic building blocks. This approach resembles the design principle of Σ-protocol theory, which has now been strengthened with a compression mechanism. We believe this perspective to be useful for handling many more application scenarios in an intuitive manner.

1.3 Publications

This dissertation provides a conceptual framework for the modular design of communication-efficient zero-knowledge proofs. The results from the publications listed below are presented within this framework. There is no one-to-one correspondence between the chapters and the publications. All chapters are based on results that have appeared in different publications and, vice versa, most publications contribute to multiple chapters. In each chapter, it is indicated which publications it is based on. Moreover, this dissertation provides additional details and proofs that have not been published before.

[AFK22] Thomas Attema, Serge Fehr, and Michael Klooß. "Fiat-Shamir Transformation of Multi-Round Interactive Proofs." In: *Theory of Cryptography Conference (TCC)*. Vol. 13747. Lecture Notes in Computer Science. Springer, 2022, pp. 113–142.

[AF22] Thomas Attema and Serge Fehr. "Parallel Repetition of $(k_1, \ldots, k_\mu)$-Special-Sound Multi-Round Interactive Proofs." In: *CRYPTO*. Vol. 13507. Lecture Notes in Computer Science. Springer, 2022, pp. 415–443.

[ACR21] Thomas Attema, Ronald Cramer, and Matthieu Rambaud. "Compressed Σ-Protocols for Bilinear Group Arithmetic Circuits and Application to Logarithmic Transparent Threshold Signatures." In: *ASIACRYPT*. Vol. 13093. Lecture Notes in Computer Science. Springer, 2021, pp. 526–556.

[ACF21] Thomas Attema, Ronald Cramer, and Serge Fehr. "Compressing Proofs of k-out-of-n Partial Knowledge." In: *CRYPTO*. Vol. 12828. Lecture Notes in Computer Science. Springer, 2021, pp. 65–91.

[ACK21] Thomas Attema, Ronald Cramer, and Lisa Kohl. "A Compressed Σ-Protocol Theory for Lattices." In: *CRYPTO*. Vol. 12826. Lecture Notes in Computer Science. Springer, 2021, pp. 549–579.

[AC20] Thomas Attema and Ronald Cramer. "Compressed Σ-Protocol Theory and Practical Application to Plug & Play Secure Algorithmics." In: *CRYPTO*. Vol. 12172. Lecture Notes in Computer Science. Springer, 2020, pp. 513–543.

Further, the author has co-authored the following publications in the field of zero-knowledge proof systems. The results of these publications, although strongly related, are not covered in this dissertation.

[ACC+22] Thomas Attema, Ignacio Cascudo, Ronald Cramer, Ivan Damgård, and Daniel Escudero. "Vector Commitments over Rings and Compressed Σ-Protocols." In: *Theory of Cryptography Conference (TCC)*. Vol. 13747. Lecture Notes in Computer Science. Springer, 2022, pp. 173–202.

[ACX21] Thomas Attema, Ronald Cramer, and Chaoping Xing. "A Note on Short Invertible Ring Elements and Applications to Cyclotomic and Trinomials Number Fields." In: *Mathematical Cryptology* (2021), pp. 45–70.

[ALS20] Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler. "Practical Product Proofs for Lattice Commitments." In: *CRYPTO*. Vol. 12171. Lecture Notes in Computer Science. Springer, 2020, pp. 470–499.

Finally, the author has co-authored the following publications, which are not directly related to compressed Σ-protocol theory. Therefore, also the results of these publications are not covered in this dissertation.

[ACK23] Thomas Attema, Pedro Capitão, and Lisa Kohl. "On Homomorphic Secret Sharing from Polynomial-Modulus LWE." In: *Practice and Theory of Public-Key Cryptography (PKC)*. 2023.

[SMA+22] Gabriele Spini, Emiliano Mancini, Thomas Attema, Mark Abspoel, Jan de Gier, Serge Fehr, Thijs Veugen, Maran van Heesch, Daniël Worm, Andrea De Luca, Ronald Cramer, and Peter M. A. Sloot. "A New Approach to Privacy-Preserving Clinical Decision Support Systems for HIV Treatment." In: *Journal of Medical Systems (JMS)* 46.84 (2022), pp. 1–11.

[ADE+22] Thomas Attema, Vincent Dunning, Maarten H. Everts, and Peter Langenkamp. "Efficient Compiler to Covert Security with Public Verifiability for Honest Majority MPC." In: *Applied Cryptography and Network Security (ACNS)*. Vol. 13269. Lecture Notes in Computer Science. Springer, 2022, pp. 663–683.

[ABN21] Thomas Attema, Joost W. Bosman, and Niels M. P. Neumann. "Optimizing the Decoy-State BB84 QKD Protocol Parameters." In: *Quantum Information Processing (QIP)* 20.154 (2021), pp. 1–26.

[HMA+20] Roy van Houte, Jesse Mulderij, Thomas Attema, Irina Chiscop, and Frank Phillipson. "Mathematical Formulation of Quantum circuit Design Problems in Networks of Quantum Computers." In: *Quantum Information Processing (QIP)* 19.5 (2020), pp. 1–22.

[NHA20] Niels M. P. Neumann, Roy van Houte, and Thomas Attema. "Imperfect Distributed Quantum Phase Estimation." In: *International Conference on Computational Science (ICCS)*. Vol. 12142. Lecture Notes in Computer Science. Springer, 2020, pp. 605–615.

[SHA+19] Alex Sangers, Maran van Heesch, Thomas Attema, Thijs Veugen, Mark Wiggerman, Jan Veldsink, Oscar Bloemen, and Daniël Worm. "Secure Multiparty PageRank Algorithm for Collaborative Fraud Detection." In: *Financial Cryptography and Data Security (FC)*. Vol. 11598. Lecture Notes in Computer Science. Springer, 2019, pp. 605–623.

[AGM+21] Thomas Attema, Nicole Gervasoni, Michiel Marcus, and Gabriele Spini. "Post-Quantum Cryptography: Computational-Hardness Assumptions and Beyond." In: *IACR Cryptology ePrint Archive* (2021). IACR ePrint: 2021/571.

[VAS19] Thijs Veugen, Thomas Attema, and Gabriele Spini. "An Implementation of the Paillier Cryptosystem with Threshold Decryption without a Trusted Dealer." In: *IACR Cryptology ePrint Archive* (2019). IACR ePrint: 2019/1136.

[HAA+19] Maran van Heesch, Niels L. M. van Adrichem, Thomas Attema, and Thijs Veugen. "Towards Quantum-Safe VPNs and Internet." In: *IACR Cryptology ePrint Archive* (2019). IACR ePrint: 2019/1277.

CHAPTER 2

2 Preliminaries

2.1 Basic Notation

We first introduce the basic notation used throughout this dissertation. For a more detailed introduction to concepts such as groups, rings, fields, ideals, modules, homomorphisms, endomorphisms and tensor products, we refer the reader to textbooks such as [Lan02].

By $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$ and $\mathbb{R}_{\geq 0}$ we denote the set of the positive integers, the integers, the real numbers and the nonnegative real numbers, respectively. We write $[a, b] = \{x \in \mathbb{R} : a \leq x \leq b\}$ for the set of real numbers bounded by a and b. For a set S, $2^S = \{A \subseteq S\}$ denotes the powerset of S, containing all subsets of S. Moreover, we adhere to the convention in cryptography by defining $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$ as the ring of integers modulo $q \in \mathbb{Z}$, i.e., $\mathbb{Z}_q$ does not refer to the ring of q-adic integers. Oftentimes q is prime, in which case the ring $\mathbb{Z}_q$ is a field.

The set of bitstrings of length $n \in \mathbb{N}$ is denoted as $\{0,1\}^n$. Moreover, $|x|$ denotes the length of a bitstring x, i.e., $|x| = n$ for all $x \in \{0,1\}^n$. The set of arbitrarily long bitstrings is denoted as $\{0,1\}^* = \cup_{n\in\mathbb{N}}\{0,1\}^n$. Further, vectors $\mathbf{x} = (x_1, \ldots, x_n)$ are written in boldface.

A group $\mathbb{G}$ with group operation $+$ is denoted as $(\mathbb{G}, +)$. If the group operation is clear from context we simply write $\mathbb{G}$. All groups in this work are assumed to be abelian, i.e., the group operation is commutative. The group of homomorphisms from $\mathbb{G}$ to $\mathbb{H}$ is denoted as $\mathrm{Hom}(\mathbb{G}, \mathbb{H})$. Its group operation is defined as the addition of homomorphisms, i.e., $f + g \colon \mathbb{G} \to \mathbb{H}$, $x \mapsto f(x) + g(x)$ for $f, g \in \mathrm{Hom}(\mathbb{G}, \mathbb{H})$. The set $\mathrm{End}(\mathbb{G}) := \mathrm{Hom}(\mathbb{G}, \mathbb{G})$ contains the endomorphisms of $\mathbb{G}$. The composition of homomorphisms defines a second binary operation (multiplication), i.e., $\mathrm{End}(\mathbb{G})$ is a ring.[1]

Sometimes we use multiplicative notation for the group operation instead and write $(\mathbb{H}, \cdot)$. If the group operation is written additively we denote the identity element by 0, and if the group operation is written multiplicatively we denote the identity element by 1.

Recall that an abelian group $(\mathbb{G}, +)$ is a $\mathbb{Z}$-module, i.e., it has a well-defined

[1] Every ring is defined to contain a multiplicative unit, and all ring homomorphisms are defined to map the multiplicative unit to the multiplicative unit.

multiplication by integers operation

$$\cdot\colon \mathbb{Z} \times \mathbb{G} \to \mathbb{G}, \quad (a, g) \mapsto a \cdot g\,.$$

More generally, let $\mathcal{R}$ be a commutative ring, then an $\mathcal{R}$-module is an abelian group $(\mathbb{G}, +)$ together with a ring homomorphism

$$\phi\colon \mathcal{R} \to \mathrm{End}(\mathbb{G}), \quad a \mapsto \phi_a\,.$$

In particular, the multiplication of $g \in \mathbb{G}$ by $a \in \mathcal{R}$ is defined as $a \cdot g := \phi_a(g)$. Further, $M \otimes_{\mathcal{R}} N$ denotes the tensor product of two $\mathcal{R}$-modules M and N.

The exponent q of an abelian group $(\mathbb{G}, +)$ is the smallest positive integer $q \in \mathbb{N}$, such that $q \cdot g = 0$ for all $g \in \mathbb{G}$. If no such integer q exist, we define $q = \infty$. It is easily seen that an abelian group $(\mathbb{G}, +)$ with exponent q is a $\mathbb{Z}_q$-module.

Let now $(\mathbb{G}_1, +)$, $(\mathbb{G}_2, +)$ and $(\mathbb{H}, \cdot)$ be groups of prime order q, hence they are $\mathbb{Z}_q$-modules. Then a mapping $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{H}$ is said to be a *pairing* if it is bilinear, nondegenerate (i.e., e is not identically equal to the identity) and there exists an efficient algorithm to compute e. The tuple $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}, e)$ is also referred to as a *bilinear group*.

Finally, we recall the definitions of negligible and noticeable functions.

Definition 2.1 (Negligible Function)**.** A function $\eta\colon \mathbb{N} \to \mathbb{R}$ is said to be negligible, denoted by $\eta(\lambda) \leq \mathrm{negl}(\lambda)$, if for all $c \in \mathbb{N}$ there exists an $N_c \in \mathbb{N}$ such that $|\eta(\lambda)| \leq \lambda^{-c}$ for all $\lambda \geq N_c$.

Definition 2.2 (Noticeable Function)**.** A function $\eta\colon \mathbb{N} \to \mathbb{R}$ is said to be noticeable if there exists a $c \in \mathbb{N}$ and $N \in \mathbb{N}$ such that $|\eta(\lambda)| \geq \lambda^{-c}$ for all $\lambda \geq N$.

These definitions have straightforward adaptations to functions $\eta\colon \{0,1\}^* \to \mathbb{R}$ taking arbitrary bitstrings as input. For instance, a function $\eta\colon \{0,1\}^* \to \mathbb{R}$ is said to be negligible if for all $c \in \mathbb{N}$ there exists an $N_c \in \mathbb{N}$ such that $|\eta(x)| \leq |x|^{-c}$ for all $|x| \geq N_c$.

2.2 Algorithms

Given a probabilistic algorithm $\mathcal{A}$, we write $y = \mathcal{A}(x; r)$ for the output produced by $\mathcal{A}$ on input x and randomness r. Sometimes the randomness is left implicit, i.e., we write $y \leftarrow \mathcal{A}(x)$ for the process of sampling the bits in r uniformly at random and evaluating $y = \mathcal{A}(x; r)$. The randomness r is also referred to as the random coins or the random tape of $\mathcal{A}$. Note that a function is simply a deterministic algorithm. An algorithm is said to be *efficient* or *polynomial time* if $\mathcal{A}(x)$ runs in a number of steps that is polynomial in the input size $|x|$.

Definition 2.3 (Polynomial Time Algorithm)**.** An algorithm $\mathcal{A}$ is a (strict) polynomial time algorithm if there exists a polynomial $p \in \mathbb{Z}[X]$ such that, for all inputs x and random coins r, $\mathcal{A}(x; r)$ runs in at most $p(|x|)$ steps.

The following weaker, but oftentimes sufficient, notion of efficiency only requires $\mathcal{A}(x)$ to run in a polynomial number of steps on *expectation* over the algorithm's randomness.

Definition 2.4 (Expected Polynomial Time Algorithm)**.** An algorithm $\mathcal{A}$ is an expected polynomial time algorithm if there exists a polynomial $p \in \mathbb{Z}[X]$ such that, for all inputs x, $\mathcal{A}(x)$ runs in an expected number of at most $p(|x|)$ steps, where the expectation is over the randomness r of $\mathcal{A}$.

An algorithm $\mathcal{B}$ is said to have *oracle*, or *black-box*, access to another algorithm $\mathcal{A}$ if $\mathcal{B}$ can invoke $\mathcal{A}$ on arbitrary inputs x and random coins r, which is denoted as $\mathcal{B}^{\mathcal{A}}$. The algorithm $\mathcal{B}$ is also said to be an *oracle algorithm*. If, for all inputs x, random coins r and algorithms $\mathcal{A}$, $\mathcal{B}^{\mathcal{A}}$ invokes $\mathcal{A}$ at most Q times, $\mathcal{B}$ is called a *Q-query* oracle algorithm.

2.3 Arithmetic Circuits

The main model of computation used in this dissertation is the arithmetic circuit model. Arithmetic circuits model the evaluation of multivariate polynomials $f(X_1, \ldots, X_n)$ defined over a finite field $\mathbb{F}$. They express a polynomial in terms of the basic arithmetic operations: addition and multiplication. More precisely, an arithmetic circuit is a directed acyclic graph. Its nodes are referred to as gates and its edges as wires. The gates with indegree 0 are called the input gates. Input gates have unbounded outdegree and are assigned a constant $a \in \mathbb{F}$ or a variable X_i. The remaining gates are addition or multiplication gates. They have indegree 2 and unbounded outdegree. As such, all wires naturally correspond to a multivariate polynomial in $\mathbb{F}[X_1, \ldots, X_n]$, where n is the number of variable input gates. An arithmetic circuit corresponding to the polynomial $f(X_1, \ldots, X_n)$ has a unique output gate with outdegree 0. Slightly abusing terminology, we also allow an arithmetic circuit C to have multiple output gates. In this case the circuit C corresponds to a vector of polynomials $(f_1, \ldots, f_s)$.

The evaluation of an arithmetic circuit entails assigning values to the n variables $X_1, \ldots, X_n$ and computing all wire values. For this reason, an arithmetic circuit with s output gates can also be viewed as a mapping $C\colon \mathbb{F}^n \to \mathbb{F}^s$.

The size $|C|$ of an arithmetic circuit C is the number of wires it contains. It is a measure for its computational complexity. There are many arithmetic circuits corresponding to the same function $f\colon \mathbb{F}^n \to \mathbb{F}^s$. A natural question is therefore to find the smallest arithmetic circuit computing a given function.

The *circuit satisfiability problem* asks to decide whether a given arithmetic circuit $C\colon \mathbb{F}^n \to \mathbb{F}$ admits a satisfiable input $\mathbf{x} \in \mathbb{F}^n$, i.e., an input $\mathbf{x}$ such that $C(\mathbf{x}) = 0$. The circuit satisfiability problem is NP-complete,[2] i.e., every problem in NP can be written as a circuit satisfiability problem, demonstrating its versatility.

2.4 Probability Distributions

Let us now recall some basic discrete probability theory. In this work, we will not require continuous probability theory.

[2] Recall that NP denotes the class of problems that admit an efficiently verifiable solution.

Definition 2.5 (Discrete Probability Space)**.** A discrete probability space is a tuple (Ω, p), containing a countable sample space Ω and a probability mass function $p\colon \Omega \to [0,1]$ such that $\sum_{\omega\in\Omega} p(\omega) = 1$. A subset $E \subseteq \Omega$ is called an event. Every event is associated to a probability via the probability measure

$$\Pr\colon 2^{\Omega} \to [0,1], \quad E \mapsto \sum_{\omega\in E} p(\omega)\,.$$

Definition 2.6 (Random Variable)**.** A random variable is a function $X\colon \Omega \to \mathcal{X}$ for some nonempty set $\mathcal{X}$. Moreover, the probability distribution of X is the function

$$D_X\colon \mathcal{X} \to [0,1], \quad x \mapsto \Pr(X = x) := \sum_{\omega\in X^{-1}(x)} p(\omega)\,.$$

For any $x \in \mathcal{X}$ and $C \subseteq \mathcal{X}$, the events $X^{-1}(x) \subseteq \Omega$ and $X^{-1}(C) \subseteq \Omega$ are simply denoted as $X = x$ and $X \in C$, respectively. The support of a random variable is $\operatorname{supp}(X) = \{x \in \mathcal{X} : \Pr(X = x) > 0\}$. Further, X is said to be uniformly distributed over $\mathcal{X}$ if $\mathcal{X}$ is a finite set and $\Pr(X = x) = 1/|\mathcal{X}|$ for all $x \in \mathcal{X}$. Sampling an element x from a distribution D_X is denoted as $x \leftarrow_R D_X$, i.e., $\Pr(x = y\colon x \leftarrow_R D_X) = \Pr(X = y)$ for all $y \in \mathcal{X}$. If D_X is the uniform distribution over some finite set $\mathcal{X}$, we also write $x \leftarrow_R \mathcal{X}$ instead of $x \leftarrow_R D_X$. For an algorithm $\mathcal{A}\colon \mathcal{X} \to \mathcal{Y}$, $\mathcal{A}(X)$ denotes the random variable with $\Pr(\mathcal{A}(X) = y) = \Pr\big(\mathcal{A}(x) = y : x \leftarrow_R D_X\big)$, where the probability is also over the randomness of $\mathcal{A}$.

Definition 2.7 (Statistical Distance)**.** The statistical distance between two random variables $X_0, X_1\colon \Omega \to \mathcal{X}$ is defined as

$$\Delta(X_0, X_1) = \frac{1}{2} \sum_{x\in\mathcal{X}} |\Pr(X_0 = x) - \Pr(X_1 = x)|\,.$$

The statistical distance is also called the *total variation distance*.

Towards proving the security of cryptographic protocols, we are often interested in algorithms $\mathcal{D}$ aiming to distinguish two random variables X_0 and X_1. For instance, the inability of an adversary to distinguish the encryption of a secret message from a uniformly random bitstring proves the security of the considered encryption scheme.

In order to quantify how well an algorithm $\mathcal{D}$ can distinguish two random variables X_0 and X_1 let us consider the following distinguishing game. First, a bit $b \leftarrow_R \{0,1\}$ is sampled uniformly at random. Second, an element $x \leftarrow_R D_{X_b}$ is sampled from the distribution of X_b. Finally, the algorithm $\mathcal{D}$, on input x, outputs a bit $b' \in \{0,1\}$. The algorithm $\mathcal{D}$ wins the distinguishing game if $b = b'$. For this reason, a probabilistic algorithm that always outputs a bit is also called a *distinguisher*.

The advantage $\operatorname{Adv}_{\mathcal{D}}(X_0, X_1)$ of a distinguisher now measures how well $\mathcal{D}$ succeeds in winning this game. For instance, the advantage equals 1 if the distinguisher always wins, and it equals 0 if $\mathcal{D}$ ignores the input x and outputs a random bit b', thereby always winning with probability 1/2.

Definition 2.8 (Advantage of a Distinguisher)**.** Let $X_0, X_1 \colon \Omega \to \mathcal{X}$ be two random variables and let $\mathcal{D} \colon \mathcal{X} \to \{0,1\}$ be a (probabilistic) distinguisher. Then, the advantage of $\mathcal{D}$ in distinguishing X_0 and X_1 is

$$\mathrm{Adv}_{\mathcal{D}}(X_0, X_1) := \left|\Pr\big(\mathcal{D}(X_0) = 0\big) - \Pr\big(\mathcal{D}(X_1) = 0\big)\right| .$$

Moreover, the advantage of a class of distinguishers $\mathcal{F}$ is

$$\mathrm{Adv}_{\mathcal{F}}(X_0, X_1) := \sup_{\mathcal{D} \in \mathcal{F}} \mathrm{Adv}_{\mathcal{D}}(X_0, X_1) .$$

The following lemma shows that the distinguishing advantage of a family of distinguishers is closely related to the statistical distance.

Lemma 2.1. *Let $X_0, X_1 \colon \Omega \to \mathcal{X}$ be random variables. Then*

$$\Delta(X_0, X_1) = \sup_{\mathcal{D}} \mathrm{Adv}_{\mathcal{D}}(X_0, X_1) ,$$

where the supremum is over all distinguishers $\mathcal{D}$.

Proof. See [CDN15, page 20]. □

We are now ready to define what it means for two families of random variables to be statistically or computationally indistinguishable.

Definition 2.9 (Statistical Indistinguishability)**.** Two families $\{X_s\}_{s \in S}$ and $\{Y_s\}_{s \in S}$ of random variables, indexed by a set of bitstrings $S \subseteq \{0,1\}^*$, are said to be statistically indistinguishable if the function

$$\Delta(s) := \Delta(X_s, Y_s)$$

is negligible in $|s|$. If $\Delta(s) = 0$ for all $s \in S$, $\{X_s\}_{s \in S}$ and $\{Y_s\}_{s \in S}$ are said to be perfectly indistinguishable.

Definition 2.10 (Computational Indistinguishability)**.** Let $\mathcal{F}$ be the class of polynomial time distinguishers. Two families $\{X_s\}_{s \in S}$ and $\{Y_s\}_{s \in S}$ of random variables, indexed by a set of bitstrings $S \subseteq \{0,1\}^*$, are said to be computationally indistinguishable if

$$\Delta(s) := \mathrm{Adv}_{\mathcal{F}}(X_s, X_s)$$

is negligible in $|s|$.

2.4.1 Geometric Distribution

A random variable B with two distinct possible outcomes, denoted 0 (failure) and 1 (success), is said to follow a Bernoulli distribution with parameter $p = \Pr(B = 1)$. Sampling from a Bernoulli distribution is also referred to as running a Bernoulli trial. The probability distribution of the number X of independent and identical Bernoulli trials needed to obtain a success is called the geometric distribution with parameter $p = \Pr(X = 1)$. In this case $\Pr(X = k) = (1-p)^{k-1}p$ for all $k \in \mathbb{N}$ and we write $X \sim \mathrm{Geo}(p)$. For two independent geometric distributions we have the following lemma.

Lemma 2.2. *Let $X \sim \mathrm{Geo}(p)$ and $Y \sim \mathrm{Geo}(q)$ be independently distributed. Then,*

$$\Pr(X \leq Y) = \frac{p}{p+q-pq} \geq \frac{p}{p+q}.$$

Proof. It holds that

$$\begin{aligned}\Pr(X \leq Y) &= \sum_{k=1}^{\infty} \Pr(X = k) \Pr(Y \geq k) = \sum_{k=1}^{\infty} (1-p)^{k-1} p \cdot (1-q)^{k-1} \\ &= p \sum_{\ell=0}^{\infty} (1-p)^{\ell} (1-q)^{\ell} = \frac{p}{1-(1-p)(1-q)} \\ &= \frac{p}{p+q-pq} \geq \frac{p}{p+q}\,,\end{aligned}$$

which completes the proof of the lemma. □

2.4.2 Negative Hypergeometric Distribution

Consider a bucket containing ℓ green balls and $N-\ell$ red balls, i.e., a total of N balls. In the negative hypergeometric experiment, balls are drawn uniformly at random from this bucket, without replacement, until k green balls have been found, or until the bucket is empty. The number of red balls X drawn in this experiment is said to have a *negative hypergeometric distribution* with parameters N, ℓ, k, which is denoted by $X \sim \mathrm{NHG}(N, \ell, k)$.

Lemma 2.3 (Negative Hypergeometric Distribution)**.** *Let $N, \ell, k \in \mathbb{N}$ with $\ell, k \leq N$, and let $X \sim \mathrm{NHG}(N, \ell, k)$. Then*

$$\mathbb{E}[X] \leq k \frac{N-\ell}{\ell+1}\,.$$

Proof. If $\ell < k$, it clearly holds that $\Pr(X = N-\ell) = 1$. Hence, in this case, $\mathbb{E}[X] = N - \ell \leq k\frac{N-\ell}{\ell+1}$, which proves the claim.

So let us now consider the case $\ell \geq k$. Then, for all $0 \leq x \leq N-\ell$,

$$\Pr(X = x) = \frac{\binom{x+k-1}{x}\binom{N-x-k}{N-\ell-x}}{\binom{N}{N-\ell}}\,.$$

Hence,

$$\begin{aligned}\mathbb{E}[X] &= \sum_{x=0}^{N-\ell} \Pr(X = x) \cdot x = \sum_{x=1}^{N-\ell} x \frac{\binom{x+k-1}{x}\binom{N-x-k}{N-\ell-x}}{\binom{N}{N-\ell}} \\ &= k\frac{N-\ell}{\ell+1} \sum_{x=1}^{N-\ell} \frac{\frac{x}{k}\binom{x+k-1}{x}\binom{N-x-k}{N-\ell-x}}{\frac{N-\ell}{\ell+1}\binom{N}{N-\ell}} = k\frac{N-\ell}{\ell+1} \sum_{x=1}^{N-\ell} \frac{\binom{x+k-1}{x-1}\binom{N-x-k}{N-\ell-x}}{\binom{N}{N-\ell-1}} \\ &= k\frac{N-\ell}{\ell+1} \sum_{x=1}^{N-\ell} \Pr(Y = x-1) = k\frac{N-\ell}{\ell+1}\,,\end{aligned}$$

where $Y \sim \mathrm{NHG}(N, \ell+1, k-1)$. This completes the proof of the lemma. □

Remark 2.1. Typically, negative hypergeometric experiments are restricted to the nontrivial case $\ell \geq k$. For reasons to become clear later, we also allow parameter choices with $\ell < k$ resulting in a trivial negative hypergeometric experiment in which all balls are always drawn.

Remark 2.2. The above negative hypergeometric experiment has a straightforward generalization to buckets with balls of more than 2 colors. Namely, say the bucket contains ℓ green balls and m_i balls of color i for $1 \leq i \leq M$. The experiment proceeds as before, i.e., drawing until either k green balls have been found or the bucket is empty. Let X_i be the number of balls of color i that are drawn in this experiment. Then $X_i \sim \mathrm{NHG}(\ell + m_i, \ell, k)$ for all i. To see this, simply run the generalized negative hypergeometric experiment without counting the balls that are neither green nor of color i.

2.5 Commitment Schemes

Commitment schemes allow a party, also referred to as a prover, to commit to (secret) input data. When a prover has made a commitment, the input data can no longer be changed, i.e., the commitment is *binding*. Moreover, the commitment itself does not reveal anything about the input data, i.e., it is *hiding*. Finally, at some later point in time, the prover can reveal his input data and prove that this was indeed the data it committed to before. This is called *opening* a commitment. Commitment schemes are one the most important building blocks in cryptography.

The following gives a formal definition for commitment schemes. The binding and hiding properties are not incorporated in this definition; we consider these as desirable security properties.

Definition 2.11 (Commitment Scheme)**.** A commitment scheme is defined by a probabilistic polynomial time setup algorithm Setup, which takes as input the (unary encoding of) a security parameter[3] λ and outputs a public key $\mathsf{pk} \leftarrow \textsc{Setup}(1^\lambda)$. Every public key defines a message set $\mathcal{M}_{\mathsf{pk}}$, a randomness set $\mathsf{Rand}_{\mathsf{pk}}$, a commitment set $\mathcal{C}_{\mathsf{pk}}$ and a deterministic function

$$\textsc{com}_{\mathsf{pk}} \colon \mathcal{M}_{\mathsf{pk}} \times \mathsf{Rand}_{\mathsf{pk}} \to \mathcal{C}_{\mathsf{pk}}, \quad (m;\gamma) \mapsto \textsc{com}_{\mathsf{pk}}(m;\gamma)\,.$$

To commit to a message $m \in \mathcal{M}_{\mathsf{pk}}$, a prover samples $\gamma \leftarrow_R \mathsf{Rand}_{\mathsf{pk}}$ uniformly at random and outputs the commitment $P = \textsc{com}_{\mathsf{pk}}(m;\gamma)$. A commitment is opened by revealing the message m together with the commitment randomness γ. An opening $(m;\gamma)$ of a commitment P is verified by checking that $\textsc{com}_{\mathsf{pk}}(m;\gamma) = P$. Let us now formally define what it means for a commitment scheme to be binding and hiding.

Definition 2.12 (Binding Commitment Scheme)**.** A commitment scheme defined by the setup algorithm Setup is (statistically) *binding* if, for every probabilistic

[3]The security parameter controls the expected amount of security a cryptographic primitive offers, i.e., there exists a monotone function f such that the cost of breaking the primitive instantiated with security parameter λ is at least $f(\lambda)$. Typically, we require the function f to grow faster than any polynomial $p(X) \in \mathbb{Z}[X]$.

algorithm $\mathcal{A}$,

$$\Pr\left(m_0 \neq m_1 \wedge P_0 = P_1 \;\middle|\; \begin{array}{c} \mathsf{pk} \leftarrow \textsc{Setup}\left(1^\lambda\right) \\ (m_0, \gamma_0, m_1, \gamma_1) \leftarrow \mathcal{A}(\mathsf{pk}) \\ P_0 = \textsc{com}_{\mathsf{pk}}(m_0; \gamma_0) \\ P_1 = \textsc{com}_{\mathsf{pk}}(m_1; \gamma_1) \end{array}\right) \leq \mathrm{negl}(\lambda)\,.$$

If the above probability equals 0, the commitment scheme is said to be *perfectly* binding. If the above only holds for polynomial time algorithms $\mathcal{A}$, the commitment scheme is said to be *computationally* binding.

Definition 2.13 (Hiding Commitment Scheme)**.** A commitment scheme defined by the setup algorithm SETUP is (statistically) *hiding* if, for every pair of probabilistic algorithms $(\mathcal{A}_1, \mathcal{A}_2)$,

$$\left|\Pr\left(\mathcal{A}_2(\mathsf{pk}, P) = b \;\middle|\; \begin{array}{c} \mathsf{pk} \leftarrow \textsc{Setup}\left(1^\lambda\right) \\ (m_0, m_1) \leftarrow \mathcal{A}_1(\mathsf{pk}) \\ b \leftarrow_R \{0,1\}, \gamma \leftarrow_R \mathsf{Rand}_{\mathsf{pk}} \\ P = \textsc{com}_{\mathsf{pk}}(m_b; \gamma) \end{array}\right) - \frac{1}{2}\right| \leq \mathrm{negl}(\lambda)\,.$$

If the above probability equals $1/2$, the commitment scheme is said to be *perfectly* hiding. If the above only holds for polynomial time algorithm pairs $(\mathcal{A}_1, \mathcal{A}_2)$, the commitment scheme is said to be *computationally* hiding.

Note that if the commitment scheme is perfectly hiding, then $\textsc{com}(m;\gamma)$ and $\textsc{com}(m';\gamma)$ are identically distributed for all $m, m' \in \mathcal{M}_{\mathsf{pk}}$, where $\gamma \leftarrow_R \mathsf{Rand}_{\mathsf{pk}}$ is uniformly distributed.

A commitment scheme is said to be *homomorphic* if, for all public keys pk, the sets $\mathcal{M}_{\mathsf{pk}}$, $\mathsf{Rand}_{\mathsf{pk}}$ and $\mathcal{C}_{\mathsf{pk}}$ are groups, and the function $\textsc{com}_{\mathsf{pk}} \colon \mathcal{M}_{\mathsf{pk}} \times \mathsf{Rand}_{\mathsf{pk}} \to \mathcal{C}_{\mathsf{pk}}$ is a group homomorphism. Typically, the group operations in $\mathcal{M}_{\mathsf{pk}}$ and $\mathsf{Rand}_{\mathsf{pk}}$ are written additively and the group operation in $\mathcal{C}_{\mathsf{pk}}$ is written multiplicatively.

We say that a commitment scheme is a *vector* commitment scheme if the setup algorithm additionally takes as input a dimension n and, for every public key $\mathsf{pk} \leftarrow \textsc{Setup}(1^\lambda, n)$, the message set is an n-fold Cartesian product $\mathcal{M}_{\mathsf{pk}}^n$, i.e.,

$$\textsc{com}_{\mathsf{pk}} \colon \mathcal{M}_{\mathsf{pk}}^n \times \mathsf{Rand}_{\mathsf{pk}} \to \mathcal{C}_{\mathsf{pk}}\,.$$

A vector commitment scheme thus allows a prover to commit to vectors of arbitrary length n. If the commitment scheme is homomorphic and $n' < n$, we also write $\textsc{com}_{\mathsf{pk}}(m_1, \ldots, m_{n'}; \gamma) := \textsc{com}_{\mathsf{pk}}(m_1, \ldots, m_{n'}, 0, \ldots, 0; \gamma)$ where $(m_1, \ldots, m_{n'}, 0, \ldots, 0) \in \mathcal{M}_{\mathsf{pk}}^n$. Sometimes, if $n' > n$, we abuse notation and still write $\textsc{com}_{\mathsf{pk}}(m_1, \ldots, m_{n'}; \gamma)$. In this case, we implicitly assume that the commitment scheme was actually instantiated with dimension at least n'.

A vector commitment scheme is said to be *compact* if the size of a commitment is constant in n. Moreover, it is said to be *compressing* if the size of a commitment is sublinear in n, i.e., the size of a commitment grows sublinearly in the dimension n of the committed vector. In particular, any compact vector commitment scheme is compressing. It is easily seen that a compressing commitment scheme can be at most computationally binding.

2.6 Group-Based Cryptographic Assumptions

The security of many cryptographic protocols is based on the intractability of certain computational problems. In this section, we introduce and formalize the group-based cryptographic hardness assumptions that are used in this dissertation.

One of the best-known computational problems used in cryptography is the *discrete logarithm* (DL) problem. Let $(\mathbb{G}, \cdot)$ be a group of prime order q and let $g \neq 1$. Then g generates $\mathbb{G}$, i.e., for all $h \in \mathbb{G}$ there exists an $x \in \mathbb{Z}_q$ such that $g^x = h$. The exponent x is also called the discrete logarithm of h with respect to generator g. The DL problem asks to find x given g and h. In suitable groups, this problem is assumed to be intractable, i.e., polynomial-time algorithms succeed with at most negligible probability in solving this problem. The following definition formalizes the discrete logarithm assumption.

Definition 2.14 (Discrete Logarithm Assumption)**.** Let $\mathcal{G}$ be a probabilistic polynomial time algorithm that, on input a security parameter λ, outputs a prime q, a group $(\mathbb{G}, \cdot)$ of order q and a generator g of $\mathbb{G}$. The *discrete logarithm* (DL) assumption holds for $\mathcal{G}$ if for all probabilistic polynomial time algorithms $\mathcal{A}$

$$\Pr\big(h = g^x : (q, \mathbb{G}, g) \leftarrow \mathcal{G}(1^\lambda) \wedge h \leftarrow_R \mathbb{G} \wedge x \leftarrow \mathcal{A}(q, \mathbb{G}, g, h)\big) \leq \mathrm{negl}(\lambda)\,.$$

The second group based hardness assumption is the *decisional Diffie-Hellman* (DDH) assumption [Bon98]. This assumption states that it is hard for an adversary to distinguish triples of the form (g^x, g^y, g^{xy}) from those of the form (g^x, g^y, g^z), where $x, y, z \leftarrow_R \mathbb{Z}_q$ are sampled uniformly at random. The DL assumption is implied by the DDH assumption, i.e., if the DDH assumption holds, so does the DL assumption.

Definition 2.15 (Decisional Diffie-Hellman Assumption)**.** Let $\mathcal{G}$ be a probabilistic polynomial time algorithm that, on input a security parameter λ, outputs a prime q, a group $(\mathbb{G}, \cdot)$ of order q and a generator g of $\mathbb{G}$. The *decisional Diffie-Hellman* (DDH) assumption holds for $\mathcal{G}$ if for all probabilistic polynomial time algorithms $\mathcal{A}$

$$\big|\Pr\big(\mathcal{A}(q, \mathbb{G}, g, g^x, g^y, g^{xy}) = 1\big) - \Pr\big(\mathcal{A}(q, \mathbb{G}, g, g^x, g^y, g^z) = 1\big)\big| \leq \mathrm{negl}(\lambda)\,,$$

where the probabilities are over $(q, \mathbb{G}, g) \leftarrow \mathcal{G}(1^\lambda)$, $x, y, z \leftarrow_R \mathbb{Z}_q$ and $\mathcal{A}$'s randomness.

We also refer to the algorithm $\mathcal{G}$ in definitions 2.14 and 2.15 as a prime order group generator. In some settings, the algorithm $\mathcal{G}$ actually outputs a bilinear group $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}, e)$. In this case, we must specify in which of the groups $\mathbb{G}_1$, $\mathbb{G}_2$ or $\mathbb{H}$ the DL or DDH assumption holds. In particular, if the DDH assumption holds in both $\mathbb{G}_1$ and $\mathbb{G}_2$, we say that the *symmetrical external Diffie-Hellman* (SXDH) assumption [BGM+05] holds. It is easily seen that, for a bilinear group $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}, e)$, the existence of an efficiently computable isomorphism $\psi\colon \mathbb{G}_1 \to \mathbb{G}_2$ contradicts the DDH assumption in $\mathbb{G}_1$, and vice-versa the existence of an efficiently computable isomorphism $\psi\colon \mathbb{G}_2 \to \mathbb{G}_1$ contradicts the DDH assumption in $\mathbb{G}_2$. Hence, the SXDH assumption can only hold if there do not exist efficiently

computable isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$. This class of bilinear groups (or pairings) is also referred to as Type III [GPS08].

The product N of two primes p and q is called an *RSA-modulus*. It is assumed to be hard to find the prime factors p and q of N. Further, the group $\mathbb{Z}_N^*$ of multiplicative units modulo N, also referred to as an *RSA-group*, has cardinality $\phi(N) = (p-1)(q-1)$. From this it follows that, without knowledge of p and q, it is intractable to find the order of the group $\mathbb{Z}_N^*$; if not, one could efficiently factor RSA-moduli. For this reason, the group $\mathbb{Z}_N^*$ is also said to be of *hidden order*.

There exists a broad variety of hardness assumptions based on groups with hidden order; we introduce two of them. First, the *strong-RSA* assumption [BP97] states that it is hard to compute nontrivial roots in a group $\mathbb{G}$ with hidden order. Second, the *hidden order* assumption states that it is hard to find the order of group elements $g \leftarrow_R \mathbb{G}$ sampled uniformly at random. The hidden order assumption is implied by the strong-RSA assumption.

A disadvantage of RSA-groups is that their order is only hidden from parties that are oblivious to the prime factors p and q of N. In practice, this means that the RSA-group typically has to be generated by a trusted dealer. An alternative candidate for groups of hidden order are class groups of imaginary quadratic number fields [Wes19; BFS20; BHR+21]. Class groups can be generated in a transparent manner and thus do not require a trusted dealer.

Definition 2.16 (Strong-RSA Assumption)**.** Let $\mathcal{G}$ be a probabilistic polynomial time algorithm that, on input a security parameter λ, outputs a group $(\mathbb{G}, \cdot)$ (with hidden order). The *strong-RSA* assumption holds for $\mathcal{G}$ if for all probabilistic polynomial time algorithms $\mathcal{A}$,

$$\Pr\big(g = P^x \wedge x > 1 : \mathbb{G} \leftarrow \mathcal{G}(1^\lambda) \wedge g \leftarrow_R \mathbb{G} \wedge (P, x) \leftarrow \mathcal{A}(\mathbb{G}, g)\big) \leq \mathrm{negl}(\lambda)\,.$$

Definition 2.17 (Hidden Order Assumption)**.** Let $\mathcal{G}$ be a probabilistic polynomial time algorithm that, on input a security parameter λ, outputs a group $(\mathbb{G}, \cdot)$ (with hidden order). The *hidden order* assumption holds for $\mathcal{G}$ if for all probabilistic polynomial time algorithms $\mathcal{A}$,

$$\Pr\big(g^x = 1 \wedge x > 1 : \mathbb{G} \leftarrow \mathcal{G}(1^\lambda) \wedge g \leftarrow_R \mathbb{G} \wedge x \leftarrow \mathcal{A}(\mathbb{G}, g)\big) \leq \mathrm{negl}(\lambda)\,.$$

2.7 Lattices and Lattice Problems

A disadvantage of the group-based assumptions of the previous section is that, once available, a quantum computer will be able to solve the corresponding computational problems efficiently [Sho94]. Therefore, cryptographic primitives based on these assumptions will in general not be secure against adversaries with access to a quantum computer. By contrast, *post-quantum cryptography* studies the design of cryptographic primitives based on computational problems that are intractable even for quantum adversaries. One of the most promising areas in this field of research is *lattice-based* cryptography, where the underlying problems are so-called lattice problems. In this section, we introduce a number of variants of the short integer solution (SIS) problem.

A lattice Λ is a discrete additive subgroup of $\mathbb{R}^m$. The lattice Λ is said to be q-ary if $q\mathbb{Z}^m \subseteq \Lambda \subseteq \mathbb{Z}^m$. For instance, for any $A \in \mathbb{Z}_q^{k\times m}$ the sets

$$\begin{aligned}\Lambda_q(A) &= \{\mathbf{y} \in \mathbb{Z}^k : \exists \mathbf{x} \in \mathbb{Z}^m \ A\mathbf{x} = \mathbf{y} \mod q\} \quad \text{and}\\ \Lambda_q^\perp(A) &= \{\mathbf{x} \in \mathbb{Z}^m : A\mathbf{x} = 0 \mod q\}\end{aligned}$$

are q-ary lattices in $\mathbb{Z}^k$ and $\mathbb{Z}^m$ respectively. Finding a nonzero and "short" element in the lattice $\Lambda_q^\perp(A) \subseteq \mathbb{Z}^m$ is referred to as the *Short Integer Solution* (SIS) problem [Ajt96].

Definition 2.18 ($\mathrm{SIS}_{q,k,m,\beta}$-Problem [Ajt96])**.** The $\mathrm{SIS}_{q,k,m,\beta}$-problem is defined as follows: Given a matrix $A \leftarrow_R \mathbb{Z}_q^{k\times m}$ sampled uniformly at random, find a nonzero vector $\mathbf{s} \in \mathbb{Z}^m$, such that $A\mathbf{s} = 0 \mod q$ and $\|\mathbf{s}\|_2 \leq \beta$.

Let $\mathcal{R} = \mathbb{Z}[X]/f(X)$ for a monic[4] polynomial $f(X)$ of degree d. The coefficient embedding

$$\psi\colon \mathcal{R} \to \mathbb{Z}^d, \quad \sum_{i=1}^{d} a_i X^{i-1} \mapsto (a_1, \ldots, a_d)$$

is a group isomorphism. Hence, $\mathcal{R}$ corresponds to the lattice $\mathbb{Z}^d$. Moreover, every ideal $I \subseteq \mathcal{R}$ corresponds to a sublattice $\psi(I) \subseteq \mathbb{Z}^d$. The lattice $\psi(I)$ is said to be a *structured* or *ideal* lattice.

For $q \in \mathbb{N}$, we write $\mathcal{R}_q = \mathcal{R}/q\mathcal{R} = \mathbb{Z}_q[X]/\big(f(X)\big)$. Further, to $a_1, \ldots, a_m \in \mathcal{R}_q$, we associate the following q-ary lattice

$$\Lambda_q^\perp(a_1, \ldots, a_m) = \{\mathbf{x} \in \mathcal{R}^m : \textstyle\sum_{i=1}^m a_i x_i = 0 \mod q\}.$$

The coefficient embedding ψ also equips the rings $\mathcal{R}$ and $\mathcal{R}^m$ with a geometry. More precisely, we define $\|\mathbf{x}\| = \|\psi(\mathbf{x})\|$ for any $\mathbf{x} \in \mathcal{R}^m$ and any norm $\|\cdot\|$ on $\mathbb{Z}^{dm}$. Finding a nonzero and short element in the lattice $\Lambda_q^\perp(a_1, \ldots, a_m) \subseteq \mathcal{R}^m$ is referred to as the *Ring-SIS* (RSIS) problem [PR06; LM06].

Definition 2.19 ($\mathrm{RSIS}_{q,m,\beta}$-Problem [Ajt96])**.** Let $\mathcal{R} = \mathbb{Z}[X]/f(X)$ for a monic polynomial $f(X)$. The $\mathrm{RSIS}_{q,m,\beta}$-problem over $\mathcal{R}$ is defined as follows: Given $a_1, \ldots, a_m \leftarrow_R \mathcal{R}_q$ sampled uniformly at random, find a nonzero vector $\mathbf{s} = (s_1, \ldots, s_m) \in \mathcal{R}^m$, such that $\sum_{i=1}^m a_i s_i = 0 \mod q$ and $\|\mathbf{s}\|_2 \leq \beta$.

For $A \in \mathcal{R}_q^{k\times m}$, $\Lambda_q^\perp(A) = \{\mathbf{x} \in \mathcal{R}^m : A\mathbf{x} = 0 \mod q\}$ corresponds to a q-ary sublattice of $\mathbb{Z}^{dm}$. The set $\Lambda_q^\perp(A) \subseteq \mathcal{R}^m$ is a finitely generated $\mathcal{R}$-module. For this reason, the corresponding lattice is also called a *module lattice*. Finding a nonzero and short element in a lattice $\Lambda_q^\perp(A)$, for $A \in \mathcal{R}_q^{k\times m}$, is referred to as the *Module-SIS* (MSIS) problem [LS15]. The MSIS-problem is a generalization of both the SIS- and the RSIS-problem. It is assumed to be intractable, even for quantum computers.

Definition 2.20 ($\mathrm{MSIS}_{q,k,m,\beta}$-Problem [LS15])**.** Let $\mathcal{R} = \mathbb{Z}[X]/f(X)$ for a monic polynomial $f(X)$. The $\mathrm{MSIS}_{q,k,m,\beta}$-problem over $\mathcal{R}$ is defined as follows: Given a matrix $A \leftarrow_R \mathcal{R}_q^{k\times m}$ sampled uniformly at random, find a nonzero vector $\mathbf{s} \in \mathcal{R}^m$, such that $A\mathbf{s} = 0 \mod q$ and $\|\mathbf{s}\|_2 \leq \beta$.

[4] Recall that a polynomial $f(X) = \sum_{i=0}^n a_i X^n$ is said to be monic if its leading coefficient a_n equals 1.

The *Gaussian heuristic* states that the length $\lambda_1\big(\Lambda_q^\perp(A)\big) = \|\mathbf{s}\|_2 \in \mathbb{R}_{\geq 0}$ of the shortest vector $\mathbf{s}$ of a q-ary lattice $\Lambda_q^\perp(A)$, for $A \in \mathcal{R}_q^{k\times m}$, is approximately equal to $\sqrt{m/(2\pi e)}q^{k/m}$ [MR09]. The quality of an algorithm χ for finding short vectors in a lattice can be characterized by its root Hermite factor δ, which is defined such that χ is expected to output basis vectors $\mathbf{s}$ with

$$\|\mathbf{s}\|_2 \approx \min(q, \delta^{dm} q^{k/m})\,. \tag{2.1}$$

In particular, smaller values of δ require better algorithms or a longer runtime. Given the current state-of-the-art, a (quantum) algorithm with $\delta \approx 1.0045$ is assumed to take at least 2^{128} operations [APS15; ESS+19], i.e., $\delta \approx 1.0045$ plausibly provides 128-bit post-quantum security.

Micciancio and Regev [MR09] showed that, from Equation 2.1, it follows that it is often suboptimal to apply the algorithm χ directly to the lattice of interest. For simplicity, let us consider the SIS-problem, i.e., we consider a lattice $\Lambda_q(A)$ with $A \in \mathbb{Z}_q^{k\times m}$, and aim to find a short vector in $\Lambda_q(A)$. For large enough m, the algorithm χ should be applied to a related lattice in $\Lambda_q(A') \subseteq \mathbb{Z}^{m'}$ with

$$m' = \sqrt{\frac{k\log_2(q)}{\log_2(\delta)}}\,.$$

More precisely, if $m > m'$, let $A' \in \mathbb{Z}_q^{k\times m'}$ be a submatrix of A obtained by removing $m - m'$ columns of A. The short vector output by χ applied to $\Lambda_q(A')$ can be appended with $m-m'$ zeros to obtain an element of $\Lambda_q(A)$ with exactly the same norm. Interestingly, for a fixed root Hermite factor δ, this approach outputs shorter vectors than applying χ directly to $\Lambda_q(A)$. In fact, the above approach is expected to output vectors of length

$$\|\mathbf{s}\|_2 \geq \min\Big(q, 2^{2\sqrt{k\log\delta\log q}}\Big)\,.$$

Note that this norm-bound is independent of the dimension m. Hence, when m is large enough, the parameter m does not influence the hardness of the SIS-problem. The same approach applied to the MSIS-problems, where $A \in \mathcal{R}_q^{k\times m}$, is expected to output lattice elements $\mathbf{s} \in \Lambda_q(A) \subseteq \mathcal{R}_q^m$ of norm

$$\|\mathbf{s}\|_2 \geq \min\Big(q, 2^{2\sqrt{dk\log\delta\log q}}\Big)\,, \tag{2.2}$$

where d is the degree the ring extension $\mathcal{R} = \mathbb{Z}[X]/f(X)$ over $\mathbb{Z}$.

In this work, we will mainly be interested in vectors that are short with respect to the ℓ_∞-norm. For this reason we also consider the following variant of the MSIS-problem, where "shortness" is defined in terms of the ℓ_∞-norm. Clearly, the hardness of $\mathrm{MSIS}^\infty_{q,k,m,\beta}$ is implied by the hardness of $\mathrm{MSIS}_{q,k,m,\sqrt{dm}\beta}$.

Definition 2.21 ($\mathrm{MSIS}^\infty_{q,k,m,\beta}$ Problem)**.** Let $\mathcal{R} = \mathbb{Z}[X]/f(X)$ for a monic polynomial $f(X)$. The $\mathrm{MSIS}^\infty_{q,k,m,\beta}$ problem over $\mathcal{R}$ is defined as follows: Given a matrix $A \leftarrow_R \mathcal{R}_q^{k\times m}$ sampled uniformly at random, find a nonzero vector $\mathbf{s} \in \mathcal{R}^m$ such that $A\mathbf{s} = 0 \mod q$ and $\|\mathbf{s}\|_\infty \leq \beta$.

2.8 Interactive (Zero-Knowledge) Proofs

A *binary relation* $\mathfrak{R}$ is a subset of the Cartesian product $X \times Y$ of two sets X and Y. It describes a connection between elements of X and elements of Y. Unless stated otherwise, we assume X and Y to be the set of arbitrary length bit strings $\{0,1\}^*$, and thus relations $\mathfrak{R}$ to be subsets of $\{0,1\}^* \times \{0,1\}^*$.

Following standard terminology, a string $w \in \{0,1\}^*$ is called a *witness* for the *statement* $x \in \{0,1\}^*$ if $(x;w) \in \mathfrak{R}$. The set of valid witnesses for a statement x is denoted by $\mathfrak{R}(x)$, i.e., $\mathfrak{R}(x) = \{w : (x;w) \in \mathfrak{R}\}$. A statement that admits a witness is said to be a *true* or *valid* statement. The set of true statements is denoted by $L_{\mathfrak{R}}$, i.e., $L_{\mathfrak{R}} = \{x : \exists\, w \text{ s.t. } (x;w) \in \mathfrak{R}\}$. A binary relation is said to be an NP relation if the validity of a witness w can be verified in time polynomial in the size $|x|$ of the statement x. In particular, for an NP relation, it holds that the size $|w|$ of a witness $w \in \mathfrak{R}(x)$ is polynomial in $|x|$. From now on we assume all relations to be NP relations.

An interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ aims for a prover $\mathcal{P}$ to convince a verifier $\mathcal{V}$ that a statement x admits a witness, or even that the prover *knows* a witness $w \in \mathfrak{R}(x)$.

Definition 2.22 (Interactive Proof)**.** An *interactive proof* $\Pi = (\mathcal{P}, \mathcal{V})$ for relation $\mathfrak{R}$ is an interactive protocol between two probabilistic machines, a prover $\mathcal{P}$ and a polynomial time verifier $\mathcal{V}$. Both $\mathcal{P}$ and $\mathcal{V}$ take as public input a statement $x \in \{0,1\}^*$, and additionally, $\mathcal{P}$ takes as private input a witness $w \in \mathfrak{R}(x)$, which is denoted as $\Pi(x;w)$ or $(\mathcal{P}(w), \mathcal{V})(x)$. As the output of the protocol, $\mathcal{V}$ either accepts or rejects the statement. Accordingly, we say the corresponding transcript (i.e., the set of all messages exchanged in the protocol execution) is *accepting* or *rejecting*.

An interactive proof Π is *complete* if the verifier $\mathcal{V}$ accepts honest executions with a public-private input pair $(x;w) \in \mathfrak{R}$ with large probability, i.e., the claims made by honest provers are accepted with large probability. It is *sound* if the verifier rejects false statements $x \notin L_{\mathfrak{R}}$ with large probability, i.e., the claims made by dishonest provers are rejected with large probability. Originally interactive proofs were defined to be complete and sound [GMR85]. By contrast, we do not require interactive protocols to satisfy these properties by definition, but consider them as desirable security properties.

Definition 2.23 (Completeness)**.** An interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ for relation $\mathfrak{R}$ is *complete* with completeness error $\rho\colon \mathbb{N} \to [0,1]$ if for all $(x;w) \in \mathfrak{R}$,

$$\Pr\big((\mathcal{P}(w), \mathcal{V})(x) = \mathsf{reject}\big) \leq \rho(|x|)\,.$$

If $\rho(|x|) = 0$ for all x, $(\mathcal{P}, \mathcal{V})$ is said to be perfectly complete.

Definition 2.24 (Soundness)**.** An interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ for relation $\mathfrak{R}$ is *sound* with soundness error $\sigma\colon \mathbb{N} \to [0,1]$ if for all $x \notin L_{\mathfrak{R}}$ and every prover $\mathcal{P}^*$,

$$\Pr\big((\mathcal{P}^*, \mathcal{V})(x) = \mathsf{accept}\big) \leq \sigma(|x|)\,.$$

If this property only holds for (probabilistic) polynomial time (i.e., computationally bounded) provers $\mathcal{P}^*$, then Π is said to be *computationally* sound.

Let us consider some additional (desirable) properties of interactive proofs. We assume that the prover $\mathcal{P}$ sends the first and the last message in any interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$. If this is not the case, the interactive proof can be appended with an empty message. Hence, the number of communication rounds $2\mu + 1$ is always odd. We also say Π is a $(2\mu + 1)$-round protocol. We will refer to *multi-round* protocols as a way of emphasizing that we are not restricting to 3-round protocols.

Definition 2.25 (Public-Coin)**.** An interactive proof $(\mathcal{P}, \mathcal{V})$ is *public-coin* if all of $\mathcal{V}$'s random choices are made public.

If a protocol is public-coin, the verifier only needs to send its random choices to the prover. In this case, $\mathcal{V}$'s messages are also referred to as *challenges*, and the set from which $\mathcal{V}$ samples its messages uniformly at random is called the challenge set.

We refer to a 3-round public-coin interactive proof as a Σ-*protocol*. Note that often a Σ-protocol is required to be (perfectly) complete, special-sound and special honest-verifier zero-knowledge (SHVZK) by definition. However, we do not require a Σ-protocol to have these additional properties.

Definition 2.26 (Σ-Protocol)**.** A Σ-protocol is a 3-round public-coin interactive proof.

2.8.1 Knowledge Soundness

If an interactive proof is complete and sound, it "merely" allows a prover to convince a verifier that a statement x admits a witness, i.e., $x \in L_{\mathfrak{R}}$. It does not necessarily convince a verifier that the prover "knows" a witness $w \in \mathfrak{R}(x)$. Informally, a prover $\mathcal{P}^*$ is said to know a witness w if it can *compute* this witness efficiently. More precisely, knowledge of w requires the existence of an efficient algorithm that, given x and *oracle* access to $\mathcal{P}^*$, outputs a witness $w \in \mathfrak{R}(x)$. For a more elaborate discussion on the definition of knowledge we refer to [Gol04].

The above allows us to define what it means for an interactive proof to prove knowledge of a witness w. This stronger notion of soundness is called *knowledge soundness* and is formally defined in Definition 2.27.

Definition 2.27 (Knowledge Soundness)**.** An interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ for relation $\mathfrak{R}$ is *knowledge sound* with knowledge error $\kappa \colon \mathbb{N} \to [0, 1]$ if there exists a positive polynomial q and an algorithm $\mathcal{E}$, called a *knowledge extractor*, with the following properties: The extractor $\mathcal{E}^{\mathcal{P}^*}(x)$, given input x and oracle access to a (potentially dishonest) prover $\mathcal{P}^*$, runs in an expected number of steps that is polynomial in $|x|$ and outputs a witness $w \in \mathfrak{R}(x)$ with probability

$$\Pr\big((x; \mathcal{E}^{\mathcal{P}^*}(x)) \in \mathfrak{R}\big) \geq \frac{\epsilon(x, \mathcal{P}^*) - \kappa(|x|)}{q(|x|)},$$

where $\epsilon(x, \mathcal{P}^*) := \Pr\big((\mathcal{P}^*, \mathcal{V})(x) = \mathsf{accept}\big)$.

If these properties only hold for probabilistic polynomial time (i.e., computationally bounded) provers $\mathcal{P}^*$, then Π is said to be *computationally* knowledge sound.

The extraction algorithm of Definition 2.27 only has oracle or black-box access to $\mathcal{P}^*$. For this reason, this is also referred to as *black-box extraction*. Moreover, the efficiency of an extractor is oftentimes measured in the (expected) number of times it invokes, or queries, $\mathcal{P}^*$.

If $\epsilon(x, \mathcal{P}^*) = \Pr\big((\mathcal{P}^*, \mathcal{V})(x) = \mathsf{accept}\big) > \kappa(|x|)$, then the success probability of the knowledge extractor of Definition 2.27 is positive. Hence, $\epsilon(x, \mathcal{P}^*) > \kappa(|x|)$ implies that x admits a witness, i.e., $x \in L_{\mathfrak{R}}$. It therefore follows that knowledge soundness with knowledge error $\kappa(|x|)$ implies soundness with soundness error $\sigma(|x|) = \kappa(|x|)$. Hence, knowledge soundness is indeed a stronger property than soundness.

Remark 2.3. It is straightforward to verify that, in order to satisfy Definition 2.27, it is sufficient to show that the required property holds for *deterministic* provers $\mathcal{P}^*$. Namely, let $\mathcal{P}^*$ be an arbitrary probabilistic dishonest prover, and let $\mathcal{P}^*[r]$ be the deterministic prover obtained by fixing $\mathcal{P}^*$'s randomness to r. Then $\epsilon(x, \mathcal{P}^*) = \mathbb{E}_r[\epsilon(x, \mathcal{P}^*[r])]$, where $\mathbb{E}_r$ denotes the expectation over the random choice of r. Furthermore, if $\mathcal{E}^{\mathcal{P}^*}(x)$ is declared to run $\mathcal{E}^{\mathcal{P}^*[r]}(x)$ for a random choice of r, then the same holds for the success probability of the extractor:

$$\Pr\big((x; \mathcal{E}^{\mathcal{P}^*}(x)) \in \mathfrak{R}\big) = \mathbb{E}_r\big[\Pr\big((x; \mathcal{E}^{\mathcal{P}^*[r]}(x)) \in \mathfrak{R}\big)\big] .$$

It follows that in order to satisfy Definition 2.27, it is sufficient to show that the required property holds for *deterministic* provers $\mathcal{P}^*$. For this reason, we may assume provers to be deterministic, in particular, we will consider the prover's first message to be deterministic. This will significantly simplify our analysis.

Definition 2.27 deviates from the more common textbook definition of knowledge soundness [Gol04; HL10] given in Definition 2.28. Instead of requiring the existence of an extractor that runs in expected polynomial time and succeeds with probability at least $(\epsilon(x, \mathcal{P}^*) - \kappa(|x|))/q(|x|)$, the textbook definition requires the existence of an extractor that, as long as $\epsilon(x, \mathcal{P}^*) > \kappa(|x|)$, *always* succeeds, but has an expected runtime that is inversely proportional to $\epsilon(x, \mathcal{P}^*) - \kappa(|x|)$. In particular, the latter extractor does not necessarily run in polynomial time. The two definitions are known to be equivalent [Gol04, Proposition 4.7.4] and therefore display a trade-off between the success probability and the expected runtime of the extractor. We will be using Definition 2.27, since this formulation simplifies our analysis. It is, for instance, much less obvious that it is sufficient to consider only deterministic provers if one uses Definition 2.28 directly.

Definition 2.28 (Knowledge Soundness - Equivalent Definition)**.** An interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ for relation $\mathfrak{R}$ is *knowledge sound* with knowledge error $\kappa\colon \mathbb{N} \to [0, 1]$ if there exists a positive polynomial q and an algorithm $\mathcal{E}$, called a *knowledge extractor*, with the following properties: The extractor $\mathcal{E}^{\mathcal{P}^*}(x)$, given input x and oracle access to a (potentially dishonest) prover $\mathcal{P}^*$ with $\epsilon(x, \mathcal{P}^*) := \Pr\big((\mathcal{P}^*, \mathcal{V})(x) = \mathsf{accept}\big) > \kappa(|x|)$, outputs a witness $w \in \mathfrak{R}(x)$ in an expected number of steps bounded by

$$\frac{q(|x|)}{\epsilon(x, \mathcal{P}^*) - \kappa(|x|)} .$$

Remark 2.4. By Definition 2.28 it is obvious that, in order to prove knowledge soundness, it is enough to consider statements $x \in \{0,1\}^*$ for which the prover $\mathcal{P}^*$ succeeds with probability $\epsilon(x, \mathcal{P}^*) > \kappa(|x|)$, i.e., there are no requirements on the behavior of the extractor for statements x with $\epsilon(x, \mathcal{P}^*) \leq \kappa(|x|)$. By contrast, Definition 2.27 requires extractors to be efficient for *all* statements x. This seems to be a stronger requirement, however the equivalence between these two definitions proves the contrary. Therefore, also towards satisfying Definition 2.27, it is enough to consider statements x with $\epsilon(x, \mathcal{P}^*) > \kappa(|x|)$. Since almost all our knowledge extractors are efficient for all x, we typically do not have to distinguish between statements x with $\epsilon(x, \mathcal{P}^*) > \kappa(|x|)$ and statements x with $\epsilon(x, \mathcal{P}^*) \leq \kappa(|x|)$.

Remark 2.5. In principle one could allow the completeness, soundness and knowledge error to be functions of the statement x instead of its size $|x|$. Both versions appear in literature, e.g., Goldreich [Gol04] defines these errors as functions of $|x|$, whereas Hazay and Lindell [HL10] define them as functions of x.

Remark 2.6. Sometimes a slightly weaker definition of knowledge soundness is used [BG92; Gol04; HL10]. This weaker definition decouples knowledge soundness from soundness by only requiring the extractor to run in expected polynomial time on inputs $x \in L_{\mathfrak{R}}$, i.e., it does not require the protocol to be sound. The reason is that in some applications the public input is guaranteed to be a *true* statement, i.e., admitting a witness. In these applications it does not matter how the protocol behaves on inputs $x \notin L_{\mathfrak{R}}$, i.e., the protocol does not need to be sound. It is straightforward to show that a *sound* protocol satisfying this weaker notion of knowledge soundness is also knowledge sound in the stronger sense of Definition 2.27.

Definition 2.29 (Proof of Knowledge)**.** An interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ that is both complete with completeness error $\rho(\cdot)$ and knowledge sound with knowledge error $\kappa(\cdot)$ is a *Proof of Knowledge* (PoK) if there exists a polynomial q such that $1 - \rho(|x|) \geq \kappa(|x|) + 1/q(|x|)$ for all x.

Definition 2.30 (Argument of Knowledge)**.** An interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ that is both complete with completeness error $\rho(\cdot)$ and *computationally* knowledge sound with knowledge error $\kappa(\cdot)$ is an *Argument of Knowledge* (AoK) if there exists a polynomial q such that $1 - \rho(|x|) \geq \kappa(|x|) + 1/q(|x|)$ for all x.

Sometimes the alternative, nonequivalent, notion of knowledge soundness presented in Definition 2.31 is used [Cra96; HM98; Unr12]. In this alternative notion, the knowledge extractor is required to run in *strict* polynomial time instead of *expected* polynomial time. However, its success probability is allowed to be proportional to $\big(\epsilon(x, \mathcal{P}^*) - \kappa(|x|)\big)^c$ for an arbitrary constant $c \geq 1$, whereas Definition 2.27 requires the success probability of the extractor to be proportional to $\epsilon(x, \mathcal{P}^*) - \kappa(|x|)$. For some interactive proofs this degradation of the success probability indeed allows the construction of *strict*, instead of *expected*, polynomial time knowledge extractors. Note that, since the success probability of the extractor degrades exponentially in c, this alternative definition only gives a meaningful notion of knowledge soundness if the exponent c is indeed constant.

Definition 2.31 (Knowledge Soundness - Alternative Notion)**.** An interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ for relation $\mathfrak{R}$ is said to satisfy the alternative notion of *knowledge soundness* with knowledge error $\kappa\colon \mathbb{N} \to [0,1]$ if there exists a positive polynomial q, a constant $c \geq 1$ and an algorithm $\mathcal{E}$, called a *knowledge extractor*, with the following properties: The extractor $\mathcal{E}^{\mathcal{P}^*}(x)$, given input x and oracle access to a (potentially dishonest) prover $\mathcal{P}^*$, runs in an expected number of steps that is polynomial in $|x|$ and, if $\epsilon(x, \mathcal{P}^*) > \kappa(|x|)$, outputs a witness $w \in \mathfrak{R}(x)$ with probability

$$\Pr\big((x; \mathcal{E}^{\mathcal{P}^*}(x)) \in \mathfrak{R}\big) \geq \frac{\big(\epsilon(x, \mathcal{P}^*) - \kappa(|x|)\big)^c}{q(|x|)},$$

where $\epsilon(x, \mathcal{P}^*) := \Pr\big((\mathcal{P}^*, \mathcal{V})(x) = \mathsf{accept}\big)$.

2.8.2 Special-Soundness

We recall the notion of (general) *special-soundness*. It is typically easier to prove that an interactive proof is special-sound than to prove that it is knowledge sound. Note that we require special-sound protocols to be public-coin.

Definition 2.32 (k-out-of-N Special-Soundness)**.** Let $k, N \in \mathbb{N}$. A 3-round public-coin interactive proof Π for relation $\mathfrak{R}$, with challenge set of cardinality $N \geq k$, is *k-out-of-N special-sound* if there exists a polynomial time algorithm that, on input a statement x and k accepting transcripts $(a, c_1, z_1), \ldots (a, c_k, z_k)$ with common first message a and pairwise distinct challenges $c_1, \ldots, c_k$, outputs a witness $w \in \mathfrak{R}(x)$. We also say Π is k-special-sound and, if $k = 2$, it is simply said to be special-sound.

In order to generalize k-special-soundness to multi-round protocols, we introduce the notion of a tree of transcripts.

Definition 2.33 (Tree of Transcripts)**.** Let $\mathbf{k} = (k_1, \ldots, k_\mu) \in \mathbb{N}^\mu$. A $\mathbf{k}$-tree of transcripts for a $(2\mu + 1)$-round public-coin interactive proof Π is a set of $K = \prod_{i=1}^{\mu} k_i$ transcripts arranged in the following tree structure. The nodes in this tree correspond to the prover's messages and the edges to the verifier's challenges. Every node at depth i has precisely k_i children corresponding to k_i pairwise distinct challenges. Every transcript corresponds to exactly one path from the root node to a leaf node. For a graphical representation we refer to Figure 2.1. We refer to the corresponding tree of challenges as a $\mathbf{k}$-tree of challenges.

Definition 2.34 ($\mathbf{k}$-out-of-$\mathbf{N}$ Special-Soundness)**.** Let $\mathbf{k} = (k_1, \ldots, k_\mu)$, $\mathbf{N} = (N_1, \ldots, N_\mu) \in \mathbb{N}^\mu$. A $(2\mu + 1)$-round public-coin interactive proof Π for relation $\mathfrak{R}$, where $\mathcal{V}$ samples the i-th challenge from a set of cardinality $N_i \geq k_i$ for $1 \leq i \leq \mu$, is $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound if there exists a polynomial time algorithm that, on input a statement x and a $\mathbf{k}$-tree of accepting transcripts, outputs a witness $w \in \mathfrak{R}(x)$. We also say Π is $\mathbf{k}$-special-sound.

In contrast to the extractor $\mathcal{E}$ of Definition 2.27 that has only oracle access to the prover, the special-soundness algorithm obtains the transcripts directly. For this reason, it is nontrivial to show that special-soundness implies knowledge

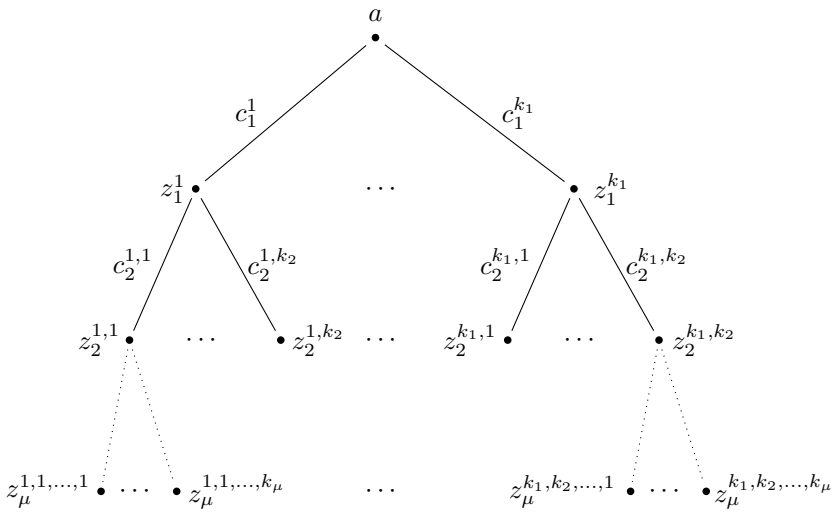

Figure 2.1: $(k_1, \ldots, k_\mu)$-tree of transcripts [ACK21].

soundness. While it is well known that for 3-round protocols special-soundness implies knowledge soundness, previously there was no known generalization to $2\mu+1$-round protocols. In Chapter 6 we show that, also for multi-round protocols, special-soundness tightly implies knowledge soundness.

2.8.3 Zero-Knowledge

In many applications, the prover $\mathcal{P}$ wishes to convince the verifier $\mathcal{V}$ without releasing any information besides the veracity of the claim. In particular, a protocol execution should not reveal any additional information about the secret witness $w \in \mathfrak{R}(x)$, even if the verifier behaves maliciously. An interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ that satisfies this security property is said to be *zero-knowledge* and also called a *zero-knowledge proof* (ZKP).

In Definition 2.35 this security property is formalized by means of a so-called *simulator*. A simulator takes as input the public statement x and outputs protocol transcripts that are distributed statistically close to transcripts generated by interactions with the honest prover $\mathcal{P}$. The existence of a simulator shows that a (potentially dishonest) verifier $\mathcal{V}^*$ can generate transcripts *without* interacting with the honest prover $\mathcal{P}$, i.e., the interactions with $\mathcal{P}$ do not reveal any information that $\mathcal{V}^*$ could not have obtained on its own.

Definition 2.35 (Zero-Knowledge)**.** An interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ for relation $\mathfrak{R}$ is said to be (statistical) *zero-knowledge* (ZK) if, for every (potentially dishonest) polynomial time verifier $\mathcal{V}^*$, there exists a polynomial time simulator $\mathcal{S}^*$ such that the following families of random variables are statistically indistinguishable:

- $\{\mathsf{view}^{\mathcal{P}}_{\mathcal{V}^*}(x; w) : (x; w) \in \mathfrak{R}\}$, where $\mathsf{view}^{\mathcal{P}}_{\mathcal{V}^*}(x; w)$ describes $\mathcal{P}$'s messages and

$\mathcal{V}^*$'s random tape when evaluating $(\mathcal{P}, \mathcal{V}^*)$ on input $(x; w)$;

- $\{\mathcal{S}^*(x) : (x; w) \in \mathfrak{R}\}$.

If these families of random variables are only computationally indistinguishable, Π is said to be *computationally* zero-knowledge.

Remark 2.7. Sometimes it is convenient to make the statistical distance between the distributions of Definition 2.35 explicit. In this case, we say Π is δ-statistical zero-knowledge, for some $\delta \colon \mathbb{N} \to [0, 1]$, if

$$\Delta\big(\mathsf{view}^{\mathcal{P}}_{\mathcal{V}^*}(x; w), \mathcal{S}^*(x)\big) \leq \delta(|x|) \quad \forall (x; w) \in \mathfrak{R}.$$

We also consider a weaker notion of zero-knowledge: *honest-verifier zero-knowledge*. This notion only requires the existence of a simulator for the *honest* verifier $\mathcal{V}$, i.e., a simulator that outputs transcripts distributed statistically close to transcripts of *honest* executions of Π. Typically, a prover cannot distinguish between interactions with honest and dishonest verifiers, therefore in most applications this weaker security property does not suffice. However, there exist generic transformations that transform certain classes of HVZK interactive proofs, such as public-coin ones, into zero-knowledge interactive proofs [OVY93; Dam93; DGO+95]. Alternatively, public-coin interactive proofs can be made non-interactive by applying the Fiat-Shamir transform [FS86]. In this transformation, the verifier's messages (challenges) are replaced by random oracle queries. In the Fiat-Shamir mode, honest-verifier zero-knowledge does suffice. For these reasons, it is often enough to show that an interactive proof is honest-verifier zero-knowledge.

Definition 2.36 ((Special) Honest-Verifier Zero-Knowledge)**.** An interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ for relation $\mathfrak{R}$ is said to be (statistical) *honest-verifier zero-knowledge* (HVZK) if there exists a polynomial time simulator $\mathcal{S}$ such that the following families of random variables are statistically indistinguishable:

- $\{\mathsf{view}^{\mathcal{P}}_{\mathcal{V}}(x; w) : (x; w) \in \mathfrak{R}\}$, where $\mathsf{view}^{\mathcal{P}}_{\mathcal{V}}(x; w)$ describes $\mathcal{P}$'s messages and $\mathcal{V}$'s random tape when evaluating $\Pi = (\mathcal{P}, \mathcal{V})$ on input $(x; w)$;
- $\{\mathcal{S}(x) : (x; w) \in \mathfrak{R}\}$.

If $\Delta\big(\mathsf{view}^{\mathcal{P}}_{\mathcal{V}}(x; w), \mathcal{S}(x)\big) = 0$ for all $(x; w) \in \mathfrak{R}$, Π is said to be *perfectly* HVZK. If these families of random variables are only computationally indistinguishable, Π is said to be *computationally* HVZK. Further, if the simulator proceeds by first sampling the verifier's messages uniformly at random, Π is said to be *special* honest-verifier zero-knowledge (SHVZK).

Finally, we consider yet another relaxation of the zero-knowledge property. For some interactive proofs $\Pi = (\mathcal{P}, \mathcal{V})$, honest executions do reveal information about the secret witness $w \in \mathfrak{R}(x)$, but *only* if the prover $\mathcal{P}$ aborts during the protocol execution. These protocols admit a simulator that can simulate *non-aborting* transcripts and are said to be *non-abort honest-verifier zero-knowledge* (NA-HVZK). It is typically straightforward to transform an NA-HVZK interactive proof into

one that is HVZK. Moreover, in the non-interactive Fiat-Shamir instantiation of a public-coin interactive proof, aborting executions are never published, and the weaker notion of NA-HVZK suffices. In the literature NA-HVZK is often simply referred to as HVZK. We use a different notation to emphasize the difference.

Definition 2.37 (Non-Abort Honest-Verifier Zero-Knowledge)**.** An interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ for relation $\mathfrak{R}$ is said to be (statistical) *non-abort honest-verifier zero-knowledge* (NA-HVZK) if there exists a polynomial time simulator $\mathcal{S}$ such that the following families of random variables are statistically indistinguishable:

- $\{\mathsf{NA\text{-}view}_{\mathcal{V}}^{\mathcal{P}}(x;w) : (x;w) \in \mathfrak{R}\}$, where $\mathsf{NA\text{-}view}_{\mathcal{V}}^{\mathcal{P}}(x;w)$ describes $\mathcal{P}$'s messages and $\mathcal{V}$'s random tape when evaluating $\Pi = (\mathcal{P}, \mathcal{V})$ on input $(x;w)$, conditioned on $\mathcal{P}$ not aborting;
- $\{\mathcal{S}(x) : (x;w) \in \mathfrak{R}\}$.

If the simulator proceeds by first sampling the verifier's messages uniformly at random, then Π is said to be non-abort *special* honest-verifier zero-knowledge (NA-SHVZK).

Remark 2.8. Definition 2.37 allows the abort probability of an honest prover to depend on the secret witness $w \in \mathfrak{R}(x)$. However, the generic transformations from NA-HVZK to HVZK typically require the abort probability to be essentially independent of the witness w. Moreover, also in the non-interactive Fiat-Shamir mode, it is preferable to have an abort probability independent of the witness; otherwise, the non-interactive proof might be susceptible to side-channel attacks, e.g., timing attacks. For this reason, in addition, we typically require that the abort probability of an honest prover is essentially independent of the witness w.

2.9 Non-Interactive Proofs in the Random Oracle Model

In the *random oracle model* (ROM), algorithms have oracle access to a function $\mathsf{RO}\colon \{0,1\}^* \to \{0,1\}^\eta$, called a random oracle, sampled uniformly at random from the set of functions with domain $\{0,1\}^*$ and codomain $\{0,1\}^\eta$ for some $\eta \in \mathbb{N}$. A random oracle RO is implicitly instantiated by *lazy sampling*, i.e., every time the random oracle is queried on a new input $x \in \{0,1\}^*$, the evaluation $\mathsf{RO}(x) \in \{0,1\}^\eta$ is sampled uniformly at random and fixed from that point onward. In particular, if the random oracle is queried on the same input x as before, possibly by a different algorithm, it will return the same output $\mathsf{RO}(x)$.

A random oracle $\mathsf{RO}\colon \{0,1\}^* \to \{0,1\}^\eta$ outputs bitstrings of length η. However, the codomain of a random oracle is adapted easily. For instance, if one requires bitstrings of length $\eta' \leq \eta$, the evaluation $\mathsf{RO}(x)$ can be truncated to its first η' bits and, if one requires bitstrings of length $k \cdot \eta$, simply define

$$\mathsf{RO}'\colon \{0,1\}^* \to \{0,1\}^{k\cdot\eta}, \quad x \mapsto \mathsf{RO}(1\|x)\|\cdots\|\mathsf{RO}(k\|x)\,,$$

where $i\|x$ denotes the bitstring x prepended with the bit decomposition of $i \in \mathbb{N}$. In fact, the random oracle $\mathsf{RO}\colon \{0,1\}^* \to \{0,1\}^\eta$ can be adapted to output elements in any finite set $\mathcal{Y}$. Therefore, we allow the codomain of a random oracle

to be an arbitrary finite set $\mathcal{Y}$. Moreover, for convenience, we sometimes leave the codomain $\mathcal{Y}$ implicit and write $\mathcal{RO}$ for the set of all random oracles. Further, to avoid technical difficulties, we sometimes limit the domain from $\{0,1\}^*$ to $\{0,1\}^{\leq u}$, the finite set of all bitstrings of length at most u, for a sufficiently large $u \in \mathbb{N}$.

An algorithm $\mathcal{A}$ with oracle access to a random oracle RO, which is denoted as $\mathcal{A}^{\mathsf{RO}}$, is called a *random oracle algorithm*. The algorithm $\mathcal{A}$ is said to be a *Q-query* random oracle algorithm if, for all inputs x, random tapes r and random oracles RO, $\mathcal{A}^{\mathsf{RO}}$ makes at most Q queries to RO.

The Fiat-Shamir transformation (Section 2.9.2) allows *public-coin* interactive proofs $\Pi = (\mathcal{P}, \mathcal{V})$ to be made non-interactive in the random oracle model. The high level idea is that the verifier's challenges are replaced by random oracle queries. This way the prover can generate a proof for knowledge of a witness $w \in \mathfrak{R}(x)$ without interacting with the verifier. The resulting protocol is called a *non-interactive random oracle proof* (NIROP). Vice versa, a non-interactive random oracle proof also corresponds to an interactive proof, obtained by replacing the random oracle queries with challenges sampled by the verifier.

Definition 2.38 (Non-Interactive Random Oracle Proof)**.** A *non-interactive random oracle proof* (NIROP) for relation $\mathfrak{R}$ is a pair $\Pi = (\mathcal{P}, \mathcal{V})$ of (probabilistic) random-oracle algorithms, a prover $\mathcal{P}$ and a polynomial time verifier $\mathcal{V}$, such that: Given $(x; w) \in \mathfrak{R}$ and access to a random oracle RO, the prover $\mathcal{P}^{\mathsf{RO}}(x; w)$ outputs a proof π. Given $x \in \{0,1\}^*$, a purported proof π, and access to a (random) oracle RO, the verifier $\mathcal{V}^{\mathsf{RO}}(x, \pi)$ outputs 0 to reject or 1 to accept the proof.

Remark 2.9. Standard techniques for "domain separation" allow multiple random oracles $\mathsf{RO}_1, \ldots, \mathsf{RO}_k$ to be constructed from a single one [BR93], e.g., by defining $\mathsf{RO}_i(x) := \mathsf{RO}(i \| x)$ for all $1 \leq i \leq k$. For this reason, if required or convenient, we allow the prover $\mathcal{P}$ and the verifier $\mathcal{V}$ of a NIROP $\Pi = (\mathcal{P}, \mathcal{V})$ to have access to multiple independent random oracles $\mathsf{RO}_1, \ldots, \mathsf{RO}_k$, possibly with different codomains.

The following definition is a natural adaptation of the completeness property for interactive proofs. Note that here, besides $\mathcal{P}$'s and $\mathcal{V}$'s randomness, the probability is over the randomness of the random oracle RO.

Definition 2.39 (Completeness - NIROP)**.** A non-interactive random oracle proof $\Pi = (\mathcal{P}, \mathcal{V})$ for relation $\mathfrak{R}$ is *complete* with completeness error $\rho \colon \mathbb{N} \to [0,1]$ if, for all $(x; w) \in \mathfrak{R}$,

$$\Pr\big((\mathcal{P}^{\mathsf{RO}}(w), \mathcal{V}^{\mathsf{RO}})(x) = \mathsf{reject} : \mathsf{RO} \leftarrow_R \mathcal{RO}\big) \leq \rho(|x|)\,.$$

If $\rho(|x|) = 0$ for all x, $(\mathcal{P}, \mathcal{V})$ is said to be perfectly complete.

Similarly, the soundness property of interactive proofs can be adapted to a soundness property for non-interactive random oracle proofs. Note that the soundness error $\sigma(|x|, Q)$ is allowed to depend on the query complexity Q of the prover $\mathcal{P}^*$ attacking the considered NIROP. For many NIROPs, it is indeed the case that the success probability of a cheating prover $\mathcal{P}^*$ increases with the number of random oracle queries Q admitted to the prover $\mathcal{P}^*$.

Definition 2.40 (Soundness - NIROP)**.** A non-interactive random oracle proof $\Pi = (\mathcal{P}, \mathcal{V})$ for relation $\mathfrak{R}$ is *sound* with soundness error $\sigma\colon \mathbb{N} \times \mathbb{N} \to [0,1]$ if for all $x \notin L_{\mathfrak{R}}$ and every Q-query prover $\mathcal{P}^*$,

$$\Pr\big((\mathcal{P}^{*,\mathsf{RO}}, \mathcal{V}^{\mathsf{RO}})(x) = \mathsf{accept} : \mathsf{RO} \leftarrow_R \mathcal{RO}\big) \leq \sigma(|x|, Q)\,.$$

If this property only holds for (probabilistic) polynomial time (i.e., computationally bounded) provers $\mathcal{P}^*$, then Π is said to be *computationally* sound.

Also the knowledge soundness definition can be adapted to non-interactive random oracle proofs. As before, knowledge soundness requires the existence of an extractor $\mathcal{E}$ that, given input x and oracle access to a prover $\mathcal{P}^*$, aims to output a witness $w \in \mathfrak{R}(x)$. However, a crucial difference with Definition 2.27, for interactive proofs, is that now the prover $\mathcal{P}^*$ attacking the considered NIROP is a *random oracle* algorithm, instead of a "normal" algorithm. Giving the knowledge extractor $\mathcal{E}$ oracle access to the random oracle algorithm $\mathcal{P}^*$ means that $\mathcal{E}$ can invoke $\mathcal{P}^{*,\mathsf{RO}}$ for any random oracle RO. More precisely, $\mathcal{E}$ observes all the random oracle queries made by $\mathcal{P}^*$ and is free to decide how to answer these queries. We also say that $\mathcal{E}$ implements RO for $\mathcal{P}^*$. Hence, instead of extracting a witness by controlling the verifier's challenge, an extractor for NIROPs aims to output a witness by controlling the random oracle responses.

Definition 2.41 (Knowledge Soundness - NIROP)**.** A non-interactive random oracle proof $\Pi = (\mathcal{P}, \mathcal{V})$ for relation $\mathfrak{R}$ is *knowledge sound* with *knowledge error* $\kappa\colon \mathbb{N} \times \mathbb{N} \to [0,1]$, if there exists a positive polynomial q and an algorithm $\mathcal{E}$, called a *knowledge extractor*, with the following properties: The extractor $\mathcal{E}^{\mathcal{P}^*}(x)$, given input x and oracle access to a (potentially dishonest) Q-query random oracle prover $\mathcal{P}^*$, runs in an expected number of steps that is polynomial in $|x|$ and Q and outputs a witness $w \in \mathfrak{R}(x)$ with probability

$$\Pr\big((x; \mathcal{E}^{\mathcal{P}^*}(x)) \in \mathfrak{R}\big) \geq \frac{\epsilon(x, \mathcal{P}^*) - \kappa(|x|, Q)}{q(|x|)}\,,$$

where $\epsilon(x, \mathcal{P}^*) = \Pr\big(\mathcal{V}^{\mathsf{RO}}(x, \mathcal{P}^{*,\mathsf{RO}}(x)) = \mathsf{accept} : \mathsf{RO} \leftarrow_R \mathcal{RO}\big)$.

It is easy to see that any cheating strategy for the interactive proof corresponding to a NIROP gives a cheating strategy for the NIROP itself that succeeds with exactly the same probability. Hence, $\kappa_{\mathrm{IP}}(|x|) \leq \kappa_{\mathrm{NI}}(|x|, Q)$ for all $x \in \{0,1\}^*$ and $Q \in \mathbb{N}$, where $\kappa_{\mathrm{IP}}(|x|)$ and $\kappa_{\mathrm{NI}}(|x|, Q)$ are the knowledge errors of the interactive and non-interactive proofs, respectively. For this reason we also refer to the ratio

$$\frac{\kappa_{\mathrm{NI}}(|x|, Q)}{\kappa_{\mathrm{IP}}(|x|)}$$

as the security loss of the NIROP. We are typically interested in how this security loss scales as a function of Q.

Finally, let us consider the zero-knowledge property. As for interactive proofs, a non-interactive random oracle proof $\Pi = (\mathcal{P}, \mathcal{V})$ is said to be zero-knowledge if there exists a simulator that aims to output a proof π that is indistinguishable

from honestly generated proofs. To this end, it is given as input a statement x and oracle access to a random oracle RO. However, in contrast to honest provers, the simulator is allowed to *reprogram* the random oracle $\mathsf{RO}\colon \{0,1\}^* \to \mathcal{Y}$ at arbitrary inputs. Let $L = \{(x_1, y_1), \ldots, (x_k, y_k)\} \subseteq \{0,1\}^* \times \mathcal{Y}$ with pairwise distinct x_i, then we write $\mathsf{RO}[L]$ for the random oracle that is reprogrammed in L, i.e.,

$$\mathsf{RO}[L](x) = \begin{cases} y_i, & \text{if } \exists i \text{ s.t. } x_i = x, \\ \mathsf{RO}(x), & \text{otherwise.} \end{cases}$$

This zero-knowledge property for non-interactive random oracle proofs is formalized in the following definition. It is easily seen that replacing the challenges of an honest-verifier zero-knowledge interactive proof by random oracle queries results in a NIROP that is zero-knowledge.

Definition 2.42 (Zero-Knowledge - NIROP)**.** A non-interactive random oracle proof $\Pi = (\mathcal{P}, \mathcal{V})$ for relation $\mathfrak{R}$ is said to be (statistical) *zero-knowledge* if there exists a polynomial time random oracle simulator $\mathcal{S}$ such that, for every distinguisher $\mathcal{D}\colon \{0,1\}^* \to \{0,1\}$, the two families $\{X(x;w) : (x;w) \in \mathfrak{R}\}$ and $\{Y(x;w) : (x;w) \in \mathfrak{R}\}$ of distributions defined as

- $X(x;w) = \mathcal{D}^{\mathsf{RO}[L]}(\pi)$, where $(\pi, L) \leftarrow \mathcal{S}^{\mathsf{RO}}(x)$ and $\mathsf{RO} \leftarrow_R \mathcal{RO}$;
- $Y(x;w) = \mathcal{D}^{\mathsf{RO}}(\pi)$, where $\pi \leftarrow \mathcal{P}^{\mathsf{RO}}(x;w)$ and $\mathsf{RO}_R \leftarrow \mathcal{RO}$;

are statistically indistinguishable. If the above only holds for polynomial time distinguishers $\mathcal{D}$, Π is said to be *computationally* zero-knowledge.

2.9.1 Adaptive Knowledge Soundness

Thus far, knowledge soundness has been defined with respect to *static* or *non-adaptive* provers $\mathcal{P}^*$ attacking the considered (non-)interactive proof for a *fixed* statement x. However, in many practical scenarios the dishonest provers are free to *choose* the statement x adaptively. Hence, in these cases static security is not sufficient. For *interactive* proofs, it is well known that static knowledge soundness implies adaptive knowledge soundness. However, this does not carry over to non-interactive proofs. For instance, it is easy to see that the static Fiat-Shamir transformation (see Definition 2.44) is in general not adaptively sound.

For this reason, let us formalize adaptive knowledge soundness for non-interactive random oracle proofs. An adaptive prover $\mathcal{P}^{\mathsf{a}}$ attacking the considered NIROP is given oracle access to a random oracle RO and outputs a statement x of fixed length $|x| = n$ together with a proof π. As in the static definition, adaptive knowledge soundness requires the existence of a knowledge extractor. However, formalizing the requirements of this extractor introduces some subtle issues. Namely, because $\mathcal{P}^{\mathsf{a}}$ chooses the statement x adaptively, it is not immediately clear for which statement the extractor should extract a witness. For instance, granting the extractor the same freedom of adaptively choosing the statement x, for which it needs to extract a witness w, renders knowledge extraction trivial; the extractor could simply output an arbitrary statement-witness pair $(x;w)$. For this reason, we require the extractor to output statement-witness pairs $(x;w)$ corresponding

to the *valid* pairs (x, π) output by the adaptive prover $\mathcal{P}^{\mathsf{a}}$. To formalize these requirements, we also write (x, π, v), with $v \in \{0,1\}$ indicating whether π is a valid proof for statement x. Given this notation, the extractor should output a triple (x, π, v) with the same distribution as the triples (x, π, v) produced by $\mathcal{P}^{\mathsf{a}}$; furthermore, if π is a valid proof for statement x, i.e., $v = 1$, then the extractor should additionally aim to output a witness $w \in \mathfrak{R}(x)$. As before, the success probability of the extractor is allowed to depend on the success probability of $\mathcal{P}^{\mathsf{a}}$. Finally, to ensure that the knowledge extractor can be used in compositional settings, where the NIROP is deployed as a component of a larger protocol, the prover $\mathcal{P}^{\mathsf{a}}$ is also allowed to additionally output arbitrary auxiliary information $\mathsf{aux} \in \{0,1\}^*$, and the extractor is then required to simulate the tuple $(x, \pi, \mathsf{aux}, v)$, rather than the triple (x, π, v). The following definition formalizes adaptive knowledge soundness along these lines. For alternative definitions see, e.g., [Unr17; DFM+19].

Definition 2.43 (Adaptive Knowledge Soundness - NIROP)**.** A non-interactive random oracle proof $(\mathcal{P}, \mathcal{V})$ for relation $\mathfrak{R}$ is *adaptively knowledge sound* with *knowledge error* $\kappa\colon \mathbb{N} \times \mathbb{N} \to [0,1]$, if there exists a positive polynomial q and an algorithm $\mathcal{E}$, called a *knowledge extractor*, with the following properties: The extractor, given input $n \in \mathbb{N}$ and oracle access to any adaptive Q-query random oracle prover $\mathcal{P}^{\mathsf{a}}$ that outputs statements x with $|x| = n$, runs in an expected number of steps that is polynomial in n and Q and outputs a tuple $(x, \pi, \mathsf{aux}, v; w)$ such that $\{(x, \pi, \mathsf{aux}, v) : (x, \pi, \mathsf{aux}) \leftarrow \mathcal{P}^{\mathsf{a},\mathsf{RO}} \wedge v \leftarrow \mathcal{V}^{\mathsf{RO}}(x, \pi)\}$ and $\{(x, \pi, \mathsf{aux}, v) : (x, \pi, \mathsf{aux}, v; w) \leftarrow \mathcal{E}^{\mathcal{P}^{\mathsf{a}}}(n)\}$ are identically distributed and

$$\Pr\big(v = \mathsf{accept} \wedge (x; w) \in \mathfrak{R} : (x, \pi, \mathsf{aux}, v; w) \leftarrow \mathcal{E}^{\mathcal{P}^{\mathsf{a}}}(n)\big) \geq \frac{\epsilon(\mathcal{P}^{\mathsf{a}}) - \kappa(n, Q)}{q(n)}\,,$$

where $\epsilon(\mathcal{P}^{\mathsf{a}}) = \Pr\big(\mathcal{V}^{\mathsf{RO}}(x, \pi) = 1 : (x, \pi) \leftarrow \mathcal{P}^{\mathsf{a},\mathsf{RO}}\big)$. Here, $\mathcal{E}$ implements RO for $\mathcal{P}^{\mathsf{a}}$; in particular, $\mathcal{E}$ can arbitrarily program RO. Moreover, the randomness is over the randomness of $\mathcal{E}$, $\mathcal{V}$, $\mathcal{P}^{\mathsf{a}}$ and RO.

Remark 2.10. We note that, while the tuple $(x, \pi, \mathsf{aux}, v)$ is required to have the same distribution for $\mathcal{P}^{\mathsf{a}}$ and $\mathcal{E}(n)$, by default the respective executions of $\mathcal{P}^{\mathsf{a}}$ and $\mathcal{E}(n)$ give rise to two different probability spaces. Looking ahead though, we remark that the extractor that we eventually construct (Section 6.6) first does an honest run of $\mathcal{P}^{\mathsf{a}}$ by faithfully simulating the answers to $\mathcal{P}^{\mathsf{a}}$'s random oracle queries (this produces the tuple $(x, \pi, \mathsf{aux}, v)$ that $\mathcal{E}(n)$ eventually outputs and which so trivially has the right distribution), and then, if π is a valid proof, $\mathcal{E}(n)$ starts rewinding $\mathcal{P}^{\mathsf{a}}$ and reprogramming the random oracle to try to find enough valid proofs to compute a witness. Thus, in this sense, we can then say that $\mathcal{E}(n)$ aims to find a witness $w \in \mathfrak{R}(x)$ for *the statement x output by $\mathcal{P}^{\mathsf{a}}$*.

2.9.2 Fiat-Shamir Transformation

The Fiat-Shamir transformation [FS86] turns a public-coin interactive proof into a non-interactive random oracle proof (NIROP). The general idea is to compute the i-th challenge c_i as a hash (i.e., the output of a random oracle which in practice is a hash function) of the i-th prover message a_i and (some part of) the previous communication transcript. For a Σ-protocol, the challenge c is computed as $c =$

$H(a)$, or as $c = H(x, a)$, where the former is sufficient for *static* security, where the statement x is given as input to the dishonest prover, and the latter is necessary for *adaptive* security, where the dishonest prover can choose the statement x for which it wants to forge a proof.

For multi-round public-coin interactive proofs, there is some degree of freedom in the computation of the i-th challenge. For concreteness and simplicity, we consider a particular version where all previous prover messages are hashed along with the current message. As for Σ-protocols, we consider a static and an adaptive variant of this version of the Fiat-Shamir transformation. In contrast to the static variant, the adaptive Fiat-Shamir transformation includes the statement x in all hash function evaluations. If it is not made explicit which variant is used, the considered result holds for both variants.

Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a $(2\mu + 1)$-round public-coin interactive proof, where the challenge from the i-th round is sampled from set $\mathcal{C}_i$. For simplicity, we consider μ random oracles $\mathsf{RO}_i \colon \{0,1\}^{\leq u} \to \mathcal{C}_i$ that map into the respective challenge spaces.

Definition 2.44 (Fiat-Shamir Transformation)**.** Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a public-coin interactive proof. The static Fiat-Shamir transformation $\mathsf{FS}[\Pi] = (\mathcal{P}_{\mathsf{fs}}, \mathcal{V}_{\mathsf{fs}})$ is the NIROP where $\mathcal{P}_{\mathsf{fs}}^{\mathsf{RO}_1,\ldots,\mathsf{RO}_\mu}(x; w)$ runs $\mathcal{P}(x; w)$, but instead of asking the verifier for the challenge c_i on message a_i, the challenges are computed as

$$c_i = \mathsf{RO}_i(a_1, \ldots, a_{i-1}, a_i)\,; \tag{2.3}$$

the output is then the proof $\pi = (a_1, \ldots, a_{\mu+1})$. On input a statement x and a proof $\pi = (a_1, \ldots, a_{\mu+1})$, $\mathcal{V}_{\mathsf{fs}}^{\mathsf{RO}_1,\ldots,\mathsf{RO}_\mu}(x, \pi)$ accepts if, for c_i as above $\mathcal{V}$ accepts the transcript $(a_1, c_1, \ldots, a_\mu, c_\mu, a_{\mu+1})$ on input x.

If the challenges are computed as

$$c_i = \mathsf{RO}_i(x, a_1, \ldots, a_{i-1}, a_i)\,; \tag{2.4}$$

the resulting NIROP is referred to as the *adaptive* Fiat-Shamir transformation.

By means of reducing the security of other variants of the Fiat-Shamir transformation to Definition 2.44, appropriately adjusted versions of our results also apply to other variants of doing the "chaining" (Equations 2.3 and 2.4) in the Fiat-Shamir transformation, for instance when c_i is computed as $c_i = \mathsf{RO}_i(i, c_{i-1}, a_i)$, or $c_i = \mathsf{RO}_i(x, i, c_{i-1}, a_i)$, where c_0 is the empty string.

2.10 Secret-Sharing Schemes

A secret-sharing scheme allows a secret to be distributed amongst a set of players, such that sufficiently small subsets of players do not have any information about the secret, while large enough subsets are able to reconstruct the secret. A secret-sharing scheme is said to be *linear* if its secret space is a finite field $\mathbb{F}$ and every share can be computed as the linear combination of the secret $s \in \mathbb{F}$ and a number of random field elements. Because a more general treatment is not required in this dissertation, we will restrict ourselves to linear secret-sharing schemes (LSSSs) for which each share is a single field element. For a more general definition, in terms

of error correcting codes and allowing shares to consist of multiple field elements, we refer to [CDN15]. Further, a *packed* LSSS, also called a ramp scheme, considers secret vectors $\mathbf{x} \in \mathbb{F}^m$, i.e. this notion generalizes the secret space dimension from $m = 1$ to arbitrary $m \in \mathbb{N}$.

Definition 2.45 (Packed Linear Secret-Sharing Scheme)**.** Let $m, n, t \in \mathbb{N}$ and $\mathbb{F}$ a finite field. A linear secret-sharing scheme $\mathcal{S}$ for sharing m-dimensional vectors $\mathbf{x} \in \mathbb{F}^m$ amongst a set of n players is defined by a matrix $M \in \mathbb{F}^{n\times(m+t)}$. A secret sharing of $\mathbf{x} \in \mathbb{F}^m$ is computed by sampling a vector $\mathbf{r} \leftarrow_R \mathbb{F}^t$ uniformly at random and outputting the share vector

$$[\mathbf{x}; \mathbf{r}]_{\mathcal{S}} = M \begin{pmatrix} \mathbf{x} \\ \mathbf{r} \end{pmatrix} \in \mathbb{F}^n .$$

If the scheme $\mathcal{S}$ is clear from context, we simply write $[\mathbf{x}; \mathbf{r}]$.

Every player in a linear secret-sharing scheme $\mathcal{S}$ thus corresponds to one row of the matrix M. For all k-subsets[5] $A \subseteq \{1, \ldots, n\}$ of players, $M_A \in \mathbb{F}^{k\times(m+t)}$ is defined to be the matrix consisting of the rows of the players in A. Hence,

$$M_A \begin{pmatrix} \mathbf{x} \\ \mathbf{r} \end{pmatrix} \in \mathbb{F}^k$$

is the vector containing the shares of the players in A. The privacy property of a secret-sharing scheme states that sufficiently small subsets A are not able to deduce any information about the secret vector $\mathbf{x} \in \mathbb{F}^m$ from their shares. These subsets are also referred to as *unqualified*, and the set of all unqualified subsets is referred to as the *adversary structure*.

Definition 2.46 (Secret Sharing - Privacy)**.** Let $m, n, t, p \in \mathbb{N}$ with $p \leq n$ and $\mathbb{F}$ a finite field. A linear secret-sharing scheme $\mathcal{S}$, defined by the matrix $M \in \mathbb{F}^{n\times(m+t)}$, is said to have *p-privacy* if for every p-subset $A \subseteq \{1, \ldots, n\}$, the distribution

$$\left\{ M_A \begin{pmatrix} \mathbf{x} \\ \mathbf{r} \end{pmatrix} \in \mathbb{F}^p : \mathbf{r} \leftarrow_R \mathbb{F}^t \right\}$$

is independent of $\mathbf{x} \in \mathbb{F}^m$.

The reconstruction property of a secret-sharing scheme states that sufficiently large subsets of players are able to reconstruct the secret given their shares. These subsets are also referred to as *qualified*, and the set of all qualified subsets is referred to as the *access structure*. In this dissertation, the definitions are restricted to *threshold* access structures, i.e., an access structure containing all subsets of a certain minimal cardinality. For a treatment of more general access structures we refer to [CDN15].

Definition 2.47 (Secret Sharing - Reconstruction)**.** Let $m, n, t, r \in \mathbb{N}$ with $r \leq n$ and $\mathbb{F}$ a finite field. A linear secret-sharing scheme $\mathcal{S}$, defined by the

[5] A k-subset is a subset of cardinality k.

matrix $M \in \mathbb{F}^{n\times(m+t)}$, is said to have *r-reconstruction* if, for every r-subset $A \subseteq \{1,\dots,n\}$, $\mathbf{x} \in \mathbb{F}^m$ is uniquely determined by

$$M_A \begin{pmatrix} \mathbf{x} \\ \mathbf{r} \end{pmatrix} \in \mathbb{F}^r .$$

It can be shown that an LSSS with r-reconstruction, for every r-subset $A \subseteq \{1,\dots,n\}$ admits a matrix $U_A \in \mathbb{F}^{m\times r}$ such that

$$U_A M_A \begin{pmatrix} \mathbf{x} \\ \mathbf{r} \end{pmatrix} = \mathbf{x} \in \mathbb{F}^m ,$$

for all $\mathbf{x}$ and $\mathbf{r}$ [CDN15]. Hence, also the reconstruction of a secret from its shares is a linear operation.

If the secret space dimension m of $\mathcal{S}$ equals 1, every subset $A \subseteq \{1,\dots,n\}$ of players is either qualified or unqualified [CDN15, Theorem 6.8]. In this case, there exists a $k \in \mathbb{N}$ such that $\mathcal{S}$ has k-reconstruction and $(k-1)$-privacy, and $\mathcal{S}$ is called a k-out-of-n or a (k,n)-secret-sharing scheme. Note that the above does not hold for arbitrary $m \in \mathbb{N}$. More precisely, if $m > 1$, there might exist subsets A that are neither qualified nor unqualified.

The component-wise product

$$[\mathbf{x};\mathbf{r_x}]_{\mathcal{S}} * [\mathbf{y};\mathbf{r_y}]_{\mathcal{S}} \in \mathbb{F}^n$$

of two share-vectors turns out to be a linear secret sharing of the component-wise product $\mathbf{x} * \mathbf{y} \in \mathbb{F}^m$ of the two secret vectors, however, with respect to a different LSSS $\widehat{\mathcal{S}}$. Namely, let $\widehat{M} \in \mathbb{F}^{n\times(m+t)^2}$ be such that its i-th row is the tensor product of the i-th row of M with itself. Then it is easily seen that

$$[\mathbf{x};\mathbf{r_x}]_{\mathcal{S}} * [\mathbf{y};\mathbf{r_y}]_{\mathcal{S}} = M \begin{pmatrix} \mathbf{x} \\ \mathbf{r_x} \end{pmatrix} * M \begin{pmatrix} \mathbf{y} \\ \mathbf{r_y} \end{pmatrix} = \widehat{M} \left(\begin{pmatrix} \mathbf{x} \\ \mathbf{r_x} \end{pmatrix} \otimes \begin{pmatrix} \mathbf{y} \\ \mathbf{r_y} \end{pmatrix} \right) .$$

Since the vector $\mathbf{x} \otimes \mathbf{y}$ contains the component-wise product $\mathbf{x} * \mathbf{y}$ as a subvector, the above equation shows that $[\mathbf{x};\mathbf{r_x}]_{\mathcal{S}} * [\mathbf{y};\mathbf{r_y}]_{\mathcal{S}}$ is indeed a secret sharing of $\mathbf{x} * \mathbf{y}$ with respect to the LSSS $\widehat{\mathcal{S}}$ defined by $\widehat{M}$. The scheme $\mathcal{S}$ is said to have *product-reconstruction* if $\widehat{S}$ has the reconstruction property. In this case, $\mathcal{S}$ is also said to be *multiplicative*.

Definition 2.48 (Secret Sharing - Product-Reconstruction)**.** Let $m, n, t, R \in \mathbb{N}$ with $R \leq n$ and $\mathbb{F}$ a finite field. A linear secret-sharing scheme $\mathcal{S}$, defined by the matrix $M \in \mathbb{F}^{n\times(m+t)}$, is said to have *R-product-reconstruction* if the secret-sharing scheme $\widehat{\mathcal{S}}$, defined as above by the matrix $\widehat{M}$, has R-reconstruction.

2.10.1 Shamir Secret-Sharing

Shamir's scheme [Sha79] is perhaps the best-known example of a linear secret sharing scheme. Its secret space is a finite field $\mathbb{F}$ with at least $n+1$ elements[6],

[6] When additionally taking the point at infinity into account, this requirement can be relaxed to $|\mathbb{F}| \geq n$. For more details see [CDN15].

where n is the number of players. Instantiated with privacy parameter $1 \leq p \leq n$, it is defined by the Vandermonde matrix

$$M = \begin{pmatrix} 1 & \alpha_1 & \cdots & \alpha_1^p \\ 1 & \alpha_2 & \cdots & \alpha_2^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_n & \cdots & \alpha_n^p \end{pmatrix} \in \mathbb{F}^{n \times p+1} ,$$

where $\alpha_1, \ldots, \alpha_n \in \mathbb{F} \setminus \{0\}$ are pairwise distinct. A Shamir secret sharing $[s; \mathbf{r}]$ of $s \in \mathbb{F}$ thus corresponds to n evaluations of the polynomial $f(X) = s + r_1 X + \cdots r_p X^p \in \mathbb{F}[X]$ of degree at most p, i.e.,

$$[s; \mathbf{r}] = M \begin{pmatrix} s \\ \mathbf{r} \end{pmatrix} = (f(\alpha_1), \ldots, f(\alpha_n)) \in \mathbb{F}^n .$$

By Lagrange interpolation it follows that the polynomial $f(X)$ is uniquely determined by any set containing at least $p + 1$ of its evaluations. Hence, this instantiation of Shamir's secret-sharing scheme has $(p + 1)$-reconstruction. Moreover, again by Lagrange interpolation, for any $s \in \mathbb{F}$ and any p-subset $A \subseteq \{1, \ldots, n\}$, the mapping

$$L \colon \mathbb{F}^p \to \mathbb{F}^p, \quad \mathbf{r} \mapsto M_A \begin{pmatrix} s \\ \mathbf{r} \end{pmatrix}$$

is bijective. Therefore, it follows that this scheme has p-privacy, i.e., it is a $(p+1)$-out-of-n secret sharing scheme.

Further, observe that the component-wise product of two share-vectors equals

$$[s_1; \mathbf{r}_1] * [s_2; \mathbf{r}_2] = (f(\alpha_1), \ldots, f(\alpha_n)) * (g(\alpha_1), \ldots, g(\alpha_n)) = (h(\alpha_1), \ldots, h(\alpha_n)) ,$$

for polynomials $f(X)$, $g(X)$ and $h(X) = f(X)g(X)$. Hence, since $h(X)$ is of degree at most $2p$, this secret-sharing scheme has $(2p + 1)$-product-reconstruction, i.e., if $2p + 1 \leq n$, it is multiplicative.

In the above, the secret $s \in \mathbb{F}$ is allocated to the constant coefficient of the secret-sharing polynomial $f(X)$, i.e., $s = f(0)$. Equivalently, the secret can be allocated to any other evaluation $f(\alpha)$ of $f(X)$. In this case, to secret share s, $f(X)$ is sampled uniformly at random from the set $\mathbb{F}[X]_{\leq p}$ of polynomials of degree at most p, under the condition that $f(\alpha) = s$. The shares then correspond to n evaluations of $f(X)$ in points $\alpha_1, \ldots, \alpha_n \in \mathbb{F} \setminus \{\alpha\}$. This variant has exactly the same properties as before.

Furthermore, Shamir's scheme can easily be adjusted to accommodate secrets of larger dimension m. In this packed secret-sharing variant, to share a vector $\mathbf{x} \in \mathbb{F}^m$, the polynomial $f(X)$ is sampled uniformly at random from the set $\mathbb{F}[X]_{\leq m+p-1}$ of polynomials of degree at most $m + p - 1$, under the condition that $\big(f(1), \ldots, f(m)\big) = \mathbf{x}$. The secret shares correspond to n evaluations of $f(X)$ in pairwise distinct points $\alpha_1, \ldots, \alpha_n \in \mathbb{F} \setminus \{1, \ldots, m\}$, where we assume that $|\mathbb{F}| \geq n + m$. Shamir's packed secret-sharing scheme for sharing m-dimensional vectors $\mathbf{x} \in \mathbb{F}^m$, instantiated with privacy parameter p, has $(m+p)$-reconstruction, p-privacy and $(2m + 2p - 1)$-product-reconstruction. In particular, player subsets of cardinality k, with $p < k < m + p$, are neither qualified nor unqualified.

CHAPTER 3

3
Compressible Σ-Protocols

3.1 Introduction

The theory of Σ-protocols [Cra96] provides a well-understood basis for the modular design of cryptographic protocols. Recently, Bulletproofs [BCC+16; BBB+18] have been introduced as a "drop-in replacement" for Σ-protocols in several important applications. Notably, this includes zero-knowledge for arithmetic circuit relations with communication complexity *logarithmic* in the size of the circuit. By contrast, standard Σ-protocols implement this functionality with *linear* communication complexity.

In this chapter, we reconcile Bulletproofs with Σ-Protocol Theory, allowing for a simpler and modular design of cryptographic protocols within established theory, while achieving exactly the same logarithmic communication. More precisely, we show that Bulletproofs' folding technique can be repurposed as a compression mechanism for a large class of standard Σ-protocols reducing their communication complexity from linear down to logarithmic.

We present our results in an abstract and generic language by observing that the core functionality we are aiming for is proving knowledge of a preimage of some *one-way* group homomorphism

$$\Psi_n \colon \mathbb{G}^n \to \mathbb{H}\,.$$

The desired applications then follow as appropriate instantiations of our abstract protocols.

In Section 3.2, we handle precisely this scenario. First, we present a well-known Σ-protocol for proving knowledge of a preimage of the homomorphism $\Psi_n \colon \mathbb{G}^n \to \mathbb{H}$. Second, by an appropriate adaptation of Bulletproof's folding technique, we show how to reduce the communication complexity from linear down to logarithmic in n. The resulting protocol is referred to as a *compressed Σ-protocol*. Moreover, we provide certain functionality enhancements for (compressed) Σ-protocols.

In Section 3.3, we generalize this functionality to proving knowledge of a "short" preimage. This generalization is motivated by the desired strong-RSA and lattice instantiations of our protocols. In these instantiations the one-way property of the homomorphisms of interest only holds with respect to "short" preimages, i.e., it is easy to find arbitrary preimages, but hard to find short preimages.

In Section 3.4, we discuss perhaps the most prominent instantiation of our abstract protocols; proving knowledge of a commitment opening satisfying a given, but arbitrary, linear constraint. Since the resulting protocols can be instantiated from a wide variety of commitment schemes, the results of this section are still generic; we only require the commitment scheme to be homomorphic and compact, i.e., the size of a commitment should be constant (or at the very least sublinear) in the size of the committed vector. Further, we present certain efficiency improvements for proving knowledge of commitment openings.

This chapter is based on the articles [AC20; ACF21; ACK21], co-authored by Ronald Cramer, Serge Fehr and Lisa Kohl.

3.2 Proving Knowledge of Homomorphism Preimages

Let $\Psi_n \colon \mathbb{G}^n \to \mathbb{H}$ be a homomorphism between abelian groups $(\mathbb{G}^n, +)$ and $(\mathbb{H}, \cdot)$ with prime exponent $q \geq 3$. Note that the group operations in $\mathbb{G}$ (and $\mathbb{G}^n$) are written additively and the ones in $\mathbb{H}$ are written multiplicatively. Further, recall that the exponent of a group $(\mathbb{K}, \cdot)$ is the smallest integer e such that $g^e = 1$ for all all $g \in \mathbb{K}$. In particular, it is easy to see that both $\mathbb{G}$ and $\mathbb{G}^n$ have the same exponent q. Moreover, recall that abelian groups with exponent q are $\mathbb{Z}_q$-modules, and that therefore Ψ_n is actually a $\mathbb{Z}_q$-module homomorphism.

Our goal is to construct a communication-efficient interactive proof for proving knowledge of a preimage $\mathbf{x} \in \mathbb{G}^n$ of a public element $P \in \mathbb{H}$, i.e., an interactive proof for relation

$$\mathfrak{R}_n = \{(P, \Psi_n; \mathbf{x}) : \Psi_n(\mathbf{x}) = P\}\,. \tag{3.1}$$

For technical reasons, we consider the homomorphism Ψ_n as part of the statement. However, if Ψ_n is clear from context, we will also refer to the group elements $P \in \mathbb{H}$ as statements, and thereby omit the more cumbersome statement notation (P, Ψ_n).

Obviously, an interactive proof for relation $\mathfrak{R}_n$ only bears practical relevance for statements (P, Ψ_n), where Ψ_n is a one-way homomorphism, i.e., it should be hard to invert Ψ_n and compute preimages of public elements $P \in \mathbb{H}$. In this case, Ψ_n is a *q-one-way homomorphism* [Cra96; CD98], i.e., Ψ_n is a one-way homomorphism with an efficient procedure for computing preimages of P^q for arbitrary P. However, our techniques do not need Ψ_n to be one-way, and we will therefore not impose this requirement.

In Section 3.2.1, we present a basic Σ-protocol for relation $\mathfrak{R}_n$, following the standard and well-known approach for q-one-way homomorphisms. In Section 3.2.2, we introduce a compression mechanism for reducing the communication costs of this Σ-protocol. In Section 3.2.3, we recursively compose the Σ-protocol with the compression mechanism and obtain a compressed Σ-protocol for relation $\mathfrak{R}_n$ with logarithmic round and communication complexity. To this end, we formalize what it means for two interactive proofs to be composable. In Section 3.2.4, we enhance the functionality with an amortization technique, well known from Σ-protocol theory, for proving knowledge of many preimages for the price of one. Finally, in Section 3.2.5, we present a natural generalization of the compression mechanism, and show how to achieve sublinear, although not logarithmic, communication complexity in a constant number of rounds.

3.2.1 Basic Σ-Protocol

The basic Σ-protocol $\Sigma_{\mathsf{b}} = (\mathcal{P}, \mathcal{V})$ for relation $\mathfrak{R}_n = \{(P, \Psi_n; \mathbf{x}) : \Psi_n(\mathbf{x}) = P\}$, described in Protocol 1, follows the generic design for q-one-way homomorphisms [Cra96; CD98]. Theorem 3.1 shows that Σ_{b} is perfectly complete, 2-out-of-q special-sound and special honest-verifier zero-knowledge (SHVZK). Both the communication costs from the prover $\mathcal{P}$ to the verifier $\mathcal{V}$, and vice versa, are given. Note that such Σ-protocols are oftentimes deployed non-interactively, via the Fiat-Shamir transformation [FS86], in which case the communication costs from verifier to prover might be irrelevant.

Protocol 1 Basic Σ-Protocol Σ_{b} for Relation $\mathfrak{R}_n$.

Parameters:	$n \in \mathbb{N}$, prime q, and groups $(\mathbb{G}, +)$ and $(\mathbb{H}, \cdot)$ with exponent q
Public Input:	$P \in \mathbb{H},\ \Psi_n \in \mathrm{Hom}(\mathbb{G}^n, \mathbb{H})$
Prover's Private Input:	$\mathbf{x} \in \mathbb{G}^n$
Prover's Claim:	$\Psi_n(\mathbf{x}) = P$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
$\mathbf{r} \leftarrow_R \mathbb{G}^n$		
$A = \Psi_n(\mathbf{r})$	$\xrightarrow{A}$	
		$c \leftarrow_R \mathbb{Z}_q$
	$\xleftarrow{c}$	
$\mathbf{z} = \mathbf{r} + c\mathbf{x}$		
	$\xrightarrow{\mathbf{z}}$	$\Psi_n(\mathbf{z}) \stackrel{?}{=} A \cdot P^c$

Theorem 3.1 (Basic Σ-Protocol)**.** *The Σ-protocol Σ_{b} for relation $\mathfrak{R}_n$, described in Protocol 1, is perfectly complete, 2-out-of-q special-sound and special honest-verifier zero-knowledge (SHVZK). Moreover, the communication costs are:*

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

Proof. **Completeness:** This property follows directly from the fact that Ψ_n is a homomorphism between groups with exponent q, i.e., it is a $\mathbb{Z}_q$-module homomorphism.

Special-Soundness: Let $(A, c, \mathbf{z})$ and $(A, c', \mathbf{z}')$ be two accepting transcripts with common first message A and distinct challenges $c \neq c' \in \mathbb{Z}_q$. Then $\bar{\mathbf{z}} = (c - c')^{-1}(\mathbf{z} - \mathbf{z}') \in \mathbb{G}^n$ is easily seen to satisfy $\Psi(\bar{\mathbf{z}}) = P$, i.e., $\bar{\mathbf{z}} \in \mathfrak{R}_n(P, \Psi_n)$ is a witness for statement (P, Ψ_n), which proves that Σ_{b} is 2-out-of-q special-sound.

SHVZK: Transcript are simulated as follows. Sample $c \leftarrow_R \mathbb{Z}_q$ and $\mathbf{z} \leftarrow_R \mathbb{G}^n$ uniformly at random and set $A = \Psi(\mathbf{z}) \cdot P^{-c}$. It is immediate that, if P

admits a witness, i.e., $P \in L_R = \Psi(\mathbb{G}^n)$, then simulated transcripts $(A, c, \mathbf{z})$ have exactly the same distribution as honestly generated transcripts, which completes the proof of the theorem.

□

Remark 3.1. In the proof of Theorem 3.1, it is implicitly assumed that messages of an accepting transcript $(A, c, \mathbf{z})$ for basic Σ-protocol Σ_{b} are of the "correct type." In particular, the prover's first message A is an element in the group $\mathbb{H}$ and the prover's final message $\mathbf{z}$ is a vector in $\mathbb{G}^n$. In practical implementations, this means that the verification algorithm should reject messages that are not of the correct type. In the remainder of this dissertation, without loss of generality, we assume that even dishonest provers deviating from the protocol description always send message of the correct type.

3.2.2 A Compression Mechanism

The communication complexity of Σ-protocol Σ_{b} is linear in n. More precisely, the final message $\mathbf{z} \in \mathbb{G}^n$ of this protocol is n-dimensional, i.e., it has exactly the same size as the secret witness $\mathbf{x}$. The crucial observation is now that this final message is again a witness with respect to relation $\mathfrak{R}_n$, but now for a different statement (Q, Ψ_n), i.e., $\mathbf{z} \in \mathfrak{R}_n(Q, \Psi_n)$. This is no coincidence, as it holds generically for this standard construction of Σ-protocols for q-one-way homomorphisms. The final message of protocol Σ_{b} can therefore be understood as a trivial interactive proof for relation $\mathfrak{R}_n$. Namely, the prover simply reveals the witness $\mathbf{z}$. Note that $Q = A \cdot P^c$ is efficiently computable, given the initial statement P and the first two messages A and c.

Replacing this trivial interactive proof by a more efficient one will thus reduce the communication costs without affecting the security (significantly). In particular, the alternative interactive proof does not have to be zero-knowledge, because the trivial one clearly is not.

Our compression mechanism Σ_{c} is thus again an interactive proof for relation

$$\mathfrak{R}_n = \{(P, \Psi_n; \mathbf{x}) : \Psi_n(\mathbf{x}) = P\}.$$

However, in contrast to Σ_{b}, it is not special honest-verifier zero-knowledge.

The compression mechanism Σ_{c} uses an adaptation of Bulletproofs' folding technique [BCC+16; BBB+18], and thereby reduces the communication costs by roughly a factor two. For simplicity, let us assume that n is even; if it is not, the witness $\mathbf{x} \in \mathbb{G}^n$ can be appended with a zero. The witness $\mathbf{x} = (\mathbf{x}_L, \mathbf{x}_R)$ can be divided into a left half $\mathbf{x}_L \in \mathbb{G}^{n/2}$ and a right half $\mathbf{x}_R \in \mathbb{G}^{n/2}$. We will write $(0, \mathbf{y}), (\mathbf{y}, 0) \in \mathbb{G}^n$ for the n-dimensional vectors that contain $\mathbf{y} \in \mathbb{G}^{n/2}$ appended with $n/2$ zeros on the left and right, respectively.

The compression mechanism Σ_{c}, described in Protocol 2, now proceeds as follows. The prover sends $A = \Psi_n(0, \mathbf{x}_L)$ and $B = \Psi_n(\mathbf{x}_R, 0)$ to the verifier. Then, upon receiving a challenge $c \in \mathbb{Z}_q$, sampled uniformly at random by the verifier, the prover sends $\mathbf{z} = \mathbf{x}_L + c\mathbf{x}_R \in \mathbb{G}^{n/2}$ to the verifier, who confirms that $\Psi_n(c\mathbf{z}, \mathbf{z}) = A \cdot P^c \cdot B^{c^2}$. Note that the final response $\mathbf{z}$ is the combination of the left and right halves of the witness $\mathbf{x} = (\mathbf{x}_L, \mathbf{x}_R)$. For this reason, this procedure is also referred to as *folding*. Hence, at the cost of sending two $\mathbb{H}$-elements A and B,

the prover reduces the number of $\mathbb{G}$-elements it has to send from n down to $n/2$. Moreover, the compression mechanism Σ_{c} has three rounds, and is therefore a Σ-protocol. Further, it is 3-out-of-q special-sound and thus requires $q \geq 3$. The main properties of the compression mechanism Σ_{c} are summarized in Theorem 3.2.

Protocol 2 Compression Mechanism Σ_{c} for relation $\mathfrak{R}_n$.

Parameters:	$n = 2m \in \mathbb{N}$, prime q, and groups $(\mathbb{G}, +)$ and $(\mathbb{H}, \cdot)$ with exponent $q \geq 3$
Public Input:	$P \in \mathbb{H},\ \Psi_n \in \mathrm{Hom}(\mathbb{G}^n, \mathbb{H})$
Prover's Private Input:	$\mathbf{x}_L, \mathbf{x}_R \in \mathbb{G}^{n/2}$
Prover's Claim:	$\Psi_n(\mathbf{x}_L, \mathbf{x}_R) = P$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
$A = \Psi_n(0, \mathbf{x}_L)$		
$B = \Psi_n(\mathbf{x}_R, 0)$	$\xrightarrow{A,B}$	
		$c \leftarrow_R \mathbb{Z}_q$
	$\xleftarrow{c}$	
$\mathbf{z} = \mathbf{x}_L + c\mathbf{x}_R \in \mathbb{G}^{n/2}$		
	$\xrightarrow{\mathbf{z}}$	$\Psi_n(c\mathbf{z}, \mathbf{z}) \stackrel{?}{=} A \cdot P^c \cdot B^{c^2}$

Theorem 3.2 (Compression Mechanism)**.** *Let $n \in \mathbb{N}$ be even. Then, the compression mechanism Σ_{c} for relation $\mathfrak{R}_n$, described in Protocol 2, is a perfectly complete and 3-out-of-q special-sound Σ-protocol. Moreover, the communication costs are:*

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

Proof. **Completeness:** This property follows immediately.

Special-Soundness: Let $(A, B, c_1, \mathbf{z}_1)$, $(A, B, c_2, \mathbf{z}_2)$ and $(A, B, c_3, \mathbf{z}_3)$ be three accepting transcripts with common first message (A, B) and pairwise distinct challenges $c_1, c_2, c_3 \in \mathbb{Z}_q$. Further, let us define the Vandermonde matrix

$$V = \begin{pmatrix} 1 & 1 & 1 \\ c_1 & c_2 & c_3 \\ c_1^2 & c_2^2 & c_3^2 \end{pmatrix} \in \mathbb{Z}_q^{3\times 3},$$

with determinant $(c_2 - c_1)(c_3 - c_1)(c_3 - c_2) \in \mathbb{Z}_q$. Since the challenges $c_1, c_2, c_3 \in \mathbb{Z}_q$ are pairwise distinct, this determinant is non-zero and the matrix V is invertible. Let

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = V^{-1} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \in \mathbb{Z}_q^3$$

and $\bar{\mathbf{z}} = \sum_{i=1}^{3} a_i(c_i\mathbf{z}_i, \mathbf{z}_i) \in \mathbb{G}^n$. Then

$$\begin{aligned}\Psi(\bar{\mathbf{z}}) &= \Psi(c_1\mathbf{z}_1, \mathbf{z}_1)^{a_1} \cdot \Psi(c_2\mathbf{z}_2, \mathbf{z}_2)^{a_2} \cdot \Psi(c_3\mathbf{z}_3, \mathbf{z}_3)^{a_3} \\ &= A^{a_1+a_2+a_3} \cdot P^{c_1a_1+c_2a_2+c_3a_3} \cdot B^{c_1^2a_1+c_2^2a_2+c_3^2a_3} \\ &= P\,,\end{aligned}$$

i.e., $\bar{\mathbf{z}} \in \mathfrak{R}_n(P, \Psi_n)$ is a witness for statement (P, Ψ_n), which completes the proof.

□

3.2.2.1 Intermezzo: A General View on the Compression Mechanism.

Implicitly, our compression mechanism uses the following linear encoding, parameterized by an arbitrary challenge $c \in \mathbb{Z}_q$,

$$\mathrm{Enc}_c\colon \mathbb{G}^n \to \mathbb{G}^n, \quad \mathbf{x} = (\mathbf{x}_L, \mathbf{x}_R) \mapsto (0, \mathbf{x}_L) + c(\mathbf{x}_L, \mathbf{x}_R) + c^2(\mathbf{x}_R, 0)\,.$$

This encoding has three properties that are necessary and sufficient for our purposes:

1. For fixed $c \in \mathbb{Z}_q$, $\mathrm{Enc}_c(\mathbf{x})$ is a linear combination of $(0, \mathbf{x}_L)$, $\mathbf{x}$ and $(\mathbf{x}_R, 0)$. Hence, $\Psi_n\big(\mathrm{Enc}_c(\mathbf{x})\big)$ is a linear combination of $\Psi_n(0, \mathbf{x}_L)$, $\Psi_n(\mathbf{x})$ and $\Psi_n(\mathbf{x}_R, 0)$, i.e., $\Psi_n\big(\mathrm{Enc}_c(\mathbf{x})\big)$ is a linear combination of elements that are *independent* of $c \in \mathbb{Z}_q$.
2. For pairwise distinct $c_1, c_2, c_3 \in \mathbb{Z}_q$ and fixed $A, B, P \in H$, there exist efficiently computable $a_1, a_2, a_3 \in \mathbb{Z}_q$, such that $Q_1^{a_1} \cdot Q_2^{a_2} \cdot Q_3^{a_3} = P$, where $Q_i = A \cdot P^{c_i} \cdot B^{c_i^2}$ for $1 \le i \le 3$.
3. For fixed $c \in \mathbb{Z}_q$,

$$\mathrm{Enc}_c(\mathbf{x}_L, \mathbf{x}_R) = \big(c(\mathbf{x}_L + c\mathbf{x}_R), \mathbf{x}_L + c\mathbf{x}_R\big) \in \{(c\mathbf{z}, \mathbf{z}) \in \mathbb{G}^n : \mathbf{z} \in \mathbb{G}^{n/2}\}\,,$$

   i.e., the image $\mathrm{Enc}_c(\mathbb{G}^n)$ is a linear subspace of $\mathbb{G}^n$ of dimension $n/2$.

The first property allows the prover to send $A = \Psi_n(0, \mathbf{x}_L)$ and $B = \Psi_n(\mathbf{x}_R, 0)$ to the verifier *before* receiving the challenge $c \in \mathbb{Z}_q$, while still being able to efficiently compute a preimage of $A \cdot P^c \cdot B^{c^2}$, after receiving the challenge c. This property therefore implies completeness of $\Sigma_{\mathbf{c}}$. The second property of the encoding directly implies 3-out-of-q special-soundness. Finally, the third property shows that the preimage of $A \cdot P^c \cdot B^{c^2}$, requested by the verifier, lies in a subspace of dimension $n/2$. For this reason, the final message can be reduced to a vector of dimension $n/2$ instead of n, i.e., a reduction of roughly a factor two in the communication costs.

3.2.3 The Compressed Σ-Protocol

Analogously to the previous section, we observe that the final message $\mathbf{z} \in \mathbb{G}^{n/2}$ of compression mechanism $\Sigma_{\mathbf{c}}$ is a witness, but now with respect to relation $\mathfrak{R}_{n/2}$ and for statement $(Q, \Psi_{n/2})$, where $Q = A \cdot P^c \cdot B^{c^2} \in \mathbb{H}$ and

$$\Psi_{n/2}\colon \mathbb{G}^{n/2} \to \mathbb{H}, \quad \mathbf{x} \mapsto \Psi_n(c\mathbf{x}, \mathbf{x})\,.$$

Therefore, the final message can again be understood as a trivial interactive proof, but now for relation $\mathfrak{R}_{n/2}$ instead of $\mathfrak{R}_n$. To further reduce the communication costs, this message can be replaced by another appropriate instantiation of compression mechanism Σ_{c}. Continuing in this manner until the final message is of constant dimension, e.g., dimension 1, results in an interactive proof with a logarithmic (in n) communication complexity.

Our compressed Σ-protocol is thus the recursive composition of Σ-protocol Σ_{b} and appropriate instantiations of compression mechanism Σ_{c}. For this reason, let us define what it means for two interactive proofs to be composable. Informally, two interactive proofs $\Pi_1 = (\mathcal{P}_1, \mathcal{V}_1)$ and $\Pi_2 = (\mathcal{P}_2, \mathcal{V}_2)$, for relations $\mathfrak{R}_1$ and $\mathfrak{R}_2$ respectively, are composable if the verifier $\mathcal{V}_1$ accepts if and only if $\mathcal{P}_1$'s final message is a witness for some statement (that may depend on the protocol transcript) with respect to relation $\mathfrak{R}_2$. The following definition formalizes this notion of composability.

Definition 3.1 (Composable Interactive Proofs)**.** Let Π_1 be a $(2\mu_1 + 1)$-round interactive proof for relation $\mathfrak{R}_1$ and let Π_2 be a $(2\mu_2 + 1)$-round interactive proof for relation $\mathfrak{R}_2$. Then Π_1 and Π_2 are said to be *composable* if there exists an efficiently computable function ϕ, such that a transcript $(a_1, c_1, a_2, \ldots, c_{\mu_1}, a_{\mu_1+1})$ of $\Pi_1(x_1)$, on public input $x_1 \in \{0,1\}^*$, is accepting if and only if a_{μ_1+1} is a witness for statement $x_2 = \phi(x_1, a_1, c_1, \ldots, c_{\mu_1})$, i.e., $a_{\mu_1+1} \in \mathfrak{R}_2(x_2)$.

In this case, we write $\Pi_c = \Pi_2 \diamond \Pi_1$ for their composition, which proceeds as follows. On input statement-witness pair $(x_1; w_1)$, the prover and verifier run $\Pi_1(x_1; w_1)$ without the prover sending the final message, i.e., the prover obtains a complete protocol transcript $(a_1, c_1, \ldots, c_{\mu_1}, a_{\mu_1+1})$ and the verifier obtains a partial protocol transcript $(a_1, c_1, \ldots, c_{\mu_1})$. Both the prover and the verifier compute $x_2 = \phi(x, a_1, c_1, \ldots, c_{\mu_1})$ and run Π_2 on statement-witness pair $(x_2; a_{\mu_1+1}) \in \mathfrak{R}_2$. The verifier accepts if the verification for Π_2 succeeds.

The following lemma summarizes the main properties of the composition $\Pi_2 \diamond \Pi_1$ of two interactive proofs.

Lemma 3.1 (Composable Interactive Proofs)**.** *Let* Π_1 *and* Π_2 *be* composable *interactive proofs for relations* $\mathfrak{R}_1$ *and* $\mathfrak{R}_2$*, respectively. Moreover, let* $\mu_1, \mu_2 \in \mathbb{N}$ *such that* Π_1 *has* $2\mu_1 + 1$ *rounds and* Π_2 *has* $2\mu_2 + 1$ *rounds. Then:*

- $\Pi_2 \diamond \Pi_1$ *is an interactive proof for relation* $\mathfrak{R}_1$ *with* $2(\mu_1 + \mu_2) + 1$ *rounds;*
- *if* Π_1 *has completeness error* $\rho_1 \colon \{0,1\}^* \to [0,1]$ *and* Π_2 *has* constant *completeness error* $\rho_2 \in [0,1]$*, then* $\Pi_2 \diamond \Pi_1$ *has completeness error*

$$\rho \colon \{0,1\}^* \to [0,1], \quad x \mapsto (1 - \rho_2)\rho_1(x) + \rho_2\,;$$

- *if* Π_1 *is* $\mathbf{k}_1$*-out-of-*$\mathbf{N}_1$ *special-sound and* Π_2 *is* $\mathbf{k}_2$*-out-of-*$\mathbf{N}_2$ *special-sound, then* $\Pi_2 \diamond \Pi_1$ *is* $(\mathbf{k}_1, \mathbf{k}_2)$*-out-of-*$(\mathbf{N}_1, \mathbf{N}_2)$ *special-sound;*
- *if* Π_1 *is special honest-verifier zero-knowledge, then so is* $\Pi_2 \diamond \Pi_1$*.*

Proof. It follows by construction that $\Pi_2 \diamond \Pi_1$ is an interactive proof for relation $\mathfrak{R}_1$ with $2(\mu_1 + \mu_2) + 1$ rounds. So let us prove the remaining claims of the lemma.

Completeness: Let $(a_1, c_1, \ldots, c_{\mu_1}, a_{\mu_1+1})$ be a transcript output by Π_1 evaluated on statement-witness pair $(x_1; w_1) \in \mathfrak{R}_1$. Then, if the verifier of $\Pi_2 \diamond \Pi_1$ rejects, it must hold that either $a_{\mu+1}$ is not a witness for statement $x_2 = \phi(x_1, a_1, c_1, \ldots, c_{\mu_1})$ with respect to relation $\mathfrak{R}_2$, or the Π_2-verifier rejects the transcript output by $\Pi_2(x_2; a_{\mu+1})$. By the composability of Π_1 and Π_2 and the completeness of Π_1, the former happens with probability at most $\rho_1(x_1)$. By the completeness of Π_2, the latter event happens with probability at most ρ_2. Note that ρ_2 is assumed to be constant. Hence, the probability that the output of $\Pi_2 \diamond \Pi_1$, on input $(x_1; w_1) \in \mathfrak{R}_1$ is rejected, is at most

$$1 - (1 - \rho_1(x_1))\,(1 - \rho_2) = (1 - \rho_2)\rho_1(x) + \rho_2\,,$$

which proves the claimed completeness error.

Special-Soundness: Let us write $\mathbf{k}_1 = (k_1, \ldots, k_{\mu_1})$. Then any $(\mathbf{k}_1, \mathbf{k}_2)$-tree of accepting transcripts for $\Pi_2 \diamond \Pi_1$, on input $x \in \{0,1\}^*$, is the composition of $K_1 = \prod_{i=1}^{\mu_1} k_i$ accepting $(1, \ldots, 1, \mathbf{k}_2)$-trees $\mathcal{Y}_1, \ldots, \mathcal{Y}_{K_1}$.

For all $1 \leq j \leq K_1$, all transcripts in the tree $\mathcal{Y}_j$ have the same first $2\mu_1$ messages $(a_{1,j}, c_{1,j}, a_{2,j}, \ldots, c_{\mu_1,j})$ which, by the composability property, corresponds to a statement $x_{2,j} = \phi(x, a_{1,j}, c_{1,j}, a_{2,j}, \ldots, c_{\mu_1,j}) \in \{0,1\}^*$. By the special-soundness property of Π_2, a witness $w_{2,j} \in \mathfrak{R}_2(x_{2,j})$ can be computed efficiently from the $(1, \ldots, 1, \mathbf{k}_2)$-tree $\mathcal{Y}_j$ of accepting transcripts. Namely note that, by construction of $\Pi_2 \diamond \Pi_1$, $\mathcal{Y}_j$ contains a $\mathbf{k}_2$-tree of accepting transcripts for Π_2 on public input $x_{2,j}$. By the composability of Π_1 and Π_2, it follows that the transcript $(a_{1,j}, c_{1,j}, a_{2,j} \ldots, c_{\mu_1,j}, w_{2,j})$ must be an accepting transcript for Π_1 on input x, i.e., every $(1, \ldots, 1, \mathbf{k}_2)$-tree of accepting transcripts $\mathcal{Y}_j$ corresponds to an accepting transcript for interactive proof Π_1.

Moreover, the K_1 accepting transcripts corresponding to the trees $\mathcal{Y}_1, \ldots, \mathcal{Y}_{K_1}$ form a $\mathbf{k}_1$-tree of transcripts. By the special-soundness property of Π_1, a witness $w \in \mathfrak{R}_1(x)$ can be computed efficiently from this $\mathbf{k}_1$-tree of accepting transcripts for Π_1. Hence, a witness w can be computed efficiently from every $(\mathbf{k}_1, \mathbf{k}_2)$-tree of accepting transcripts, which proves the claimed special-soundness property for $\Pi_2 \diamond \Pi_1$.

SHVZK: The simulator $\mathcal{S}$ proceeds as follows. It samples $\mu_1 + \mu_2$ challenges for $\Pi_2 \diamond \Pi_1$ uniformly at random. Then it uses the first μ_1 challenges to run the simulator for Π_1 and obtains a transcript $(a_1, c_1, \ldots, c_{\mu_1}, a_{\mu_1+1})$. Subsequently, $\mathcal{S}$ runs Π_2 on input $(\phi(a_1, c_1, \ldots, c_{\mu_1}); a_{\mu_1+1}) \in \mathfrak{R}_2$ and obtains a transcript $(a'_1, c'_1, \ldots, c'_{\mu_2}, a'_{\mu_2+1})$ for Π_2, using the μ_2 challenges sampled before. The simulator then outputs the transcript

$$(a_1, c_1, \ldots, c_{\mu_1}, a'_1, c'_1, \ldots, c'_{\mu_2}, a'_{\mu_2+1})$$

for $\Pi_2 \diamond \Pi_1$ of length $2(\mu_1 + \mu_2) + 1$. It follows immediately that simulated transcripts have the same distribution as honestly generated ones, which completes the proof of the lemma.

□

Remark 3.2. Lemma 3.1 assumes that the completeness error of Π_2 is constant. In general, the completeness error is a function of the statement $x \in \{0,1\}^*$. However, this more general treatment would significantly complicate the analysis of $\Pi_2 \diamond \Pi_1$. More precisely, in this general treatment, the completeness error $\phi_2(x_2)$ of Π_2 is a function of the public statement x_2 used in the instantiation of Π_2 within $\Pi_2 \diamond \Pi_1$, and not a function of the input statement x_1 of $\Pi_2 \diamond \Pi_1$. Since we typically consider interactive proofs with constant completeness error, we have omitted this more general treatment.

Let us now return to Σ-protocol Σ_{b} and compression mechanism Σ_{c} and show that they are composable. To this end, let

$$\phi\colon \{0,1\}^* \to \{0,1\}^*, \quad (P, \Psi_n, A, c) \mapsto (A \cdot P^c, \Psi_n)\,.$$

Then a transcript $(A, c, \mathbf{z})$ for Σ_{b}, on public input (P, Ψ_n), is accepting if and only if $\mathbf{z}$ is a witness for $\phi(P, \Psi_n, A, c)$, i.e., Σ_{b} and Σ_{c} are indeed composable and their composition $\Sigma_{\mathsf{c}} \diamond \Sigma_{\mathsf{b}}$ is well defined. Similarly, by defining the function

$$\phi'\colon \{0,1\}^* \to \{0,1\}^*, \quad (P, \Psi_n, A, B, c) \mapsto \big(AP^cB^{c^2}, \Psi_{n/2}\colon \mathbf{x} \mapsto \Psi_n(c\mathbf{x}, \mathbf{x})\big)\,,$$

it follows that the compression mechanism Σ_{c} instantiated for relation $\mathfrak{R}_n$ is composable with Σ_{c} instantiated for $\mathfrak{R}_{n/2}$.

Our compressed Σ-protocol Σ_{comp} for relation $\mathfrak{R}_n$, i.e., the recursive composition of Σ_{b} and appropriate instantiations Σ_{c}, is therefore well defined. In every application of the compression mechanism, at the cost of sending two $\mathbb{H}$-elements, the dimension of the witness is reduced by a factor two. For simplicity, let us assume that the initial dimension n of the witness is a power of two, i.e., $n = 2^\mu$. If this is not the case, the witness can be appended with zeros. The optimal amount of recursions depends on the bit-size of $\mathbb{G}$- and $\mathbb{H}$-elements. For instance, reducing the witness dimension from two down to one, would reduce the communication costs by one element of $\mathbb{G}$, but it would increase the communication costs by 2 elements of $\mathbb{H}$; this is only beneficial if $\mathbb{G}$-elements are at least twice as large of $\mathbb{H}$-elements. For simplicity, we optimize the communication cost for instantiations where elements of $\mathbb{G}$ and $\mathbb{H}$ have the same bit-size, by continuing the compression until the witness has dimension two. However, we note that Σ_{comp} is easily adapted to other scenarios.

Altogether, the compressed Σ-protocol is therefore defined as

$$\Sigma_{\mathsf{comp}} = \underbrace{\Sigma_{\mathsf{c}} \diamond \cdots \diamond \Sigma_{\mathsf{c}}}_{\mu-1 \text{ times}} \diamond \Sigma_{\mathsf{b}}\,.$$

The main properties of Σ_{comp} follow (recursively) from Lemma 3.1 and are summarized in Theorem 3.3. For completeness, a full protocol description is given in Protocol 3.

Theorem 3.3 (Compressed Σ-Protocol)**.** *Let $n = 2^\mu$ for some $\mu \in \mathbb{N}$. Then the compressed Σ-protocol Σ_{comp} for relation $\mathfrak{R}_n$, described in Protocol 3, is perfectly complete, $(2, 3, \ldots, 3)$-out-of-$(q, \ldots, q)$ special-sound and special honest-verifier zero-knowledge (SHVZK). Moreover, it has $(2\mu + 1)$ communication rounds and the communication costs are:*

- $\mathcal{P} \to \mathcal{V}$: *2 elements of* $\mathbb{G}$ *and* $2\mu - 1$ *elements of* $\mathbb{H}$*;*
- $\mathcal{V} \to \mathcal{P}$: *$\mu$ elements of* $\mathbb{Z}_q$*.*

Protocol 3 Compressed Σ-Protocol Σ_{comp} for Relation $\mathfrak{R}_n$.

PARAMETERS:	$n = 2^\mu \in \mathbb{N}$, prime q, and groups $(\mathbb{G}, +)$ and $(\mathbb{H}, \cdot)$ with exponent $q \geq 3$
PUBLIC INPUT:	$P \in \mathbb{H},\ \Psi_n \in \mathrm{Hom}(\mathbb{G}^n, \mathbb{H})$
PROVER'S PRIVATE INPUT:	$\mathbf{x} \in \mathbb{G}^n$
PROVER'S CLAIM:	$\Psi_n(\mathbf{x}) = P$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
$\mathbf{r} \leftarrow_R \mathbb{G}^n$		
$A_0 = \Psi_n(\mathbf{r})$	$\xrightarrow{A_0}$	
		$c_1 \leftarrow_R \mathbb{Z}_q$
$\mathbf{x}^1 = (\mathbf{x}_L^1, \mathbf{x}_R^1) = \mathbf{r} + c_1\mathbf{x}$	$\xleftarrow{c_1}$	
		$Q_1 = A_0 P^{c_1}$
$A_1 = \Psi_n(0, \mathbf{x}_L^1)$		
$B_1 = \Psi_n(\mathbf{x}_R^1, 0)$	$\xrightarrow{A_1, B_1}$	
		$c_2 \leftarrow_R \mathbb{Z}_q$
	$\xleftarrow{c_2}$	
$\mathbf{x}^2 = \mathbf{x}_L^1 + c_2\mathbf{x}_R^1 \in \mathbb{G}^{n/2}$		$Q_2 = A_1 {Q_1}^{c_2} B_1^{c_2^2}$
$\vdots$	$\vdots$	$\vdots$
$A_{\mu-1} = \Psi_4(0, \mathbf{x}_L^{\mu-1})$		
$B_{\mu-1} = \Psi_4(\mathbf{x}_R^{\mu-1}, 0)$	$\xrightarrow{A_{\mu-1}, B_{\mu-1}}$	
		$c_\mu \leftarrow_R \mathbb{Z}_q$
	$\xleftarrow{c_\mu}$	
$\mathbf{z} = \mathbf{x}_L^{\mu-1} + c_\mu\mathbf{x}_R^{\mu-1} \in \mathbb{G}^2$		$Q_\mu = A_{\mu-1} {Q_{\mu-1}}^{c_\mu} B_{\mu-1}^{c_\mu^2}$
	$\xrightarrow{\mathbf{z}}$	
		$\Psi_2(\mathbf{z}) \stackrel{?}{=} Q_\mu$

The homomorphisms Ψ_ℓ, for $\ell \in \{2, 4, \ldots, 2^{\mu-1}\}$, are defined recursively:

$$\Psi_\ell \colon \mathbb{G}^\ell \to \mathbb{H}, \quad \mathbf{y} \mapsto \Psi_{2\ell}(c_{\mu-\log(\ell)+1}\mathbf{y}, \mathbf{y})\,.$$

3.2.4 Amortizing the Communication Costs

Various techniques from Σ-protocol theory are directly applicable to compressed Σ-protocols. As an example we show how to prove knowledge of many preim-

ages of the homomorphism Ψ_n with the same communication costs as before, i.e., amortizing the communication costs over many statement-witness pairs.

Protocol 4 describes the standard Σ-protocol Σ_{a} for this amortized setting, i.e., it is a Σ-protocol for relation

$$\mathfrak{R}_{\mathsf{A}} = \{(P_1, \ldots, P_s, \Psi_n; \mathbf{x}_1, \ldots, \mathbf{x}_s) : \Psi_n(\mathbf{x}_i) = P_i \ \ \forall i\} .$$

The properties of Σ_{a} are summarized in Theorem 3.4. In particular, note that the communication costs of Σ_{a}, while linear in n, are independent of the number of statements s. Moreover, Σ_{a} is $(s+1)$-out-of-q special sound and therefore requires $q \geq s+1$.

Theorem 3.4 (Amortized Σ-Protocol)**.** *The amortized Σ-protocol Σ_{a} for relation $\mathfrak{R}_{\mathsf{A}}$, described in Protocol 4, is perfectly complete, $(s+1)$-out-of-q special-sound and special honest-verifier zero-knowledge (SHVZK). Moreover, the communication costs are:*

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

Proof. **Completeness:** This property follows immediately.

Special-Soundness: Let $(A, c_0, \mathbf{z}_0), \ldots, (A, c_s, \mathbf{z}_s)$ be $s+1$ accepting transcripts with common first message A and pairwise distinct challenges $c_j \in \mathbb{Z}_q$. Further, let us define the Vandermonde matrix

$$V = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ c_0 & c_1 & \cdots & c_s \\ \vdots & \vdots & \ddots & \vdots \\ c_0^s & c_1^s & \cdots & c_s^s \end{pmatrix} \in \mathbb{Z}_q^{(s+1)\times(s+1)} ,$$

with determinant $\prod_{i<j}(c_j - c_i) \in \mathbb{Z}_q$. Since the challenges $c_j \in \mathbb{Z}_q$ are pairwise distinct, this determinant is nonzero and the matrix V is invertible. Let $(a_{j,i})_{0\leq j,i\leq s} = V^{-1}$, i.e., the $a_{j,i}$'s are the entries of the inverse of V, and, for $1 \leq \ell \leq s$, let $\bar{\mathbf{z}}_\ell = \sum_{j=0}^{s} a_{j,\ell}\mathbf{z}_j \in \mathbb{G}^n$. Then

$$\Psi(\bar{\mathbf{z}}_\ell) = A^{e_0} \cdot \prod_{i=1}^{s} P_i^{e_i} ,$$

where $e_0 = \sum_{j=0}^{s} a_{j,\ell}$ and $e_i = \sum_{j=0}^{s} a_{j,\ell} c_j^i$ for all $1 \leq i \leq s$. Hence, $e_i = 0$ for all $i \neq \ell$ and $e_\ell = 1$. It follows that $\Psi(\bar{\mathbf{z}}_\ell) = P_\ell$, i.e., $(\bar{\mathbf{z}}_1, \ldots, \bar{\mathbf{z}}_s)$ is a witness for statement $(P_1, \ldots, P_s)$, which proves the claimed special-soundness property.

SHVZK: Transcripts are simulated as follows. Sample $c \leftarrow_R \mathbb{Z}_q$ and $\mathbf{z} \leftarrow_R \mathbb{G}^n$ uniformly at random and set $A = \Psi(\mathbf{z}) \cdot \prod_{i=1}^{s} P_i^{-c^i}$. It is immediate that, if $(P_1, \ldots, P_s)$ admits a witness, then simulated transcripts $(A, c, \mathbf{z})$ have exactly the same distribution as honestly generated transcripts, which completes the proof of theorem.

□

Protocol 4 Amortized Σ-Protocol Σ_{a} for Relation $\mathfrak{R}_{\mathsf{A}}$.

PARAMETERS:	$n, s \in \mathbb{N}$, prime q, and groups $(\mathbb{G}, +)$ and $(\mathbb{H}, \cdot)$ with exponent $q \geq s+1$
PUBLIC INPUT:	$P_1, \ldots, P_s \in \mathbb{H},\ \Psi_n \in \mathrm{Hom}(\mathbb{G}^n, \mathbb{H})$
PROVER'S PRIVATE INPUT:	$\mathbf{x}_1, \ldots, \mathbf{x}_s \in \mathbb{G}^n$
PROVER'S CLAIM:	$P_i = \Psi_n(\mathbf{x}_i)\ \ \forall i$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
$\mathbf{r} \leftarrow_R \mathbb{G}^n$		
$A = \Psi_n(\mathbf{r})$	$\xrightarrow{A}$	
		$c \leftarrow_R \mathbb{Z}_q$
	$\xleftarrow{c}$	
$\mathbf{z} = \mathbf{r} + \sum_{i=1}^{s} c^i \mathbf{x}_i$		
	$\xrightarrow{\mathbf{z}}$	$\Psi_n(\mathbf{z}) \stackrel{?}{=} A \cdot \prod_{i=1}^{s} P_i^{c^i}$

The final message of Σ_{a} is a witness for relation $\mathfrak{R}_n$. Therefore, Σ-protocol Σ_{a} is amenable for our compression mechanism. This underlines our viewpoint that the compression mechanism is a strengthening of the well-established Σ-protocol theory. Let us write

$$\Sigma_{\mathsf{A}} = \underbrace{\Sigma_{\mathsf{c}} \diamond \cdots \diamond \Sigma_{\mathsf{c}}}_{\mu-1 \text{ times}} \diamond \Sigma_{\mathsf{a}}\,.$$

for the resulting compressed Σ-protocol for relation $\mathfrak{R}_{\mathsf{A}}$. Its properties are summarized in Theorem 3.5. Note that compression has reduced the communication complexity from linear down to logarithmic in n.

Theorem 3.5 (Amortized Compressed Σ-Protocol)**.** *Let $n = 2^\mu \in \mathbb{N}$. Then the amortized compressed Σ-protocol Σ_{A} for relation $\mathfrak{R}_{\mathsf{A}}$ is perfectly complete, unconditionally $(s+1, 3, \ldots, 3)$-out-of-$(q, \ldots, q)$ special-sound and special honest-verifier zero-knowledge (SHVZK). Moreover, it has $(2\mu + 1)$ communication rounds and the communication costs are:*

- *$\mathcal{P} \to \mathcal{V}$: 2 elements of $\mathbb{G}$ and $2\mu - 1$ elements of $\mathbb{H}$;*
- *$\mathcal{V} \to \mathcal{P}$: μ elements of $\mathbb{Z}_q$.*

3.2.5 Sublinear Communication in Constant Rounds

Towards reducing the dimension, the compression mechanism Σ_{c} divides the witness $\mathbf{x} = (\mathbf{x}_L, \mathbf{x}_R) \in \mathbb{G}^n$ in two parts $\mathbf{x}_L$ and $\mathbf{x}_R$. This approach has a straightforward generalization, where the witness is divided into k parts, i.e., $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_k)$. This generalization, denoted by Σ_{k}, is described in Protocol 5 and its properties are summarized in Theorem 3.6. In particular, Σ_{k} is $(2k-1)$-out-of-q special-sound and therefore requires $q \geq 2k - 1$.

Protocol 5 Generalized Compression Mechanism Σ_{k} with k-fold folding.

Parameters:	$n = k \cdot m \in \mathbb{N}$, prime q, and groups $(\mathbb{G}, +)$ and $(\mathbb{H}, \cdot)$ with exponent $q \geq 2k-1$
Public Input:	$P \in \mathbb{H},\ \Psi_n \in \mathrm{Hom}(\mathbb{G}^n, \mathbb{H})$
Prover's Private Input:	$\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_k) \in \mathbb{G}^n$
Prover's Claim:	$\Psi_n(\mathbf{x}_1, \ldots, \mathbf{x}_k) = P$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
$A_1 = \Psi_n(0, \ldots, 0, \mathbf{x}_1)$ $A_2 = \Psi_n(0, \ldots, 0, \mathbf{x}_1, \mathbf{x}_2)$ $\vdots$ $A_{2k-1} = \Psi_n(\mathbf{x}_k, 0, \ldots, 0)$	$\xrightarrow[A_{k+1}, \ldots, A_{2k-1}]{A_1, \ldots, A_{k-1}}$	
		$c \leftarrow_R \mathbb{Z}_q$
	$\xleftarrow{c}$	
$\mathbf{z} = \sum_{i=1}^{k} c^{i-1} \mathbf{x}_i \in \mathbb{G}^{n/k}$		
	$\xrightarrow{\mathbf{z}}$	$\Psi_n(c^{k-1}\mathbf{z}, \ldots, c\mathbf{z}, \mathbf{z})$
		$\stackrel{?}{=} P \cdot \prod_{i \neq k} A_i^{c^i}$

Theorem 3.6 (Generalized Compression Mechanism)**.** *The generalized compression mechanism Σ_k for relation $\mathfrak{R}_n$, described in Protocol 5, is a perfectly complete and $(2k-1)$-out-of-q special-sound Σ-protocol. Moreover, the communication costs are:*

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

Proof. **Completeness:** This property follows immediately.
Special-Soundness: Let

$$\begin{gathered}(A_1, \ldots, A_{k-1}, A_{k+1}, \ldots, A_{2k-1}, c_0, \mathbf{z}_0)\,, \\ \vdots \\ (A_1, \ldots, A_{k-1}, A_{k+1}, \ldots, A_{2k-1}, c_{2k-2}, \mathbf{z}_{2k-2})\,,\end{gathered}$$

be $2k-1$ accepting transcripts with common first message and pairwise distinct challenges $c_j \in \mathbb{Z}_q$. Further, let us define the Vandermonde matrix

$$V = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ c_0 & c_1 & \cdots & c_{2k-2} \\ \vdots & \vdots & \ddots & \vdots \\ c_0^{2k-2} & c_1^{2k-2} & \cdots & c_{2k-2}^{2k-2} \end{pmatrix} \in \mathbb{Z}_q^{(2k-1)\times(2k-1)}\,,$$

with determinant $\prod_{i<j}(c_j - c_i) \in \mathbb{Z}_q$. Since the challenges $c_j \in \mathbb{Z}_q$ are pairwise distinct, this determinant is nonzero and the matrix V is invertible. Let $\mathbf{a} = (a_0, \ldots, a_{2k-2})^T = V^{-1}\mathbf{e}_k$, where $\mathbf{e}_k$ is the k-th unit vector, i.e., $\mathbf{e}_k$'s k-th entry is 1 and its remaining entries are zero. Then

$$\bar{\mathbf{z}} = \sum_{i=0}^{2k-2} a_i(c_i^{k-1}\mathbf{z}_i, \ldots, c\mathbf{z}_i, \mathbf{z}_i) \in \mathbb{G}^n$$

is easily seen to satisfy $\Psi_n(\bar{\mathbf{z}}) = P$, i.e., it is a witness for statement (P, Ψ_n), which completes the proof.

□

Assuming, for simplicity, that n is a power k, i.e., $n = k^\mu$ for some $\mu \in \mathbb{N}$, allows this generalized compression mechanism to be applied recursively to our basic Σ-protocol Σ_{b}, resulting in the composition

$$\underbrace{\Sigma_{\mathsf{k}} \diamond \cdots \diamond \Sigma_{\mathsf{k}}}_{\mu-1 \text{ times}} \diamond \Sigma_{\mathsf{b}} .$$

This composite protocol has the following communications costs:

- $\mathcal{P} \to \mathcal{V}$: k elements of $\mathbb{G}$ and $(2k-2)\log_k(n) - 2k + 3$ elements of $\mathbb{H}$;
- $\mathcal{V} \to \mathcal{P}$: $\log_k(n)$ element of $\mathbb{Z}_q$.

If $\mathbb{G}$ and $\mathbb{H}$ elements are of the same size, the communication costs from prover to verifier are minimized for $k = 2$, resulting in exactly the compressed Σ-protocol Σ_{comp} from Section 3.2.3.

However, while the communication costs are minimized for $k = 2$, this instantiation does result in a logarithmic number of rounds. By contrast, taking $k = \sqrt{n}$, results in a 5-round interactive proof, with communication costs:

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

Hence, the resulting instantiation achieves a sublinear communication complexity in a constant number of rounds. Of course, in the non-interactive Fiat-Shamir mode the $k = 2$ instantiation with logarithmic communication might be preferable. Altogether the generalization of this section demonstrates a trade-off between the communication costs and the round complexity.

3.3 Proving Knowledge of *Short* Preimages

Certain cryptographic functions only admit desirable one-way properties with respect to "short" preimages, i.e., for these functions it is in general easy to find a preimage, but hard to find a short preimage of a given element. The most prominent examples are one-way functions based on lattice assumptions, but also certain one-way functions based on the strong-RSA assumption require preimages

to be short. In these cryptographic scenarios, the goal is therefore not to prove knowledge of just any preimage, but to prove knowledge of a *short* preimage. For this reason, towards accommodating lattice and strong-RSA based cryptographic platforms, we will generalize our compressed Σ-protocols.

To this end, let us assume that the group $\mathbb{G}$ is equipped with an absolute value (norm)

$$|\cdot| : \mathbb{G} \mapsto \mathbb{R}_{\geq 0}, \quad x \mapsto |x| \ .$$

Moreover, we assume a norm $\|\cdot\|_p$ on $\mathbb{G}^n$ to be defined as a natural extension of this absolute value. More precisely,

$$\|\cdot\|_p : \mathbb{G}^n \mapsto \mathbb{R}_{\geq 0}, \quad \mathbf{x} = (x_1, \ldots, x_n) \mapsto \|\mathbf{x}\|_p = \left(|x_1|^p + \cdot + |x_n|^p\right)^{1/p} \ .$$

for some $p \in \mathbb{R}_{\geq 1} \cup \{\infty\}$, where $p = \infty$ corresponds to the ℓ_∞-norm. The results in this section hold for any choice of p.

Then our goal is to construct a communication-efficient interactive proof for proving knowledge of a preimage of the homomorphism $\Psi_n \colon \mathbb{G}^n \to \mathbb{H}$ with bounded norm, i.e., an interactive proof for relation

$$\mathfrak{S}_n = \{(P, \Psi_n, \alpha; \mathbf{x}) : \Psi_n(\mathbf{x}) = P \wedge \|\mathbf{x}\|_p \leq \alpha\} \, . \tag{3.2}$$

As before, for technical reasons, we consider the homomorphism Ψ_n and the norm bound α to be part of the statement. However, if Ψ_n and α are clear from context, we will also refer to the group elements $P \in \mathbb{H}$ as statements, and thereby omit the more cumbersome notation (P, Ψ_n, α) of the statement.

In order to accommodate lattice based instantiations, a second generalization is required. Namely, thus far we assumed $\mathbb{G}$ and $\mathbb{H}$ to be abelian groups with exponent q, i.e., $\mathbb{Z}_q$-modules. However, in this section we allow $\mathbb{G}$ and $\mathbb{H}$ to be $\mathcal{R}$-modules for an arbitrary commutative ring $\mathcal{R}$. In fact, the homomorphisms encountered in lattice based cryptography are typically of the form $\Psi \colon \mathcal{R}^n \to \mathcal{R}_q^s$ for some ring $\mathcal{R}$ and $n, s, q \in \mathbb{N}$, where we recall that $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$.

Our approach is to generalize the interactive proofs of Section 3.2. For this reason, in Section 3.3.1, we construct a basic Σ-protocol for $\mathfrak{S}_n$. Subsequently, in Section 3.3.2, we adapt the compression mechanism to this more general scenario. Finally, in Section 3.3.3, we recursively compose these building blocks to obtain a compressed Σ-protocol for relation $\mathfrak{S}_n$.

3.3.1 Basic Σ-Protocol

The main difficulty in generalizing the basic Σ-protocol Σ_{b} of Section 3.2.1 comes from the fact that, in this generalization, witnesses have to be of small norm. In Σ_{b} the prover samples a vector $\mathbf{r} \in \mathbb{G}^n$, sends $\Psi_n(\mathbf{r})$ to the verifier and, after receiving a challenge c, it sends the response $\mathbf{z} = \mathbf{r} + c\mathbf{x}$. Since the vector $\mathbf{r}$ is sampled uniformly at random, responses $\mathbf{z}$ are also uniformly distributed, i.e., $\mathbf{r}$ masks $c\mathbf{x}$. For this reason, basic Σ-protocol Σ_{b} is *perfectly* special honest-verifier zero-knowledge (SHVZK). However, even if the witness $\mathbf{x}$ is of small norm, the same does not have to hold for responses $\mathbf{z}$. Hence, following the above approach, it cannot be guaranteed that extracted witnesses have small norm.

For this reason, in our generalization, we require the random vector $\mathbf{r}$ and challenges c to be of small norm too. This allows us to bound the norm of the prover's

final message $\mathbf{z} = \mathbf{r} + c\mathbf{x}$ and thereby also the norm of extracted witnesses. However, as a consequence, $\mathbf{r}$ is no longer uniformly distributed in $\mathbb{G}^n$ and therefore no longer perfectly masks $c\mathbf{x}$, i.e., the resulting protocol is not perfectly SHVZK. A first solution is to sample $\mathbf{r}$, such that the distribution of $\mathbf{z}$ is *statistically* close to a distribution independent of the witness $\mathbf{x}$. This will result in a Σ-protocol that is statistically SHVZK with responses $\mathbf{z}$ of bounded norm. Altogether, the random vector $\mathbf{r}$ should be sampled such that:

1. the norm of $\mathbf{r}$ is not much larger than that of the secret witness $\mathbf{x}$, but;
2. $\mathbf{r}$ still (statistically) masks $c\mathbf{x}$ for arbitrary challenges c.

A more efficient strategy was introduced by Lyubashevsky.[1] By using rejection sampling, he showed how to reduce the norm of responses $\mathbf{z} = \mathbf{r} + c\mathbf{x}$ significantly, while still achieving a meaningful zero-knowledge property [Lyu09; Lyu12]. In his approach, after receiving the challenge and computing the response $\mathbf{z} = \mathbf{r} + c\mathbf{x}$, the prover decides whether to abort or to send $\mathbf{z}$ to the verifier. Informally, this allows a prover to only complete protocol executions that do not reveal information about the secret witness $\mathbf{x}$. Rejection sampling does introduce an abort probability or completeness error to the protocol. Moreover, it weakens the special honest-verifier zero-knowledge property. More precisely, aborting transcripts of the form $(A, c, \perp)$ might reveal information about the secret witness $\mathbf{x}$. The resulting protocol is therefore only *non-abort* SHVZK. Fortunately, non-abort SHVZK is sufficient for most practical purposes. Namely, there exist generic approaches for transforming a non-abort SHVZK interactive proof into one that is SHVZK. Moreover, in the non-interactive Fiat-Shamir mode the prover only outputs non-aborting transcripts, so in this mode non-abort SHVZK indeed suffices.

In the following definition we abstract Lyubashevsky's rejection sampling by a distribution $\mathcal{D}$ and an algorithm $\mathcal{F}\colon \mathbb{G}^n \times \mathbb{G}^n \to \mathbb{G}^n \cup \{\perp\}$ such that:

1. elements $\mathbf{r}$ sampled from $\mathcal{D}$ (statistically) mask elements $\mathbf{v} \in V \subseteq \mathbb{G}^n$;
2. masked elements $\mathbf{v} + \mathbf{r}$ have bounded norm;
3. the abort probability $\Pr(\mathcal{F}(\mathbf{v}; \mathbf{r}) = \perp : \mathbf{r} \leftarrow_R \mathcal{D})$ is essentially independent of $\mathbf{v} \in V$.

Definition 3.2 ((V, δ)-Hiding and β-Bounded Sampling)**.** Let $\mathcal{R}$ be a commutative ring, $\mathbb{G}$ an $\mathcal{R}$-module and $n \in \mathbb{N}$. Let $V \subseteq \mathbb{G}^n$ and $\delta \in [0, 1]$. Further, Let $\mathcal{D}$ be an efficiently sampleable distribution with support in $\mathbb{G}^n$ and $\mathcal{F}$ a polynomial time algorithm. We say $(\mathcal{D}, \mathcal{F})$ is (V, δ)-*hiding* if there exists a polynomial time algorithm $\mathcal{F}'$ such that, for every $\mathbf{v} \in V$:

- $\mathcal{F}$, on input $\mathbf{v}$ and $\mathbf{r} \leftarrow_R \mathcal{D}$, outputs $\mathbf{v} + \mathbf{r}$ or $\perp$;
- $\mathcal{F}'$ outputs an element $\mathbf{z} \in \mathbb{G}^n$ or $\perp$,

such that the output distributions of $(\mathcal{D}, \mathcal{F})$ and $\mathcal{F}'$ have statistical distance at most δ, i.e.,

$$\Delta\left(\{\mathcal{F}(\mathbf{v}; \mathbf{r}) : \mathbf{r} \leftarrow_R \mathcal{D}\}, \{\mathcal{F}'\}\right) \leq \delta \quad \forall \mathbf{v} \in V\,.$$

[1] In fact, in the full version [Gro05] of [Gro03] predating Lyubashevsky's work, Groth already describes this rejection sampling strategy.

If $\delta = 0$, we say $(\mathcal{D}, \mathcal{F})$-is *perfectly* V-hiding. Further, we define

$$\rho := \min(\Pr(\mathcal{F}' = \perp) + \delta, 1) \in [0, 1]$$

to be the *abort probability* of $(\mathcal{D}, \mathcal{F})$.

Finally, let $\beta \in \mathbb{R}_{\geq 0}$. We say that $(\mathcal{D}, \mathcal{F})$ is β-*bounded* if

$$\Pr(\|\mathbf{z}\|_p \leq \beta : \mathbf{z} \leftarrow_R \mathcal{F}(\mathbf{v}; \mathbf{r}) \wedge \mathbf{r} \leftarrow_R \mathcal{D} \wedge \mathbf{z} \neq \perp) = 1 \quad \forall \mathbf{v} \in V\,.$$

Note that, if $(\mathcal{D}, \mathcal{F})$ is (V, δ)-hiding, the abort probability of $(\mathcal{D}, \mathcal{F})$ satisfies

$$\Pr(\mathcal{F}(\mathbf{v}; \mathbf{r}) = \perp : \mathbf{r} \leftarrow_R \mathcal{D}) \leq \Pr(\mathcal{F}' = \perp) + \delta \quad \forall \mathbf{v} \in V\,,$$

where the right-hand side is independent of $\mathbf{v}$.

Even with the use of rejection sampling, a knowledge extractor will in general only be able to extract preimages of Ψ_n with norm larger than the norm bound claimed by honest provers. More precisely, an extractor outputs preimages of norm at most $\tau \cdot \alpha$ for some $\tau \in \mathbb{R}_{\geq 0}$, while an honest prover claims to know a witness of norm at most α. The factor τ is referred as the *soundness slack* and introduces a relaxed notion of knowledge soundness and special-soundness. Interactive proofs for relation $\mathfrak{S}_n$ that satisfy this relaxed notion are said to be knowledge sound, or special-sound, with soundness slack τ. As long as it is hard to find preimages of norm $\tau \cdot \alpha$ this relaxation is still meaningful.

There are two sources introducing soundness slack. First, $\mathbf{z} = \mathbf{r} + c\mathbf{x}$ itself will in general already have larger norm than $\mathbf{x}$. Second, even worse, extracting a witness $\bar{\mathbf{z}}$ from two accepting transcripts, introduces additional slack. This slack is more difficult to control, as it depends on the (multiplicative) inverse of challenge differences.

In fact, differences of ring elements $c, c' \in \mathcal{R}$ are not necessarily invertible, let alone have short inverses. For this reason, we introduce a second relaxation to the knowledge soundness notion. Namely, for some fixed element $\zeta \in \mathcal{R}$, we allow the knowledge extractor to output a preimage of $P^\zeta \in \mathbb{H}$ instead of P. The element ζ is referred to as an *approximation factor*, and interactive proofs that admit such an extractor are said to be knowledge sound, or special-sound, with approximation factor ζ.

Let us now introduce the notion of an ζ-*exceptional subset*. This notion captures precisely the challenge sets required to guarantee the existence of a knowledge extractor with the above, relaxed, properties.

Definition 3.3 (ζ-Exceptional Subset)**.** Let $\mathcal{R}$ be a ring, $\zeta \in \mathcal{R}$, and $\mathcal{C} \subseteq \mathcal{R}$ be a set. We say $\mathcal{C}$ is a ζ-*exceptional subset* of $\mathcal{R}$ if for all $c, c' \in \mathcal{C}$ with $c \neq c'$ there exists an $a \in \mathcal{R}$ such that $a(c - c') = \zeta$. If $\mathcal{C}$ is a 1-exceptional subset of $\mathcal{R}$, we simply say that $\mathcal{C}$ is an *exceptional subset*.

Note that the 1-exceptional subsets are precisely the subsets of $\mathcal{R}$ with invertible nonzero differences, i.e., these are indeed the exceptional subsets of $\mathcal{R}$. Moreover, every subset of $\mathcal{R}$ is 0-exceptional.

Instantiating the Σ-protocol for relation $\mathfrak{S}_n$ with rejection sampling and a ζ-exceptional challenge set $\mathcal{C} \subseteq \mathcal{R}$ results in an interactive proof that is 2-out-of-$|\mathcal{C}|$

special-sound with soundness slack τ and approximation factor ζ, for some $\tau \in \mathbb{R}_{\geq 0}$. Before we present this Σ-protocol and its properties, we need to introduce some notation allowing us to specify the soundness slack τ. To this end, for ζ-exceptional subsets $\mathcal{C} \subseteq \mathcal{R}$ we define $w(\mathcal{C})$ and $\overline{w}(\mathcal{C}, \zeta)$ as follows:

$$\begin{aligned} w(\mathcal{C}) &= \max_{c \in \mathcal{C}, x \in \mathbb{G} \setminus \{0\}} \frac{|cx|}{|x|}\,, \\ \overline{w}(\mathcal{C}, \zeta) &= \max_{c \neq c' \in \mathcal{C}, x \in \mathbb{G} \setminus \{0\}} \frac{\left|\zeta(c-c')^{-1}x\right|}{|x|}\,. \end{aligned} \tag{3.3}$$

In the above, we assume that $\mathcal{R}$ does not have zero-divisors, i.e., the element $(c-c')^{-1}$ is well defined in the field of fractions of $\mathcal{R}$. Moreover, since $\mathcal{C}$ is ζ-exceptional it follows that $\zeta(c-c')^{-1} \in \mathcal{R}$.

The value $w(\mathcal{C})$ gives an upper bound on how much the norm of a vector $\mathbf{x} \in \mathbb{G}^n$ increases when multiplied by an element in $\mathcal{C}$, i.e., $w(\mathcal{C})$ is such that

$$\|c\mathbf{x}\|_p \leq w(\mathcal{C}) \cdot \|\mathbf{x}\|_p \quad \forall c \in \mathcal{C}, \quad \forall \mathbf{x} \in \mathbb{G}^n\,.$$

Note that if $\mathcal{R} = \mathbb{G} = \mathbb{Z}$, we simply have $w(\mathcal{C}) = \max\{|c| : c \in \mathcal{C} \subseteq \mathbb{Z}\}$.

The value $\overline{w}(\mathcal{C}, \zeta)$ gives an upper bound on how much the norm of a vector $\mathbf{x} \in \mathbb{G}^n$ increases when multiplied with the "approximation" $\zeta(c-c')^{-1}$ of a challenge difference inverse $(c-c')^{-1}$, i.e., $\overline{w}(\mathcal{C}, \zeta)$ is such that

$$\left\|\zeta(c-c')^{-1}\mathbf{x}\right\|_p \leq \overline{w}(\mathcal{C}, \zeta) \cdot \|\mathbf{x}\|_p \quad \forall \mathbf{x} \in \mathbb{G}^n, \quad \forall c, c' \in \mathcal{C} \text{ with } c \neq c'\,.$$

Now that all the required notation has been introduced, we are ready to present our Σ-protocol Π_{b} for relation $\mathfrak{S}_n$. This generalization of basic Σ-protocol Σ_{b} from Section 3.2 allows a prover to prove knowledge of a *short* preimage of P with respect to homomorphism Ψ_n. It is described in Protocol 6 and its main properties are summarized in Theorem 3.7.

Theorem 3.7 (Basic Σ-Protocol for Short Preimages)**.** *The Σ-protocol Π_{b} for relation*

$$\mathfrak{S}_n = \{(P, \Psi_n, \alpha; \mathbf{x}) : \Psi_n(\mathbf{x}) = P \wedge \|\mathbf{x}\|_p \leq \alpha\}\,,$$

described in Protocol 6, is complete with completeness error ρ, it is 2-out-of-$|\mathcal{C}|$ special-sound with soundness slack $2\overline{w}(\mathcal{C}, \zeta)\beta/\alpha$ and approximation factor ζ and it is δ-statistical non-abort special honest-verifier zero-knowledge (SHVZK). Moreover, the communication costs are:

- *$\mathcal{P} \to \mathcal{V}$: 1 element of $\mathbb{G}^n$ with norm at most β and 1 element of $\mathbb{H}$;*
- *$\mathcal{V} \to \mathcal{P}$: 1 element of $\mathcal{C} \subseteq \mathcal{R}$.*

Proof. **Completeness:** This property follows directly, because $(\mathcal{D}, \mathcal{F})$ is β-bounded and has abort probability ρ, and Ψ_n is an $\mathcal{R}$-module homomorphism.

Special-Soundness: Let $(A, c, \mathbf{z})$ and $(A, c', \mathbf{z}')$ be two accepting transcripts with common first message A and distinct challenges $c \neq c' \in \mathcal{C}$. Define $\bar{\mathbf{z}} = a(\mathbf{z} - \mathbf{z}') \in \mathbb{G}^n$, where a is such that $a(c-c') = \zeta \in \mathcal{R}$. Note that

such an a exists, because $\mathcal{C}$ is ζ-exceptional. Then it is easily seen that $\Psi(\bar{\mathbf{z}}) = P^{\zeta}$. Moreover,

$$\|\bar{\mathbf{z}}\|_p = \|a(\mathbf{z}-\mathbf{z}')\|_p \leq \overline{w}(\mathcal{C},\zeta)\,\|\mathbf{z}-\mathbf{z}'\|_p \leq 2\overline{w}(\mathcal{C},\zeta)\beta\,,$$

which proves the required norm bound on extracted preimages.

Non-Abort SHVZK: Transcripts are simulated as follows. Let $\mathcal{F}'$ be the algorithm corresponding to the V-hiding property of $(\mathcal{D},\mathcal{F})$. Given a challenge c, the simulator runs $\mathcal{F}'$. If $\mathcal{F}'$ outputs $\perp$, the simulator returns $(\perp, c, \perp)$. Else, the simulator sets $\mathbf{z} \leftarrow \mathcal{F}'$, computes the first message as $A = \Psi(\mathbf{z}) \cdot P^{-c}$ and outputs $(A, c, \mathbf{z})$. By the V-hiding property the output distributions of $\mathcal{F}$ and $\mathcal{F}'$ have statistical distance at most δ, and A can be derived deterministically from the values $c, \mathbf{z}$ and P. Therefore, δ-statistical non-abort SHVZK follows, which completes the proof of theorem.

□

Protocol 6 Basic Σ-Protocol Π_{b} for Relation $\mathfrak{S}_n$.

Parameters:	$n \in \mathbb{N}$, ring $\mathcal{R}$, $\mathcal{R}$-modules $(\mathbb{G},+)$ and $(\mathbb{H},\cdot)$, ζ-exceptional subset $\mathcal{C} \subseteq \mathcal{R}$ with $\lvert\mathcal{C}\rvert \geq 2$, $V = \{c\mathbf{x} \in \mathbb{G}^n : \|\mathbf{x}\|_p \leq \alpha \wedge c \in \mathcal{C}\}$ and (V,δ)-hiding and β-bounded pair $(\mathcal{D},\mathcal{F})$ with abort probability $\rho \in [0,1]$
Public Input:	$P \in \mathbb{H},\ \Psi_n \in \mathrm{Hom}(\mathbb{G}^n,\mathbb{H}),\ \alpha \in \mathbb{R}_{\geq 0}$
Prover's Private Input:	$\mathbf{x} \in \mathbb{G}^n$
Prover's Claim:	$\Psi_n(\mathbf{x}) = P \wedge \|\mathbf{x}\|_p \leq \alpha$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
$\mathbf{r} \leftarrow_R \mathcal{D}$		
$A = \Psi_n(\mathbf{r})$	$\xrightarrow{A}$	
		$c \leftarrow_R \mathcal{C} \subseteq \mathcal{R}$
	$\xleftarrow{c}$	
If $\mathcal{F}(c\mathbf{x};\mathbf{r}) = \perp$: Abort		
Else: $\mathbf{z} = \mathbf{r} + c\mathbf{x}$	$\xrightarrow{\mathbf{z}}$	$\|\mathbf{z}\|_p \overset{?}{\leq} \beta$
		$\Psi_n(\mathbf{z}) \overset{?}{=} A \cdot P^c$

Remark 3.3. The set V in Σ-protocol Π_{b} depends on the public parameter α. Therefore, the set V, the distribution-algorithm pair $(\mathcal{D},\mathcal{F})$ and its properties should technically be parameterized by α. However, to avoid an even more cumbersome notation, we decided to omit this parameterization.

Remark 3.4. Our definitions require the approximation factor ζ to be a fixed element of the ring $\mathcal{R}$. However, in some settings it is beneficial to allow for arbitrary approximation factors in some fixed subset $\Omega \subseteq \mathcal{R}$. In this case the

extractor does not output a preimage of P^{ζ}, but it outputs a preimage of P^{ω} for some $\omega \in \Omega$. Hence, the extractor is free to choose an approximation factor $\omega \in \Omega$. In some instantiations, this relaxation allows for a smaller soundness slack. However, it introduces additional difficulties when composing the Σ-protocol with other protocols, such as a compression mechanism. These difficulties can be handled, but in most settings the required adjustments negate the benefits of this additional relaxation, which is why we do not consider it further.

The Σ-protocol Σ_{b} of Section 3.2 is actually a specific instantiation of Σ-protocol Π_{b}. It can be derived by setting $V = \mathbb{G}^n$, $\mathcal{C} = \mathbb{Z}_q$, $\mathcal{D}$ as the uniform distribution over $\mathbb{G}^n$ and

$$\mathcal{F}\colon \mathbb{G}^n \times \mathbb{G}^n, \quad (\mathbf{v}; \mathbf{r}) \mapsto \mathbf{v} + \mathbf{r}\,.$$

Then $(\mathcal{D}, \mathcal{F})$ is perfectly $\mathcal{V}$-hiding with abort probability 0. Finally, note that, since this instantiation does not require the witness to be small, we do not need to consider a norm. Hence, Π_{b} is indeed a generalization of Σ_{b}.

3.3.1.1 From Non-Abort SHVZK to SHVZK

Rejection sampling, and therefore also our abstraction of rejection sampling, in general does not allow to simulate the first message for aborting transcripts (see, e.g., the simulator in the proof of Theorem 3.7). For this reason, Σ-protocol Π_{b} provides only non-abort SHVZK. In the non-interactive Fiat-Shamir mode this is not a problem, because the prover simply does not output aborting transcripts. But, when using the Σ-protocol interactively, we have to apply an additional measure in order to guarantee SHVZK. In [DOT+21] it was recently shown how to deal with this problem for the purpose of constructing a lattice-based multi-signature scheme. However, this is a more challenging task than enhancing an interactive proof from non-abort SHVZK to standard SHVZK. Therefore, their solution requires to either rely on random oracles or trapdoor commitments. We observe that in our case to go from non-abort SHVZK to standard SHVZK, it suffices to replace the first message by a statistically hiding and computationally binding commitment scheme. The cost of this transformation is that the special-soundness property is only preserved under the (computational) assumption that the commitment scheme is binding, i.e., the resulting protocol is only *computationally* special-sound. Alternatively, one could instantiate this approach with a computationally hiding and statistically binding commitment scheme. This would preserve the *unconditional* special-soundness, but would result in *computational* SHVZK.

Lemma 3.2 (Non-Abort SHVZK to SHVZK)**.** *Let* Π *be a complete,* 2*-out-of-N special-sound and non-abort special honest-verifier zero-knowledge* Σ*-protocol. Further, let* COM *be a statistically hiding and computationally binding commitment scheme. Then there exists a* Σ*-protocol* Π′ *that is complete,* computationally 2*-out-of-N special-sound, under the assumption that the commitment scheme is binding, and* special honest-verifier zero-knowledge*.*

Proof. The idea is simply to replace the first message of the protocol by a commitment to the first message. More precisely, Σ-protocol Π′ proceeds as follows. First, the prover computes the first message A according to Π. Further, the prover

samples randomness γ for the commitment scheme and sends $C = \text{COM}(A;\gamma)$ to the Verifier, who responds with a challenge c. In the last round the prover computes z according to the second prover's message in Π, depending on A and the challenge c. If Π does not abort, the prover sends A, γ, and z to the verifier. The verifier accepts if A, γ is a valid opening of the commitment C, and (A, c, z) is an accepting transcript for Π. It is left to show that Π' indeed satisfies the required properties.

Completeness: This property follows immediately.

Computational Special-Soundness: Let (C, c, A, γ, z) and (C, c', A', γ', z') be two accepting transcripts. Then, either we have that $A' = A$ and we can rely on the 2-out-of-N special-soundness of Π, or the prover broke the computational binding property of COM by finding two valid and distinct openings A, γ and A', γ' for commitment C.

SHVZK: Given a challenge c, the simulator runs the simulator for the underlying protocol Π. If the underlying simulator returns $(\bot, c, \bot)$, the simulator samples randomness γ and outputs $(\text{COM}(0;\gamma), c, \bot)$. If the underlying simulator returns (A, c, z), then the simulator samples randomness γ and outputs $(\text{COM}(A;\gamma), c, A, \gamma, z)$. SHVZK follows by the statistical hiding property of COM and the non-abort SHVZK property of the underlying protocol Π.

□

Remark 3.5. Applying this transformation to Σ-protocol Π_{b}, the prover does not have to send A, because the verifier can first compute A as $\Psi(\mathbf{z}) \cdot P^{-c}$ and then verify if A, γ is indeed a valid opening of commitment C. Therefore, if A has a larger bit-size than the commitment C and its randomness γ combined, the transformation of Π_{b} actually has smaller communication costs than the original Σ-protocol Π_{b}.

3.3.2 A Compression Mechanism

As before, we observe that the final message of Σ-protocol Π_{b} is a witness for statement $(A \cdot P^c, \Psi_n, \beta)$ with respect to relation

$$\mathfrak{S}_n = \{(P, \Psi_n, \alpha; \mathbf{x}) : \Psi_n(\mathbf{x}) = P \wedge \|\mathbf{x}\|_p \leq \alpha\}\,.$$

Moreover, the verifier accepts if and only if the final message is a valid witness. Hence, the final message is a trivial interactive proof for relation $\mathfrak{S}_n$, and our goal is to replace this trivial interactive proof by a more efficient one. This more efficient interactive proof does not have to be zero-knowledge.

The compression mechanism is thus an interactive proof for relation $\mathfrak{S}_n$ that is not zero-knowledge. Since it is not required to be zero-knowledge, rejection sampling can be avoided. In particular, there is no need for a (V, δ)-hiding and β-bounded distribution-algorithm pair $(\mathcal{D}, \mathcal{F})$. For this reason, the compression mechanism Π_{c} for $\mathfrak{S}_n$ is a straightforward adaptation of compression mechanism Σ_{c} of Section 3.2.2. It is presented in Protocol 7 and its properties are summarized in Theorem 3.8. Note that, as before, the compression mechanism reduces the dimension of the witness from n down to $n/2$. However, in contrast to Section 3.2.2, here compression comes at the cost of increasing the soundness slack.

Theorem 3.8 (Compression Mechanism)**.** *The compression mechanism* Π_{c} *for relation* $\mathfrak{S}_n$*, described in Protocol 7, is a perfectly complete and* 3*-out-of-q special-sound* Σ*-protocol with soundness slack*[2]

$$6 \cdot \overline{w}(\mathcal{C},\zeta)^3 \cdot \big(w(\mathcal{C})^2 + w(\mathcal{C})^3\big) \cdot \big(1 + w(\mathcal{C})^p\big)^{1/p}$$

and approximation factor ζ^3*. Moreover, the communication costs are:*

- $\mathcal{P} \to \mathcal{V}$*:* 1 *element of* $\mathbb{G}^{n/2}$ *with norm at most* $\big(1 + w(\mathcal{C})\big)\alpha$ *and* 2 *elements of* $\mathbb{H}$*;*
- $\mathcal{V} \to \mathcal{P}$*:* 1 *element of* $\mathcal{C} \subseteq \mathcal{R}$*.*

Protocol 7 Compression Mechanism Π_{c} for relation $\mathfrak{S}_n$.

PARAMETERS:	$n \in 2\mathbb{N}$, ring $\mathcal{R}$, $\mathcal{R}$-modules $(\mathbb{G},+)$ and $(\mathbb{H},\cdot)$, ζ-exceptional subset $\mathcal{C} \subseteq \mathcal{R}$ with $\lvert\mathcal{C}\rvert \geq 3$
PUBLIC INPUT:	$P \in \mathbb{H},\ \Psi_n \in \mathrm{Hom}(\mathbb{G}^n, \mathbb{H}),\ \alpha \in \mathbb{R}_{\geq 0}$
PROVER'S PRIVATE INPUT:	$\mathbf{x} = (\mathbf{x}_L, \mathbf{x}_R) \in \mathbb{G}^n$
PROVER'S CLAIM:	$\Psi_n(\mathbf{x}_L, \mathbf{x}_R) = P \wedge \lVert\mathbf{x}\rVert_p \leq \alpha$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
$A = \Psi_n(0, \mathbf{x}_L)$ $B = \Psi_n(\mathbf{x}_R, 0)$	$\xrightarrow{A,B}$	
		$c \leftarrow_R \mathcal{C} \subseteq \mathcal{R}$
	$\xleftarrow{c}$	
$\mathbf{z} = \mathbf{x}_L + c\mathbf{x}_R \in \mathbb{G}^{n/2}$		
	$\xrightarrow{\mathbf{z}}$	$\lVert\mathbf{z}\rVert_p \overset{?}{\leq} \big(1 + w(\mathcal{C})\big) \cdot \alpha$ $\Psi_n(c\mathbf{z}, \mathbf{z}) \overset{?}{=} A \cdot P^c \cdot B^{c^2}$

Proof. Recall that

$$\lVert\cdot\rVert_p : \mathbb{G}^n \to \mathbb{R}_{\geq 0}, \quad \mathbf{x} = (x_1, \ldots, x_n) \mapsto \lVert\mathbf{x}\rVert_p = \big(\lvert x_1\rvert^p + \cdot + \lvert x_n\rvert^p\big)^{1/p} .$$

for some $p \in \mathbb{R}_{\geq 1} \cup \{\infty\}$, and that $w(\mathcal{C})$ and $\overline{w}(\mathcal{C},\zeta)$ are independent of the dimension n. Let us now prove that Π_{c} has the desired completeness and special-soundness properties.

Completeness: This property follows, since Ψ_n is a homomorphism and

$$\begin{aligned} \lVert\mathbf{z}\rVert_p = \lVert\mathbf{x}_L + c\mathbf{x}_R\rVert_p &\leq \lVert\mathbf{x}_L\rVert_p + w(\mathcal{C}) \lVert\mathbf{x}_R\rVert_p \\ &\leq \big(1 + w(\mathcal{C})\big) \lVert\mathbf{x}\rVert_p \\ &\leq \big(1 + w(\mathcal{C})\big)\alpha\,, \end{aligned}$$

[2]For $p = \infty$, we define $\big(1 + w(\mathcal{C})^p\big)^{1/p} = w(\mathcal{C})$.

where we use that

$$\|\mathbf{x}_L\|_p \leq \|(\mathbf{x}_L, \mathbf{x}_R)\|_p = \|\mathbf{x}\|_p \quad \text{and} \quad \|\mathbf{x}_R\|_p \leq \|(\mathbf{x}_L, \mathbf{x}_R)\|_p = \|\mathbf{x}\|_p \,.$$

Special-Soundness: Let $(A, B, c_1, \mathbf{z}_1)$, $(A, B, c_2, \mathbf{z}_2)$ and $(A, B, c_3, \mathbf{z}_3)$ be three accepting transcripts with common first message (A, B) and pairwise distinct challenges $c_1, c_2, c_3 \in \mathcal{C}$. Further, let

$$(a_1, a_2, a_3) = \left(c_3^2 - c_2^2, c_1^2 - c_3^2, c_2^2 - c_1^2\right) \,,$$

then

$$\begin{pmatrix} 1 & 1 & 1 \\ c_1 & c_2 & c_3 \\ c_1^2 & c_2^2 & c_3^2 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} = \tilde{c} \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} ,$$

where $\tilde{c} = (c_1 - c_2)(c_1 - c_3)(c_2 - c_3) \in \mathcal{R}$.

Let a be such that $a \cdot \tilde{c} = \zeta^3$, which exists because $\mathcal{C}$ is ζ-exceptional, and let

$$\bar{\mathbf{z}} = a \cdot \sum_{i=1}^{3} a_i (c_i \mathbf{z}_i, \mathbf{z}_i) \in \mathbb{G}^n \,.$$

Then

$$\begin{aligned} \Psi(\bar{\mathbf{z}}) &= \left(\Psi(c_1\mathbf{z}_1, \mathbf{z}_1)^{a_1} \cdot \Psi(c_2\mathbf{z}_2, \mathbf{z}_2)^{a_2} \cdot \Psi(c_3\mathbf{z}_3, \mathbf{z}_3)^{a_3}\right)^a \\ &= \left(A^{a_1+a_2+a_3} \cdot P^{c_1a_1+c_2a_2+c_3a_3} \cdot B^{c_1^2a_1+c_2^2a_2+c_3^2a_3}\right)^a \\ &= P^{a\cdot\tilde{c}} = P^{\zeta^3} \,, \end{aligned}$$

i.e., $\bar{\mathbf{z}}$ is a preimage of P^{ζ^3} with respect to homomorphism Ψ_n. Let us now bound the norm of the extracted preimage $\bar{\mathbf{z}}$. It holds that

$$\begin{aligned} \|\bar{\mathbf{z}}\|_p &\leq \overline{w}(\mathcal{C}, \zeta)^3 \cdot \sum_{i=1}^{3} \|a_i(c_i\mathbf{z}_i, \mathbf{z}_i)\|_p \\ &\leq \overline{w}(\mathcal{C}, \zeta)^3 \cdot \sum_{i=1}^{3} 2 \cdot w(\mathcal{C})^2 \cdot \|(c_i\mathbf{z}_i, \mathbf{z}_i)\|_p \,. \end{aligned}$$

Now observe that, for all i,

$$\|(c_i\mathbf{z}_i, \mathbf{z}_i)\|_p^p = \|c_i\mathbf{z}_i\|_p^p + \|\mathbf{z}_i\|_p^p \leq w(\mathcal{C})^p \|\mathbf{z}_i\|_p^p + \|\mathbf{z}_i\|_p^p = \left(1 + w(\mathcal{C})^p\right) \|\mathbf{z}_i\|_p^p \,.$$

Hence,

$$\begin{aligned} \|\bar{\mathbf{z}}\|_p &\leq 2 \cdot w(\mathcal{C})^2 \cdot \overline{w}(\mathcal{C}, \zeta)^3 \cdot \sum_{i=1}^{3} \left(1 + w(\mathcal{C})^p\right)^{1/p} \cdot \|\mathbf{z}_i\|_p \\ &\leq 6 \cdot w(\mathcal{C})^2 \cdot \overline{w}(\mathcal{C}, \zeta)^3 \cdot \left(1 + w(\mathcal{C})\right) \cdot \left(1 + w(\mathcal{C})^p\right)^{1/p} \cdot \alpha \\ &= 6 \cdot \overline{w}(\mathcal{C}, \zeta)^3 \cdot \left(w(\mathcal{C})^2 + w(\mathcal{C})^3\right) \cdot \left(1 + w(\mathcal{C})^p\right)^{1/p} \cdot \alpha \,, \end{aligned}$$

which proves the required norm bound and completes the proof. □

3.3.3 The Compressed Σ-Protocol for Short Preimages

It is easily seen that Σ-protocol Π_{b} and compression mechanism Π_{c} are composable (Definition 3.1). Assuming that $n = 2^{\mu}$ for some $\mu \in \mathbb{N}$, the compressed Σ-protocol Π_{comp} for proving knowledge of a short preimage is thus defined as the recursive composition

$$\Pi_{\mathsf{comp}} = \underbrace{\Pi_{\mathsf{c}} \diamond \cdots \diamond \Pi_{\mathsf{c}}}_{\mu \text{ times}} \diamond \Pi_{\mathsf{b}} \,.$$

For simplicity, we applied the compression mechanism μ times, i.e., until the dimension of the witness has been reduced to 1. However, depending on bit-size of elements in the $\mathcal{R}$-modules $\mathbb{G}$ and $\mathbb{H}$, a different number of compressions might be required to minimize the communication costs.

Most properties of Π_{comp} follow directly from Lemma 3.1. What remains is to determine the soundness slack and approximation factor of the recursive composition Π_{comp}. However, it is easily seen that the soundness slack and approximation factors accumulate multiplicatively under recursive composition. In general, if Π_1 has soundness slack τ_1 and approximation factor ζ_1 and Π_2 has soundness slack τ_2 and approximation factor ζ_2, then $\Pi_2 \diamond \Pi_1$ has soundness slack $\tau_1 \cdot \tau_2$ and approximation factor $\zeta_1 \cdot \zeta_2$.

Protocol 8 provides a complete description of compressed Σ-protocol Π_{comp} for relation $\mathfrak{S}_n$, its properties are summarized in Theorem 3.9.

Note that the soundness slack τ_n grows exponentially in the number of rounds and therefore polynomially in the dimension n of the secret witness $\mathbf{x} \in \mathbb{G}^n$. Since the interactive proof Π_{comp} has to be instantiated such that it is hard to find preimages of norm at most $\tau_n \cdot \alpha$, even though the prover claims to know a preimage of norm at most α, larger soundness slack typically implies larger protocol parameters and larger communication costs. For this reason, while the number of elements communicated is logarithmic in the dimension n, the communication costs of Π_{comp}, expressed in the number of bits transmitted, are typically not logarithmic in n. For instance, in Section 5.6, we show that an appropriate lattice-instantiation of compressed Σ-protocol Π_{comp} has *polylogarithmic* communication complexity.

Theorem 3.9 (Compressed Σ-Protocol for Short Preimages)**.** *Let $n = 2^{\mu}$ for some $\mu \in \mathbb{N}$. Then the compressed Σ-protocol*

$$\Pi_{comp} = \underbrace{\Pi_{c} \diamond \cdots \diamond \Pi_{c}}_{\mu \text{ times}} \diamond \Pi_{b} \,,$$

for relation $\mathfrak{S}_n$, described in Protocol 8, is complete with completeness error ρ, it is $(2, 3, \ldots, 3)$-out-of-$(|\mathcal{C}|, \ldots, |\mathcal{C}|)$ special-sound with soundness slack

$$\tau = 2 \cdot 6^{\mu} \cdot \overline{w}(\mathcal{C}, \zeta)^{3\mu+1} \cdot \left(w(\mathcal{C})^2 + w(\mathcal{C})^3\right)^{\mu} \cdot \left(1 + w(\mathcal{C})^p\right)^{\mu/p} \cdot \beta/\alpha$$

and approximation factor $\zeta^{3\mu+1}$, and it is δ-statistical non-abort special honest-verifier zero-knowledge.

Moreover, it has $2\mu+3$ communication rounds and the communication costs are:

Protocol 8 Compressed Σ-Protocol Π_{comp} for Relation $\mathfrak{S}_n$.

Parameters:	$n = 2^\mu \in \mathbb{N}$, ring $\mathcal{R}$, $\mathcal{R}$-modules $(\mathbb{G}, +)$ and $(\mathbb{H}, \cdot)$, ζ-exceptional subset $\mathcal{C} \subseteq \mathcal{R}$ with $\lvert\mathcal{C}\rvert \geq 3$, $V = \{c\mathbf{x} \in \mathbb{G}^n : \lVert\mathbf{x}\rVert_p \leq \alpha \wedge c \in \mathcal{C}\}$ and (V, δ)-hiding and β-bounded pair $(\mathcal{D}, \mathcal{F})$ with abort probability $\rho \in [0, 1]$
Public Input:	$P \in \mathbb{H},\ \Psi_n \in \mathrm{Hom}(\mathbb{G}^n, \mathbb{H}),\ \alpha \in \mathbb{R}_{\geq 0}$
Prover's Private Input:	$\mathbf{x} \in \mathbb{G}^n$
Prover's Claim:	$\Psi_n(\mathbf{x}) = P \wedge \lVert\mathbf{x}\rVert_p \leq \alpha$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
$\mathbf{r} \leftarrow_R \mathcal{D}$		
$A_0 = \Psi_n(\mathbf{r})$	$\xrightarrow{A_0}$	
		$c_0 \leftarrow_R \mathcal{C}$
If $\mathcal{F}(c_0\mathbf{x}; \mathbf{r}) = \perp$: Abort	$\xleftarrow{c_0}$	
Else:		
$\mathbf{x}^1 = (\mathbf{x}_L^1, \mathbf{x}_R^1) = \mathbf{r} + c_0\mathbf{x}$		
		$Q_1 = A_0 P^{c_0}$
$A_1 = \Psi_n(0, \mathbf{x}_L^1)$		
$B_1 = \Psi_n(\mathbf{x}_R^1, 0)$	$\xrightarrow{A_1, B_1}$	
		$c_1 \leftarrow_R \mathcal{C}$
	$\xleftarrow{c_1}$	
$\mathbf{x}^2 = \mathbf{x}_L^1 + c_1\mathbf{x}_R^1 \in \mathbb{G}^{n/2}$		$Q_2 = A_1 Q_1^{c_1} B_1^{c_1^2}$
$\vdots$	$\vdots$	$\vdots$
$A_\mu = \Psi_2(0, \mathbf{x}_L^\mu)$		
$B_\mu = \Psi_2(\mathbf{x}_R^\mu, 0)$	$\xrightarrow{A_\mu, B_\mu}$	
	$\xleftarrow{c_\mu}$	$c_\mu \leftarrow_R \mathcal{C}$
$z = \mathbf{x}_L^\mu + c_\mu\mathbf{x}_R^\mu \in \mathbb{G}$		$Q_\mu = A_\mu Q_\mu^{c_\mu} B_\mu^{c_\mu^2}$
	$\xrightarrow{z}$	
		$\lVert z\rVert_p \stackrel{?}{\leq} \left(1 + w(\mathcal{C})\right)^\mu \cdot \beta$
		$\Psi_1(z) \stackrel{?}{=} Q_\mu$

The homomorphisms Ψ_ℓ, for $\ell \in \{1, 2, 4, \ldots, 2^{\mu-1}\}$, are defined recursively:

$$\Psi_\ell \colon \mathbb{G}^\ell \to \mathbb{H}, \quad \mathbf{y} \mapsto \Psi_{2\ell}(c_{\mu - \log(\ell)}\mathbf{y}, \mathbf{y})\,.$$

- $\mathcal{P} \to \mathcal{V}$: 1 *element of* $\mathbb{G}$ *with norm at most* $\big(1+w(\mathcal{C})\big)^{\mu}\beta$ *and* $2\mu+1$ *elements of* $\mathbb{H}$*;*
- $\mathcal{V} \to \mathcal{P}$: $\mu+1$ *element of* $\mathcal{C} \subseteq \mathcal{R}$*.*

3.3.4 Enlarging the Challenge Set

In Chapter 6, we show that **k**-out-of-**N** special-sound interactive proofs are knowledge sound with knowledge error

$$\mathrm{Er}(\mathbf{k};\mathbf{N}) = 1 - \prod_{i=1}^{\mu}\left(1 - \frac{k_i - 1}{N_i}\right) ,$$

where $\mathbf{k} = (k_1, \ldots, k_\mu)$ and $\mathbf{N} = (N_1, \ldots, N_\mu)$. In fact, **k**-out-of-**N** special-sound interactive proofs typically admit a cheating strategy with success probability $\mathrm{Er}(\mathbf{k};\mathbf{N})$, i.e., this knowledge error is optimal. If this knowledge error is not small enough it must be reduced.

A standard approach for reducing the knowledge error is to run t instances of the same interactive proof in parallel. The verifier accepts if and only if the prover succeeds in all t instances. In Section 6.5, we show that this approach indeed reduces the knowledge error from $\mathrm{Er}(\mathbf{k};\mathbf{N})$ down to $\mathrm{Er}(\mathbf{k};\mathbf{N})^t$. For instance, let us consider a $(2,\ldots,2)$-out-of-$(2,\ldots,2)$ special-sound interactive proof with $2\log_2(n)+1$ rounds, i.e., the verifier sends $\log_2 n$ challenges sampled from a set of cardinality two. This interactive proof has knowledge error

$$1 - \left(1 - \frac{1}{2}\right)^{\log_2 n} = 1 - \frac{1}{n} .$$

Now let t be the number of parallel repetitions required to reduce the knowledge error down to $2^{-\lambda}$. Then,

$$t \geq \frac{-\lambda}{\log_2(1 - \frac{1}{n})} \geq \lambda \cdot n .$$

A similar analysis applies to the $(2,3,\ldots,3)$-out-of-$(|\mathcal{C}|,\ldots,|\mathcal{C}|)$ special-sound compressed Σ-protocol Π_{comp} of Theorem 3.9. More precisely, if the size of the challenge set $\mathcal{C}$ is constant in $n+\lambda$, then the required number of parallel repetitions is *linear* in n. Therefore, after parallel repetition, the communication complexity becomes *superlinear* in n, which completely defeats the purpose of compressing the linear communication complexity of the basic Σ-protocol.

Hence, in some scenarios, parallel repetition does not allow for a sufficient knowledge error reduction while maintaining a sublinear communication complexity. For this reason, we introduce an alternative approach. Instead of repeating the interactive protocol, we aim to increase the size of the challenge set $\mathcal{C}$ in order to decrease the knowledge error. Let us now describe this approach.

Recall that our goal is to construct an interactive proof for proving knowledge of a short preimage of the $\mathcal{R}$-module homomorphism $\Psi\colon \mathbb{G}^n \to \mathbb{H}$. To increase the size of the challenge set $\mathcal{C}$, we extend the scalar ring $\mathcal{R}$ of the modules $\mathbb{G}^n$ and $\mathbb{H}$ to an extension $\mathcal{S}$ of $\mathcal{R}$. More precisely, we consider the tensor products $\mathcal{S} \otimes_{\mathcal{R}} \mathbb{G}^n$

and $\mathcal{S} \otimes_{\mathcal{R}} \mathbb{H}$, also referred to as *base extensions* over $\mathcal{S}$. These base extensions are $\mathcal{S}$-modules and the mapping

$$\Psi_S \colon \mathcal{S} \otimes_{\mathcal{R}} \mathbb{G}^n \to \mathcal{S} \otimes_{\mathcal{R}} \mathbb{H}\,, \quad \text{such that } s \otimes \mathbf{x} \mapsto s \otimes \Psi(\mathbf{x})\,,$$

is a well-defined $\mathcal{S}$-module homomorphism [AM69, p.27].

Let us assume that $s_1, \ldots, s_d \in \mathcal{S}$ is an $\mathcal{R}$-basis of $\mathcal{S}$. Then every element of $\mathcal{S} \otimes_{\mathcal{R}} \mathbb{G}^n$ has a unique representation of the form $s_1 \otimes \mathbf{x}_1 + \cdots + s_d \otimes \mathbf{x}_d$, with $\mathbf{x}_1, \ldots, \mathbf{x}_d \in \mathbb{G}^n$. Moreover, $\mathbf{x}$ is a Ψ-preimage of $P \in \mathbb{G}^n$ if and only if $s_1 \otimes \mathbf{x}$ is a $\Psi_{\mathcal{S}}$-preimage of $s_1 \otimes P$. Finally, if $s_1 \otimes \mathbf{x}_1 + \cdots + s_d \otimes \mathbf{x}_d$ is a $\Psi_{\mathcal{S}}$-preimage of $s_1 \otimes P$, it follows that $\mathbf{x}_1$ is a Ψ-preimage of P. Hence, proving knowledge of a (short) Ψ-preimage can be reduced to proving knowledge of a (short) $\Psi_{\mathcal{S}}$-preimage.

Note that an element $\mathbf{x} \in \mathcal{S} \otimes_{\mathcal{R}} \mathbb{G}^n$ is not an n-dimensional vector. Instead it is of the form

$$\mathbf{x} = \sum_{i=1}^{d} s_i \otimes \mathbf{x}_i = \sum_{i=1}^{d} s_i \otimes (\mathbf{x}_{i,L}, \mathbf{x}_{i,R}) \in \mathcal{S} \otimes_{\mathcal{R}} \mathbb{G}^n\,.$$

However, also $\mathbf{x}$ has naturally defined left and right parts, i.e.,

$$\mathbf{x}_L = \sum_{i=1}^{d} s_i \otimes \mathbf{x}_i = \sum_{i=1}^{d} s_i \otimes \mathbf{x}_{i,L} \in \mathcal{S} \otimes_{\mathcal{R}} \mathbb{G}^{n/2} \text{ and}$$
$$\mathbf{x}_R = \sum_{i=1}^{d} s_i \otimes \mathbf{x}_i = \sum_{i=1}^{d} s_i \otimes \mathbf{x}_{i,R} \in \mathcal{S} \otimes_{\mathcal{R}} \mathbb{G}^{n/2}\,.$$

For this reason, the compressed Σ-protocols are easily seen to also apply to the base extended homomorphism $\Psi_{\mathcal{S}}$.

Instantiating compressed Σ-protocol Π_{comp} for $\Psi_{\mathcal{S}}$ allows the challenge sets to be chosen as subsets of $\mathcal{S}$ instead of $\mathcal{R}$. Appropriately chosen ring extensions therefore allow for larger challenge sets. For instance, the ring $\mathbb{Z}$ only contains exceptional subsets (Definition 3.3) of cardinality two, while the ring extension $\mathbb{Z}[\omega_p]$, for a prime p and a primitive p-th root of unity ω_p, contains the exceptional subset

$$\left\{ \frac{\omega_p^k - 1}{\omega_p - 1} : 1 \leq k \leq p \right\}$$

of cardinality p.

Let us now return to our simplified example of a $(2, \ldots, 2)$-out-of-$(2, \ldots, 2)$ special-sound interactive proof. Although we focus on this simple example, the analysis below has a straightforward generalization to arbitrary $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs.

Suppose that by choosing an appropriate degree d ring extension, the challenge sets can be enlarged to challenge sets of cardinality d, i.e., the base extended interactive proof is $(2, \ldots, 2)$-out-of-$(d, \ldots, d)$ special-sound and has knowledge error

$$1 - \left(1 - \frac{1}{d}\right)^{\log_2 n}\,.$$

Moreover, the base extension increases the communication costs by a factor d. Before we continue our analysis, we derive the following lemma.

Lemma 3.3. *Let $N \in \mathbb{N}$ and $0 \leq x \leq 1/(4N)$, then*

$$1 - (1-x)^N \geq \frac{2Nx}{3}\,.$$

Proof.

$$\begin{aligned} 1 - (1-x)^N &= Nx - \sum_{i=2}^{N} \binom{N}{k} (-x)^i \geq Nx - \sum_{i=2}^{\infty} (Nx)^i \\ &= Nx - \frac{(Nx)^2}{1-Nx} = Nx\frac{1-2Nx}{1-Nx} \geq \frac{2Nx}{3}\,, \end{aligned}$$

where the final inequality follows because $Nx \leq 1/4$. □

From Lemma 3.3 it follows that the knowledge error of a $(2,\ldots,2)$-out-of-$(d,\ldots,d)$ special-sound interactive proof with $2\log_2(n)+1$ rounds and $d \geq 4\log_2 n$ satisfies

$$1 - \left(1 - \frac{1}{d}\right)^{\log_2 n} \geq \frac{2\log_2(n)}{3d}\,.$$

Hence, to reduce the knowledge error down to $2^{-\lambda}$, the degree d of the ring extension must be such that

$$d \geq \frac{2}{3} \cdot 2^{\lambda} \cdot \log_2 n\,.$$

In other words, the degree scales *logarithmically* in the input dimension n, but *exponentially* in the security parameter λ. Hence, besides parallel repetition, also base extension results in undesirable (communication) costs. More precisely, using parallel repetition, the communication costs scale linearly in the dimension n of the witness $\mathbf{x}$. And, using base extension, the communication costs scale exponentially in the security parameter λ.

However, it turns out that, by combining the two techniques, the knowledge error can be sufficiently reduced with only a limited increase of communication costs. More precisely, taking $t = \lambda$ parallel repetitions of the $(2,\ldots,2)$-out-of-$(d,\ldots,d)$ interactive proof with degree $d = 2\log_2(n)$, results in knowledge error

$$\left(1 - \left(1 - \frac{1}{d}\right)^{\log_2 n}\right)^t \leq \left(\frac{\log_2 n}{d}\right)^t = 2^{-\lambda}\,.$$

Moreover, the prover has to send $\mathcal{O}(\lambda \cdot \log_2^2 n)$ elements to the verifier, i.e., the communication complexity of the t-fold parallel repetition of the degree d base extended interactive proof is polylogarithmic in n.

Altogether, one should choose the ring extension $\mathcal{S}$ and the challenge set $\mathcal{C} \subseteq \mathcal{S}$ as a function of n, such that the knowledge error of the base extended interactive proof is *constant* in n and the degree of the ring extension is at most *polylogarithmic* in n. Then, $\mathcal{O}(\lambda)$ parallel repetitions are required to decrease the knowledge

error down to $2^{-\lambda}$ and the communication complexity only increases with a factor $\mathcal{O}\big(\lambda \cdot \mathrm{polylog}(n)\big)$ with respect to the basic interactive proof.

In theory the size of the challenge set can also grow exponentially in the degree d of the ring extension, e.g., if $\mathcal{R}$ and $\mathcal{S}$ are fields and the soundness slack is irrelevant. This would change the above trade-off significantly. In fact, in this case the knowledge error can be made negligible by using merely base extension, and no parallel repetitions are required. However, when taking the soundness slack and approximation factor into account, "good" challenge sets typically grow linearly in the degree of the ring extension. For this reason, our analysis has been restricted to this specific situation. Finding good challenge sets, resulting in small soundness slack and an appropriate approximation factor, is a difficult task on its own. In Chapter 5, we will give some concrete examples and for more details we refer to [LS18; ACX21].

Remark 3.6. The degree d base extended interactive proof allows a prover to prove knowledge of d different Ψ-preimages simultaneously without increasing the costs. More precisely, if $\mathcal{S}$ has basis $s_1, \ldots, s_d \in \mathcal{S}$ over $\mathcal{R}$, then proving knowledge of the Ψ-preimages of $P_1, \ldots, P_d \in \mathbb{H}$ is equivalent to proving knowledge of the $\Psi_{\mathcal{S}}$-preimage of

$$s_1 \otimes P_1 + s_2 \otimes P_2 + \cdots + s_d \otimes P_d \in \mathcal{S} \otimes_{\mathcal{R}} \mathbb{H}\,.$$

Remark 3.7. The compressed Σ-protocols of Section 3.2 allow a prover to prove knowledge of a preimage for a homomorphism between groups of prime exponent $q \geq 3$. Because the compression mechanism is 3-out-of-q special-sound, these interactive proofs require $q \geq 3$. By using the base extension techniques, the compressed Σ-protocols of Section 3.2 can be adapted to work for groups with arbitrary (not necessarily prime) exponent $m \geq 2$.

3.4 Compact Commitments and Linear Forms

Perhaps the most prominent application of our compressed Σ-protocols is proving knowledge of a commitment opening satisfying an arbitrary *linear* constraint. More precisely, compressed Σ-protocol are oftentimes instantiated with a homomorphism of the form

$$\Psi = (\textsc{com}, L)\colon \mathbb{G}^n \times \mathsf{Rand} \to \mathbb{H} \times \mathbb{G}, \quad (\mathbf{x}; \gamma) \mapsto \big(\textsc{com}(\mathbf{x}; \gamma), L(\mathbf{x})\big)\,,$$

where $\textsc{com}\colon \mathbb{G}^n \times \mathsf{Rand} \to \mathbb{H}$ is a vector commitment scheme. Hence, both the commitment scheme COM and the function L are homomorphisms. Moreover, the set Rand, from which the commitment randomness is sampled, is assumed to be an abelian group.

The resulting compressed Σ-protocol thus allows a prover to prove knowledge of an opening $(\mathbf{x}; \gamma)$ to some commitment P satisfying the linear constraint $L(\mathbf{x}) = y$ for some public value $y \in \mathbb{G}$. If $\mathbb{G} = \mathbb{Z}_q$, the homomorphism $L\colon \mathbb{Z}_q^n \to \mathbb{Z}_q$ is a linear form. For this reason, we also refer to the above functionality as *opening a linear form*. Moreover, we will also refer to homomorphisms $L\colon \mathbb{G}^n \to \mathbb{G}$ as linear forms. All the result in this section hold verbatim when we replace linear forms by *affine forms*, where we recall that an affine form is a linear form plus a constant.

Compressed Σ-protocols require the prover to send a logarithmic number of elements in the codomain of Ψ to the verifier. Therefore, to achieve a logarithmic communication complexity, we additionally require the commitment scheme to be *compact*, i.e., the size of a commitment $P = \text{COM}(\mathbf{x}; \gamma)$ should be independent of, or constant in, the dimension n of $\mathbf{x} \in \mathbb{G}^n$. In strong-RSA and lattice based platforms, due to their soundness slack, the communication complexity is polylogarithmic instead of logarithmic.

In this section, we will take a closer look at these compressed Σ-protocol instantiations. For simplicity, we ignore the norm bounds and restrict ourselves to the compressed Σ-protocols of Section 3.2 and assume $(\mathbb{G}, +)$ and $(\mathbb{H}, \cdot)$ to be $\mathbb{Z}_q$-modules. However, by using the techniques from Section 3.3, the constructions of this section are easily generalized towards *short* preimages.

In Section 3.4.1, we reduce the communication costs of the naive compressed Σ-protocol instantiation with roughly a factor two. In Section 3.4.2, we show how to amortize the costs of opening many linear forms $L_1, \ldots, L_s \colon \mathbb{G}^n \to \mathbb{G}$. These reduction and amortization approaches are only *computationally* special-sound. In Section 3.4.3, we show how to achieve the same functionality with unconditional special-soundness, without increasing the communication costs. Finally, in Section 3.4.4, we construct an interactive proof for proving knowledge of openings to many different commitments satisfying different linear constraints.

3.4.1 Opening Linear Forms on Committed Vectors

The compressed Σ-protocol for opening a linear form $L \colon \mathbb{G}^n \to \mathbb{G}$ on a compactly committed vector $\mathbf{x} \in \mathbb{G}^n$ is an interactive proof for relation

$$\mathfrak{R}_{\text{COM}} = \{(P, y; \mathbf{x}, \gamma) : \text{COM}(\mathbf{x}; \gamma) = P \wedge L(\mathbf{x}) = y\}. \tag{3.4}$$

This protocol is a straightforward instantiation of compressed Σ-protocol Σ_{comp} of Section 3.2.3. However, since the homomorphism (COM, L) has domain $\mathbb{G}^n \times \mathsf{Rand}$, it is not of the form $\Psi_n \colon \mathbb{G}^n \to \mathbb{H}$ required by Σ_{comp}. For this reason, one minor adaptation is required. Namely, the prover $\mathcal{P}$ simply sends the masked commitment randomness to the verifier after receiving the first challenge in the Σ-protocol. More precisely, the first steps of the compressed Σ-protocol for relation $\mathfrak{R}_{\text{COM}}$ proceed as follows:

- The prover samples $\mathbf{r} \leftarrow_R \mathbb{G}^n$ and $\rho \leftarrow_R \mathsf{Rand}$ uniformly at random, and sends $A = \text{COM}(\mathbf{r}; \rho)$ and $t = L(\mathbf{r})$ to the verifier;
- After receiving the challenge $c \in \mathbb{Z}_q$, the prover sends $\phi = \rho + c\gamma$.

Now observe that $\mathbf{z} = \mathbf{r} + c\mathbf{x}$ is a preimage of $(A \cdot P^c, t + cy)$ with respect to the homomorphism

$$\Psi(\cdot, \phi) \colon \mathbb{G}^n \to \mathbb{H} \times \mathbb{G}, \quad \mathbf{x} \mapsto \big(\text{COM}(\mathbf{x}; \phi), L(\mathbf{x})\big).$$

This homomorphism is of the required form and thus the compression mechanism applies as before. Assuming that $n = 2^\mu$ is a power-of-two, the resulting compressed Σ-protocol has communication costs:

- $\mathcal{P} \to \mathcal{V}$: $2\mu + 1$ elements of $\mathbb{G}$, $2\mu - 1$ elements of $\mathbb{H}$ and 1 element of Rand;

- $\mathcal{V} \to \mathcal{P}$: μ elements of $\mathbb{Z}_q$.

Note that, since Ψ has codomain $\mathbb{G} \times \mathbb{H}$, the prover must also send logarithmically many $\mathbb{G}$-elements. By contrast, in protocol Σ_{comp} for proving knowledge of preimages of $\Psi_n \colon \mathbb{G}^n \to \mathbb{H}$, the prover sends a constant number of $\mathbb{G}$-elements and logarithmically many $\mathbb{H}$-elements.

Remark 3.8. Typically the commitment randomness is sampled from $\mathsf{Rand} = \mathbb{G}^s$ for some $s \in \mathbb{N}$. In this case, the homomorphism $(\textsc{com}, L) \colon \mathbb{G}^{n+s} \to \mathbb{H} \times \mathbb{G}$ is already of the form required by compressed Σ-protocol Σ_{comp}, and the above adaptation can be omitted.

The aforementioned approach describes the naive compressed Σ-protocol instantiation for opening linear forms on compactly committed vectors. Let us now describe a more efficient technique for achieving exactly the same functionality. This technique was introduced by Bünz et al. [BBB+18].

Before we describe this improvement, recall that a vector commitment scheme allows a prover to commit to input vectors of arbitrary dimension. More precisely, by convention,

$$\textsc{com}(\mathbf{x}; \gamma) = \textsc{com}(\mathbf{x}, 0, \ldots, 0; \gamma)$$

for any number of zeros. If the number of zeros is clear from context, we simply write $\textsc{com}(\mathbf{x}, 0; \gamma)$, where now 0 represents a 0-vector with the appropriate dimension. Hence, if $\textsc{com}$ is a homomorphic vector commitment scheme, a committed vector $\mathbf{x} \in \mathbb{G}^n$ can always be appended with a vector $\mathbf{y} \in \mathbb{G}^m$:

$$\textsc{com}(\mathbf{x}; \gamma) \cdot \textsc{com}(0, \mathbf{y}; 0) = \textsc{com}(\mathbf{x}, 0; \gamma) \cdot \textsc{com}(0, \mathbf{y}; 0) = \textsc{com}(\mathbf{x}, \mathbf{y}; \gamma)\,.$$

The improved compressed Σ-protocol can now be described as follows. Instead of asking the prover to prove knowledge of a preimage of (P, y) with respect to $\Psi = (\textsc{com}, L)$, the verifier asks to prove knowledge of a preimage of $P \cdot \textsc{com}(0, cy; 0)$ with respect to the homomorphism

$$\Psi' \colon \mathbb{G}^n \times \mathsf{Rand} \to \mathbb{H}, \quad (\mathbf{x}; \gamma) \mapsto \textsc{com}(\mathbf{x}, c \cdot L(\mathbf{x}); \gamma)\,.$$

Note that, if $(\mathbf{x}, \gamma)$ is a preimage of Ψ, then it is also a preimage of Ψ', i.e., an honest prover can complete both tasks. This technique reduces relation $\mathfrak{R}_{\textsc{com}}$ to the relation

$$\{(P; \mathbf{x}, \gamma) : \textsc{com}(\mathbf{x}, c \cdot L(\mathbf{x}); \gamma) = P\}\,,$$

where the linear form is incorporated into the commitment. Since the codomain of Ψ' is $\mathbb{H}$ instead of $\mathbb{H} \times \mathbb{G}$, this technique reduces[3] the communication costs by roughly a factor two.

The reduction is an interactive proof for relation $\mathfrak{R}_{\textsc{com}}$, denoted by Π_{r} and described in Protocol 9. Its main properties are summarized in Theorem 3.10. Note that Π_{r} is clearly not zero-knowledge. However, since the prover only sends one message, the composition of Π_{r} with an appropriate instantiation of compressed Σ-protocol Σ_{comp} is easily seen to be special honest-verifier zero-knowledge. Moreover,

[3] Technically, the improvement depends on the bit-size of elements in $\mathbb{G}$ and $\mathbb{H}$. Here we assume $\mathbb{G}$- and $\mathbb{H}$-elements to be of the same size.

the special-soundness property only holds if the commitment scheme is binding, i.e., the cost of this reduction is a degradation from unconditional to computational special-soundness. In most practical applications, the commitment scheme is required to be binding anyway. For this reason, this degradation in security is almost always acceptable.

Protocol 9 Interactive Proof Π_{r} for Incorporating the Linear Form Into the Commitment.

PARAMETERS:	$n \in \mathbb{N}$, prime q, groups $(\mathbb{G}, +)$ and $(\mathbb{H}, \cdot)$ with exponent q, $L \in \mathrm{Hom}(\mathbb{G}^n, \mathbb{G})$ and $\mathrm{COM}\colon \mathbb{G}^n \times \mathsf{Rand} \to \mathbb{H}$ (homomorphic)
PUBLIC INPUT:	$P \in \mathbb{H}$, $y \in \mathbb{G}$
PROVER'S PRIVATE INPUT:	$\mathbf{x} \in \mathbb{G}^n$, $\gamma \in \mathsf{Rand}$
PROVER'S CLAIM:	$\mathrm{COM}(\mathbf{x}; \gamma) = P \wedge L(\mathbf{x}) = y$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
		$c \leftarrow_R \mathbb{Z}_q$
	$\xleftarrow{c}$	
	$\xrightarrow{\mathbf{x},\gamma}$	$\mathrm{COM}\big(\mathbf{x}, c \cdot L(\mathbf{x}); \gamma\big) \overset{?}{=} P \cdot \mathrm{COM}(0, c \cdot y; 0)$

Theorem 3.10 (Incorporating the Linear Form Into the Commitment)**.** *The interactive proof Π_{r} for relation $\mathfrak{R}_{\mathrm{COM}}$, described in Protocol 9, is perfectly complete and computationally 2-out-of-q special-sound, under the assumption that the commitment scheme is binding. Moreover, the communication costs are:*

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

Proof. Note that Π_{r} only has two communication rounds. By appending this protocol with an empty first message, from the prover to the verifier, it becomes a Σ-protocol. For this reason, we will also refer to Π_{r} as a Σ-protocol. Let us now show that Π_{r} has the desired completeness and special-soundness properties.

Completeness: This property follows immediately.

Special-Soundness: Let $(c, \mathbf{x}, \gamma)$ and $(c', \mathbf{x}', \gamma')$ be two accepting transcripts with distinct challenges $c \neq c' \in \mathbb{Z}_q$.

Then

$$\begin{aligned}
&\mathrm{COM}\big(\mathbf{x}, cL(\mathbf{x}); \gamma\big) \cdot \mathrm{COM}\big(\mathbf{x}', c'L(\mathbf{x}'); \gamma'\big)^{-1} \\
&\quad = \mathrm{COM}\big(\mathbf{x} - \mathbf{x}', cL(\mathbf{x}) - c'L(\mathbf{x}'); \gamma - \gamma'\big) \\
&\quad = \mathrm{COM}\big(0, (c - c')y; 0\big)\,.
\end{aligned}$$

Hence, either we have found two distinct openings

$$\big(\mathbf{x}-\mathbf{x}', cL(\mathbf{x})-c'L(\mathbf{x}');\gamma-\gamma'\big) \quad \text{and} \quad \big(0,(c-c')y;0\big)$$

for the same commitment, breaking its binding property, or $\mathbf{x}=\mathbf{x}'$, $\gamma=\gamma'$ and $cL(\mathbf{x})-c'L(\mathbf{x}')=(c-c')y$. In the latter case it follows that $L(\mathbf{x})=y$ and

$$\textsc{com}(\mathbf{x};\gamma)=\textsc{com}(\mathbf{x},0;\gamma)=\textsc{com}\big(\mathbf{x},cL(\mathbf{x});\gamma\big)\cdot\textsc{com}(0,cy;0)^{-1}=P\,.$$

Hence, $(\mathbf{x};\gamma)$ is a witness for statement (P,y) with respect to relation $\mathfrak{R}_{\textsc{com}}$, which completes the proof of the theorem.

□

Let us finally describe the improved interactive proof for opening linear forms on compactly committed vectors. This interactive proof is simply the composition $\Sigma_{\mathsf{comp}}\diamond\Pi_{\mathsf{r}}$ of the reduction Π_{r} with an appropriate instantiation of compressed Σ-protocol Σ_{comp}. The properties of this protocol are described in Theorem 3.11. Note in particular that, instead of $2\mu+1$ elements, the prover only sends 2 elements of $\mathbb{G}$ to the verifier. Hence, in comparison to the naive approach, the total number of elements sent by the prover has been reduced from $4\mu+1$ down to $2\mu+2$.

Theorem 3.11 (Compressed Σ-Protocol for Opening a Linear Form)**.** *Let $n=2^\mu$ for some $\mu\in\mathbb{N}$. Then the compressed Σ-protocol $\Sigma_{\mathsf{comp}}\diamond\Pi_{\mathsf{r}}$ for relation $\mathfrak{R}_{\textsc{com}}$ is perfectly complete, computationally $(2,2,3,\ldots,3)$-out-of-$(q,\ldots,q)$ special-sound, under the assumption that the commitment scheme is binding, and special honest-verifier zero-knowledge (SHVZK). Moreover, it has $(2\mu+2)$ communication rounds and the communication costs are:*

- $\mathcal{P}\to\mathcal{V}$*: 2 elements of $\mathbb{G}$, $2\mu-1$ elements of $\mathbb{H}$ and 1 element of* Rand*;*
- $\mathcal{V}\to\mathcal{P}$*: $\mu+1$ elements of $\mathbb{Z}_q$.*

3.4.2 Amortization - Opening Many Linear Forms

The previous section demonstrated how to efficiently open a linear form on a compactly committed vector. Moreover, by the amortization technique of Section 3.4.2, we know how to extend this functionality to opening *one* linear form on *many* different commitments, without increasing the communication costs. In this section, we consider the task of opening *many* different linear forms on *one* commitment. More precisely, our goal is to construct a communication-efficient interactive proof for relation

$$\mathfrak{R}^s_{\textsc{com}}=\{(P,y_1,\ldots,y_s;\mathbf{x},\gamma):\textsc{com}(\mathbf{x};\gamma)=P\wedge L_i(\mathbf{x})=y_i\ \forall 1\le i\le s\}\,.$$

There are several ways to realize this functionality. For instance, one could generalize the reduction of Section 3.4.1 and consider a commitment

$$\textsc{com}(\mathbf{x},c\cdot L_1(\mathbf{x}),\ldots,c\cdot L_s(\mathbf{x});\gamma)\,,$$

where $c \in \mathbb{Z}_q$ is a challenge sampled uniformly at random by the verifier. Hence, the linear forms are incorporated in different slots of the committed vector. Composing this reduction with an appropriate instantiation of compressed Σ-protocol Σ_{comp} would already result in an interactive proof for relation $\mathfrak{R}^s_{\mathrm{COM}}$ with communication complexity logarithmic in $n + s$.

However, we apply a different reduction and incorporate all the linear forms in a single slot of the commitment. Our reduction uses a "polynomial amortization trick" (known, e.g., from MPC). After composing this reduction with a compressed Σ-protocol, one obtains an interactive proof for relation $\mathfrak{R}^s_{\mathrm{COM}}$ with communication costs independent of s. Hence, the communication costs for opening many linear forms are exactly the same as for opening a single linear form. As before, the cost of this reduction is a degradation from unconditional to computational special-soundness. Moreover, the reduction is $(s+1)$-out-of-q special-sound.

For completeness, the reduction, denoted by Π_{R}, is described in Protocol 10 and its properties are summarized in Theorem 3.12.

Protocol 10 Interactive Proof Π_{R} for Incorporating *Many* Linear Forms Into the Commitment.

PARAMETERS:	$n, s \in \mathbb{N}$, prime q, groups $(\mathbb{G}, +)$ and $(\mathbb{H}, \cdot)$ with exponent q, $L_1, \ldots, L_s \in \mathrm{Hom}(\mathbb{G}^n, \mathbb{G})$ and $\mathrm{COM}\colon \mathbb{G}^n \times \mathsf{Rand} \to \mathbb{H}$ (homomorphic)
PUBLIC INPUT:	$P \in \mathbb{H}$, $y_1, \ldots, y_s \in \mathbb{G}$
PROVER'S PRIVATE INPUT:	$\mathbf{x} \in \mathbb{G}^n$, $\gamma \in \mathsf{Rand}$
PROVER'S CLAIM:	$\mathrm{COM}(\mathbf{x}; \gamma) = P \wedge L_i(\mathbf{x}) = y_i \ \forall 1 \leq i \leq s$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
		$c \leftarrow_R \mathbb{Z}_q$
	$\xleftarrow{c}$	
	$\xrightarrow{\mathbf{x};\gamma}$	$\mathrm{COM}\Big(\mathbf{x}, \sum_{i=1}^{s} c^i \cdot L_i(\mathbf{x}); \gamma\Big)$
		$\stackrel{?}{=}$
		$P \cdot \mathrm{COM}\Big(0, \sum_{i=1}^{s} c^i \cdot y_i; 0\Big)$

Theorem 3.12 (Incorporating Many Linear Forms Into the Commitment)**.** *The interactive proof Π_{R} for relation $\mathfrak{R}^s_{\mathrm{COM}}$, described in Protocol 10, is perfectly complete and computationally $(s+1)$-out-of-q special-sound, under the assumption that the commitment scheme is binding. Moreover, the communication costs are:*

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

Proof. **Completeness:** This property follows immediately.

Special-Soundness: Let $(c_0, \mathbf{x}_0, \gamma_0), \ldots, (c_s, \mathbf{x}_s, \gamma_s)$ be $s+1$ accepting transcripts with pairwise distinct challenges $c_0, \ldots, c_s \in \mathbb{Z}_q$.

For $0 \leq k \leq s$, let us write $f_k(\cdot) = \sum_{i=1}^{s} c_k^i L_i(\cdot)$. Then, for all $k \neq \ell$,

$$\begin{aligned} &\textsc{com}(\mathbf{x}_k, f_k(\mathbf{x}_k); \gamma_k) \cdot \textsc{com}(\mathbf{x}_\ell, f_\ell(\mathbf{x}_\ell); \gamma_\ell)^{-1} \\ &\quad = \textsc{com}(\mathbf{x}_k - \mathbf{x}_\ell, f_k(\mathbf{x}_k) - f_\ell(\mathbf{x}_\ell); \gamma_k - \gamma_\ell) \\ &\quad = \textsc{com}\Big(0, \sum_{i=1}^{s} (c_k^i - c_\ell^i) y_i; 0\Big) . \end{aligned}$$

Hence, either we have found two distinct openings for the same commitment, breaking its binding property, or $\mathbf{x}_k = \mathbf{x}_\ell$, $gamma_k = \gamma_\ell$ and

$$f_k(\mathbf{x}_k) - f_\ell(\mathbf{x}_\ell) = \sum_{i=1}^{s} (c_k^i - c_\ell^i) y_i \,, \tag{3.5}$$

for all $0 \leq k, \ell \leq s$. In the latter case, let $\mathbf{x} = \mathbf{x}_0 = \cdots = \mathbf{x}_s$ and $\gamma = \gamma_0 = \cdots = \gamma_s$, then it is easily seen that $\textsc{com}(\mathbf{x}; \gamma) = P$. Moreover, let $Q(X) = \sum_{i=1}^{s} (L_i(\mathbf{x}) - y_i) X^i \in \mathbb{G}[X]$. Then, by Equation 3.5

$$Q(c_k) = f_k(\mathbf{x}) - \sum_{i=1}^{s} c_k^i \cdot y_i = f_\ell(\mathbf{x}) - \sum_{i=1}^{s} c_\ell^i \cdot y_i = Q(c_\ell) \,,$$

for all $0 \leq k, \ell \leq s$. Since the $s+1$ evaluation points c_k are pairwise distinct and Q is a polynomial of degree at most s with constant term 0, it follows that $Q(X) = Q(0) = 0$ is identically zero, i.e., $L_i(\mathbf{x}) = y_i$ for all $1 \leq i \leq s$.

Hence, $(\mathbf{x}; \gamma)$ is a witness for statement $(P, y_1, \ldots, y_s)$ with respect to relation $\mathfrak{R}^s_{\textsc{com}}$, which completes the proof of the theorem.

□

3.4.3 Opening Linear Forms with Unconditional Soundness

The interactive proofs of the previous two sections reduce the communication costs of opening linear forms on a compactly committed vector. However, these reductions are only *computationally* special-sound. In this section, we describe an alternative approach with roughly the same communication costs and *unconditional* special-soundness.

First observe that, since q is prime and thus $\mathbb{Z}_q$ is a field, the $\mathbb{Z}_q$-module $\mathbb{G}^n$ is a vector space admitting a $\mathbb{Z}_q$-basis $\mathbf{b}_1, \ldots, \mathbf{b}_m \in \mathbb{G}^n$. Note that the $\mathbb{Z}_q$-dimension m of $\mathbb{G}^n$ is not necessarily equal to n. For simplicity, let us assume that $\mathbb{G} = \mathbb{Z}_q$. Then $m = n$ and a basis $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{G}^n$ can be computed efficiently. Moreover, there exists an efficient algorithm to express elements of $\mathbb{G}^n = \mathbb{Z}_q^n$ as linear combinations of these basis vectors. Therefore, in this case, proving knowledge of a commitment opening $(\mathbf{x}; \gamma) \in \mathbb{G}^n \times \mathsf{Rand}$ is equivalent to proving knowledge of a preimage of the homomorphism

$$\Psi \colon \mathbb{Z}_q^n \times \mathsf{Rand} \to \mathbb{H}, \quad (\mathbf{y}; \gamma) \mapsto \textsc{com}(B \cdot \mathbf{y}; \gamma) \,,$$

where

$$B = \begin{pmatrix} \mathbf{b}_1 & \cdots & \mathbf{b}_n \end{pmatrix} \in \mathbb{G}^{n \times n}.$$

We now observe that proving that a committed vector $\mathbf{x} \in \mathbb{G}^n$ satisfies $L(\mathbf{x}) = y$, for some linear form L and scalar y, is equivalent to proving that $\mathbf{x}$ lies in the affine subspace $A_{L,y} = \{\mathbf{z} \in \mathbb{G}^n : L(\mathbf{z}) = y\}$. We assume (without loss of generality) that $y = 0$ and $L \neq 0$. Then $V_L := A_{L,0} \subset \mathbb{G}^n$ is a linear subspace of dimension $n-1$. Both prover and verifier use the same deterministic algorithm to compute a basis $\mathbf{v}_1, \ldots, \mathbf{v}_{n-1} \in \mathbb{G}^n$ for V_L and set

$$\Psi' \colon \mathbb{Z}_q^{n-1} \times \mathsf{Rand} \to \mathbb{H}, \quad (\mathbf{y}; \gamma) \mapsto \textsc{com}(B' \cdot \mathbf{y}; \gamma),$$

where

$$B' = \begin{pmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_{n-1} \end{pmatrix} \in \mathbb{G}^{n \times n-1}.$$

By black-box application of the compressed Σ-protocol for proving knowledge of Ψ'-preimages, the prover shows that it knows a Ψ'-preimage $(\mathbf{y}; \gamma)$ of P. Let $\mathbf{x} = B' \cdot \mathbf{y} \in \mathbb{G}^n$, then $(\mathbf{x}; \gamma)$ is an opening of commitment P. Moreover, $\mathbf{x}$ lies in the linear subspace V_L and therefore $L(\mathbf{x}) = y = 0$.

Hence, opening the linear form L on a committed vector is reduced to proving knowledge of a Ψ'-preimage. As before, since the homomorphism Ψ' has codomain $\mathbb{H}$ instead of $\mathbb{H} \times \mathbb{G}$, this approach reduces the communication costs by roughly a factor two. However, in contrast to the reduction of Section 3.4.1, this reduction is unconditionally special-sound. Moreover, this reduction reduces the dimension of the secret witness from n down to $n-1$. In general, opening s linearly independent linear forms on the same commitment, reduces the dimension of the witness from n down to $n-s$. For this reason, this unconditionally secure approach even results in (slightly) smaller communication costs.

Although this view may be superior from a conceptual standpoint, it does increase the computational costs for both the prover and the verifier. Both have to compute a basis for V_L, and the prover has to express the secret witness $\mathbf{x}$ as a $\mathbb{Z}_q$-linear combination of the basis vectors. If $\mathbb{G} = \mathbb{Z}_q$ this can be done efficiently. However, if the discrete logarithm problem is hard in $\mathbb{G}$, there does not exist an efficient algorithm for expressing arbitrary witnesses $\mathbf{x}$ as $\mathbb{Z}_q$-linear combination of basis vectors. For these reasons, our protocols will be based on the computationally special-sound reductions of Section 3.4.1 and Section 3.4.2.

3.4.4 Compactification

So far, we have shown how to handle two different amortization scenarios efficiently:

1. opening *one* linear form on *many* compact commitments (Section 3.2.4);
2. opening *many* linear forms on *one* compact commitment (Section 3.4.2).

For both cases, we presented a protocol with roughly the same communication costs as opening *one* linear form on *one* compact commitment. More precisely, in the first case the communication costs are exactly the same, and in the second case the verifier has to send one additional challenge to the prover. A straightforward combination of these techniques results in an interactive proof for opening *many*

linear forms on *many* compact commitments, without increasing the communication costs.

However, in many practical applications these amortization techniques do not suffice. For instance, in Section 7.2, we will see that to prove that a committed vector $\mathbf{x}$ satisfies a *nonlinear* constraint, the vector $\mathbf{x}$ needs to be appended with auxiliary information $\mathsf{aux} \in \mathbb{G}^t$ for some $t \in \mathbb{N}$. This auxiliary information *linearizes* the nonlinear constraint. More precisely, if the committed vector $(\mathbf{x}, \mathsf{aux})$ satisfies certain *linear* constraints, it follows that $\mathbf{x}$ satisfies the required nonlinear constraint. For more details we refer to Section 7.2. Now, from a practical application perspective, it is likely that the prover is *already* committed to $\mathbf{x}$ before the start of the interactive proof. The prover can be committed to $\mathbf{x}$ in a *single* compact commitment, but it can also be committed to the coefficients of $\mathbf{x}$ *individually*. The latter is relevant in practical situations with a natural dynamic, where provers deliver committed data in subsequent transactions, and only periodically prove some property on the compound information.

In order to deal with each of these scenarios, we need some further utility enhancements. It turns out that this is just a matter of "technology," i.e., plug and play with our compressed Σ-protocols and their basic theory suffices. We consider the following two extreme cases:

Case 1: Opening a linear form L_i on a compact commitment $P_i = \text{COM}(\mathbf{x}_i; \gamma_i)$ for $1 \leq i \leq s$. Because the prover does not wish to reveal the "cross-terms" $L_i(\mathbf{x}_j)$ for $i \neq j$, this is different from the standard amortization scenarios.

Case 2: Opening a linear form $L(\mathbf{x})$ evaluated on an input vector $\mathbf{x} = (x_1, \ldots, x_n)$ dispersed over n different commitments $P_i = \text{COM}(x_i; \gamma_i)$.

Besides these extreme cases one can consider hybrid scenarios in which the secret-vector-of-interest $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_s)$ is dispersed over s compact commitments to vectors $\mathbf{x}_i \in \mathbb{G}^{n_i}$. The methods described below *both* carry over to hybrid scenarios. The optimal approach depends on specific properties of the scenario. Namely, the communication complexity of the "Case 1 enhancement" is linear in the number of commitments, whereas the communication complexity of the "Case 2 enhancement" is quadratic in the (maximum) dimension of the committed vectors. Both enhancements reduce the situation to that of a prover with a single compact commitment to all relevant data (i.e., input data and auxiliary data). For this reason, these techniques are referred to as *compactification*.

Case 1. To further emphasize the practical relevance of this case, let us consider the commit-and-proof scenario, where a prover is already committed to the secret input vector $\mathbf{x}$ in a compact commitment $P = \text{COM}(\mathbf{x}, \gamma)$ and wishes to prove that $\mathbf{x}$ satisfies some *nonlinear* constraint. To handle this scenario, the prover sends a commitment $Q = \text{COM}(0, \mathsf{aux}; \rho)$ to the required auxiliary information aux to the verifier, and both the prover and verifier compute the new commitment $P' := P \cdot Q = \text{COM}(\mathbf{x}, \mathsf{aux}; \gamma + \rho)$ to the vector $\mathbf{x}$ appended with the auxiliary data aux. Subsequently, the prover opens the required linear forms on commitment P' for proving that $\mathbf{x}$ satisfies the given nonlinear constraint (for more details see Section 7.2). Additionally, the prover must show that input $\mathbf{x}$ and the auxiliary

information aux "live on different coefficients" of the appended vector $(\mathbf{x}, \mathsf{aux})$, i.e., it must show that the opening $(0, \mathsf{aux}; \rho)$ of commitment Q starts with the appropriate number of zeros. If this is not the case, a dishonest prover could simply use the auxiliary information to modify the coefficients of $\mathbf{x}$. Note that proving that the i-th coefficient of a committed vector equals zero boils down to opening the linear form $L(\mathbf{x}) = x_i$. Combined with the amortization technique for opening many linear forms on a single commitment, we are therefore exactly in the Case 2 scenario (with $s = 2$);

- opening a linear form L_i on $P_i = \text{COM}(\mathbf{x}_i; \gamma_i)$ for $1 \leq i \leq s$.

The straightforward approach for handling this case, simply invokes s different compressed Σ-protocols for the commitments. This would clearly incur a multiplicative factor s loss in the communication efficiency. We show how to avoid this loss.

For simplicity, we restrict ourselves to the case $s = 2$, but this compactification technique has a straightforward generalization to arbitrary s. More precisely, let us consider the two linear forms $L_1, L_2 : \mathbb{G}^n \to \mathbb{G}$ and two compact commitments $P_1 = \text{COM}(\mathbf{x}_1; \gamma_1)$ and $P_2 = \text{COM}(\mathbf{x}_2; \gamma_2)$ to $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{G}^n$. The goal is to efficiently open $L_1(\mathbf{x}_1)$ and $L_2(\mathbf{x}_2)$ in zero-knowledge. In particular, the *cross-terms* $L_1(\mathbf{x}_2)$ and $L_2(\mathbf{x}_1)$ are to remain secret.

The main idea is to build a *shell* around the compact commitments that allows the prover to mask linear form evaluations that are not supposed to be revealed, i.e., the cross-terms. Thereby, the problem can be reduced to a standard amortization scenario, where the entire "matrix" of linear form evaluations

$$\begin{pmatrix} L_1(\mathbf{x}_1) & L_1(\mathbf{x}_2) \\ L_2(\mathbf{x}_1) & L_2(\mathbf{x}_2) \end{pmatrix}$$

is revealed. More precisely, *intended* evaluations, on the diagonal of this matrix, will return the correct value and *unintended* evaluations will return a random, i.e., masked, value.

Let us now consider the details of our solution. The relation is somewhat relaxed by allowing the prover to append the committed vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ with two additional (random) coefficients $u, w \in \mathbb{G}$. However, it is essential that first coefficient u is only used to equip commitment P_1 with a shell, and the second coefficient w is only used to equip commitment P_2 with a shell. Shelled commitments $\text{COM}(\mathbf{x}_1, u, 0; \gamma_1'')$ to $\mathbf{x}_1$ and $\text{COM}(\mathbf{x}_2, 0, w; \gamma_2'')$ to $\mathbf{x}_2$ are obtained by multiplying P_1 and P_2 with shells $\text{COM}(0, u, 0; \gamma_1')$ and $\text{COM}(0, 0, w; \gamma_2')$, respectively.

We show how to prove knowledge of "shelled" openings $(\mathbf{x}_1, u, 0; \gamma_1)$ and $(\mathbf{x}_2, 0, w; \gamma_1)$ of the initial commitments P_1 and P_2, such that $L_1(\mathbf{x}_1) = y_1$ and $L_2(\mathbf{x}_2) = y_2$. More precisely, our compactification technique is an interactive proof for relation:

$$\mathfrak{R}_{\text{shell}} = \left\{ (P_1, P_2, y_1, y_2; \mathbf{x}_1, \mathbf{x}_2, u, w, \gamma_1, \gamma_2) : \begin{array}{l} P_1 = \text{COM}(\mathbf{x}_1, u, 0; \gamma_1) \wedge \\ P_2 = \text{COM}(\mathbf{x}_2, 0, w; \gamma_2) \wedge \\ L_1(\mathbf{x}_1) = y_1 \ \wedge \ L_2(\mathbf{x}_2) = y_2 \end{array} \right\}.$$

In particular, there is no constraint on the shells u and w. This is essential because the shells will be used to mask the cross-terms $L_1(\mathbf{x}_2)$ and $L_2(\mathbf{x}_1)$ that are to remain secret.

Next, we describe how this relation can be reduced to the standard amortization scenario where cross terms *are* revealed. To this end, let $\rho \in \mathbb{Z}_q^*$ be a challenge, sampled uniformly at random by the verifier, and let us consider the following linear forms:

$$L_1^\rho \colon \mathbb{G}^{n+2} \to \mathbb{G}, \quad (\mathbf{x}, a, b) \mapsto L_1(\mathbf{x}) + \rho \cdot b\,,$$
$$L_2^\rho \colon \mathbb{G}^{n+2} \to \mathbb{G}, \quad (\mathbf{x}, a, b) \mapsto L_2(\mathbf{x}) + \rho \cdot a\,.$$

Then

$$\begin{pmatrix} L_1^\rho(\mathbf{x}_1, u, 0) & L_1^\rho(\mathbf{x}_2, 0, w) \\ L_2^\rho(\mathbf{x}_1, u, 0) & L_2^\rho(\mathbf{x}_2, 0, w) \end{pmatrix} = \begin{pmatrix} y_1 & L_1(\mathbf{x}_2) + \rho \cdot w \\ L_2(\mathbf{x}_1) + \rho \cdot u & y_2 \end{pmatrix}, \tag{3.6}$$

i.e., the cross-terms $L_1(\mathbf{x}_2)$ and $L_2(\mathbf{x}_1)$ are masked by the elements $\rho \cdot w$ and $\rho \cdot u$, respectively. If the prover chooses the shells $u, w \in \mathbb{G}$ uniformly at random, then the masks $\rho \cdot w$ and $\rho \cdot u$ are uniformly distributed, and the distribution of the evaluations $L_1^\rho(\mathbf{x}_2, 0, w)$ and $L_2^\rho(\mathbf{x}_1, u, 0)$ is independent of the secret vectors $\mathbf{x}_1$ and $\mathbf{x}_2$.

Hence, if a prover appends the commitments to the secret vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ with uniformly random shells $u, w \in \mathbb{G}$, the case 1 scenario can be reduced to a standard amortization scenario where the prover opens all four linear form evaluations. To this end, the prover sends commitments $R_1 = \textsc{com}(0, u, 0; \rho_1)$ and $R_2 = \textsc{com}(0, 0, w; \rho_2)$, to uniformly random shells $u, w \in \mathbb{G}$, to the verifier. Moreover, by means of a standard Σ-protocol, the prover shows that R_1 and R_2 are 1-dimensional commitments to u and v. Note that the communication costs of this standard Σ-protocol do not depend on n. Subsequently, after receiving a challenge $\rho \leftarrow_R \mathbb{Z}_q^*$, the prover opens the linear forms L_1^ρ and L_2^ρ, as defined above, on the shelled commitments $Q_1 = P_1 \cdot R_1$ and $Q_2 = P_2 \cdot R_2$, i.e., by invoking the appropriate compressed Σ-protocol it proves that Equation 3.6 holds.

The compactification protocol Π_{shell} for relation $\mathfrak{R}_{\text{shell}}$ is described in Protocol 11. Its main properties are summarized in Theorem 3.13. Interactive proof Π_{shell} has essentially the same communications costs as compressed Σ-protocol Σ_{comp} for opening *one* linear form on *one* compact commitment. Hence, we have indeed avoided the multiplicative factor two loss of the naive approach.

Note that, in contrast to all interactive proofs presented before, Π_{shell} requires the commitment scheme to be perfectly hiding. The reason is that, for Π_{shell} to be perfectly special honest-verifier zero-knowledge, the first message containing the commitments $R_1 = \textsc{com}(0, u, 0; \rho_1)$ and $R_2 = \textsc{com}(0, 0, w; \rho_2)$ should not reveal any information about the masks u and w. The protocol can also be instantiated with statistically or computationally hiding commitment schemes, this would affect the zero-knowledge property accordingly.

Theorem 3.13 (Compactification Protocol for Shelled Commitments)**.** *Let $n + 2 = 2^\mu$ for some $\mu \in \mathbb{N}$. Then the interactive proof Π_{shell} for relation $\mathfrak{R}_{shell}$, described in Protocol 11, is perfectly complete, computationally $(2, 2, 3, \ldots, 3)$-out-of-$(q, q-1, q, \ldots, q)$ special-sound, under the assumption that the commitment scheme is binding, and special honest-verifier zero-knowledge (SHVZK), under the assumption that the commitment scheme is perfectly hiding. Moreover, it has $(2\mu + 7)$ communication rounds and the communication costs are:*

- $\mathcal{P} \to \mathcal{V}$: 6 *elements of* $\mathbb{G}$, $2\mu + 3$ *elements of* $\mathbb{H}$ *and* 3 *elements of* Rand;
- $\mathcal{V} \to \mathcal{P}$: $\mu + 3$ *elements of* $\mathbb{Z}_q$.

Proof. First, observe that the amortized compressed Σ-protocol, invoked by interactive proof Π_{shell}, uses both the amortization technique from Section 3.2.4, over the two commitments, and the amortization technique from Section 3.4.2, over the two linear forms. Therefore, the compressed Σ-protocol is perfectly complete, computationally $(3, \ldots, 3)$-out-of-$(q, \ldots, q)$ special-sound, under the assumption that the commitment scheme is binding, and special honest-verifier zero-knowledge (SHVZK). Moreover, it has $(2\mu+2)$ communication rounds and the communication costs are:

- $\mathcal{P} \to \mathcal{V}$: 2 elements of $\mathbb{G}$, $2\mu - 1$ elements of $\mathbb{H}$ and 1 element of Rand;
- $\mathcal{V} \to \mathcal{P}$: $\mu + 1$ elements of $\mathbb{Z}_q$.

From this the communication costs of Π_{shell} follow. Let us now prove the remaining properties.

Completeness: This property follows immediately.

Special-Soundness: Suppose we are given an accepting $(1, 2, 3, \ldots, 3)$-tree of transcripts, i.e., all transcripts in this tree start with the same messages

$$(R_1, R_2, A_1, A_2, c, z_1, z_2, \phi_1, \phi_2)\,.$$

Further we have two distinct challenges $\rho \neq \rho' \in \mathbb{Z}_q^*$, corresponding to the two different $(1, 1, 3, \ldots, 3)$-trees of accepting transcripts.

By the $(3, \ldots, 3)$-out-of-$(q, \ldots, q)$ special-soundness of the compressed Σ-protocol that is invoked, for both ρ and ρ', openings of the commitments $P_1 \cdot R_1$ and $P_2 \cdot R_2$ can be computed given these trees (under the assumption that the commitment scheme is binding). Hence, either we have found distinct openings for the same commitments, breaking the binding property of COM, or the commitment openings found for ρ and ρ' coincide.

Let us assume the latter and write $(\bar{\mathbf{z}}_1, \bar{u}, \bar{w}', \bar{\gamma}_1)$ and $(\bar{\mathbf{z}}_2, \bar{u}', \bar{w}, \bar{\gamma}_2)$ for the openings of $P_1 \cdot R_1$ and $P_2 \cdot R_2$, respectively. Then, by the same special-soundness property,

$$L_1^{\rho}(\bar{\mathbf{z}}_1, \bar{u}, \bar{w}', \bar{\gamma}_1) = L_1^{\rho'}(\bar{\mathbf{z}}_1, \bar{u}, \bar{w}', \bar{\gamma}_1) = y_1\,,$$
$$L_2^{\rho}(\bar{\mathbf{z}}_2, \bar{u}', \bar{w}, \bar{\gamma}_2) = L_2^{\rho'}(\bar{\mathbf{z}}_2, \bar{u}', \bar{w}, \bar{\gamma}_2) = y_2\,.$$

Therefore, by definition of L_1^{ρ}, $L_1^{\rho'}$, L_2^{ρ} and $L_2^{\rho'}$, it is easily seen to follow that $\bar{w}' = \bar{u}' = 0$, $L_1(\bar{\mathbf{z}}_1) = y_1$ and $L_2(\bar{\mathbf{z}}_2) = y_2$.

Hence, the pair $(\bar{\mathbf{z}}_1, \bar{u}, 0, \bar{\gamma}_1)$ and $(\bar{\mathbf{z}}_2, 0, \bar{w}, \bar{\gamma}_2)$ is a witness for statement $(P_1 \cdot R_1, P_2 \cdot R_2, y_1, y_2)$ with respect to relation $\mathfrak{R}_{\text{shell}}$.

The desired special-soundness property of Π_{shell} now follows from the special-soundness of the Σ-protocol used to prove knowledge of appropriate openings

of R_1 and R_2. More precisely, this Σ-protocol shows that the prover knows an opening of R_1 with zeros everywhere except in the first shell coefficient, and an opening of R_2 with zeros everywhere except in the second shell coefficient. Combined with the previously extracted witness for $(P_1 \cdot R_1, P_2 \cdot R_2, y_1, y_2)$, this corresponds to a witness for statement (P_1, P_2, y_1, y_2).

SHVZK: Transcript for statements (P_1, P_2, y_1, y_2) that admit a witness are simulated as follows. Sample $\mu + 3$ challenges $(c, \rho, c_0, \ldots, c_\mu)$ and elements $z_1, z_2, y_{1,2}, y_{2,1} \leftarrow_R \mathbb{G}$ and $\gamma_1', \gamma_2', \phi_1, \phi_2 \leftarrow_R \mathsf{Rand}$ uniformly at random. Then compute $R_1 = \textsc{com}(0; \gamma_1')$, $R_2 = \textsc{com}(0; \gamma_2')$, $A_1 = \textsc{com}(0, z_1, 0; \phi_1) \cdot R_1^{-c}$ and $A_2 = \textsc{com}(0, 0, z_2; \phi_2) \cdot R_2^{-c}$.

Then, since $\rho \neq 0$ and (P_1, P_2, y_1, y_2) admits a witness, the public statement $(P_1 \cdot R_1, P_1 \cdot R_1, y_1, y_{1,2}, y_{2,1}, y_2)$ for the amortized compressed Σ-protocol admits a witness. Therefore, it is possible to run the SHVZK simulator for this protocol, given this statement and the $\mu + 1$ challenges sampled before, to obtain a protocol transcript tr. The SHVZK simulator for Π_{shell} then outputs transcript

$$(R_1, R_2, A_1, A_2, c, z_1, z_2, \phi_1, \phi_2, \rho, y_{1,2}, y_{2,1}, \mathsf{tr}) .$$

Because $\rho \neq 0$ and the commitment scheme is perfectly hiding, simulated transcripts have exactly the same distribution as honestly generated ones, which completes the proof of theorem.

□

Interactive proof Π_{shell} shows how to handle the case 1 compactification scenario if $s = 2$, i.e., opening linear form evaluations $L_1(\mathbf{x}_1)$ and $L_2(\mathbf{x}_2)$ given compact commitments to $\mathbf{x}_1$ and $\mathbf{x}_2$. This technique has a straightforward generalization to arbitrary s, where the matrix of linear form evaluations is an $s \times s$ matrix containing s public values on the diagonal and $s^2 - s$ secret values, the cross-terms. Hence, in general, commitments must be appended with $s^2 - s$ different shells. For this reason, the communication costs grow quadratically in s. However, this quadratic loss in communication efficiency is *additive*, i.e., the communication costs are in $\mathcal{O}(s^2 + \log n)$. By contrast, the communication costs of the naive approach are in $\mathcal{O}(s \log n)$. The optimal approach thus depends on specific properties of the application scenario.

Case 2. Let us now consider the case where the prover has n individual commitments $P_i = \textsc{com}(x_i; \gamma_i)$ to the coefficients of $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{G}^n$, and wishes to prove that $L(\mathbf{x}) = y$ for some public linear form $L \colon \mathbb{G}^n \to \mathbb{G}$ and $y \in \mathbb{G}$. Hence, in this case the relevant information is dispersed over many different commitments. Our goal is thus to construct an interactive proof for relation

$$\mathfrak{R}_{\mathsf{d}} = \{(P_1, \ldots, P_n, y; \mathbf{x}, \gamma_1, \ldots, \gamma_n) : \textsc{com}(x_i; \gamma_i) = P_i \wedge L(\mathbf{x}) = y\} .$$

To bring about the desired starting point of the compressed Σ-protocols, our approach is to *compactify* all the coefficients x_i into one single compact commitment P.

Protocol 11 Compactification Protocol Π_{shell} for Shelled Commitments.

PARAMETERS:	$n+2=2^\mu \in \mathbb{N}$, prime q, groups $(\mathbb{G},+)$ and $(\mathbb{H},\cdot)$ with exponent q, $L_1, L_2 \in \mathrm{Hom}(\mathbb{G}^n, \mathbb{G})$ and $\mathrm{COM}\colon \mathbb{G}^n \times \mathsf{Rand} \to \mathbb{H}$ (homomorphic)
PUBLIC INPUT:	$P_1, P_2 \in \mathbb{H}$, $y_1, y_2 \in \mathbb{G}$
PROVER'S PRIVATE INPUT:	$\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{G}^n$, $u, w \in \mathbb{G}^n$, $\gamma_1, \gamma_2 \in \mathsf{Rand}$
PROVER'S CLAIM:	$\mathrm{COM}(\mathbf{x}_1, u, 0; \gamma_1) = P_1 \wedge L_1(\mathbf{x}_1) = y_1 \wedge$ $\mathrm{COM}(\mathbf{x}_2, 0, w; \gamma_2) = P_2 \wedge L_2(\mathbf{x}_2) = y_2$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
$u', w', a_1, a_2 \leftarrow_R \mathbb{G}$		
$\gamma_1', \gamma_2', \psi_1, \psi_2 \leftarrow_R \mathsf{Rand}$		
$R_1 = \mathrm{COM}(0, u', 0; \gamma_1')$		
$R_2 = \mathrm{COM}(0, 0, w'; \gamma_2')$		
$A_1 = \mathrm{COM}(0, a_1, 0; \psi_1)$		
$A_2 = \mathrm{COM}(0, 0, a_2; \psi_2)$		
	$\xrightarrow{R_1, R_2, A_1, A_2}$	
	$\xleftarrow{c}$	$c \leftarrow_R \mathbb{Z}_q$
$z_1 = a_1 + cu'$		
$z_2 = a_2 + cw'$		
$\phi_1 = \psi_1 + c\gamma_1'$		
$\phi_2 = \psi_2 + c\gamma_2'$	$\xrightarrow{z_1, z_2, \phi_1, \phi_2}$	$\mathrm{COM}\big(0, z_1, 0; \phi_1\big) \stackrel{?}{=} A_1 \cdot R_1^c$
		$\mathrm{COM}\big(0, 0, z_2; \phi_2\big) \stackrel{?}{=} A_2 \cdot R_2^c$
	$\xleftarrow{\rho}$	$\rho \leftarrow_R \mathbb{Z}_q^*$
$y_{1,2} = L_1(\mathbf{x}_2) + \rho \cdot (w + w')$		
$y_{2,1} = L_2(\mathbf{x}_1) + \rho \cdot (u + u')$	$\xrightarrow{y_{1,2}, y_{2,1}}$	

Run an amortized compressed Σ-protocol for proving knowledge of openings $(\mathbf{x}_1, u + u', 0; \gamma_1 + \gamma_1')$ and $(\mathbf{x}_2, 0, w + w'; \gamma_2 + \gamma_2')$ of commitments $P_1 \cdot R_1$ and $P_2 \cdot R_2$, respectively, such that:

$$\begin{array}{rclcrcl} L_1^\rho(\mathbf{x}_1, u, w') & = & y_1\,, & & L_1^\rho(\mathbf{x}_2, u', w) & = & y_{1,2}\,, \\ L_2^\rho(\mathbf{x}_1, u, w') & = & y_{2,1}\,, & & L_2^\rho(\mathbf{x}_2, u', w) & = & y_2\,, \end{array}$$

where

$$L_1^\rho(\mathbf{x}, a, b) := L_1(\mathbf{x}) + \rho \cdot b \quad \text{and} \quad L_2^\rho(\mathbf{x}, a, b) := L_2(\mathbf{x}) + \rho \cdot a\,.$$

The first component of our interactive proof is a standard (amortized) Σ-protocol for proving knowledge of the openings $(x_i; \gamma_i)$ of the commitments P_i. Let us recall this Σ-protocol:

1. The prover samples $r \leftarrow_R \mathbb{G}$ and $\gamma \leftarrow \mathsf{Rand}$ uniformly at random and sends $A = \mathrm{COM}(r; \rho)$ to the verifier;

2. After receiving a challenge $c \leftarrow_R \mathbb{Z}_q$, sampled uniformly at random by the verifier, the prover sends $z = r + \sum_{i=1}^{n} c^i x_i$ and $\phi = \rho + \sum_{i=1}^{n} c^i \gamma_i$;
3. The verifier checks that $\text{COM}(z; \phi) = A \cdot \prod_{i=1}^{n} P_i^{c_i}$.

Note that communication costs of this Σ-protocol are independent of n.

We now observe that $z = r + \sum_{i=1}^{n} c^i x_i$ is a *linear form* L_c evaluated in the secret vector $(\mathbf{x}, r)$ containing all the relevant coefficients $x_1, \ldots, x_n$. For this reason, in our interactive proof Π_{d} for relation $\mathfrak{R}_{\mathsf{d}}$ the prover appends the first message of this Σ-protocol with a compact commitment $P = \text{COM}(\mathbf{x}, r; \gamma)$ to $(\mathbf{x}, r)$. After receiving the verifier's challenge c, the prover additionally invokes a compressed Σ-protocol to prove knowledge of an opening $(\mathbf{x}; r)$ of P that satisfies $L(\mathbf{x}) = y$ and $L_c(\mathbf{x}, r) = r + \sum_{i=1}^{n} c^i x_i = z$.

Interactive proof Π_{d} for relation $\mathfrak{R}_{\mathsf{d}}$ is described in Protocol 12 and its main properties are summarized in Theorem 3.14.

Protocol 12 Compactification Protocol Π_{d} for Dispersed Commitments.

Parameters:	$n + 1 = 2^\mu \in \mathbb{N}$, prime q, groups $(\mathbb{G}, +)$ and $(\mathbb{H}, \cdot)$ with exponent q, $L \in \text{Hom}(\mathbb{G}^n, \mathbb{G})$ and $\text{COM}\colon \mathbb{G}^n \times \mathsf{Rand} \to \mathbb{H}$ (homomorphic)
Public Input:	$P_1, \ldots, P_n \in \mathbb{H}$, $y \in \mathbb{G}$
Prover's Private Input:	$\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{G}^n$, $\gamma_1, \ldots, \gamma_s \in \mathsf{Rand}$
Prover's Claim:	$\text{COM}(x_i; \gamma_i) = P_i \ \forall i \wedge L(\mathbf{x}) = y$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
$r \leftarrow_R \mathbb{G}$		
$\rho, \gamma \leftarrow_R \mathsf{Rand}$		
$A = \text{COM}(r; \rho)$		
$P = \text{COM}(\mathbf{x}, r; \gamma)$		
	$\xrightarrow{A,P}$	
	$\xleftarrow{c}$	$c \leftarrow_R \mathbb{Z}_q$
$z = r + \sum_{i=1}^{n} c^i x_i$		
$\phi = \rho + \sum_{i=1}^{n} c^i \gamma_i$		
	$\xrightarrow{z,\phi}$	$\text{COM}(z; \phi) \stackrel{?}{=} A \cdot \prod_{i=1}^{n} P_i^{c^i}$

Run an amortized compressed Σ-protocol for proving knowledge of a commitment opening $(\mathbf{x}, r; \gamma)$ of P such that:

$$L(\mathbf{x}) = y \quad \text{and} \quad L_c(\mathbf{x}, r) := r + \sum_{i=1}^{n} c^i x_i = z\,.$$

Theorem 3.14 (Compactification Protocol for Dispersed Commitments)**.** *Let $n + 1 = 2^\mu$ for some $\mu \in \mathbb{N}$. Then the interactive proof Π_{d} for relation $\mathfrak{R}_{\mathsf{d}}$, described in Protocol 12, is perfectly complete, computationally $(n+1, 3, 2, 3, \ldots, 3)$-out-of-$(q, \ldots, q)$ special-sound, under the assumption that the commitment scheme*

is binding, and special honest-verifier zero-knowledge (SHVZK), under the assumption that the commitment scheme is perfectly hiding. Moreover, it has $(2\mu + 5)$ *communication rounds and the communication costs are:*

- $\mathcal{P} \to \mathcal{V}$*:* 3 *elements of* $\mathbb{G}$*,* $2\mu + 1$ *elements of* $\mathbb{H}$ *and* 2 *elements of* Rand*;*
- $\mathcal{V} \to \mathcal{P}$*:* $\mu + 2$ *elements of* $\mathbb{Z}_q$*.*

Proof. First observe that the amortized compressed Σ-protocol, invoked by interactive proof Π_{d}, amortizes the costs of opening the two linear forms by using the amortization technique from Section 3.4.2. Therefore, this compressed Σ-protocol is perfectly complete, computationally $(3, 2, 3 \ldots, 3)$-out-of-$(q, \ldots, q)$ special-sound, under the assumption that the commitment scheme is binding, and special honest-verifier zero-knowledge (SHVZK). Moreover, it has $(2\mu + 2)$ communication rounds and the communication costs are:

- $\mathcal{P} \to \mathcal{V}$: 2 elements of $\mathbb{G}$, $2\mu - 1$ elements of $\mathbb{H}$ and 1 element of Rand;
- $\mathcal{V} \to \mathcal{P}$: $\mu + 1$ elements of $\mathbb{Z}_q$.

From this the communication costs of Π_{d} follow. Let us now prove the remaining properties.

Completeness: This property follows immediately.

Special-Soundness: Suppose we are given an accepting $(1, 3, 2, 3, \ldots, 3)$-tree of protocol transcripts, i.e., all transcripts in the this tree start with the same messages (A, P, c, z, ϕ). By the $(3, 2, 3, \ldots, 3)$-out-of-$(q, \ldots, q)$ special-soundness of the compressed Σ-protocol that is invoked, an opening $(\bar{\mathbf{z}}, \bar{r}; \bar{\gamma})$ of P can be computed given this tree (under the assumption that the commitment scheme is binding). Moreover, this opening satisfies $L(\bar{\mathbf{z}}) = y$ and $L_c(\bar{\mathbf{z}}, \bar{r}) = z$.

An $(n+1, 3, 2, 3, \ldots, 3)$-tree of accepting transcripts corresponds to $n+1$ of these trees with common first message (A, P) and pairwise distinct challenges $c_0, \ldots, c_n \in \mathbb{Z}_q$. Hence, this tree corresponds to tuples

$$(A, P, c_i, z_i, \phi_i) \quad \text{and} \quad (\bar{\mathbf{z}}_i, \bar{r}_i; \bar{\gamma}_i)\,,$$

such that

$$\textsc{com}(\bar{\mathbf{z}}_i, \bar{r}_i; \bar{\gamma}_i) = P \wedge L(\bar{\mathbf{z}}_i) = y \wedge L_{c_i}(\bar{\mathbf{z}}_i, \bar{r}_i) = z_i \quad \forall 0 \leq i \leq n\,.$$

Therefore, we have either found distinct openings for the same commitment P, breaking the binding property of COM, or $(\bar{\mathbf{z}}_i, \bar{r}_i; \bar{\gamma}_i) = (\bar{\mathbf{z}}_j, \bar{r}_j; \bar{\gamma}_j)$ for all $i \neq j$. Let us assume the latter and write $(\bar{\mathbf{z}}, \bar{r}; \bar{\gamma}) := (\bar{\mathbf{z}}_i, \bar{r}_i; \bar{\gamma}_i)$.

The remainder of the proof now follows from the standard extraction procedure for the amortized Σ-protocol. More precisely, let

$$V = \begin{pmatrix} 1 & c_0 & \cdots & c_0^n \\ 1 & c_1 & \cdots & c_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & c_n & \cdots & c_n^n \end{pmatrix} \in \mathbb{Z}_q^{(n+1)\times(n+1)}\,,$$

be the invertible Vandermonde matrix defined by the pairwise distinct challenges $c_0, \ldots, c_n$. Further, let

$$\begin{pmatrix} \tilde{z}_0 \\ \vdots \\ \tilde{z}_n \end{pmatrix} = V^{-1} \begin{pmatrix} z_0 \\ \vdots \\ z_n \end{pmatrix} \in \mathbb{G}^{n+1} \quad \text{and} \quad \begin{pmatrix} \tilde{\phi}_0 \\ \vdots \\ \tilde{\phi}_n \end{pmatrix} = V^{-1} \begin{pmatrix} \phi_0 \\ \vdots \\ \phi_n \end{pmatrix} \in \mathsf{Rand}^{n+1} .$$

Then, $\textsc{com}(\tilde{z}_i; \tilde{\phi}_i) = P_i$ for all $1 \leq i \leq n$. Moreover, since $L_{c_i}(\bar{\mathbf{z}}, \bar{r}) = z_i$, it follows that $\bar{\mathbf{z}} = (\tilde{z}_1, \ldots, \tilde{z}_n)$. Finally, recall that $L(\bar{\mathbf{z}}) = y$, i.e., $(\bar{\mathbf{z}}, \tilde{\phi}_1, \ldots, \tilde{\phi}_n)$ is a witness for $(P_1, \ldots, P_n, y)$, which proves the required special-soundness property.

SHVZK: Transcript for statements $(P_1, \ldots, P_n, y)$ that admit a witness are simulated as follows. Sample $\mu + 2$ challenges $(c, c_0, \ldots, c_\mu)$ and $z \leftarrow_R \mathbb{G}$ and $\gamma, \phi \leftarrow_R \mathsf{Rand}$ uniformly at random, and compute $P = \textsc{com}(0; \gamma)$ and $A = \textsc{com}(z; \phi) \cdot \prod_{i=1}^{n} P_i^{-c^i}$.

Then, since $(P_1, \ldots, P_n, y)$ admits a witness and COM is perfectly hiding, the public statement (P, y, z), for the invoked compressed Σ-protocol, admits a witness. Therefore, it is possible to run the SHVZK simulator for this protocol, given the statement (P, y, z) and the $\mu + 1$ challenges $(c_0, \ldots, c_\mu)$ sampled before, to obtain a protocol transcript tr. The SHVZK simulator for Π_{d} then outputs transcript

$$(A, P, c, z, \phi, \mathsf{tr}) .$$

Because the commitment scheme is perfectly hiding, simulated transcripts have exactly the same distribution as honestly generated ones, which completes the proof of theorem.

□

CHAPTER 4

4

Compressed Σ-Protocols: Higher Level Functionalities

4.1 Introduction

Instantiating compressed Σ-protocols with a homomorphic and compact vector commitment scheme establishes an honest-verifier zero-knowledge proof for opening linear forms L on committed vectors $\mathbf{x}$. More precisely, its most basic variant is a protocol for proving knowledge of a commitment opening $(\mathbf{x}; \gamma)$ satisfying the *linear* constraint $L(\mathbf{x}) = y$. This functionality might seem somewhat restrictive; in many practical scenarios the statement one wishes to prove cannot be captured by a linear constraint *directly*. In this chapter, we enhance this basic linear functionality by treating two (classes of) relations that cannot be captured by a homomorphism directly. In both cases our strategy is to reduce the relation to our desired starting point, i.e., a prover claiming to know a homomorphism preimage.

First, in Section 4.2, we consider proving the correctness of a large set of committed multiplication triples $(\alpha_i, \beta_i, \gamma_i = \alpha_i\beta_i) \in \mathbb{Z}_q^3$. The corresponding multiplicative relation is clearly nonlinear and therefore cannot be captured by a homomorphism directly. Our approach is to *linearize* this relation to bring about the desired starting point and, subsequently, apply a compressed Σ-protocol. This approach is based on the work of [CDP12] that shows how to prove *arbitrary constraints* on committed vectors by exploiting techniques from secure multi-party computation (MPC) based on arithmetic secret sharing. More concretely, our work is based on the ideas underlying the *commitment multiplication* protocol from [CDM00]. It is this combination of "compact commitments with linear openings" and arithmetic secret sharing that allows for "linearizing nonlinear relations." This section is based on the article [AC20], co-authored by Ronald Cramer.

Second, in Section 4.3, we consider a prover that claims to know the homomorphism preimages for a *subset* of public elements $P_1, \ldots, P_n$, i.e., a prover claiming to have *partial* knowledge about the preimages of these elements. Proofs of partial knowledge were introduced in [CDS94]. Their solution combines Σ-protocol theory and linear secret sharing, and achieves linear communication complexity. We present a Σ-protocol, inspired by the [CDS94]-approach, for linearizing the proof of partial knowledge relation. However, a careful re-design of the original protocol is necessary to allow for compression. After composing with the appropriate

compressed Σ-protocol, we establish a proof of partial knowledge with logarithmic communication complexity. This section is based on the article [ACF21], co-authored by Ronald Cramer and Serge Fehr.

4.2 An Arithmetic Secret-Sharing Based Linearization Technique

The main result of [CDP12] is a zero-knowledge protocol for proving the correctness of a large number of committed multiplication triples $(\alpha_i, \beta_i, \gamma_i = \alpha_i\beta_i) \in \mathbb{Z}_q^3$. Their technique requires some adaptations to make it work for us here. In Section 4.2.1, we first outline the technique from [CDP12] and then discuss the required adaptations. These adaptations allow us to linearize the nonlinear relations defined by multiplication triples. Combined with our compressed Σ-protocols for opening linear forms, we obtain an interactive proof that allows a prover to commit to a large vector of multiplication triples and prove that the committed vector is of the appropriate form.

In practice, it may happen that the prover is already committed to the vector of multiplication triples *before* being asked to prove its correctness. This is referred to as the "commit-and-prove" scenario. In order to deal with this scenario some further utility enhancements are needed. The required enhancements, based on the compactification techniques of Section 3.4.4, are described in Section 4.2.2.

Finally, the linearization technique of Section 4.2.1 requires $q > 3m$, i.e., the multiplication triples must be defined over a large enough field $\mathbb{Z}_q$. In Section 4.2.3, we show how to handle the case $q \leq 3m$.

4.2.1 Proving Correctness of Multiplication Triples

Let us first outline the technique from [CDP12] for proving the correctness of committed multiplication triples. Subsequently, we describe our adaptations to this technique. The work of [CDP12] considers homomorphic commitment schemes where the secret committed to is not a vector in $\mathbb{Z}_q^n$, but a *single element* of $\mathbb{Z}_q$ instead. Their primary result is a Σ-protocol showing the correctness of commitments to m multiplication triples $(\alpha_i, \beta_i, \gamma_i := \alpha_i\beta_i)$. In other words, each of the α_i's, β_i's and γ_i's is individually committed to, and the protocol verifies the multiplicative relations $\gamma_i = \alpha_i \cdot \beta_i$. The communication complexity of the [CDP12]-approach is linear in the number of multiplication triples m. Adaptations are required to make the protocol amenable for compression and reduce the communication complexity down to logarithmic.

Their solution employs multiplicative packed secret sharing (Section 2.10). For instance, consider Shamir's scheme over $\mathbb{Z}_q$, with privacy parameter $p = 1$, but with secret-space dimension m. This scheme uses random polynomials of degree $\leq m$, subject to the evaluations on the points $1, \ldots, m$ comprising the desired secret vector. Note that, for each sharing, a single random $\mathbb{Z}_q$-element is required (which can be taken as the evaluation at 0). Moreover, this packed secret scheme can be instantiated with $q - m$ players, with shares corresponding to the $q - m$ evaluations outside the points $1, \ldots, m$.

Remark 4.1. Actually, the above scheme can be instantiated with $q-m+1$ players by taking the evaluation at infinity as an additional share. Because this adaptation only has a minor impact on the properties of the protocol, we will ignore the point at infinity in our analysis. For more details see [CDN15].

It is important to note that, given a secret vector $\mathbf{x} \in \mathbb{Z}_q^m$ and random element $r \in \mathbb{Z}_q$, it holds by *Lagrange Interpolation* that, for each $c \in \mathbb{Z}_q$, the evaluation $f(c)$ of such polynomial $f(X)$ is some public $\mathbb{Z}_q$-linear combination over the coordinates of the secret vector and the random element. Namely, consider the map that, on input $m+1$ arbitrary evaluations on the points $0, \ldots, m$, outputs the (coefficients of the) unique polynomial $f(X)$ of degree $\leq m$ that maps the points $0, \ldots, m$ to these given evaluations. A transformation matrix describing this map corresponds to the inverse of the Vandermonde-matrix

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 4 & \cdots & 2^m \\ \vdots & \vdots & \ddots & \vdots & \\ 1 & m & m^2 & \cdots & m^m \end{pmatrix} \in \mathbb{Z}_q^{(m+1)\times(m+1)}.$$

Composed with the *linear* evaluation at c mapping, this transformation describes the desired $\mathbb{Z}_q$-linear combination.

Now, assume that $3m < q$. In this case, the total number of shares $q - m$ is at least $2m + 1$ and the above instantiation of Shamir's secret-sharing scheme is multiplicative. More precisely, the secret-sharing scheme has $(2m + 1)$-product-reconstruction. In Section 4.2.3, we describe how to handle the case $3m \geq q$. The [CDP12]-protocol goes as follows.

- The vectors of commitments to the multiplication triples are assumed to be part of the common input.
- The prover selects a random polynomial $f(X)$ of degree at most m that defines a packed secret sharing of the vector $(\alpha_1, \ldots, \alpha_m)$. The prover also selects a random polynomial $g(X)$ of degree at most m that defines a packed secret sharing of the vector $(\beta_1, \ldots, \beta_m)$. Finally, the prover computes the product polynomial $h(X) := f(X)g(X)$ of degree at most $2m < q$.
- The prover commits to the random $\mathbb{Z}_q$-element for the sharing based on $f(X)$, i.e., $f(0)$, and commits to the random $\mathbb{Z}_q$-element for the sharing based on $g(X)$, i.e., $g(0)$. The prover also commits to the evaluations of $h(X)$ on the points $0, m + 1, \ldots, 2m$.[1] Note that the "absent" evaluations at $1, \ldots, m$ comprise the γ_i's and their commitments are already assumed to be part of the common input.
- The prover sends these $m + 3$ commitments to the verifier.
- The verifier selects a random challenge $c \in \mathbb{Z}_q$, distinct from $1, \ldots, m$, and sends it to the prover.

[1] By Lagrange interpolation these points, together with the γ_i's, determine $h(X)$.

- By public linear combinations, both prover and verifier can compute three commitments: one to $u := f(c)$, one to $v := g(c)$ and one to $w := h(c)$. The prover opens each of these (assuming, of course, that $c \notin \{1, \ldots, m\}$).
- The verifier checks the three openings and verifies that $u \cdot v = w$.

If the committed polynomials do not satisfy $f(X)g(X) = h(X)$, and under the assumption that the commitment scheme is binding, there are at most $2m$ values of c out of the $q - m$ possibilities such that the final check goes through. So a dishonest prover succeeds with probability at most $2m/(q-m)$, which is smaller than 1 since $3m < q$. More precisely, the protocol can be shown to be $(2m+1)$-out-of-$(q-m)$ special-sound under the assumption that the commitment scheme is binding. Honest-verifier zero-knowledge essentially follows from 1-privacy of the secret-sharing scheme.

We now make the following observation. In the above protocol, the prover may as well use our compressed Σ-protocol for opening linear forms as a black-box. Indeed, the entire $(4m+3)$-dimensional $\mathbb{Z}_q$-vector

$$\mathbf{y} = \big(\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_m, \gamma_1, \ldots, \gamma_m, f(0), g(0), h(0), h(m+1), \ldots, h(2m)\big)$$

of data that the prover commits to in the protocol above can be committed to in a *single* compact commitment. Note that, by definition, $\gamma_i = h(i)$ for all $1 \leq i \leq m$, i.e., the γ_i's comprise the "missing" evaluations of $h(X)$. Furthermore, all of the data *opened* to the verifier is some fixed linear form on the (long) secret committed vector $\mathbf{y}$. Indeed, each of the values u, v and w corresponds to an opening of a public linear form applied to $\mathbf{y}$. The linear form is determined by Lagrange interpolation as addressed above, under the convention that the form takes zeros on the portion of the coordinates of $\mathbf{y}$ not relevant to the computation, i.e., all three linear forms correspond to the evaluation of a polynomial whose coefficients are defined by a different part of $\mathbf{y}$.

Overall, in this adaptation of the [CDP12]-protocol, the prover sends a single compact commitment to $\mathbf{y}$ to the verifier and, after receiving a challenge $c \leftarrow_R \mathbb{Z}_q \setminus \{1, \ldots, m\}$, the prover and verifier proceed by running a compressed Σ-protocol to open three different linear forms. This interactive proof for committing to m multiplication triples and proving the correctness of these triples only requires the prover to send $\mathcal{O}\big(\log(m)\big)$ elements.

4.2.1.1 Generalizing to Arbitrary Packed Secret-Sharing Schemes

For concreteness, the linearization technique has thus far been instantiated with Shamir's packed secret-sharing scheme. This scheme allows an m-dimensional vector with coefficients in a finite field to be secret shared amongst $q - m$ players. Moreover, the deployed scheme has 1-privacy, $(m+1)$-reconstruction and $(2m+1)$-product-reconstruction. Hence, if $2m + 1 \leq q - m$, or equivalently $q > 3m$, this scheme is multiplicative.

More generally, as long as $n \geq R$, the linearization technique can be instantiated with any n-player linear secret-sharing scheme (LSSS) $\mathcal{S}$ for m-dimensional vectors that has R-product-reconstruction and $(p \geq 1)$-privacy. As in Section 2.10, we

let $\widehat{\mathcal{S}}$ denote the LSSS such that every component-wise product $[\mathbf{a};\mathbf{r_a}]_{\mathcal{S}} * [\mathbf{b};\mathbf{r_b}]_{\mathcal{S}}$ of secret sharings is a secret sharing of the component-wise product $\mathbf{a} * \mathbf{b}$ with respect to $\widehat{\mathcal{S}}$. The linearization technique, now instantiated with $\mathcal{S}$, proceeds as follows.

- The prover samples $\mathbf{r}_\alpha, \mathbf{r}_\beta \leftarrow_R \mathbb{Z}_q^t$ uniformly at random and computes $\mathbf{r}_\gamma$ such that

$$[\alpha_1, \ldots, \alpha_m; \mathbf{r}_\alpha]_{\mathcal{S}} * [\beta_1, \ldots, \beta_m; \mathbf{r}_\beta]_{\mathcal{S}} = [\gamma_1, \ldots, \gamma_m; \mathbf{r}_\gamma]_{\widehat{\mathcal{S}}}.$$

- The prover commits to the long vector

$$\mathbf{y} = \left(\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_m, \gamma_1, \ldots, \gamma_m, \mathbf{r}_\alpha, \mathbf{r}_\beta, \mathbf{r}_\gamma\right).$$

  in a single compact commitment.

- The verifier selects an index $i \leftarrow_R \{1, \ldots, n\}$ uniformly at random and sends it to the prover.

- By linearity of the secret-sharing scheme, both prover and verifier can determine three linear forms L_1, L_2 and L_3: one corresponding to the i-th share $L_1(\mathbf{y}) = u$ of $[\alpha_1, \ldots, \alpha_m; \mathbf{r}_\alpha]_{\mathcal{S}}$ when evaluated in $\mathbf{y}$, one to the i-th share $L_2(\mathbf{y}) = v$ of $[\beta_1, \ldots, \beta_m; \mathbf{r}_\beta]_{\mathcal{S}}$ and one to the i-th share $L_3(\mathbf{y}) = w$ of $[\gamma_1, \ldots, \gamma_m; \mathbf{r}_\gamma]_{\widehat{\mathcal{S}}}$.

- The prover uses a compressed Σ-protocol to open the three linear forms L_1, L_2 and L_3 on the compactly committed vector $\mathbf{y}$.

- The verifier checks the three openings and checks whether $u \cdot v = w$.

The following lemma shows that, if $\alpha_i \cdot \beta_i \neq \gamma_i$ for some i, then $u \cdot v = w$ with probability at most $(R-1)/n$. Hence, assuming that the commitment scheme is binding, a dishonest prover succeeds with probability at most $(R-1)/n$ in convincing the verifier.

As before, honest-verifier zero-knowledge essentially follows from $(p \geq 1)$-privacy of the secret-sharing scheme. In fact, the verifier can ask the prover to open the shares of p different players instead of only one. For $p > 1$, this reduces the success probability of a dishonest prover from $(R-1)/n$ down to $\binom{R-1}{p}/\binom{n}{p}$.

Lemma 4.1 (Arithmetic Secret Sharing Based Linearization)**.** *Let $m, n, t, R \in \mathbb{N}$ with $R \leq n$, q prime and $\mathcal{S}$ the linear secret-sharing scheme (LSSS) defined by $M \in \mathbb{Z}_q^{n \times (m+t)}$. Further, let $\widehat{\mathcal{S}}$ be the LSSS defined by*

$$\widehat{M} = \begin{pmatrix} M_1 \otimes M_1 \\ \vdots \\ M_n \otimes M_n \end{pmatrix} \in \mathbb{Z}_q^{n \times (m+t)^2},$$

where M_i denotes the i-th row of M. Suppose that $\widehat{\mathcal{S}}$ has R-reconstruction or, equivalently, that $\mathcal{S}$ has R-product-reconstruction.

Then, for all $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{Z}_q^m$ *with* $\mathbf{a} * \mathbf{b} \neq \mathbf{c}$ *and for all* $\mathbf{r_a}$, $\mathbf{r_b}$ *and* $\mathbf{r_c}$, *it holds that the vectors*

$$[\mathbf{a}; \mathbf{r_a}]_{\mathcal{S}} * [\mathbf{b}; \mathbf{r_b}]_{\mathcal{S}} \in \mathbb{Z}_q^n \quad \textit{and} \quad [\mathbf{c}; \mathbf{r_c}]_{\widehat{\mathcal{S}}} \in \mathbb{Z}_q^n$$

coincide in at most $R - 1$ *coefficients.*

Proof. First, recall that the component-wise product $[\mathbf{a}; \mathbf{r_a}]_{\mathcal{S}} * [\mathbf{b}; \mathbf{r_b}]_{\mathcal{S}}$ of $\mathcal{S}$-sharings is a secret sharing of $\mathbf{a} * \mathbf{b}$ with respect to $\widehat{\mathcal{S}}$ (see Section 2.10), i.e.,

$$[\mathbf{a}; \mathbf{r_a}]_{\mathcal{S}} * [\mathbf{b}; \mathbf{r_b}]_{\mathcal{S}} = [\mathbf{a} * \mathbf{b}; \mathbf{r}]_{\widehat{\mathcal{S}}} \in \mathbb{Z}_q^n$$

for some vector $\mathbf{r}$.

Since $\widehat{\mathcal{S}}$ has R-reconstruction, any A of cardinality R of the secret sharing $[\mathbf{a} * \mathbf{b}; \mathbf{r}]_{\widehat{\mathcal{S}}}$ uniquely determines $\mathbf{a} * \mathbf{b}$. Hence, if there exists an R-subset A for which the shares of $[\mathbf{a} * \mathbf{b}; \mathbf{r}]_{\widehat{\mathcal{S}}}$ and $[\mathbf{c}; \mathbf{r_c}]_{\widehat{\mathcal{S}}}$ coincide, it follows that $\mathbf{a} * \mathbf{b} = \mathbf{c}$. This contradicts the assumption $\mathbf{a} * \mathbf{b} \neq \mathbf{c}$ and therefore such an R-subset A cannot exist. This completes the proof of the lemma. □

4.2.2 A Commit-and-Prove Variant

The compressed Σ-protocol for proving the correctness of m multiplication triples, described in Section 4.2.1, requires the prover to commit to the vector of triples

$$\mathbf{x} = \big(\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_m, \gamma_1, \ldots, \gamma_m\big) \in \mathbb{Z}_q^{3m}$$

and the auxiliary information

$$\mathsf{aux} = \big(f(0), g(0), h(0), h(m+1), \ldots, h(2m)\big) \in \mathbb{Z}_q^{m+3}$$

in a single compact commitment. By contrast, the original [CDP12]-protocol allows the prover to generate this auxiliary information *after* it has committed to the multiplication triples. A protocol where the prover can first commit to the secret input data and at a later point in time prove that the committed input satisfies some constraint, unknown at the time of committing to the input data, is called a *commit-and-prove* protocol. Hence, whereas the original [CDP12]-protocol is commit-and-prove, the compressed Σ-protocol of Section 4.2.1 is not.

In particular, note that the [CDP12]-protocol can be repeated arbitrarily many times, e.g., to prove to multiple verifiers that a fixed set of commitments comprises a set of committed multiplication triples. In each repetition the protocol generates fresh auxiliary information. By contrast, the compressed Σ-protocol variant outputs a commitment to multiplication triples together with the auxiliary information. Hence, repeating this protocol would output different commitments, allowing a dishonest prover to commit to different sets of multiplication triples in each invocation. Moreover, since the deployed secret-sharing scheme only has 1-privacy, a prover can not reuse a compact commitment to the long vector $\mathbf{y} = (\mathbf{x}, \mathsf{aux}) \in \mathbb{Z}_q^{4m+3}$. More precisely, it is crucial that the prover only opens the evaluations $f(c)$, $g(c)$ and $h(c)$ for a single challenge $c \in \mathbb{Z}_q \setminus \{1, \ldots, m\}$. This prevents the prover from reusing a given commitment to the long vector $\mathbf{y}$.

In many practical scenarios, commit-and-prove functionality is crucial. Fortunately, enhancing the compressed Σ-protocol for multiplication triples with this

functionality turns out to be merely a matter of plug-and-play with the basic theory.

To see this, suppose $P \in \mathbb{H}$ is a fixed compact commitment to the vector of m multiplication triples $\mathbf{x} \in \mathbb{Z}_q^{3m}$. We aim to bring about the desired starting point, i.e., a single compact commitment to multiplication triples and freshly generated auxiliary information $\mathsf{aux} \in \mathbb{Z}_q^{m+3}$. Let now Q be a commitment to the vector aux prepended with $3m$ zeros, i.e., to $(0, \mathsf{aux}) \in \mathbb{Z}_q^{4m+3}$ (here 0 denotes a vector of $3m$ zeros). Then, since the commitment scheme is assumed to be homomorphic, $P \cdot Q$ is the required compact commitment to the vector $\mathbf{y} = (\mathbf{x}, \mathsf{aux})$ containing both the multiplication triples and the auxiliary information. What remains is for the prover to convince the verifier that the commitment Q is of the appropriate form. More precisely, it must prove that Q is a commitment to a vector $(0, \mathsf{aux})$ starting with at least $3m$ zeros. This simply amounts to opening the $3m$ linear forms

$$L_i \colon \mathbb{Z}_q^{4m+3} \to \mathbb{Z}_q, \quad \mathbf{x} \mapsto x_i \,,$$

for $1 \leq i \leq 3m$. Namely, opening L_i to 0 on a compactly committed vector shows that the i-th coordinate of this vector equals 0.

The commit-and-prove variant thus runs two amortized compressed Σ-protocols for opening linear forms as subroutines. The first one opens the three linear forms, corresponding to the polynomial evaluations $f(c)$, $g(c)$ and $h(c)$, on the commitment $P \cdot Q$. The second one opens the n linear forms L_i, corresponding to the required nullity checks, on commitment Q. Recall that the costs of opening n different linear forms on a single compact commitment can be amortized (Section 3.4.2). Therefore, the naive commit-and-prove approach incurs roughly a factor two loss in communication efficiency. By deploying the compactification techniques of Section 3.4.4, this factor two loss can be avoided.

The foregoing describes how to handle the scenario where the prover is already committed to all multiplication triples in a single compact commitment P. Section 3.4.4 also shows how to handle the case where the prover is committed to all coefficients of the vector $\mathbf{x}$ of multiplication triples individually, i.e., in separate 1-dimensional commitments. Further, in Section 7.2.2, we deploy similar techniques to achieve a commit-and-prove circuit satisfiability protocol.

4.2.3 Correctness of Multiplication Triples in Small Fields

The linearization technique of Section 4.2.1 requires $q > 3m$, where m is the number of multiplication triples in the finite field $\mathbb{Z}_q$. In fact, linearization is well defined as long as $q > 2m$; the prover must commit to $2m + 1$ distinct evaluations of the polynomial $h(X) \in \mathbb{Z}_q[X]$. However, since the challenges are sampled from $\mathbb{Z}_q \setminus \{1, \ldots, m\}$, the linearization step itself is a $(2m + 1)$-out-of-$(q - m)$ special-sound Σ-protocol. In Chapter 6, we show that this implies a knowledge error $\kappa \geq 2m/(q-m)$. Hence, to ensure a nontrivial knowledge error $\kappa < 1$, we must even require $q > 3m$. In the discrete logarithm instantiation the prime q is exponential in the security parameter and, with m polynomial in the security parameter, this gives a negligible knowledge error. However, when the multiplication triples are defined over smaller fields, possibly even with cardinality $q \leq 3m$, the approach above does not suffice. In this section, we show that only minor adaptations are required when considering multiplication triples in small fields $\mathbb{Z}_q$.

So let, as before,

$$\mathbf{x} = \big(\alpha_1, \ldots, \alpha_m, \beta_1, \ldots, \beta_m, \gamma_1, \ldots, \gamma_m\big) \in \mathbb{Z}_q^{3m}$$

be a vector of m multiplication triples $(\alpha_i, \beta_i, \gamma_i)$, but now defined over a small field with cardinality $q \leq 3m$. To handle the fact that $q \leq 3m$, we simply define Shamir's secret-sharing scheme over a field extension $\mathbb{F}/\mathbb{Z}_q$ with cardinality at least $3m + 1$. More precisely, in this case $f(X) \in \mathbb{F}[X]$, $g(X) \in \mathbb{F}[X]$ and $h(X) = f(X)g(X)$ define packed secret sharings, with 1-privacy, of the α_i's, the β_i's and the γ_i's. Hence, the difference with before is that the polynomials are now defined over the field extension $\mathbb{F}$ instead of the base field $\mathbb{Z}_q$. Let d be the degree of the extension $\mathbb{F}/\mathbb{Z}_q$, i.e., $|\mathbb{F}| = q^d > 3m$. For simplicity, we enumerate the elements of $\mathbb{F}$, i.e., every $0 \leq i < q^d$ uniquely corresponds to a field element. Then, by choosing an appropriate basis of the extension $\mathbb{F}/\mathbb{Z}_q$, the vector

$$\mathbf{y} = \big(\mathbf{x}, f(0), g(0), h(0), h(m+1), \ldots, h(2m)\big) \in \mathbb{F}^{4m+3}\,,$$

containing all relevant information, can be viewed as a vector in $\mathbb{Z}_q^{3m+dm+3d}$. Here, we use that the coefficients of $\mathbf{x}$ are elements of the base field $\mathbb{Z}_q$. What remains is to observe that, also in this generalization, all evaluations of $f(X)$, $g(X)$ and $h(X)$ are accessible as $\mathbb{Z}_q$-affine combinations of the coefficients of $\mathbf{y}$.

Taking d such that $|\mathbb{F}| = q^d > 2m$ results in a well-defined linearization technique for multiplication triples in $\mathbb{Z}_q$. It is a perfectly complete and $(2m+1)$-out-of-$(q^d - m)$ special-sound Σ-protocol with knowledge error

$$\kappa = \frac{2m+1}{q^d - m}\,,$$

i.e., $q^d > 3m$ is required for the knowledge error to be nontrivial. Moreover, when composed with a compressed Σ-protocol for opening linear forms, the prover only needs to send $\mathcal{O}\big(\log(n + dm)\big)$ elements to the verifier.

Exactly the same approach applies to multiplication triples defined over a ring $\mathcal{R}$. In this case, the evaluation points of the Shamir polynomials $f(X), g(X), h(X) \in \mathcal{R}[X]$ should be chosen from an *exceptional* subset of $\mathcal{R}$. If the maximal size of such an exceptional subset is too small, i.e., at most $3m$, one simply defines the secret-sharing scheme over an appropriate ring extension of $\mathcal{R}$.

4.3 Proofs of Partial Knowledge

In a k-out-of-n proof of partial knowledge [CDS94] a prover knowing witnesses for some k-subset, i.e., subset of cardinality k, of n given public statements can convince the verifier of this claim without revealing which k-subset. Typically, the secrets are solutions to public instances of intractable problems, such as the discrete logarithm problem. The work of [CDS94] gives an elegant solution with linear communication complexity that combines Σ-protocol theory with linear secret sharing. Especially the "1-out-of-n" case $k = 1$ has seen myriad applications during the last decades, e.g., in electronic voting, ring signatures, and confidential transaction systems. Our goal is to construct proofs of partial knowledge with *logarithmic* communication complexity in both k and n.

4.3.1 Knowledge of k-out-of-n Homomorphism Preimages

Before we present our solution, let us formalize the k-out-of-n proof of partial knowledge problem. To this end, for a prime q and a group $(\mathbb{G}_T, +)$ with exponent q, let

$$\psi\colon \mathbb{Z}_q \to \mathbb{G}_T$$

be a homomorphism. The prover now claims to know the preimages for some k-subset of a set of n public group elements $y_1, \ldots, y_n \in \mathbb{G}_T$. We aim to construct an interactive proof for convincing a verifier of the veracity of this claim, without revealing the preimages or the k-subset. More precisely, we aim to construct an interactive proof for relation

$$\mathfrak{R}_{k\text{-out-of-}n} = \left\{ (y_1, \ldots, y_n; S, \mathbf{x}) : |S| = k,\, y_i = \psi(x_i)\ \forall i \in S \subseteq \{1, \ldots, n\} \right\}.$$

Note that, for notational convenience, the secret $\mathbf{x}$ is defined as a vector in $\mathbb{Z}_q^n$, while only the k coefficients $(x_i)_{i\in S}$ are relevant in this relation. Further, for simplicity we assume the domain of the homomorphism Ψ to be $\mathbb{Z}_q$. Our techniques are easily generalized to arbitrary domains $\mathbb{G}$. However, this would require a vector commitment scheme for *mixed* vectors with coefficients in both $\mathbb{Z}_q$ and $\mathbb{G}$, such as the commitment schemes presented in Section 5.3.

Inspired by the design principle of [CDS94], we reduce the k-out-of-n scenario to the n-out-of-n scenario by having the prover "eliminate" the preimages $(x_i)_{i\notin S}$ that it does not know, and then we apply an amortized compressed Σ-protocol to prove the n instances in one go. However, the original solution of [CDS94] to reduce the k-out-of-n to the n-out-of-n scenario, achieved by secret sharing the challenge, does not work for us, as the resulting protocol is not in the shape for the compression mechanism to apply. More precisely, the third message in the [CDS94]-protocol includes a consistent secret sharing of the challenge, which cannot be compressed.

Instead, we use the following solution. The prover first chooses an $(n-k+1)$-out-of-n Shamir secret sharing $\big(s_1 = p(1), \ldots, s_n = p(n)\big) \in \mathbb{Z}_q^n$ of the default secret $s = 1$, where it selects the non-constant "random" coefficients $a_1, \ldots, a_{n-k}$ of the sharing polynomial

$$p(X) = 1 + a_1 X + \cdots + a_{n-k} X^{n-k} \in \mathbb{Z}_q[X]$$

such that $s_i = 0$ for $i \notin S$. Hence, $p(X)$ is the unique polynomial of degree at most $n-k$ such that $p(0) = 1$ and $p(i) = 0$ for all $i \notin S$.

Now let $t_i = s_i x_i$ for any i, i.e., $t_i = 0$ for all $i \notin S$. The prover then commits to the vector

$$(\mathbf{a}, \mathbf{t}) = (a_1, \ldots, a_{n-k}, t_1, \ldots, t_n) \in \mathbb{Z}_q^{2n-k}$$

in a single compact commitment $P = \textsc{com}(\mathbf{x}; \gamma)$. We assume the commitment scheme $\textsc{com}\colon \mathbb{Z}_q^{2n-k} \times \mathsf{Rand} \to \mathbb{H}$ to be homomorphic.

What remains is for the prover to show that

$$\psi(t_i) = s_i y_i \tag{4.1}$$

for all $i \in \{1, \ldots, n\}$. Recall that $s_i = p(i) = 1 + \sum_{j=1}^{n-k} a_j i^j$. Thus, Equation 4.1 can be rewritten as

$$\phi_i(\mathbf{a}, \mathbf{t}) := \psi(t_i) - y_i \cdot \sum_{j=1}^{n-k} a_j i^j = y_i \,,$$

where the left hand side is a group homomorphism $\phi_i \colon \mathbb{Z}_q^{2n-k} \to \mathbb{G}_T$ evaluated in the committed vector $(\mathbf{a}, \mathbf{t})$. Hence, proving knowledge of an opening of commitment P that satisfies Equation 4.1 for all $1 \leq i \leq n$, is reduced to proving knowledge of a Ψ-preimage of $(P, y_1, \ldots, y_n)$, where

$$\Psi \colon \mathbb{Z}_q^{2n-k} \times \mathsf{Rand} \to \mathbb{H} \times \mathbb{G}_T^n, \quad (\mathbf{a}, \mathbf{t}; \gamma) \mapsto \big(\mathrm{COM}(\mathbf{a}, \mathbf{t}; \gamma), \phi_1(\mathbf{a}, \mathbf{t}), \ldots, \phi_n(\mathbf{a}, \mathbf{t})\big) \,.$$

In other words, in the final step of the k-out-of-n proof of partial knowledge, the prover opens n homomorphisms ϕ_i on the compactly committed vector $(\mathbf{a}, \mathbf{t})$.

For efficiency reasons, the costs of opening these n homomorphism can be amortized. More precisely, instead of opening the homomorphisms ϕ_i individually, the prover opens a single homomorphism $\Phi_c = \sum_{i=1}^{n} c^{i-1} \phi_i$ for a challenge $c \leftarrow_R \mathbb{Z}_q$ sampled uniformly at random by the verifier. This approach is a minor adaptation of the amortization technique presented in Section 3.4.2. The difference is that here the coefficients of the committed vector $(\mathbf{a}, \mathbf{t}) \in \mathbb{Z}_q^t$ are of a different type than the homomorphism openings $y_1, \ldots, y_n \in \mathbb{G}_T$. For this reason, the evaluations y_i can not be incorporated into the commitment.

Opening the homomorphism Φ_c with a standard Σ-protocol gives a novel secret-sharing based realization of [CDS94], with linear communication complexity. However, in contrast to the original [CDS94]-approach, this novel realization is now amenable to the compression techniques of Chapter 3, allowing us to reduce the communication complexity form linear down to logarithmic.

The resulting interactive proof, denoted $\Pi_{k\text{-out-of-}n}$, is formalized in Protocol 13. Its main properties are summarized in Theorem 4.1. In particular, note that the communication costs are logarithmic in both k and n. In this theorem, we minimize the communication costs by applying the compression mechanism $\log_2(2n-k) - 2$ times to reduce the dimension of the secret vector $(\mathbf{a}, \mathbf{t})$ from $2n-k$ down to 4. Namely, since every factor two reduction of the dimension comes at the cost of sending two $\mathbb{H}$-elements and two $\mathbb{G}_T$-elements, it is suboptimal to continue further and reduce the dimension of the witness down to two or even one. This also means that we implicitly assume that $n \geq 4$.

Theorem 4.1 (k-out-of-n Proof of Partial Knowledge)**.** *Let q be a prime and $k, n, \mu \in \mathbb{N}$ such that $k \leq n$ and $2n - k = 2^\mu$. Further, let $\psi \colon \mathbb{Z}_q \to \mathbb{G}_T$ be a homomorphism and* $\mathrm{COM} \colon \mathbb{Z}_q^n \times \mathsf{Rand} \to \mathbb{H}$ *a homomorphic vector commitment scheme.*

Then the compressed Σ-protocol $\Pi_{k\text{-out-of-}n}$ for relation $\mathfrak{R}_{k\text{-out-of-}n}$, described in Protocol 13, is perfectly complete, $(n, 2, 3, \ldots, 3)$-out-of-$(q, \ldots, q)$ special-sound, under assumption that the commitment scheme is binding, and special honest-verifier zero-knowledge (SHVZK), under the assumption that the commitment scheme is hiding. Moreover, it has $2\mu + 1$ communication rounds and the communication costs are:

- $\mathcal{P} \to \mathcal{V}$: *4 elements of* $\mathbb{Z}_q$, $2\mu - 3$ *elements of* $\mathbb{G}_T$, $2\mu - 2$ *elements of* $\mathbb{H}$ *and* 1 *element of* Rand;
- $\mathcal{V} \to \mathcal{P}$: μ *elements of* $\mathbb{Z}_q$.

Protocol 13 k-out-of-n Proof of Partial Knowledge $\Pi_{k\text{-out-of-}n}$.

Parameters:	$k, n \in \mathbb{N}$, prime q, groups $(\mathbb{G}_T, +)$ and $(\mathbb{H}, \cdot)$ with exponent q, $\psi \in \mathrm{Hom}(\mathbb{Z}_q, \mathbb{G}_T)$ and $\mathrm{COM}\colon \mathbb{Z}_q^{2n-k} \times \mathsf{Rand} \to \mathbb{H}$ (homomorphic)
Public Input:	$y_1, \ldots, y_n \in \mathbb{G}_T$
Prover's Private Input:	$S \subseteq \{1, \ldots, n\}$ with $\lvert S\rvert = k$, $x_1, \ldots, x_n \in \mathbb{Z}_q^n$
Prover's Claim:	$\psi(x_i) = y_i$ for all $i \in S$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
$p(X) = 1 + \sum_{i=1}^{n-k} a_i X^i$ s.t. $p(i) = 0 \quad \forall i \notin S$		
$\mathbf{t} = (p(1)x_1, \ldots, p(n)x_n)$		
$\gamma \leftarrow_R \mathsf{Rand}$ $P = \mathrm{COM}(\mathbf{a}, \mathbf{t}; \gamma)$		
	$\xrightarrow{P}$	
		$c \leftarrow_R \mathbb{Z}_q$
	$\xleftarrow{c}$	

Run the compressed Σ-protocol Σ_{comp} of Section 3.2.3 to prove knowledge of a preimage of $(P, \sum_{i=1}^{n} c^{i-1} y_i)$ with respect to homomorphism

$$\Psi\colon \mathbb{Z}_q^{2n-k} \times \mathsf{Rand} \to \mathbb{H} \times \mathbb{G}_T, \quad (\mathbf{a}, \mathbf{t}; \gamma) \mapsto \left(\mathrm{COM}(\mathbf{a}, \mathbf{t}; \gamma), \sum_{i=1}^{n} c^{i-1} \phi_i(\mathbf{a}, \mathbf{t})\right),$$

where $\phi_i(\mathbf{a}, \mathbf{t}) := \psi(t_i) - y_i \cdot \sum_{j=1}^{n-k} a_j i^j$ for all $1 \le i \le n$.

Proof. **Completeness:** This property follows from the completeness of the compressed Σ-protocol Σ_{comp}.

SHVZK: This property follows from the fact that the commitment P is hiding and from the corresponding zero-knowledge property of Σ_{comp}.

Special-Soundness: Similar to the proof of Theorem 3.12 it follows that, under the assumption that the commitment scheme is binding, there exists an extractor that, on input an $(n, 2, 3, \ldots, 3)$-tree of accepting transcripts, outputs an opening $(\mathbf{a}, \mathbf{t}; \gamma)$ to the commitment P such that $\phi_i(\mathbf{a}, \mathbf{t}) := y_i$ for all $1 \le i \le n$.

Let $p(X) = 1 + \sum_{j=1}^{n-k} a_j X^j$. Then, $\phi_i(\mathbf{a}, \mathbf{t}) := y_i$ can be rewritten as $\psi(t_i) = p(i)y_i$. Given the bounded degree of p and the non-zero constant coefficient, $p(i) = 0$ for at most $n-k$ choices of $i \in \{1, \ldots, n\}$. Thus, setting $S = \{i : p(i) \neq 0\}$, we have $|S| \geq k$, and for any $i \in S$ we can set $x_i := t_i/p(i)$. This then implies that $\psi(x_i) = y_i$ for all $i \in S$, which completes the proof. □

Example 4.1 (Discrete Logarithm Instantiations). Taking $\psi\colon \mathbb{Z}_q \to \mathbb{H}$, $x \mapsto h^x$ allows one to prove knowledge of the discrete logarithms of a k-subset of public group elements $P_1, \ldots, P_n \in (\mathbb{H}, \cdot)$. Moreover, it is easily seen that the proofs of partial knowledge immediately generalize to homomorphism $\psi\colon \mathbb{Z}_q^s \to \mathbb{G}_T$ with arbitrary input dimension s. This observation allows one to instantiate ψ as the Pedersen (vector) commitment function and prove knowledge of k-out-of-n commitment openings.

Remark 4.2. Similar to the linearization technique of Section 4.2, the proof of partial knowledge deploys Shamir's linear secret-sharing scheme (LSSS). However, the linearization technique for multiplication triples crucially depends on the multiplicativity of the LSSS. By contrast, the k-out-of-n proof of partial knowledge does not require multiplicativity and can be instantiated with any n player linear secret-sharing scheme that has $(n-k+1)$-reconstruction and $(n-k)$-privacy.

4.3.2 Pairing-Based Reduction of the Communication Costs

The amortized communication costs for opening the homomorphisms $\phi_i\colon \mathbb{Z}_q^{2n-k} \to \mathbb{G}_T$ are roughly $4\log_2(2n-k)$ elements. This is approximately a factor two larger than the communication costs for opening n linear forms $L_i\colon \mathbb{Z}_q^{2n-k} \to \mathbb{Z}_q$ (Section 3.4.2). The reason is that, for a linear form, the input and output coefficients are of the same type; both are $\mathbb{Z}_q$ elements. Therefore, using the techniques of Section 3.4.2, the linear form evaluations can be "incorporated" into the commitment. More precisely, opening n linear forms L_i on a compact commitment $P = \text{COM}(\mathbf{x}; \gamma)$ can be reduced to proving knowledge of a preimage for the homomorphism

$$\Psi_c\colon \mathbb{Z}_q^{2n-k} \times \mathsf{Rand} \to \mathbb{H}, \quad (\mathbf{x}; \gamma) \mapsto \text{COM}\Big(\mathbf{x}, \sum_{i=1}^{n} c^i L_i(\mathbf{x}); \gamma\Big),$$

where $c \leftarrow_R \mathbb{Z}_q$ is a challenge sampled uniformly at random by the verifier. Applying the same technique for the homomorphisms $\phi_i\colon \mathbb{Z}_q^{2n-k} \to \mathbb{G}_T$ requires a compact commitment scheme for *mixed* vectors $\big(\mathbf{x}, \sum_{i=1}^{n} c^i L_i(\mathbf{x})\big) \in \mathbb{Z}_q^{2n-k} \times \mathbb{G}_T$ containing both $\mathbb{Z}_q$ and $\mathbb{G}_T$ coefficients. In some settings, e.g., when proving knowledge of k-out-of-n discrete logarithms or Pedersen commitment openings, pairing-based commitment schemes with the required properties exist (Section 5.3). These commitment schemes allow the communication costs of the corresponding k-out-of-n proof of partial knowledge protocol to be reduced with a factor two, down to roughly $2\log_2(2n-k)$ elements. For more details we refer to [ACF21].

4.3.3 General Access Structures

Thus far, we have restricted ourselves to provers that claim to know the preimages of some (secret) subset S, of cardinality at least k, of n (public) elements $P_1, \ldots, P_n$, i.e., the secret subset S is an element of a *threshold* access structure

$$\Gamma_{k,n} = \{A \subseteq \{1, \ldots, n\} : |A| \geq k\} \subseteq 2^{\{1,\ldots,n\}}.$$

Here, we describe how the proof of partial knowledge can easily be generalized to arbitrary monotone access structures $\Gamma \subseteq 2^{\{1,\ldots,n\}}$, i.e., to provers that claim to know the preimages of some subset of $S \in \Gamma$ of n public elements. Recall that Γ is called a monotone access structure if for all $A \in \Gamma$ and for all $B \subseteq 2^{\{1,\ldots,n\}}$ with $A \subseteq B$ it holds that $B \in \Gamma$. The proofs of partial knowledge of [CDS94] already considered arbitrary access structures and we adapt their techniques by combining them with our compression framework.

Our k-out-of-n proofs of partial knowledge implicitly deploy a linear secret-sharing scheme (LSSS) for access structure $\Gamma^*_{k,n} = \Gamma_{n-k,n}$. Here, Γ^* denotes the *dual* of access structure Γ, generally given by

$$\Gamma^* = \{A \subseteq \{1, \ldots, n\} : A^c \notin \Gamma\}.$$

More concretely the protocol of Section 4.3.1 uses Shamir's secret-sharing scheme and the polynomial $p(X) = 1 + \sum_{j=1}^{n-k} a_j X^j$ defines a secret sharing of the field element 1.

Now let Γ be a monotone access structure and $\mathcal{S}$ an LSSS for sharing field elements for access structure Γ^*. This implies that the adversary structure of $\mathcal{S}$ equals $\{S : S \notin \Gamma^*\}$, i.e., all player subsets are either qualified or unqualified [CDN15]. Depending on the access structure Γ^*, it might be required that shares are allowed to consist of several field elements.

Then, to construct a proof of partial knowledge for Γ, we simply replace $p(i)$ by the i-th share of a secret sharing of 1, with the randomness chosen so that the "right" shares (i.e., those corresponding to the x_i's that the prover does not know) vanish. Since the adversary structure of $\mathcal{S}$ equals $\{S : S \notin \Gamma\}$, the randomness can always be chosen such that the appropriate shares vanish, showing completeness of the generalized proof of partial knowledge. Special-soundness follows from the following observation. Let $A \subseteq \{1, \ldots, n\}$ be the subset for which all the corresponding shares vanish. Then, by linearity of the secret-sharing scheme and since the secret sharing reconstructs to 1, it follows that $A \notin \Gamma^*$. Hence, $A^c \in \Gamma$ and special-soundness follows as before.

The communication complexity of the resulting protocol depends logarithmically on the size of the LSSS for Γ^*, which is given by the monotone-span-program complexity of Γ^* [SJM91] and which coincides with the monotone-span-program complexity of Γ [Gál95].

CHAPTER 5

5 Suitable Cryptographic Platforms

5.1 Introduction

Thus far, we have seen how to prove knowledge of homomorphism preimages. One of the main applications of this functionality is *opening linear forms on compactly committed vectors*. More precisely, proving knowledge of a (vector) commitment opening that satisfies some arbitrary linear constraint captured by a linear form. Our compressed Σ-protocols require the vector commitment scheme to be *homomorphic*. Moreover, since in every iteration of the compression mechanism the prover sends two commitments, the communication complexity is only reduced if the commitment scheme is *compact*, or at least *compressing*. Recall that the size of a compact vector commitment is constant in the dimension n of the committed vector and the size of a compressing commitment is merely sublinear in n.

It is easy to see that compact commitments can be at most *computationally* binding; the domain of the commitment function is much larger than its codomain. For this reason, compact and homomorphic commitment schemes are to be based on computational assumptions. In this chapter, we will present a number of cryptographic platforms in which commitment schemes with the desired properties, and their corresponding compressed Σ-protocols, can be instantiated. The instantiations of this chapter are based on the papers [AC20; ACK21; ACR21], co-authored by Ronald Cramer, Lisa Kohl and Matthieu Rambaud.

5.2 Discrete Logarithm Assumption

The most prominent example of a compact and homomorphic vector commitment scheme is the Pedersen vector commitment scheme [Ped91]. This scheme allows a prover to commit to n-dimensional vectors[1] of field elements $\mathbf{x} \in \mathbb{Z}_q^n$, where q is a prime. A commitment is a single group element, regardless of the dimension n, i.e., commitments are indeed compact. The commitment scheme is perfectly hiding and computationally binding under the discrete logarithm assumption. Its formal definition is given below.

[1] Actually, Pedersen only introduced a commitment scheme for single elements $x \in \mathbb{Z}_q$. The vector commitment scheme presented here is a natural generalization and is therefore typically referred to as the Pedersen vector commitment scheme.

Definition 5.1 (Pedersen Vector Commitment Scheme [Ped91])**.** The Pedersen vector commitment scheme is defined by the following setup algorithm and commitment function:

- $\mathsf{pk} = (q, \mathbb{H}, g_1, \ldots, g_n, h) \leftarrow \textsc{Setup}(1^\lambda, n)$, where $(q, \mathbb{H}, \cdot) \leftarrow \mathcal{G}(1^\lambda)$ for a prime order group generator $\mathcal{G}(\cdot)$, i.e., $q = |\mathbb{H}|$ is prime, and

$$(\mathbf{g}, h) = (g_1, \ldots, g_n, h) \leftarrow_R \mathbb{H}^{n+1}$$

  are sampled uniformly at random;

- $\textsc{com}_{\mathsf{pk}} \colon \mathbb{Z}_q^n \times \mathbb{Z}_q \to \mathbb{H}, \quad (\mathbf{x}; \gamma) \mapsto \mathbf{g}^{\mathbf{x}} h^\gamma := h^\gamma \prod_{i=1}^n g_i^{x_i}$.

Recall that to commit to a vector $\mathbf{x} \in \mathbb{Z}_q^n$, the prover samples $\gamma \leftarrow_R \mathbb{Z}_q$ uniformly at random and outputs the commitment $\textsc{com}_{\mathsf{pk}}(\mathbf{x}; \gamma)$.

The Pedersen commitment function is a homomorphism, i.e., the compressed Σ-protocols of Chapter 3 apply. Since the randomness $\gamma \in \mathbb{Z}_q$ and coefficients $x_1, \ldots, x_n \in \mathbb{Z}_q$ of a Pedersen commitment $\textsc{com}_{\mathsf{pk}}(\mathbf{x}; \gamma)$ are all field elements, the randomness can be compressed too. More precisely, instead of compressing a vector $\mathbf{x}$ of dimension n, a vector $(\mathbf{x}; \gamma)$ of dimension $n+1$ will be compressed. This yields a minor improvement with respect to the abstract treatment of Section 3.4.

Theorem 5.1 now summarizes the main properties of the resulting compressed Σ-protocol for opening linear forms on Pedersen commitments. We immediately consider the most efficient variant of Theorem 3.11, where the linear form evaluation is incorporated into the commitment. More precisely, compression is applied to the homomorphism

$$\Psi \colon \mathbb{Z}_q^{n+1} \to \mathbb{H}, \quad (\mathbf{x}; \gamma) \mapsto \textsc{com}_{\mathsf{pk}}(\mathbf{x}, c \cdot L(\mathbf{x}); \gamma)\,,$$

for some challenge $c \leftarrow_R \mathbb{Z}_q$ sent by the verifier in the first round of the protocol. This variant has *computational* special-soundness. At the cost of increasing the communication costs by roughly a factor two, or by using the techniques from Section 3.4.3, this compressed Σ-protocol can be made *unconditionally* special-sound.

Theorem 5.1 (Compressed Σ-Protocol for Pedersen Commitments)**.** *Let $n + 1 = 2^\mu$ for some $\mu \in \mathbb{N}$, $\textsc{com}_{pk}$ the Pedersen vector commitment scheme instantiated with public key $\mathsf{pk} = (q, \mathbb{H}, g_1, \ldots, g_n, h)$ and $L \colon \mathbb{Z}_q^n \to \mathbb{Z}_q$ a linear form.*

Then the compressed Σ-protocol for relation

$$\mathfrak{R}_{Ped} = \{(P, y; \mathbf{x}, \gamma) : \mathbf{g}^{\mathbf{x}} h^\gamma = P \wedge L(\mathbf{x}) = y\}\,,$$

is perfectly complete, computationally $(2, 2, 3, \ldots, 3)$-out-of-$(q, \ldots, q)$ special-sound, under the discrete logarithm assumption, and special honest-verifier zero-knowledge (SHVZK). Moreover, it has $(2\mu + 2)$ communication rounds and the communication costs are:

- *$\mathcal{P} \to \mathcal{V}$: 2 elements of $\mathbb{Z}_q$ and $2\mu - 1$ elements of $\mathbb{H}$;*
- *$\mathcal{V} \to \mathcal{P}$: $\mu + 1$ elements of $\mathbb{Z}_q$.*

5.3 Pairing-Based Platform

In a pairing-based platform, the Pedersen commitment scheme has a straightforward adaptation to accommodate vectors of group, rather than field, elements [AFG+10]. More precisely, let $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{H}$ be a (nondegenerate) bilinear mapping between groups $(\mathbb{G}_1, +)$, $(\mathbb{G}_2, +)$ and $(\mathbb{H}, \cdot)$ of prime order q, i.e., e is a pairing and $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}, e)$ is a bilinear group. The adapted Pedersen vector commitment scheme allows a prover to commit to vectors $\mathbf{x}$ in $\mathbb{G}_1^n$ or $\mathbb{G}_2^n$. Lai et al. further extended this approach to commitments to mixed vectors in $\mathbb{Z}_q^{n_0} \times \mathbb{G}_1^{n_1}$ or, analogously, $\mathbb{Z}_q^{n_0} \times \mathbb{G}_2^{n_2}$ [LMR19]. Definition 5.2 formalizes this commitment scheme.

Definition 5.2 (Extended Pedersen Commitment Scheme [AFG+10; LMR19])**.** The following setup algorithm and commitment function define a pairing-based extension of the Pedersen Vector commitment scheme:

- Setup:
$$\mathsf{pk} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}, e, \mathbf{g}, \mathbf{h}, h) \leftarrow \textsc{Setup}(1^\lambda, n_0, n_1)\,,$$
where $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}, e) \leftarrow \mathcal{G}(1^\lambda)$ for a bilinear group generator $\mathcal{G}(\cdot)$ and $(\mathbf{g}, \mathbf{h}, h) \leftarrow_R \mathbb{G}_2^{n_1} \times \mathbb{H}^{n_0} \times \mathbb{H}$ are sampled uniformly at random.

- Commitment Function:
$$\textsc{com}_{\mathsf{pk}}\colon \mathbb{Z}_q^{n_0} \times \mathbb{G}_1^{n_1} \times \mathbb{Z}_q \to \mathbb{H}, \quad (\mathbf{x}, \mathbf{y}; \gamma) \mapsto \mathbf{h}^{\mathbf{x}} \cdot e(\mathbf{y}, \mathbf{g}) \cdot h^\gamma\,,$$
where $\mathbf{h}^{\mathbf{x}} := \prod_{i=1}^{n_0} h_i^{x_i}$ and $e(\mathbf{y}, \mathbf{g}) := \prod_{i=1}^{n_1} e(y_i, g_i)$.

This commitment scheme is perfectly hiding and computationally binding under the *double pairing assumption*. Informally, this assumption states that it is hard to find elements $r_1, r_2 \in \mathbb{G}_1$ such that $e(r_1, g_1)e(r_2, g_2) = 1$ for random $g_1, g_2 \in \mathbb{G}_2$. Abe et al. showed that the double pairing assumption is implied by the decisional Diffie-Hellman (DDH) assumption in $\mathbb{G}_2$ [AFG+10]. Therefore, the above commitment scheme is computationally binding under the DDH assumption in $\mathbb{G}_2$.

Note that the double pairing assumption does not hold in *symmetric* bilinear groups, i.e., when $\mathbb{G}_1 = \mathbb{G}_2$. Namely, in this case $e(-g_2, g_1)e(g_1, g_2) = 1$ for all $g_1, g_2 \in (\mathbb{G}_2, +)$. Similarly, it is easily seen that the DDH assumption does not hold in $\mathbb{G}_2$ if there exists a pairing $e\colon \mathbb{G}_2 \times \mathbb{G}_2 \to \mathbb{H}$. For this reason, we require the bilinear group to be asymmetrical. If the DDH assumption holds in both $\mathbb{G}_1$ and $\mathbb{G}_2$, we also say that the *symmetrical external Diffie-Hellman* (SXDH) assumption holds.

Abe et al. observed that the commitment scheme of Definition 5.2 introduces an alternative for Pedersen commitments to vectors of field elements [AFG+10]. Namely, a commitment to n different n-dimensional Pedersen commitments is a commitment to an n^2-dimensional $\mathbb{Z}_q$-vector. This two-tiered commitment scheme only requires $2n + 1$ public group elements. By contrast, Pedersen's commitment scheme requires $n^2 + 1$ public group elements to commit to an n^2-dimensional $\mathbb{Z}_q$-vector. Replacing the Pedersen vector commitment scheme in Theorem 5.1 by this two-tiered approach results in a compressed Σ-protocol with exactly the same

communication costs, but with a square root improvement in the size of the public parameters.

In addition, Lai et al. show how this approach can be extended to construct a commitment scheme for vectors with coefficients in $\mathbb{Z}_q$, $\mathbb{G}_1$ and $\mathbb{G}_2$ [LMR19]. In contrast to previous schemes, a commitment to a vector $\mathbf{x} \in \mathbb{Z}_q^{n_0} \times \mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2}$ consists of two elements in the group $\mathbb{H}$. The reason is that $(x, y) = (g_1, -g_2)$ is a nontrivial solution for the equation $e(x, g_2)e(g_1, y) = 1$ for any $(g_1, g_2) \in \mathbb{G}_1 \times \mathbb{G}_2$. Such a solution would break the binding property of the naive generalization of Definition 5.2 in which commitments consist of only one target group element. However, with high probability, there does not exist a solution $(x, y) \in \mathbb{G}_1 \times \mathbb{G}_2$ to the system of equations $e(x, g_2)e(g_1, y) = 1$ and $e(x, g_2')e(g_1', y) = 1$, where $(g_1, g_2), (g_1', g_2') \in \mathbb{G}_1 \times \mathbb{G}_2$ are sampled uniformly at random. For this reason, the commitments consist of two target group elements and, under the SXDH assumption, breaking their binding property can be reduced to solving a similar system of equations. The resulting commitment scheme is described in Definition 5.3. It is computationally hiding under the DDH assumption in $\mathbb{G}_T$, and it is computationally binding under the SXDH assumption. The scheme can be made perfectly hiding by introducing an additional randomizer $\gamma_2 \in \mathbb{Z}_q$.

Definition 5.3 (Compact Commitments to $(\mathbb{Z}_q, \mathbb{G}_1, \mathbb{G}_2)$-Vectors [LMR19])**.** The following setup algorithm and commitment function define a pairing-based extension of the Pedersen Vector commitment scheme:

- Setup:

$$\mathsf{pk} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}, e, \mathbf{g}_1, \mathbf{g}_1', \mathbf{g}_2, \mathbf{g}_2', \mathbf{h}, \mathbf{h}', h, h') \leftarrow \textsc{Setup}(1^\lambda, n_0, n_1, n_2)\,,$$

  where $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}, e) \leftarrow \mathcal{G}(1^\lambda)$ for a bilinear group generator $\mathcal{G}(\cdot)$ and $(\mathbf{g}_1, \mathbf{g}_1', \mathbf{g}_2, \mathbf{g}_2', \mathbf{h}, \mathbf{h}', h, h') \leftarrow_R \mathbb{G}_1^{2n_2} \times \mathbb{G}_2^{2n_1} \times \mathbb{H}^{2n_0+2}$ are sampled uniformly at random.

- Commitment Function: $\textsc{com}_{\mathsf{pk}} \colon \mathbb{Z}_q^{n_0} \times \mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2} \times \mathbb{Z}_q^2 \to \mathbb{H}$,

$$(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2; \gamma) \mapsto \begin{pmatrix} \mathbf{h}^{\mathbf{x}_0} \cdot e(\mathbf{x}_1, \mathbf{g}_2) \cdot e(\mathbf{g}_1, \mathbf{x}_2) \cdot h_1^{\gamma} \\ \mathbf{h}'^{\mathbf{x}_0} \cdot e(\mathbf{x}_1, \mathbf{g}_2') \cdot e(\mathbf{g}_1', \mathbf{x}_2) \cdot h_1'^{\gamma} \end{pmatrix}.$$

The commitment scheme of Definition 5.3 is a homomorphic and compact commitment scheme for mixed vectors $\mathbf{x} \in \mathbb{Z}_q^{n_0} \times \mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2}$. However, it does not allow a prover to commit to elements of the target group $\mathbb{H}$ of the pairing $e \colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{H}$. Unfortunately, we do not know how to do this compactly while preserving the required homomorphic properties. For this reason, we introduce the homomorphic commitment scheme of Definition 5.4. This scheme is based on the ElGamal encryption scheme [ElG84]. The commitment scheme is unconditionally binding and computationally hiding under the DDH assumption in $\mathbb{G}_T$. However, in contrast to the previous commitment schemes, it is not compact. More precisely, an ElGamal commitment to a vector $\mathbf{x}_T \in \mathbb{H}^{n_T}$ contains $n_T + 1$ group elements.

Definition 5.4 (ElGamal Commitment Scheme)**.** The ElGamal vector commitment scheme is defined by the following setup algorithm and commitment function:

- $\mathsf{pk} = (q, \mathbb{H}, G_1, \ldots, G_{n_T}, H) \leftarrow \textsc{Setup}(1^\lambda, n_T)$, where $(q, \mathbb{H}, \cdot) \leftarrow \mathcal{G}(1^\lambda)$ for a prime order group generator $\mathcal{G}(\cdot)$ and $(\mathbf{G}, H) = (G_1, \ldots, G_{n_T}, H) \leftarrow_R \mathbb{H}^{n_T+1}$ are sampled uniformly at random;
- $\textsc{com}_{\mathsf{pk}} \colon \mathbb{H}^{n_T} \times \mathbb{Z}_q \to \mathbb{H}^{N_T+1}, \quad (\mathbf{x}_T; \rho) \mapsto \begin{pmatrix} H^\rho \\ \mathbf{G}^\rho * \mathbf{x}_T \end{pmatrix}$,
  where $\mathbf{G}^\rho := (G_1^\rho, \ldots, G_{n_T}^\rho)$ and $*$ denotes the component-wise product.

Combined, the commitment schemes of Definition 5.3 and Definition 5.4 provide a homomorphic commitment scheme for *bilinear group vectors*

$$\mathbf{x} \in \mathbb{Z}_q^{n_0} \times \mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2} \times \mathbb{H}^{n_T} .$$

This commitment scheme is only compact in the dimension n_0, n_1 and n_2; the size of a commitment is linear in the dimension n_T of the $\mathbb{H}$-component. For completeness we have included the definition of the resulting commitment scheme for bilinear group vectors.

Definition 5.5 (Bilinear Group Vector Commitment Scheme [LMR19])**.** The following setup algorithm and commitment function define a bilinear group vector commitment scheme:

- Setup:

$$\mathsf{pk} = \begin{pmatrix} q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}, e, \mathbf{g}_1, \mathbf{g}_1', \mathbf{g}_2, \\ \mathbf{g}_2', \mathbf{h}, \mathbf{h}', \mathbf{G}, h, h', H \end{pmatrix} \leftarrow \textsc{Setup}(1^\lambda, n_0, n_1, n_2, n_T) ,$$

  where $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}, e) \leftarrow \mathcal{G}(1^\lambda)$ for a bilinear group generator $\mathcal{G}(\cdot)$ and

$$(\mathbf{g}_1, \mathbf{g}_1', \mathbf{g}_2, \mathbf{g}_2', \mathbf{h}, \mathbf{h}', \mathbf{G}, h, h', H) \leftarrow_R \mathbb{G}_1^{2n_2} \times \mathbb{G}_2^{2n_1} \times \mathbb{H}^{2n_0+n_T+3}$$

  are sampled uniformly at random.

- Commitment Function: $\textsc{com}_{\mathsf{pk}} \colon \mathbb{Z}_q^{n_0} \times \mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2} \times \mathbb{H}^{n_T} \times \mathbb{Z}_q^2 \to \mathbb{H}^{n_T+3}$,

$$(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_T; \gamma, \rho) \mapsto \begin{pmatrix} \mathbf{h}^{\mathbf{x}_0} \cdot e(\mathbf{x}_1, \mathbf{g}_2) \cdot e(\mathbf{g}_1, \mathbf{x}_2) \cdot h_1^\gamma \\ \mathbf{h}'^{\mathbf{x}_0} \cdot e(\mathbf{x}_1, \mathbf{g}_2') \cdot e(\mathbf{g}_1', \mathbf{x}_2) \cdot h_1'^\gamma \\ H^\rho \\ \mathbf{G}^\rho * \mathbf{x} \end{pmatrix} .$$

A compressed Σ-protocol, instantiated with the above bilinear group vector commitment scheme, allows a prover to prove knowledge of a commitment opening satisfying a linear constraint $L(\mathbf{x}) = \mathbf{y}$ captured by a linear mapping

$$L \colon \mathbb{Z}_q^{n_0} \times \mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2} \times \mathbb{H}^{n_T} \to \mathbb{Z}_q \times \mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{H} .$$

As before, we apply the compressed Σ-protocol of Theorem 3.11, where the linear form evaluations are incorporated into the commitment. Note that, since the commitment scheme is only compact in its $(\mathbb{Z}_q, \mathbb{G}_1, \mathbb{G}_2)$-part, only the $(\mathbb{Z}_q, \mathbb{G}_1, \mathbb{G}_2)$-part of the L-evaluation should be incorporated into the commitment. For the

same reason, compression is only applied to the $(\mathbb{Z}_q, \mathbb{G}_1, \mathbb{G}_2)$-part of the committed vector. Theorem 5.2 summarizes the main properties of the compressed Σ-protocol for bilinear group vectors. For simplicity, we assume that $n_0 + 1 = n_1 = n_2$, but the result is easily extended to arbitrary input dimensions. Note that the communication complexity of this compressed Σ-protocol is logarithmic in n_0, n_1 and n_2, but linear in n_T.

Theorem 5.2 (Compressed Σ-Protocol for Bilinear Group Vectors)**.** *Let $n_0 + 1 = n_1 = n_2 = 2^\mu$ for some $\mu \in \mathbb{N}$, $n_T \in \mathbb{N}$, COM_{pk} the bilinear group vector commitment function instantiated with the bilinear group $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}, e)$ and $L\colon \mathbb{Z}_q^{n_0} \times \mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2} \times \mathbb{H}^{n_T} \to \mathbb{Z}_q \times \mathbb{G}_1 \times \mathbb{G}_2 \times \mathbb{H}$ linear.*

Then the compressed Σ-protocol for relation

$$\mathfrak{R}_{Bil} = \{(P, \mathbf{y}; \mathbf{x}, \gamma, \rho) : \text{COM}_{pk}(\mathbf{x}; \gamma, \rho) = P \wedge L(\mathbf{x}) = \mathbf{y}\},$$

is perfectly complete, computationally $(2, 2, 3, \ldots, 3)$-out-of-$(q, \ldots, q)$ special-sound, under the symmetrical external Diffie-Hellman (SXDH) assumption, and special honest-verifier zero-knowledge (SHVZK). Moreover, it has $(2\mu + 2)$ communication rounds and the communication costs are:

- *$\mathcal{P} \to \mathcal{V}$: 3 elements of $\mathbb{Z}_q$, 2 elements of $\mathbb{G}_1$, 2 elements of $\mathbb{G}_2$ and $6\mu + 2n_T - 3$ elements of $\mathbb{H}$;*
- *$\mathcal{V} \to \mathcal{P}$: $\mu + 1$ elements of $\mathbb{Z}_q$.*

5.4 Knowledge of Exponent Assumption

If one desires, the functionality of opening linear forms on compactly committed vectors can also be achieved from the *Knowledge-of-Exponent Assumption* (KEA). In order to introduce this assumption, let $\mathbb{H}$ be a group of prime order q and let us consider the following problem: on input $g, h = g^a \in \mathbb{H}$ output a pair $G, H \in \mathbb{H}$ with $G = H^a$. A simple solution to this problem is to output $G = g^c$ and $H = h^c$ for an arbitrary $c \in \mathbb{Z}_q$. Informally, the KEA states that this is the *only* way to solve this problem. More precisely, for any adversary that successfully outputs a pair (G, H) there exists an extractor that outputs the exponent c such that $G = g^c$ and thus $H = h^c$. We stress that the KEA is of a different nature than the discrete logarithm or decisional Diffie-Hellman assumption. KEA is not an intractability assumption and it is unfalsifiable [Nao03; BCP+14]. For these reasons, its application is not completely without controversy.

Opening linear forms on compact commitments instantiated from the KEA does not proceed by the standard compression paradigm. Namely, the basic protocol for this functionality already has *constant* communication complexity, i.e., compression is not needed. However, since the techniques of Section 7.2 only require *black-box* access to a protocol for opening linear forms on compactly committed vectors, they are equally applicable to a KEA instantiation. For this reason, we present the KEA approach here, even though it is not an instantiation of the compressed Σ-protocols of Chapter 3. Basing the linear form openings on the KEA results in *constant* communication complexity instead of logarithmic. However,

the resulting protocol does require a trusted setup. Below, we will elaborate on this trusted setup requirement.

We now describe the KEA based vector commitment scheme together with its protocol for opening linear forms. Our approach uses the techniques of [Gro10] and only minor adaptations are required.

A compact commitment to a vector $\mathbf{x} \in \mathbb{Z}_q^n$ is, as before, a Pedersen vector commitment $P = \mathbf{g}^{(\gamma,\mathbf{x})} := g_0^\gamma \prod_{i=1}^n g_i^{x_i}$. To prove knowledge of a commitment opening of P, the prover simply sends another Pedersen commitment $Q = \mathbf{h}^{(\gamma,\mathbf{x})}$ to $\mathbf{x}$, under the same randomness γ, using a different vector of group elements $\mathbf{h} = \mathbf{g}^\alpha = (g_0^\alpha, \ldots, g_n^\alpha) \in \mathbb{H}^{n+1}$. The value $\alpha \in \mathbb{Z}_q$ is sampled uniformly at random by a *trusted* party and is only shared with a *designated* verifier. Both vectors of groups elements are public. The proof Q is verified by checking that $Q = P^\alpha$, i.e., only a designated verifier that knows the secret value α can verify a proof. It is crucial that the prover does not know α, otherwise it can simply forge a proof by computing $Q = P^\alpha$.

The knowledge-of-exponent assumption states that an adversary capable of computing pairs (P, Q) with $Q = P^\alpha$, either knows α or an opening to P. From this assumption knowledge soundness follows. Correctness follows immediately and zero-knowledge follows since the proof Q is uniquely determined by P and α. In fact, the verifier can compute the proof $Q = P^\alpha$ without knowledge of $\mathbf{x}$. Hence, the proof Q does not reveal any additional information about the witness $\mathbf{x}$. Note that the resulting protocol only has one round, i.e., it is non-interactive, and its communication costs are independent of the dimension n.

Given a bilinear pairing $e\colon \mathbb{H} \times \mathbb{H} \to \mathbb{H}_T$, the verification procedure can be made public, i.e., given e, even parties that do not know α can verify a proof. In this case verification amounts to checking that $e(P, h_0) = e(g_0, Q)$.

If during the setup phase, the prover is only given the group elements h_0 and h_i for $i \in S \subseteq \{1, \ldots, n\}$, then the proof Q can only be computed if $x_i = 0$ for all $i \notin S$. Groth [Gro10] refers to the resulting proof as a *restriction* proof, since it actually shows that the nonzero entries of the committed vector $\mathbf{x}$ are restricted to the subset S of indices. The restriction proof is an important building block of our KEA-based protocol for opening linear forms on Pedersen commitments. Therefore, it is described in Protocol 14.

To additionally prove that the committed vector $\mathbf{x}$ satisfies the linear constraint $L(\mathbf{x}) = y$ some adaptations are required. More precisely, in this case, the group elements are sampled under the condition that $\mathbf{g} = (g, g^\beta, \ldots, g^{\beta^n})$ for some secret $\beta \in \mathbb{Z}_q$. The KEA that takes this additional structure into account is called the n-power Knowledge-of-Exponent Assumption (n-PKEA).

Groth [Gro10] showed that, using this additional structure, efficient circuit zero-knowledge protocols exist, i.e., protocols for proving knowledge of a secret vector $\mathbf{x} \in \mathbb{Z}_q^n$ such that $C(\mathbf{x}) = 0$ for some arbitrary arithmetic circuit C. Note that an arithmetic circuit constraint $C(\mathbf{x}) = 0$ is not necessarily linear. Groth's protocols can be adapted to our situation, where we simply wish to prove the validity of a *linear* constraint $L(\mathbf{x}) = y$ for some linear form $L\colon \mathbb{Z}_q^n \to \mathbb{Z}_q$. In Section 7.2, we will show how to handle nonlinear instances.

The adaptation of Groth's protocol relies on the following observation. Suppose that $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{Z}_q^n$ is such that $L(\mathbf{x}) = \langle \mathbf{a}, \mathbf{x} \rangle$ for all $\mathbf{x} \in \mathbb{Z}_q^n$, and let us

Protocol 14 KEA Restriction Proof for Pedersen Commitments.

PARAMETERS:	$n \in \mathbb{N}$, group $(\mathbb{H}, \cdot)$ of prime order q, pairing $e\colon \mathbb{H} \times \mathbb{H} \to \mathbb{H}_T$, $\mathbf{g} \in \mathbb{H}^{n+1}$, $h_0 = g_0^{\alpha}$, $h_i = g_i^{\alpha}$ for $i \in S \subseteq \{1, \ldots, n\}$ for a (secret) $\alpha \in \mathbb{Z}_q$
PUBLIC INPUT:	$P \in \mathbb{H}$
PROVER'S PRIVATE INPUT:	$\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_q^n$
PROVER'S CLAIM:	$\mathbf{g}^{(\gamma, \mathbf{x})} = g_0^{\gamma} \prod_{i=1}^{n} g_i^{x_i} = P \wedge x_i = 0 \; \forall i \notin S$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
$Q = h_0^{\gamma} \prod_{i \in S} h_i^{x_i}$	$\xrightarrow{\quad Q \quad}$	$e(P, h_0) \stackrel{?}{=} e(g_0, Q)$

define the following polynomials:

$$F(Y) = \gamma + \sum_{i=1}^{n} x_i Y^i\,, \quad G(Y) = \sum_{i=0}^{n-1} a_{n-i} Y^i\,,$$
$$\text{and} \quad H(Y) = F(Y)G(Y) = \sum_{i=0}^{2n-1} c_i Y^i\,.$$

Then the $(n+1)$-th coefficient of $H(Y)$ equals $c_n = \langle \mathbf{x}, \mathbf{a} \rangle = L(\mathbf{x})$. Moreover, since $\mathbf{g} = (g, g^{\beta}, \ldots, g^{\beta^n})$ for some secret $\alpha, \beta \in \mathbb{Z}_q$,

$$P = \mathbf{g}^{(\gamma, \mathbf{x})} = g^{F(\beta)}, \quad R := \mathbf{g}^{(a_n \ldots, a_1, 0)} = g^{G(\beta)} \quad \text{and} \quad e(P, R) = e\big(g, g^{H(\beta)}\big)\,.$$

Since $\mathbf{a}$ is public, both the prover and the verifier can compute the group element R. Hence, to prove that $L(\mathbf{x}) = \langle \mathbf{a}, \mathbf{x} \rangle = y$, the prover must convince the verifier that

$$e(P, R) = e\big(g, \mathbf{g}^{(c_0, \ldots, c_{n-1}, y, c_{n+1}, \ldots, c_{2n-1})}\big)\,, \tag{5.1}$$

for some $c_i \in \mathbb{Z}_q$. Note that we make some abuse of notation by implicitly assuming the vector $\mathbf{g}$ to be long enough, i.e., $\mathbf{g} = (g, g^{\beta}, \ldots, g^{\beta^{2n-1}}) \in \mathbb{H}^{2n}$.

To prove the validity of Equation 5.1, the prover sends the group element $S = \prod_{i \neq n} g_i^{c_i}$, where $c_0, \ldots, c_{2n-1}$ are the coefficients of the polynomial $H(Y)$. Subsequently, the verifier checks that

$$e(P, R) = e\big(g, S \cdot g_n^{y}\big)\,.$$

The proof is completed by adding the following group elements:

- A commitment opening proof Q for P, proving knowledge of an opening $(\mathbf{x}; \gamma) \in \mathbb{Z}_q^{n+1}$ of P;
- A restriction proof T for S, showing that the exponent vector $(c_0, \ldots, c_{2n-1})$ of S is zero in its $(n+1)$-th coordinate.

Note that the element Q is in fact a restriction proof; it shows that $\mathbf{x}$ is an n-dimensional vector, i.e., the commitment P does not make use of the public group elements

$$g^{\beta^{n+1}}, \ldots, g^{\beta^{2n-1}} \in \mathbb{H}\,.$$

The KEA based non-interactive proof for opening linear forms on Pedersen commitments is described in Protocol 15. It is crucial that the prover does not know the secret values $\alpha, \alpha', \beta \in \mathbb{Z}_q$. Therefore, these values must be generated in a trusted setup phase. The size of the proof is independent of the dimension n of the committed vector. This non-interactive proof is an adaptation of Groth's product argument [Gro10, Section 6]. Its (security) analysis requires somewhat different techniques and formalization then the ones used before. For this reason, we refer to [Gro10] for a more formal analysis.

Protocol 15 KEA Protocol for Opening Linear Forms.

Parameters:	$n \in \mathbb{N}$, group $(\mathbb{H}, \cdot)$ of prime order q, pairing $e\colon \mathbb{H} \times \mathbb{H} \to \mathbb{H}_T$ and vectors of $\mathbb{H}$-elements $\mathbf{g} = (g_0, \ldots, g_{2n-1}) = (g, g^{\beta}, \ldots, g^{\beta^{2n-1}})$, $\mathbf{k} = (g_0^{\alpha'}, \ldots, g_{n-1}^{\alpha'}, 1, g_{n+1}^{\alpha'}, \ldots, g_{2n-1}^{\alpha'})$ and $\mathbf{h} = (g_0^{\alpha}, \ldots, g_n^{\alpha})$ for (secret) $\alpha, \alpha', \beta \in \mathbb{Z}_q$
Public Input:	$P \in \mathbb{H}$, $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{Z}_q$ and $y \in \mathbb{Z}_q$
Prover's Private Input:	$\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{Z}_q^n$
Prover's Claim:	$g_0^{\gamma} \prod_{i=1}^{n} g_i^{x_i} = P \wedge L(\mathbf{x}) = \langle \mathbf{a}, \mathbf{x} \rangle = y$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
$F(Y) = \gamma + x_1 Y + \cdots + x_n Y^n$		
$G(Y) = a_n + a_{n-1} Y + \cdots + a_1 Y^{n-1}$		
$F(Y)G(Y) = c_0 + \cdots + c_{2n-1} Y^{2n-1}$		
$Q = \mathbf{h}^{(\gamma, \mathbf{x})}$		
$S = \prod_{i \neq n} g_i^{c_i}$		$R = \mathbf{g}^{(a_n, \ldots, a_1, 0)}$
$T = \prod_{i \neq n} k_i^{c_i}$		
	$\xrightarrow{Q,S,T}$	
		$e(P, h_0) \stackrel{?}{=} e(g_0, Q)$
		$e(S, k_0) \stackrel{?}{=} e(g_0, T)$
		$e(P, R) \stackrel{?}{=} e(g, S \cdot g_n^y)$

5.5 Strong-RSA Assumption

Let us now move to a compressed Σ-protocol instantiation based on the assumption that a dishonest prover does not know the order of some given group. More precisely, its security is based on the strong-RSA assumption (Definition 2.16). This instantiation is inspired by the strong-RSA based polynomial commitment scheme DARK [BFS20]. A polynomial commitment scheme allows a prover to commit to a polynomial $f \in \mathbb{Z}_q[X]$ of arbitrary degree and admits a protocol for "opening polynomial evaluations," i.e., a protocol for proving that a committed polynomial f satisfies $f(x) = y$ for some public $x, y \in \mathbb{Z}_q$. DARK is a strong-RSA based adaptation of the Bulletproof protocol [BCC+16; BBB+18], and it allows a prover to open polynomial evaluations with logarithmic communication complexity. However, Block et al. [BHR+21] identified a gap in the security analysis of DARK. Fortunately, they also proposed an adaptation of DARK, solving the aforementioned security gap at the cost of increasing the communication complexity from logarithmic to polylogarithmic.

Note that a polynomial $f(X) = \sum_{i=0}^{n} a_i X^i$ is uniquely defined by its coefficient vector and an evaluation of a polynomial is a special type of linear form evaluation, i.e.,

$$f(x) = \langle (a_0, \ldots, a_n), (1, x, \ldots, x^n) \rangle \,.$$

Hence, the functionality of a polynomial commitment scheme is strictly weaker than "opening linear forms on compactly committed vectors." Some of the techniques introduced in DARK [BFS20] and its adaptation [BHR+21] crucially depend on the structure of linear forms corresponding to polynomial evaluations, and are therefore not applicable to opening arbitrary linear forms. For this reason, we must modify the aforementioned approaches.

An important building block in these strong-RSA based interactive proofs is the following *integer* commitment scheme. To simplify the exposition, and in order to focus on the important aspects, we consider a *non-hiding* variant. For a statistically hiding variant of this commitment scheme we refer the reader to [DF02].

Definition 5.6 (Non-Hiding Integer Commitment Scheme [FO97; DF02])**.** The following setup algorithm and commitment function define a non-hiding integer commitment scheme:

- $\mathsf{pk} = (\mathbb{H}, g) \leftarrow \textsc{Setup}(1^\lambda)$, where $(\mathbb{H}, \cdot) \leftarrow \mathcal{G}(1^\lambda)$ for a hidden-order group generator $\mathcal{G}(\cdot)$ and $g \leftarrow_R \mathbb{H}$ is sampled uniformly at random;
- $\textsc{com}_{\mathsf{pk}} \colon \mathbb{Z} \to \mathbb{H}, \quad x \mapsto g^x$.

The commitment scheme of Definition 5.6 is homomorphic and computationally binding under the hidden order assumption (Definition 2.17), which is implied by the Strong-RSA assumption.

By appropriately encoding vectors of integers $\mathbf{x} \in \mathbb{Z}^n$, this commitment scheme allows a prover to commit to vectors of *bounded* integers. More precisely, let $\mathbb{Z}(\alpha) = \{x \in \mathbb{Z} : |x| < \alpha\}$ and

$$\mathrm{Enc}_Q \colon \mathbb{Z}(\alpha)^n \to \mathbb{Z}, \quad (x_1, \ldots, x_n) \mapsto \sum_{i=1}^{n} x_i Q^{i-1} \,,$$

for some integer $Q \geq 2\alpha$. Since $Q \geq 2\alpha$, this encoding is injective. Moreover, base Q decomposition provides an efficient decoding algorithm Dec_Q. A commitment to a bounded integer vector $\mathbf{x} \in \mathbb{Z}(\alpha)^n$ is simply an integer commitment to $\mathrm{Enc}_Q(\mathbf{x}) \in \mathbb{Z}$. Definition 5.7 formalizes this commitment scheme. By the injectivity of Enc_Q, this commitment scheme is computationally binding under the strong-RSA assumption.

Definition 5.7 (Non-Hiding Bounded Integer Vector Commitment Scheme)**.** The following setup algorithm and commitment function define a non-hiding bounded integer vector commitment scheme:

- $\mathsf{pk} = (\mathbb{H}, g) \leftarrow \textsc{Setup}(1^\lambda)$, where $(\mathbb{H}, \cdot) \leftarrow \mathcal{G}(1^\lambda)$ for a hidden-order group generator $\mathcal{G}(\cdot)$ and $g \leftarrow_R \mathbb{H}$ is sampled uniformly at random;
- $\textsc{com}_{\mathsf{pk}} \colon \mathbb{Z}(\alpha)^n \to \mathbb{H}, \quad \mathbf{x} \mapsto g^{\mathrm{Enc}_Q(\mathbf{x})}$, where $\mathrm{Enc}_Q(\mathbf{x}) = \sum_{i=1}^n x_i Q^{i-1}$ for some $Q \geq 2\alpha$.

Our goal is to construct an interactive proof for proving knowledge of an opening $\mathbf{x} \in \mathbb{Z}(\alpha)^n$ of the commitment $P \in \mathbb{H}$ satisfying the linear constraint $L(\mathbf{x}) = y$ for some linear form $L \colon \mathbb{Z}^n \to \mathbb{Z}$, i.e., for proving knowledge of a short Ψ_Q-preimage, where

$$\Psi_Q \colon \mathbb{Z}^n \to \mathbb{H} \times \mathbb{Z}, \quad \mathbf{x} \mapsto \left(g^{\mathrm{Enc}_Q(\mathbf{x})}, L(\mathbf{x})\right).$$

The interactive proofs of Section 3.3 have soundness slack τ and approximation factor ζ, i.e., they allow a prover to prove knowledge of a Ψ_Q-preimage $\mathbf{x}'$ of $(P^\zeta, \zeta y) \in \mathbb{H} \times \mathbb{Z}$ with $\|\mathbf{x}'\|_\infty \leq \tau\alpha$. Oftentimes, this relaxation is acceptable as long as the commitment scheme is also binding with respect to (τ, ζ)-*relaxed* openings. More precisely, given a commitment P, it should be hard for a prover to find distinct openings $\mathbf{x}, \mathbf{x}' \in \mathbb{Z}(\tau\alpha)^n$ of P^ζ. Concretely, this means that the encoding should be instantiated such that $Q \geq 2\tau\alpha$ instead of $Q \geq 2\alpha$. Hence, the soundness slack τ directly influences the efficiency of the interactive proof and should thus be kept to a minimum.

Further, the mapping Ψ_Q is a $\mathbb{Z}$-module homomorphism. For this reason, instantiating the compression mechanism of Section 3.3.2 directly, requires a challenge set $\mathcal{C} \subseteq \mathbb{Z}$. This either leaves us with a small challenge set, e.g., $\mathcal{C} = \{-1, 0, 1\}$, or with a large soundness slack τ. For instance, challenge sets of the form $\mathcal{C} = \mathbb{Z}(B) = \{x \in \mathbb{Z} : |x| < B\}$ result in a soundness slack that grows *exponentially* in B. For this reason, we first apply the base extension techniques of Section 3.3.4. More precisely, we extend the base $\mathbb{Z}$ of the $\mathbb{Z}$-module homomorphism Ψ_Q to the $2d$-th cyclotomic number ring $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$ for $d = 2^{d'}$ a power of two, i.e., we consider the $\mathcal{R}$-module homomorphism

$$\Psi_{Q,\mathcal{R}} \colon \mathcal{R} \otimes_{\mathbb{Z}} \mathbb{Z}^n \to \mathcal{R} \otimes_{\mathbb{Z}} (\mathbb{H} \times \mathbb{Z}), \quad \text{such that} \quad r \otimes \mathbf{x} \mapsto r \otimes \Psi_Q(\mathbf{x}).$$

Moreover, via the $\mathbb{Z}$-basis $\{1, \ldots, X^{d-1}\}$ of $\mathcal{R}$, we define the following ℓ_∞-norm on $\mathcal{R} \otimes_{\mathbb{Z}} \mathbb{Z} \cong \mathcal{R}$:

$$\left\|1 \otimes x_1 + X \otimes x_2 + \cdots + X^{d-1} \otimes x_d\right\|_\infty = \max_{1 \leq i \leq d} |x_i| \,,$$

where $x_i \in \mathbb{Z}$ for all i. This norm has a natural extension to $\mathcal{R} \otimes_{\mathbb{Z}} \mathbb{Z}^n$, i.e.,

$$\left\|1 \otimes \mathbf{x}_1 + X \otimes \mathbf{x}_2 + \cdots + X^{d-1} \otimes \mathbf{x}_d\right\|_\infty = \max_{1 \le i \le d} \|\mathbf{x}_i\|_\infty \,.$$

The reason for extending the base to a power-of-two cyclotomic number ring is that these rings contain challenge sets resulting in small soundness slack. To see this, we recall the following lemma by Benhamouda et al. [BCK+14].

Lemma 5.1 (Lemma 3.1 of [BCK+14])**.** *Let $d = 2^{d'} \in \mathbb{N}$ be a power of two and let $\mathbb{Z}[X]/(X^d+1)$ be the $2d$-th cyclotomic number ring. Then, for all $i \neq j$*

$$\frac{2}{X^i - X^j} \in \mathbb{Z}[X]/(X^d+1)\,.$$

Moreover, this polynomial only has coefficients in $\{-1, 0, 1\}$.

Proof. Without loss of generality, we may assume that $i = 0$ and $1 < j < 2d$. Now let k be the smallest positive integer such that $kj = 0 \mod d$. Since d is a power of two and $j \neq 0 \mod 2d$, it holds that $kj \neq 0 \mod 2d$ and $X^{kj} = -1$. Therefore,

$$\frac{2}{X^i - X^j} = \frac{2}{1 - X^j} = \frac{2}{1 - X^{kj}} \cdot (1 + X^j + X^{2j} + \cdots X^{(k-1)j}) = 1 + X^j + \cdots X^{(k-1)j}\,,$$

which proves the first claim of the lemma.

What remains to show is that no two exponents ℓj and $\ell' j$, for $0 \le \ell < \ell' < k$, are the same modulo d. Assuming the contrary, it follows that $(\ell' - \ell)j = 0 \mod d$ with $0 < \ell' - \ell < k$. This contradicts the assumption that k is the smallest positive integer such that $kj = 0 \mod d$ and completes the proof. □

Lemma 5.1 shows that the challenge set $\mathcal{C} = \{0, \pm 1, \pm X, \ldots, \pm X^{d-1}\}$ is a 2-exceptional subset of the power-of-two cyclotomic number ring $\mathcal{R} = \mathbb{Z}[X]/(X^d+1)$. Moreover, it immediately implies the following corollary.

Corollary 5.1. *Let $d = 2^{d'} \in \mathbb{N}$ be a power of two and let $\mathcal{R} = \mathbb{Z}[X]/(X^d+1)$ be the $2d$-th cyclotomic number ring. Further, let $\mathcal{C} = \{0, \pm 1, \pm X, \ldots, \pm X^{d-1}\} \subset \mathcal{R}$. Then,*

$$w(\mathcal{C}) = \max_{c \in \mathcal{C}, x \in \mathcal{R} \otimes_{\mathbb{Z}} \mathbb{Z} \setminus \{0\}} \frac{\|cx\|_\infty}{\|x\|_\infty} = 1\,,$$

$$\overline{w}(\mathcal{C}, 2) = \max_{c \neq c' \in \mathcal{C}, x \in \mathcal{R} \otimes_{\mathbb{Z}} \mathbb{Z} \setminus \{0\}} \frac{\left\|2(c - c')^{-1} x\right\|_\infty}{\|x\|_\infty} = d\,.$$

The following theorem summarizes the properties of the compression mechanism of Section 3.3.2 instantiated for the homomorphism

$$\Psi_{Q,\mathcal{R}} \colon \mathcal{R} \otimes_{\mathbb{Z}} \mathbb{Z}^n \to \mathcal{R} \otimes_{\mathbb{Z}} (\mathbb{H} \times \mathbb{Z})\,, \quad \text{such that} \quad r \otimes \mathbf{x} \mapsto r \otimes \Psi_Q(\mathbf{x})\,,$$

with challenge set $\mathcal{C} = \{0, \pm 1, \pm X, \ldots, \pm X^{d-1}\} \subset \mathcal{R} = \mathbb{Z}[X]/(X^d+1)$. The theorem is a direct consequence of Theorem 3.8 and Corollary 5.1. It is valid for

all values of $Q \in \mathbb{N}$. However, the compression mechanism has soundness slack $12d^3$ and approximation factor 8. More precisely, while the prover claims to know a Ψ_Q-preimage of (P, y) with ℓ_∞-norm at most α, it is only capable of proving knowledge of a Ψ_Q-preimage of $(P^8, 8 \cdot y)$ with ℓ_∞-norm at most $12d^3\alpha$. Therefore, the vector commitment scheme should be instantiated with $Q \geq 24d^3\alpha$.

Theorem 5.3 (Strong-RSA Based Compression Mechanism)**.** *Let $\alpha, Q \in \mathbb{N}$, $n \in \mathbb{N}$ even, $d \in \mathbb{N}$ a power of two and $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$. Then the compression mechanism Π_c, described in Protocol 7, instantiated for the base-$\mathcal{R}$ extension $\Psi_{Q,\mathcal{R}}$ of the strong-RSA homomorphism*

$$\Psi_Q \colon \mathbb{Z}^n \to \mathbb{H} \times \mathbb{Z}, \quad \mathbf{x} \mapsto \left(g^{\mathrm{Enc}_Q(\mathbf{x})}, L(\mathbf{x})\right),$$

with challenge set $\mathcal{C} = \{0, \pm 1, \pm X, \ldots, \pm X^{d-1}\}$, is an interactive proof for relation

$$\mathfrak{R}_{\mathsf{RSA}} = \left\{(P, y, \alpha; \mathbf{x}) : \Psi_{Q,\mathcal{R}}(\mathbf{x}) = (P, y) \wedge \|\mathbf{x}\|_\infty \leq \alpha\right\}.$$

It is perfectly complete and 3-out-of-$(2d + 1)$ special-sound with soundness slack $12d^3$ and approximation factor 8. Moreover, the communication costs are:

- *$\mathcal{P} \to \mathcal{V}$: $\frac{dn}{2} + 2d$ elements of $\mathbb{Z}$ and $2d$ elements of $\mathbb{H}$;*
- *$\mathcal{V} \to \mathcal{P}$: 1 element of $\mathcal{C} \subseteq \mathcal{R}$.*

The following theorem now summarizes the properties of the μ-fold recursive composition of the strong-RSA based interactive proof of Theorem 5.3. As before this theorem holds for any Q, but to account for the soundness slack it should be instantiated with $Q \geq 2 \cdot 12^\mu \cdot d^{3\mu} \cdot \alpha$.

Theorem 5.4 (Recursive Strong-RSA Based Compression Mechanism)**.** *Let $n = 2^\mu \in \mathbb{N}$ be a power of two. Then, the μ-fold recursive composition of the strong-RSA compression mechanism of Theorem 5.3 is a $(2\mu+1)$-round interactive proof for relation $\mathfrak{R}_{\mathsf{RSA}}$. It is perfectly complete and $(3, \ldots, 3)$-out-of-$(2d+1, \ldots, 2d+1)$ special-sound with soundness slack $12^\mu \cdot d^{3\mu}$ and approximation factor 8^μ. Moreover, the communication costs are:*

- *$\mathcal{P} \to \mathcal{V}$: $d + 2d\log_2 n$ elements of $\mathbb{Z}$ and $2d\log_2 n$ elements of $\mathbb{H}$;*
- *$\mathcal{V} \to \mathcal{P}$: μ element of $\mathcal{C} = \{0, \pm 1, \pm X, \ldots, \pm X^{d-1}\} \subset \mathcal{R}$.*

If the interactive proof of Theorem 5.4 is instantiated with the degree d of the base extension equal to $\log_2 n$, its knowledge error is constant in n (see Section 3.3.4). Therefore, to reduce the knowledge error down to $2^{-\lambda}$, $t = \mathcal{O}(\lambda)$ parallel repetitions are required. The communication complexity of the resulting protocol, measured in the number of elements, is $\mathcal{O}(\lambda \cdot \log_2^2 n)$, i.e., it is polylogarithmic in n. Moreover, the soundness slack equals

$$12^\mu d^{3\mu} = n^{\log_2(\log_2 n)+2+\log_2 3},$$

i.e., it is subexponential in n. Taking $\alpha = (q-1)/2$ and $L \colon \mathbb{Z}^n \to \mathbb{Z}_q$ for some odd prime q, shows that this protocol allows a prover to commit to a vector $\mathbf{x} \in \mathbb{Z}_q^n$ and proves that it satisfies an arbitrary $\mathbb{Z}_q$-linear constraint.

An advantage of this strong-RSA based interactive proof, over the discrete logarithm instantiation of Section 5.2, is that the public key size of the underlying commitment scheme is constant in n. By contrast, a Pedersen commitment to an n-dimensional vector requires $n+1$ group elements, i.e., there the public key size is linear in n. However, note that this improvement comes at the cost of increasing the communication complexity from logarithmic to polylogarithmic.

Our approach differs from the polynomial commitment schemes of [BFS20] and [BHR+21]. Restricting to polynomial commitment schemes allows for an adaptation that reduces the verification complexity, measured in the number of group exponentiations, from quasilinear down to polylogarithmic in n. However, this adaptation requires the use of *proofs of exponentiation* [Wes19]. Moreover, our instantiation is unconditionally sound, whereas the aforementioned polynomial commitment schemes have conditional soundness based on the strong-RSA assumption.

5.6 A Lattice Assumption: Short Integer Solutions

The final compressed Σ-protocol instantiation that we shall discuss is based on a lattice assumption and therefore plausibly secure against quantum adversaries. More precisely, its security is based on the hardness of the *Module Short Integer Solution* (MSIS) problem (Definition 2.20). As before, our goal is to construct an efficient protocol for opening linear forms on compactly committed vectors.

Before we describe the underlying MSIS-based commitment scheme, we introduce some notation. Let $\mathcal{R} = \mathbb{Z}[X]/f(X)$ for a monic and irreducible polynomial $f(x) \in \mathbb{Z}[X]$ of degree d. For any $p \in \mathbb{Z}$, we write $\mathcal{R}_p = \mathcal{R}/p\mathcal{R}$. Moreover, we equip $\mathcal{R}$ with the following ℓ_∞-norm:

$$\left\| \sum_{i=0}^{d-1} a_i X^i \right\|_\infty = \max_{0 \le i \le d-1} |a_i| \,, \quad \text{for all } \sum_{i=0}^{d-1} a_i X^i \in \mathcal{R} \,.$$

This norm has a natural extension to $\mathcal{R}^n$, i.e., $\|\mathbf{x}\|_\infty = \max_{1 \le i \le n} \|x_i\|_\infty$ for all vectors $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{R}^n$. Further, for $\alpha \in \mathbb{R}_{\ge 0}$, we write

$$\mathcal{R}(\alpha) = \{x \in \mathcal{R} : \|x\|_\infty \le \alpha\} \,.$$

The MSIS-based commitment scheme, described in Definition 5.8, allows a prover to commit to vectors $\mathbf{x} \in \mathcal{R}^n$ of bounded ℓ_∞-norm, i.e., $\mathbf{x} \in \mathcal{R}(\alpha)^n$ for some $\alpha \in \mathbb{R}_{\ge 0}$. It is based on Ajtai's seminal work [Ajt96] and different variants of this commitment scheme have been presented in prior works, e.g., in [BKL+15; BBC+18; BDL+18]. This commitment scheme is oftentimes instantiated with norm bound $\alpha = \lceil (p-1)/2 \rceil$ for some $p \in \mathbb{N}$. This instantiation allows a prover to commit to vectors in $\mathcal{R}_p^n$.

Definition 5.8 (Lattice-Based Commitment Scheme)**.** Let $\mathcal{R} = \mathbb{Z}[X]/f(X)$ for a monic and irreducible polynomial $f(x) \in \mathbb{Z}[X]$ of degree d and let $\alpha \in \mathbb{R}_{\ge 0}$. Then, the following setup algorithm and commitment function define a lattice-based vector commitment scheme:

- $\mathsf{pk} = (A_1, A_2) \leftarrow \textsc{Setup}(1^\lambda, \mathcal{R}, q, k, r, \alpha, n)$, where $q > 2\alpha$ is a rational prime, $k, r \in \mathbb{N}$ and $(A_1, A_2) \leftarrow_R \mathcal{R}_q^{k\times(n+r)}$ is sampled uniformly at random;
- $\textsc{com}_{\mathsf{pk}} \colon \mathcal{R}(\alpha)^n \times \mathcal{R}(\alpha)^r \to \mathcal{R}_q^k, \quad (\mathbf{x}; \gamma) \mapsto A_1\mathbf{x} + A_2\gamma \mod q.$

When considered as a function on $\mathcal{R}^n \times \mathcal{R}^k$, the commitment function $\textsc{com}_{\mathsf{pk}}$ is an $\mathcal{R}$-module *homomorphism*. Moreover, the following lemma shows that the commitment scheme is computationally *binding* under the MSIS assumption. Note that, for large enough $n + r$, the hardness of the $\mathrm{MSIS}^\infty_{k,n+r,2\alpha}$ problem is independent of $n + r$ (see Equation 2.2). Therefore, this vector commitment scheme is *compact*, i.e., the size of a commitment is constant in the input dimension n.

Lemma 5.2 (Binding)**.** *The commitment scheme of Definition 5.8 is binding, conditioned on the hardness of the* $\mathrm{MSIS}^\infty_{q,k,n+r,2\alpha}$*-problem over* $\mathcal{R}$*.*

Proof. Suppose that $(\mathbf{x}; \gamma) \neq (\mathbf{x}'; \gamma')$ are two distinct openings of the same commitment P. Then $\mathbf{s} = (\mathbf{x} - \mathbf{x}'; \gamma - \gamma') \neq 0$ satisfies $\|\mathbf{s}\|_\infty \leq 2\alpha$ and $[A_1, A_2]\mathbf{s} = 0$, i.e., $\mathbf{s}$ is a solution of the $\mathrm{MSIS}^\infty_{k,n+r,2\alpha}$ problem, which completes the proof. □

The following lemma shows that if q is chosen to be inert in $\mathcal{R}$, i.e., if $\mathcal{R}_q$ is a field, and the randomness dimension r is large enough, then the commitment scheme is statistically *hiding*. The assumption that q is inert in $\mathcal{R}$ is only made to simplify the exposition. In this case, $\mathcal{R}_q$ is a field and it is easily seen that

$$\Pr\big(A\mathbf{x} = A\mathbf{y} : A \leftarrow_R \mathcal{R}_q^{k\times r}\big) \leq \frac{1}{\big|\mathcal{R}_q^k\big|} = \frac{1}{q^{dk}} \quad \forall \mathbf{x} \neq \mathbf{y} \in \mathcal{R}_q^r\,,$$

i.e., the family of hash functions $h_A \colon \mathcal{R}_q^r \to \mathcal{R}_q^k,\ \mathbf{x} \mapsto A\mathbf{x}$ is universal. By contrast, if $\mathcal{R}_q$ is not a field and contains zero-divisors, this family of hash functions is not universal. Based on [LS18], Baum et al. [BDL+18] show how this lemma can be generalized to arbitrary (not necessarily inert) primes q. The results of [LS18], and thus the generalization of [BDL+18], are only applicable to cyclotomic number rings $\mathcal{R}$. Fortunately, the generalization [ACX21] of [LS18] allows one to handle arbitrary number rings $\mathcal{R} = \mathbb{Z}[X]/f(X)$.

Lemma 5.3 (Hiding)**.** *Let* $\mathcal{R} = \mathbb{Z}[X]/f(X)$ *for a monic and irreducible polynomial* $f(x) \in \mathbb{Z}[X]$ *of degree* $d \in \mathbb{N}$*, and let* λ *denote the security parameter. If* q *is inert in* $\mathcal{R}$ *and* $r \in \mathbb{N}$ *is such that*

$$r \geq \frac{2\lambda + dk\log_2 q}{d\log_2(2\alpha + 1)}\,,$$

then the commitment scheme of Definition 5.8 is statistically hiding.

Proof. Since q is inert in $\mathcal{R}$, it follows that $\mathcal{R}_q$ is a field and the family of functions $h_A \colon \mathcal{R}_q^r \to \mathcal{R}_q^k,\ \mathbf{x} \mapsto A\mathbf{x}$, indexed by $A \in \mathcal{R}_q^{k\times r}$, is a universal hash family. Further, the min-entropy of the uniform distribution over $\mathcal{R}(\alpha)^r$ equals

$$dr\log_2(2\alpha + 1) \geq 2\lambda + dk\log_2 q\,.$$

Since $q > 2\alpha$ and by the leftover hash lemma [ILL89], it therefore follows that the statistical distance between the distribution

$$\mathcal{X} = \{(A, A\gamma) : A \leftarrow_R \mathcal{R}_q^{k\times r}, \gamma \leftarrow_R \mathcal{R}(\alpha)^r\}$$

and the uniform distribution $\mathcal{U}$ over $\mathcal{R}_q^{k\times r} \times \mathcal{R}_q^k$ is at most $2^{-\lambda}$, which proves the lemma. □

As in the strong-RSA instantiation, due to the soundness slack and approximation factor, our compressed Σ-protocols only allow a prover to prove knowledge of a *relaxed* opening. The following definition formalizes the notion of a relaxed commitment opening for the lattice-based commitment scheme of Definition 5.8.

Definition 5.9 ((τ, ζ)-Relaxed Commitment Opening)**.** Let $\tau \in \mathbb{R}_{\geq 0}$, $\zeta \in \mathcal{R}$ and let P be a commitment for the commitment scheme of Definition 5.8. A (τ, ζ)-relaxed opening of P is a pair $(\mathbf{x}; \gamma) \in \mathcal{R}^{n+r}$, such that $\text{COM}(\mathbf{x}; \gamma) = \zeta \cdot P \in \mathcal{R}_q^k$ and $\|(\mathbf{x}; \gamma)\|_\infty \leq \tau\alpha$.

A (τ, ζ)-relaxed opening of a commitment P differs in two ways from a standard opening. First, it contains an approximation factor ζ, such that the relaxed opening gives a short preimage for $\zeta \cdot P \in \mathcal{R}_q^k$ instead of P. Second, the norm-bound $\tau\alpha$ of relaxed openings differs from the norm bound α on honestly committed vectors (typically $\tau > 1$).

As long as it is infeasible to find two distinct (τ, ζ)-relaxed openings $(\mathbf{x}; \gamma)$ and $(\mathbf{x}'; \gamma')$ of a commitment P with $(\mathbf{x}; \gamma) \neq (\mathbf{x}'; \gamma')$, proving knowledge of relaxed opening is sufficient in most practical scenarios. In this case, we say the commitment scheme is binding with respect to (τ, ζ)-relaxed openings. The following lemma reduces breaking the "binding with respect to relaxed openings" property to solving the MSIS-problem. Note that the hardness of the corresponding MSIS-problem does not depend on the approximation factor ζ.

Lemma 5.4 (Binding with respect to (τ, ζ)-Relaxed Openings)**.** *Let $\tau \in \mathbb{R}_{\geq 0}$ and $\zeta \in \mathcal{R}$. The commitment scheme of Definition 5.8 is binding with respect to (τ, ζ)-relaxed openings, conditioned on the hardness of the* $\text{MSIS}^\infty_{q,k,n+r,2\tau\alpha}$*-problem over $\mathcal{R}$.*

Proof. Suppose that $(\mathbf{x}; \gamma)$ and $(\mathbf{x}'; \gamma')$ are distinct (τ, ζ)-relaxed openings of a commitment P. Then $\mathbf{s} = (\mathbf{x} - \mathbf{x}'; \gamma - \gamma') \neq 0$ satisfies $\|\mathbf{s}\|_\infty \leq 2\tau\alpha$ and $[A_1, A_2]\mathbf{s} = 0$, i.e., $\mathbf{s}$ is a solution of the $\text{MSIS}^\infty_{k,n+r,2\tau\alpha}$ problem, which completes the proof. □

Our goal is to prove knowledge of a (τ, ζ)-relaxed commitment opening $(\mathbf{x}; \gamma)$, for appropriate $\tau \in \mathbb{R}_{\geq 0}$ and $\zeta \in \mathcal{R}$, that satisfies the constraint $L(\mathbf{x}) = \zeta \cdot y$, where $L\colon \mathcal{R}^n \to \mathcal{R}'$ is an $\mathcal{R}$-module homomorphism for some arbitrary $\mathcal{R}'$. To this end, we consider the following $\mathcal{R}$-module homomorphism:

$$\Psi\colon \mathcal{R}^n \times \mathcal{R}^r \to \mathcal{R}_q^k \times \mathcal{R}', \quad (\mathbf{x}; \gamma) \mapsto \left(A_1\mathbf{x} + A_2\gamma, L(\mathbf{x})\right).$$

Typically, $\mathcal{R}' = \mathcal{R}_p$ for some rational prime $p \neq q$. Note that, if the approximation factor ζ is invertible in $\mathcal{R}'$, then $L(\mathbf{x}) = \zeta \cdot y$ implies that $L(\zeta^{-1} \cdot \mathbf{x}) = y$.

For this reason, in most practical scenarios, the approximation factor is required to be invertible in $\mathcal{R}'$.

The Ψ-instantiation of the compressed Σ-protocol of Section 3.3, with some challenge set $\mathcal{C} \subseteq \mathcal{R}$, requires rejection sampling in order to be special honest-verifier zero-knowledge (SHVZK). More precisely, it requires a distribution-algorithm pair $(\mathcal{D}, \mathcal{F})$ that is V-hiding, for $V = \{c\mathbf{x} : \mathbf{x} \in \mathcal{R}(\alpha)^{n+r} \wedge c \in \mathcal{C}\}$, and β-bounded for some reasonably small $\beta \in \mathbb{R}_{\geq 0}$ (Definition 3.2). In our instantiation, we let $\mathcal{D}$ be the uniform distribution over an appropriate subset of $\mathcal{R}^{n+r}$. The following lemma shows that this approach gives the required properties.

Lemma 5.5 (Uniform Rejection Sampling)**.** *Let $\mathcal{R} = \mathbb{Z}[X]/f(X)$ for a monic and irreducible polynomial $f(X) \in \mathbb{Z}[X]$ of degree d, $\mathcal{C} \subseteq \mathcal{R}$ and $n, r \in \mathbb{N}$. Recall that $\mathcal{R}(\alpha) = \{\mathbf{x} \in \mathcal{R} : \|\mathbf{x}\|_\infty \leq \alpha\}$ and*

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

Further, let $V = \{c\mathbf{x} \in \mathcal{R}^{n+r} : \mathbf{x} \in \mathcal{R}(\alpha)^{n+r} \wedge c \in \mathcal{C}\}$, $\gamma > w(\mathcal{C})\alpha$, $\mathcal{D}$ the uniform distribution over $\mathcal{R}(\gamma)^{n+r}$ and

$$\mathcal{F}(\mathbf{r}, \mathbf{v}) = \begin{cases} \perp, & \text{if } \|\mathbf{v} + \mathbf{r}\|_\infty > \gamma - w(\mathcal{C})\alpha, \\ \mathbf{v} + \mathbf{r}, & \text{otherwise.} \end{cases}$$

Then $(\mathcal{D}, \mathcal{F})$ is perfectly V-hiding and $(\gamma - w(\mathcal{C})\alpha)$-bounded, with abort probability

$$\delta \leq (n + r)d \cdot \frac{2w(\mathcal{C})\alpha + 2}{2\gamma + 1} .$$

Proof. Note that, for all $\mathbf{v} \in V$, it holds that $\|\mathbf{v}\|_\infty \leq w(\mathcal{C})\alpha$. Hence, the abort probability of the probabilistic algorithm $\{\mathcal{F}(\mathbf{r}, \mathbf{v}) \mid \mathbf{r} \leftarrow \mathcal{D}\}$ equals

$$\begin{aligned} \delta &= 1 - \left(\frac{2\lfloor \gamma - w(\mathcal{C})\alpha \rfloor + 1}{2\lfloor \gamma \rfloor + 1} \right)^{(n+r)d} \\ &\leq 1 - \left(1 - \frac{2w(\mathcal{C})\alpha + 2}{2\gamma + 1} \right)^{(n+r)d} \\ &\leq (n + r)d \cdot \frac{2w(\mathcal{C})\alpha + 2}{2\gamma + 1} . \end{aligned}$$

where the final step follows from Bernoulli's inequality.

Now let $\mathcal{F}'$ be the algorithm that aborts with probability δ and otherwise outputs $\mathbf{z} \leftarrow_R \mathcal{R}\big(\gamma - w(\mathcal{C})\alpha\big)^{n+r}$ sampled uniformly at random. Then it is easily seen that $\{\mathcal{F}(\mathbf{r}, \mathbf{v}) \mid \mathbf{r} \leftarrow_R \mathcal{D}\}$ and $\{\mathcal{F}'\}$ have exactly the same output distributions, i.e., $(\mathcal{D}, \mathcal{F})$ is perfectly V-hiding.

Finally, $(\mathcal{D}, \mathcal{F})$ is $(\gamma - w(\mathcal{C})\alpha)$-bounded, which completes the proof. □

Remark 5.1. The smallest lattice-based signatures actually take $\mathcal{D}$ to be a Gaussian distribution. Namely, when the secrets have a bounded ℓ_2-norm, the Gaussian

distribution results in better protocol parameters. In our instantiation, this is not the case; our secrets are bounded with respect to the ℓ_∞-norm. For this reason, it is beneficial to resort to a uniform distribution over an appropriate subset of $\mathcal{R}^{n+r}$. An additional benefit is that uniform sampling is less prone to side-channel attacks. This is the reason that the lattice-based digital signature scheme Dilithium also deploys a uniform rejection sampling approach [DKL+18].

Let now Π_{MSIS} be the compressed Σ-protocol of Section 3.3.3 instantiated for homomorphism Ψ, with the rejection sampling approach of Lemma 5.5 and some arbitrary ζ-exceptional challenge set $\mathcal{C} \subseteq \mathcal{R}$. The properties of Π_{MSIS}, summarized in the following theorem, follow immediately from Theorem 3.9 and Lemma 5.5.

Theorem 5.5 (MSIS-based Compressed Σ-Protocol)**.** *Let $n + r = 2^\mu$ for some $\mu \in \mathbb{N}$. Let Π_{MSIS} be the compressed Σ-protocol Π_{comp}, described in Protocol 8, instantiated with a ζ-exceptional challenge set $\mathcal{C} \subseteq \mathcal{R}$ of cardinality at least 3, the distribution-algorithm pair $(\mathcal{D}, \mathcal{F})$ of Lemma 5.5 and MSIS-based homomorphism*

$$\Psi \colon \mathcal{R}^n \times \mathcal{R}^r \to \mathcal{R}_q^k \times \mathcal{R}', \quad (\mathbf{x}; \gamma) \mapsto \big(A_1\mathbf{x} + A_2\gamma, L(\mathbf{x})\big)\,.$$

Then Π_{MSIS} is an interactive proof for relation

$$\mathfrak{R}_{\mathrm{MSIS}} = \{(P, y, \alpha; \mathbf{x}) : \Psi(\mathbf{x}) = (P, y) \wedge \|\mathbf{x}\|_\infty \leq \alpha\}\,.$$

It is complete with completeness error

$$\delta \leq (n + r)d \cdot \frac{2w(\mathcal{C})\alpha + 2}{2\gamma + 1}\,,$$

$(2, 3, \ldots, 3)$-out-of-$(|\mathcal{C}|, \ldots, |\mathcal{C}|)$ special-sound with soundness slack

$$\tau = 2 \cdot 6^\mu \cdot \overline{w}(\mathcal{C}, \zeta)^{3\mu+1} \cdot \big(w(\mathcal{C})^2 + w(\mathcal{C})^3\big)^\mu \cdot w(\mathcal{C}) \cdot \frac{\gamma - w(\mathcal{C})\alpha}{\alpha}$$

and approximation factor $\zeta^{3\mu+1}$, and it is non-abort special honest-verifier zero-knowledge (NA-SHVZK).

Moreover, it has $2\mu+3$ communication rounds and the communication costs are:

- *$\mathcal{P} \to \mathcal{V}$: 1 element of $\mathcal{R}$ with norm at most $\big(1+w(\mathcal{C})\big)^\mu \cdot \big(\gamma - w(\mathcal{C})\alpha\big)$, $2\mu+1$ elements of $\mathcal{R}'$ and $2\mu+1$ elements of $\mathcal{R}_q^k$;*
- *$\mathcal{V} \to \mathcal{P}$: $\mu + 1$ elements of $\mathcal{C} \subseteq \mathcal{R}$.*

As a concrete example of the compressed Σ-protocol Π_{MSIS}, let us consider the cyclotomic number ring $\mathcal{R} = \mathbb{Z}[X]/(X^d + 1)$ with $d = 2^{d'}$ a power-of-two and challenge set $\mathcal{C} = \{0, \pm 1, \pm X, \ldots, \pm X^{d-1}\} \subset \mathcal{R}$. Further, by taking norm-bound $\alpha = (p - 1)/2$ for some odd prime p and $\mathcal{R}' = \mathcal{R}_p$, we consider a prover that wishes to commit to vectors $\mathbf{x} \in \mathcal{R}_p^n$ and prove that $L(\mathbf{x}) = y$ for some linear form $L \colon \mathcal{R}_p^n \to \mathcal{R}_p$. In Section 5.5, we used exactly the same ring for the base-extension of our strong-RSA instantiation. Moreover, in Lemma 5.1 and Corollary 5.1, we showed that $\mathcal{C}$ is a 2-exceptional subset with $w(\mathcal{C}) = 1$ and $\overline{w}(\mathcal{C}, 2) = d$. Note that, since p is odd, the approximation factor $\zeta = 2$ is invertible in $\mathcal{R}_p$. Let us now analyze the communication complexity of this example.

This instantiation of Π_{MSIS} is $(2,3,\dots,3)$-out-of-$(2d+1,\dots,2d+1)$ special-sound. In Chapter 6, we will see that it therefore has knowledge error

$$1-\left(1-\frac{1}{2d+1}\right)\left(1-\frac{2}{2d+1}\right)^{\mu}\leq 1-\left(1-\frac{2}{2d+1}\right)^{\mu+1}\leq\frac{2\mu+2}{2d+1}\,,$$

where $\mu=\log_2(n+r)$. For simplicity, let us assume that $d\geq 2\mu+2$. Then, this compressed Σ-protocol has knowledge error at most $1/2$, and $t\leq\lambda$ parallel repetitions are required to reduce the knowledge error down to $2^{-\lambda}$. If $d<2\log_2(n+r)+2$, the base extension techniques of Section 3.3.4 can be deployed to increase the size of the challenge set.

Further, we let $\gamma=\Theta((n+r)td\alpha)=\Theta((n+r)tdp)$. By Theorem 5.5, this is enough to achieve a constant completeness error. Altogether, this instantiation allows a prover to prove knowledge of $(\tau,2^{3\mu+1})$-relaxed commitment openings, where

$$\tau=2d\cdot(12d^3)^{\mu}\cdot\frac{\gamma-\alpha}{\alpha}=\Theta\big(t\cdot d^2\cdot(n+r)^{3+\log_2 3+3\log_2 d}\big)\,.$$

Hence, in practice, the commitment scheme must be instantiated to be binding with respect to $(\tau,2^{3\mu+1})$-relaxed openings, i.e., the $\mathrm{MSIS}^{\infty}_{q,k,n+r,2\tau\alpha}$-problem over $\mathcal{R}$ must be computationally hard (Lemma 5.4). From the Micciancio-Regev bound (Equation 2.2) it follows that this problem is hard if

$$dk\log_2 q\geq\frac{\log_2^2(2\tau\alpha\sqrt{n+r})}{4\log_2\delta}=\Theta\left(\frac{\log^2 d\cdot\log^2\big(tdp\cdot(n+r)\big)}{\log\delta}\right),\tag{5.2}$$

where δ is the root Hermite factor.

By Theorem 5.5 and the fact that $t=\mathcal{O}(\lambda)$, it therefore follows that the resulting t-fold parallel repetition of Π_{MSIS} has communication complexity

$$\mathcal{O}\left(\frac{\lambda\cdot\log(n+r)\cdot\log^2 d\cdot\log^2\big(\lambda dp\cdot(n+r)\big)}{\log\delta}\right).$$

Finally, by Lemma 5.3 and Equation 5.2, we observe that $r=\mathcal{O}(\lambda+\frac{\log\lambda pn}{\log\delta})$. Hence, the resulting protocol has *polylogarithmic* communication complexity.

Note that, in the discrete logarithm instantiation over the group $\mathbb{G}$, the secret vector $\mathbf{x}$ has coefficients in the finite field $\mathbb{Z}_q$, where q is the exponent of $\mathbb{G}$. For the discrete logarithm problem to be hard in $\mathbb{G}$, the size of the prime q must therefore be exponential in the security parameter. The discrete logarithm instantiation does not allow a prover to directly prove relations over fields $\mathbb{Z}_p$ of small characteristic p. By contrast, the above lattice-based instantiation does not suffer from this limitation. In fact, smaller primes p correspond to harder MSIS-problem instantiations.

Remark 5.2. For simplicity, we have deployed a standard parallel repetition approach to reduce the knowledge error down to $2^{-\lambda}$. More precisely, in the considered t-fold parallel repetition, the verifier only accepts if the prover succeeds in all t parallel instances. However, while decreasing the knowledge error, this approach also increases the completeness error. To account for this effect, we have

chosen the protocol parameter γ to increase linearly in the number of parallel repetitions t. In Section 6.5.4, we describe a threshold parallel repetition approach that decreases both the completeness and knowledge error simultaneously. This approach would therefore allow for a further improvement of the above lattice instantiation.

CHAPTER 6

6

Knowledge Soundness of Compressed Σ-Protocols

6.1 Introduction

In a compressed Σ-protocol for relation $\mathfrak{R}$ a prover aims to convince a verifier to know a witness $w \in \mathfrak{R}$ for some statement $x \in \{0,1\}^*$. A dishonest prover, without knowledge of a witness w, should not be able to convince a verifier. This property is called *knowledge soundness* and is formally captured by Definition 2.27. Knowledge soundness requires the existence of an extraction algorithm, called a knowledge extractor that, on input x and given oracle access to a prover $\mathcal{P}^*$, aims to output a witness $w \in \mathfrak{R}$.

Thus far, we have only shown compressed Σ-protocols to satisfy the weaker notion *special-soundness*. It is well known that 3-round k-out-of-N special-sound interactive proofs are knowledge sound with knowledge error $(k-1)/N$, i.e., for 3-round interactive proofs, special-soundness implies knowledge soundness. Further, the t-fold parallel repetition of a 3-round 2-out-of-N special-sound interactive proof is easily seen to decrease the knowledge error from $1/N$ down to $1/N^t$. Finally, the security loss of the Fiat-Shamir transformation of a 3-round interactive proof is known to be linear in the number of random oracle queries admitted to a prover attacking the considered non-interactive proof. However, for multi-round interactive proofs, the situation is significantly more complicated. In this chapter, we discuss knowledge soundness of certain (natural variations of) multi-round special-sound interactive proofs.

In Section 6.2, we explain the difficulties that arise when generalizing existing knowledge extractors for 3-round interactive proofs to $(2\mu+1)$-round interactive proofs.

In Section 6.3, we describe an extraction algorithm for special-sound multi-round interactive proofs that runs in *strict* polynomial time. The success probability of this extractor is not large enough to prove knowledge soundness; it only shows that a subclass of special-sound interactive proofs satisfies an alternative notion of knowledge soundness (Definition 2.31). It is not known how to increase the success probability of the extractor in strict polynomial time. In fact, unless one allows for smaller success probability, strict polynomial time extraction is impossible for nontrivial constant-round zero-knowledge proofs [BL02]. This section is based on

the article [AC20], co-authored by Ronald Cramer.

In Section 6.4, we show that the success probability of the extraction algorithm can be increased if it is allowed to run in *expected* polynomial time. The resulting knowledge extractor shows that, also for multi-round interactive proofs, special-soundness *tightly* implies knowledge soundness. This section is based on the article [ACK21], co-authored by Ronald Cramer and Lisa Kohl.

In Section 6.5, we consider the t-fold parallel repetition of multi-round special-sound interactive proofs. In many occasions, the knowledge error κ of an interactive proof is not small enough and parallel repetition is used to decrease it. We show that, also for multi-round protocols, the t-fold parallel repetition of a special-sound interactive proof reduces the knowledge error from κ down to κ^t. This section is based on the article [AF22], co-authored by Serge Fehr.

In Section 6.6, we show that the security loss of the Fiat-Shamir transformation of a special-sound interactive proof is independent of the number of rounds. More precisely, we show that, similar to the 3-round case, the security loss is linear in the number of random oracle queries admitted to the prover $\mathcal{P}^*$ attacking the considered non-interactive protocol. This section is based on the article [AFK22], co-authored by Serge Fehr and Michael Klooß.

Table 6.1 summarizes the main properties, i.e., the efficiency and success probability, of the different knowledge extractors described in this chapter.

Protocol	Section	Number of $\mathcal{P}^*$-queries X	Success probability P
Π	6.3	$X \leq K$	$P \geq (\epsilon - \kappa)^K$
Π	6.4	$\mathbb{E}[X] \leq K$	$P \geq \epsilon - \kappa$
Π^t	6.5	$\mathbb{E}[X] \leq t \cdot 2^\mu \cdot K \leq t \cdot K^2$	$P \geq \frac{1}{2K}(\epsilon - \kappa^t)$
$\mathsf{FS}[\Pi]$	6.6	$\mathbb{E}[X] \leq K + (K-1)Q$	$P \geq \epsilon - (Q+1)\kappa$

Table 6.1: The efficiency and success probability of different knowledge extractors for (variations of) a $(k_1, \ldots, k_\mu)$-special-sound interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$. Here, Π^t and $\mathsf{FS}[\Pi]$ denote the t-fold parallel repetition and the Fiat-Shamir transformation of Π. Moreover, $\epsilon = \epsilon(x, \mathcal{P}^*)$ is the success probability of the prover $\mathcal{P}^*$ attacking the considered protocol on statement x and κ is the knowledge error of Π. Finally, Q is the number of random-oracle queries admitted to a non-interactive prover attacking $\mathsf{FS}[\Pi]$ and $K = \prod_{i=1}^{\mu} k_i$.

6.2 The Knowledge Soundness Problem for Multi-Round Special-Sound Interactive Proofs

A 3-round public-coin interactive proof is said to be 2-special-sound if there exists an efficient algorithm that, on input a *colliding* pair of accepting transcripts (a, c, z) and (a, c', z'), i.e., with common first message a and distinct challenges $c \neq c'$,

outputs a witness $w \in \mathfrak{R}(x)$ for the statement x. By contrast, knowledge soundness requires the existence of an extractor that is given oracle access to a prover; it should extract a witness by interacting with a prover in a black-box manner. In particular, a knowledge extractor does not receive protocol transcripts as input. It should either generate these transcripts or extract a witness by some other means. In order to prove that 2-special-soundness implies knowledge soundness, one must show how to *efficiently* output a colliding pair of accepting transcripts given only oracle access to a prover. By special-soundness a witness can then be extracted efficiently from this pair of transcripts. Together these two steps define a knowledge extractor.

In the theory of Σ-protocols, i.e., 3-round interactive proofs, it is well known that 2-out-of-N special-soundness implies knowledge soundness with knowledge error $1/N$, where N is the size of the challenge set. This can be shown by a heavy-row type approach [Dam10; HL10]. The alternative knowledge soundness notion of Definition 2.31, requiring a *strict* polynomial time extractor that is allowed to have somewhat smaller success probability, is also implied by special-soundness. In [Cra96], it is shown how this follows by an application of Jensen's inequality.

The knowledge error $1/N$ of a 2-out-of-N special-sound interactive proof equals the probability that the prover guessed the challenge correctly before receiving it from the verifier. For this reason, it corresponds to the success probability of a trivial cheating strategy admitted by typical special-sound interactive proofs. In particular, every 3-round interactive proof that is *special honest-verifier zero-knowledge* admits such a cheating strategy. For this reason, the knowledge error $1/N$ is the best one can hope for and the implication from 2-out-of-N special-soundness to knowledge soundness is *tight*.

Recently, and in particular for compressed Σ-protocols, a natural generalization of special-soundness has become relevant: $\mathbf{k}$-*out-of-*$\mathbf{N}$ *special-soundness*, where $\mathbf{k} = (k_1, \ldots, k_\mu)$ and $\mathbf{N} = (N_1, \ldots, N_\mu)$. This is a generalization in two ways: (1) from requiring a colliding pair of transcripts to requiring a k-collision of k transcripts, i.e., k accepting transcripts with common first message and pairwise distinct challenges, and (2) from 3-round interactive proofs with 1 challenge, sent from the verifier to the prover, to $(2\mu + 1)$-round interactive proofs with μ challenges.

Typical $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs admit a cheating strategy that succeeds if at least one of the μ random challenges c_i, received from the verifier, hits a certain set Γ_i of size $k_i - 1$ chosen by the dishonest prover. The success probability of this cheating strategy is

$$\mathrm{Er}(k_1, \ldots, k_\mu; N_1, \ldots, N_\mu) := 1 - \prod_{i=1}^{\mu} \left(1 - \frac{k_i - 1}{N_i}\right). \tag{6.1}$$

This cheating strategy is a generalization of the one for 2-out-of-N Σ-protocols, where a dishonest prover succeeds if it guesses the challenge correctly before receiving it. Indeed, the latter has a success probability $\mathrm{Er}(2, N) = 1/N$, which matches the knowledge error of a 2-out-of-N special-sound Σ-protocol. It is not unnatural to expect $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs to be knowledge sound with knowledge error $\mathrm{Er}(\mathbf{k}, \mathbf{N})$.

However, in particular due to the generalization to *multi-round* interactive proofs, the mentioned extractor analyses are no longer directly applicable. Hence, it is not straightforward to show that **k**-out-of-**N** special-soundness also implies knowledge soundness if $\mu > 1$. For this reason, prior works resort to alternative arguments. Bootle et al. [BCC+16] give an asymptotic analysis of **k**-out-of-**N** special-sound interactive proofs. Their analysis is only applicable to interactive proofs with exponentially large challenge sets and it does not give an exact knowledge error. Wikström generalizes Bootle et al.'s analysis and constructs a knowledge extractor with a *configurable* knowledge error [Wik18]. At the cost of increasing the runtime, the knowledge error of this extractor can be configured to be arbitrarily close to

$$\sum_{i=1}^{\mu} \frac{k_i - 1}{N_i} > \mathrm{Er}(\mathbf{k}, \mathbf{N}) \,.$$

However, as the knowledge error moves closer to $\sum_{i=1}^{\mu} \frac{k_i-1}{N_i}$, the runtime of the extractor grows indefinitely. Moreover, also Wikström's analysis only applies to interactive proofs with exponentially large challenge sets.

As a consequence of these seemingly suboptimal extractor analyses, Hoffman, Klooß and Rupp raised the question whether there even exists an efficient knowledge extractor with knowledge error $\sum_{i=1}^{\mu} \frac{k_i-1}{N_i}$ [HKR19, Question D.4.]. Hence, at that time, it was unclear whether the knowledge error $\sum_{i=1}^{\mu} \frac{k_i-1}{N_i}$ was achievable, let alone the strictly smaller knowledge error $\mathrm{Er}(\mathbf{k}, \mathbf{N})$.

Recent works, while remaining non-tight, have improved the tightness and generalized the extraction to interactive proofs with smaller, i.e., not necessarily exponentially large, challenge sets [PLS19; JT20; AL21]. A common characteristic of all aforementioned approaches for analyzing multi-round knowledge extractors is the use of tail bounds, such as Markov's inequality, to bound the success probability and/or the (expected) runtime of the knowledge extractor. Using tail bounds, non-tightness appears to be unavoidable. In particular, Albrecht and Lai deemed a knowledge extractor with knowledge error $\sum_{i=1}^{\mu} \frac{k_i-1}{N_i}$ out of reach with current techniques [AL21].

6.3 A Partial Solution: Strict Polynomial Time Extraction

This section provides a partial solution towards our goal of proving that **k**-out-of-**N** special-soundness implies knowledge soundness. More precisely, we show that every **k**-out-of-**N** special-sound interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ admits a *strict* polynomial time extractor that, given a statement x and oracle access to a prover $\mathcal{P}^*$, succeeds in extracting a witness $w \in \mathfrak{R}(x)$ with probability at least

$$\big(\epsilon(x, \mathcal{P}^*) - \mathrm{Er}(\mathbf{k}; \mathbf{N})\big)^K \,,$$

where $\epsilon(x, \mathcal{P}^*)$ is the success probability of $\mathcal{P}^*$ attacking Π on public input x, $\mathrm{Er}(\mathbf{k}; \mathbf{N})$ is the knowledge error as defined in Equation (6.1) and $K = \prod_{i=1}^{\mu} k_i$.

This is only a partial solution for two reasons. First, the standard notion of knowledge soundness requires an extractor with success probability proportional

in $\epsilon(x, \mathcal{P}^*) - \mathrm{Er}(\mathbf{k}; \mathbf{N})$ instead of

$$\left(\epsilon(x, \mathcal{P}^*) - \mathrm{Er}(\mathbf{k}; \mathbf{N})\right)^K .$$

Therefore, this strict polynomial time extractor only shows that, for appropriate $\mathbf{k}$, $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs satisfy the alternative notion of knowledge soundness of Definition 2.31. Second, the requirements of Definition 2.31 are only satisfied if $K = \prod_{i=1}^{\mu} k_i$ is constant in the size of the input x. In fact, since the success probability of the extractor degrades exponentially in K, this result only gives a meaningful security notion if K is constant. Unfortunately, for many protocols of interest this is not the case and K even grows superlinearly in $|x|$.

Hence, the extractor presented in this section shows that, for a subclass of interactive proofs, $\mathbf{k}$-out-of-$\mathbf{N}$ special-soundness implies a meaningful, but alternative, notion of knowledge soundness. However, in contrast to the full solution that will be presented in Section 6.4, this extractor runs in *strict* polynomial time, which is known to be impossible if one insist on the standard notion of knowledge soundness [BL02].

6.3.1 Σ-Protocols

To simplify the exposition, we start with the simpler case of Σ-protocols, i.e., 3-round interactive proofs. The general case of multi-round interactive proofs will be treated in the subsequent section.

It is well known that a 2-out-of-N special-sound Σ-protocol admits a strict polynomial time extractor that succeeds with probability at least $(\epsilon(x, \mathcal{P}^*) - 1/N)^2$ [Cra96]. This result follows from an application of Jensen's inequality to the convex function $f(X) = X(X - 1/N)$. More precisely, Cramer defined the *collision-game* described below. This is essentially the game played by the knowledge extractor and Lemma 6.1 gives a lower bound for the probability of winning the game. Both the game and the lemma presented here are almost identical to the ones found in [Cra96]. We note that Bellare and Neven use similar techniques to prove the security of non-interactive protocols in the Fiat-Shamir paradigm [BN06].

Collision-Game. Consider a 0/1-matrix H with n rows and N columns. The rows correspond to the prover's randomness and the columns to the verifier's randomness. Therefore, every entry of H corresponds to a protocol transcript. An entry of the matrix is 1 if the transcript is accepting and 0 otherwise.

The game goes as follows. Select an entry of H uniformly at random. If this entry is a 1, select another entry of the same row uniformly at random. If this entry is again a 1, the game outputs success. If any of the selected entries equals a 0, the game is lost.

To bound the probability of winning the collision-game, Jensen's inequality is used, which states that, if X is a real-valued random variable and f is a continuous convex function defined on the support of X, it holds that

$$f\left(\mathbb{E}[X]\right) \leq \mathbb{E}[f(X)] .$$

Lemma 6.1 (Lemma 2.1 of [Cra96])**.** *Let H be a 0/1-matrix with n rows and N columns, and let ϵ denote the fraction of 1-entries in H. Then the probability of winning the collision-game is greater than or equal to $\epsilon(\epsilon - 1/N)$.*

Proof. For $1 \leq i \leq n$, let ϵ_i denote the fraction of 1-entries in the i-th row. Clearly, the probability of winning the collision-game is equal to[1]

$$\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\left(\frac{N\epsilon_i - 1}{N-1}\right) = \frac{1}{n}\frac{N}{N-1}\sum_{i=1}^{n}\epsilon_i\left(\epsilon_i - \frac{1}{N}\right) \geq \frac{1}{n}\sum_{i=1}^{n}\epsilon_i\left(\epsilon_i - \frac{1}{N}\right).$$

To complete the proof, observe that $\mathbb{E}[\epsilon_i] = \epsilon$, put $f(x) = x(x - 1/N)$ on the interval $[0, 1]$ and apply Jensen's inequality (using that f is a convex function). □

Using Lemma 6.1, it is straightforward to construct a strict polynomial time knowledge extractor that succeeds with probability at least $(\epsilon(x, \mathcal{P}^*) - 1/N)^2$ for 2-out-of-N special-sound Σ-protocols.

Instead, we show that the above argument can be adapted to show that, in order to satisfy the alternative notion of knowledge soundness (Definition 2.31), it is enough to consider *deterministic* provers $\mathcal{P}^*$. This observation simplifies the extractor analysis of interactive proofs and allows us to immediately handle the more general case of k-out-of-N special-sound Σ-protocols. In particular, the first message a sent by a deterministic prover is fixed, i.e., a does not vary over different invocations of the prover. Moreover, this observation will allow us to recursively generalize the analysis to *multi-round* interactive proofs.

Recall that for the standard definition of knowledge soundness (Definition 2.27), it is straightforward to see that one only needs to consider deterministic provers (Remark 2.3). The main reason is that $\epsilon(x, \mathcal{P}^*) - \kappa(|x|)$ is *linear* in $\mathcal{P}^*$'s success probability $\epsilon(x, \mathcal{P}^*)$ and linear functions commute with the expected value operator. This reasoning does not apply to the alternative notion of knowledge soundness, where extractors have success probability $\big(\epsilon(x, \mathcal{P}^*) - \kappa(|x|)\big)^c$ for some constant $c \geq 1$. However, by an appropriate application of Jensen's inequality, the argument can be adapted.

The following lemma shows that, also for the alternative notion of knowledge soundness, it is enough to consider deterministic provers.

Lemma 6.2 (Deterministic and Probabilistic Provers)**.** *Let $\Pi = (\mathcal{P}, \mathcal{V})$ be an interactive proof for relation $\mathfrak{R}$, $\kappa\colon \mathbb{N} \to [0, 1]$, $c \geq 1$ a constant and q a positive polynomial. Further, let $\mathcal{E}_{\mathsf{det}}$ be a knowledge extractor for Π that, given input x and oracle access to a* deterministic *prover $\mathcal{P}^*_{\mathsf{det}}$, runs in strict polynomial time and, if $\epsilon(x, \mathcal{P}^*_{\mathsf{det}}) \geq \kappa(|x|)$, succeeds in outputting a witness $w \in \mathfrak{R}(x)$ with probability*

$$\Pr\big((x; \mathcal{E}_{\mathsf{det}}^{\mathcal{P}^*_{\mathsf{det}}}(x)) \in \mathfrak{R}\big) \geq \frac{\big(\epsilon(x, \mathcal{P}^*_{\mathsf{det}}) - \kappa(|x|)\big)^c}{q(|x|)},$$

where $\epsilon(x, \mathcal{P}^*_{\mathsf{det}}) := \Pr\big((\mathcal{P}^*_{\mathsf{det}}, \mathcal{V})(x) = \mathsf{accept}\big)$.

[1] This is minor correction of the original proof, which incorrectly states that the success probability is equal to $\frac{1}{n}\sum_{i=1}^{n}\epsilon_i\left(\epsilon_i - \frac{1}{N}\right)$.

Then there exists a knowledge extractor $\mathcal{E}$ that, given input x and oracle access to a (possibly probabilistic) prover $\mathcal{P}^*$, runs in strict polynomial time and, if $\epsilon(x, \mathcal{P}^*) \geq \kappa(|x|)$, succeeds in outputting a witness $w \in \mathfrak{R}(x)$ with probability

$$\Pr\big((x; \mathcal{E}^{\mathcal{P}^*}(x)) \in \mathfrak{R}\big) \geq \frac{\big(\epsilon(x, \mathcal{P}^*) - \kappa(|x|)\big)^c}{q(|x|)}\,.$$

Proof. Let $\mathcal{P}^*$ be an arbitrary randomized dishonest prover, and let $\mathcal{P}^*[r]$ be the deterministic prover obtained by fixing $\mathcal{P}^*$'s randomness to r. Then $\epsilon(x, \mathcal{P}^*) = \mathbb{E}_r[\epsilon(x, \mathcal{P}^*[r])]$, where $\mathbb{E}_r$ denotes the expectation over the random choice of r.

Given input x and oracle access to $\mathcal{P}^*$, the knowledge extractor $\mathcal{E}$ is declared to run $\mathcal{E}_{\mathsf{det}}^{\mathcal{P}^*[r]}(x)$ for a random choice of r. Clearly, $\mathcal{E}^{\mathcal{P}^*}(x)$ runs in strict polynomial time. So let us analyze its success probability.

The extractor $\mathcal{E}^{\mathcal{P}^*}(x)$ succeeds with probability

$$\begin{aligned}
\Pr\big((x; \mathcal{E}^{\mathcal{P}^*}(x)) \in \mathfrak{R}\big) &= \mathbb{E}_r\big[\Pr\big((x; \mathcal{E}_{\mathsf{det}}^{\mathcal{P}^*[r]}(x)) \in \mathfrak{R}\big)\big] \\
&\geq \mathbb{E}_r\left[\frac{\big(\epsilon(x, \mathcal{P}^*[r]) - \kappa(|x|)\big)^c}{q(|x|)}\right] \\
&\geq \frac{\mathbb{E}_r\big[f\big(\epsilon(x, \mathcal{P}^*[r])\big)\big]}{q(|x|)}\,,
\end{aligned}$$

where the function f is defined as follows

$$f\colon \mathbb{R} \to \mathbb{R}\colon \quad \alpha \mapsto \begin{cases} \big(\alpha - \kappa(|x|)\big)^c, & \text{if } \alpha \geq \kappa(|x|)\,, \\ 0\,, & \text{otherwise.} \end{cases} \tag{6.2}$$

Note that, in the above, x is an arbitrary but *fixed* statement.

It is easily seen that f is twice-differentiable and, for all $\alpha \in \mathbb{R} \setminus \{\kappa(|x|)\}$, $f''(\alpha) \geq 0$. Moreover, for $\alpha_0 = \kappa(|x|)$ it holds that

$$\lim_{\alpha \uparrow \alpha_0} \frac{f(\alpha) - f(\alpha_0)}{\alpha - \alpha_0} = 0 \leq \lim_{\alpha \downarrow \alpha_0} \frac{f(\alpha) - f(\alpha_0)}{\alpha - \alpha_0}\,.$$

Hence, f is a convex function.

Therefore, by Jensen's inequality, it follows that, if $\epsilon(x, \mathcal{P}^*) \geq \kappa(|x|)$, the extractor $\mathcal{E}^{\mathcal{P}^*}(x)$ succeeds with probability

$$\begin{aligned}
\Pr\big((x; \mathcal{E}^{\mathcal{P}^*}(x)) \in \mathfrak{R}\big) &\geq \frac{\mathbb{E}_r\big[f\big(\epsilon(x, \mathcal{P}[r]^*)\big)\big]}{q(|x|)} \\
&\geq \frac{f\big(\mathbb{E}_r\big[\epsilon(x, \mathcal{P}[r]^*)\big]\big)}{q(|x|)} \\
&= \frac{f\big(\epsilon(x, \mathcal{P}^*)\big)}{q(|x|)} \\
&= \frac{\big(\epsilon(x, \mathcal{P}^*) - \kappa(|x|)\big)^c}{q(|x|)}\,,
\end{aligned}$$

which completes the proof.

□

Let us now return to the extractor analysis of k-out-of-N special-sound Σ-protocols. For multiple reasons, we will state and prove our core technical results in a more abstract language. One reason is that this allows us to focus on the important aspects. Another reason is that we will actually exploit the considered abstraction, and thus generalization, of the considered problem in the subsequent sections, where we consider parallel repetitions and Fiat-Shamir transformations. In particular, it allows us to unify the notation over the different sections of this chapter. The abstraction crucially depends on Lemma 6.2, showing that it is sufficient to consider deterministic provers $\mathcal{P}^*$.

In our abstraction, we consider an arbitrary function $V\colon \mathcal{C} \times \{0,1\}^* \to \{0,1\}$, $(c, y) \mapsto V(c, y)$, and an arbitrary (possibly probabilistic) algorithm $\mathcal{A}$ that takes as input an element $c \in \mathcal{C}$ and outputs a string $y \leftarrow \mathcal{A}(c)$. The *success probability* of $\mathcal{A}$ is then naturally defined as

$$\epsilon^V(\mathcal{A}) := \Pr\big(V(C, \mathcal{A}(C)) = 1\big)\,,$$

where, here and below, the probability space is defined by means of the randomness of $\mathcal{A}$ and the random variable C being uniformly random in $\mathcal{C}$.

The obvious instantiation of $\mathcal{A}$ is given by a *deterministic* dishonest prover $\mathcal{P}^*$ attacking the considered k-out-of-N special-sound Σ-protocol $\Pi = (\mathcal{P}, \mathcal{V})$ on input x. More precisely, on input c, $\mathcal{A}$ runs $\mathcal{P}^*$, sending c as the challenge, and outputs $\mathcal{P}^*$'s (fixed) first message a and its response z, and the function V is defined as the verification check that $\mathcal{V}$ performs. In this instantiation

$$\epsilon^V(\mathcal{A}) = \epsilon(x, \mathcal{P}^*)\,.$$

Moreover, we point out that this instantiation gives rise to a deterministic $\mathcal{A}$. However, later on, when generalizing the approach to the parallel composition of interactive proofs (Section 6.5), it will be crucial that in our abstract treatment, $\mathcal{A}$ may be an arbitrary *randomized* algorithm that decides on its output y in a randomized manner given the input c, and that V is arbitrary. Moreover, the more general treatment of probabilistic $\mathcal{A}$ does not complicate the analysis. Therefore, the abstraction will not be restricted to deterministic $\mathcal{A}$.

Motivated by the k-out-of-N special-soundness of the considered Σ-protocol, given oracle access to $\mathcal{A}$, the goal of the extractor will be to find correct responses $y_1, \ldots, y_k$ for k pairwise distinct challenges $c_1, \ldots, c_k \in \mathcal{C}$, i.e., such that $V(c_i, y_i) = 1$ for all i. This extractor $\mathcal{E}$ is formally described in Figure 6.1 and Lemma 6.3 shows that it runs in *strict* polynomial time and succeeds with probability at least

$$\left(\epsilon^V(\mathcal{A}) - \frac{k-1}{N}\right)^k\,.$$

Lemma 6.3 (Strict Polynomial Time Extraction - Σ-Protocols)**.** *Let $k \in \mathbb{N}$, $\mathcal{C}$ a finite set with cardinality $N \geq k$ and let $V\colon \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. Then there exists an oracle algorithm $\mathcal{E}$, described in Figure 6.1, with the following properties: The*

algorithm $\mathcal{E}^{\mathcal{A}}$, given oracle access to a (probabilistic) algorithm $\mathcal{A}\colon \mathcal{C} \to \{0,1\}^*$, requires at most k queries to $\mathcal{A}$ and, if $\epsilon^V(\mathcal{A}) \geq (k-1)/N$, with probability at least

$$\left(\epsilon^V(\mathcal{A}) - \frac{k-1}{N}\right)^k,$$

it outputs k pairs $(c_1, y_1), (c_2, y_2), \ldots, (c_k, y_k) \in \mathcal{C} \times \{0,1\}^*$ with $V(c_i, y_i) = 1$ for all i and $c_i \neq c_j$ for all $i \neq j$.

Proof. The extractor $\mathcal{E}^{\mathcal{A}}$ is described in Figure 6.1 and proceeds as follows. It samples $c_1 \in \mathcal{C}$ uniformly at random and evaluates $y_1 \leftarrow \mathcal{A}(c_1)$. If $V(c_1, y_1) = 0$, $\mathcal{E}^{\mathcal{A}}$ aborts. Otherwise, it samples $c_2 \in \mathcal{C} \setminus \{c_1\}$ uniformly at random and evaluates $y_2 \leftarrow \mathcal{A}(c_2)$. The extractor $\mathcal{E}^{\mathcal{A}}$ continues in this manner, until either it aborts, i.e., it finds a pair (c_i, y_i) with $V(c_i, y_i) = 0$, or until it has extracted k pairs $(c_1, y_1), (c_2, y_2), \ldots, (c_k, y_k) \in \mathcal{C} \times \{0,1\}^*$ with $V(c_i, y_i) = 1$ for all i and $c_i \neq c_j$ for all $i \neq j$.

Clearly, $\mathcal{E}^{\mathcal{A}}$ makes at most k queries to $\mathcal{A}$. Moreover, if $\epsilon^V(\mathcal{A}) \geq (k-1)/N$, its success probability is at least

$$\prod_{j=0}^{k-1} \frac{N}{N-j}\left(\epsilon^V(\mathcal{A}) - \frac{j}{N}\right) \geq \left(\epsilon^V(\mathcal{A}) - \frac{k-1}{N}\right)^k,$$

which completes the proof of the lemma. □

Figure 6.1: Strict Polynomial Time Extractor $\mathcal{E}$.

Parameters: $k \in \mathbb{N}$.
Oracle access to: Algorithm $\mathcal{A}\colon \mathcal{C} \to \{0,1\}^*$ and verification function $V\colon \mathcal{C} \times \{0,1\}^* \to \{0,1\}$.

- For $i \in \{1, \ldots, k\}$:
  - Sample $c_i \in \mathcal{C} \setminus \{c_1, \ldots, c_{i-1}\}$ uniformly at random and evaluate $y_i \leftarrow \mathcal{A}(c_i)$.
  - If $V(c_i, y_i) = 0$, abort. Else, continue.

Output: If $\mathcal{A}$ has not aborted, output k pairs $(c_1, y_1), \ldots, (c_k, y_k) \in \mathcal{C} \times \{0,1\}^*$, with $V(c_i, y_i) = 1$ for all i and $c_i \neq c_j$ for all $i \neq j$.

The following theorem is an immediate consequence of Lemma 6.2 and Lemma 6.3. It shows that, if k is constant in $|x|$, k-out-of-N special-sound Σ-protocols satisfy the alternative knowledge soundness notion of Definition 2.31.

Theorem 6.1 (Strict Polynomial Time Extraction for Σ-Protocols)**.** *Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a k-out-of-N special-sound Σ-protocol for relation $\mathfrak{R}$. Then there exists an extraction algorithm $\mathcal{E}$ with the following properties: The extractor $\mathcal{E}^{\mathcal{P}^*}(x)$, given input x and oracle access to a (potentially dishonest) prover $\mathcal{P}^*$, requires at*

most k queries to $\mathcal{P}^*$ and, if $\epsilon(x,\mathcal{P}^*) \geq (k-1)/N$, outputs a witness $w \in \mathfrak{R}(x)$ with probability

$$\Pr\big((x;\mathcal{E}^{\mathcal{P}^*}(x)) \in \mathfrak{R}\big) \geq \left(\epsilon(x,\mathcal{P}^*) - \frac{k-1}{N}\right)^k ,$$

where $\epsilon(x,\mathcal{P}^*) := \Pr\big((\mathcal{P}^*,\mathcal{V})(x) = \textsf{accept}\big)$.

6.3.2 Multi-Round Interactive Proofs

Let us now move to multi-round interactive proofs and show that **k**-out-of-**N** special-soundness, for $\mathbf{k} = (k_1,\ldots,k_\mu)$ and $\mathbf{N} = (N_1,\ldots,N_\mu)$, implies the existence of an extraction algorithm that requires at most $K = \prod_{i=1}^{\mu} k_i$ queries to $\mathcal{P}^*$ and succeeds in extracting a witness with probability at least

$$\left(\epsilon(x,\mathcal{P}^*) - \mathrm{Er}(\mathbf{k};\mathbf{N})\right)^K ,$$

where $\mathrm{Er}(\mathbf{k};\mathbf{N})$ is as defined in Equation 6.1. Note that $\mathrm{Er}(k;N) = (k-1)/N$, i.e., this is indeed a multi-round generalization of the result of Section 6.3.1.

As a multi-round generalization of the abstraction of the previous section, we now consider a (possibly randomized) algorithm $\mathcal{A}$ that takes as input a vector $(c_1,\ldots,c_\mu) \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$ of challenges and outputs a string y, and we consider a function

$$V\colon \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu \times \{0,1\}^* \to \{0,1\} .$$

The obvious instantiation is a deterministic prover $\mathcal{P}^*$ attacking the considered multi-round interactive proof $\Pi = (\mathcal{P},\mathcal{V})$ on input x. Formally, on input $(c_1,\ldots,c_\mu)$, $\mathcal{A}$ runs $\mathcal{P}^*$, sending c_1 in the first challenge round, c_2 in the second, etc., and eventually $\mathcal{A}$ outputs all of $\mathcal{P}^*$'s messages. Then the function V captures the verification procedure of $\mathcal{V}$, i.e., $V(c_1,\ldots,c_\mu,y) = 1$ if and only if the corresponding transcript is accepting for statements x. As before, this instantiation actually results in a deterministic algorithm $\mathcal{A}$. However, our analysis also allows probabilistic instantiations of $\mathcal{A}$.

Syntactically identical to the previous section, the *success probability* of $\mathcal{A}$ is defined as

$$\epsilon^V(\mathcal{A}) := \Pr\big(V(C,\mathcal{A}(C)) = 1\big) ,$$

where $C = (C_1,\ldots,C_\mu)$ is uniformly random in $\mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$. However, here the goal of the extractor is different: the goal is to find correct responses for a **k**-tree of challenge vectors (Definition 2.33). Note that, since the prover $\mathcal{P}^*$ is deterministic, any **k**-tree of challenge vectors corresponds uniquely to a **k**-tree of transcripts.

Towards constructing a knowledge extractor, we make the following observation. For notational convenience, let us write $\mathbf{k}_m = (1,\ldots,1,k_{m+1},\ldots,k_\mu)$ for all $1 \leq m \leq \mu$. Then, a **k**-tree of challenge vectors has the following recursive nature:

- A $(1,\ldots,1)$-tree of challenge vectors is simply a challenge vector $(c_1,\ldots,c_\mu)$;
- A $\mathbf{k}_{m-1}$-tree of challenge vectors is a set of k_m $\mathbf{k}_m$-trees, where all $\prod_{i=m}^{\mu} k_i$ challenge vectors have the first $m-1$ coordinates in common.

The following lemma exploits the recursive nature of $\mathbf{k}$-trees of challenge vectors and shows the existence of an extraction algorithm with the desired runtime and success probability.

Lemma 6.4 (Strict Polynomial Time Extraction - Multi-Round Protocols)**.** *Let $\mathbf{k} = (k_1, \ldots, k_\mu), \mathbf{N} = (N_1, \ldots, N_\mu) \in \mathbb{N}^\mu$, $\mathcal{C}_1, \ldots, \mathcal{C}_\mu$ finite sets with cardinality $|\mathcal{C}_i| = N_i \geq k_i$ and let $V \colon \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu \times \{0,1\}^* \to \{0,1\}$. Then there exists an algorithm $\mathcal{E}$ with the following properties: The algorithm $\mathcal{E}^{\mathcal{A}}$, given oracle access to a (probabilistic) algorithm $\mathcal{A} \colon \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu \to \{0,1\}^*$, requires at most $K = \prod_{i=1}^{\mu} k_i$ queries to $\mathcal{A}$ and, if $\epsilon^V(\mathcal{A}) \geq \mathrm{Er}(\mathbf{k}; \mathbf{N})$, with probability at least*

$$\left(\epsilon^V(\mathcal{A}) - \mathrm{Er}(\mathbf{k}; \mathbf{N})\right)^K,$$

it outputs K pairs $(\mathbf{c}_1, y_1), \ldots, (\mathbf{c}_K, y_K) \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu \times \{0,1\}^*$ with $V(\mathbf{c}_i, y_i) = 1$ for all i and such that the vectors $\mathbf{c}_i \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$ form a $\mathbf{k}$-tree of challenge vectors, where we recall that

$$\mathrm{Er}(\mathbf{k}; \mathbf{N}) = 1 - \prod_{i=1}^{\mu} \left(1 - \frac{k_i - 1}{N_i}\right).$$

Proof. The extraction algorithm $\mathcal{E}$ is defined recursively. To this end, we write $\mathbf{k}_m = (1, \ldots, 1, k_{m+1}, \ldots, k_\mu)$ and $K_m = \prod_{i=m+1}^{\mu} k_i$ for all $0 \leq m \leq \mu$, with the understanding that $\mathbf{k}_\mu = (1, \ldots, 1)$ and $K_\mu = 1$.

For all m and $\vec{c}_m = (c_1, \ldots, c_m) \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_m$, we let $\mathcal{E}_m^{\mathcal{A}}(\vec{c}_m)$ be the algorithm that, given oracle access to $\mathcal{A}$, aims to output K_m pairs $(\mathbf{c}_1, y_1), \ldots, (\mathbf{c}_{K_m}, y_{K_m}) \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu \times \{0,1\}^*$ with $V\big(\mathbf{c}_i, y_i\big) = 1$ for all i and such that the vectors $\mathbf{c}_i \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$ form a $\mathbf{k}_m$-tree of challenge vectors with the first m coordinates equal to $\vec{c}_m = (c_1, \ldots, c_m)$.

Let us now define the extraction algorithm $\mathcal{E}_m^{\mathcal{A}}(\vec{c}_m)$. For $m = \mu$ and $\vec{c}_\mu = (c_1, \ldots, c_\mu)$, $\mathcal{E}_\mu^{\mathcal{A}}(\vec{c}_\mu)$ simply evaluates $y \leftarrow \mathcal{A}(\vec{c}_\mu)$. If $V(\vec{c}_\mu, y) = 1$, $\mathcal{E}_\mu^{\mathcal{A}}(\vec{c}_\mu)$ successfully outputs $(\vec{c}_\mu, y)$. In this case we write $\mathcal{E}_\mu^{\mathcal{A}}(\vec{c}_\mu) \neq \perp$.

For $m < \mu$ and $\vec{c}_m = (c_1, \ldots, c_m)$, $\mathcal{E}_m^{\mathcal{A}}(\vec{c}_m)$ runs $\mathcal{E}_{m+1}^{\mathcal{A}}(\vec{c}_m, y_\ell)$ for $1 \leq \ell \leq k_{m+1}$ and $y_\ell \in \mathcal{C}_{m+1}$ sampled uniformly at random such that $y_i \neq y_j$ for all $i \neq j$. We say $\mathcal{E}_m^{\mathcal{A}}(\vec{c}_m)$ aborts if any of its $\mathcal{E}_{m+1}^{\mathcal{A}}$-invocations fails, i.e., if $\mathcal{E}_{m+1}^{\mathcal{A}}(\vec{c}_m, y_\ell) = \perp$ for some ℓ. If $\mathcal{E}_m^{\mathcal{A}}(\vec{c}_m)$ does not abort, it is easily seen that the k_{m+1} $\mathbf{k}_{m+1}$-trees, output by its $\mathcal{E}_{m+1}^{\mathcal{A}}$-invocations, form a $\mathbf{k}_m$-tree of challenge vectors.

The extraction algorithm $\mathcal{E}^{\mathcal{A}}$ simply runs $\mathcal{E}_0^{\mathcal{A}}$. Let us now analyze the expected number of $\mathcal{A}$-queries and success probability of $\mathcal{E}^{\mathcal{A}}$.

Expected Number of $\mathcal{A}$-Queries. By induction, it immediately follows that, for all m and $\vec{c}_m = (c_1, \ldots, c_m)$, $\mathcal{E}_m^{\mathcal{A}}(\vec{c}_m)$ makes at most $K_{m+1} = k_{m+1} \cdots k_\mu$ queries to $\mathcal{A}$. Hence, $\mathcal{E}^{\mathcal{A}}$ requires at most K queries to $\mathcal{A}$, which proves the claimed number of $\mathcal{A}$-queries.

Success Probability. For all m and $\vec{c}_m = (c_1, \ldots, c_m)$, let

$$\epsilon(\vec{c}_m) = \Pr\big(V(C, \mathcal{A}(C)) = 1 \mid C_1 = c_1 \wedge \cdots \wedge C_m = c_m\big),$$

where $C = (C_1, \ldots, C_\mu)$ is uniformly random in $\mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$, i.e., $\epsilon(\vec{c}_m)$ denotes the success probability of $\mathcal{A}$ conditioned on the first m challenges being equal to $\vec{c}_m$.

Moreover, similar to the convex function of Equation 6.2, we let

$$f_m \colon \mathbb{R} \to \mathbb{R} : \quad \alpha \mapsto \begin{cases} \big(\alpha - \mathrm{Er}(\mathbf{k}_m; \mathbf{N})\big)^{K_m}, & \text{if } \alpha \geq \mathrm{Er}(\mathbf{k}_m; \mathbf{N}), \\ 0, & \text{otherwise.} \end{cases}$$

By induction, for all $0 \leq m \leq \mu$ and for all $\vec{c}_m$, we will show that

$$\Pr\big(\mathcal{E}_m^{\mathcal{A}}(\vec{c}_m) \neq \perp\big) \geq f_m\big(\epsilon(\vec{c}_m)\big). \tag{6.3}$$

It holds that $\mathbf{k}_\mu = (1, \ldots, 1)$, $K_\mu = 1$ and $\mathrm{Er}(\mathbf{k}_\mu; \mathbf{N}) = 0$. Therefore,

$$\Pr\big(\mathcal{E}_\mu^{\mathcal{A}}(\vec{c}_\mu) \neq \perp\big) = \epsilon(\vec{c}_\mu) = f_\mu\big(\epsilon(\vec{c}_\mu)\big),$$

which proves the base case $m = \mu$.

So let us assume that the induction hypothesis of Equation 6.3 is satisfied for m and for all $\vec{c}_m \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_m$. Then, for all $\vec{c}_{m-1}$,

$$\begin{aligned} \Pr\big(\mathcal{E}_{m-1}^{\mathcal{A}}(\vec{c}_{m-1}) \neq \perp\big) &= \mathbb{E}_{y_1, \ldots, y_{k_m}} \left[\prod_{\ell=1}^{k_m} \Pr\big(\mathcal{E}_m^{\mathcal{A}}(\vec{c}_{m-1}, y_\ell) \neq \perp\big) \right] \\ &\geq \mathbb{E}_{y_1, \ldots, y_{k_m}} \left[\prod_{\ell=1}^{k_m} f_m\big(\epsilon(\vec{c}_{m-1}, y_\ell)\big) \right], \end{aligned}$$

where the expected value is over the random choices of pairwise distinct $y_1, \ldots, y_{k_m} \in \mathcal{C}_m$.

By basic probability theory it now follows that

$$\mathbb{E}_{y_1, \ldots, y_{k_m}} \left[\prod_{\ell=1}^{k_m} f_m\big(\epsilon(\vec{c}_{m-1}, y_\ell)\big) \right] = \prod_{\ell=1}^{k_m} \mathbb{E}_{y_\ell | y_1, \ldots, y_{\ell-1}} \Big[f_m\big(\epsilon(\vec{c}_{m-1}, y_\ell)\big) \Big],$$

where the latter expected values are over the random choices of the variables $y_\ell \in \mathcal{C}_m \setminus \{y_1, \ldots, y_{\ell-1}\}$, i.e., conditioned on the first $\ell - 1$ choices to be equal to $y_1, \ldots, y_{\ell-1}$.

Using the fact that f_m is convex and applying Jensen's inequality shows that

$$\Pr\big(\mathcal{E}_{m-1}^{\mathcal{A}}(\vec{c}_{m-1}) \neq \perp\big) \geq \prod_{\ell=1}^{k_m} f_m\Big(\mathbb{E}_{y_\ell | y_1, \ldots, y_{\ell-1}} \big[\epsilon(\vec{c}_{m-1}, y_\ell)\big]\Big).$$

Since y_ℓ is sampled uniformly at random from $\mathcal{C}_m \setminus \{y_1, \ldots, y_{\ell-1}\}$, it holds that

$$\begin{aligned} \mathbb{E}_{y_\ell | y_1, \ldots, y_{\ell-1}} \big[\epsilon(\vec{c}_{m-1}, y_\ell)\big] &= \frac{N_m \cdot \epsilon(\vec{c}_{m-1}) - \sum_{j=1}^{\ell-1} \epsilon(\vec{c}_{m-1}, y_j)}{N_m - \ell + 1} \\ &\geq \frac{N_m}{N_m - \ell + 1} \left(\epsilon(\vec{c}_{m-1}) - \frac{\ell - 1}{N_m} \right) \\ &= 1 - \frac{N_m}{N_m - \ell + 1} \big(1 - \epsilon(\vec{c}_{m-1})\big). \end{aligned}$$

Hence, since f_m is monotonically increasing,

$$\begin{aligned}\Pr\big(\mathcal{E}^{\mathcal{A}}_{m-1}(\vec{c}_{m-1}) \neq \perp\big) &\geq \prod_{\ell=1}^{k_m} f_m\left(1 - \frac{N_m}{N_m - \ell + 1}\left(1 - \epsilon(\vec{c}_{m-1})\right)\right) \\ &\geq f_m\left(1 - \frac{N_m}{N_m - k_m + 1}\big(1 - \epsilon(\vec{c}_{m-1})\big)\right)^{k_m}.\end{aligned}$$

To complete the proof, we must express this lower bound in terms of the function f_{m-1} instead of f_m. To this end, we first consider the case $\epsilon(\vec{c}_{m-1}) < \mathrm{Er}(\mathbf{k}_{m-1}; \mathbf{N})$. In this case

$$\begin{aligned}1 - \frac{N_m}{N_m - k_m + 1}\big(1 - \epsilon(\vec{c}_{m-1})\big) &< 1 - \frac{N_m}{N_m - k_m + 1}\left(1 - \mathrm{Er}(\mathbf{k}_{m-1}; \mathbf{N})\right) \\ &= \mathrm{Er}(\mathbf{k}_m; \mathbf{N})\,.\end{aligned}$$

Hence, in this case

$$f_m\left(1 - \frac{N_m}{N_m - k_m + 1}\big(1 - \epsilon(\vec{c}_{m-1})\big)\right)^{k_m} = f_{m-1}\big(\epsilon(\vec{c}_{m-1})\big) = 0\,.$$

So let us consider the other case, i.e., $\epsilon(\vec{c}_{m-1}) \geq \mathrm{Er}(\mathbf{k}_m; \mathbf{N})$. Then

$$\begin{aligned}&f_m\left(1 - \frac{N_m}{N_m - k_m + 1}\big(1 - \epsilon(\vec{c}_{m-1})\big)\right)^{k_m} \\ &= \left(1 - \frac{N_m}{N_m - k_m + 1}\big(1 - \epsilon(\vec{c}_{m-1})\big) - \mathrm{Er}(\mathbf{k}_m; \mathbf{N})\right)^{K_{m-1}} \\ &= \left(\left(\frac{N_m}{N_m - k_m + 1}\right)\left(\epsilon(\vec{c}_{m-1}) - 1 + \frac{N_m - k_m + 1}{N_m}\big(1 - \mathrm{Er}(\mathbf{k}_m; \mathbf{N})\big)\right)\right)^{K_{m-1}} \\ &= \left(\frac{N_m}{N_m - k_m + 1}\right)^{K_{m-1}} \cdot \left(\epsilon(\vec{c}_{m-1}) - \mathrm{Er}(\mathbf{k}_{m-1}; \mathbf{N})\big)\right)^{K_{m-1}} \\ &\geq \left(\epsilon(\vec{c}_{m-1}) - \mathrm{Er}(\mathbf{k}_{m-1}; \mathbf{N})\big)\right)^{K_{m-1}} \\ &= f_{m-1}\big(\epsilon(\vec{c}_{m-1})\big)\,.\end{aligned}$$

Altogether it follows that, for all $\vec{c}_{m-1} \in \mathcal{C}_1 \times \cdots \mathcal{C}_{m-1}$,

$$f_m\left(1 - \frac{N_m}{N_m - k_m + 1}\big(1 - \epsilon(\vec{c}_{m-1})\big)\right)^{k_m} \geq f_{m-1}\big(\epsilon(\vec{c}_{m-1})\big)\,,$$

and therefore,

$$\Pr\big(\mathcal{E}^{\mathcal{A}}_{m-1}(\vec{c}_{m-1}) \neq \perp\big) \geq f_{m-1}\big(\epsilon(\vec{c}_{m-1})\big)\,,$$

which proves the induction hypothesis of Equation 6.3.

In particular, if $\epsilon^V(\mathcal{A}) \geq \mathrm{Er}(\mathbf{k};\mathbf{N})$,

$$\Pr\big(\mathcal{E}^{\mathcal{A}} \neq \perp\big) = \Pr\big(\mathcal{E}_0^{\mathcal{A}} \neq \perp\big) \geq \big(\epsilon^V(\mathcal{A}) - \mathrm{Er}(\mathbf{k};\mathbf{N})\big)^K \ ,$$

which completes the proof of the lemma. □

The following theorem is an immediate consequence of Lemma 6.4. It shows that, if $K = k_1 \dots k_\mu$ is constant in $|x|$, $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs satisfy the alternative knowledge soundness notion of Definition 2.31 with knowledge error $\mathrm{Er}(\mathbf{k};\mathbf{N})$. The knowledge error $\mathrm{Er}(\mathbf{k};\mathbf{N})$ is tight, since $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs typically admit a cheating strategy that succeeds with probability $\mathrm{Er}(\mathbf{k};\mathbf{N})$.

Theorem 6.2 (Strict Polynomial Time Extraction for Multi-Round Protocols)**.** *Let $\mathbf{k} = (k_1,\dots,k_\mu), \mathbf{N} = (N_1,\dots,N_\mu) \in \mathbb{N}^\mu$ and let $K = k_1 \dots k_\mu$. Let $\Pi = (\mathcal{P},\mathcal{V})$ be a $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proof for relation $\mathfrak{R}$. Then there exists an extraction algorithm $\mathcal{E}$ with the following properties: The extractor $\mathcal{E}^{\mathcal{P}^*}(x)$, given input x and oracle access to a (potentially dishonest) prover $\mathcal{P}^*$, requires at most K queries to $\mathcal{P}^*$ and, if $\epsilon(x,\mathcal{P}^*) \geq \mathrm{Er}(\mathbf{k};\mathbf{N})$, outputs a witness $w \in \mathfrak{R}(x)$ with probability*

$$\Pr\big((x;\mathcal{E}^{\mathcal{P}^*}(x)) \in \mathfrak{R}\big) \geq (\epsilon(x,\mathcal{P}^*) - \mathrm{Er}(\mathbf{k};\mathbf{N}))^K \ ,$$

where $\epsilon(x,\mathcal{P}^*) := \Pr\big((\mathcal{P}^*,\mathcal{V})(x) = \mathsf{accept}\big)$ and

$$\mathrm{Er}(\mathbf{k};\mathbf{N}) = 1 - \prod_{i=1}^{\mu}\left(1 - \frac{k_i - 1}{N_i}\right).$$

6.4 A Complete Solution in Expected Polynomial Time

In this section, we present a complete solution to the knowledge soundness problem for $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs, i.e., we prove that $\mathbf{k}$-out-of-$\mathbf{N}$ special-soundness implies knowledge soundness with knowledge error $\mathrm{Er}(\mathbf{k},\mathbf{N})$. More precisely, towards satisfying Definition 2.27, we construct a knowledge extractor that, given a statement x and oracle access to a prover $\mathcal{P}^*$, runs in *expected* polynomial time and succeeds in extracting a witness $w \in \mathfrak{R}(x)$ with probability at least

$$\epsilon(x,\mathcal{P}^*) - \mathrm{Er}(\mathbf{k},\mathbf{N}) \,,$$

where $\epsilon(x,\mathcal{P}^*)$ is the success probability of $\mathcal{P}^*$ on input x. Therefore, with respect to the partial solution of Section 6.3 we show that, at the cost of relaxing from *strict* to *expected* polynomial time extraction, the success probability can be increased from

$$\big(\epsilon(x,\mathcal{P}^*) - \mathrm{Er}(\mathbf{k},\mathbf{N})\big)^K \quad \text{to} \quad \epsilon(x,\mathcal{P}^*) - \mathrm{Er}(\mathbf{k},\mathbf{N}) \,,$$

where $K = \prod_{i=1}^{\mu} k_i$. This shows that indeed $\mathbf{k}$-out-of-$\mathbf{N}$ special-soundness *tightly* implies knowledge soundness with knowledge error $\mathrm{Er}(\mathbf{k},\mathbf{N})$.

6.4.1 Σ-Protocols

As before, to simplify the exposition, we start with the simpler case of Σ-protocols, i.e., 3-round interactive proofs. Moreover, we use the same abstract notation, i.e., we consider an arbitrary algorithm $\mathcal{A}\colon \mathcal{C} \to \{0,1\}^*$ and an arbitrary verification function $V\colon \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. Recall that $\mathcal{A}$ has a naturally defined success probability

$$\epsilon^V(\mathcal{A}) := \Pr\big(V(C, \mathcal{A}(C)) = 1\big)\,,$$

where C is uniformly random in $\mathcal{C}$. The obvious instantiation of $\mathcal{A}$ is given by a *deterministic*[2] prover $\mathcal{P}^*$ attacking the considered k-out-of-N special-sound Σ-protocol $\Pi = (\mathcal{P}, \mathcal{V})$ on input x.

As before, given oracle access to $\mathcal{A}$, the goal is to find correct responses $y_1, \ldots, y_k$ for k pairwise distinct challenges $c_1, \ldots, c_k \in \mathcal{C}$, i.e., such that $V(c_i, y_i) = 1$ for all i. However, this time we follow a different approach. The first step of the extractor is the same as in Section 6.3.1, i.e., it samples a random challenge c_1 and evaluates $y_1 \leftarrow \mathcal{A}(c_1)$. If $V(c_1, y_1) = 0$, the extractor aborts. Otherwise, i.e., if $V(c_1, y_1) = 1$, the extractor samples challenges from $\mathcal{C} \setminus \{c_1\}$, without replacement, until either $k-1$ additional pairs $(c_2, y_2), \ldots, (c_k, y_k)$, with $V(c_i, y_i) = 1$ for all i, have been found or until the entire challenge set $\mathcal{C}$ has been exhausted. This extraction algorithm is also described in Figure 6.2 and its properties are summarized in Lemma 6.5.

Recall that the strict polynomial-time extractor of Section 6.3.1 aborts if any pair (c, y) with $V(c, y) = 0$ is encountered. By contrast, if $V(c_1, y_1) = 1$, the expected polynomial time extractor described here continues searching until it has succeeded or until there are no more challenges to try. Lemma 6.5 shows that this adaptation increases the success probability from $(\epsilon^V(\mathcal{A}) - (k-1)/N)^k$ to $\epsilon^V(\mathcal{A}) - (k-1)/N$, where $N = |\mathcal{C}|$. The cost of this improvement is a degradation from strict to expected polynomial runtime. However, this is still sufficient for proving knowledge soundness.

The proof of the following lemma can be simplified by restricting to deterministic algorithms $\mathcal{A}$. This would still be sufficient for proving knowledge soundness. However, in the next section, for multi-round protocols, we will apply this lemma recursively and there it is crucial that $\mathcal{A}$ is allowed to be probabilistic.

Lemma 6.5 (Expected Polynomial Time Extraction - Σ-Protocols)**.** *Let $k \in \mathbb{N}$, $\mathcal{C}$ a finite set with cardinality $N \geq k$ and let $V\colon \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. Then there exists an oracle algorithm $\mathcal{E}$ with the following properties: The algorithm $\mathcal{E}^{\mathcal{A}}$, given oracle access to a (probabilistic) algorithm $\mathcal{A}\colon \mathcal{C} \to \{0,1\}^*$, requires an expected number of at most k queries to $\mathcal{A}$ and with probability at least*

$$\frac{N}{N-k+1}\left(\epsilon^V(\mathcal{A}) - \frac{k-1}{N}\right)\,,$$

it outputs k pairs $(c_1, y_1), (c_2, y_2), \ldots, (c_k, y_k) \in \mathcal{C} \times \{0,1\}^*$ with $V(c_i, y_i) = 1$ for all i and $c_i \neq c_j$ for all $i \neq j$.

[2] Recall that, in order to prove knowledge soundness, it is sufficient to consider deterministic provers (Remark 2.3).

Figure 6.2: Expected Polynomial Time Extractor $\mathcal{E}$.

Parameters: $k \in \mathbb{N}$.
Oracle access to: Algorithm $\mathcal{A}\colon \mathcal{C} \to \{0,1\}^*$ and verification function $V\colon \mathcal{C} \times \{0,1\}^* \to \{0,1\}$.

- Sample $c_1 \in \mathcal{C}$ uniformly at random and evaluate $y_1 \leftarrow \mathcal{A}(c_1)$.
- If $V(c_1, y_1) = 0$, abort.
- Else, repeat
  - sample $c \in \mathcal{C} \setminus \{c_1\}$ uniformly at random (without replacement) and evaluate $y \leftarrow \mathcal{A}(c)$;

  until either $k-1$ additional pairs $(c_2, y_2), \ldots, (c_k, y_k)$, with $V(c_i, y_i) = 1$ for all i, have been found or until all challenges $c \in \mathcal{C} \setminus \{c_1\}$ have been tried.

Output: In the former case, output k pairs $(c_1, y_1), \ldots, (c_k, y_k) \in \mathcal{C} \times \{0,1\}^*$ with $V(c_i, y_i) = 1$ for all i and $c_i \neq c_j$ for all $i \neq j$.

Proof. The extractor $\mathcal{E}^{\mathcal{A}}$, given oracle access to $\mathcal{A}$, is described in Figure 6.2 and proceeds as follows. It samples a random challenge c_1 and evaluates $y_1 \leftarrow \mathcal{A}(c_1)$. If $V(c_1, y_1) = 0$, the extractor aborts. Otherwise, if $V(c_1, y_1) = 1$, the extractor samples challenges from $\mathcal{C} \setminus \{c_1\}$, without replacement, until either $k-1$ additional pairs $(c_2, y_2), \ldots, (c_k, y_k)$, with $V(c_i, y_i) = 1$ for all i, have been found or until the entire challenge set $\mathcal{C}$ has been exhausted.

We write C_1 for the random variable denoting the first challenge sampled by the extractor, i.e., C_1 is uniformly random in $\mathcal{C}$. Moreover, we write $\Gamma = 0$ and $\Gamma = 1$ for the events $V\big(C_1, \mathcal{A}(C_1)\big) = 0$ and $V\big(C_1, \mathcal{A}(C_1)\big) = 1$, respectively. In particular, note that

$$\epsilon^V(\mathcal{A}) = \Pr(\Gamma = 1)\,.$$

Let us now analyze the expected number of $\mathcal{A}$-queries and the success probability of the extractor $\mathcal{E}^{\mathcal{A}}$.

Expected Number of $\mathcal{A}$-Queries. Let T denote the number of $\mathcal{A}$-queries made by $\mathcal{E}^{\mathcal{A}}$. Moreover, let S denote the number of challenges $c \in \mathcal{C}$ for which $\mathcal{A}$ returns a correct response, i.e., $S = \big|\{c \in \mathcal{C} \mid V\big(c, \mathcal{A}(c)\big) = 1\}\big|$. Note that, since $\mathcal{A}$ is probabilistic, S is a random variable with support $\{0, \ldots, N\}$.

Let us now assume that the first $\mathcal{A}$-query by $\mathcal{E}^{\mathcal{A}}$ is successful, i.e., $\Gamma = 1$. Then conditioned on $S = \ell > 0$, the remainder of the extraction algorithm can be modeled by a *negative hyper geometric distribution*; challenges are drawn (without replacement) from a set of size $N-1$ containing $\ell - 1$ correct responses.

Therefore, by Lemma 2.3,

$$\mathbb{E}[T \mid \Gamma = 1 \wedge S = \ell > 0] \leq k + (k-1)\frac{N-\ell}{\ell} = 1 + (k-1)\frac{N}{\ell}\,.$$

Moreover, $\Gamma = 0$ implies $T = 1$ and thus $\mathbb{E}[T \mid \Gamma = 0 \wedge S = \ell] = 1$. Hence, for all

$0 < \ell \leq N$,

$$\begin{aligned}
\mathbb{E}[T \mid S = \ell] &= \Pr\big(\Gamma = 0 \mid S = \ell\big) \cdot \mathbb{E}[T \mid \Gamma = 0 \wedge S = \ell] \\
&\quad + \Pr\big(\Gamma = 1 \mid S = \ell\big) \cdot \mathbb{E}[T \mid \Gamma = 1 \wedge S = \ell] \\
&= \frac{N-\ell}{N} \cdot \mathbb{E}[T \mid \Gamma = 0 \wedge S = \ell] + \frac{\ell}{N} \cdot \mathbb{E}[T \mid \Gamma = 1 \wedge S = \ell] \\
&\leq \frac{N-\ell}{N} + \frac{\ell}{N} \cdot \left(1 + (k-1)\frac{N}{\ell}\right) \\
&= 1 + k - 1 = k\,.
\end{aligned}$$

Since $\mathbb{E}[T \mid S = 0] = 1$, it follows that $\mathbb{E}[T] \leq k$, which proves the claimed expected number of $\mathcal{A}$-queries made by $\mathcal{E}^{\mathcal{A}}$.

Success Probability. The extractor succeeds if $S \geq k$ and the first challenge returns a correct response, i.e.,

$$\begin{aligned}
\Pr\big(\mathcal{E}^{\mathcal{A}} \neq \perp\big) = \Pr\big(\Gamma = 1 \wedge S \geq k\big) &= \sum_{\ell=k}^{N} \Pr\big(S = \ell\big) \Pr\big(\Gamma = 1 \mid S = \ell\big) \\
&= \sum_{\ell=k}^{N} \Pr\big(S = \ell\big) \frac{\ell}{N}\,.
\end{aligned}$$

Now, for $\ell \leq N$, note that

$$\begin{aligned}
\frac{\ell}{N} &= 1 - \left(1 - \frac{\ell}{N}\right) \geq 1 - \frac{N}{N-k+1}\left(1 - \frac{\ell}{N}\right) \\
&= \frac{N}{N-k+1}\left(\frac{N-k+1}{N} - 1 + \frac{\ell}{N}\right) = \frac{N}{N-k+1}\left(\frac{\ell}{N} - \frac{k-1}{N}\right).
\end{aligned}$$

Hence,

$$\begin{aligned}
\Pr\big(\mathcal{E}^{\mathcal{A}} \neq \perp\big) &\geq \sum_{\ell=k}^{N} \Pr\big(S = \ell\big) \frac{N}{N-k+1}\left(\frac{\ell}{N} - \frac{k-1}{N}\right) \\
&\geq \sum_{\ell=0}^{N} \Pr\big(S = \ell\big) \frac{N}{N-k+1}\left(\frac{\ell}{N} - \frac{k-1}{N}\right) \\
&= \frac{N}{N-k+1}\left(\Pr\big(\Gamma = 1\big) - \frac{k-1}{N}\right) \\
&= \frac{N}{N-k+1}\left(\epsilon^{V}(\mathcal{A}) - \frac{k-1}{N}\right),
\end{aligned}$$

which completes the proof of the lemma. □

From Lemma 6.5 it immediately follows that k-out-of-N special-soundness tightly implies knowledge soundness.

Theorem 6.3 (Knowledge Soundness of Σ-Protocols)**.** *Let* $\Pi = (\mathcal{P}, \mathcal{V})$ *be a* k*-out-of-*N *special-sound* Σ*-protocol for relation* $\mathfrak{R}$*. Then* Π *is knowledge sound with knowledge error* $\mathrm{Er}(k; N) = (k-1)/N$*.*

An alternative knowledge extractor for 2-out-of-N special-sound Σ-protocols, proving Theorem 6.3 for this special case, can be found in [HL10]. Their extractor follows a heavy-row type approach and is designed towards satisfying the equivalent, but different, knowledge soundness definition (Definition 2.28). Therefore, in order to compare the two approaches, one must perform a generic transformation [Gol04]. Concretely, towards satisfying Definition 2.28, our extractor can be repeated until it succeeds resulting in a knowledge extractor for 2-out-of-N special-sound Σ-protocols that, if $\epsilon^V(\mathcal{A}) > 1/N$, always succeeds and requires an expected number of at most

$$\frac{2}{\epsilon^V(\mathcal{A}) - 1/N}$$

queries to $\mathcal{A}$.

Our approach simplifies the extraction algorithm and its analysis. The crucial difference is that, instead of sampling challenges with replacement, our extractor samples new challenges *without* replacement. Most importantly, as we will show in the next section, our approach allows for a generalization to the multi-round case. By contrast, all known multi-round generalizations of the heavy-row approach of [HL10] result in suboptimal knowledge errors and expected runtimes.

6.4.2 Multi-Round Interactive Proofs

We are now ready to prove that **k**-out-of-**N** special-sound multi-round interactive proofs are indeed knowledge sound with knowledge error $\mathrm{Er}(\mathbf{k}; \mathbf{N})$. We use the same abstract notation as in Section 6.3.2, i.e., we consider an arbitrary probabilistic algorithm $\mathcal{A}\colon \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu \to \{0,1\}^*$ and an arbitrary verification function

$$V\colon \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu \times \{0,1\}^* \to \{0,1\}\,.$$

The obvious instantiation of $\mathcal{A}$ is given by a deterministic prover $\mathcal{P}^*$ attacking the considered interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ on input x. Recall that $\mathcal{A}$'s success probability is denoted as

$$\epsilon^V(\mathcal{A}) := \Pr\big(V(C, \mathcal{A}(C)) = 1\big)\,,$$

where $C = (C_1, \ldots, C_\mu)$ is uniformly random in $\mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$.

The goal of the extractor is, given oracle access to $\mathcal{A}$, to find correct responses for a **k**-tree of challenge vectors (Definition 2.33). The following lemma shows the existence of an extractor with the desired properties. The extractor is a recursive application of the 3-round extractor of Lemma 6.5.

Lemma 6.6 (Expected Polynomial Time Extraction - Multi-Round Protocols)**.** *Let* $\mathbf{k} = (k_1, \ldots, k_\mu), \mathbf{N} = (N_1, \ldots, N_\mu) \in \mathbb{N}^\mu$*,* $\mathcal{C}_1, \ldots, \mathcal{C}_\mu$ *finite sets with cardinality* $|\mathcal{C}_i| = N_i \geq k_i$ *and let* $V\colon \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu \times \{0,1\}^* \to \{0,1\}$*. Then there exists an oracle algorithm* $\mathcal{E}$ *with the following properties: The algorithm* $\mathcal{E}^{\mathcal{A}}$*, given oracle*

access to a (probabilistic) algorithm $\mathcal{A}\colon \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu \to \{0,1\}^*$*, requires an expected number of at most* $K = \prod_{i=1}^{\mu} k_i$ *queries to* $\mathcal{A}$ *and with probability at least*

$$\frac{1}{1 - \mathrm{Er}(\mathbf{k}; \mathbf{N})} \left(\epsilon^V(\mathcal{A}) - \mathrm{Er}(\mathbf{k}; \mathbf{N}) \right) ,$$

it outputs K *pairs* $(\mathbf{c}_1, y_1), \ldots, (\mathbf{c}_K, y_K) \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu \times \{0,1\}^*$ *with* $V(\mathbf{c}_i, y_i) = 1$ *for all* i *and such that the vectors* $\mathbf{c}_i \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$ *form a* $\mathbf{k}$*-tree of challenge vectors, where we recall that*

$$\mathrm{Er}(\mathbf{k}; \mathbf{N}) = 1 - \prod_{i=1}^{\mu} \left(1 - \frac{k_i - 1}{N_i} \right) .$$

Proof. The proof goes by induction on μ. For $\mu = 1$, the lemma directly follows from Lemma 6.5. So let $\mu > 1$ and let us assume the lemma holds for $\mu = M$ and consider the case $\mu = M + 1$.

For any $c \in \mathcal{C}_1$, let $\mathcal{A}_c$ be the algorithm that takes as input a vector $(c^2, \ldots, c^\mu) \in \mathcal{C}_2 \times \cdots \times \mathcal{C}_\mu$ and runs $\mathcal{A}(c, c^2, \ldots, c^\mu)$. The function V_c is defined accordingly, i.e.,

$$V_c\colon \mathcal{C}_2 \times \cdots \times \mathcal{C}_\mu \times \{0,1\}^* \to \{0,1\}, \quad (\mathbf{c}, y) \mapsto V(c, \mathbf{c}, y) .$$

Moreover, let $\mathbf{k}' = (k_2, \ldots, k_\mu), \mathbf{N}' = (N_2, \ldots, N_\mu) \in \mathbb{N}^{\mu-1}$ and $K' = \prod_{i=2}^{\mu} k_i$.

By the induction hypothesis there exists an algorithm $\mathcal{E}_{\mu-1}^{\mathcal{A}_c}$ that, given oracle access to $\mathcal{A}_c$, aims to output a set $\mathcal{Y}$ of K' pairs $(\mathbf{c}_1, y_1), \ldots, (\mathbf{c}_{K'}, y_{K'}) \in \mathcal{C}_2 \times \cdots \times \mathcal{C}_\mu \times \{0,1\}^*$ with $V(c, \mathbf{c}_i, y_i) = 1$ for all i such that the vectors $\mathbf{c}_i \in \mathcal{C}_2 \times \cdots \times \mathcal{C}_\mu$ form a $\mathbf{k}'$-tree of challenge vectors. Moreover, $\mathcal{E}_{\mu-1}^{\mathcal{A}_c}$ requires an expected number of at most K' queries to $\mathcal{A}$ and succeeds with probability at least

$$\frac{1}{1 - \mathrm{Er}(\mathbf{k}'; \mathbf{N}')} \left(\epsilon^{V_c}(\mathcal{A}_c) - \mathrm{Er}(\mathbf{k}'; \mathbf{N}') \right) .$$

We define $W\colon \mathcal{C}_1 \times \{0,1\}^* \to \{0,1\}$, by setting $W(c, \mathcal{Y}) = 1$ if and only if $\mathcal{Y}$ is a set satisfying the above properties.

Now let $\mathcal{B}^{\mathcal{A}}\colon \mathcal{C}_1 \to \{0,1\}^*$ be the algorithm that, given oracle access to $\mathcal{A}$, takes as input an element $c \in \mathcal{C}_1$ and runs $\mathcal{E}_{\mu-1}^{\mathcal{A}_c}$. By Lemma 6.5, there exists an expected polynomial time algorithm $\mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$ that, given oracle access to $\mathcal{B}^{\mathcal{A}}$, aims to output k_1 pairs $(c_1, \mathcal{Y}_1), \ldots, (c_{k_1}, \mathcal{Y}_{k_1}) \in \mathcal{C}_1 \times \{0,1\}^*$ with $W(c_i, \mathcal{Y}_i) = 1$ for all i and $c_i \neq c_j$ for all $i \neq j$. Clearly, the set of k_1 $\mathbf{k}'$-trees of challenge vectors forms a $\mathbf{k}$-tree. For this reason, the extractor $\mathcal{E}^{\mathcal{A}}$ is simply defined to run $\mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$. Note that, by the associativity of the composition of oracle algorithms, $\mathcal{E}^{\mathcal{A}} = \mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}} = (\mathcal{E}_1^{\mathcal{B}})^{\mathcal{A}}$ is indeed an oracle algorithm given oracle access to $\mathcal{A}$.

Let us now analyze the success probability and the expected number of $\mathcal{A}$-queries of the algorithm $\mathcal{E}^{\mathcal{A}} = \mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$.

Success Probability. By Lemma 6.5, and the induction hypothesis, it follows

that $\mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$ succeeds with probability at least

$$\begin{aligned}
&\frac{N_1}{N_1-k_1+1}\left(\epsilon^W\big(\mathcal{B}^{\mathcal{A}}\big)-\frac{k_1-1}{N_1}\right)\\
&\quad=\frac{N_1}{N_1-k_1+1}\left(\mathbb{E}_c\big[\Pr\big(\mathcal{E}_{\mu-1}^{\mathcal{A}_c}\neq\perp\big)\big]-\frac{k_1-1}{N_1}\right)\\
&\quad\geq\frac{N_1}{N_1-k_1+1}\left(\mathbb{E}_c\left[\frac{1}{1-\mathrm{Er}(\mathbf{k}';\mathbf{N}')}\left(\epsilon^{V_c}(\mathcal{A}_c)-\mathrm{Er}(\mathbf{k}';\mathbf{N}')\right)\right]-\frac{k_1-1}{N_1}\right)\\
&\quad=\frac{N_1}{N_1-k_1+1}\left(\frac{1}{1-\mathrm{Er}(\mathbf{k}';\mathbf{N}')}\left(\epsilon^{V}(\mathcal{A})-\mathrm{Er}(\mathbf{k}';\mathbf{N}')\right)-\frac{k_1-1}{N_1}\right)\\
&\quad=\frac{1}{1-\mathrm{Er}(\mathbf{k};\mathbf{N})}\left(\epsilon^{V}(\mathcal{A})-\mathrm{Er}(\mathbf{k}';\mathbf{N}')-\frac{k_1-1}{N_1}\big(1-\mathrm{Er}(\mathbf{k}';\mathbf{N}')\big)\right)\\
&\quad=\frac{1}{1-\mathrm{Er}(\mathbf{k};\mathbf{N})}\left(\epsilon^{V}(\mathcal{A})-1+\frac{N_1-k_1+1}{N_1}\big(1-\mathrm{Er}(\mathbf{k}';\mathbf{N}')\big)\right)\\
&\quad=\frac{1}{1-\mathrm{Er}(\mathbf{k};\mathbf{N})}\left(\epsilon^{V}(\mathcal{A})-\mathrm{Er}(\mathbf{k};\mathbf{N})\right)\,,
\end{aligned}$$

where we (twice) use the recursive relation

$$1-\mathrm{Er}(\mathbf{k};\mathbf{N})=\frac{N_1-k_1+1}{N_1}\left(1-\mathrm{Er}(\mathbf{k}';\mathbf{N}')\right)\,.$$

This shows that $\mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$ has the desired success probability.

Expected Number of $\mathcal{A}$-Queries. By Lemma 6.5 it follows that $\mathcal{E}_{\mu-1}^{\mathcal{A}_c}$ requires an expected number of at most k_1 queries to $\mathcal{B}^{\mathcal{A}}$ for all c. Moreover, by the induction hypothesis, $\mathcal{B}^{\mathcal{A}}$ requires an expected number of at most K' queries to $\mathcal{A}$. Hence, $\mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$ requires an expected number of at most K queries to $\mathcal{A}$, which completes the proof of the lemma. □

Remark 6.1. In the proof of Lemma 6.6, it is crucial that the algorithm $\mathcal{B}^{\mathcal{A}}$ is allowed to be *probabilistic*. For this reason, we did not restrict Lemma 6.5 to deterministic algorithms, even though this would have been sufficient for proving knowledge soundness of k-out-of-N special-sound Σ-protocols.

From Lemma 6.6 it immediately follows that, also for multi-round protocols, $\mathbf{k}$-out-of-$\mathbf{N}$ special-soundness tightly implies knowledge soundness. This result is summarized in the following theorem.

Theorem 6.4 (Knowledge Soundness of Multi-Round Interactive Proofs)**.** *Let $\Pi=(\mathcal{P},\mathcal{V})$ be a $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proof for relation $\mathfrak{R}$, where $\mathbf{k}=(k_1,\ldots,k_\mu)$, $\mathbf{N}=(N_1,\ldots,N_\mu)\in\mathbb{N}^\mu$. Then Π is knowledge sound with knowledge error*

$$\mathrm{Er}(\mathbf{k};\mathbf{N})=1-\prod_{i=1}^{\mu}\left(1-\frac{k_i-1}{N_i}\right).$$

6.4.3 A Note on Witness Extended Emulation

A technical issue arises when using proofs of knowledge as sub-protocols in larger cryptographic protocols [GK96; Lin01; Lin03]. More precisely, to prove security of the compound protocol, a simulator is typically required to run the extractor of the proof of knowledge. However, the naive simulation approach does not necessarily run in polynomial time. To this end, Lindell defined the notion of *witness-extended emulation*, capturing precisely the properties required when proofs of knowledge are used as sub-protocols [Lin01; Lin03]. Moreover, he showed that any proof of knowledge, with negligible knowledge error, has witness-extended emulation, thereby solving this technical issue for all proofs of knowledge at once. Hence, from our extraction analysis it follows that any $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proof has witness-extended emulation if $\mathrm{Er}(\mathbf{k}, \mathbf{N})$ is negligible.

The first multi-round extractor analysis for $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs considered witness emulation directly [BCC+16], i.e., it did not show that $\mathbf{k}$-out-of-$\mathbf{N}$ special-soundness implies knowledge soundness, but merely that it implies witness extended emulation. In particular, their analysis does not provide a concrete knowledge error and only applies to protocols with exponentially large challenge sets.

6.5 Solving the Parallel Repetition Problem

In certain occasions, the knowledge error of a "basic" proof of knowledge (and thereby the cheating probability of a dishonest prover) is not small enough, and thus needs to be reduced. In particular, this is the case for lattice-based proofs of knowledge (PoKs), where typically challenge sets are only of polynomial size resulting in nonnegligible knowledge errors [LS18; ACX21]. Reducing the knowledge error can be done generically by repeating the PoK. Indeed, repeating a PoK t times *sequentially*, i.e., one after the other, is known to reduce the knowledge error from κ down to κ^t [Gol01]. However, this approach also increases the number of communication rounds by a factor t. This is often undesirable, and sometimes even insufficient, e.g., because the security loss of the Fiat-Shamir transformation, transforming interactive into non-interactive protocols, is oftentimes exponential in the number of rounds (see Section 6.6).

Therefore, it is much more attractive to try to reduce the knowledge error by *parallel* repetition. However, analyzing parallel repetitions is significantly more complicated than analyzing sequential repetitions, because a dishonest prover does not have to treat all t parallel instances independently, i.e., a message corresponding to a specific instance may depend on the messages and challenges of the other parallel instances. In fact, it is not true in general that the t-fold parallel repetition decreases the knowledge error from κ down to κ^t; there even exist interactive arguments for which parallel repetition does not decrease the success probability of a dishonest prover at all [BIN97; PW07].

For this reason, parallel repetition of interactive proofs has been studied extensively, but mainly in the context of decreasing the *soundness error* [HPW+10; CL10; CP15]. However, knowledge soundness is a strictly stronger requirement than soundness; there exist interactive proofs that are sound but not knowledge

sound. More precisely, proving the existence of an efficient knowledge extractor is a much more delicate task than proving that the verifier is unlikely to accept a false statement.

In the special case of 2-out-of-N special-sound interactive proofs such a parallel repetition is much easier to analyze: the t-fold parallel repetition of a 2-special-sound interactive proof with challenge space of cardinality N is again 2-special-sound, but now with a challenge space of size N^t, and so knowledge-soundness with knowledge error $1/N^t$ follows immediately from the generic reduction of Theorem 6.3. Unfortunately, this reasoning does not extend to k-out-of-N special-sound interactive proofs with $k > 2$: even though we still have that the t-fold parallel repetition of a k-out-of-N special-sound interactive proof is ℓ-out-of-N^t special-sound, but now with $\ell = (k-1)^t + 1$, this large increase in the special-soundness parameter ℓ renders the extractor, obtained via the generic reduction, inefficient. More precisely, the runtime of an ℓ-out-of-N^t special-sound interactive proof scales linearly in ℓ, and therefore exponentially in t for $\ell = (k-1)^t + 1$, unless $k = 2$. In case of multi-round interactive proofs, it is not even clear that the t-fold parallel repetition of a **k**-out-of-**N** special-sound $(2\mu+1)$-round interactive proof satisfies any meaningful notion of special-soundness.

We consider parallel repetition of interactive proofs in the context of decreasing the *knowledge error*. In Section 6.5.1, we show, based on a result from [CP15], that the t-fold parallel repetition of any *public-coin* interactive proof reduces the knowledge error from κ down to $\kappa^t + \nu$ for any noticeable term ν. This generic result is tight, since there are interactive proofs for which parallel repetition does not allow the knowledge error to be reduced down to a negligible function [DJM+12]. However, it is also suboptimal in that, when applied to a k-out-of-N special-sound protocol for instance, it does not give the knowledge error $\mathrm{Er}(k;N)^t$ that one expects (and that one should get when $k = 2$) and, worse, the knowledge error remains nonnegligible.

For this reason, in Section 6.5.2, we restrict the analysis to k-out-of-N special-sound Σ-protocols, i.e., 3-round interactive proofs, and derive a *strong* parallel repetition result. Here, as usual in the general context of parallel repetition, the term "strong" means that the figure of merit κ, here the knowledge error, drops from κ to κ^t under a t-fold parallel repetition. In Section 6.5.3, we generalize this result to **k**-out-of-**N** special-sound *multi-round* interactive proofs. Finally, in Section 6.5.4, we consider the more general case of s-out-of-t threshold parallel repetition, where the verifier accepts if s-out-of-t instantiations of the basic interactive proof are accepting. Threshold parallel repetition allows both the knowledge and the completeness error to be reduced simultaneously.

6.5.1 A Generic but Suboptimal Solution

In this section, we establish a *weak* parallel repetition theorem. We write $\Pi^t = (\mathcal{P}^t, \mathcal{V}^t)$ for the t-fold parallel repetition of an interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$, which runs t instances of Π in parallel and the verifier $\mathcal{V}^t$ accepts if all the parallel instances are accepted. Then we show that, if Π is public-coin and knowledge sound with knowledge error κ, Π^t has knowledge error $\kappa^t + \nu$ for any noticeable ν. The result is weak in that it does not reduce the knowledge error from κ down to κ^t. However, it is generically applicable to any *public-coin* interactive proof.

Our main building block is a result by Chung and Pass [CP15] summarized in Theorem 6.5. This theorem shows the existence of a prover $\mathfrak{P}$ that, given input x and oracle access to a dishonest prover $\mathcal{P}^*$ attacking interactive proof $\Pi^t = (\mathcal{P}^t, \mathcal{V}^t)$, succeeds in convincing $\mathcal{V}$ with probability approximately $\epsilon(x, \mathcal{P}^*)^{1/t}$, where $\epsilon(x, \mathcal{P}^*)$ is the probability that the prover $\mathcal{P}^*$ successfully convinces verifier $\mathcal{V}^t$ on input x. This theorem immediately shows that the t-fold parallel repetition reduces the soundness error from σ down to approximately σ^t. Subsequently, in Theorem 6.6, we show how this result can be used to derive our parallel repetition theorem for reducing the knowledge error instead of the soundness error.

Theorem 6.5 (Theorem 2 of [CP15])**.** *Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a public-coin interactive proof for relation $\mathfrak{R}$. Let $t \in \mathbb{N}$, and let $\Pi^t = (\mathcal{P}^t, \mathcal{V}^t)$ be the t-fold parallel repetition of Π. Then there exists an oracle algorithm $\mathfrak{P}^{(\cdot)}$ such that for every $\xi, \delta \colon \{0,1\}^* \to (0,1)$, every $x \in \{0,1\}^*$, and every PPT prover $\mathcal{P}^*$, it holds that if*

$$\Pr\left[\left(\mathcal{P}^*, \mathcal{V}^t\right)(x) = \textsf{accept}\right] \geq \underbrace{\left(1 + \xi(x)\right) \cdot \delta(x)^t}_{\epsilon(x) :=},$$

then

$$\Pr\left[\left(\mathfrak{P}^{\mathcal{P}^*}, \mathcal{V}\right)(x) = \textsf{accept}\right] \geq \delta(x)\,.$$

Furthermore, $\mathfrak{P}^{\mathcal{P}^*}$ runs in time $\operatorname{poly}\left(|x|, t, \xi(x)^{-1}, \epsilon(x)^{-1}, (1 - \delta(x))^{-1}\right)$.

Theorem 6.5 was actually established specifically in the context of decreasing the soundness error of *computationally sound* interactive proofs. Recall that computational soundness only requires the success probability of *computationally bounded* dishonest provers to be smaller than the soundness error. For this reason, in contrast to the case of unconditional soundness, analyzing the parallel repetition of computationally sound interactive proofs is significantly more complicated. More precisely, from Theorem 6.5 it follows by contraposition that parallel repetition decreases the soundness error; given a prover $\mathcal{P}^*$ attacking the parallel repetition Π^t with success probability ϵ, an oracle prover $\mathfrak{P}^{(\cdot)}$ attacking the basic interactive proof Π with success probability approximately $\epsilon^{1/t}$ is constructed. Applying this argument in the context of computational soundness requires the oracle prover $\mathfrak{P}^{(\cdot)}$ to be *efficient*. In the context of unconditional soundness the oracle prover $\mathfrak{P}^{(\cdot)}$ is not required to be efficient. In fact, it is well known that the t-fold parallel repetition of an unconditionally sound public-coin interactive proof decreases the soundness error σ down to σ^t [BGG90; Gol98], i.e., for these protocols there exists a *strong* parallel repetition result.

By contrast, both the unconditional and computational variant of *knowledge soundness* require the existence of an *efficient* extractor. Therefore, restricting to either of the two variations does not simplify the analysis. However, in the following theorem we show that, using the above oracle prover $\mathfrak{P}^{(\cdot)}$, a knowledge extractor for the parallel repetition Π^t can be constructed. The extractor invokes $\mathfrak{P}^{(\cdot)}$ a polynomial number of times and is therefore efficient as long as $\mathfrak{P}^{(\cdot)}$ is efficient. Altogether, Theorem 6.6 shows that t-fold parallel repetition decreases

the knowledge error from κ down to $\kappa^t + \nu$ for any noticeable ν. However, we cannot show that Π^t has negligible knowledge error for any fixed negligible function, because the running time of $\mathfrak{P}^{\mathcal{P}^*}$ scales with $\epsilon(x, \mathcal{P}^*)^{-1}$.

While it might seem that this barrier is rather an artifact of the proof technique of [CP15] on which we build, it was shown by [DJM+12] that Theorem 6.5 is tight when considering soundness amplification of interactive proofs in general. More precisely, based on some cryptographic assumptions they showed that, for some protocols, parallel repetition does not amplify security beyond negligible, meaning that for any negligible function η one can find an instantiation that when starting with nonnegligible soundness error, the protocol can always be broken with probability $\eta(x)$, no matter how many parallel repetitions one runs.

Theorem 6.6 (Generic Parallel Repetition Theorem)**.** *Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a public-coin interactive proof for relation $\mathfrak{R}$ that is knowledge sound with knowledge error $\kappa\colon \mathbb{N} \to [0,1]$. Let $\nu\colon \mathbb{N} \to (0,1)$ be an arbitrary noticeable function. Then, the t-fold parallel repetition $\Pi^t = (\mathcal{P}^t, \mathcal{V}^t)$ of Π is knowledge sound with knowledge error $\kappa' = \kappa^t + \nu$.*

Proof. We construct a knowledge extractor $\mathcal{E}^t$ for $\Pi^t = (\mathcal{P}^t, \mathcal{V}^t)$ as follows. Let $\mathcal{P}^*$ be some (potentially dishonest) prover attacking Π with success probability $\epsilon(x, P^*)$ on input x. Let $\xi\colon \mathbb{N} \to (0,1)$ be such that $\xi(n) = \nu(n)/\kappa(n)^t$ for all $n \in \mathbb{N}$. Then, by Theorem 6.5, there exists an oracle prover $\mathfrak{P}^{(\cdot)}$ such that

$$\epsilon(x, \mathfrak{P}^{\mathcal{P}^*}) = \Pr\big((\mathfrak{P}^{\mathcal{P}^*}, \mathcal{V})(x) = \mathsf{accept}\big) \geq \delta(x)\,,$$

where

$$\delta(x) = \left(\frac{\epsilon(x, \mathcal{P}^*)}{1 + \xi(|x|)}\right)^{1/t}.$$

By assumption $\Pi = (\mathcal{P}, \mathcal{V})$ is knowledge sound with knowledge error κ and, therefore, there exists a knowledge extractor $\mathcal{E}$ for Π. We define $\mathcal{E}^t$ as the algorithm that executes the knowledge extractor $\mathcal{E}$ on the prover $\mathfrak{P}^{\mathcal{P}^*}$.

Let us now analyze the expected runtime and success probability of extractor $\mathcal{E}^t$ for interactive proof Π^t. Recall that, in order to prove knowledge soundness with knowledge error $\kappa'(|x|)$, it is enough to consider statements $x \in \{0,1\}^*$ with $\epsilon(x, \mathcal{P}^*) > \kappa'(|x|)$ (Remark 2.4). Therefore, it is left to show that the following holds:

Claim. If $\epsilon(x, \mathcal{P}^*) > \kappa'(|x|)$, then the extractor $\mathcal{E}^t$ as defined above runs in an expected polynomial number of steps and there exists a positive polynomial q such that $\mathcal{E}^t$ is successful with probability at least $(\epsilon(x, \mathcal{P}^*) - \kappa'(|x|))/q(|x|)$.

Expected Runtime. We start proving the claim by showing that $\mathfrak{P}^{\mathcal{P}^*}$ runs in an expected polynomial number of steps. By Theorem 6.5, we have that the runtime of $\mathfrak{P}^{\mathcal{P}^*}$ is in $\mathrm{poly}(|x|, t, \xi(|x|)^{-1}, \epsilon(x, \mathcal{P}^*)^{-1}, (1 - \delta(x))^{-1})$. It holds that

$$\xi(|x|) = \nu(|x|)/\kappa(|x|)^t \geq \nu(|x|)$$

and $\epsilon(x, \mathcal{P}^*) > \kappa'(|x|) \geq \nu(|x|)$ and therefore also $\xi(|x|)^{-1}, \epsilon(x, \mathcal{P}^*)^{-1} \leq \mathrm{poly}(|x|)$. It is left to show that $1 - \delta(x)$ is noticeable. Via the Taylor approximation of the

function $f(a) = a^{1/t}$ in $a = 1$, we obtain

$$\delta(x) = \left(\frac{\epsilon(x,\mathcal{P}^*)}{1+\xi(|x|)}\right)^{1/t} \leq 1 - \frac{1}{t}\left(1 - \frac{\epsilon(x,\mathcal{P}^*)}{1+\xi(|x|)}\right).$$

Therefore, we also have

$$1-\delta(x) \geq \frac{1}{t}\left(1 - \frac{\epsilon(x,\mathcal{P}^*)}{1+\xi(|x|)}\right) = \frac{1}{t}\left(\frac{1+\xi(|x|)-\epsilon(x,\mathcal{P}^*)}{1+\xi(|x|)}\right) \overset{\xi,\epsilon\leq 1}{\geq} \frac{\xi(|x|)}{2t} \geq \frac{\nu(|x|)}{2t},$$

as required.

Next, note that if $\epsilon(x,\mathcal{P}^*) > \kappa'(|x|)$, then $\delta(x) > \kappa(|x|)$. This is a simple consequence of the definition of $\xi(|x|)$ and $\delta(x)$, because

$$\epsilon(x,\mathcal{P}^*) > \kappa(|x|)^t + \nu(|x|) = \kappa(|x|)^t\big(1+\xi(|x|)\big)$$

implies $\delta(x) = (\epsilon(x,\mathcal{P}^*)/(1+\xi(|x|))^{1/t} > \kappa(|x|)$ as required.

Altogether, this shows that if $\epsilon(x,\mathcal{P}^*) > \kappa'(|x|)$, then $\mathcal{E}^t$ runs in an expected polynomial number of steps.

Success Probability. Let us now consider the success probability of $\mathcal{E}^t$. By definition of the knowledge extractor $\mathcal{E}$, there exists a positive polynomial p such that $\mathcal{E}^t$ outputs a witness $w \in \mathfrak{R}(x)$ with probability at least

$$\frac{\delta(x) - \kappa(|x|)}{p(|x|)}.$$

Therefore, it is left to show that if $\epsilon(x,\mathcal{P}^*) > \kappa'(|x|)$, there exists a positive polynomial q such that

$$\frac{\delta(x) - \kappa(|x|)}{p(|x|)} \geq \frac{\epsilon(x,\mathcal{P}^*) - \kappa(|x|)^t - \nu(|x|)}{q(|x|)}.$$

To express the success probability of $\mathcal{E}^t$ in terms of $\epsilon(x,\mathcal{P}^*)$, let us define the functions $f(a) = t(a^{1/t} - b)$ and $g(a) = a - b^t$, for $b \in [0,1]$. Observe that $f(a)$ is concave for $a \geq 0$. Moreover, $f(b^t) = g(b^t) = 0$ and $f(1) = t(1-b) \geq (1-b)\sum_{i=0}^{t-1} b^i = g(1)$. Hence $\max(f(a),0) \geq g(a)$ for all $a \in [0,1]$.

From this inequality we have that whenever $\delta(x) > \kappa(|x|)$, it holds that

$$\begin{aligned}
\delta(x) - \kappa(|x|) &= \max(\delta(x) - \kappa(|x|), 0) \\
&= \max\left(\left(\frac{\epsilon(x,\mathcal{P}^*)}{(1+\xi(|x|))}\right)^{1/t} - \kappa(|x|), 0\right) \\
&\geq \frac{1}{t}\left(\frac{\epsilon(x,\mathcal{P}^*)}{(1+\xi(|x|))} - \kappa(|x|)^t\right) \\
&= \frac{1}{t(1+\xi(|x|))}\left(\epsilon(x,\mathcal{P}^*) - (1+\xi(|x|))\kappa(|x|)^t\right) \\
&\geq \frac{1}{2t}\left(\epsilon(x,\mathcal{P}^*) - \kappa(|x|)^t - \nu(|x|)\right).
\end{aligned}$$

Thus, choosing $q(|x|) = 2t \cdot p(|x|)$ yields the desired result, which proves the claim and completes the proof of the theorem. □

Remark 6.2. Let M be the total size of the challenge set, i.e., $M = \prod_{i=1}^{\mu} |\mathcal{C}_i|$ where the i^{th} challenge is sampled from challenge set $\mathcal{C}_i$. If M is polynomial in the size of the input x, the analysis can be simplified significantly. In this case the knowledge extractor can query all possible challenges and still run in polynomial time. A parallel repetition theorem then follows by a simple counting argument. This is the approach in the analysis of the 5-round $(2,2)$-out-of-$(q,2)$ special-sound signature scheme MQDSS [SSH11; CHR+16]. It is much more challenging to construct efficient knowledge extractors when M is not polynomial in $|x|$.

6.5.2 Parallel Repetition of k-out-of-n Special-Sound Σ-Protocols

Let us now restrict ourselves to *special-sound* interactive proofs. To simplify the exposition, we start with the simpler case of Σ-protocols; the general case of multi-round protocols will then be treated in the subsequent section. Thus, for the remainder of this section, we consider a k-out-of-N special-sound interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ with challenge set $\mathcal{C}$ of cardinality $N \geq k$. We have seen in Section 6.4 that Π is knowledge sound with knowledge error $\mathrm{Er}(k; N) = (k-1)/N$. In this section, we prove that the t-fold parallel repetition $\Pi^t = (\mathcal{P}^t, \mathcal{V}^t)$ of Π is then again knowledge sound, but now with knowledge error $\mathrm{Er}(k; N)^t$, which is optimal. Thus, we show what is sometimes referred to as *strong* parallel repetition, meaning that the figure of merit decreases with power t under parallel repetition. This is well known to hold for special-sound Σ-protocols, i.e., for $k = 2$, but was open for general k.

The standard way to reason about parallel repetition for the special case $k = 2$ uses the fact that Π^t is ℓ-out-of-N^t special-sound with $\ell = (k-1)^t + 1$. However, this reasoning does not apply in general, because ℓ grows exponentially in t for $k > 2$. Instead, our result crucially depends on the fact that Π^t is the t-fold parallel repetition of a k-out-of-N special-sound protocol Π^t. In this section, we first construct a novel extraction algorithm for k-out-of-N special-sound interactive proofs Π, thereby reproving that k-out-of-N special-soundness implies knowledge soundness (Theorem 6.3). Subsequently, we show how this extraction algorithm can be used to deduce a strong parallel repetition result for Π^t. In Section 6.5.3, we then extend our results to multi-round interactive proofs.

On a high level, the crucial ingredient in our analyses is to introduce and work with a more "fine-grained" notion of success probability of a dishonest prover, as explained below.

Knowledge Soundness of a Single Invocation

Consider a dishonest deterministic prover $\mathcal{P}^*$ attacking the considered k-out-of-N special-sound interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ on public input x. The goal of the extractor is to run $\mathcal{P}^*$ sufficiently many times so as to obtain k correct answers $z_1, \ldots, z_k$ for k pairwise distinct challenges $c_1, \ldots, c_k \in \mathcal{C}$. By the special-soundness property, a witness $w \in \mathfrak{R}(x)$ can be computed efficiently from the resulting set of protocol transcripts. Recall that, without loss of generality, we may assume $\mathcal{P}^*$ to be deterministic and therefore its first message a to be fixed (Remark 2.3).

The crucial question is how often $\mathcal{P}^*$ needs to be invoked, and thus what is

the (expected) running time of the extractor. Alternatively, towards satisfying Definition 2.27, we would like to have an extractor that runs in a fixed (expected) polynomial time, but may fail with some probability. It is quite clear that in both cases the figure of merit (i.e., the running time in the former and the success probability in the latter) depends on the *success probability* $\epsilon(x, \mathcal{P}^*)$ of $\mathcal{P}^*$ on input x; for instance, if $\epsilon(x, \mathcal{P}^*)$ is below the knowledge error $\mathrm{Er}(k; N)$ then we cannot expect extraction to work in general. However, a crucial observation is that for a given dishonest prover $\mathcal{P}^*$, its success probability $\epsilon(x, \mathcal{P}^*)$ does actually not characterize (very well) whether extraction is possible or not: if in a special-sound Σ-protocol $\mathcal{P}^*$ provides the correct response with probability $\epsilon(x, \mathcal{P}^*)$ (and fails to do so with probability $1 - \epsilon(x, \mathcal{P}^*)$) *for every* possible choice of the challenge, then extraction is still possible even when $\epsilon(x, \mathcal{P}^*) < \mathrm{Er}(k; N)$ (but not negligible), simply by trying sufficiently many times for two distinct challenges. Below, we will identify an alternative, in some sense more fine-grained, "quality measure" of $\mathcal{P}^*$, and we show that this measure does characterize when extraction is possible. This will be helpful when it comes to more complicated settings, like a *parallel repetition*, or a *multi-round* protocol, or, ultimately, a *parallel repetition* of a *multi-round* protocol.

As before, we will state and prove our core technical results in a more abstract language, i.e., we consider an arbitrary probabilistic algorithm $\mathcal{A} \colon \mathcal{C} \to \{0,1\}^*$ and an arbitrary verification function $V \colon \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. The success probability of $\mathcal{A}$ is denoted as

$$\epsilon(\mathcal{A}) := \Pr\big(V\big(C, \mathcal{A}(C)\big) = 1\big)\,,$$

where C is uniformly random in $\mathcal{C}$. The obvious instantiation of $\mathcal{A}$ is given by a deterministic dishonest prover $\mathcal{P}^*$ attacking the considered k-out-of-N special-sound Σ-protocol Π on input x.

Given oracle access to $\mathcal{A}$, the goal of the extractor is to find correct responses $y_1, \ldots, y_k$ for k pairwise distinct challenges $c_1, \ldots, c_k \in \mathcal{C}$, i.e., such that $V(c_i, y_i) = 1$ for all i. In Section 6.4.1, we showed how to do this in expected polynomial time and with success probability at least

$$\epsilon(\mathcal{A}) - \mathrm{Er}(k; N)\,.$$

Below we follow a different approach and show that a more fine-grained measure, capturing how well extraction can be done, is

$$\delta_k(\mathcal{A}) := \min_{S \subseteq \mathcal{C} : |S| < k} \Pr\big(V(C, \mathcal{A}(C)) = 1 \mid C \notin S\big)\,.$$

More precisely, we argue existence of an extraction algorithm $\mathcal{E}^{\mathcal{A}}$ with oracle access to $\mathcal{A}$, that runs in expected polynomial time and succeeds with probability at least $\delta_k(\mathcal{A})/k$.

Lemma 6.7 (Extraction Algorithm - Σ-protocols)**.** *Let $k \in \mathbb{N}$, $\mathcal{C}$ a finite set with cardinality $N \geq k$ and let $V \colon \mathcal{C} \times \{0,1\}^* \to \{0,1\}$. Then there exists an oracle algorithm $\mathcal{E}$ with the following properties: The algorithm $\mathcal{E}^{\mathcal{A}}$, given oracle access to a (probabilistic) algorithm $\mathcal{A} \colon \mathcal{C} \to \{0,1\}^*$, requires an expected number of at most $2k - 1$ queries to $\mathcal{A}$ and, with probability at least $\delta_k(\mathcal{A})/k$, it outputs k pairs*

$(c_1, y_1), (c_2, y_2), \ldots, (c_k, y_k) \in \mathcal{C} \times \{0,1\}^*$ *with* $V(c_i, y_i) = 1$ *for all* i *and* $c_i \neq c_j$ *for all* $i \neq j$.

Figure 6.3: Recursive Expected Polynomial Time Extractor $\mathcal{E}_k(\mathcal{D})$.

Parameters: $k \in \mathbb{N}$ and $\mathcal{D} \subseteq \mathcal{C}$.
Oracle access to: Algorithm $\mathcal{A}\colon \mathcal{C} \to \{0,1\}^*$ and verification function $V\colon \mathcal{C} \times \{0,1\}^* \to \{0,1\}$.

- Sample $c_1 \in \mathcal{D}$ uniformly at random and evaluate $y_1 \leftarrow \mathcal{A}(c_1)$.
- If $V(c_1, y_1) = 0$, abort.
- If $V(c_1, y_1) = 1$ and $k = 1$, output $(c_1, y_1) \in \mathcal{D} \times \{0,1\}^*$.
- Else, set $\textsc{coin} = 0$ and repeat
  - run $\mathcal{E}_{k-1}(\mathcal{D} \setminus \{c_1\})$;
  - set $\textsc{coin} \leftarrow V\big(d, \mathcal{A}(d)\big)$ for $d \in \mathcal{D}$ sampled uniformly at random;

  until either $\mathcal{E}_{k-1}(\mathcal{D} \setminus \{c_1\})$ outputs $k-1$ pairs $(c_2, y_2), \ldots, (c_k, y_k)$ with $V(c_i, y_i) = 1$ for all i have been found or until $\textsc{coin} = 1$.

Output: In the former case, output k pairs $(c_1, y_1), \ldots, (c_k, y_k) \in \mathcal{D} \times \{0,1\}^*$ with $V(c_i, y_i) = 1$ for all i and $c_i \neq c_j$ for all $i \neq j$.

Proof. The extraction algorithm is defined recursively over k. For this reason, we add a subscript k and write $\mathcal{E}_k^{\mathcal{A}}$ for the extraction algorithm that, given oracle access to $\mathcal{A}$, aims to output k pairs (c_i, y_i). In this proof, we also make the set $\mathcal{D} \subseteq \mathcal{C}$, from which the extractor samples the challenges c_i, explicit by writing $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$. This allows the extractor to be deployed on subsets $\mathcal{D}$ of the full challenge set $\mathcal{C}$, i.e., extractor $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ aims to output k pairs (c_i, y_i) with pairwise distinct challenges $c_i \in \mathcal{D}$ and $V(c_i, y_i) = 1$ for all i. When writing $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ we will always implicitly assume that $|\mathcal{D}| \geq k$. Accordingly, we also write

$$\epsilon^V(\mathcal{A}, \mathcal{D}) := \Pr\big(V(C, \mathcal{A}(C)) = 1\big)\,,$$
$$\delta_k^V(\mathcal{A}, \mathcal{D}) := \min_{S \subseteq \mathcal{D} : |S| < k} \Pr\big(V(C, \mathcal{A}(C)) = 1 \mid C \notin S\big)\,,$$

where the probability space is defined over of the randomness of $\mathcal{A}$ and the random variable C being uniformly random in $\mathcal{D} \subseteq \mathcal{C}$. If V is clear from context we sometimes simply write $\epsilon(\mathcal{A}, \mathcal{D})$ and $\delta_k(\mathcal{A}, \mathcal{D})$. Note that for all $k \geq 1$,

$$\delta_{k+1}(\mathcal{A}, \mathcal{D}) \leq \delta_k(\mathcal{A}, \mathcal{D}) \leq \delta_1(\mathcal{A}, \mathcal{D}) = \epsilon(\mathcal{A}, \mathcal{D})\,.$$

Let us now define the extraction algorithm. The extraction algorithm is defined recursively over k and also described in Figure 6.3. Let $\mathcal{D} \subseteq \mathcal{C}$ be an arbitrary subset with cardinality at least k. For $k = 1$, the extractor $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$ simply samples a challenge $c_1 \in \mathcal{D}$ uniformly at random and computes $y_1 \leftarrow \mathcal{A}(c_1)$. If

$V(c_1, y_1) = 0$, it outputs $\perp$ and aborts. Otherwise, if $V(c_1, y_1) = 1$, it successfully outputs (c_1, y_1). This extractor queries $\mathcal{A}$ once and it succeeds with probability $\epsilon(\mathcal{A}, \mathcal{D}) = \delta_1(\mathcal{A}, \mathcal{D})$.

For $k > 1$, the extractor $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ first runs the extractor $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$. If $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$ fails, $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ outputs $\perp$ and aborts; otherwise, if $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$ succeeds to output a pair (c_1, y_1), $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ proceeds as follows. It defines the set $\mathcal{D}' = \mathcal{D} \setminus \{c_1\}$ and runs $\mathcal{E}_{k-1}^{\mathcal{A}}(\mathcal{D}')$. If $\mathcal{E}_{k-1}^{\mathcal{A}}(\mathcal{D}')$ succeeds to output $k-1$ pairs $(c_2, y_2), \ldots (c_k, y_k)$, $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ successfully outputs k pairs $(c_1, y_1), \ldots, (c_k, y_k)$. On the other hand, if $\mathcal{E}_{k-1}^{\mathcal{A}}(\mathcal{D}')$ fails, $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ tosses a coin that returns heads with probability $\epsilon(\mathcal{A}, \mathcal{D})$. This coin can be implemented by running $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$, i.e., sampling a random challenge $d \leftarrow \mathcal{D}$ and evaluating $V\big(d, \mathcal{A}(d)\big)$. If the coin returns heads, $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ outputs $\perp$ and aborts. If the coin returns tails, $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ runs $\mathcal{E}_{k-1}^{\mathcal{A}}(\mathcal{D}')$ once more and performs the same steps as before. The algorithm proceeds in this manner until either it has successfully found k pairs (c_i, y_i) or until the coin returns heads.

Let us now analyze the success probability and the expected number of $\mathcal{A}$-queries of this algorithm.

Success Probability. We aim to show that, for all $k \in \mathbb{N}$ and for all $\mathcal{D} \subseteq \mathcal{C}$ with $|\mathcal{D}| \geq k$, the success probability $\Delta_k(\mathcal{D})$ of the extractor $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ satisfies

$$\Delta_k(\mathcal{D}) \geq \delta_k(\mathcal{A}, \mathcal{D})/k\,.$$

The analysis goes by induction. Since $\Delta_1(\mathcal{D}) = \epsilon(\mathcal{A}, \mathcal{D}) = \delta_1(\mathcal{A}, \mathcal{D})/1$, the induction hypothesis is satisfied for the case $k = 1$.

Let us now consider $k > 1$ and assume that the induction hypothesis holds for $k' = k-1$ and all $\mathcal{D}'$ with $|\mathcal{D}'| \geq k-1$. We consider arbitrary $\mathcal{D} \subseteq \mathcal{C}$ with $|\mathcal{D}| \geq k$. Then, if in its first step $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ successfully runs extractor $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$ (outputting a pair (c_1, y_1) with $V(c_1, y_1) = 1$), it starts running two geometric experiments until one of them finishes. In the first geometric experiment the extractor aims to find an additional set of $k-1$ pairs (c_i, y_i) by running $\mathcal{E}_{k-1}^{\mathcal{A}}(\mathcal{D}')$, where $\mathcal{D}' = \mathcal{D} \setminus \{c_1\}$. By the induction hypothesis, the parameter p of this geometric distribution satisfies

$$p := \Delta_{k-1}(\mathcal{D}') \geq \delta_{k-1}(\mathcal{A}, \mathcal{D}')/(k-1) \geq \delta_k(\mathcal{A}, \mathcal{D})/(k-1)\,.$$

In the second geometric experiment the extractor tosses a coin that returns heads with probability

$$q := \epsilon(\mathcal{A}, \mathcal{D})\,.$$

The second step of the extractor succeeds if the second geometric experiment does not finish before the first, and so by Lemma 2.2 this probability is lower bounded by

$$\begin{aligned}
\Pr\big(\mathrm{Geo}(p) \leq \mathrm{Geo}(q)\big) &\geq \frac{p}{p+q} = \frac{\Delta_{k-1}(\mathcal{D}')}{\Delta_{k-1}(\mathcal{D}') + \epsilon(\mathcal{A}, \mathcal{D})} \\
&\geq \frac{\delta_k(\mathcal{A}, \mathcal{D})/(k-1)}{\delta_k(\mathcal{A}, \mathcal{D})/(k-1) + \epsilon(\mathcal{A}, \mathcal{D})} \\
&\geq \frac{\delta_k(\mathcal{A}, \mathcal{D})/(k-1)}{\epsilon(\mathcal{A}, \mathcal{D})/(k-1) + \epsilon(\mathcal{A}, \mathcal{D})} \\
&= \frac{\delta_k(\mathcal{A}, \mathcal{D})}{k \cdot \epsilon(\mathcal{A}, \mathcal{D})}\,,
\end{aligned}$$

where the second inequality follows from the monotonicity of the function $x \mapsto \frac{x}{x+q}$. Since the first step of the extractor succeeds with probability $\epsilon(\mathcal{A}, \mathcal{D})$, it follows that $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ succeeds with probability at least $\delta_k(\mathcal{A}, \mathcal{D})/k$.

Therefore, by induction it follows that for all k and $\mathcal{D}$ with $|\mathcal{D}| \geq k$, the extractor $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ succeeds with probability at least $\delta_k(\mathcal{A}, \mathcal{D})/k$. In particular, the extractor $\mathcal{E}_k^{\mathcal{A}}(\mathcal{C})$ succeeds with probability at least $\delta_k(\mathcal{A})/k$, which proves that this extractor has the desired success probability.

Expected Number of $\mathcal{A}$-Queries. For $\mathcal{D} \subseteq \mathcal{C}$ with $|\mathcal{D}| \geq k$, we let $Q_k(\mathcal{D})$ be the expected number of $\mathcal{A}$-queries made by the extractor $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$. We will prove that $Q_k(\mathcal{D}) \leq 2k - 1$ for all $k \in \mathbb{N}$ and $\mathcal{D} \subseteq \mathcal{C}$ with $|\mathcal{D}| \geq k$. The proof of this claim goes by induction. First note that, since $Q_1(\mathcal{D}) = 1$ for all $\mathcal{D} \neq \emptyset$, this claim is clearly satisfied for the base case $k = 1$.

Let us now consider $k > 1$ and assume the claim is satisfied for $k' = k - 1$. Let $\mathcal{D} \subseteq \mathcal{C}$ be arbitrary with $|\mathcal{D}| \geq k$. Then $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ first runs $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$, which requires exactly $Q_1(\mathcal{D}) = 1$ query. Then with probability $\epsilon(\mathcal{A}, \mathcal{D})$ it continues to the second step. In each iteration of the second step $\mathcal{E}_k^{\mathcal{A}}(\mathcal{D})$ runs $\mathcal{E}_{k-1}^{\mathcal{A}}(\mathcal{D}')$, for some $D' \subseteq \mathcal{C}$ with $|\mathcal{D}'| \geq k-1$, and it tosses a coin by running $\mathcal{E}_1^{\mathcal{A}}(\mathcal{D})$. Therefore, each iteration requires an expected number of at most $Q_{k-1}(\mathcal{D}') + 1 \leq 2k - 2$ queries. Moreover, the expected number of tosses until the coin returns heads is $1/\epsilon(\mathcal{A}, \mathcal{D})$. Hence, the expected number of iterations in the second step of this extraction algorithm is at most $1/\epsilon(\mathcal{A}, \mathcal{D})$. It follows that

$$Q_k(\mathcal{D}) \leq 1 + \epsilon(\mathcal{A}, \mathcal{D}) \frac{1}{\epsilon(\mathcal{A}, \mathcal{D})}(2k - 2) = 2k - 1\,.$$

Here it is crucial that the above inequality holds for all $\mathcal{D} \subseteq \mathcal{C}$. This proves the claimed upper bound on the expected number of $\mathcal{A}$-queries and completes the proof of the lemma.

□

In the context of a deterministic dishonest prover $\mathcal{P}^*$ attacking a k-out-of-N special-sound protocol, we make the following observation. First, by basic probability theory and for any $S \subseteq \mathcal{C}$,

$$\begin{aligned} \Pr\big(V(C, \mathcal{A}(C)) = 1 \mid C \notin S\big) &= \frac{\Pr\big(V(C, \mathcal{A}(C)) = 1 \wedge C \notin S\big)}{\Pr\big(C \notin S\big)} \\ &\geq \frac{\Pr\big(V(C, \mathcal{A}(C)) = 1) - \Pr\big(C \in S\big)}{\Pr\big(C \notin S\big)}\,. \end{aligned} \tag{6.4}$$

Thus, extractor $\mathcal{E}^{\mathcal{A}}$ succeeds with positive probability as soon as $\epsilon(\mathcal{A}) > \Pr\big(C \in S\big)$ for every $S \subseteq \mathcal{C}$ with $|S| < k$. More precisely,

$$\Pr\big(\mathcal{E}^{\mathcal{A}} \neq \perp\big) \geq \frac{\delta_k(\mathcal{A})}{k} \geq \frac{\epsilon(\mathcal{A}) - \mathrm{Er}(k; N)}{k(1 - \mathrm{Er}(k; N))}\,, \tag{6.5}$$

where $\mathrm{Er}(k; N) = (k - 1)/N$.

This observation confirms that k-out-of-N special-soundness implies knowledge soundness with knowledge error $\mathrm{Er}(k; N)$, i.e., it provides an alternative proof

for Theorem 6.3. Hence, in comparison to $\epsilon(\mathcal{A})$, $\delta_k(\mathcal{A})$ is indeed a more fine-grained measure capturing how well extraction can be done.

Note that the extractor of Lemma 6.7 does not strictly outperform the extractor of Lemma 6.5. Namely, it behaves somewhat worse in the (expected) polynomial runtime, and also in the success probability when the measure $\delta_k(\mathcal{A})$ is bounded by $\epsilon(\mathcal{A}) - (k-1)/N$; the expected runtime is roughly a factor two larger and the success probability is roughly a factor k smaller. However, this is still sufficient for proving that k-out-of-N special-soundness tightly implies knowledge soundness. Moreover, by exploiting the definition of δ, as we show below, we can obtain an extractor for a *parallel repetition* of the considered interactive proof by running the extractor individually on each instance of the parallel repetition. Thus, our extractor is well suited to handle parallel repetitions of k-out-of-N special-sound Σ-protocols. Nevertheless, it remains an interesting problem whether our extractor can be improved to match up with the extractor from Lemma 6.5 while still giving rise to our parallel-repetition results.

Knowledge-Soundness of the Parallel Repetition

When moving to the t-fold parallel repetition $\Pi^t = (\mathcal{P}^t, \mathcal{V}^t)$ of the k-out-of-N special-sound Σ-protocol $\Pi = (\mathcal{P}, \mathcal{V})$, we consider an algorithm $\mathcal{A}$ that takes as input a row $(c^1, \ldots, c^t) \in \mathcal{C}^t$ of challenges[3] and outputs a string y, and the *success probability* of $\mathcal{A}$ is then defined as

$$\epsilon(\mathcal{A}) = \Pr\big(V(C^1, \ldots, C^t, \mathcal{A}(C^1, \ldots, C^t)) = 1\big)\,,$$

for some given $V\colon \mathcal{C}^t \times \{0,1\}^* \to \{0,1\}$ and where the C^j are understood to be independently and uniformly distributed over $\mathcal{C}$. We use superscripts to distinguish between the different parallel instantiations of basic Σ-protocol Π, so that later, when considering multi-round interactive proofs, the subscripts can be used to distinguish between the different rounds of the protocol.

The obvious instantiation of $\mathcal{A}$ is given by a deterministic prover $\mathcal{P}^*$ attacking the considered t-fold parallel repetition $\Pi^t = (\mathcal{P}^t, \mathcal{V}^t)$ of Π on input x. More precisely, on input $(c^1, \ldots, c^t)$, $\mathcal{A}$ runs $\mathcal{P}^*$ sending $(c^1, \ldots, c^t)$ as the challenges for the t repetitions of Π, and outputs $\mathcal{P}^*$'s (fixed) first messages $(a^1, \ldots, a^t)$ and its responses $(z^1, \ldots, z^t)$, and the function V is defined as the verification procedure of $\mathcal{V}^t$, which checks each repetition independently and accepts only if all are correct.

Such an $\mathcal{A}$ naturally induces t algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_t$ as considered above in the context of a single execution of a k-out-of-N special-sound protocol, taking *one* challenge as input: on input c^j, the algorithm $\mathcal{A}_j$ runs $y \leftarrow \mathcal{A}(c^1, \ldots, c^t)$ with c^i chosen uniformly at random from $\mathcal{C}$ for $i \neq j$, and outputs y along with the c^i's for $i \neq j$. We can thus run the extractor from above on all of the $\mathcal{A}_j$'s individually, with the goal being that at least one of them succeeds. We know that for each $\mathcal{A}_j$ individually, the extraction succeeds with probability

$$\delta_k^V(\mathcal{A}_j) = \min_{S^j \subseteq \mathcal{C} : |S^j| < k} \Pr\big(V(C^j, \mathcal{A}_j(C^j)) = 1 \mid C^j \notin S^j\big), \tag{6.6}$$

[3] There is no rigorous meaning in the list of challenges forming a *row*; it is merely that later we will also consider a *column* of challenges, which will then play a different *contextual* role.

where V is understood to appropriately reorder its inputs. The following lemma allows us to bound the probability that at least one of the extractors $\mathcal{E}^{\mathcal{A}_j}$ succeeds to produce k challenge-response pairs $\big((c^1,\dots,c^t),y\big)$ that all verify V and for which the k choices of c^j are all distinct for the considered j.

Lemma 6.8. *Let $k,t\in\mathbb{N}$, $\mathcal{C}$ a set with $|\mathcal{C}|=N\geq k$, $V\colon \mathcal{C}^t\times\{0,1\}^*\to\{0,1\}$, and $\mathcal{A}$ a (probabilistic) algorithm that takes as input a vector $(c^1,\dots,c^t)\in\mathcal{C}^t$ and outputs a string $y\in\{0,1\}^*$. Then*

$$\sum_{j=1}^{t}\delta_k^V(\mathcal{A}_j)\geq\frac{\epsilon(\mathcal{A})-\mathrm{Er}(k;N)^t}{1-\mathrm{Er}(k;N)}\,,$$

where $\mathrm{Er}(k;N)=(k-1)/N$.

Proof. Let Λ denote the event $V\big(C^1,\dots,C^t,\mathcal{A}(C^1,\dots,C^t)\big)=1$ and, for $1\leq j\leq t$, let S^j be such that it minimizes Equation 6.6. Moreover, let Γ_j denote the event $C^j\notin S^j$.

Without loss of generality, we may assume that $|S^j|=k-1$ for all j. Then, for all j,

$$\Pr(\Gamma_j)=\Pr(c^j\notin S^j)=1-\mathrm{Er}(k;N)\,.$$

Moreover, using elementary probability theory,

$$\begin{aligned}\sum_{j=1}^{t}\delta_k(\mathcal{A}_j)&=\sum_{j=1}^{t}\Pr\big(V(C^j,\mathcal{A}_j(C^j))=1\mid C^j\notin S^j\big)=\sum_{j=1}^{t}\Pr\big(\Lambda\mid\Gamma_j\big)\\&=\sum_{j=1}^{t}\frac{\Pr\big(\Lambda\wedge\Gamma_j\big)}{\Pr\big(\Gamma_j\big)}=\sum_{j=1}^{t}\frac{\Pr\big(\Lambda\wedge\Gamma_j\big)}{1-\mathrm{Er}(k;N)}\geq\frac{\Pr\big(\Lambda\wedge\exists\,j:\Gamma_j\big)}{1-\mathrm{Er}(k;N)}\\&\geq\frac{\Pr\big(\Lambda\big)-\Pr\big(\neg\Gamma_j\ \forall j\big)}{1-\mathrm{Er}(k;N)}=\frac{\epsilon(\mathcal{A})-\mathrm{Er}(k;N)^t}{1-\mathrm{Er}(k;N)}\,,\end{aligned}$$

which completes the proof. □

Lemma 6.8 readily provides a lower bound on $\max_i\delta_k^V(\mathcal{A}_i)\geq\sum_i\delta_k^V(\mathcal{A}_i)/t$, and thus on the success probability of the extractor. However, we can do slightly better. For this purpose, let $\Delta=\min\big(1,\sum_{i=1}^{t}\delta_k^V(\mathcal{A}_i)/k\big)$. Then, by the inequality of the arithmetic and the geometric mean,

$$\left(\prod_{i=1}^{t}\Big(1-\frac{\delta_k^V(\mathcal{A}_i)}{k}\Big)\right)^{1/t}\leq\frac{1}{t}\sum_{i=1}^{t}\Big(1-\frac{\delta_k^V(\mathcal{A}_i)}{k}\Big)\leq 1-\frac{\Delta}{t}\,.$$

Hence, the probability that at least one extractor $\mathcal{E}^{\mathcal{A}_i}$ succeeds equals

$$1-\prod_{i=1}^{t}\left(1-\frac{\delta_k^V(\mathcal{A}_i)}{k}\right)\geq 1-\left(1-\frac{\Delta}{t}\right)^t\geq 1-e^{-\Delta}\geq(1-e^{-1})\Delta\geq\frac{1}{2}\Delta\,,\quad(6.7)$$

where the third inequality uses that $(1 - e^{-x}) \geq (1 - e^{-1})x$ for all $0 \leq x \leq 1$, which is easily verified.[4] Hence, by Lemma 6.8, the probability of at least one of the extractors $\mathcal{E}^{\mathcal{A}_i}$ being successful is at least

$$\frac{\Delta}{2} \geq \frac{\epsilon^V(\mathcal{A}) - \mathrm{Er}(k;N)^t}{2k(1 - \mathrm{Er}(k;N))} .$$

From this it follows that the t-fold parallel repetition Π^t of a k-out-of-N special-sound protocol Π is knowledge sound with knowledge error $\mathrm{Er}(k;N)^t$, where $\mathrm{Er}(k;N) = (k-1)/N$ is the knowledge error of a single execution of Π. This strong parallel repetition result for k-out-of-N special-sound Σ-protocols is formalized in Theorem 6.7.

Theorem 6.7 (Parallel Repetition of k-Special-Sound Σ-Protocols)**.** *Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a k-out-of-N special-sound Σ-protocol. Let $\Pi^t = (\Pi^t, \mathcal{V}^t)$ be the t-fold parallel repetition of Π. Then Π^t is knowledge sound with knowledge error $\mathrm{Er}(k;N)^t$, where $\mathrm{Er}(k;N) = (k-1)/N$.*

Also here we have that the knowledge error $\mathrm{Er}(k;N)^t$ matches the trivial cheating probability, which succeeds if in each instance of the parallel repetition the challenge falls into a given set of size $k - 1$.

Remark 6.3. The above parallel repetition result (and also the generalization of Section 6.5.3) directly generalizes to the parallel composition of t *different* protocols or to the parallel composition of t different instances of the same protocol. In this case, the knowledge error will be the product of the individual knowledge errors.

6.5.3 Parallel Repetition of Multi-Round Interactive Proofs

We now consider the general case of multi-round interactive proofs. The line of reasoning is quite similar to that of 3-round protocols, but with an appropriately adjusted definition of δ. So, for the remainder of this section, we consider a $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound $(2\mu+1)$-round interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$, where the verifier samples its i-th challenge uniformly at random from a finite set $\mathcal{C}_i$ for $1 \leq i \leq \mu$. Eventually, we want to analyze its t-fold parallel repetition $\Pi^t = (\mathcal{P}^t, \mathcal{V}^t)$, but again we first consider a single invocation.

Knowledge Soundness of a Single Invocation

Similar to Section 6.4.2, we consider a probabilistic algorithm $\mathcal{A}$ that takes as input a vector $(c_1, \ldots, c_\mu) \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$ of challenges and outputs a string y, and we consider a function

$$V \colon \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu \times \{0,1\}^* \to \{0,1\} .$$

As before, the success probability of $\mathcal{A}$ is defined as

$$\epsilon^V(\mathcal{A}) := \Pr\big(V(C, \mathcal{A}(C)) = 1\big) ,$$

[4] For instance by observing that the two sides are equal for $x = 0$ and $x = 1$, and that the left hand side is a concave function while the right hand side is linear.

where $C = (C_1, \ldots, C_\mu)$ is uniformly random in $\mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$. The obvious instantiation of $\mathcal{A}$ is a deterministic prover $\mathcal{P}^*$ attacking the considered protocol. The goal of the extractor is to find correct responses for a $\mathbf{k}$-tree of challenges (Definition 2.33), where $\mathbf{k} = (k_1, \ldots, k_\mu)$. Generalizing the case of ordinary Σ-protocols, i.e., 3-round interactive proofs, the figure of merit here is

$$\delta^V_{\mathbf{k}}(\mathcal{A}) := \min_{S_1, S_2(\cdot), \ldots, S_\mu(\cdot)} \Pr\left(\Lambda \,\middle|\, \begin{array}{r} C_1 \notin S_1 \wedge C_2 \notin S_2(C_1) \wedge \cdots \\ \cdots \wedge C_\mu \notin S_\mu(C_1, \ldots, C_{\mu-1}) \end{array}\right), \tag{6.8}$$

where Λ denotes the event $V(C, \mathcal{A}(C)) = 1$ and the minimum is over all sets $S_1 \in \mathcal{C}_1|_{<k_1}$, and over all functions $S_2 \colon \mathcal{C}_1 \to \mathcal{C}_2|_{<k_2}$, $S_3 \colon \mathcal{C}_1 \times \mathcal{C}_2 \to \mathcal{C}_3|_{<k_3}$, etc. Here for any set $\mathcal{C}$ and $k \in \mathbb{N}$, $\mathcal{C}|_{<k}$ denotes the set of subsets of $\mathcal{C}$ with cardinality smaller than k.

Indeed, the following lemma shows that there exists an expected polynomial time extractor $\mathcal{E}^{\mathcal{A}}$ with oracle access to $\mathcal{A}$ that, with probability at least $\delta^V_{\mathbf{k}}(\mathcal{A}) / \prod_{i=1}^{\mu} k_i$, succeeds to extract correct responses for a $\mathbf{k}$-tree of challenges. Exploiting the abstract notation of Lemma 6.7, the proof of this lemma follows by induction over the number of challenges μ sent by the verifier. In particular, the extractor of the following lemma follows the same recursive approach as the one in Lemma 6.6, where we also considered knowledge extraction for multi-round interactive proofs. However, instead of Lemma 6.5, here we apply Lemma 6.7 for the base case of 3-round Σ-protocols. Subsequently, we will show that this adaptation allows us to handle parallel repetitions of multi-round interactive proofs.

Lemma 6.9 (Multi-Round Extraction Algorithm)**.** *Let $\mathbf{k} = (k_1, \ldots, k_\mu)$, $\mathbf{N} = (N_1, \ldots, N_\mu) \in \mathbb{N}^\mu$, $K = \prod_{i=1}^{\mu} k_i$, $\mathcal{C}_1, \ldots, \mathcal{C}_\mu$ finite sets $\mathcal{C}_i$ with cardinality $N_i \geq k_i$ and let $V \colon \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu \times \{0,1\}^* \to \{0,1\}$. Then there exists an oracle algorithm $\mathcal{E}$ with the following properties: The algorithm $\mathcal{E}^{\mathcal{A}}$, given oracle access to a (probabilistic) algorithm $\mathcal{A} \colon \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu \to \{0,1\}^*$, requires an expected number of at most $2^\mu \cdot K$ queries to $\mathcal{A}$ and, with probability at least $\delta^V_{\mathbf{k}}(\mathcal{A})/K$, outputs K pairs $(\mathbf{c}_1, y_1), \ldots, (\mathbf{c}_K, y_K) \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu \times \{0,1\}^*$ with $V(\mathbf{c}_i, y_i) = 1$ for all i and such that the vectors $\mathbf{c}_i \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$ form a $\mathbf{k}$-tree.*

Proof. The proof goes by induction on μ. For the base case $\mu = 1$, the lemma directly follows from Lemma 6.7. So let us assume the lemma holds for $\mu' = \mu - 1$. Then, for any $c \in \mathcal{C}_1$, let $\mathcal{A}_c$ be the algorithm that takes as input a vector $(c_2, \ldots, c_\mu) \in \mathcal{C}_2 \times \cdots \times \mathcal{C}_\mu$ and runs $\mathcal{A}(c, c_2, \ldots, c_\mu)$. The function V_c is defined accordingly, i.e.,

$$V_c \colon \mathcal{C}_2 \times \cdots \times \mathcal{C}_\mu \times \{0,1\}^* \to \{0,1\}, \quad (\mathbf{c}, y) \mapsto V(c, \mathbf{c}, y)\,.$$

Moreover, let $\mathbf{k}' = (k_2, \ldots, k_\mu), \mathbf{N}' = (N_2, \ldots, N_\mu) \in \mathbb{N}^{\mu-1}$ and $K' = \prod_{i=2}^{\mu} k_i$. By the induction hypothesis, there exists an algorithm $\mathcal{E}^{\mathcal{A}_c}_{\mu-1}$ that aims to output a set $\mathcal{Y}$ of K' pairs $(\mathbf{c}_1, y_1), \ldots, (\mathbf{c}_{K'}, y_{K'}) \in \mathcal{C}_2 \times \cdots \times \mathcal{C}_\mu \times \{0,1\}^*$ with $V(c, \mathbf{c}_i, y_i) = 1$ for all i and such that the vectors $\mathbf{c}_i \in \mathcal{C}_2 \times \cdots \times \mathcal{C}_\mu$ form a $\mathbf{k}'$-tree of challenge vectors. Moreover, $\mathcal{E}^{\mathcal{A}_c}_{\mu-1}$ requires an expected number of at most $2^{\mu-1} \cdot K'$ queries to $\mathcal{A}$ and succeeds with probability at least $\delta^{V_c}_{\mathbf{k}'}(\mathcal{A}_c)/K'$. We define $W \colon \mathcal{C}_1 \times \{0,1\}^* \to \{0,1\}$, by setting $W(c, \mathcal{Y}) = 1$ if and only if $\mathcal{Y}$ is a set satisfying the above properties.

Now let $\mathcal{B}^{\mathcal{A}}\colon \mathcal{C}_1 \to \{0,1\}^*$ be the algorithm, with oracle access to $\mathcal{A}$, that takes as input an element $c \in \mathcal{C}_1$ and runs $\mathcal{E}_{\mu-1}^{\mathcal{A}_c}$. By Lemma 6.7, there exists an expected polynomial time algorithm $\mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$, with oracle access to $\mathcal{B}^{\mathcal{A}}$, that aims to output k_1 pairs $(c_1, \mathcal{Y}_1), \ldots, (c_{k_1}, \mathcal{Y}_{k_1}) \in \mathcal{C}_1 \times \{0,1\}^*$ with $W(c_i, \mathcal{Y}_i) = 1$ for all i and $c_i \neq c_j$ for all $i \neq j$. The extractor $\mathcal{E}^{\mathcal{A}}$ simply runs $\mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$. Note that, by the associativity of the composition of oracle algorithms, $\mathcal{E}^{\mathcal{A}} = \mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}} = (\mathcal{E}_1^{\mathcal{B}})^{\mathcal{A}}$ is indeed an algorithm with oracle access to $\mathcal{A}$.

Let us now analyze the success probability and the expected number of $\mathcal{A}$-queries of the algorithm $\mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$ and therefore of $\mathcal{E}^{\mathcal{A}}$.

Success Probability. Again by Lemma 6.7 it follows that $\mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$ succeeds with probability at least

$$\begin{aligned}
\delta_{k_1}^{W}(\mathcal{B}^{\mathcal{A}})/k_1 &= \min_{S_1 \subseteq \mathcal{C}_1, |S_1| < k_1} \frac{\Pr\big(W(C, \mathcal{B}^{\mathcal{A}}(C)) = 1 \mid C \notin S_1\big)}{k_1} \\
&= \min_{S_1 \subseteq \mathcal{C}_1, |S_1| < k_1} \frac{\Pr\big(W(C, \mathcal{B}^{\mathcal{A}}(C)) = 1 \wedge C \notin S_1\big)}{k_1 \cdot \Pr(C \notin S_1)} \\
&= \min_{S_1 \subseteq \mathcal{C}_1, |S_1| < k_1} \frac{\sum_{c \notin S_1} \Pr(C = c) \cdot \Pr\big(W(c, \mathcal{B}^{\mathcal{A}}(c)) = 1\big)}{k_1 \cdot \Pr(C \notin S_1)}\,,
\end{aligned}$$

where C is uniformly random in $\mathcal{C}$. Hence, by the induction hypothesis it follows that

$$\begin{aligned}
\delta_{k_1}^{W}(\mathcal{B}^{\mathcal{A}})/k_1 &\geq \min_{S_1 \subseteq \mathcal{C}_1, |S_1| < k_1} \frac{\sum_{c \notin S_1} \Pr(C = c) \cdot \delta_{\mathbf{k}'}^{V_c}(\mathcal{A}_c)}{k_1 \cdot K' \cdot \Pr(C \notin S_1)} \\
&= \min_{S_1 \subseteq \mathcal{C}_1, |S_1| < k_1} \frac{\sum_{c \notin S_1} \Pr(C = c) \cdot \delta_{\mathbf{k}'}^{V_c}(\mathcal{A}_c)}{K \cdot \Pr(C \notin S_1)}\,. \qquad (6.9)
\end{aligned}$$

Now note that

$$\delta_{\mathbf{k}'}^{V_c}(\mathcal{A}_c) = \min_{S_2(\cdot), \ldots, S_\mu(\cdot)} \Pr\left(\Lambda \,\middle|\, \begin{array}{l} C_1 = c \wedge C_2 \notin S_2(C_1) \wedge \cdots \\ \quad \cdots \wedge C_\mu \notin S_\mu(C_1, \ldots, C_{\mu-1}) \end{array}\right),$$

where Λ denotes the event $V(C, \mathcal{A}(C)) = 1$. Hence,

$$\begin{aligned}
&\sum_{c \notin S_1} \Pr(C = c) \cdot \delta_{\mathbf{k}'}^{V_c}(\mathcal{A}_c) = \\
&\qquad \min_{S_2(\cdot), \ldots, S_\mu(\cdot)} \Pr\left(\Lambda \wedge C_1 \notin S_1 \,\middle|\, \begin{array}{l} C_2 \notin S_2(C_1) \wedge \cdots \\ \quad \cdots \wedge C_\mu \notin S_\mu(C_1, \ldots, C_{\mu-1}) \end{array}\right).
\end{aligned}$$

Combining this equality with Equation 6.9, shows that

$$\delta_{k_1}^{W}(\mathcal{B}^{\mathcal{A}})/k_1 \geq \frac{\delta_{\mathbf{k}}^{V}(\mathcal{A})}{K}\,,$$

which shows that $\mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$ has the desired success probability.

Expected Number of $\mathcal{A}$-Queries. By Lemma 6.7, it follows that $\mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$ requires an expected number of at most $2k_1$ queries to $\mathcal{B}^{\mathcal{A}}$. By the induction hypothesis it follows that $\mathcal{B}^{\mathcal{A}}(c)$ requires an expected number of at most $2^{\mu-1} \cdot K'$ queries to $\mathcal{A}$ for all $c \in \mathcal{C}$. Hence, $\mathcal{E}^{\mathcal{A}} = \mathcal{E}_1^{\mathcal{B}^{\mathcal{A}}}$ requires an expected number of at most $2^{\mu} \cdot K$ queries to $\mathcal{A}$, which completes the proof of the lemma.

□

Let $S_1, S_2(\cdot), \ldots, S_\mu(\cdot)$ be the arguments minimizing Equation 6.8. Further, let Λ denote the event $V(C, \mathcal{A}(C)) = 1$ and let Γ denote the event

$$\Gamma \;=\; C_1 \notin S_1 \wedge C_2 \notin S_2(C_1) \wedge \cdots \wedge C_\mu \notin S_\mu(C_1, \ldots, C_{\mu-1})\,.$$

Then, using the same kind of reasoning as in Equation 6.4, we have

$$\delta_{\mathbf{k}}^{V}(\mathcal{A}) = \Pr(\Lambda \mid \Gamma) = \frac{\Pr(\Lambda \wedge \Gamma)}{\Pr(\Gamma)} \geq \frac{\Pr(\Lambda) - \Pr(\neg\Gamma)}{\Pr(\Gamma)} = \frac{\epsilon^{V}(\mathcal{A}) - \mathrm{Er}(\mathbf{k};\mathbf{N})}{1 - \mathrm{Er}(\mathbf{k};\mathbf{N})}\,,$$

where

$$\mathrm{Er}(\mathbf{k};\mathbf{N}) = \Pr(\neg\Gamma) = 1 - \prod_{i=1}^{\mu}\left(1 - \frac{k_i - 1}{N_i}\right).$$

This confirms that a $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proof is knowledge sound with knowledge error $\mathrm{Er}(\mathbf{k};\mathbf{N})$, i.e., it provides an alternative proof for Theorem 6.4. This alternative approach, and in particular the quality measure $\delta_{\mathbf{k}}^{V}(\mathcal{A})$, allows us to generalize to parallel repetitions of $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs.

Knowledge-Soundness of the Parallel Repetition

We finally move towards stating and proving our main general parallel repetition result for multi-round protocols. Thus, consider the t-fold parallel repetition $\Pi^t = (\mathcal{P}^t, \mathcal{V}^t)$ of the given $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound $(2\mu+1)$-round interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$.

We consider an algorithm $\mathcal{A}$ that takes as input a *row* $(\mathbf{c}^1, \ldots, \mathbf{c}^t)$ of *columns* $\mathbf{c}^j = (c_1^j, \ldots, c_\mu^j) \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$ of challenges and outputs a string y. Furthermore, we consider a verification function V, which then defines the *success probability* of $\mathcal{A}$ as

$$\epsilon^{V}(\mathcal{A}) = \Pr\big(V(C, \mathcal{A}(C)) = 1\big)\,,$$

where $C = (C^1, \ldots, C^t)$ with C^j distributed uniformly over $\mathcal{C}_1 \times \cdots \mathcal{C}_\mu$ for all $1 \leq j \leq t$.

Again, the obvious instantiation for $\mathcal{A}$ is a deterministic dishonest prover $\mathcal{P}^*$ attacking $\Pi^t = (\mathcal{P}^t, \mathcal{V}^t)$ on input x. More precisely, on input a row $(\mathbf{c}^1, \ldots, \mathbf{c}^t)$ of columns, $\mathcal{A}$ runs $\mathcal{P}^*$ sending $(\mathbf{c}^1, \ldots, \mathbf{c}^t)$ as the challenges, and outputs all of $\mathcal{P}^*$'s messages, and the function V is defined as the verification check that $\mathcal{V}^t$ performs.

Such an $\mathcal{A}$ naturally induces t algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_t$ as considered before in the context of a single execution of a multi-round protocol, taking one challenge-column as input and outputting one string: on input $\mathbf{c}^j$, the algorithm $\mathcal{A}_j$ runs $y \leftarrow \mathcal{A}(\mathbf{c}^1, \ldots, \mathbf{c}^\mu)$ with $\mathbf{c}^i$ chosen uniformly at random from $\mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$ for $i \neq j$,

and outputs y along with the $\mathbf{c}^i$'s for $i \neq j$. Thus, we can run the extractor from Lemma 6.9 on all of the $\mathcal{A}_j$'s individually, with the goal being that at least one of them succeeds. For each $\mathcal{A}_j$ individually, the extraction succeeds with probability at least

$$\delta^V_{\mathbf{k}}(\mathcal{A}_j)/K = \min_{S^j_1, S^j_2(\cdot), \ldots, S^j_\mu(\cdot)} \Pr\left(\Lambda_j \,\middle|\, \begin{array}{c} C^j_1 \notin S^j_1 \wedge C^j_2 \notin S^j_2(C^j_1) \wedge \cdots \\ \cdots \wedge C^j_\mu \notin S^j_\mu(C^j_1, \ldots, C^j_{\mu-1}) \end{array}\right)/K\,, \tag{6.10}$$

where Λ_j denotes the event $V(C^j, \mathcal{A}_j(C^j)) = 1$, V is understood to appropriately reorder its inputs and $K = \prod_{i=1}^{\mu} k_i$. The following lemma allows us to bound the probability that at least one of the extractors $\mathcal{E}^{\mathcal{A}_j}$ succeeds.

Lemma 6.10. *Let $\mathbf{k} = (k_1, \ldots, k_\mu), \mathbf{N} = (N_1, \ldots, N_\mu) \in \mathbb{N}^\mu$, $t \in \mathbb{N}$, $\mathcal{C}_1, \ldots, \mathcal{C}_\mu$ finite sets $\mathcal{C}_i$ with cardinality $N_i \geq k_i$ and let $V\colon \left(\mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu\right)^t \times \{0,1\}^* \to \{0,1\}$. Further, let $\mathcal{A}$ be a (probabilistic) algorithm that takes as input a row $(\mathbf{c}^1, \ldots, \mathbf{c}^t)$ of columns $\mathbf{c}^j = (c^j_1, \ldots, c^j_\mu) \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$ and outputs a string $y \in \{0,1\}^*$. Then*

$$\sum_{j=1}^{t} \delta^V_{\mathbf{k}}(\mathcal{A}_j) \geq \frac{\epsilon^V(\mathcal{A}) - \mathrm{Er}(\mathbf{k}; \mathbf{N})^t}{1 - \mathrm{Er}(\mathbf{k}; \mathbf{N})}\,,$$

where

$$\mathrm{Er}(\mathbf{k}; \mathbf{N}) = 1 - \prod_{i=1}^{\mu}\left(1 - \frac{k_i - 1}{N_i}\right).$$

Proof. Let Λ denote the event $V(C, \mathcal{A}(C)) = 1$ and, for $1 \leq j \leq t$, let S^j_1, $S^j_2(\cdot)$, $\ldots$, $S^j_\mu(\cdot)$ be such that they minimize Equation 6.10. Moreover, let Γ_j denote the event

$$C^j_1 \notin S^j_1 \wedge C^j_2 \notin S^j_2(C^j_1) \wedge \cdots \wedge C^j_\mu \notin S^j_\mu(C^j_1, \ldots, C^j_{\mu-1})\,.$$

Without loss of generality, we may assume that $\left|S^j_1\right| = k_1 - 1$ and

$$S^j_i : \mathcal{C}_1 \times \cdots \times \mathcal{C}_{i-1} \to \{S \subseteq \mathcal{C}_i : |S| = k_i - 1\}$$

for all $2 \leq i \leq \mu$ and $1 \leq j \leq t$. Then, for all $1 \leq j \leq t$,

$$\Pr(\Gamma_j) = \prod_{i=1}^{\mu}\left(1 - \frac{k_i - 1}{N}\right) = 1 - \mathrm{Er}(\mathbf{k}; \mathbf{N})\,.$$

Moreover, using elementary probability theory,

$$\begin{aligned}\sum_{j=1}^{t} \delta^V_{\mathbf{k}}(\mathcal{A}_j) &= \sum_{j=1}^{t} \Pr\big(\Lambda \mid \Gamma_j\big) = \sum_{j=1}^{t} \frac{\Pr\big(\Lambda \wedge \Gamma_j\big)}{\Pr\big(\Gamma_j\big)} = \sum_{j=1}^{t} \frac{\Pr\big(\Lambda \wedge \Gamma_j\big)}{1 - \mathrm{Er}(\mathbf{k}; \mathbf{N})} \\ &\geq \frac{\Pr\big(\Lambda \wedge \exists\, j : \Gamma_j\big)}{1 - \mathrm{Er}(\mathbf{k}; \mathbf{N})} \geq \frac{\Pr\big(\Lambda\big) - \Pr\big(\neg\Gamma_j\ \forall j\big)}{1 - \mathrm{Er}(\mathbf{k}; \mathbf{N})} = \frac{\epsilon^V(\mathcal{A}) - \mathrm{Er}(\mathbf{k}; \mathbf{N})^t}{1 - \mathrm{Er}(\mathbf{k}; \mathbf{N})}\,,\end{aligned}$$

which completes the proof. □

As for the parallel repetition of a 3-round protocol, it follows that the probability of at least one of the extractors $\mathcal{E}^{\mathcal{A}_j}$ being successful is at least

$$\frac{\Delta}{2} \geq \frac{\epsilon^V(\mathcal{A}) - \mathrm{Er}(\mathbf{k};\mathbf{N})^t}{2K(1-\mathrm{Er}(\mathbf{k};\mathbf{N}))},$$

where $\Delta = \min\bigl(1, \sum_{j=1}^{t} \delta_{\mathbf{k}}^V(\mathcal{A}_j)/K\bigr)$ and $K = \prod_{i=1}^{\mu} k_i$. This gives us the following strong parallel repetition result for $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound protocols.

Theorem 6.8 (Parallel Repetition Theorem for Multi-Round Protocols)**.** *Let* $\Pi = (\mathcal{P}, \mathcal{V})$ *be a* $\mathbf{k}$*-out-of-*$\mathbf{N}$ *special-sound interactive proof. Then the* t*-fold parallel repetition* $\Pi^t = (\mathcal{P}^t, \mathcal{V}^t)$ *of* Π *is knowledge sound with knowledge error* $\mathrm{Er}(\mathbf{k};\mathbf{N})^t$*, where*

$$\mathrm{Er}(\mathbf{k};\mathbf{N}) = 1 - \prod_{i=1}^{\mu}\left(1 - \frac{k_i - 1}{N_i}\right),$$

is the knowledge error of Π*.*

Also here, the knowledge error $\mathrm{Er}(\mathbf{k};\mathbf{N})^t$ coincides with success probability $\prod_j \Pr(\neg\Gamma_j)$ of the trivial cheating strategy, which typical $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs admit.

6.5.4 Threshold Parallel Repetition

In the previous section we have shown that the knowledge error $\mathrm{Er}(\mathbf{k};\mathbf{N})^t$ of the t-fold parallel repetition $\Pi^t = (\mathcal{P}^t, \mathcal{V}^t)$ of a $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proof $\Pi = (\mathcal{P}, \mathcal{V})$ decreases exponentially with t. However, the completeness error of Π^t equals $\rho' = 1 - (1-\rho)^t$, where ρ is the completeness error of Π. Hence, if $\rho \notin \{0, 1\}$, the completeness error of Π^t increases quickly with t. In order to decrease both the knowledge and the completeness error simultaneously, we consider a *threshold parallel repetition*. The s-out-of-t threshold parallel repetition of an interactive proof Π, denoted by $\Pi_s^t = (\mathcal{P}_s^t, \mathcal{V}_s^t)$, runs t instances of Π in parallel and $\mathcal{V}_s^t$ accepts if at least s-out-of-t instances are accepted. In particular, it holds that $\Pi_t^t = \Pi^t$. In this section, we show that if Π is $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound then Π_s^t is knowledge sound. We will immediately consider the general case of multi-round protocols.

As in Section 6.5.3, we consider an algorithm $\mathcal{A}$ that takes as input a row $\mathbf{c} = (\mathbf{c}^1, \ldots, \mathbf{c}^t)$ of columns $\mathbf{c}^j = (c_1^j, \ldots, c_\mu^j) \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$ of challenges and outputs a string y. However, this time we consider t different verification functions

$$V_j \colon \bigl(\mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu\bigr)^t \times \{0,1\}^* \to \{0,1\},$$

together with one additional *threshold* verification function defined as follows:

$$V(\mathbf{c}, y) = \begin{cases} 1 & \text{if } \displaystyle\sum_{j=1}^{t} V_j(\mathbf{c}, y) \geq s, \\ 0 & \text{otherwise}. \end{cases} \tag{6.11}$$

The obvious instantiation for $\mathcal{A}$ is a deterministic dishonest prover $\mathcal{P}^*$ attacking Π_s^t. This instantiation defines V_j as the verification performed by the j-th instance of $\mathcal{V}$. The verification function V then captures the verification performed by $\mathcal{V}_s^t$.

As before, such $\mathcal{A}$ induces t algorithms $\mathcal{A}_1, \ldots, \mathcal{A}_t$ as considered in the context of a single execution of Π, taking one challenge-column as input and outputting one string: on input $\mathbf{c}^j$, the algorithm $\mathcal{A}_j$ runs $y \leftarrow \mathcal{A}(\mathbf{c}^1, \ldots, \mathbf{c}^\mu)$ with $\mathbf{c}^i$ chosen uniformly at random from $\mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$ for $i \neq j$, and outputs y along with the $\mathbf{c}^i$'s for $i \neq j$. For each $\mathcal{A}_j$, we can run the extractor from Lemma 6.9, which succeeds with probability at least

$$\delta_{\mathbf{k}}^{V_j}(\mathcal{A}_j)/K = \min_{S_1^j, S_2^j(\cdot), \ldots, S_\mu^j(\cdot)} \Pr\left(\Lambda_j \,\middle|\, \begin{array}{c} C_1^j \notin S_1^j \wedge C_2^j \notin S_2^j(C_1^j) \wedge \cdots \\ \cdots \wedge C_\mu^j \notin S_\mu^j(C_1^j, \ldots, C_{\mu-1}^j) \end{array}\right)/K\,, \tag{6.12}$$

where Λ_j denotes the event $V_j(C_j, \mathcal{A}_j(C_j)) = 1$ and $K = \prod_{i=1}^{\mu} k_i$. The following lemma is a generalization of Lemma 6.10 and it allows us to bound the probability that at least one of the extractors $\mathcal{E}^{\mathcal{A}_j}$ succeeds.

Lemma 6.11. *Let $\mathbf{k} = (k_1, \ldots, k_\mu)$, $\mathbf{N} = (N_1, \ldots, N_\mu) \in \mathbb{N}^\mu$, $t \in \mathbb{N}$, $\mathcal{C}_1, \ldots, \mathcal{C}_\mu$ finite sets $\mathcal{C}_i$ with cardinality $N_i \geq k_i$, let $V\colon \left(\mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu\right)^t \times \{0,1\}^* \to \{0,1\}$ be the threshold verification function as defined in Equation (6.11). Further, let $\mathcal{A}$ be a (probabilistic) algorithm that takes as input a row $(\mathbf{c}^1, \ldots, \mathbf{c}^t)$ of columns $\mathbf{c}^j = (c_1^j, \ldots, c_\mu^j) \in \mathcal{C}_1 \times \cdots \times \mathcal{C}_\mu$ and outputs a string $y \in \{0,1\}^*$. Then*

$$\sum_{j=1}^{t} \delta_{\mathbf{k}}^{V_j}(\mathcal{A}_j) \geq \frac{\epsilon^V(\mathcal{A}) - \mathrm{Er}_s^t(\mathbf{k}; \mathbf{N})}{1 - \mathrm{Er}(\mathbf{k}; \mathbf{N})}\,,$$

where

$$\mathrm{Er}_s^t(\mathbf{k}; \mathbf{N}) = \sum_{\ell=s}^{t} \binom{t}{\ell} \mathrm{Er}(\mathbf{k}; \mathbf{N})^\ell (1 - \mathrm{Er}(\mathbf{k}; \mathbf{N}))^{t-\ell}$$

and

$$\mathrm{Er}(\mathbf{k}; \mathbf{N}) = 1 - \prod_{i=1}^{\mu} \left(1 - \frac{k_i - 1}{N_i}\right).$$

Note that $\mathrm{Er}_s^t(\mathbf{k}; \mathbf{N})$ is the probability of being successful at least s times when given t trials, when each trial is successful with independent probability $\mathrm{Er}(\mathbf{k}; \mathbf{N})$.

Proof. For $1 \leq j \leq t$, let Λ_j denote the event $V_j(C, \mathcal{A}_j(C)) = 1$ and let $S_1^j, S_2^j(\cdot), \ldots, S_\mu^j(\cdot)$ such that they minimize Equation 6.12. Moreover, let Γ_j denote the event

$$C_1^j \notin S_1^j \wedge C_2^j \notin S_2^j(C_1^j) \wedge \cdots \wedge C_\mu^j \notin S_\mu^j(C_1^j, \ldots, C_{\mu-1}^j)\,.$$

Without loss of generality, we may assume that $|S_1^j| = k_1 - 1$ and

$$S_i^j : \mathcal{C}_1 \times \cdots \mathcal{C}_{i-1} \to \{S \subset \mathcal{C}_i : |S| = k_i - 1\}$$

for all $2 \leq i \leq \mu$ and $1 \leq j \leq t$. Then, for all $1 \leq j \leq t$,

$$\Pr(\Gamma_j) = \prod_{i=1}^{\mu} \left(1 - \frac{k_i - 1}{N_i}\right) = 1 - \mathrm{Er}(\mathbf{k}; \mathbf{N})\,.$$

Moreover, using elementary probability theory,

$$\begin{aligned}\sum_{j=1}^{t} \delta_{\mathbf{k}}^{V_j}(\mathcal{A}_j) &= \sum_{j=1}^{t} \Pr\big(\Lambda_j \mid \Gamma_j\big) = \sum_{j=1}^{t} \frac{\Pr\big(\Lambda_j \wedge \Gamma_j\big)}{\Pr\big(\Gamma_j\big)} = \sum_{j=1}^{t} \frac{\Pr\big(\Lambda_j \wedge \Gamma_j\big)}{1-\mathrm{Er}(\mathbf{k};\mathbf{N})} \\ &\geq \frac{\Pr\big(\exists\, j : \Lambda_j \wedge \Gamma_j\big)}{1-\mathrm{Er}(\mathbf{k};\mathbf{N})} \geq \frac{\Pr\big(|\{j : \Lambda_j\}| \geq s \wedge |\{j : \Gamma_j\}| \geq t-s+1\big)}{1-\mathrm{Er}(\mathbf{k};\mathbf{N})} \\ &\geq \frac{\Pr\big(|\{j : \Lambda_j\}| \geq s\big) - \Pr\big(|\{j : \Gamma_j\}| \leq t-s\big)}{1-\mathrm{Er}(\mathbf{k};\mathbf{N})} \geq \frac{\epsilon^V(\mathcal{A}) - \mathrm{Er}_s^t(\mathbf{k};\mathbf{N})}{1-\mathrm{Er}(\mathbf{k};\mathbf{N})}\,,\end{aligned}$$

which completes the proof. □

As before (see Equation 6.7), it follows that the probability of at least one of the extractors $\mathcal{E}^{\mathcal{A}_j}$ being successful is at least

$$\frac{\Delta}{2} \geq \frac{\epsilon^V(\mathcal{A}) - \mathrm{Er}_s^t(\mathbf{k};\mathbf{N})}{2K(1-\mathrm{Er}(\mathbf{k};\mathbf{N}))}\,,$$

where $\Delta = \min\big(1, \sum_{j=1}^{t} \delta_{\mathbf{k}}^{V_j}(\mathcal{A}_j)/K\big)$ and $K = \prod_{i=1}^{\mu} k_i$. This proves the following threshold parallel repetition result for $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs.

Theorem 6.9 (Threshold Parallel Repetition Theorem)**.** *Let $\Pi = (\mathcal{P}, \mathcal{V})$ be a $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proof. Then the s-out-of-t threshold parallel repetition $\Pi_s^t = (\mathcal{P}_s^t, \mathcal{V}_s^t)$ of Π is knowledge sound with knowledge error*

$$\mathrm{Er}_s^t(\mathbf{k};\mathbf{N}) = \sum_{\ell=s}^{t} \binom{t}{\ell} \mathrm{Er}(\mathbf{k};\mathbf{N})^{\ell} (1-\mathrm{Er}(\mathbf{k};\mathbf{N}))^{t-\ell}\,,$$

where

$$\mathrm{Er}(\mathbf{k};\mathbf{N}) = 1 - \prod_{i=1}^{\mu} \left(1 - \frac{k_i - 1}{N_i}\right),$$

is the knowledge error of Π.

As before, the knowledge error $\mathrm{Er}_s^t(\mathbf{k};\mathbf{N})$ coincides with the trivial cheating probability for Π_s^t, confirming the tightness of Theorem 6.9.

Note that the completeness error of Π_s^t equals

$$\rho_s^t = \sum_{\ell=0}^{s-1} \binom{t}{\ell} \rho^{t-\ell} (1-\rho)^{\ell}\,.$$

Hence, the completeness error ρ_s^t increases and the knowledge error $\mathrm{Er}_s^t(\mathbf{k};\mathbf{N})$ decreases in s. Moreover, it is easily seen that for t large enough and $\mathrm{Er}(\mathbf{k};\mathbf{N}) \cdot t < s < (1-\rho)t$ the threshold parallel repetition Π_s^t has a smaller knowledge and a smaller completeness error than Π, i.e., $\mathrm{Er}_s^t(\mathbf{k};\mathbf{N}) < \mathrm{Er}(\mathbf{k};\mathbf{N})$ and $\rho_s^t < \rho$. In contrast to standard parallel repetition, threshold parallel repetition therefore allows both these errors to be reduced simultaneously.

6.6 Non-Interactivity: Knowledge Extraction under the Fiat-Shamir Transformation

The celebrated and broadly used Fiat-Shamir transformation turns any public-coin interactive proof into a *non-interactive* proof, which inherits the main security properties (in the random oracle model) of the interactive version. The rough idea is to replace the random challenges, which are provided by the verifier in the interactive version, by the hash of the current message (concatenated with the message-challenge pairs from previous rounds). By a small adjustment, where also the to-be-signed message is included in the hashes, the transformation turns any public-coin interactive proof into a signature scheme. Indeed, the latter is a commonly used design principle for constructing very efficient signature schemes.

While originally considered in the context of 3-round public-coin interactive proofs, i.e., so-called Σ-protocols, the Fiat-Shamir transformation also applies to *multi-round* protocols. However, a major drawback in the case of multi-round protocols is that, in general, the security loss obtained by applying the Fiat-Shamir transformation grows exponentially with the number of rounds. Concretely, for any $(2\mu+1)$-round interactive proof Π that admits a cheating probability of at most ϵ, captured by the knowledge or soundness error, the Fiat-Shamir-transformed protocol $\mathsf{FS}[\Pi]$ admits a cheating probability of (approximately) at most $Q^\mu \cdot \epsilon$, where Q denotes the number of random-oracle queries admitted to the dishonest prover. More precisely, a tight reduction is due to [BCS16] with a security loss $\binom{Q}{\mu} \approx \frac{Q^\mu}{\mu^\mu}$, where the approximation holds whenever μ is much smaller than Q, which is the typical case. More concretely, [BCS16] introduces the notions of *state-restoration soundness (SRS)* and *state-restoration knowledge (SRK)*, and it shows that any (knowledge) sound protocol Π satisfies these notions with the claimed security loss.[5] The security of $\mathsf{FS}[\Pi]$ (with the same loss) then follows from the fact that these soundness notions imply the security of the Fiat-Shamir transformation.

Furthermore, there are (contrived) examples of multi-round protocols Π for which this Q^μ security loss is almost tight. For instance, the μ-fold sequential repetition Π of a special-sound Σ-protocol with challenge space $\mathcal{C}$ is ϵ-sound with $\epsilon = 1/|\mathcal{C}|^\mu$, while it is easy to see that, by attacking the sequential repetitions round by round, investing Q/μ queries per round to try to find a "good" challenge, and assuming $|\mathcal{C}|$ to be much larger than Q, its Fiat-Shamir transformation $\mathsf{FS}[\Pi]$ can be broken with probability approximately $\left(\frac{Q}{\mu}\frac{1}{|\mathcal{C}|}\right)^\mu = \frac{Q^\mu}{\mu^\mu} \cdot \epsilon$.[6]

For μ beyond 1 or 2, let alone for non-constant μ (e.g., for compressed Σ-protocols, IOP-based protocols [BCS16; AHI+17; BCR+19] and also other Bulletproofs-like protocols [BCC+16; BBB+18]), this is a very unfortunate situation when it comes to choosing concrete security parameters. If one wants to rely on the proven security reduction, one needs to choose a large security parameter for Π, in order to compensate for the order Q^μ security loss, affecting its efficiency; alternatively, one has to give up on proven security and simply *assume*

[5] As a matter of fact, [BCS16] considers arbitrary *interactive oracle proofs (IOPs)*, but these notions are well defined for ordinary interactive proofs too.

[6] This is clearly a contrived example since the natural construction would be to apply the Fiat-Shamir transformation to the *parallel* repetition of the original Σ-protocol, where no such huge security loss would then occur.

that the security loss is much milder than what the general bound suggests.

This situation gives rise to the following question: *Do there exist natural classes of multi-round public-coin interactive proofs for which the security loss behaves more benign than what the general reduction suggests?* Ideally, the general Q^μ loss appears for contrived examples only.

So far, the only positive results, establishing a security loss linear in Q, were established in the context of *straight-line/online* extractors that do not require rewinding. These extractors either rely on the algebraic group model (AGM) [GT21], or are restricted to protocols using hash-based commitment schemes in the random oracle model [BCS16]. To analyze the properties of straight-line extractors, new auxiliary soundness notions were introduced: *round-by-round (RBR) soundness* [CCH+19] and *RBR knowledge* [CMS19]. However, it is unclear if and how these notions can be used in scenarios where straight-line extraction does not apply.

In this section, we address the above question (in the plain random-oracle model, and without restricting to schemes that involve hash-based commitments), and give both positive and negative answers.

6.6.1 Technical Overview

Positive Result. We show that the Fiat-Shamir transformation of any $(k_1, \ldots, k_\mu)$-out-of-$(N_1, \ldots, N_\mu)$ special-sound interactive proof has a security loss of at most $Q + 1$. More concretely, we consider the *knowledge error* κ as the figure of merit, i.e., informally, the maximal probability of the verifier accepting the proof when the prover does not have a witness for the claimed statement, and we prove the following result. For any $(k_1, \ldots, k_\mu)$-out-of-$(N_1, \ldots, N_\mu)$-special-sound $(2\mu + 1)$-round interactive proof Π with knowledge error $\kappa = \mathrm{Er}(k_1, \ldots, k_\mu, N_1, \ldots, N_\mu)$, the Fiat-Shamir transformed protocol $\mathsf{FS}[\Pi]$ has a knowledge error at most $(Q + 1) \cdot \kappa$.

Since in the Fiat-Shamir transformation of any $(2\mu + 1)$-round protocol Π, a dishonest prover can simulate any attack against Π, and can try Q/μ times when allowed to do Q queries in total, our new upper bound $(Q + 1) \cdot \kappa$ is close to the trivial lower bound $1-(1-\kappa)^{Q/\mu} \approx Q\kappa/\mu$. Another, less explicit security measure in the context of knowledge soundness is the run time of the knowledge extractor. Our bound on the knowledge error holds by means of a knowledge extractor that makes an expected number of $K+Q\cdot(K-1)$ queries, where $K = k_1 \cdots k_\mu$. This is a natural bound: K is the number of necessary distinct "good" transcripts (which form a certain tree-like structure). The loss of $Q \cdot (K-1)$ captures the fact that a prover may finish different proofs, depending on the random oracle answers, and only one out of Q proofs may be useful for extraction, as explained below.

Our result on the *knowledge* soundness of $\mathsf{FS}[\Pi]$ for special-sound protocols Π immediately carries over to *ordinary* soundness of $\mathsf{FS}[\Pi]$, with the same security loss $Q + 1$. However, proving knowledge soundness is more intricate; showing a linear-in-Q loss for ordinary soundness can be obtained via simpler arguments (e.g., there is no need to argue efficiency of the extractor).

The construction of our knowledge extractor is motivated by the extractor from Section 6.4 in the interactive case, but the analysis here in the context of a non-

interactive proof is more involved. We analyze the extractor in an inductive manner, and capture the induction step (and the base case) by means of an abstract experiment. The crucial idea for the analysis (and extractor) is how to deal with accepting transcripts that are not useful.

To see the core problem, consider a Σ-protocol, i.e., a 3-round k-special-sound interactive proof, and a semi-honest prover that knows a witness and behaves as follows. It prepares, independently, Q first messages $a^1, \ldots, a^Q$ and asks for all hashes $c^i = \mathsf{RO}(a^i)$, and then decides "randomly" (e.g., using a hash over all random oracle answers) which thread to complete, i.e., for which i^* to compute the response z and then output the valid proof (a^{i^*}, z). When the extractor then reprograms the random oracle at the point a^{i^*} to try to obtain another valid response but now for a different challenge, this affects i^*, and most likely the prover will then use a different thread j^* and output the proof (a^{j^*}, z') with $a^{j^*} \neq a^{i^*}$. More precisely, $\Pr(j^* = i^*) = 1/Q$. Hence, an overhead of Q appears in the run-time.

In case of an *arbitrary* dishonest prover with an unknown strategy for computing the a^i's above, and with an arbitrary (unknown) success probability ϵ, the intuition remains: after reprogramming, we still expect $\Pr(j^* = i^*) \geq 1/Q$ and thus a linear-in-Q overhead in the run-time of the extractor. However, providing a rigorous proof is complicated by the fact that the event $j^* = i^*$ is not necessarily independent of the prover producing a *valid* proof (again) after the reprogramming. Furthermore, conditioned on the prover having been successful in the first run and conditioned on the corresponding i^*, the success probability of the prover after the reprogramming may be skewed, i.e., may not be ϵ anymore. As a warm-up for our general multi-round result, we first give a rigorous analysis of the above case of a Σ-protocol. For that purpose, we introduce an abstract sampling game that mimics the behavior of the extractor in finding two valid proofs with $j^* = i^*$, and we bound the success probability and the "cost" (i.e., the number of samples needed) of the game, which directly translate to the success probability and the run-time of the extractor.

Perhaps surprisingly, when moving to *multi-round* protocols, dealing with the knowledge error is relatively simple by recursively composing the extractor for the Σ-protocol. However, controlling the run-time is intricate. If the extractor is recursively composed, i.e., it makes calls to a sub-extractor to obtain a subtree, then a naive construction and analysis gives a blow-up of Q^μ in the run-time. Intuitively, because only $1/Q$ of the sub-extractor runs produce useful subtrees, i.e., subtrees which extend the current a^{i^*}. The other trees belong to some a^{j^*} with $j^* \neq i^*$ and are thus useless. This overhead of Q then accumulates per round (i.e., per sub-extractor).

The crucial observation that we exploit in order to overcome the above issue is that the very first (accepting) transcript sampled by a sub-extractor already determines whether a subtree will be (potentially) useful, or not. Thus, if this very first transcript already shows that the subtree will not be useful, there is no need to run the full-fledged subtree extractor, saving precious time.

To illustrate this more, we again consider the simple case of a dishonest prover that succeeds with certainty. Then, after the first run of the sub-extractor to produce the first subtree (which requires expected time linear in Q) and having reprogrammed the random oracle with the goal to find another subtree that ex-

tends the current a^{i^*}, it is cheaper to first do a single run of the prover to learn j^* and only run the full fledged sub-extractor if $j^* = i^*$, and otherwise reprogram and re-try again. With this strategy, we expect Q tries, followed by the run of the sub-extractor, to find a second fitting subtree. Altogether, this amounts to linear-in-Q runs of the prover, compared to the Q^2 using the naive approach.

Again, what complicates the rigorous analysis is that the prover may succeed with bounded probability ϵ only, and the event $j^* = i^*$ may depend on the prover/sub-extractor being successful (again) after the reprogramming. Furthermore, as an additional complication, conditioned on the sub-extractor having been successful in the first run and conditioned on the corresponding i^*, both the success probability of the prover and the run-time of the sub-extractor after the reprogramming may be skewed now. Again, we deal with this by considering an abstract sampling game that mimics the behavior of the extractor, but where the cost function is now more fine-grained in order to distinguish between a single run of the prover and a run of the sub-extractor. Because of this more fine-grained way of defining the "cost," the analysis of the game also becomes substantially more intricate.

Negative Result. We also show that the general exponential security loss of the Fiat-Shamir transformation, when applied to a multi-round protocol, is not an artifact of contrived examples, but there exist natural protocols that indeed have such an exponential loss. For instance, our negative result applies to the lattice-based protocols in [BLN+20; ACK21]. Concretely, we show that the t-fold parallel repetition Π^t of a typical $(k_1, \ldots, k_\mu)$-special-sound $(2\mu{+}1)$-round interactive proof Π features this behavior when $t \geq \mu$. For simplicity, let us assume that t and Q are multiples of μ. Then, in more detail, we show that for any typical $(k_1, \ldots, k_\mu)$-special-sound protocol Π there exists a polynomial time Q-query prover $\mathcal{P}^*$ against $\mathsf{FS}[\Pi^t]$ that succeeds in making the verifier accept with probability approximately $\frac{1}{2}Q^\mu\kappa^t/\mu^{\mu+t}$ for *any* statement x, where κ is the knowledge error (as well as the soundness error) of Π. Thus, with the claimed probability, $\mathcal{P}^*$ succeeds in making the verifier accept for statements x that are not in the language and/or for which $\mathcal{P}^*$ does not know a witness. Given that, by Section 6.5, κ^t is the knowledge error of Π^t (i.e., the soundness error of Π^t as an interactive proof), this shows that the knowledge error of Π^t grows proportionally with Q^μ when applying the Fiat-Shamir transformation.

6.6.2 Related Work

Independent Concurrent Work. In independent and to a large extent concurrent work,[7] Wikström [Wik21] achieves a similar positive result on the Fiat-Shamir transformation, using a different approach and different techniques: [Wik21] reduces non-interactive extraction to a form of interactive extraction and then applies a generalized version of [Wik18], while our construction adapts the interactive extractor from Section 6.4 and offers a direct analysis. One difference in the results, which is mainly of theoretical interest, is that our result holds and is meaningful for *any* $Q < |\mathcal{C}|$, whereas [Wik21] requires the challenge set $\mathcal{C}$ to be large.

[7] When finalizing our write-up [AFK22], we were informed by Wikström that he derived similar results a few months earlier, subsequently made available online [Wik21].

The Forking Lemma. The security of the Fiat–Shamir transformation of k-out-of-N special-sound Σ-protocols is widely used for construction of signatures. There, unforgeability is typically proven via a forking lemma [PS96; BN06], which extracts, with probability roughly ϵ^k/Q, a witness from a signature-forging adversary with success probability ϵ, where Q is the number of queries to the random oracle. The loss ϵ^k is due to *strict* polynomial time extraction (and can be decreased, but in general not down to ϵ). Such a k-th power loss in the success probability for a constant k is fine in certain settings, e.g., for proving the security of signature schemes; however, not for proofs of knowledge (which, on the other hand, consider *expected* polynomial time extraction [BL02]).

We are not aware of forking lemmas being used in the context of the Fiat–Shamir transformation for multi-round interactive proofs, i.e., for $(2\mu+1)$-round interactive proofs with $\mu > 1$. The techniques for interactive proofs are not directly applicable to the Fiat-Shamir mode. First, incorporating the query complexity Q of a dishonest prover $\mathcal{P}^*$ attacking the non-interactive Fiat–Shamir transformation complicates the analysis. Second, a naive adaptation of the forking lemmas for interactive proofs gives a blow-up of Q^μ in the run-time.

6.6.3 An Abstract Sampling Game

Towards the goal of constructing and analyzing a knowledge extractor for the Fiat-Shamir transformation $\mathsf{FS}[\Pi]$ of special-sound interactive proofs Π, we define and analyze an abstract sampling game. Given access to a deterministic Q-query prover $\mathcal{P}^*$, attacking the non-interactive random oracle proof $\mathsf{FS}[\Pi]$, our extractor will essentially play this abstract game in the case Π is a Σ-protocol, and it will play this game recursively in the general case of a multi-round protocol. The abstraction allows us to focus on the crucial properties of the extraction algorithm, without unnecessarily complicating the notation.

The game considers an arbitrary but fixed U-dimensional array M, where, for all $1 \le j_1, \ldots, j_U \le N$, the entry $M(j_1, \ldots, j_U) = (v, i)$ contains a bit $v \in \{0,1\}$ and an index $i \in \{1, \ldots, U\}$. Think of the bit v indicating whether this entry is "good" or "bad," and the index i pointing to one of the U dimensions. The goal will be to find k "good" entries with the same index i, and with all of them lying in the 1-dimensional array $M(j_1, \ldots, j_{i-1}, \cdot, j_{i+1}, \ldots, j_U)$ for some $1 \le j_1, \ldots, j_{i-1}, j_{i+1}, \ldots, j_U \le N$.

Looking ahead, considering the case of a Σ-protocol first, this game captures the task of our extractor to find k proofs that are valid and feature the same first message, but have different hash values assigned to the first message. Thus, in our application, the sequence $j_1, \ldots, j_U$ specifies the function table of the random oracle

$$\mathsf{RO}\colon \{1, \ldots, U\} \to \{1, \ldots, N\}, \quad i \mapsto j_i$$

while the entry $M(j_1, \ldots, j_U) = (v, i)$ captures the relevant properties of the proof produced by the considered prover when interacting with that particular specification of the random oracle. Concretely, the bit v indicates whether the proof is valid, and the index i is the first message a of the proof. Replacing j_i by j_i' then means to reprogram the random oracle at the point $i = a$. Note that after the reprogramming, we want to obtain another valid proof with the *same* first

message, i.e., with the same index i (but now a different challenge, due to the reprogramming).

The game is formally defined in Figure 6.4 and its core properties are summarized in Lemma 6.12 below. Looking ahead, we note that for efficiency reasons, the extractor will naturally not sample the entire sequence $j_1, \ldots, j_U$ (i.e., function table), but will sample the relevant components on the fly using lazy sampling.

It will be useful to define, for all $1 \le i \le U$, the function

$$\begin{aligned} a_i \colon \{1,\ldots,N\}^U &\to \mathbb{N}_{\ge 0}, \\ (j_1,\ldots,j_U) &\mapsto \left|\left\{ j : \ M(j_1,\ldots,j_{i-1},j,j_{i+1},\ldots,j_U) = (1,i) \ \right\}\right| . \end{aligned} \tag{6.13}$$

The value $a_i(j_1,\ldots,j_U)$ counts the number of entries that are "good" and have index i in the 1-dimensional array $M(j_1,\ldots,j_{i-1},\cdot,j_{i+1},\ldots,j_U)$. Note that a_i does not depend on the i-th entry of the input vector $(j_1,\ldots,j_U)$, and so, by a slight abuse of notation, we sometimes also write $a_i(j_1,\ldots,j_{i-1},j_{i+1},\ldots,j_U)$.

Lemma 6.12 (Abstract Sampling Game)**.** *Consider the game in Figure 6.4. Let $J = (J_1,\ldots,J_U)$ be uniformly distributed in $\{1,\ldots,N\}^U$, indicating the first entry sampled, and let $(V,I) = M(J_1,\ldots,J_U)$. Further, for all $1 \le i \le U$, let $A_i = a_i(J)$. Moreover, let X be the number of entries of the form $(1,i)$ with $i = I$ sampled (including the first one), and let Λ be the total number of entries sampled in this game. Then*

$$\mathbb{E}[\Lambda] \le 1 + (k-1)P \quad \textit{and}$$
$$\Pr(X = k) \ge \frac{N}{N-k+1}\left(\Pr(V=1) - P\cdot\frac{k-1}{N}\right),$$

where $P = \sum_{i=1}^{U} \Pr(A_i > 0)$.

Remark 6.4. Note the abstractly defined parameter P. In our application, where the index i of $(v,i) = M(j_1,\ldots,j_U)$ is determined by the output of a prover making no more than Q queries to the random oracle with function table $j_1,\ldots,j_U$, the parameter P will be bounded by $Q+1$. We show this formally (yet again somewhat abstractly) in Lemma 6.13. Intuitively, the reason is that the events $A_i > 0$ are *disjoint* for all but Q indices i (those that the considered prover does *not* query), and so their probabilities add up to at most 1. Indeed, if $a_i(j_1,\ldots,j_U) > 0$ for an index i that the algorithm did *not* query, then $M(j_1,\ldots,j_U) \in \{(0,i),(1,i)\}$; namely, since i has not been queried, the index i output by the algorithm is oblivious to the value of j_i. Therefore, given $j_1,\ldots,j_U$, there is at most one *unqueried* index i with $a_i(j_1,\ldots,j_U) > 0$.

Proof (of Lemma 6.12). **Expected Number of Samples.** Let us first derive an upper bound on the expected value of Λ. To this end, let X' denote the number of sampled entries of the form $(1,i)$ with $i = I$, but, in contrast to X, *without* counting the first one. Similarly, let Y' denote the number of sampled entries of the form (v,i) with $v = 0$ or $i \ne I$, again without counting the first one. Then $\Lambda = 1 + X' + Y'$ and

$$\Pr(X' = 0 \mid V = 0) = \Pr(Y' = 0 \mid V = 0) = 1 .$$

Figure 6.4: Abstract Sampling Game.

Parameters: $k, N, U \in \mathbb{N}$, and M a U-dimensional array with entries in $M(j_1, \ldots, j_U) \in \{0,1\} \times \{1, \ldots, U\}$ for all $1 \leq j_1, \ldots, j_U \leq N$.

- Sample $(j_1, \ldots, j_U) \in \{1, \ldots, N\}^U$ uniformly at random and set $(v, i) = M(j_1, \ldots, j_U)$.
- If $v = 0$, abort.
- Else, repeat
  - sample $j' \in \{1, \ldots, N\} \setminus \{j_i\}$ (without replacement),
  - compute $(v', i') = M(j_1, \ldots, j_{i-1}, j', j_{i+1}, \ldots, j_U)$,

  until either $k-1$ additional entries equal to $(1, i)$ have been found, or until all indices j' have been tried.

Hence, $\mathbb{E}[X' \mid V = 0] = \mathbb{E}[Y' \mid V = 0] = 0$.

Let us now consider the expected value $\mathbb{E}[Y' \mid V = 1]$. To this end, we observe that, conditioned on the event $V = 1 \wedge I = i \wedge A_i = a$ with $a > 0$, Y' follows a negative hypergeometric distribution with parameters $N-1$, $a-1$ and $k-1$. Hence, by Lemma 2.3,

$$\mathbb{E}[Y' \mid V = 1 \wedge I = i \wedge A_i = a] \leq (k-1)\frac{N-a}{a}\,,$$

and thus, using that $\Pr(X' \leq k-1 \mid V = 1) = 1$,

$$\mathbb{E}[X' + Y' \mid V = 1 \wedge I = i \wedge A_i = a] \leq (k-1) + (k-1)\frac{N-a}{a} = (k-1)\frac{N}{a}\,.$$

On the other hand

$$\Pr(V = 1 \wedge I = i \mid A_i = a) = \frac{a}{N}$$

and thus

$$\Pr(V = 1 \wedge I = i \wedge A_i = a) = \Pr(A_i = a)\frac{a}{N}\,. \tag{6.14}$$

Therefore, and since $\Pr(V = 1 \wedge I = i \wedge A_i = 0) = 0$,

$$\begin{aligned}
\Pr(V = 1) \cdot \mathbb{E}[X' + Y' \mid V = 1] &= \sum_{i=1}^{U}\sum_{a=1}^{N} \Pr(V = 1 \wedge I = i \wedge A_i = a) \\
&\qquad \cdot \mathbb{E}[X' + Y' \mid V = 1 \wedge I = i \wedge A_i = a] \\
&\leq \sum_{i=1}^{U}\sum_{a=1}^{N} \Pr(A_i = a)(k-1) \\
&= (k-1)\sum_{i=1}^{U} \Pr(A_i > 0) = (k-1)P\,,
\end{aligned}$$

where $P = \sum_{i=1}^{U} \Pr(A_i > 0)$. Hence,

$$\begin{aligned}\mathbb{E}[\Lambda] &= \mathbb{E}[1 + X' + Y'] \\ &= 1 + \Pr(V = 0) \cdot \mathbb{E}[X' + Y' \mid V = 0] + \Pr(V = 1) \cdot \mathbb{E}[X' + Y' \mid V = 1] \\ &\leq 1 + (k-1)P\,,\end{aligned}$$

which proves the claimed upper bound on $\mathbb{E}[\Lambda]$.

Success Probability. Let us now find a lower bound for the "success probability" $\Pr(X = k)$ of this game. Using (6.14) again, we can write

$$\Pr(X = k) = \sum_{i=1}^{U} \Pr(V = 1 \wedge I = i \wedge A_i \geq k) = \sum_{i=1}^{U} \sum_{a=k}^{N} \Pr(A_i = a) \frac{a}{N}\,.$$

Now, using $a \leq N$, note that

$$\begin{aligned}\frac{a}{N} &= 1 - \left(1 - \frac{a}{N}\right) \geq 1 - \frac{N}{N-k+1}\left(1 - \frac{a}{N}\right) \\ &= \frac{N}{N-k+1}\left(\frac{N-k+1}{N} - 1 + \frac{a}{N}\right) = \frac{N}{N-k+1}\left(\frac{a}{N} - \frac{k-1}{N}\right).\end{aligned}$$

Therefore, combining the two, and using that the summand becomes negative for $a < k$ to argue the second inequality, and using (6.14) once more, we obtain

$$\begin{aligned}\Pr(X = k) &\geq \sum_{i=1}^{U} \sum_{a=k}^{N} \Pr(A_i = a) \frac{N}{N-k+1} \left(\frac{a}{N} - \frac{k-1}{N}\right) \\ &\geq \sum_{i=1}^{U} \sum_{a=1}^{N} \Pr(A_i = a) \frac{N}{N-k+1} \left(\frac{a}{N} - \frac{k-1}{N}\right) \\ &= \frac{N}{N-k+1} \sum_{i=1}^{U} \sum_{a=1}^{N} \left(\Pr(V = 1 \wedge I = i \wedge A_i = a) - \Pr(A_i = a) \cdot \frac{k-1}{N}\right).\end{aligned}$$

Hence,

$$\begin{aligned}\Pr(X = k) &\geq \frac{N}{N-k+1} \left(\Pr(V = 1) - \frac{k-1}{N} \sum_{i=1}^{U} \Pr(A_i > 0)\right) \\ &= \frac{N}{N-k+1} \left(\Pr(V = 1) - P \cdot \frac{k-1}{N}\right),\end{aligned}$$

where, as before, we have used that $\Pr(V = 1 \wedge I = i \wedge A_i = 0) = 0$ for all $1 \leq i \leq U$, and finally that $P = \sum_{i=1}^{U} \Pr(A_i > 0)$. This completes the proof of the lemma. □

Our knowledge extractor will instantiate the abstract sampling game via a deterministic Q-query prover $\mathcal{P}^*$ attacking the Fiat-Shamir transformation $\mathsf{FS}[\Pi]$. The index i of $M(v, i) = (j_1, \ldots, j_U)$ is then determined by the output of $\mathcal{P}^*$, with the random oracle being given by the function table $j_1, \ldots, j_U$. Since the index i is thus determined by Q queries to the random oracle, the following shows that the parameter P will in this case be bounded by $Q + 1$.

Lemma 6.13. *Consider the game in Figure 6.4. Let v and idx be functions such that $M(j) = \big(v(j), \mathsf{idx}(j)\big)$ for all $j \in \{1,\dots,N\}^U$. Furthermore, let $J = (J_1,\dots,J_U)$ be uniformly distributed in $\{1,\dots,N\}^U$, and set $A_i = a_i(J)$ for all $1 \le i \le U$. Let us additionally assume that for all $j \in \{1,\dots,N\}^U$ there exists a subset $S(j) \subseteq \{1,\dots,U\}$ of cardinality at most Q such that $\mathsf{idx}(j) = \mathsf{idx}(j')$ for all j' with $j'_\ell = j_\ell$ for all $\ell \in S(j)$. Then*

$$P = \sum_{i=1}^{U} \Pr(A_i > 0) \le Q + 1\,.$$

Proof. By basic probability theory, it follows that[8]

$$\begin{aligned}
P &= \sum_{i=1}^{U} \Pr(A_i > 0) \\
&= \sum_{j \in \{1,\dots,N\}^U} \Pr(J = j) \sum_{i=1}^{U} \Pr(A_i > 0 \mid J = j) \\
&= \sum_{j} \Pr(J = j) \Bigg(\sum_{i \in S(j)} \Pr(A_i > 0 \mid J = j) + \sum_{i \notin S(j)} \Pr(A_i > 0 \mid J = j) \Bigg) \\
&\le \sum_{j} \Pr(J = j) \Bigg(Q + \sum_{i \notin S(j)} \Pr(A_i > 0 \mid J = j) \Bigg) \\
&= Q + \sum_{j} \Pr(J = j) \sum_{i \notin S(j)} \Pr(A_i > 0 \mid J = j)\,,
\end{aligned}$$

where the inequality follows from the fact that $|S(j)| \le Q$ for all j.

Now note that, by definition of the sets $S(j)$, for all $j \in \{1,\dots,N\}^U$, $i \notin S(j)$ and $j^* \in \{1,\dots,N\}$, it holds that

$$\Pr\big(\mathsf{idx}(J_1,\dots,J_{i-1},j^*,J_{i+1},\dots,J_U) = \mathsf{idx}(j) \mid J = j\big) = 1\,.$$

Therefore, for all $i \notin S(j) \cup \{\mathsf{idx}(j)\}$,

$$\Pr(A_i > 0 \mid J = j) = 0\,.$$

Hence,

$$\sum_{i \notin S(j)} \Pr(A_i > 0 \mid J = j) \le \Pr(A_{\mathsf{idx}(j)} > 0 \mid J = j) \le 1.$$

Altogether, it follows that

$$P \le Q + \sum_{j} \Pr(J = j) = Q + 1\,,$$

which completes the proof. □

[8] The probabilities $\Pr(A_i > 0 \mid J = j)$ are all 0 or 1; however, it's still convenient to use probability notation here.

6.6.4 The Fiat-Shamir Transformation of Σ-Protocols

Let us first consider the Fiat-Shamir transformation $\mathsf{FS}[\Pi]$ of a k-out-of-N special-sound Σ-protocol Π, i.e., a 3-round interactive proof with challenge set $\mathcal{C}$ of cardinality N. Subsequently, in Section 6.6.6, we move to general *multi-round* interactive proofs.

Let $\mathcal{P}^*$ be a deterministic dishonest Q-query random-oracle prover, attacking the Fiat-Shamir transformation $\mathsf{FS}[\Pi]$ of Π on input x. Given a statement x as input, after making Q queries to the random oracle $\mathsf{RO}\colon \{0,1\}^{\leq u} \to \mathcal{C}$, $\mathcal{P}^*$ outputs a proof $\pi = (a, z)$. For reasons to become clear later, we re-format (and partly rename) the output and consider $I := a$ and π as $\mathcal{P}^*$'s output. We refer to the output I as the *index*. Furthermore, we extend $\mathcal{P}^*$ to an algorithm $\mathcal{A}$ that additionally checks the correctness of the proof π. Formally, $\mathcal{A}$ runs $\mathcal{P}^*$ to obtain I and π, queries RO to obtain $c := \mathsf{RO}(I)$, and then outputs

$$I = a\,, \quad y := (a, c, z) \qquad \text{and} \qquad v := V(y)\,,$$

where $V(y) = 1$ if y is an accepting transcript for the interactive proof Π on input x and $V(y) = 0$ otherwise. Hence, $\mathcal{A}$ is a random-oracle algorithm making at most $Q + 1$ queries; indeed, it relays the oracle queries done by $\mathcal{P}^*$ and makes the one needed to do the verification. We may write $\mathcal{A}^{\mathsf{RO}}$ to make the dependency of $\mathcal{A}$'s output on the choice of the random oracle RO explicit. The random-oracle algorithm $\mathcal{A}$ has a naturally defined success probability

$$\epsilon(\mathcal{A}) := \Pr\bigl(v = 1 : (I, y, v) \leftarrow \mathcal{A}^{\mathsf{RO}}\bigr)\,,$$

where $\mathsf{RO}\colon \{0,1\}^{\leq u} \to \mathcal{C}$ is chosen uniformly at random. The probability $\epsilon(\mathcal{A})$ corresponds to the success probability $\epsilon(x, \mathcal{P}^*)$ of the random-oracle prover $\mathcal{P}^*$ on input x.

Our goal is now to construct an extraction algorithm that, when given oracle access to $\mathcal{A}$, aims to output k accepting transcripts $y_1, \ldots, y_k$ with common first message a and distinct challenges. By the k-out-of-N special-soundness of Π, a witness for statement x can be computed efficiently from these transcripts. Recall that an extractor with oracle access to a random oracle algorithm is free to choice the answers to the random oracle queries made by the algorithm. However, the answers provided by the extractor must be indistinguishable from those provided by a true random oracle algorithm.

The extractor $\mathcal{E}$ is defined in Figure 6.5. We note that, after a successful first run of $\mathcal{A}$, having produced a first accepting transcript (a, c, z), we rerun $\mathcal{A}$ from the very beginning and answer all oracle queries consistently, except the query to a; i.e., we only reprogram the oracle at the point $I = a$. Note that since $\mathcal{P}^*$ (and thus $\mathcal{A}$) is deterministic, and we only reprogram the oracle at the point $I = a$, in each iteration of the repeat loop $\mathcal{A}$ is ensured to make the query to I again.[9]

A crucial observation is the following. Within a run of $\mathcal{E}$, all the queries that are made by the different invocations of $\mathcal{A}$ are answered *consistently* using lazy sampling, except for the queries to the index I, where different responses $c, c', \ldots$

[9] Of course, it would be sufficient to rewind $\mathcal{A}$ to the point where it makes the (first) query to a, but this would make the description more clumsy.

Figure 6.5: Extractor $\mathcal{E}$ for Random Oracle Algorithms.

Parameters: $k, Q \in \mathbb{N}$.
Oracle access to: The $(Q+1)$-query random oracle algorithm $\mathcal{A}$ as above.

- Run $\mathcal{A}$ as follows to obtain (I, y_1, v): answer all (distinct) oracle queries with uniformly random values in $\mathcal{C}$. Let c be the response to query I.
- If $v = 0$, abort.
- Else, repeat
  - sample $c' \in \mathcal{C} \setminus \{c\}$ (without replacement);
  - run $\mathcal{A}$ as follows to obtain (I', y', v'): answer the query to I with c', while answering all other queries consistently if the query was performed by $\mathcal{A}$ already on a previous run, and with a fresh random value in $\mathcal{C}$ otherwise;

  until either $k-1$ additional challenges c' with $v' = 1$ and $I' = I$ have been found or until all challenges $c' \in \mathcal{C} \setminus \{c\}$ have been tried.
- In the former case, output the k accepting transcripts $y_1, \ldots, y_k$. In the latter case, the algorithm aborts.

are given. This is indistinguishable from having them answered by a full-fledged random oracle, i.e., by means of a pre-chosen function $\mathsf{RO}\colon \{0,1\}^{\leq u} \to \mathcal{C}$, but then replacing the output $\mathsf{RO}(I)$ at I by fresh challenges c' for the runs of $\mathcal{A}$ in the repeat loop. By enumerating the elements in the domain and codomain of RO, it is easily seen that the extractor is actually running the abstract game from Figure 6.4. Thus, bounds on the success probability and the expected run time (in terms of queries to $\mathcal{A}$) follow from Lemma 6.12 and Lemma 6.13. Altogether we obtain the following result.

Lemma 6.14 (Extractor for Random Oracle Algorithms)**.** *The extractor $\mathcal{E}$ of Figure 6.5 makes an expected number of at most $k + Q \cdot (k-1)$ queries to $\mathcal{A}$ and succeeds in outputting k transcripts $y_1, \ldots, y_k$ with common first message a and distinct challenges with probability at least*

$$\frac{N}{N-k+1}\left(\epsilon(\mathcal{A}) - (Q+1)\cdot\frac{k-1}{N}\right).$$

Proof. By enumerating all the elements in the domain and codomain of the random oracle RO, we may assume that $\mathsf{RO}\colon \{1,...,U\} \to \{1,...,N\}$, and thus RO can be represented by the function table $(j_1,...,j_U) \in \{1,\ldots,N\}^U$ for which $\mathsf{RO}(i) = j_i$. Further, since $\mathcal{P}^*$ is deterministic, the outputs I, y and v of the algorithm $\mathcal{A}$ can be viewed as functions taking as input the function table $(j_1,\ldots,j_U) \in \{1,\ldots,N\}^U$ of RO, and so we can consider the array $M(j_1,\ldots,j_U) = \big(I(j_1,\ldots,j_U), v(j_1,\ldots,j_U)\big)$.

Then, a run of the extractor perfectly matches up with the abstract sampling game of Figure 6.4 instantiated with array M. The only difference is that, in

this sampling game, we consider full-fledged random oracles encoded by vectors $(j_1, \ldots, j_U) \in \{1, \ldots, N\}^U$, while the actual extractor implements these random oracles by lazy sampling. Thus, we can apply Lemma 6.12 to obtain bounds on the success probability and the expected run time. However, in order to control the parameter P, which occurs in the bound of Lemma 6.12, we make the following observation, so that we can apply Lemma 6.13 to bound $P \leq Q + 1$.

For every $(j_1, \ldots, j_U)$, let $S(j_1, \ldots, j_U) \subseteq \{1, \ldots, U\}$ be the set of points that $\mathcal{P}^*$ queries to the random oracle when $(j_1, \ldots, j_U)$ corresponds to the entire function table of the random oracle. Then, $\mathcal{P}^*$ will produce the same output when the random oracle is reprogrammed at an index $i \notin S(j_1, \ldots, j_U)$. In particular, $I(j_1, \ldots, j_{i-1}, j, j_{i+1}, \ldots, j_U) = I(j_1, \ldots, j_{i-1}, j', j_{i+1}, \ldots, , j_U)$ for all j, j' and for all $i \notin S(j_1, \ldots, j_U)$. Furthermore, $|S(j_1, \ldots, j_U)| \leq Q$. Hence, the conditions of Lemma 6.13 are satisfied and $P \leq Q+1$. The bounds on the success probability and the expected run time now follow, completing the proof. □

The existence of the above extractor, combined with the k-out-of-N special-soundness property, implies the following theorem. This theorem shows that the security loss of the Fiat-Shamir transformation for k-out-of-N Σ-protocols is $Q+1$, i.e., the security loss is linear in the query complexity Q of a prover $\mathcal{P}^*$ attacking the Fiat-Shamir transformation.

Theorem 6.10 (Fiat-Shamir Transformation of a Σ-Protocol)**.** *The Fiat-Shamir transformation* $\mathsf{FS}[\Pi]$ *of a k-out-of-N special-sound Σ-protocol* Π *is knowledge sound with knowledge error*

$$\kappa_{\mathrm{fs}}(Q) = (Q + 1) \cdot \kappa\,,$$

where $\kappa := \mathrm{Er}(k; N) = (k - 1)/N$ *is the knowledge error of the (interactive) Σ-protocol* Π*.*

6.6.5 A Refined Analysis of the Abstract Sampling Game

Before we prove knowledge soundness of the Fiat-Shamir transformation of *multi-round* interactive protocols, we reconsider the abstract game of Section 6.6.3, and present a refined analysis of the cost of playing the game. The multi-round knowledge extractor will essentially play a recursive composition of this game; however, the analysis of Section 6.6.3 is insufficient for our purposes (resulting in a super-polynomial bound on the run-time of the knowledge extractor). Fortunately, it turns out that a refinement allows us to prove the required (polynomial) upper bound.

In Section 6.6.3, the considered cost measure is the number of entries visited during the game. For Σ-protocols, every entry corresponds to a single invocation of the dishonest prover $\mathcal{P}^*$. For multi-round protocols, every entry will correspond to a single invocation of a sub-tree extractor. The key observation is that some invocations of the sub-tree extractor are expensive while others are *cheap*. For this reason, we introduce a cost function Γ and a constant cost γ to our abstract game, allowing us to differentiate between these two cases. Γ and γ assign a cost to every entry of the array M; Γ corresponds to the cost of an expensive invocation of the sub-tree extractor, and γ corresponds to the cost of a cheap invocation. While this

refinement presents a natural generalization of the abstract game of Section 6.6.3, its analysis becomes significantly more involved.

The following lemma provides an upper bound for the total cost of playing the abstract game in terms of these two cost functions.

Lemma 6.15 (Abstract Sampling Game - Weighted Version)**.** *Consider again the game of Figure 6.4, as well a cost function* $\Gamma\colon \{1,\dots,N\}^U \to \mathbb{R}_{\geq 0}$ *and a constant cost* $\gamma \in \mathbb{R}_{\geq 0}$*. Let* $J = (J_1,\dots,J_U)$ *be uniformly distributed in* $\{1,\dots,N\}^U$*, indicating the first entry sampled, and let* $(V,I) = M(J_1,\dots,J_U)$*. Further, for all* $1 \leq i \leq U$*, let* $A_i = a_i(J)$*, where the function* a_i *is as defined in Equation 6.13.*

We define the cost of sampling an entry $M(j_1,\dots,j_U) = (v,i)$ *with index* $i = I$ *to be* $\Gamma(j_1,\dots,j_U)$ *and the cost of sampling an entry* $M(j_1,\dots,j_U) = (v,i)$ *with index* $i \neq I$ *to be* γ*. Let* Δ *be the total cost of playing this game. Then*

$$\mathbb{E}[\Delta] \leq k \cdot \mathbb{E}[\Gamma(J)] + (k-1) \cdot T \cdot \gamma$$

where $T = \sum_{i=1}^{U} \Pr(I \neq i \wedge A_i > 0) \leq P$*.*

Remark 6.5. Note that the parameter T in the statement here differs slightly from its counterpart $P = \sum_i \Pr(A_i > 0)$ in Lemma 6.12. Recall the informal discussion of P in the context of our application (Remark 6.4), where the array M is instantiated via a Q-query prover $\mathcal{P}^*$ attacking the Fiat-Shamir transformation of an interactive proof. We immediately see that now the defining events $I \neq i \wedge A_i > 0$ are *empty* for all $U - Q$ indices that the prover does not query, giving the bound $T \leq Q$ here, compared to the bound $P \leq Q + 1$ on P. The formal (and more abstract) statement and proof is given in Lemma 6.16.

Proof. Let us split up Δ into the cost measures Δ_1, Δ_2 and Δ_3, defined as follows. Δ_1 denotes the total costs of the elements $M(j_1,\dots,j_U) = (1,i)$ with $i = I$ sampled in the game, i.e., the elements with bit $v = 1$ and index $i = I$; correspondingly, X denotes the number of entries of the form $(1,i)$ with $i = I$ sampled (including the first one if $V = 1$). Second, Δ_2 denotes the total costs of the elements $M(j_1,\dots,j_U) = (0,i)$ with $i = I$ sampled, i.e., the elements with bit $v = 0$ and index $i = I$; correspondingly, Y denotes the number of entries of the form $(0,i)$ with $i = I$ sampled (including the first one if $V = 0$). Finally, Δ_3 denotes the total costs of the elements $M(j_1,\dots,j_U) = (v,i)$ with $i \neq I$ sampled; correspondingly, Z denotes the number of entries of this form sampled.

Clearly $\Delta = \Delta_1 + \Delta_2 + \Delta_3$. Moreover, since the cost γ is constant, it follows that $\mathbb{E}[\Delta_3] = \gamma \cdot \mathbb{E}[Z]$. In a similar manner, we now aim to relate $\mathbb{E}[\Delta_1]$ and $\mathbb{E}[\Delta_2]$ to $\mathbb{E}[Y]$ and $\mathbb{E}[Z]$, respectively. However, since the cost function $\Gamma\colon \{1,\dots,N\}^U \to \mathbb{R}_{\geq 0}$ is not necessarily constant, this is more involved.

For $1 \leq i \leq U$ let us write $J_i^* = (J_1,\dots,J_{i-1},J_{i+1},\dots,J_U)$, which is uniformly random with support $\{1,\dots,N\}^{U-1}$. Moreover, for all $1 \leq i \leq U$ and $j^* = (j_1^*,\dots,j_{i-1}^*,j_{i+1}^*,\cdots,j_U) \in \{1,\dots,N\}^{U-1}$, let $\Lambda(i,j^*)$ denote the event

$$\Lambda(i,j^*) = [I = i \wedge J_i^* = j^*]\,.$$

We note that conditioned on the event $\Lambda(i,j^*)$, all samples are picked from the subarray $M(j_1^*,\dots,j_{i-1}^*,\cdot,j_{i+1}^*,\cdots,j_U^*)$; the first one uniformly at random subject to the index I being i, and the remaining ones (if $V = 1$) uniformly at random (without replacement).

We first analyze and bound $\mathbb{E}[\Delta_1 \mid \Lambda(i,j^*)]$. We observe that, for all i and j^* with $\Pr(\Lambda(i,j^*)) > 0$,

$$\mathbb{E}[\Delta_1 \mid \Lambda(i,j^*)] = \sum_{\ell=0}^{N} \Pr\big(X = \ell \mid \Lambda(i,j^*)\big) \cdot \mathbb{E}[\Delta_1 \mid \Lambda(i,j^*) \wedge X = \ell]\,.$$

Since, conditioned on $\Lambda(i,j^*) \wedge X = \ell$ for $\ell \in \{0,\ldots,N\}$, any size-ℓ subset of elements with $v = 1$ and index i is equally likely to be sampled, it follows that

$$\mathbb{E}[\Delta_1 \mid \Lambda(i,j^*) \wedge X = \ell] = \mathbb{E}[\Gamma(J) \mid V = 1 \wedge \Lambda(i,j^*)] \cdot \ell\,.$$

Hence,

$$\begin{aligned}\mathbb{E}[\Delta_1 \mid \Lambda(i,j^*)] &= \mathbb{E}[\Gamma(J) \mid V = 1 \wedge \Lambda(i,j^*)] \cdot \sum_{\ell} \Pr\big(X = \ell \mid \Lambda(i,j^*)\big) \cdot \ell \\ &= \mathbb{E}[\Gamma(J) \mid V = 1 \wedge \Lambda(i,j^*)] \cdot \mathbb{E}[X \mid \Lambda(i,j^*)]\,.\end{aligned}$$

Similarly,

$$\mathbb{E}[\Delta_2 \mid \Lambda(i,j^*)] = \mathbb{E}[\Gamma(J) \mid V = 0 \wedge \Lambda(i,j^*)] \cdot \mathbb{E}[Y \mid \Lambda(i,j^*)]\,.$$

Next, we bound the expected values of X and Y conditioned on $\Lambda(i,j^*)$. The analysis is a more fine-grained version of the proof of Lemma 6.12. Bounding $\mathbb{E}[X \mid \Lambda(i,j^*)]$ is quite easy: since $V = 0$ implies $X = 0$ and $V = 1$ implies $X \le k$, it immediately follows that

$$\begin{aligned}\mathbb{E}[X \mid \Lambda(i,j^*)] &= \Pr(V = 0 \mid \Lambda(i,j^*)) \cdot \mathbb{E}[X \mid V = 0 \wedge \Lambda(i,j^*)] \\ &\qquad + \Pr(V = 1 \mid \Lambda(i,j^*)) \cdot \mathbb{E}[X \mid V = 1 \wedge \Lambda(i,j^*)] \\ &\le \Pr(V = 1 \mid \Lambda(i,j^*)) \cdot k\,.\end{aligned}$$

Hence,

$$\mathbb{E}[\Delta_1 \mid \Lambda(i,j^*)] \le k \cdot \Pr(V = 1 \mid \Lambda(i,j^*)) \cdot \mathbb{E}[\Gamma(J) \mid V = 1 \wedge \Lambda(i,j^*)]\,. \qquad (6.15)$$

Suitably bounding the expectation $\mathbb{E}[Y \mid \Lambda(i,j^*)]$, and thus $\mathbb{E}[\Delta_2 \mid \Lambda(i,j^*)]$, is more involved. For that purpose, we introduce the following parameters. For the considered fixed choice of the index $1 \le i \le U$ and of $j^* = (j_1^*,\ldots,j_{i-1}^*,j_{i+1}^*,\cdots,j_U^*)$, we let[10]

$$\begin{aligned}a &:= a_i(j^*) = \big|\{j :\ (v_j,i_j) = M(j_1^*,\ldots,j_{i-1}^*,j,j_{i+1}^*,\ldots,j_U^*) = (1,i)\ \}\big| \quad \text{and} \\ b &:= b_i(j^*) := \big|\{j :\ (v_j,i_j) = M(j_1^*,\ldots,j_{i-1}^*,j,j_{i+1}^*,\ldots,j_U^*) = (0,i)\ \}\big|\,.\end{aligned}$$

Let us first note that

$$\Pr\big(V = 1 \mid \Lambda(i,j^*)\big) = \frac{a}{a+b} \quad \text{and} \quad \Pr\big(V = 0 \mid \Lambda(i,j^*)\big) = \frac{b}{a+b}$$

[10] Recall that we use the notation $a_i(j_1,\ldots,j_U)$ and $a_i(j_1,\ldots,j_{i-1},j_{i+1},\ldots,j_U)$ interchangeably, exploiting that $a_i(j_1,\ldots,j_U)$ does not depend on the i-th input j_i.

for all i and j^* with $\Pr\big(\Lambda(i,j^*)\big) > 0$. Therefore, if we condition on the event $V = 1 \wedge \Lambda(i,j^*)$ we implicitly assume that i and j^* are so that a is positive. Now, towards bounding $\mathbb{E}[Y \mid \Lambda(i,j^*)]$, we observe that conditioned on the event $V = 1 \wedge \Lambda(i,j^*)$, the random variable Y follows a negative hypergeometric distribution with parameters $a+b-1$, $a-1$ and $k-1$ (see also Remark 2.2). Hence, by Lemma 2.3,

$$\mathbb{E}[Y \mid V = 1 \wedge \Lambda(i,j^*)] \leq (k-1)\frac{b}{a}\,,$$

and thus

$$\begin{aligned}
\mathbb{E}[Y \mid \Lambda(i,j^*)] &= \Pr(V = 0 \mid \Lambda(i,j^*)) \cdot \mathbb{E}[Y \mid V = 0 \wedge \Lambda(i,j^*)] \\
&\quad + \Pr(V = 1 \mid \Lambda(i,j^*)) \cdot \mathbb{E}[Y \mid V = 1 \wedge \Lambda(i,j^*)] \\
&\leq \Pr\big(V = 0 \mid \Lambda(i,j^*)\big) + \Pr\big(V = 1 \mid \Lambda(i,j^*)\big) \cdot (k-1)\frac{b}{a} \\
&= \frac{b}{a+b} + \frac{a}{a+b} \cdot (k-1)\frac{b}{a} = k\frac{b}{a+b} \\
&= k \cdot \Pr(V = 0 \mid \Lambda(i,j^*))\,,
\end{aligned}$$

where we use that $\mathbb{E}[Y \mid V = 0 \wedge \Lambda(i,j^*)] = 1$. Hence,

$$\mathbb{E}[\Delta_2 \mid \Lambda(i,j^*)] \leq k \cdot \Pr(V = 0 \mid \Lambda(i,j^*)) \cdot \mathbb{E}[\Gamma(J) \mid V = 0 \wedge \Lambda(i,j^*)]\,,$$

and thus, combined with Equation 6.15,

$$\mathbb{E}[\Delta_1 + \Delta_2 \mid \Lambda(i,j^*)] \leq k \cdot \mathbb{E}[\Gamma(J) \mid \Lambda(i,j^*)]\,.$$

Since this inequality holds for all i and j^* with $\Pr\big(\Lambda(i,j^*)\big) > 0$, it follows that

$$\mathbb{E}[\Delta_1 + \Delta_2] \leq k \cdot \mathbb{E}[\Gamma(J)]\,.$$

What remains is to show that $\mathbb{E}[Z] \leq (k-1)T$, from which it follows that $\mathbb{E}[\Delta_3] = \gamma\mathbb{E}[Z] \leq (k-1)T\gamma$. The slightly weaker bound $\mathbb{E}[Z] \leq (k-1)P$ follows immediately from observing that $Z \leq Y'$ for Y' as in the proof of Lemma 6.12 (the number of entries counted by Z is a subset of those counted by Y'), and using that $\mathbb{E}[Y'] \leq \mathbb{E}[X' + Y'] \leq (k-1)P$ as derived in the proof of Lemma 6.12. In order to get the slightly better bound in terms of T, we bound $\mathbb{E}[Z]$ from scratch below. We use a similar approach as above for bounding the expectation of Y. Thus, we consider a fixed choice of i and j^* and set $a := a_i(j^*)$ and $b := b_i(j^*)$. Then, conditioned on $V = 1 \wedge \Lambda(i,j^*)$, also Z follows a negative hypergeometric distribution, but now with parameters $N-b-1$, $a-1$ and $k-1$. Therefore, for all i and j^* with $\Pr\big(V = 1 \wedge \Lambda(i,j^*)\big) > 0$,

$$\mathbb{E}[Z \mid V = 1 \wedge \Lambda(i,j^*)] \leq (k-1)\frac{N-a-b}{a}\,.$$

Using that $\mathbb{E}[Z \mid V = 0 \wedge \Lambda(i,j^*)] = 0$, but also recalling that $\Pr\big(V = 1 \mid \Lambda(i,j^*)\big) = a/(a+b)$ and exploiting $\Pr(I = i \mid J_i^* = j^*) = (a+b)/N$,

it follows that

$$\begin{aligned}
\mathbb{E}[Z \mid \Lambda(i,j^*)] &= \Pr\big(V=1 \mid \Lambda(i,j^*)\big) \cdot \mathbb{E}[Z \mid V=1 \wedge \Lambda(i,j^*)] \\
&\leq \frac{a}{a+b} \cdot (k-1) \cdot \frac{N-a-b}{a} = (k-1) \cdot \frac{N-a-b}{a+b} \\
&= (k-1) \cdot \Big(\frac{1}{\Pr(I=i \mid J_i^*=j^*)} - 1 \Big) \\
&= (k-1) \cdot \frac{\Pr(J_i^*=j^*) - \Pr(I=i \wedge J_i^*=j^*)}{\Pr(I=i \wedge J_i^*=j^*)} \\
&= (k-1) \cdot \frac{\Pr(I \neq i \wedge J_i^*=j^*)}{\Pr(I=i \wedge J_i^*=j^*)} = (k-1) \cdot \frac{\Pr(I \neq i \wedge J_i^*=j^*)}{\Pr(\Lambda(i,j^*))} \,.
\end{aligned}$$

We recall that the above holds for all i and j^* for which $a = a_i(j^*) > 0$, so that $\Pr(V=1 \wedge \Lambda(i,j^*)) > 0$. For i and j^* with $a = a_i(j^*) = 0$, it holds that $\Lambda(i,j^*)$ implies $V=0$, and thus $\mathbb{E}[Z \mid \Lambda(i,j^*)] = 0$. Therefore

$$\begin{aligned}
\mathbb{E}[Z] &= \sum_{i=1}^{U} \sum_{\substack{j^* \text{ s.t.} \\ a_i(j^*)>0}} \Pr[\Lambda(i,j^*)] \cdot \mathbb{E}[Z \mid \Lambda(i,j^*)] \\
&\leq (k-1) \cdot \sum_{i=1}^{U} \sum_{\substack{j^* \text{ s.t.} \\ a_i(j^*)>0}} \Pr(I \neq i \wedge J_i^*=j^*) \\
&\leq (k-1) \cdot \sum_{i=1}^{U} \Pr(I \neq i \wedge A_i > 0) = (k-1) \cdot T \,.
\end{aligned}$$

Hence $\mathbb{E}[\Delta_3] \leq (k-1) \cdot T \cdot \gamma$, as intended, and altogether it follows that

$$\mathbb{E}[\Delta] = \mathbb{E}[\Delta_1 + \Delta_2 + \Delta_3] \leq k \cdot \mathbb{E}[\Gamma(J)] + (k-1) \cdot T \cdot \gamma \,,$$

which completes the proof of the lemma.

□

Lemma 6.16. *Consider the game in Figure 6.4. Let v and idx be functions such that $M(j) = \big(v(j), \mathsf{idx}(j)\big)$ for all $j \in \{1,\dots,N\}^U$. Furthermore, let $J = (J_1,\dots,J_U)$ be uniformly distributed in $\{1,\dots,N\}^U$ and set $A_i = a_i(J)$ for all $1 \leq i \leq U$ as in Equation 6.13. Let us additionally assume that for all $j \in \{1,\dots,N\}^U$ there exists a subset $S(j) \subseteq \{1,\dots,U\}$ of cardinality at most Q such that $\mathsf{idx}(j) = \mathsf{idx}(j')$ for all j, j' with $j_\ell = j'_\ell$ for all $\ell \in S(j)$. Then*

$$T = \sum_{i=1}^{U} \Pr\big(\mathsf{idx}(J) \neq i \wedge A_i > 0\big) \leq Q \,.$$

Proof. The proof is analogous to the proof of Lemma 6.13. By basic probability

theory, it follows that

$$\begin{aligned} T &= \sum_{i=1}^{U} \Pr(\mathsf{idx}(J) \neq i \wedge A_i > 0) \\ &= \sum_{j} \Pr(J = j) \Bigg(\sum_{i \in S(j)} \Pr(\mathsf{idx}(J) \neq i \wedge A_i > 0 \mid J = j) \\ &\qquad\qquad + \sum_{i \notin S(j)} \Pr(\mathsf{idx}(J) \neq i \wedge A_i > 0 \mid J = j) \Bigg) \\ &\leq Q + \sum_{j} \Pr(J = j) \sum_{i \notin S(j)} \Pr(\mathsf{idx}(J) \neq i \wedge A_i > 0 \mid J = j)\,, \end{aligned}$$

where the inequality follows from the fact that $|S(j)| \leq Q$ for all j.

Now note that, by definition of the sets $S(j)$, for all $j \in \{1, \ldots, N\}^U$, $i \notin S(j)$ and $j_i \in \{1, \ldots, N\}$, it holds that

$$\Pr\big(\mathsf{idx}(J_1, \ldots, J_{i-1}, j_i, J_{i+1}, \ldots, J_U) = \mathsf{idx}(j) \mid J = j\big) = 1\,.$$

Therefore, for all $i \notin S(j) \cup \{\mathsf{idx}(j)\}$,

$$\Pr(A_i > 0 \mid J = j) = 0\,.$$

Hence,

$$\begin{aligned} &\sum_{i \notin S(j)} \Pr\big(\mathsf{idx}(J) \neq i \wedge A_i > 0 \mid J = j\big) \\ &\qquad \leq \Pr\big(\mathsf{idx}(J) \neq \mathsf{idx}(j) \wedge A_{\mathsf{idx}(j)} > 0 \mid J = j\big) = 0\,. \end{aligned}$$

Altogether, it follows that

$$T \leq Q + \sum_{j} \Pr(J = j) \sum_{i \notin S(j)} \Pr\big(\mathsf{idx}(J) \neq i \wedge A_i > 0 \mid J = j\big) = Q\,,$$

which completes the proof. □

6.6.6 The Fiat-Shamir Transformation of Multi-Round Protocols

Let us now move to multi-round interactive proofs. More precisely, we consider the Fiat-Shamir transformation $\mathsf{FS}[\Pi]$ of a $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound $(2\mu + 1)$-round interactive proof Π, with $\mathbf{k} = (k_1, \ldots, k_\mu)$. While the multi-round extractor has a natural recursive construction, it requires a more fine-grained analysis to show that it indeed implies knowledge soundness.

To avoid a cumbersome notation, we first handle $(2\mu + 1)$-round interactive proofs in which the verifier samples all μ challenges uniformly at random from the *same* set $\mathcal{C}$. Subsequently, we argue that our techniques have a straightforward generalization to interactive proofs where the verifier samples its challenges from different challenge sets.

Multi-Round Interactive Proofs with a Single Challenge Set

Consider a deterministic dishonest Q-query random-oracle prover $\mathcal{P}^*$, attacking the Fiat-Shamir transformation $\mathsf{FS}[\Pi]$ of a $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proof Π on input x. We assume all challenges to be elements of the same set $\mathcal{C}$. After making at most Q queries to the random oracle, $\mathcal{P}^*$ outputs a proof $\pi = (a_1, \ldots, a_{\mu+1})$. We re-format the output and consider

$$I_1 := a_1\,,\ I_2 := (a_1, a_2)\,, \ldots,\ I_\mu := (a_1, \ldots, a_\mu) \quad \text{and} \quad \pi$$

as $\mathcal{P}^*$'s output. Sometimes it will be convenient to also consider

$$I_{\mu+1} := (a_1, \ldots, a_{\mu+1})\,.$$

Furthermore, we extend $\mathcal{P}^*$ to a random-oracle algorithm $\mathcal{A}$ that additionally checks the correctness of the proof π. Formally, relaying all the random oracle queries that $\mathcal{P}^*$ is making, $\mathcal{A}$ runs $\mathcal{P}^*$ to obtain $\mathbf{I} = (I_1, \ldots, I_\mu)$ and π, additionally queries the random oracle to obtain $c_1 := \mathsf{RO}(I_1), \ldots, c_\mu := \mathsf{RO}(I_\mu)$, and then outputs

$$\mathbf{I}\,, \quad y := (a_1, c_1, \ldots, a_\mu, c_\mu, a_{\mu+1}) \qquad \text{and} \qquad v := V(x, y)\,,$$

where $V(x, y) = 1$ if y is an accepting transcript for the interactive proof Π on input x, and $V(x, y) = 0$ otherwise. Hence, $\mathcal{A}$ makes at most $Q + \mu$ queries (the queries done by $\mathcal{P}^*$, and the queries to $I_1, \ldots, I_\mu$). Moreover, $\mathcal{A}$ has a naturally defined success probability

$$\epsilon(\mathcal{A}) := \Pr\bigl(v = 1 : (I, y, v) \leftarrow \mathcal{A}^{\mathsf{RO}}\bigr)\,,$$

where $\mathsf{RO}\colon \{0,1\}^{\leq u} \to \mathcal{C}$ is distributed uniformly. As before, $\epsilon(\mathcal{A}) = \epsilon(x, \mathcal{P}^*)$.

Our goal is now to construct an extraction algorithm that, when given oracle access to $\mathcal{A}$, and thus to $\mathcal{P}^*$, aims to output a $\mathbf{k}$-tree of accepting transcripts (Definition 2.33). By the $\mathbf{k}$-out-of-$\mathbf{N}$ special-soundness of Π, a witness for statement x can then be computed efficiently from these transcripts.

To this end, we recursively introduce a sequence of "sub-extractors" $\mathcal{E}_1, \ldots, \mathcal{E}_\mu$, where $\mathcal{E}_m$ aims to find a $(1, \ldots, 1, k_m, \ldots, k_\mu)$-tree of accepting transcripts. The main idea behind this recursion is that such a $(1, \ldots, 1, k_m, \ldots, k_\mu)$-tree of accepting transcripts is the composition of k_m appropriate $(1, \ldots, 1, k_{m+1}, \ldots, k_\mu)$-trees.

For technical reasons, we define the sub-extractors $\mathcal{E}_m$ as *random-oracle* algorithms, each one making $Q + \mu$ queries to a random oracle. As we will see, the recursive definition of $\mathcal{E}_m$ is very much like the extractor from the 3-round case, but with $\mathcal{A}$ replaced by the sub-extractor $\mathcal{E}_{m+1}$; however, for this to work we need the sub-extractor to be the same kind of object as $\mathcal{A}$, thus a random-oracle algorithm making the same number of queries. As base for the recursion, we consider the algorithm $\mathcal{A}$ (which outputs a single transcript, i.e., a $(1, \ldots, 1)$-tree); thus, the sub-extractor $\mathcal{E}_\mu$ (which outputs a $(1, \ldots, 1, k_\mu)$-tree) is essentially the extractor of the 3-round case, but with $\mathcal{A}$ now outputting an index *vector* $\mathbf{I} = (I_1, \ldots, I_\mu)$, and with $\mathcal{E}_\mu$ being a *random-oracle* algorithm, so that we can recursively replace the random-oracle algorithm $\mathcal{A}$ by $\mathcal{E}_\mu$ to obtain $\mathcal{E}_{\mu-1}$, etc.

Figure 6.6: Sub-extractor $\mathcal{E}_m$, as a $(Q+\mu)$-query random-oracle algorithm.

Parameters: $k_m, Q \in \mathbb{N}$.
Oracle access to: $\mathcal{E}_{m+1}$.
Random oracle queries: $\leq Q + \mu$.

- Run $\mathcal{E}_{m+1}$ as follows to obtain $(\mathbf{I}, y_1, v)$: relay the $Q+\mu$ queries to the random oracle and record all query-response pairs. Let c be the response to query I_m.
- If $v = 0$, abort with output $v = 0$.
- Else, repeat
  - sample $c' \in \mathcal{C} \setminus \{c\}$ (without replacement);
  - run $\mathcal{E}_{m+1}$ as follows to obtain $(\mathbf{I}', y', v')$, aborting right after the initial run of $\mathcal{P}^*$ if $I'_m \neq I_m$: answer the query to I_m with c', while answering all other queries consistently if the query was performed by $\mathcal{E}_{m+1}$ already on a previous run and with a fresh random value in $\mathcal{C}$ otherwise;

  until either $k_m - 1$ additional challenges c' with $v' = 1$ and $I'_m = I_m$ have been found or until all challenges $c' \in \mathcal{C} \setminus \{c\}$ have been tried.
- In the former case, output $\mathbf{I}$, the k_m accepting $(1, \ldots, 1, k_{m+1}, \ldots, k_\mu)$-trees $y_1, \ldots, y_{k_m}$, and $v := 1$; in the latter case, output $v := 0$.

Formally, the recursive definition of $\mathcal{E}_m$ from $\mathcal{E}_{m+1}$ is given in Figure 6.6, where $\mathcal{E}_{\mu+1}$ (the base case) is set to $\mathcal{E}_{\mu+1} := \mathcal{A}$, and where $\mathcal{E}_m$ exploits the following *early abort* feature of $\mathcal{E}_{m+1}$: like $\mathcal{A}$, the sub-extractor $\mathcal{E}_{m+1}$ computes the index vector it eventually outputs by running $\mathcal{P}^*$ *as its first step* (see Lemma 6.17 below). This allows the executions of $\mathcal{E}_{m+1}$ in the repeat loop in Fig. 6.6 to abort after a single run of $\mathcal{P}^*$ if the requirement $I'_m = I_m$ on its index vector $\mathbf{I}$ is not satisfied, without proceeding to produce the remaining parts y', v' of the output (which would invoke more calls to $\mathcal{P}^*$).

The actual extractor $\mathcal{E}$ is then given by a run of $\mathcal{E}_1$, with the $Q+\mu$ random-oracle queries made by $\mathcal{E}_1$ being answered using lazy-sampling.

Remark 6.6. Let us emphasize that within *one* run of $\mathcal{E}_m$, except for the query to I_m for which the response is "reprogrammed," all the queries made by the multiple runs of the sub-extractor $\mathcal{E}_{m+1}$ in the repeat loop are answered *consistently*, both with the run of $\mathcal{E}_{m+1}$ in the first step and among the runs in the repeat loop. This means that a query to a value ξ that has been answered by η in a previous run on $\mathcal{E}_{m+1}$ (within the considered run of $\mathcal{E}_m$) is again answered by η, and a query to a value ξ' that has not been queried yet in a previous run on $\mathcal{E}_{m+1}$ (within the considered run of $\mathcal{E}_m$) is answered with a freshly chosen uniformly random $\eta' \in \mathcal{C}$. In *multiple* runs of $\mathcal{E}_m$, very naturally the random tape of $\mathcal{E}_m$ will be refreshed, and thus there is no guaranteed consistency among the answers to the query calls of $\mathcal{E}_{m+1}$ across multiple runs of $\mathcal{E}_m$.

The following lemma captures some technical property of the sub-extractors $\mathcal{E}_m$. Subsequently, Proposition 6.1 shows that $\mathcal{E}_m$, if successful, indeed outputs a $(1,\dots,1,k_m\dots,k_\mu)$-tree of accepting transcripts. Proposition 6.2 bounds the success probability and expected run time of $\mathcal{E}_m$. All statements are understood to hold for any statement x and any $m \in \{1,\dots,\mu+1\}$.

Lemma 6.17 (Consistency of $\mathcal{P}^*$ and $\mathcal{E}_m$)**.** *$\mathcal{E}_m$ obtains the index vector $\mathbf{I}$, which it eventually outputs, by running $(\mathbf{I},\pi) \leftarrow \mathcal{P}^*$ as its first step. In particular, for any fixed choice of the random oracle RO, the index vector $\mathbf{I}$ output by $\mathcal{E}_m^{\mathsf{RO}}$ matches the one output by $\mathcal{P}^{*,\mathsf{RO}}$.*

Proof. The first claim holds for $\mathcal{E}_{\mu+1} = \mathcal{A}$ by definition of $\mathcal{A}$, and it holds for $\mathcal{E}_m$ with $m \le \mu$ by induction, given that $\mathcal{E}_m$ runs $\mathcal{E}_{m+1}$ as a first step. The claim on the matching index vectors then follows trivially. □

Proposition 6.1 (Correctness)**.** *For any fixed choice of the random oracle let $(\mathbf{I}, y_1,\dots,y_{k_m}, v) \leftarrow \mathcal{E}_m^{\mathsf{RO}}(x)$. If $v = 1$ then $(y_1,\dots,y_{k_m})$ forms a $(1,\dots,1,k_m,\dots,k_\mu)$-tree of accepting transcripts.*

Proof. All $\prod_{j=m+1}^{\mu} k_j$ transcripts in a $(1,\dots,1,k_{m+1},\dots,k_\mu)$-tree contain the same partial transcript $(a_1,c_1,\dots,c_m,a_{m+1})$, i.e., the first $2m-1$ messages in all these transcripts coincide. Hence, any $(1,\dots,1,k_{m+1},\dots,k_\mu)$-tree of transcripts has a well-defined *trunk* $(a_1,c_1,\dots,c_m,a_{m+1})$.

By induction on m, we will prove that if $v = 1$ then $(y_1,\dots,y_{k_m})$ forms a $(1,\dots,1,k_m,\dots,k_\mu)$-tree of accepting transcripts with trunk $(a_1,\mathsf{RO}(I_1),\dots,\mathsf{RO}(I_{m-1}),a_m)$, where $I_j = (a_1,\dots,a_j)$. This obviously implies the correctness claim.

For the base case $m = \mu+1$, recall that $\mathcal{E}_{\mu+1} = \mathcal{A}$, and that by definition of $\mathcal{A}$ and its output $(\mathbf{I},y,v)$, if $v = 1$, then y is an accepting transcript, and thus a $(1,\dots,1)$-tree of accepting transcripts with $(a_1,\mathsf{RO}(I_1),\dots,\mathsf{RO}(I_\mu),a_{\mu+1})$ as trunk by definition of $\mathbf{I} = (I_1,\dots,I_\mu)$.

For the induction step, by the induction hypothesis on $\mathcal{E}_{m+1}$ and its output $(\mathbf{I},y,v)$, if $v = 1$, then y is a $(1,\dots,1,k_{m+1},\dots,k_\mu)$-tree of accepting transcripts with trunk $(a_1,\mathsf{RO}(I_1),\dots,a_m,\mathsf{RO}(I_m),a_{m+1})$, where $I_{m+1} = (a_1,\dots,a_{m+1})$. This holds for $(\mathbf{I},y_1,v)$ output by $\mathcal{E}_{m+1}$ in the first step of $\mathcal{E}_m$, but also for any invocation of $\mathcal{E}_{m+1}$ in the repeat loop with output $(\mathbf{I}',y',v')$, here with trunk $(a_1',\mathsf{RO}'(I_1'),\dots,a_m',\mathsf{RO}'(I_m'),a_{m+1}')$, where RO' is such that $\mathsf{RO}'(I_j) = \mathsf{RO}(I_j)$ for all $j \ne m$, while $\mathsf{RO}(I_m) = c_i$ and $\mathsf{RO}'(I_m) = c_i'$. By definition of the output of $\mathcal{E}_m$, for y_1 and y' occurring in the output of $\mathcal{E}_m$, it is ensured that $I_m = I_m'$.

Now note that by Lemma 6.17, for the purpose of the argument, $\mathcal{E}_m$ could have run $\mathcal{P}^*$ instead of $\mathcal{E}_{m+1}$ to obtain $\mathbf{I}$ and $\mathbf{I}'$. Therefore, by definition of the index vectors output by $\mathcal{P}^*$, which is such that I_j is a (fixed-size) prefix of I_m for $j < m$, it follows that also $I_j = I_j'$ for all $j < m$.

Therefore, the output $y_1,\dots,y_{k_m}$ of $\mathcal{E}_m$ forms a $(1,\dots,1,k_m,\dots,k_\mu)$-tree of accepting transcripts with trunk $(a_1,\mathsf{RO}(I_1),\dots,a_{m-1},\mathsf{RO}(I_{m-1}),a_m)$, where $I_m = (a_1,\dots,a_m)$. This completes the proof. □

Proposition 6.2 (Run Time and Success Probability)**.** *Let $K_m = \prod_{j=m}^{\mu} k_j$. The extractor $\mathcal{E}_m$ makes an expected number of at most $K_m + Q \cdot (K_m - 1)$ queries to $\mathcal{A}$ (and thus to $\mathcal{P}^*$) and successfully outputs $v = 1$ with probability at least*

$$\frac{\epsilon(\mathcal{A}) - (Q+1) \cdot \kappa_m}{1 - \kappa_m}$$

where

$$\kappa_m := \mathrm{Er}(k_m, \ldots, k_\mu; N, \ldots, N) = 1 - \prod_{i=m+1}^{\mu} \left(1 - \frac{k_i - 1}{N}\right).$$

Proof. The proof goes by induction on m. The base case $m = \mu + 1$ holds trivially, understanding that $K_{\mu+1} = 1$ and $\mathrm{Er}(\emptyset, N) = 0$. Indeed, $\mathcal{E}_{\mu+1}$ makes one call to $\mathcal{A}$ and outputs $v = 1$ with probability $\epsilon(\mathcal{A})$. Alternatively, we can take $m = \mu$ as base case, which follows immediately from Lemma 6.14.

For the induction step, we assume now that the lemma is true for $m' = m + 1$ and consider the extractor $\mathcal{E}_m$. As in the 3-round case, we observe that, within a run of $\mathcal{E}_m$, all the queries that are made by the different invocations of $\mathcal{E}_{m+1}$ are answered *consistently* using lazy sampling, except for the queries to the index I_m, which are answered with different responses c'. This is indistinguishable from having them answered by a full-fledged random oracle $\mathsf{RO}\colon \{1, \ldots, U\} \to \{1, \ldots, N\}$, where we have enumerated the domain and codomain of RO as before. This enumeration allows RO to be identified with its function table $(j_1, \ldots, j_U) \in \{1, \ldots, N\}^U$. Thus, the extractor is actually running the abstract sampling game from Figure 6.4.

However, in contrast to the instantiation of Section 6.6.4, the entries of the array M are now *probabilistic*. Namely, while $\mathcal{A}$ is deterministic, the extractor $\mathcal{E}_{m+1}$ is a probabilistic algorithm. Fortunately, this does not influence the key properties of the abstract sampling game. Namely, for the purpose of the analysis, we may fix the randomness of the extractor $\mathcal{E}_{m+1}$. By linearity of the success probability and the expected run time, the bounds that hold for any fixed choice of randomness also hold when averaged over the randomness. Thus, we can apply Lemma 6.12 and Lemma 6.15 to bound the success probability and the expected run time.[11]

To control the parameters P and T, which occur in the bounds of these lemmas, we make the following observation. A similar observation was required in the proof of Lemma 6.14.

First, by Lemma 6.17, the index vector $\mathbf{I}$ output by $\mathcal{E}_{m+1}$ matches the index vector output by $\mathcal{P}^*$, when given the same random oracle RO. Second, since $\mathcal{P}^*$ is deterministic, its output can only change when the random oracle is reprogrammed at one of the indices $i \in \{1, \ldots, U\}$ queried by $\mathcal{P}^*$. Therefore, for every $(j_1, \ldots, j_U)$, let $S(j_1, \ldots, j_U) \subseteq \{1, \ldots, U\}$ be the set of points that $\mathcal{P}^*$ queries to the random oracle when $(j_1, \ldots, j_U)$ corresponds to the entire function table of the random oracle. Then, $\mathcal{P}^*$ will produce the same output when

[11] To be more precise, to allow for fresh randomness in the different runs of $\mathcal{E}_{m+1}$ within $\mathcal{E}_m$, we first replace the randomness of $\mathcal{E}_{m+1}$ by $F(j_1, \ldots, j_U)$ for a random function F, where $(j_1, \ldots, j_U)$ is the function table of the random oracle providing the answers to $\mathcal{E}_{m+1}$'s queries, and then we fix the choice of F and average over F after having applied Lemma 6.12 and Lemma 6.15.

the random oracle is reprogrammed at an index $i \notin S(j_1, \ldots, j_U)$. In particular, $\mathbf{I}(j_1, \ldots, j_{i-1}, j, j_{i+1}, \ldots, j_U) = \mathbf{I}(j_1, \ldots, j_{i-1}, j', j_{i+1}, \ldots, j_U)$ for all j, j' and for all $i \notin S(j_1, \ldots, j_U)$. Furthermore, $|S(j_1, \ldots, j_U)| \leq Q$. Hence, the conditions of Lemma 6.13 and Lemma 6.16 are satisfied, and it follows that $P \leq Q + 1$ and $T \leq Q$. We are now ready to analyze the success probability and the expected number of $\mathcal{A}$ queries of $\mathcal{E}_m$.

Success Probability. By the induction hypothesis, the success probability p_{m+1} of $\mathcal{E}_{m+1}$ is bounded by

$$p_{m+1} \geq \frac{\epsilon(\mathcal{A}) - (Q+1) \cdot \kappa_{m+1}}{1 - \kappa_{m+1}} .$$

Then, by Lemma 6.12 and Lemma 6.13, the success probability of $\mathcal{E}_m$ is bounded by

$$\frac{N}{N - k_m + 1} \left(p_{m+1} - (Q+1) \frac{k_m - 1}{N} \right)$$

$$\geq \frac{N}{N - k_m + 1} \left(\frac{\epsilon(\mathcal{A}) - (Q+1) \cdot \kappa_{m+1}}{1 - \kappa_{m+1}} - (Q+1) \frac{k_m - 1}{N} \right) .$$

Now observe that, for $\kappa_m = \mathrm{Er}\big(k_m, \ldots, k_\mu; N, \ldots, N\big)$, the following recursive property is easily derived:

$$\frac{N - k_m + 1}{N} (1 - \kappa_{m+1}) = 1 - \kappa_m .$$

Hence,

$$\begin{aligned}
p_m &\geq \frac{\epsilon(\mathcal{A}) - (Q+1) \cdot \kappa_{m+1}}{1 - \kappa_m} - (Q+1) \frac{k_m - 1}{N - k_m + 1} \\
&= \frac{1}{1 - \kappa_m} \left(\epsilon(\mathcal{A}) - (Q+1) \cdot \left(\kappa_{m+1} + (1 - \kappa_m) \frac{k_m - 1}{N - k_m + 1} \right) \right) \\
&= \frac{1}{1 - \kappa_m} \left(\epsilon(\mathcal{A}) - (Q+1) \cdot \left(1 - (1 - \kappa_m) \cdot \frac{N}{N - k_m + 1} \right. \right. \\
&\qquad \left. \left. + (1 - \kappa_m) \frac{k_m - 1}{N - k_m + 1} \right) \right) \\
&= \frac{\epsilon(\mathcal{A}) - (Q+1) \cdot \kappa_m}{1 - \kappa_m} ,
\end{aligned}$$

which proves the claimed success probability.

Expected Number of $\mathcal{A}$-Queries. Let the random variable T_m denote the number of $\mathcal{A}$-queries made by extractor $\mathcal{E}_m$. By the induction hypothesis, it holds that

$$\mathbb{E}[T_{m+1}] \leq K_{m+1} + Q \cdot (K_{m+1} - 1) .$$

We make one crucial observation, allowing us to achieve the claimed query complexity, linear in Q. Namely, we can view the run of a (sub)extractor as a *two-stage* algorithm that allows an *early abort*. By Lemma 6.17, after only one $\mathcal{A}$-query, $\mathcal{E}_{m+1}$ already returns the index I_m. At this stage, $\mathcal{E}_m$ can decide whether to continue the execution of $\mathcal{E}_{m+1}$ or to *early abort* this execution. If the index is incorrect, i.e., it does not match the one obtained in the first invocation of $\mathcal{E}_{m+1}$, then $\mathcal{E}_m$ early aborts the execution of $\mathcal{E}_{m+1}$. Only if the index is correct, the $\mathcal{E}_{m+1}$ execution has to be finished.

For this reason, we define the function $(j_1, \ldots, j_U) \mapsto \Gamma(j_1, \ldots, j_U)$, where $\Gamma(j_1, \ldots, j_U)$ is the (expected) costs of running $\mathcal{E}_{m+1}$ (completely) with random oracle $(j_1, \ldots, j_U)$. Moreover, we set $\gamma = 1$ indicating the cost of an early abort invocation of $\mathcal{E}_{m+1}$. These cost functions measure the expected number of calls to $\mathcal{A}$.

Hence, by Lemma 6.15 and Lemma 6.16, the expected cost of running $\mathcal{E}_m$ is at most

$$\begin{aligned}\mathbb{E}[T_m] &\leq k_m \cdot \mathbb{E}[\Gamma(C)] + \gamma \cdot Q \cdot (k_m - 1) = k_m \cdot \mathbb{E}[T_{m+1}] + Q \cdot (k_m - 1) \\ &\leq K_m + Q \cdot (K_m - k_m) + Q \cdot (k_m - 1) = K_m + Q \cdot (K_m - 1)\,,\end{aligned}$$

where C is distributed uniformly at random in $\mathcal{C}^U$. This completes the proof. □

The existence of extractor $\mathcal{E}_1$, combined with the $\mathbf{k}$-special-soundness property, implies Theorem 6.11. This theorem shows that the Fiat-Shamir security loss for $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound $(2\mu+1)$-round interactive proofs is $Q+1$, i.e., the security loss is linear in the query complexity Q of provers $\mathcal{P}^*$ attacking the considered non-interactive random oracle proof $\mathsf{FS}[\Pi]$. In particular, the Fiat-Shamir security loss is independent of the number of rounds $(2\mu + 1)$ of the interactive proof Π.

Theorem 6.11 (Fiat-Shamir Transformation of a Multi-Round Interactive Proof with a Single Challenge Set)**.** *Let* $\mathbf{k} = (k_1, \ldots, k_\mu)$, $\mathbf{N} = (N, \ldots, N) \in \mathbb{N}^\mu$. *The Fiat-Shamir transformation* $\mathsf{FS}[\Pi]$ *of a* $\mathbf{k}$*-out-of-*$\mathbf{N}$ *special-sound interactive proof* Π*, in which all challenges are sampled from a set* $\mathcal{C}$ *of size* N*, is knowledge sound with knowledge error*

$$(Q + 1) \cdot \mathrm{Er}(\mathbf{k}; \mathbf{N})\,,$$

where

$$\mathrm{Er}(\mathbf{k}; \mathbf{N}) = 1 - \prod_{i=1}^{\mu} \left(1 - \frac{k_i - 1}{N}\right)$$

is the knowledge error of the interactive proof Π*.*

Multi-Round Interactive Proofs with Arbitrary Challenge Sets

Thus far, we considered and analyzed multi-round interactive proofs in which all challenges are sampled uniformly at random from the *same* set $\mathcal{C}$ of cardinality N. However, it is straightforward to verify that our techniques also apply to multi-round interactive proofs with different challenge sets, i.e., where the i-th challenge is sampled from a set $\mathcal{C}_i$ of cardinality N_i.

A natural first step in this generalization is to consider μ random oracles $\mathsf{RO}_i\colon \{0,1\}^{\leq u} \to \mathcal{C}_i$ instead of one. Besides some additional bookkeeping, all the reasoning goes through unchanged. Indeed, everything works as is when the prover $\mathcal{P}^*$ has the additional freedom to choose which random oracle it queries. Thus, we obtain the following generalization of Theorem 6.11.

Theorem 6.12 (Fiat-Shamir Transformation of a Multi-Round Interactive Proof)**.** *Let* $\mathbf{k} = (k_1, \ldots, k_\mu) \in \mathbb{N}^\mu$ *and* $\mathbf{N} = (N_1, \ldots, N_\mu) \in \mathbb{N}^\mu$*. The Fiat-Shamir transformation of a* $\mathbf{k}$*-out-of-*$\mathbf{N}$ *special-sound interactive proof* Π *is knowledge sound with knowledge error* $(Q+1) \cdot \mathrm{Er}(\mathbf{k}; \mathbf{N})$*, where*

$$\mathrm{Er}(\mathbf{k}; \mathbf{N}) := 1 - \prod_{i=1}^{\mu} \left(1 - \frac{k_i - 1}{N_i}\right)$$

is the knowledge error of the interactive proof Π*.*

6.6.7 An Attack on the Fiat-Shamir Transformation of a Parallel Repetition

In the previous sections we have established a positive result: for a broad class of interactive proofs the Fiat-Shamir security loss is only linear in the number of queries Q admitted to a prover $\mathcal{P}^*$ attacking the considered non-interactive random oracle proof. One might therefore wonder whether the generic security loss for $(2\mu+1)$-round interactive proofs, roughly equal to Q^μ, is only tight for contrived examples. In this section, we show that this is not the case. We demonstrate a nontrivial attack on the Fiat–Shamir transformation of the *parallel repetition* of $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs.

Recall that typical $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs Π admit a cheating strategy that succeeds if at least one of the μ random challenges c_i, received from the verifier, hits a certain set Γ_i of size $k_i - 1$ chosen by the dishonest prover. The success probability of this cheating strategy matches the knowledge error

$$\mathrm{Er}(\mathbf{k}; \mathbf{N}) = 1 - \prod_{i=1}^{\mu} \left(1 - \frac{k_i - 1}{N_i}\right).$$

A straightforward analysis shows that this approach generalizes to a cheating strategy for the t-fold parallel repetition $\Pi^t = (\mathcal{P}^t, \mathcal{V}^t)$ of Π, with success probability $\mathrm{Er}(\mathbf{k}; \mathbf{N})^t$ again matching the knowledge error (now of Π^t).

The following (informal) theorem shows the existence of an attack strategy for the Fiat-Shamir transformation of Π^t that succeeds with probability roughly $Q^\mu/\mu^{t+\mu} \cdot \mathrm{Er}(\mathbf{k}; \mathbf{N})^t$. In particular, the security loss of the Fiat-Shamir transformation, when applied to the t-fold parallel repetition Π^t, is roughly $Q^\mu/\mu^{t+\mu}$. This stands in stark contrast to a single execution of a $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound protocol, where the loss is linear in Q and independent of μ. The main idea of this attack is that a dishonest prover $\mathcal{P}^*$ can attack different groups of parallel instances in different rounds of the protocol *independently*. More precisely, in every round the dishonest prover $\mathcal{P}^*$ attacks t/μ parallel instances.

In order to focus on the crucial aspects of the attack, the theorem is stated informally, allowing us to avoid certain cumbersome details. First, we do not

formalize the properties required by the basic interactive proof Π and merely state that this attacks applies to "typical" $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs. Informally, our attack applies to interactive proofs where:

1. the aforementioned cheating strategy, with success probability $\mathrm{Er}(\mathbf{k}, \mathbf{N})$, applies;
2. in the Fiat-Shamir mode the prover $\mathcal{P}^*$ can try sufficiently many message-challenge pairs in every round of the protocol.

The second property ensures that if, at some point during the attack, the random oracle returns a challenges c that does not hit the subset specified by the dishonest prover $\mathcal{P}^*$, i.e., this phase of the attack fails, then $\mathcal{P}^*$ can simply try again by querying the random oracle with a different input value. Typical $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs admit both properties. However, there exist (artificial) counterexamples. Moreover, we only give an approximation of the success probability, and the accuracy of this approximation is not discussed. For a more formal treatment of this attack we refer the reader to the article [AFK22], co-authored by Serge Fehr and Michael Klooß, on which this section is based.

Theorem 6.13 (Informal - Attack on the Fiat-Shamir Transformation of a Parallel Repetition)**.** *The Fiat-Shamir transformation of* $\mathsf{FS}[\Pi^t]$ *of the* t*-fold parallel repetition* Π^t *of a "typical"* $\mathbf{k}$*-out-of-*$\mathbf{N}$ *special-sound interactive proof* Π *admits a* Q*-query cheating strategy that succeeds with probability "roughly"*

$$\frac{Q^\mu}{\mu^{t+\mu}} \cdot \mathrm{Er}(\mathbf{k}; \mathbf{N})^t \,,$$

where $\mathrm{Er}(\mathbf{k}; \mathbf{N})$ *is the knowledge error of* Π *and, thus,* $\mathrm{Er}(\mathbf{k}; \mathbf{N})^t$ *is the knowledge error of* Π^t*.*

Proof. For simplicity, let us assume $\mathbf{k} = (k, \ldots, k)$ and $\mathbf{N} = (N, \ldots, N)$ for some $k, N \in \mathbb{N}$, and assume t and Q to be multiples of μ, i.e., $t = \mu \cdot t'$ and $Q = \mu \cdot Q'$ for some $t', \mu' \in \mathbb{N}$. For a more general treatment we refer to [AFK22].

The main idea of the cheating strategy is that a cheating prover $\mathcal{P}^*$ attacks t' parallel instances of Π in every round of the protocol. The attacks in the different rounds can be executed independently.

More precisely, the cheating strategy proceeds as follows. In the first round, the cheating prover $\mathcal{P}^*$ chooses random first messages $a_1^1, \ldots, a_1^{t'}$ together with subsets $\Gamma_1, \ldots, \Gamma_{t'} \subseteq \mathcal{C}_1$ of cardinality $k-1$, such that the following holds. If the first challenge c_1^j for instance $1 \leq j \leq t'$ lands in Γ_j, then $\mathcal{P}^*$ is able to honestly complete the execution of instance j and have the verifier accept that instance. Recall that typical $\mathbf{k}$-out-of-$\mathbf{N}$ special-sound interactive proofs admit a cheating strategy following precisely this approach. The first messages $a_1^{t'+1}, \ldots, a_1^\mu$ for the remaining $t - t'$ parallel instances are chosen at random. Then, the prover $\mathcal{P}^*$ queries the random oracle to receive the first round challenges $c_1^1, \ldots, c_1^t \in \mathcal{C}_1$ for all parallel instances. This step of the attack succeeds if $c_1^j \in \Gamma_j$ for all $1 \leq j \leq t'$, i.e., if the first t' challenges land in the previously specified subsets Γ_j, which happens with probability $(k-1)^{t'}/N^{t'}$. If this step of the attack has not succeeded, $\mathcal{P}^*$ rewinds to the start of the first round, chooses new first messages and proceeds as

before. The cheating prover $\mathcal{P}^*$ tries to attack this round at most Q' times and therefore succeeds in doing so with probability

$$1 - \left(1 - \left(\frac{k-1}{N}\right)^{t'}\right)^{Q'} \approx Q' \cdot \left(\frac{k-1}{N}\right)^{t'} ,$$

where the approximation holds if $Q' \ll N^{t'}/(k-1)^{t'}$.

If the attack of the first round has succeeded, $\mathcal{P}^*$ moves to the second round and tries to attack parallel instances $t'+1, \ldots, 2t'$ in a similar manner, again succeeding with probability roughly $Q' \cdot (k-1)^{t'}/N^{t'}$. While doing so $\mathcal{P}^*$ generates the messages for parallel instances $1, \ldots, t'$ honestly and samples the messages for instances $2t'+1, \ldots, t$ randomly. The cheating prover $\mathcal{P}^*$ continues until it has either aborted or successfully attacked all t parallel instances.

In every round, $\mathcal{P}^*$ makes at most Q' random oracle queries. Therefore, $\mathcal{P}^*$ is a $Q' \cdot \mu = Q$-query random oracle algorithm. Moreover, this attack strategy succeeds with probability roughly

$$\left(Q' \cdot \left(\frac{k-1}{N}\right)^{t'}\right)^{\mu} = \left(\frac{Q}{\mu}\right)^{\mu} \cdot \left(\frac{k-1}{N}\right)^{t} .$$

The observation that

$$\mathrm{Er}(\mathbf{k}, \mathbf{N}) = 1 - \left(1 - \frac{k-1}{N}\right)^{\mu} \leq \mu \cdot \frac{k-1}{N} ,$$

completes the proof of this informal theorem. □

CHAPTER 7

7

Applications of Compressed Σ-Protocols

7.1 Introduction

The primary functionality of compressed Σ-protocols is to prove knowledge of an opening to one or several compact commitments satisfying a linear constraint. In Chapter 4, in order to handle certain *nonlinear* relations, this functionality was enhanced. First, it was shown how to commit to a long vector of multiplication triples $(\alpha_i, \beta_i, \gamma_i = \alpha_i\beta_i)$ and prove that the committed vector satisfies the corresponding multiplicative relation. Second, a proof of partial knowledge technique was presented, allowing a prover to prove knowledge of k-out-of-n homomorphism preimages. The enhancements, required to handle these two nonlinear scenarios, can be viewed as linearization techniques; in both cases the nonlinear relation is reduced to a linear relation amenable for basic compressed Σ-protocols.

In this chapter, we present two applications of the basic compressed Σ-protocols together with these higher level (nonlinear) functionalities. First, in Section 7.2, we show how to prove *arbitrary* constraints, captured by an arithmetic circuit, on committed vectors. More precisely, we show how to prove that a committed vector $\mathbf{x} \in \mathbb{Z}_q^n$ satisfies the constraint $C(\mathbf{x}) = 0$ for some public arithmetic circuit $C\colon \mathbb{Z}_q^n \to \mathbb{Z}_q^s$. Protocols with this functionality are also referred to as *circuit zero-knowledge protocols*. It turns out that, by deploying the linearization techniques of Section 4.2, we only need *black-box* access to the basic functionality for opening linear forms. This explains also why compressed Σ-protocols do not need any *direct* provision to handle nonlinearity. Further, the number of black-box calls to this basic functionality is constant. Therefore, the (poly)logarithmic communication complexity is directly inherited when proving arbitrary constraints on committed vectors. Section 7.2 is based on the article [AC20], co-authored by Ronald Cramer.

Second, in Section 7.3, we combine the proofs of partial knowledge with an appropriate signature scheme to construct a *threshold signature scheme* (TSS). A k-out-of-n TSS is a standard signature scheme, allowing each of the n players to individually sign arbitrary messages m, enriched with a public k-aggregation algorithm. The k-aggregation algorithm takes as input k signatures, issued by any k distinct players, on the same message m and outputs a *threshold signature*. A TSS

is designed such that no adversary holding strictly less than k distinct signatures on a given message m can issue a valid threshold signature on this message. A naive TSS is obtained by exhibiting the k individual signatures directly. However, this approach results in threshold signatures with size linear in the threshold k. The main goal for TSSs is to have *succinct* threshold signatures, i.e., with size sublinear in k and n. The succinct TSS of [Sho00] immediately found an application in reducing the communication complexity of consensus protocols [CKS00; CKS05], this application was revived recently [LM18; AMS19; YMR+19; ADD+19]. The impact of succinctness is significant since, in consensus applications, the threshold k is of the same order of magnitude as n (typically $k = n/2$ or $k = 2n/3$). Although desirable in some applications, it is not required that a threshold signature *hides* the k-subset of signers. We construct a succinct TSS that has this additional security property, i.e., threshold signatures do not reveal any information about the k-subset of players that supplied valid signatures to the aggregation algorithm. Section 7.3 is based on the article [ACR21], co-authored by Ronald Cramer and Matthieu Rambaud.

7.2 Circuit Zero-Knowledge Protocols

First, in Section 7.2.1, we describe the compressed Σ-protocol for basic circuit satisfiability. This protocol allows a prover to commit to an input $\mathbf{x}$ and subsequently prove that the committed input $\mathbf{x}$ satisfies the constraint $C(\mathbf{x}) = 0$ for an arbitrary, but fixed, arithmetic circuit C. In practice, it may happen that the prover is already committed to the secret $\mathbf{x}$ *before* receiving the circuit C. This is referred to as the "commit-and-prove" scenario. In order to deal with this scenario, we need some further utility enhancements. The required enhancements are described in Section 7.2.2. Finally, in Section 7.2.3, we describe a generalization from arithmetic circuits to bilinear group arithmetic circuits.

7.2.1 The Compressed Σ-Protocol for Arithmetic Circuits

Suppose $C\colon \mathbb{Z}_q^n \to \mathbb{Z}_q^s$ is an arithmetic circuit with n inputs, s outputs and m multiplication gates. We only count multiplication gates with *variable* inputs; additions and multiplications by constants are implicitly handled and immaterial to the communication costs. We can easily turn our approach for proving correctness of multiplication triples into a solution for "circuit zero-knowledge," i.e., the prover convinces the verifier it knows an input $\mathbf{x} \in \mathbb{Z}_q^n$ for which the circuit C, without loss of generality, returns 0. We note that [CDP12] also gives a solution for circuit zero-knowledge based on linearizing multiplication triples. But that solution has a communication complexity that is linear in the size of the circuit C. We aim for a (poly)logarithmic communication complexity, so we make some changes.

The protocol goes as follows. The prover first determines the computation graph implied by instantiating the circuit C with its input vector $\mathbf{x}$. In this graph every wire is assigned a value in $\mathbb{Z}_q$. In particular, let $\alpha_1, \ldots, \alpha_m \in \mathbb{Z}_q$ be the left inputs, $\beta_1, \ldots, \beta_m \in \mathbb{Z}_q$ the right inputs and $\gamma_1, \ldots, \gamma_m \in \mathbb{Z}_q$ the outputs of the m multiplication gates in this computation graph. Hence, each $(\alpha_i, \beta_i, \gamma_i) \in \mathbb{Z}_q^3$ is a multiplication triple.

Let us now use the following simple fact about arithmetic circuits. For each i, there are affine forms[1] $u_i, v_i \colon \mathbb{Z}_q^{n+m} \to \mathbb{Z}_q$, depending only on C, such that, for all $\mathbf{x} \in \mathbb{Z}_q^n$, it holds that $\alpha_i = u_i(\mathbf{x}, \gamma_1, \ldots, \gamma_m)$ and $\beta_i = v_i(\mathbf{x}, \gamma_1, \ldots, \gamma_m)$. These forms are uniquely determined by the addition and scalar multiplication gates. In other words, a given vector $(\mathbf{x}, \gamma_1, \ldots, \gamma_m) \in \mathbb{Z}_q^{n+m}$ can be completed to a valid computation graph if and only if

$$u_i(\mathbf{x}, \gamma_1, \ldots, \gamma_m) \cdot v_i(\mathbf{x}, \gamma_1, \ldots, \gamma_m) = \gamma_i \,,$$

for all $1 \leq i \leq m$. Hence, checking whether $(\mathbf{x}, \gamma_1, \ldots, \gamma_m)$ corresponds to a valid computation graph amounts to verifying the multiplication triples defined by the γ_i's and the public linear forms u_i and v_i. This verification can be performed by deploying the arithmetic secret-sharing based linearization technique for multiplication triples of Section 4.2. Further, there are affine forms $w_j \colon \mathbb{Z}_q^{n+m} \to \mathbb{Z}_q$ corresponding to the s output gates of C. Hence, the evaluation $C(\mathbf{x})$ returns 0 if and only if $w_j(\mathbf{x}, \gamma_1, \ldots, \gamma_m) = 0$ for all $1 \leq j \leq s$. Recall that s is the dimension of the codomain of the circuit $C \colon \mathbb{Z}_q^n \to \mathbb{Z}_q^s$.

Altogether, after computing the above computation graph, the circuit satisfiability protocol therefore proceeds as follows. As in Section 4.2, the prover selects a random polynomial $f(X)$ of degree at most m that defines a packed secret sharing of the vector $(\alpha_1, \ldots, \alpha_m)$ of left inputs to the multiplication gates. The prover also selects a random polynomial $g(X)$ of degree at most m that defines a packed secret sharing of the vector $(\beta_1, \ldots, \beta_m)$ of right inputs to the multiplication gates. Finally, the prover computes the product polynomial $h(X) := f(X)g(X)$ of degree at most $2m < q$.

The prover commits to each coordinate of $\mathbf{x}$ and to the *auxiliary data*

$$\mathsf{aux} = (f(0), g(0), h(0), h(1), \ldots, h(2m)) \in \mathbb{Z}_q^{2m+3}$$

in one single compact commitment. The length of the committed vector $\mathbf{y} = (\mathbf{x}, \mathsf{aux})$ thus equals $n + 2m + 3$. Note that the vector $\mathbf{y}$ contains the outputs $\gamma_1 = h(1), \ldots, \gamma_m = h(m)$ of the multiplication gates of C evaluated on $\mathbf{x}$. However, it does not necessarily contain the inputs $\alpha_1, \beta_1, \ldots, \alpha_m, \beta_m$ of these multiplication gates. These inputs are namely affine combinations of the coefficients of $\mathbf{y}$. This explains why it is not necessary to commit explicitly to the α_i's and the β_i's as these are now implicitly committed to via said affine forms evaluated on $\mathbf{y}$. Therefore, since the values $f(0)$ and $g(0)$ are still included in $\mathbf{y}$, the polynomials $f(X)$, $g(X)$ and $h(X)$ are well defined by $\mathbf{y}$, and their evaluations are, by composition of the appropriate maps, also affine evaluations on $\mathbf{y}$. What remains is to check that the polynomial $h(X)$ is indeed the product of $f(X)$ and $g(X)$.

Therefore, with the above observations at hand, the circuit zero-knowledge protocol is reduced to opening the affine forms that, on input $\mathbf{y}$, output $\big(C(\mathbf{x}), f(c), g(c), h(c)\big) \in \mathbb{Z}_q^{s+3}$ for a challenge $c \leftarrow_R \mathbb{Z}_q \setminus \{1, \ldots, m\}$ sampled uniformly at random by the verifier. First, the verifier checks that $h(c) = f(c)g(c)$,

[1] Recall that an affine form $A \colon \mathbb{Z}_q^n \to \mathbb{Z}_q$ is a linear form L plus a constant $a \in \mathbb{Z}_q$. Hence, opening an affine form $A = L + a$ amounts to opening the linear form L and adding the (public) constant a.

which, as in Section 4.2, shows that $h(X) = f(X)g(X)$ holds with high probability. Second, the verifier checks that $C(\mathbf{x}) = 0$, which shows that the circuit is satisfiable and that the prover knows a witness $\mathbf{x}$.

This approach thus reduces the nonlinear circuit satisfiability relation to opening a constant number of affine forms on a compactly committed vector, it is therefore again a *linearization* technique. The compressed Σ-protocol for circuit satisfiability thus consists of two main building blocks: (1) the linearization technique and (2) a compressed Σ-protocol for opening linear forms.

The linearization technique itself can be presented as an interactive proof for the circuit satisfiability relation

$$\mathfrak{R}_{\mathsf{CS}} = \{(C; \mathbf{x}) : C(\mathbf{x}) = 0\}\,.$$

This interactive proof is composable with a basic compressed Σ-protocol for opening linear forms, allowing its linear communication complexity to be reduced. A formal description can be found in Protocol 16. To simplify the exposition we consider an abstract compact vector commitment scheme

$$[\cdot]\colon \bigcup_{\ell\in\mathbb{N}} \mathbb{Z}_q^\ell \to \mathbb{H}$$

that allows a prover to commit to arbitrary length vectors $\mathbf{x} \in \bigcup_{\ell\in\mathbb{N}} \mathbb{Z}_q^\ell$ in a single group element $P \in \mathbb{H}$. In this notation, we leave the commitment randomness implicit, i.e., $[\mathbf{x}]$ denotes a commitment to the vector $\mathbf{x}$.

As a stand-alone building block, the interactive proof described in Protocol 16 might seem pointless, as the prover's final message $(\mathbf{y}, u, v, w)$ contains the witness $\mathbf{x}$. Hence, it is clearly not zero-knowledge and it is less efficient than a trivial interactive proof that simply reveals the witness $\mathbf{x}$. However, the key point is that the long vector $\mathbf{y}$ can be viewed as an interactive proof for opening linear forms. This long vector, dominating the communication costs, can therefore be replaced by a basic compressed Σ-protocol. Thus, Protocol 16 indeed linearizes the nonlinear circuit satisfiability relation, making it amenable for basic compressed Σ-protocols. Below we describe the properties of this composition, but Theorem 7.1 first summarizes the main properties of the stand-alone linearization technique for circuit satisfiability. It shows that this technique is a perfectly complete and $(2m+1)$-out-of-$(q-m)$ special-sound Σ-protocol.

Theorem 7.1 (Linearization for Circuit Satisfiability)**.** *Let $n, m, s \in \mathbb{N}$, $q > 3m$ a prime, $[\cdot]\colon \bigcup_{\ell\in\mathbb{N}} \mathbb{Z}_q^\ell \to \mathbb{H}$ a homomorphic vector commitment scheme and $C\colon \mathbb{Z}_q^n \to \mathbb{Z}_q^s$ an arithmetic circuit with m multiplication gates. The Σ-protocol for relation*

$$\mathfrak{R}_{\mathsf{CS}} = \{(C; \mathbf{x}) : C(\mathbf{x}) = 0\}\,,$$

described in Protocol 16, is perfectly complete, and $(2m+1)$-out-of-$(q-m)$ special-sound, under the assumption that the commitment scheme is binding.

Proof. **Completeness:** This property follows immediately.

Protocol 16 Linearization of the Circuit Satisfiability Relation.

Parameters:	$n, m, s \in \mathbb{N}$, prime $q > 3m$, group $(\mathbb{H}, \cdot)$ with exponent q and homomorphic vector commitment scheme $[\cdot]\colon \bigcup_{\ell \in \mathbb{N}} \mathbb{Z}_q^\ell \to \mathbb{H}$
Public Input:	circuit $C\colon \mathbb{Z}_q^n \to \mathbb{Z}_q^s$ with m multiplication gates
Prover's Private Input:	$\mathbf{x} \in \mathbb{Z}_q^n$ and, for $1 \leq i \leq m$, (α_i, β_i) denote the left and right inputs to the multiplication gates of the circuit C evaluated in $\mathbf{x}$
Prover's Claim:	$C(\mathbf{x}) = 0$

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
$f(X), g(X) \leftarrow_R \mathbb{Z}_q[X]$ s.t. $\deg(f) \leq m \ \wedge \ \deg(g) \leq m$ $\wedge\ f(i) = \alpha_i \ \wedge \ g(i) = \beta_i \ \forall i$ $\wedge\ h(X) = f(X)g(X)$		
$\mathbf{y} = \big(\mathbf{x}, f(0), g(0), h(0), \ldots, h(2m)\big)$ $P = [\mathbf{y}]$		
	$\xrightarrow{P}$	
		$c \leftarrow_R \mathbb{Z}_q \setminus \{1, \ldots, m\}$
	$\xleftarrow{c}$	
$u = f(c)$ $v = g(c)$ $w = h(c)$		
	$\xrightarrow{\mathbf{y}, u, v, w}$	$u \cdot v \stackrel{?}{=} w$
		$[\mathbf{y}] \stackrel{?}{=} P$ $A_C(\mathbf{y}) \stackrel{?}{=} 0\,,\ f_{\mathbf{y}}(c) \stackrel{?}{=} u$ $g_{\mathbf{y}}(c) \stackrel{?}{=} v\,,\ h_{\mathbf{y}}(c) \stackrel{?}{=} w$

Here, $A_C\colon \mathbb{Z}_q^{n+2m+3} \to \mathbb{Z}_q^s$ is the affine mapping that, on input the vector $\mathbf{y}$ containing the secret input $\mathbf{x} \in \mathbb{Z}_q^n$ and the outputs of the multiplication gates of C evaluated in $\mathbf{x}$, outputs $C(\mathbf{x}) \in \mathbb{Z}_q^s$. Further, $f_{\mathbf{y}}(X)$, $g_{\mathbf{y}}(X)$ and $h_{\mathbf{y}}(X)$ are the polynomials defined by the vector $\mathbf{y}$ and the circuit C. They correspond to the polynomials $f(X)$, $g(X)$ and $h(X)$ constructed by an *honest* prover.

Special-Soundness: Let

$$(P, c_0, \mathbf{y}_0, u_0, v_0, w_0)\,, \ \ldots\ , (P, c_{2m}, \mathbf{y}_{2m}, u_{2m}, v_{2m}, w_{2m})$$

be $2m + 1$ accepting transcripts with common first message P and pairwise distinct challenges $c_j \in \mathbb{Z}_q \setminus \{1, \ldots, m\}$. Then, under the assumption that the commitment scheme $[\cdot]$ is binding, it follows that $\mathbf{y}_0 = \cdots = \mathbf{y}_{2m} = \mathbf{y}$,

and we may write $f(X) = f_{\mathbf{y}}(X)$, $g(X) = g_{\mathbf{y}}(X)$ and $h(X) = h_{\mathbf{y}}(X)$ for the three polynomials unique defined by $\mathbf{y}$.

Further, $\mathbf{y}$ corresponds to a wire value assignment of the circuit C. In particular, for $1 \leq i \leq m$, $\alpha_i = f(i)$, $\beta_i = g(i)$ and $\gamma_i = h(i)$ correspond to the values assigned to the left input, the right input and the output of the i-th multiplication gate in C. Moreover, since $A_C(\mathbf{y}) = 0$, the values assigned to the output wires are equal to 0. What remains is to verify that this wire value assignment is *valid*, i.e., for all gates the output should correspond to the appropriate combination of the input values.

The linear relations, defined by the addition and scalar multiplication gates, are automatically satisfied. Therefore, all that needs to be verified are the multiplicative relations $\alpha_i \cdot \beta_i = \gamma_i$. To this end, observe that, for all $0 \leq j \leq 2m$,

$$f(c_j) \cdot g(c_j) = u_j \cdot v_j = w_j = h(c_j)\,.$$

Since the polynomials $f(X)$ and $g(X)$ are of degree at most m and $h(X)$ is of degree at most $2m$, it follows that $f(X) \cdot g(X) = h(X)$. Hence,

$$\alpha_i \cdot \beta_i = f(i) \cdot g(i) = h(i) = \gamma_i$$

for all $1 \leq i \leq m$, which completes the proof.

□

The vector $\mathbf{y}$, sent in the final round of Protocol 16, is a trivial proof of knowledge for opening the $s + 3$ affine forms that return

$$\big(C(\mathbf{x}) = A_C(\mathbf{y}), f_{\mathbf{y}}(c), g_{\mathbf{y}}(c), h_{\mathbf{y}}(c)\big) \in \mathbb{Z}_q^{s+3}$$

on input $\mathbf{y}$. The compressed Σ-protocol for basic circuit satisfiability simply replaces this trivial interactive proof, i.e., the message $\mathbf{y}$, by a compressed Σ-protocol for opening the appropriate linear forms. Recall that the costs of these $s+3$ linear form openings can be amortized (Section 3.4.2), i.e., the amortized communication costs are independent of s.

The exact communication costs depend on the instantiation of the compact commitment scheme. For instance, the discrete logarithm based instantiation has logarithmic communication, while, due to soundness slack, the lattice based instantiation has polylogarithmic communication. For concreteness let us consider the discrete logarithm instantiation of Section 5.2. The following theorem summarizes the main properties of the discrete logarithm based circuit satisfiability protocol Π_{CS}. Note in particular that this compressed Σ-protocol has a communication complexity that is logarithmic in the dimension n of the input vector $\mathbf{x} \in \mathbb{Z}_q^n$ and the number of multiplication gates m. Moreover, since the Pedersen vector commitment scheme is unconditionally hiding, it is special honest-verifier zero-knowledge.

Theorem 7.2 (DL-Based Compressed Σ-Protocol for Circuit Satisfiability)**.** *Let q be a prime and $\mu, n, m, s \in \mathbb{N}$ such that $n + 2m + 4 = 2^\mu$ and $q > 3m$. Further, let* $\text{COM}\colon \mathbb{Z}_q^{n+2m+3} \times \mathbb{Z}_q \to \mathbb{H}$ *be the Pedersen vector commitment scheme.*

Then the compressed Σ*-protocol* Π_{CS} *for relation*

$$\mathfrak{R}_{\mathsf{CS}} = \{(C; \mathbf{x}) : C(\mathbf{x}) = 0\},$$

instantiated with the Pedersen commitment scheme COM*, is perfectly complete, computationally* $(2m+1, s+4, 3, \ldots, 3)$*-out-of-*$(q-m, q, \ldots, q)$ *special-sound, under the discrete logarithm assumption, and special honest-verifier zero-knowledge (SHVZK). Moreover, it has* $2\mu+5$ *communication rounds and the communication costs are:*

- $\mathcal{P} \to \mathcal{V}$*:* 5 *elements of* $\mathbb{Z}_q$ *and* 2μ *elements of* $\mathbb{H}$*;*
- $\mathcal{V} \to \mathcal{P}$*:* $\mu + 2$ *elements of* $\mathbb{Z}_q$*.*

Proof. Completeness and special-soundness follow directly from Lemma 3.1. This lemma describes the properties of the composition of interactive proofs. However, Lemma 3.1 only states that the composition $\Pi_b \diamond \Pi_a$ of interactive proofs is SHVZK if the interactive proof Π_a that is applied *first* is SHVZK.

In our case, the interactive proof that is applied first, the linearization of Protocol 16, is not SHVZK. So we need to use additional properties to prove that the compressed Σ-protocol for circuit satisfiability is SHVZK. It turns out that this property follows from the fact that the Pedersen vector commitment scheme is perfectly hiding and the deployed instantiation of Shamir's packed secret-sharing scheme has 1-privacy.

To see this, let us describe the SHVZK simulator. Assume that the circuit C admits an input $\mathbf{x}$ such that $C(\mathbf{x}) = 0$ and let $(\alpha_i, \beta_i, \gamma_i = \alpha_i\beta_i)$ denote the left input, right input and output values of the multiplication gates of C evaluated in $\mathbf{x}$. The simulator then proceeds as follows. First, it samples the challenges $c_1 \leftarrow_R \mathbb{Z}_q \setminus \{1, \ldots, m\}$ and $c_2, \ldots, c_{\mu+2} \leftarrow_R \mathbb{Z}_q$ uniformly at random. Second, it samples a Pedersen commitment $P \leftarrow_R \mathbb{H}$ and field elements $u, v \leftarrow_R \mathbb{Z}_q$ uniformly at random, and sets $w = u \cdot v$.

The first three messages P, c_1 and (u, v, w) of the circuit satisfiability protocol have now been simulated. The remaining messages are sampled by using the SHVZK simulator of the compressed Σ-protocol for opening linear forms. However, this is only possible when the commitment P admits an opening $(\mathbf{y}; \gamma)$ satisfying the appropriate linear relations.

To see that this is the case, note that, since $c_1 \notin \{1, \ldots, m\}$, there exist polynomials[2] $f, g \in \mathbb{Z}_q[X]$ of degree at most m such that $f(c_1) = u$ and $g(c_1) = v$, and $f(i) = \alpha_i$ and $g(i) = \beta_i$ for all $1 \leq i \leq m$. Let $h(X) = f(X) \cdot g(X)$.

Hence, there exists a vector $\mathbf{y} = \big(\mathbf{x}, f(0), g(0), h(0), \ldots, h(2m)\big)$ that satisfies the linear relations $A_C(\mathbf{y}) = 0$, $f_{\mathbf{y}}(c_1) = f(c_1) = u$, $g_{\mathbf{y}}(c_1) = g(c_1) = v$ and $h_{\mathbf{y}}(c_1) = h(c_1) = w$. Moreover, since the Pedersen vector commitment scheme is perfectly hiding, the random commitment P has an opening $(\mathbf{y}; \gamma)$ for some $\gamma \in \mathbb{Z}_q$. Hence, the compressed Σ-protocol for opening linear forms is instantiated with a statement P that admits a witness satisfying the appropriate linear constraints. Therefore, the simulator for our circuit satisfiability protocol can run the SHVZK simulator of the compressed Σ-protocol for opening linear forms to

[2] The existence of these polynomials follows from the 1-privacy of the secret-sharing scheme.

simulate the remaining messages. Again using the hiding property of the Pedersen vector commitment scheme and the 1-privacy of the secret-sharing scheme, it is easily seen that the simulated transcripts follow the same distribution as honestly generated ones. □

The terminology *circuit satisfiability* seems to suggest that we are only considering circuits for which it is hard to compute a satisfying witness $\mathbf{x}$, i.e., an $\mathbf{x}$ with $C(\mathbf{x}) = 0$. However, many practical scenarios consider circuits C for which it is easy to compute an $\mathbf{x}$ such that $C(\mathbf{x}) = 0$. In these scenarios the functionality offered by a circuit zero-knowledge protocol is still nontrivial. Namely, after evaluating the protocol, the prover has not only proven knowledge of a witness $\mathbf{x}$, but is also *committed* to this vector $\mathbf{x}$. Hence, in this case, a prover can show that a committed vector satisfies certain properties captured by the arithmetic circuit. These properties do not need to be captured by arithmetic circuits for which it is hard to compute an input evaluating to 0.

7.2.2 An Extension to *Commit-and-Prove* Protocols

In the previous section we treated the basic circuit satisfiability scenario, where a prover claims to know a satisfiable input $\mathbf{x} \in \mathbb{Z}_q^n$ such that $C(\mathbf{x}) = 0$ for some public arithmetic circuit $C\colon \mathbb{Z}_q^n \to \mathbb{Z}_q^s$. The compressed Σ-protocol Π_{CS} for basic circuit satisfiability requires the prover to commit to the input $\mathbf{x}$ and a vector of auxiliary data aux in a single compact commitment. However, in practice it is likely that the prover is *already* committed to the input $\mathbf{x}$ before the start of the protocol. Consider, for example, the following two extreme cases:

Case 1: The prover is committed to $\mathbf{x}$ in a *single* compact commitment.

Case 2: The prover is committed to the coordinates of $\mathbf{x}$ *individually*, i.e., each coordinate is committed to in a separate 1-dimensional commitment.

Besides these extreme cases one can consider hybrid scenarios in which the secret-vector-of-interest $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_s)$ is dispersed over s compact commitments to vectors $\mathbf{x}_i \in \mathbb{Z}_q^{n_i}$. We will focus on the two extreme cases, but hybrid scenarios can be handled similarly.

An interactive proof that allows a prover to prove statements about secret vectors that it is already committed to is called a *commit-and-prove* protocol. More precisely, given a compact vector commitment scheme $[\cdot]\colon \bigcup_{\ell\in\mathbb{N}} \mathbb{Z}_q^\ell \to \mathbb{H}$ (again leaving the commitment randomness implicit), a commit-and-prove protocol, for the case 1 scenario, is an interactive proof for relation

$$\mathfrak{R}^1_{\mathsf{CS}} = \{(P, C; \mathbf{x}) : [\mathbf{x}] = P \wedge C(\mathbf{x}) = 0\},$$

and a commit-and-prove protocol, for the case 2 scenario, is an interactive proof for relation

$$\mathfrak{R}^2_{\mathsf{CS}} = \{(P_1, \ldots, P_n, C; \mathbf{x}) : [x_i] = P_i \ \forall i \wedge C(\mathbf{x}) = 0\}.$$

In order to deal with these scenarios, we first need to bring about the desired starting point for the circuit satisfiability protocol of Section 7.2.1, i.e., the prover

needs to be committed to all coordinates of the input $\mathbf{x}$ and the required auxiliary information aux in a single compact commitment. Similar to Section 4.2.2, we handle the commit-and-prove scenario by deploying the compactification techniques of Section 3.4.4.

Let us first consider the case 1 commit-and-prove scenario. In this case, the basic circuit satisfiability protocol is adapted as follows.

- In the first round, instead of sending a compact commitment to the $n+2m+3$ dimensional vector $\mathbf{y} = (\mathbf{x}, \mathsf{aux})$, the prover sends a compact commitment Q to $(0, \mathsf{aux}) \in \mathbb{Z}_q^{n+2m+3}$ to the verifier.[3]
- Given the commitment $P = [\mathbf{x}] = [(\mathbf{x}, 0)]$ to the input vector $\mathbf{x}$ and the commitment $Q = [(0, \mathsf{aux})]$ to the auxiliary data, both the prover and verifier can compute a single compact commitment $P \cdot Q = [(\mathbf{x}, \mathsf{aux})]$ to all relevant data. This brings about the desired starting point for the circuit satisfiability protocol of Section 7.2.1.
- What remains is for the prover to show that the commitment Q is of the appropriate form. More precisely, Q should be a commitment to a vector with zeros in its first n coordinates. This boils down to opening the linear forms $L_i(\mathbf{y}) = y_i$, for $1 \leq i \leq n$, that return the first n coordinates of the vector $\mathbf{y}$. As before, the communication complexity of opening n different linear forms on the same commitment can be amortized.

The above shows how the case 1 commit-and-prove scenario is reduced to opening $s+3$ linear forms on the commitment $P \cdot Q = [(\mathbf{x}, \mathsf{aux})]$ and opening n linear forms on the commitment $Q = [(0, \mathsf{aux})]$. The naive approach of simply evaluating two (amortized) compressed Σ-protocols increases the communication costs with roughly a factor two, with respect to the basic circuit satisfiability protocol Π_{CS}. However, the factor two loss can be avoided by deploying the (case 1) compactification techniques of Section 3.4.4. Recall that compactification allows a prover to *compactify* relevant data that is dispersed over several commitments, into a single compact commitment. Phrased alternatively, compactification techniques allow a prover to open linear forms evaluated on inputs that are dispersed over several commitments.

The resulting compressed Σ-protocol, denoted by Π^1_{CS}, is an interactive proof for commit-and-prove relation $\mathfrak{R}^1_{\mathsf{CS}}$. Theorem 7.3 summarizes the main properties of the discrete logarithm instantiation of Π^1_{CS}, i.e., the instantiation using the Pedersen vector commitment scheme. The other instantiations of Chapter 5 work similarly, but may result in different communication complexities. Note that the communication costs of $\mathfrak{R}^1_{\mathsf{CS}}$ are roughly the same as those of the compressed Σ-protocol for basic circuit satisfiability.

Theorem 7.3 (DL-Based Case 1 Commit-and-Prove Protocol for Arithmetic Circuits)**.** *Let q be a prime and $\mu, n, m, s \in \mathbb{Z}_q$ such that $n+2m+6 = 2^\mu$ and $q > 3m$. Further, let* $\textsc{com}\colon \left(\bigcup_{\ell \in \mathbb{N}} \mathbb{Z}_q^\ell\right) \times \mathbb{Z}_q \to \mathbb{H}$ *the Pedersen vector commitment scheme and $C\colon \mathbb{Z}_q^n \to \mathbb{Z}_q^s$ an arithmetic circuit with m multiplication gates.*

[3] Here, $(0, \mathsf{aux})$ denotes the vector $\mathsf{aux} \in \mathbb{Z}_q^{2m+3}$ of auxiliary information prepended with n zeros.

Then the discrete logarithm based compressed Σ-protocol Π^1_{CS} *for relation*

$$\mathfrak{R}^1_{\mathsf{CS}} = \{(P, C; \mathbf{x}, \gamma) : \text{COM}(\mathbf{x}; \gamma) = P \wedge C(\mathbf{x}) = 0\}$$

is perfectly complete, computationally $(2m+1, \max(n+1, s+4), 2, 2, 3, \ldots, 3)$*-out-of-*$(q-m, q, q, q-1, q, \ldots, q)$ *special-sound, under the discrete logarithm assumption, and special honest-verifier zero-knowledge (SHVZK). Moreover, it has* $2\mu + 11$ *communication rounds and the communication costs are:*

- $\mathcal{P} \to \mathcal{V}$*:* 11 *elements of* $\mathbb{Z}_q$ *and* $2\mu + 4$ *elements of* $\mathbb{H}$*;*
- $\mathcal{V} \to \mathcal{P}$*:* $\mu + 5$ *elements of* $\mathbb{Z}_q$*.*

The case 2 commit-and-prove scenario is handled similarly. However, instead of the case 1 compactification techniques, we now deploy the case 2 compactification techniques of Section 3.4.4. The resulting protocol, denoted by Π^2_{CS}, is a compressed Σ-protocol for relation $\mathfrak{R}^2_{\mathsf{CS}}$. Theorem 7.4 summarizes the main properties of its discrete logarithm instantiation.

Theorem 7.4 (DL-Based Case 2 Commit-and-Prove Protocol for Arithmetic Circuits)**.** *Let* q *be a prime and* $\mu, n, m, s \in \mathbb{Z}_q$ *such that* $n + 2m + 5 = 2^\mu$ *and* $q > 3m$*. Further, let* COM: $\left(\bigcup_{\ell \in \mathbb{N}} \mathbb{Z}_q^\ell\right) \times \mathbb{Z}_q \to \mathbb{H}$ *be the Pedersen vector commitment scheme and* $C \colon \mathbb{Z}_q^n \to \mathbb{Z}_q^s$ *an arithmetic circuit with* m *multiplication gates.*

Then the discrete logarithm based compressed Σ-protocol Π^1_{CS} *for relation*

$$\mathfrak{R}^1_{\mathsf{CS}} = \{(P_1, \ldots, P_n, C; \mathbf{x}, \gamma_1, \ldots, \gamma_n) : \text{COM}(x_i; \gamma_i) = P_i \ \forall i \wedge C(\mathbf{x}) = 0\}\,,$$

is perfectly complete, computationally $(2m+1, n+1, s+5, 2, 3, \ldots, 3)$*-out-of-*$(q-m, q, \ldots, q)$ *special-sound, under the discrete logarithm assumption, and special honest-verifier zero-knowledge (SHVZK). Moreover, it has* $2\mu + 7$ *communication rounds and the communication costs are:*

- $\mathcal{P} \to \mathcal{V}$*:* 7 *elements of* $\mathbb{Z}_q$ *and* $2\mu + 1$ *elements of* $\mathbb{H}$*;*
- $\mathcal{V} \to \mathcal{P}$*:* $\mu + 3$ *elements of* $\mathbb{Z}_q$*.*

7.2.3 A Generalization to Bilinear Group Arithmetic Circuits

Every computable function with fixed input length can be expressed as an arithmetic circuit. Therefore interactive proofs for arithmetic circuit satisfiability are extremely powerful and widely deployed. In fact, they lead to an obvious, but indirect, approach for arbitrary relations:

1. Construct an arithmetic circuit capturing the relation;
2. Apply an efficient circuit ZK protocol to this arithmetic circuit.

However, for some relations, the associated arithmetic circuits can be large and complex, thereby losing the conceptual simplicity and possibly even the concrete efficiency over a more direct approach.

For instance, Lai et al. [LMR19] consider the *bilinear group arithmetic circuit model*. A bilinear group arithmetic circuit is defined over a bilinear group

$(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}, e)$, where $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{H}$ is a bilinear pairing between groups of prime order q. Its wires take values in either $\mathbb{Z}_q$, $\mathbb{G}_1$, $\mathbb{G}_2$ or $\mathbb{H}$, and gates are either group operations, $\mathbb{Z}_q$-scalar multiplications or bilinear pairings. Hence, bilinear group circuits are generalizations of arithmetic circuits. They directly capture relations encountered in, e.g., identity based encryption [SW05] and structure preserving signatures [AFG+16].

Every bilinear group arithmetic circuit can also be expressed as an arithmetic circuit. This requires every group element to be expressed as a vector of $\mathbb{Z}_q$-elements and every gate to be replaced by a $\mathbb{Z}_q$-circuit. For instance, for a highly optimized group of order $q \approx 2^{256}$, evaluating a single group exponentiation requires an arithmetic circuit with approximately 800 multiplication gates [HBH+20]. In the bilinear circuit model, exactly the same operation would only comprise a single gate. Hence, expressing a bilinear group arithmetic circuit as an arithmetic circuit can significantly increase its size. Therefore, avoiding this reduction might significantly reduce the communication costs.

Lai et al. [LMR19] generalize the Bulletproof framework for arithmetic circuits to handle bilinear group arithmetic circuits directly. Also compressed Σ-protocols admit a straightforward adaption for this more general model. To see this, we merely require two observations. First, the pairing-based commitment scheme of Section 5.3 allows a prover to commit to mixed vectors $\mathbf{x} \in \mathbb{Z}_q^{n_0} \times \mathbb{G}_1^{n_1} \times \mathbb{G}_2^{n_2} \times \mathbb{H}^{n_T}$. This commitment scheme is homomorphic and the size of a commitment is constant in $n_0 + n_1 + n_2$ and linear in n_T. Second, the gates in a bilinear group arithmetic circuit are either affine or bilinear. The affine gates are handled directly by our compressed Σ-protocols. Moreover, as before, the bilinear gates can be linearized via an arithmetic secret-sharing scheme. Altogether we obtain a compressed Σ-protocol for relations captured by bilinear arithmetic circuits. For more details we refer the reader to [ACR21].

7.3 Threshold Signature Scheme

In this section, as a second application of compressed Σ-protocols, we construct a transparent k-out-of-n threshold signature scheme (TSS) with threshold signatures that are $\mathcal{O}(\lambda \log n)$ bits, where λ is the security parameter. Recall that a TSS enables any set of at least k players, in a group of n, to issue a "threshold" signature on a message m, but no subset of less than k players is able to issue one. A TSS is called *transparent* if it does not require a trusted setup phase, i.e., all public parameters are random coins. Given recent advances in efficient circuit zero-knowledge, an obvious TSS construction defines a threshold signature as a proof of knowledge attesting the knowledge of k-out-of-n signatures. With the appropriate circuit zero-knowledge protocol this would immediately result in a transparent TSS with sublinear size threshold signatures. However, this approach requires an inefficient reduction from the corresponding threshold signature relation to a relation defined over an arithmetic circuit. More precisely, the arithmetic circuits capturing these relations are typically large.

For this reason, we follow a more *direct* approach avoiding this inefficient reduction. Namely, we append the BLS signature scheme [BLS01; BLS04] with a

k-aggregation algorithm. The BLS signature scheme is defined over a bilinear group. In particular, the BLS verification algorithm checks a linear constraint defined over a bilinear group. This naturally fits with the compressed Σ-protocols for opening homomorphisms. To derive the required threshold functionality, we use the proof of partial knowledge techniques from Section 4.3. The compressed Σ-protocols are interactive. To obtain a signature scheme, they can be made non-interactive by applying the Fiat-Shamir transformation [FS86].

The non-interactive proofs contain precisely the messages sent from the prover to the verifier in the interactive proof. Hence, the logarithmic TSS size is inherited from the logarithmic communication complexity of the compressed Σ-protocol.

The k-aggregation algorithm can be evaluated by any party with input at least k valid signatures from distinct signers. Besides the signatures, the k-aggregation algorithm only takes public input values. Moreover, the threshold k can be chosen at aggregation time independent of the set-up phase. By contrast, Shoup's construction [Sho00] requires a different trusted setup phase for every threshold k. Since the compressed Σ-protocol is special honest-verifier zero-knowledge, an additional property of our TSS is that a threshold signature hides the k-subset of signers S. Further, the TSS does not require a trusted setup and is therefore transparent. More precisely, the players can generate their own public-private key-pairs and the Σ-protocol only requires an unstructured public random string defined by the public parameters of the commitment scheme.

We deviate slightly from the standard TSS definitions. Therefore, in Section 7.3.1, we first formalize our security model before, in Section 7.3.2, we present our construction.

7.3.1 Definition and Security Model

We deviate from standard TSS definitions and aim for a strictly stronger functionality. In standard TSS definitions [Sho00; Bol03], a non-transparent mechanism (e.g., a trusted dealer or a multiparty computation protocol) generates a single public key and n private keys that are distributed amongst the n players. The private keys allow individual players to generate *partial* signatures on messages m. There is a public algorithm to aggregate k partial signatures into a threshold signature. The threshold signature can be verified with the public key generated by the trusted dealer.

By contrast, we define a TSS as an *extension* of a digital signature scheme. The fundamental strengthening of the definitions of [Sho00; Bol03] and related works, is that the public and private keys are generated by the players locally. Public keys are published on a *bulletin board* and thereby publicly tied to the player's identities. Since this setup does not require a trusted dealer (or another non-transparent mechanism for generating keys), it is said to be *transparent*. The players can individually sign messages by using their private keys. The aggregation algorithm now takes as input k signatures, instead of partial signatures, to generate a threshold signature. For simplicity we assume the threshold k to be fixed. We will explain later why our construction (trivially) satisfies some stronger properties.

Let us first give a definition for the basic building block of our TSS.

Definition 7.1 (Digital Signature)**.** A digital signature scheme consists of three

algorithms:

- KEYGEN is a randomized key generation algorithm that outputs a public-private key-pair $(\mathsf{pk}, \mathsf{sk})$;
- SIGN is a (possibly randomized) signing algorithm that, on input a message $m \in \{0,1\}^*$ and a secret key sk, outputs a signature $\sigma = \text{SIGN}(\mathsf{sk}, m)$;
- VERIFY is a deterministic verification algorithm that, on input a public key pk, a message m and a signature σ, outputs either accept or reject.

A signature scheme is *correct* if $\text{VERIFY}(\mathsf{pk}, m, \text{SIGN}(\mathsf{sk}, m)) = \mathsf{accept}$ with probability 1 for all key-pairs $(\mathsf{pk}, \mathsf{sk}) \leftarrow \text{KEYGEN}$ and messages $m \in \{0,1\}^*$. If $\text{VERIFY}(\mathsf{pk}, m, \sigma) = \mathsf{accept}$, we say that σ is a *valid* signature on message m. Moreover, an adversary that does not know the secret key sk should not be able to forge a valid signature. This security property is formally captured in the widely accepted definition *Existential Unforgeability under Chosen-Message Attacks* (EUF-CMA) [Bol03]. We assume digital signature schemes to be correct and EUF-CMA by definition.

Definition 7.2 (Threshold Signature)**.** A k-out-of-n threshold signature scheme (TSS) is a digital signature scheme (KEYGEN, SIGN, VERIFY) appended with two algorithms:

- k-AGGREGATE is a (possibly randomized) aggregation algorithm that, on input n public keys $(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$, k signatures $(\sigma_i)_{i \in S}$ for a k-subset $S \subseteq \{1, \ldots, n\}$ and a message $m \in \{0,1\}^*$, outputs a threshold signature Σ;
- k-VERIFY is a deterministic verification algorithm that, on input n public keys $(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$, a message m and a threshold signature Σ, outputs either accept or reject;

Let $S \subseteq \{1, \ldots, n\}$ be some k-subset of indices and let $(\sigma)_{i \in S}$ be signatures, such that $\text{VERIFY}(\mathsf{pk}_i, m, \sigma_i) = \mathsf{accept}$, for all $i \in S$, and for some message $m \in \{0,1\}^*$. Then a TSS is *correct* if for all $(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$, m, S and $(\sigma)_{i \in S}$,

$$k\text{-VERIFY}\big(\mathsf{pk}_1, \ldots, \mathsf{pk}_n, m, k\text{-AGGREGATE}(m, (\sigma_i)_{i \in S})\big) = \mathsf{accept}\,,$$

with probability 1. If $k\text{-VERIFY}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n, m, \Sigma) = \mathsf{accept}$, we say that Σ is a valid threshold signature. Moreover, an adversary with at most $k-1$ valid signatures on a message m should not be able to construct a valid threshold signature. This *unforgeability* property can be formalized by the following security game. Consider an adversary that is allowed to choose a subset of $k-1$ indices $\mathcal{I} \subset \{1, \ldots, n\}$ and impose the values of the keys pk_i in this subset. Assume that all remaining keys pk_i were generated honestly from KEYGEN and therefore correspond to secret keys sk_i. The adversary is allowed to query polynomially many signatures $\sigma_i' = \text{SIGN}(sk_i, m')$ for arbitrary messages m'. The TSS is said to be *unforgeable* if the adversary is incapable of producing a valid k-out-of-n threshold signature on some message m that has not been queried.

7.3.2 The Threshold Signature Scheme

We follow a non-standard, but conceptually simple, approach for constructing a threshold signature scheme. The starting point of our TSS is a digital signature scheme (KEYGEN, SIGN, VERIFY) and the k-aggregation algorithm k-AGGREGATE simply produces a proof of knowledge of k valid signatures on a message m, i.e., a proof of knowledge for the following relation:

$$\begin{aligned}\mathfrak{R}_T = \big\{ \, &(\mathsf{pk}_1, \ldots, \mathsf{pk}_n, m; S, (\sigma_i)_{i \in S}) : \\ &|S| = k, \ \text{VERIFY}(\mathsf{pk}_i, m, \sigma_i) = \mathsf{accept} \ \forall i \in S \big\} \, .\end{aligned} \tag{7.1}$$

The obvious approach is to capture this relation by an arithmetic circuit, i.e., reduce it to a number of constraints defined over $\mathbb{Z}_q$, and apply a communication-efficient proof of knowledge for arithmetic circuit relations in a black-box manner. A significant drawback of this *indirect* approach is that it relies on an inefficient reduction to arithmetic circuit relations. For this reason, we follow a *direct* approach avoiding these inefficient reductions.

We instantiate our TSS with the BLS signature scheme [BLS01; BLS04] defined over a bilinear group $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}, e, G, H)$. Recall that we write the group operations in $\mathbb{G}_1$ and $\mathbb{G}_2$ additively and the group operations in $\mathbb{H}$ multiplicatively. Let us now briefly recall the BLS signature scheme, instantiated in our n-player setting. All players i, $1 \leq i \leq n$, generate their own private key $u_i \in \mathbb{Z}_q$, and publish the associated public key $P_i = u_i \cdot H \in \mathbb{G}_2$. To sign a message $m \in \{0,1\}^*$, player i computes signature $\sigma_i = u_i \cdot \mathcal{H}(m) \in \mathbb{G}_1$, where $\mathcal{H} \colon \{0,1\}^* \to \mathbb{G}_1$ is some (public) collision resistant hash function. The public verification algorithm accepts a signature σ_i if

$$e(\sigma_i, H) = e(\mathcal{H}(m), P_i) \, . \tag{7.2}$$

By the bilinearity of e, all honestly generated signatures are accepted. The unforgeability follows from the so called co-CDH* assumption [BLS04].

Remark 7.1. The BLS signature scheme was originally instantiated such that $\mathbb{G}_1 = \mathbb{G}_2$, i.e., both input coordinates of the pairing e are elements of the same group. However, the authors already showed that the scheme can be instantiated in a more general setting, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are possibly different. But still, their security proof, showing that unforgeability follows from the *Computational co-Diffie-Hellman* (co-CDH) assumption, requires the existence of an efficiently computable isomorphism $\psi \colon \mathbb{G}_2 \to \mathbb{G}_1$. As discussed in Section 2.6, the existence of such an isomorphism contradicts the SXDH assumption; more precisely, the DDH assumption in $\mathbb{G}_2$ cannot hold if there exists an efficiently computable isomorphism $\psi \colon \mathbb{G}_2 \to \mathbb{G}_1$. Now recall that the binding properties of the commitment schemes of Definition 5.2 and Definition 5.3 are derived from the DDH assumption in $\mathbb{G}_2$ and the SXDH assumption, respectively. Hence, at first glance BLS signatures and these pairing-based commitments appear incompatible, i.e., they seem to require different bilinear groups. Fortunately, Boneh, Lynn and Shacham already commented on the necessity of the isomorphism ψ in the journal version of their work [BLS04]. They mention that, by relying on a slightly different complexity assumption referred to as the co-CDH* assumption [SV07], the BLS signature

scheme can also be instantiated in bilinear groups $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G, H)$ without efficiently computable isomorphisms between $\mathbb{G}_1$ and $\mathbb{G}_2$, i.e., bilinear groups of Type III [GPS08]. This shows that, under the co-CDH* assumption, we can safely instantiate the BLS signature scheme and the pairing-based commitment scheme in the same bilinear group. A more detailed analysis of certain pairing-based signature schemes, instantiated with Type III bilinear groups, is provided in [CHK+10]. In particular, they show that the co-DHP and co-DHP* assumptions are equivalent if the generators are suitably chosen and conclude that existing evidence suggests that Type III pairings offer at least as much security as Type II pairings when used to implement the BLS signature scheme.

In order to commit to mixed vectors with coefficients in both $\mathbb{Z}_q$ and $\mathbb{G}_1$, we will use the extended Pedersen vector commitment scheme of Definition 5.2:

$$\text{COM}\colon \mathbb{Z}_q^{n_0} \times \mathbb{G}_1^{n_1} \times \mathbb{Z}_q \to \mathbb{H}, \quad (\mathbf{x}, \mathbf{y}; \gamma) \mapsto \mathbf{h}^{\mathbf{x}} \cdot e(\mathbf{y}, \mathbf{g}) \cdot h^{\gamma}\,,$$

where $\mathbf{h}^{\mathbf{x}} := \prod_{i=1}^{n_0} h_i^{x_i}$ and $e(\mathbf{y}, \mathbf{g}) := \prod_{i=1}^{n_1} e(y_i, g_i)$. This commitment scheme is binding under the DDH assumption in $\mathbb{G}_1$. We do not need to be able to commit to $\mathbb{G}_2$- and $\mathbb{H}$-coefficients.

Instantiating relation $\mathfrak{R}_T$ with the BLS signature scheme therefore results in the following relation:

$$\mathfrak{R}_{TSS} = \{(P_1, \ldots, P_n, m; S, (\sigma_i)_{i\in S}) : |S| = k,\ e(\sigma_i, H) = e(\mathcal{H}(m), P_i)\ \forall i \in S\}\,.$$

The k-AGGREGATE algorithm simply computes a proof of knowledge for relation $\mathfrak{R}_{TSS}$. The main challenge is that the prover only knows k-out-of-n signatures. To handle this problem the k-out-of-n case is reduced to the n-out-of-n case by deploying the linear secret sharing based proofs of partial knowledge technique from Section 4.3. In fact, this technique allows us to reduce the nonlinear relation $\mathfrak{R}_{TSS}$ to a linear relation defined over the bilinear group $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}, e, G, H)$.

Let us recall the proof of partial knowledge technique in the context of the threshold signature relation $\mathfrak{R}_{TSS}$. First, the k-aggregator defines

$$p(X) = 1 + \sum_{j=1}^{n-k} a_j X^j \in \mathbb{Z}_q[X]$$

to be the unique polynomial of degree at most $n-k$ with $p(i) = 0$ for all $i \in \{1, \ldots, n\}\backslash S$. This polynomial defines an $(n-k+1)$-out-of-n secret sharing of 1, with shares $s_i = 0$ for all $i \notin S$. Then, the k-aggregator lets $\widetilde{\sigma}_i = p(i)\sigma_i$, where $\widetilde{\sigma}_i$ is understood to be equal to 0 for $i \notin S$, i.e., the secret sharing defined by $p(X)$ *eliminates* the signatures $(\sigma_i)_{i\notin S}$ that the k-aggregator does not know. Subsequently, the k-aggregator commits to the mixed vector

$$\mathbf{x} = (a_1, \ldots, a_{n-k}, \widetilde{\sigma}_1, \ldots, \widetilde{\sigma}_n) \in \mathbb{Z}_q^{n-k} \times \mathbb{G}_1^n\,.$$

Note that the committed vector $\mathbf{x}$ satisfies

$$f_i(\mathbf{x}) = f_i(a_1, \ldots, a_{n-k}, \widetilde{\sigma}_1, \ldots \widetilde{\sigma}_n) = e(\mathcal{H}(m), P_i)$$

for all $1 \leq i \leq n$, where

$$f_i \colon \mathbb{Z}_q^{n-k} \times \mathbb{G}_1^n \to \mathbb{H}, \quad \mathbf{x} \mapsto e(\widetilde{\sigma}_i, H) - \sum_{j=1}^{n-k} a_j i^j e(\mathcal{H}(m), P_i)\,. \tag{7.3}$$

Hence, by proving that the committed vector satisfies these relations, it follows that the k-aggregator knows a non-zero polynomial $p(X)$ of degree at most $n-k$ and group elements $\widetilde{\sigma}_1, \ldots \widetilde{\sigma}_n \in \mathbb{G}_1$ such that $e(\widetilde{\sigma}_i, H) = p(i)e(\mathcal{H}(m), P_i)$ for all $1 \leq i \leq n$. Therefore, the k-aggregator must know valid signatures for all indices i with $p(i) \neq 0$, and since $p(X)$ is non-zero and of degree at most $n-k$, at least k of its evaluations are non-zero. Because the mappings f_i are homomorphisms, the required proof of knowledge follows by applying the appropriate compressed Σ-protocol. As before, amortization can be applied to open all n homomorphisms $f_1, \ldots, f_n$ for essentially the price of one. Further, the protocol is made non-interactive by applying the Fiat-Shamir transformation. Altogether, the threshold signature contains a commitment $P \in \mathbb{H}$ to the mixed vector $\mathbf{x}$ together with a non-interactive proof of knowledge π of an opening of P that satisfies the aforementioned linear constraints. The k-AGGREGATE algorithm is summarized in Algorithm 17. The associated k-verification algorithm k-VERIFY simply runs the verifier of the compressed Σ-protocol. Correctness of the resulting threshold signature follows immediately from the completeness of the compressed Σ-protocol, and unforgeability follows from its (knowledge) soundness. The properties of the TSS are summarized in Theorem 7.5. Note that our TSS has some additional properties not required by the definition of Section 7.3.1. For instance, since the interactive proof of knowledge is special honest-verifier zero-knowledge, our threshold signatures hide the k-subset S of signers.

Theorem 7.5 (Threshold Signature Scheme)**.** *The k-out-of-n threshold signature scheme defined by the BLS signatures scheme [BLS01; BLS04], appended with the k-aggregation algorithm described in Algorithm 17, is correct and unforgeable. Moreover:*

- *a threshold signature contains exactly $4\lceil \log_2(n) \rceil + 3$ elements of $\mathbb{H}$, 1 element of $\mathbb{G}_1$ and 1 element of $\mathbb{Z}_q$;*
- *a threshold signature is zero-knowledge on the identities of the k-signers;*
- *the threshold k can be chosen at aggregation time;*
- *the threshold signature scheme resists against an adaptive adversary which, can replace the public keys of corrupted players.*

Proof. **Correctness** This immediately follows from the completeness of compressed Σ-protocol Σ_{comp}.

Unforgeability The proof is similar to the proof of Theorem 4.1, describing the properties of the proof of partial knowledge protocol. From special-soundness of the compressed Σ-protocol, it follows that there exists an efficient extractor $\mathcal{E}$ that outputs a vector $\mathbf{x}' = (\mathbf{a}', \tau_1, \ldots, \tau_n) \in \mathbb{Z}_q^{n-k} \times \mathbb{G}_1^n$ such that

Algorithm 17 Algorithm k-AGGREGATE.

PARAMETERS:	$k, n \in \mathbb{N}$, prime q, hash function $\mathcal{H}\colon \{0,1\}^* \to \mathbb{G}_1$ and bilinear group $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{H}, e, G, H)$
PUBLIC INPUT:	Public keys $P_1, \ldots, P_n \in \mathbb{G}_2$ and message $m \in \{0,1\}^*$
PRIVATE INPUT :	Subset $S \subseteq \{1, \ldots, n\}$ and signatures $\sigma_i \in \mathbb{G}_1$ $\forall i \in S$
OUTPUT :	TSS $\Sigma = (\pi, P) \in \mathbb{Z}_q \times \mathbb{G}_1 \times \mathbb{H}^{4\lceil \log_2(n) \rceil + 3} \cup \{\perp\}$

1. If $\exists i \in S$ such that $e(\sigma_i, H) \neq e(\mathcal{H}(m), P_i)$, output $\perp$ and abort.
2. Compute the unique polynomial $p(X) = 1 + \sum_{i=1}^{n-k} a_j X^j \in \mathbb{Z}_q[X]$ of degree at most $n-k$ such that $p(i) = 0$ for all $i \in \{1, \ldots, n\} \backslash S$.
3. Compute $\widetilde{\sigma}_i := p(i)\sigma_i$ for all $i \in S$ and set $\widetilde{\sigma}_i = 0$ for all $i \notin S$.
4. Let $\mathbf{x} = (a_1, \ldots, a_{n-k}, \widetilde{\sigma}_1, \ldots, \widetilde{\sigma}_n) \in \mathbb{Z}_q^{n-k} \times \mathbb{G}_1^n$ and compute commitment $P = \text{COM}(\mathbf{x}; \gamma) \in \mathbb{H}$ for $\gamma \in \mathbb{Z}_q$ sampled uniformly at random.
5. Run the non-interactive variant of compressed Σ-protocol Σ_{comp} to produce a proof π attesting that the committed vector $\mathbf{x}$ satisfies

$$f_i(\mathbf{x}) = f_i(a_1, \ldots, a_{n-k}, \widetilde{\sigma}_1, \ldots \widetilde{\sigma}_n) = e(\mathcal{H}(m), P_i)$$

   for all $1 \leq i \leq n$, where f_i are the homomorphisms defined in Equation (7.3).
6. Output commitment P and the non-interactive proof π.

$f_i(\mathbf{x}) = e(\mathcal{H}(m), P_i)$ for all $1 \leq i \leq n$, where f_i is as in Equation (7.3). Let us denote $p'(X) = 1 + \sum_{i=1}^{n-k} a'_j X^k \in \mathbb{Z}_q[X]$, then $S' = \{i : p'(i) \neq 0\}$ has cardinality at least k. Moreover, it is easily seen that $p'(i)^{-1} S_i$ is a valid BLS signature on message m associated to public key P_i. Hence, an adversary capable of forging a threshold signature is also capable of computing k distinct valid signatures on m. Since the adversary is capable of corrupting at most $k-1$ players, this contradicts the unforgeability of the BLS signature scheme.

The remaining properties are trivially verified. □

BIBLIOGRAPHY

Bibliography

[AC20] Thomas Attema and Ronald Cramer. "Compressed Σ-Protocol Theory and Practical Application to Plug & Play Secure Algorithmics." In: *CRYPTO*. Vol. 12172. Lecture Notes in Computer Science. Springer, 2020, pp. 513–543 (Cited on pages 60, 107, 123, 148, 217).

[ACF21] Thomas Attema, Ronald Cramer, and Serge Fehr. "Compressing Proofs of k-out-of-n Partial Knowledge." In: *CRYPTO*. Vol. 12828. Lecture Notes in Computer Science. Springer, 2021, pp. 65–91 (Cited on pages 60, 108, 118).

[ACK21] Thomas Attema, Ronald Cramer, and Lisa Kohl. "A Compressed Σ-Protocol Theory for Lattices." In: *CRYPTO*. Vol. 12826. Lecture Notes in Computer Science. Springer, 2021, pp. 549–579 (Cited on pages 44, 60, 123, 148, 190).

[ACR21] Thomas Attema, Ronald Cramer, and Matthieu Rambaud. "Compressed Σ-Protocols for Bilinear Group Arithmetic Circuits and Application to Logarithmic Transparent Threshold Signatures." In: *ASIACRYPT*. Vol. 13093. Lecture Notes in Computer Science. Springer, 2021, pp. 526–556 (Cited on pages 123, 218, 227).

[ACX21] Thomas Attema, Ronald Cramer, and Chaoping Xing. "A Note on Short Invertible Ring Elements and Applications to Cyclotomic and Trinomials Number Fields." In: *Mathematical Cryptology* (2021), pp. 45–70 (Cited on pages 87, 137, 167).

[ADD+19] Ittai Abraham, Srinivas Devadas, Danny Dolev, Kartik Nayak, and Ling Ren. "Synchronous Byzantine Agreement with Expected $\mathcal{O}(1)$ Rounds, Expected $\mathcal{O}(n^2)$ Communication, and Optimal Resilience." In: *Financial Cryptography and Data Security (FC)*. Vol. 11598. Lecture Notes in Computer Science. Springer, 2019, pp. 320–334 (Cited on page 218).

[AF22] Thomas Attema and Serge Fehr. "Parallel Repetition of $(k_1, \ldots, k_\mu)$-Special-Sound Multi-Round Interactive Proofs." In: *CRYPTO*. Vol. 13507. Lecture Notes in Computer Science. Springer, 2022, pp. 415–443 (Cited on page 148).

[AFG+10] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. "Structure-Preserving Signatures and Commitments to Group Elements." In: *CRYPTO*. Vol. 6223. Lecture Notes in Computer Science. Springer, 2010, pp. 209–236 (Cited on page 125).

[AFG+16] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. "Structure-Preserving Signatures and Commitments to Group Elements." In: *Journal of Cryptology* 29.2 (2016), pp. 363–421 (Cited on page 227).

[AFK22] Thomas Attema, Serge Fehr, and Michael Klooß. "Fiat-Shamir Transformation of Multi-Round Interactive Proofs." In: *Theory of Cryptography Conference (TCC)*. Vol. 13747. Lecture Notes in Computer Science. Springer, 2022, pp. 113–142 (Cited on pages 148, 190, 211).

[AHI+17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. "Ligero: Lightweight Sublinear Arguments without a Trusted Setup." In: *ACM Conference on Computer and Communications Security (CCS)*. ACM, 2017, pp. 2087–2104 (Cited on page 187).

[Ajt96] Miklós Ajtai. "Generating Hard Instances of Lattice Problems (Extended Abstract)." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 1996, pp. 99–108 (Cited on pages 37, 136).

[AL21] Martin R. Albrecht and Russell W. F. Lai. "Subtractive Sets over Cyclotomic Rings - Limits of Schnorr-Like Arguments over Lattices." In: *CRYPTO*. Vol. 12826. Lecture Notes in Computer Science. Springer, 2021, pp. 519–548 (Cited on pages 17, 150).

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. "Proof Verification and the Hardness of Approximation Problems." In: *Journal of the ACM* 45.3 (1998), pp. 501–555 (Cited on page 9).

[AM69] Michael Francis Atiyah and I. G. MacDonald. *Introduction to Commutative Algebra*. Addison-Wesley-Longman, 1969. ISBN: 978-0-201-40751-8 (Cited on page 85).

[AMS19] Ittai Abraham, Dahlia Malkhi, and Alexander Spiegelman. "Asymptotically Optimal Validated Asynchronous Byzantine Agreement." In: *ACM Symposium on Principles of Distributed Computing (PODC)*. ACM, 2019, pp. 337–346 (Cited on page 218).

[APS15] Martin R. Albrecht, Rachel Player, and Sam Scott. "On the Concrete Hardness of Learning With Errors." In: *Journal of Mathematical Cryptology* 9.3 (2015), pp. 169–203 (Cited on page 38).

[AS92] Sanjeev Arora and Shmuel Safra. "Probabilistic Checking of Proofs; A New Characterization of NP." In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 1992, pp. 2–13 (Cited on page 9).

[Bab85] László Babai. "Trading Group Theory for Randomness." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 1985, pp. 421–429 (Cited on pages 8, 255, 263).

[BBB+18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Gregory Maxwell. "Bulletproofs: Short Proofs for Confidential Transactions and More." In: *IEEE Symposium on Security and Privacy (S&P)*. IEEE Computer Society, 2018, pp. 315–334 (Cited on pages 11, 17, 59, 62, 89, 132, 187, 256, 264).

[BBC+18] Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. "Sub-Linear Lattice-Based Zero-Knowledge Arguments for Arithmetic Circuits." In: *CRYPTO*. Vol. 10992. Lecture Notes in Computer Science. Springer, 2018, pp. 669–699 (Cited on page 136).

[BCC+16] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. "Efficient Zero-Knowledge Arguments for Arithmetic Circuits in the Discrete Log Setting." In: *EUROCRYPT*. Vol. 9666. Lecture Notes in Computer Science. Springer, 2016, pp. 327–357 (Cited on pages 11, 17, 59, 62, 132, 150, 167, 187, 256, 264).

[BCK+14] Fabrice Benhamouda, Jan Camenisch, Stephan Krenn, Vadim Lyubashevsky, and Gregory Neven. "Better Zero-Knowledge Proofs for Lattice Encryption and Their Application to Group Signatures." In: *ASIACRYPT*. Vol. 8873. Lecture Notes in Computer Science. Springer, 2014, pp. 551–572 (Cited on page 134).

[BCP+14] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. "On the Existence of Extractable One-Way Functions." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 2014, pp. 505–514 (Cited on pages 15, 128).

[BCR+19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. "Aurora: Transparent Succinct Arguments for R1CS." In: *EUROCRYPT*. Vol. 11476. Lecture Notes in Computer Science. Springer, 2019, pp. 103–128 (Cited on page 187).

[BCS16] Eli Ben-Sasson, Alessandro Chiesa, and Nicholas Spooner. "Interactive Oracle Proofs." In: *Theory of Cryptography Conference (TCC)*. Vol. 9986. Lecture Notes in Computer Science. 2016, pp. 31–60 (Cited on pages 187, 188).

[BDL+18] Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. "More Efficient Commitments from Structured Lattice Assumptions." In: *International Conference on Security and Cryptography for Networks (SCN)*. Vol. 11035. Lecture Notes in Computer Science. Springer, 2018, pp. 368–385 (Cited on pages 136, 137).

[Bel53] Giovan Battista Bellaso. *La Cifra del Sig.* Venice (Italy), 1553 (Cited on page 4).

[BFS20] Benedikt Bünz, Ben Fisch, and Alan Szepieniec. "Transparent SNARKs from DARK Compilers." In: *EUROCRYPT*. Vol. 12105. Lecture Notes in Computer Science. Springer, 2020, pp. 677–706 (Cited on pages 17, 36, 132, 136).

[BG92] Mihir Bellare and Oded Goldreich. "On Defining Proofs of Knowledge." In: *CRYPTO*. Vol. 740. Lecture Notes in Computer Science. Springer, 1992, pp. 390–420 (Cited on pages 10, 42).

[BGG90] Mihir Bellare, Oded Goldreich, and Shafi Goldwasser. "Randomness in Interactive Proofs." In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 1990, pp. 563–572 (Cited on page 169).

[BGM+05] Lucas Ballard, Matthew Green, Breno de Medeiros, and Fabian Monrose. "Correlation-Resistant Storage via Keyword-Searchable Encryption." In: *IACR Cryptology ePrint Archive* (2005). IACR ePrint: 2005/417 (Cited on page 35).

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. "Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation (Extended Abstract)." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 1988, pp. 1–10 (Cited on page 7).

[BHR+21] Alexander R. Block, Justin Holmgren, Alon Rosen, Ron D. Rothblum, and Pratik Soni. "Time- and Space-Efficient Arguments from Groups of Unknown Order." In: *CRYPTO*. Vol. 12828. Lecture Notes in Computer Science. Springer, 2021, pp. 123–152 (Cited on pages 36, 132, 136).

[BIN97] Mihir Bellare, Russell Impagliazzo, and Moni Naor. "Does Parallel Repetition Lower the Error in Computationally Sound Protocols?" In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 1997, pp. 374–383 (Cited on pages 17, 18, 167).

[BKL+15] Fabrice Benhamouda, Stephan Krenn, Vadim Lyubashevsky, and Krzysztof Pietrzak. "Efficient Zero-Knowledge Proofs for Commitments from Learning with Errors over Rings." In: *European Symposium on Research in Computer Security (ESORICS)*. Vol. 9326. Lecture Notes in Computer Science. Springer, 2015, pp. 305–325 (Cited on page 136).

[BL02] Boaz Barak and Yehuda Lindell. "Strict Polynomial-Time in Simulation and Extraction." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 2002, pp. 484–493 (Cited on pages 147, 151, 191).

[BLN+20] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. "A Non-PCP Approach to Succinct Quantum-Safe Zero-Knowledge." In: *CRYPTO*. Vol. 12171. Lecture Notes in Computer Science. Springer, 2020, pp. 441–469 (Cited on pages 17, 190).

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. "Short Signatures from the Weil Pairing." In: *ASIACRYPT*. Vol. 2248. Lecture Notes in Computer Science. Springer, 2001, pp. 514–532 (Cited on pages 227, 230, 232).

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. "Short Signatures from the Weil Pairing." In: *Journal of Cryptology* 17.4 (2004), pp. 297–319 (Cited on pages 227, 230, 232).

[Blu81] Manuel Blum. "Coin Flipping by Telephone." In: *CRYPTO*. UC Santa Barbara, Department of Electrical and Computer Engineering (ECE) Report No 82-04, 1981, pp. 11–15 (Cited on page 7).

[BN06] Mihir Bellare and Gregory Neven. "Multi-Signatures in the Plain Public-Key Model and a General Forking Lemma." In: *ACM Conference on Computer and Communications Security (CCS)*. ACM, 2006, pp. 390–399 (Cited on pages 151, 191).

[Bol03] Alexandra Boldyreva. "Threshold Signatures, Multisignatures and Blind Signatures Based on the Gap-Diffie-Hellman-Group Signature Scheme." In: *Practice and Theory of Public-Key Cryptography (PKC)*. Vol. 2567. Lecture Notes in Computer Science. Springer, 2003, pp. 31–46 (Cited on pages 228, 229).

[Bon98] Dan Boneh. "The Decision Diffie-Hellman Problem." In: *Algorithmic Number Theory Symposium (ANTS)*. Vol. 1423. Lecture Notes in Computer Science. Springer, 1998, pp. 48–63 (Cited on page 35).

[BP97] Niko Baric and Birgit Pfitzmann. "Collision-Free Accumulators and Fail-Stop Signature Schemes without Trees." In: *EUROCRYPT*. Vol. 1233. Lecture Notes in Computer Science. Springer, 1997, pp. 480–494 (Cited on page 36).

[BR93] Mihir Bellare and Phillip Rogaway. "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols." In: *ACM Conference on Computer and Communications Security (CCS)*. ACM, 1993, pp. 62–73 (Cited on page 47).

[CCD88] David Chaum, Claude Crépeau, and Ivan Damgård. "Multiparty Unconditionally Secure Protocols (Extended Abstract)." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 1988, pp. 11–19 (Cited on page 7).

[CCH+19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. "Fiat-Shamir: From Practice to Theory." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 2019, pp. 1082–1090 (Cited on pages 19, 188).

[CD98] Ronald Cramer and Ivan Damgård. "Zero-Knowledge Proofs for Finite Field Arithmetic; or: Can Zero-Knowledge be for Free?" In: *CRYPTO*. Vol. 1462. Lecture Notes in Computer Science. Springer, 1998, pp. 424–441 (Cited on pages 10, 11, 60, 61, 255, 263).

[CDG87] David Chaum, Ivan Damgård, and Jeroen van de Graaf. "Multiparty Computations Ensuring Privacy of Each Party's Input and Correctness of the Result." In: *CRYPTO*. Vol. 293. Lecture Notes in Computer Science. Springer, 1987, pp. 87–119 (Cited on page 7).

[CDM00] Ronald Cramer, Ivan Damgård, and Ueli M. Maurer. "General Secure Multi-party Computation from any Linear Secret-Sharing Scheme." In: *EUROCRYPT*. Vol. 1807. Lecture Notes in Computer Science. Springer, 2000, pp. 316–334 (Cited on pages 107, 256, 264).

[CDN15] Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015. ISBN: 9781107043053 (Cited on pages 31, 52, 53, 109, 119, 256, 264).

[CDP12] Ronald Cramer, Ivan Damgård, and Valerio Pastro. "On the Amortized Complexity of Zero Knowledge Protocols for Multiplicative Relations." In: *International Conference on Information Theoretic Security (ICITS)*. Vol. 7412. Lecture Notes in Computer Science. Springer, 2012, pp. 62–79 (Cited on pages 13, 14, 107–110, 112, 218, 256, 264).

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. "Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols." In: *CRYPTO*. Vol. 839. Lecture Notes in Computer Science. Springer, 1994, pp. 174–187 (Cited on pages 10, 15, 107, 114–116, 119).

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. "The Random Oracle Methodology, Revisited." In: *Journal of the ACM* 51.4 (2004), pp. 557–594 (Cited on page 19).

[CHK+10] Sanjit Chatterjee, Darrel Hankerson, Edward Knapp, and Alfred Menezes. "Comparing Two Pairing-Based Aggregate Signature Schemes." In: *Designs, Codes and Cryptography* 55.2-3 (2010), pp. 141–167 (Cited on page 231).

[CHR+16] Ming-Shing Chen, Andreas Hülsing, Joost Rijneveld, Simona Samardjiska, and Peter Schwabe. "From 5-Pass MQ-Based Identification to MQ-Based Signatures." In: *ASIACRYPT*. Vol. 10032. Lecture Notes in Computer Science. 2016, pp. 135–165 (Cited on page 172).

[Chu36] Alonzo Church. "An Unsolvable Problem of Elementary Number Theory." In: *American Journal of Mathematics* 58.2 (1936), pp. 345–363 (Cited on page 4).

[CKS00] Christian Cachin, Klaus Kursawe, and Victor Shoup. "Random Oracles in Constantipole: Practical Asynchronous Byzantine Agreement using Cryptography (Extended Abstract)." In: *ACM Symposium on Principles of Distributed Computing (PODC)*. ACM, 2000, pp. 123–132 (Cited on page 218).

[CKS05] Christian Cachin, Klaus Kursawe, and Victor Shoup. "Random Oracles in Constantinople: Practical Asynchronous Byzantine Agreement using Cryptography." In: *Journal of Cryptology* 18.3 (2005), pp. 219–246 (Cited on page 218).

[CL10] Kai-Min Chung and Feng-Hao Liu. "Parallel Repetition Theorems for Interactive Arguments." In: *Theory of Cryptography Conference (TCC)*. Vol. 5978. Lecture Notes in Computer Science. Springer, 2010, pp. 19–36 (Cited on pages 18, 167).

[CMS19] Alessandro Chiesa, Peter Manohar, and Nicholas Spooner. "Succinct Arguments in the Quantum Random Oracle Model." In: *Theory of Cryptography Conference (TCC)*. Vol. 11892. Lecture Notes in Computer Science. Springer, 2019, pp. 1–29 (Cited on page 188).

[Coo71] Stephen A. Cook. "The Complexity of Theorem-Proving Procedures." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 1971, pp. 151–158 (Cited on page 8).

[CP15] Kai-Min Chung and Rafael Pass. "Tight Parallel Repetition Theorems for Public-Coin Arguments using KL-Divergence." In: *Theory of Cryptography Conference (TCC)*. Vol. 9015. Lecture Notes in Computer Science. Springer, 2015, pp. 229–246 (Cited on pages 18, 167–170).

[CR79] Stephen A. Cook and Robert A. Reckhow. "The Relative Efficiency of Propositional Proof Systems." In: *Journal of Symbolic Logic* 44.1 (1979), pp. 36–50 (Cited on page 8).

[Cra96] Ronald Cramer. "Modular Design of Secure yet Practical Cryptographic Protocols." PhD thesis. CWI and University of Amsterdam, 1996 (Cited on pages 10, 16, 42, 59–61, 149, 151, 152, 255, 257, 263, 265).

[Dam10] Ivan Damgård. *On Σ-Protocols*. Lecture Notes, Aarhus University, Department of Computer Science. 2010 (Cited on page 149).

[Dam93] Ivan Damgård. "Interactive Hashing can Simplify Zero-Knowledge Protocol Design without Computational Assumptions (Extended Abstract)." In: *CRYPTO*. Vol. 773. Lecture Notes in Computer Science. Springer, 1993, pp. 100–109 (Cited on page 45).

[DF02] Ivan Damgård and Eiichiro Fujisaki. "A Statistically-Hiding Integer Commitment Scheme Based on Groups with Hidden Order." In: *ASIACRYPT*. Vol. 2501. Lecture Notes in Computer Science. Springer, 2002, pp. 125–142 (Cited on page 132).

[DFM+19] Jelle Don, Serge Fehr, Christian Majenz, and Christian Schaffner. "Security of the Fiat-Shamir Transformation in the Quantum Random-Oracle Model." In: *CRYPTO*. Vol. 11693. Lecture Notes in Computer Science. Springer, 2019, pp. 356–383 (Cited on page 50).

[DGO+95] Ivan Damgård, Oded Goldreich, Tatsuaki Okamoto, and Avi Wigderson. "Honest Verifier vs Dishonest Verifier in Public Coin Zero-Knowledge Proofs." In: *CRYPTO*. Vol. 963. Lecture Notes in Computer Science. Springer, 1995, pp. 325–338 (Cited on page 45).

[DH76] Whitfield Diffie and Martin E. Hellman. "New Directions in Cryptography." In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654 (Cited on page 6).

[Din07] Irit Dinur. "The PCP Theorem by Gap Amplification." In: *Journal of the ACM* 54.3 (2007), 12–es (Cited on page 9).

[DJM+12] Yevgeniy Dodis, Abhishek Jain, Tal Moran, and Daniel Wichs. "Counterexamples to Hardness Amplification Beyond Negligible." In: *Theory of Cryptography Conference (TCC)*. Vol. 7194. Lecture Notes in Computer Science. Springer, 2012, pp. 476–493 (Cited on pages 168, 170).

[DKL+18] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. "CRYSTALS-Dilithium: A Lattice-Based Digital Signature Scheme." In: *Transactions on Cryptographic Hardware and Embedded Systems (TCHES)* 2018.1 (2018), pp. 238–268 (Cited on page 140).

[DOT+21] Ivan Damgård, Claudio Orlandi, Akira Takahashi, and Mehdi Tibouchi. "Two-Round n-out-of-n and Multi-Signatures and Trapdoor Commitment from Lattices." In: *Practice and Theory of Public-Key Cryptography (PKC)*. Vol. 12710. Lecture Notes in Computer Science. Springer, 2021, pp. 99–130 (Cited on page 78).

[ElG84] Taher ElGamal. "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms." In: *CRYPTO*. Vol. 196. Lecture Notes in Computer Science. Springer, 1984, pp. 10–18 (Cited on page 126).

[ESS+19] Muhammed F. Esgin, Ron Steinfeld, Amin Sakzad, Joseph K. Liu, and Dongxi Liu. "Short Lattice-Based One-out-of-Many Proofs and Applications to Ring Signatures." In: *Applied Cryptography and Network Security (ACNS)*. Vol. 11464. Lecture Notes in Computer Science. Springer, 2019, pp. 67–88 (Cited on page 38).

[FFS88] Uriel Feige, Amos Fiat, and Adi Shamir. "Zero-Knowledge Proofs of Identity." In: *Journal of Cryptology* 1.2 (1988), pp. 77–94 (Cited on page 10).

[FO97] Eiichiro Fujisaki and Tatsuaki Okamoto. "Statistical Zero Knowledge Protocols to Prove Modular Polynomial Relations." In: *CRYPTO*. Vol. 1294. Lecture Notes in Computer Science. Springer, 1997, pp. 16–30 (Cited on page 132).

[FS86] Amos Fiat and Adi Shamir. "How to Prove Yourself: Practical Solutions to Identification and Signature Problems." In: *CRYPTO*. Vol. 263. Lecture Notes in Computer Science. Springer, 1986, pp. 186–194 (Cited on pages 10, 12, 18, 45, 50, 61, 228).

[Gál95] Anna Gál. "Combinatorial Methods in Boolean Function Complexity." PhD thesis. University of Chicago, 1995 (Cited on page 119).

[Gen09] Craig Gentry. "Fully Homomorphic Encryption using Ideal Lattices." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 2009, pp. 169–178 (Cited on page 7).

[GH98] Oded Goldreich and Johan Håstad. "On the Complexity of Interactive Proofs with Bounded Communication." In: *Information Processing Letters* 67.4 (1998), pp. 205–214 (Cited on page 9).

[GK96] Oded Goldreich and Ariel Kahan. "How to Construct Constant-Round Zero-Knowledge Proof Systems for NP." In: *Journal of Cryptology* 9.3 (1996), pp. 167–190 (Cited on page 167).

[GM82] Shafi Goldwasser and Silvio Micali. "Probabilistic Encryption and How to Play Mental Poker Keeping Secret All Partial Information." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 1982, pp. 365–377 (Cited on page 7).

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The Knowledge Complexity of Interactive Proof-Systems (Extended Abstract)." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 1985, pp. 291–304 (Cited on pages 8–10, 39, 255, 263).

[GMW86] Oded Goldreich, Silvio Micali, and Avi Wigderson. "Proofs that Yield Nothing But their Validity and a Methodology of Cryptographic Protocol Design (Extended Abstract)." In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 1986, pp. 174–187 (Cited on pages 9, 11).

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. "How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 1987, pp. 218–229 (Cited on page 7).

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. "Proofs that Yield Nothing But Their Validity for All Languages in NP Have Zero-Knowledge Proof Systems." In: *Journal of the ACM* 38.3 (1991), pp. 691–729 (Cited on page 9).

[Göd31] Kurt Gödel. "Über Formal Unentscheidbare Sätze der Principia Mathematica und Verwandter Systeme I." In: *Monatshefte für Mathematik und Physik* 38.1 (1931), pp. 173–198 (Cited on page 3).

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1: Basic Techniques*. Cambridge University Press, 2001. ISBN: 0-521-79172-3 (Cited on page 167).

[Gol04] Oded Goldreich. *The Foundations of Cryptography - Volume 2: Basic Applications*. Cambridge University Press, 2004. ISBN: 0-521-83084-2 (Cited on pages 40–42, 164).

[Gol98] Oded Goldreich. *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*. Vol. 17. Algorithms and Combinatorics. Springer, 1998 (Cited on page 169).

[GPS08] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. "Pairings for Cryptographers." In: *Discrete Applied Mathematics* 156.16 (2008), pp. 3113–3121 (Cited on pages 36, 231).

[GQ88] Louis C. Guillou and Jean-Jacques Quisquater. "A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Transmission and Memory." In: *EUROCRYPT*. Vol. 330. Lecture Notes in Computer Science. Springer, 1988, pp. 123–128 (Cited on page 10).

[Gro03] Jens Groth. "A Verifiable Secret Shuffle of Homomorphic Encryptions." In: *Practice and Theory of Public-Key Cryptography (PKC)*. Vol. 2567. Lecture Notes in Computer Science. Springer, 2003, pp. 145–160 (Cited on page 74).

[Gro05] Jens Groth. "A Verifiable Secret Shuffle of Homomorphic Encryptions." In: *IACR Cryptology ePrint Archive* (2005). IACR ePrint: 2005/246 (Cited on page 74).

[Gro10] Jens Groth. "Short Pairing-Based Non-Interactive Zero-Knowledge Arguments." In: *ASIACRYPT*. Vol. 6477. Lecture Notes in Computer Science. Springer, 2010, pp. 321–340 (Cited on pages 129, 131).

[GS86] Shafi Goldwasser and Michael Sipser. "Private Coins versus Public Coins in Interactive Proof Systems." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 1986, pp. 59–68 (Cited on page 9).

[GT21] Ashrujit Ghoshal and Stefano Tessaro. "Tight State-Restoration Soundness in the Algebraic Group Model." In: *CRYPTO*. Vol. 12827. Lecture Notes in Computer Science. Springer, 2021, pp. 64–93 (Cited on pages 19, 188).

[Hai09] Iftach Haitner. "A Parallel Repetition Theorem for Any Interactive Argument." In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 2009, pp. 241–250 (Cited on page 18).

[HBH+20] Daira Hopwood, Sean Bowe, Taylor Hornby, and Nathan Wilcox. *Zcash Protocol Specification - Version 2020.1.7*. Aug. 30, 2020 (Cited on page 227).

[HKR19] Max Hoffmann, Michael Klooß, and Andy Rupp. "Efficient Zero-Knowledge Arguments in the Discrete Log Setting, Revisited." In: *ACM Conference on Computer and Communications Security (CCS)*. ACM, 2019, pp. 2093–2110 (Cited on pages 17, 150).

[HL10] Carmit Hazay and Yehuda Lindell. *Efficient Secure Two-Party Protocols - Techniques and Constructions*. Information Security and Cryptography. Springer, 2010. ISBN: 978-3-642-14302-1 (Cited on pages 41, 42, 149, 164).

[HM98] Shai Halevi and Silvio Micali. "More on Proofs of Knowledge." In: *IACR Cryptology ePrint Archive* (1998). IACR ePrint: 1998/015 (Cited on page 42).

[HPW+10] Johan Håstad, Rafael Pass, Douglas Wikström, and Krzysztof Pietrzak. "An Efficient Parallel Repetition Theorem." In: *Theory of Cryptography Conference (TCC)*. Vol. 5978. Lecture Notes in Computer Science. Springer, 2010, pp. 1–18 (Cited on pages 18, 167).

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. "Pseudo-Random Generation from One-Way Functions (Extended Abstracts)." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 1989, pp. 12–24 (Cited on page 138).

[JT20] Joseph Jaeger and Stefano Tessaro. "Expected-Time Cryptography: Generic Techniques and Applications to Concrete Soundness." In: *Theory of Cryptography Conference (TCC)*. Vol. 12552. Lecture Notes in Computer Science. Springer, 2020, pp. 414–443 (Cited on pages 17, 150).

[Kas63] F.W. Kasiski. *Die Geheimschriften und die Dechiffrir-Kunst: Mit Besonderer Berücksichtigung der Deutschen und der Französischen Sprache*. E. S. Mittler und Sohn, 1863 (Cited on page 5).

[Kil92] Joe Kilian. "A Note on Efficient Zero-Knowledge Proofs and Arguments (Extended Abstract)." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 1992, pp. 723–732 (Cited on pages 9, 11).

[Lan02] Serge Lang. *Algebra*. 3rd ed. Graduate Texts in Mathematics. Originally published by Addison-Wesley (1993). Springer New York, NY, 2002. ISBN: 978-0-387-95385-4 (Cited on page 27).

[Lev73] Leonid Anatolevich Levin. "Universal Sequential Search Problems (in Russian)." In: *Problemy Peredachi Informatsii* 9.3 (1973), pp. 115–116 (Cited on page 8).

[Lin01] Yehuda Lindell. "Parallel Coin-Tossing and Constant-Round Secure Two-Party Computation." In: *CRYPTO*. Vol. 2139. Lecture Notes in Computer Science. Springer, 2001, pp. 171–189 (Cited on page 167).

[Lin03] Yehuda Lindell. "Parallel Coin-Tossing and Constant-Round Secure Two-Party Computation." In: *Journal of Cryptology* 16.3 (2003), pp. 143–184 (Cited on page 167).

[LL93] Arjen K. Lenstra and Hendrik W. Lenstra. *The Development of the Number Field Sieve*. Springer Berlin, Heidelberg, 1993. ISBN: 978-3-540-57013-4 (Cited on page 6).

[LM06] Vadim Lyubashevsky and Daniele Micciancio. "Generalized Compact Knapsacks are Collision Resistant." In: *International Colloquium on Automata, Languages, and Programming (ICALP)*. Vol. 4052. Lecture Notes in Computer Science. Springer, 2006, pp. 144–155 (Cited on page 37).

[LM18] Julian Loss and Tal Moran. "Combining Asynchronous and Synchronous Byzantine Agreement: The Best of Both Worlds." In: *IACR Cryptology ePrint Archive* (2018). IACR ePrint: 2018/235 (Cited on page 218).

[LMR19] Russell W. F. Lai, Giulio Malavolta, and Viktoria Ronge. "Succinct Arguments for Bilinear Group Arithmetic: Practical Structure-Preserving Cryptography." In: *ACM Conference on Computer and Communications Security (CCS)*. ACM, 2019, pp. 2057–2074 (Cited on pages 125–127, 226, 227).

[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 (Cited on page 37).

[LS18] Vadim Lyubashevsky and Gregor Seiler. "Short, Invertible Elements in Partially Splitting Cyclotomic Rings and Applications to Lattice-Based Zero-Knowledge Proofs." In: *EUROCRYPT*. Vol. 10820. Lecture Notes in Computer Science. Springer, 2018, pp. 204–224 (Cited on pages 87, 137, 167).

[Lyu09] Vadim Lyubashevsky. "Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures." In: *ASIACRYPT*. Vol. 5912. Lecture Notes in Computer Science. Springer, 2009, pp. 598–616 (Cited on page 74).

[Lyu12] Vadim Lyubashevsky. "Lattice Signatures without Trapdoors." In: *EUROCRYPT*. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 738–755 (Cited on page 74).

[MBK+19] Mary Maller, Sean Bowe, Markulf Kohlweiss, and Sarah Meiklejohn. "Sonic: Zero-Knowledge SNARKs from Linear-Size Universal and Updatable Structured Reference Strings." In: *ACM Conference on Computer and Communications Security (CCS)*. ACM, 2019, pp. 2111–2128 (Cited on page 17).

[MR09] Daniele Micciancio and Oded Regev. "Lattice-Based Cryptography." In: *Post-Quantum Cryptography*. Springer Berlin Heidelberg, 2009, pp. 147–191 (Cited on page 38).

[MV03] Daniele Micciancio and Salil P. Vadhan. "Statistical Zero-Knowledge Proofs with Efficient Provers: Lattice Problems and More." In: *CRYPTO*. Vol. 2729. Lecture Notes in Computer Science. Springer, 2003, pp. 282–298 (Cited on page 10).

[Nao03] Moni Naor. "On Cryptographic Assumptions and Challenges." In: *CRYPTO*. Vol. 2729. Lecture Notes in Computer Science. Springer, 2003, pp. 96–109 (Cited on pages 15, 128).

[Oka92] Tatsuaki Okamoto. "Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes." In: *CRYPTO*. Vol. 740. Lecture Notes in Computer Science. Springer, 1992, pp. 31–53 (Cited on page 10).

[OVY93] Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. "Interactive Hashing Simplifies Zero-Knowledge Protocol Design." In: *EUROCRYPT*. Vol. 765. Lecture Notes in Computer Science. Springer, 1993, pp. 267–273 (Cited on page 45).

[OW93] Rafail Ostrovsky and Avi Wigderson. "One-Way Fuctions are Essential for Non-Trivial Zero-Knowledge." In: *Israel Symposium on Theory of Computing Systems (ISTCS)*. IEEE Computer Society, 1993, pp. 3–17 (Cited on page 9).

[Ped91] Torben P. Pedersen. "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing." In: *CRYPTO*. Vol. 576. Lecture Notes in Computer Science. Springer, 1991, pp. 129–140 (Cited on pages 123, 124).

[PLS19] Rafaël del Pino, Vadim Lyubashevsky, and Gregor Seiler. "Short Discrete Log Proofs for FHE and Ring-LWE Ciphertexts." In: *Practice and Theory of Public-Key Cryptography (PKC)*. Vol. 11442. Lecture Notes in Computer Science. Springer, 2019, pp. 344–373 (Cited on pages 17, 150).

[PR06] Chris Peikert and Alon Rosen. "Efficient Collision-Resistant Hashing from Worst-Case Assumptions on Cyclic Lattices." In: *Theory of Cryptography Conference (TCC)*. Vol. 3876. Lecture Notes in Computer Science. Springer, 2006, pp. 145–166 (Cited on page 37).

[PS96] David Pointcheval and Jacques Stern. "Security Proofs for Signature Schemes." In: *EUROCRYPT*. Vol. 1070. Lecture Notes in Computer Science. Springer, 1996, pp. 387–398 (Cited on page 191).

[PV07] Rafael Pass and Muthuramakrishnan Venkitasubramaniam. "An Efficient Parallel Repetition Theorem for Arthur-Merlin Games." In: *ACM Symposium on Theory of Computing (STOC)*. ACM, 2007, pp. 420–429 (Cited on page 18).

[PV12] Rafael Pass and Muthuramakrishnan Venkitasubramaniam. "A Parallel Repetition Theorem for Constant-Round Arthur-Merlin Proofs." In: *ACM Transactions on Compututation Theory (TOCT)* 4.4 (2012), 10:1–10:22 (Cited on page 18).

[PW07] Krzysztof Pietrzak and Douglas Wikström. "Parallel Repetition of Computationally Sound Protocols Revisited." In: *Theory of Cryptography Conference (TCC)*. Vol. 4392. Lecture Notes in Computer Science. Springer, 2007, pp. 86–102 (Cited on pages 17, 167).

[RAD78] Ronald L. Rivest, Len Adleman, and Michael L. Dertouzos. "On Data Banks and Privacy Homomorphisms." In: *Foundations of Secure Computation* 4.11 (1978), pp. 169–180 (Cited on page 7).

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." In: *Communications of the ACM* 21.2 (1978), pp. 120–126 (Cited on page 6).

[Sch91] Claus-Peter Schnorr. "Efficient Signature Generation by Smart Cards." In: *Journal of Cryptology* 4.3 (1991), pp. 161–174 (Cited on page 10).

[Sha48a] Claude E. Shannon. "A Mathematical Theory of Communication." In: *Bell System Technical Journal* 27.3 (1948), pp. 379–423 (Cited on page 5).

[Sha48b] Claude E. Shannon. "A Mathematical Theory of Communication." In: *Bell System Technical Journal* 27.4 (1948), pp. 623–656 (Cited on page 5).

[Sha49] Claude E. Shannon. "Communication Theory of Secrecy Systems." In: *Bell System Technical Journal* 28.4 (1949), pp. 656–715 (Cited on page 5).

[Sha79] Adi Shamir. "How to Share a Secret." In: *Communications of the ACM* 22.11 (1979), pp. 612–613 (Cited on page 53).

[Sho00] Victor Shoup. "Practical Threshold Signatures." In: *EUROCRYPT*. Vol. 1807. Lecture Notes in Computer Science. Springer, 2000, pp. 207–220 (Cited on pages 218, 228).

[Sho94] Peter W. Shor. "Algorithms for Quantum Computation: Discrete Logarithms and Factoring." In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 1994, pp. 124–134 (Cited on pages 5, 7, 36).

[SJM91] Gustavus J. Simmons, Wen-Ai Jackson, and Keith M. Martin. "The Geometry of Shared Secret Schemes." In: *Bulletin of the Institute of Combinatorics and its Applications* 1 (1991), pp. 71–88 (Cited on page 119).

[SRA81] Adi Shamir, Ronald L Rivest, and Leonard M Adleman. "Mental Poker." In: *The Mathematical Gardner*. Springer, 1981, pp. 37–43 (Cited on page 7).

[SSH11] Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. "Public-Key Identification Schemes Based on Multivariate Quadratic Polynomials." In: *CRYPTO*. Vol. 6841. Lecture Notes in Computer Science. Springer, 2011, pp. 706–723 (Cited on page 172).

[SV07] Nigel P. Smart and Frederik Vercauteren. "On Computable Isomorphisms in Efficient Asymmetric Pairing-Based Systems." In: *Discrete Applied Mathematics* 155.4 (2007), pp. 538–547 (Cited on page 230).

[SW05] Amit Sahai and Brent Waters. "Fuzzy Identity-Based Encryption." In: *EUROCRYPT*. Vol. 3494. Lecture Notes in Computer Science. Springer, 2005, pp. 457–473 (Cited on page 227).

[Tur36] Alan Mathison Turing. "On Computable Numbers, with an Application to the Entscheidungsproblem." In: *Proceedings of the London Mathematical Society* 42.2 (1936), pp. 230–265 (Cited on page 4).

[TW87] Martin Tompa and Heather Woll. "Random Self-Reducibility and Zero Knowledge Interactive Proofs of Possession of Information." In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 1987, pp. 472–482 (Cited on page 10).

[Unr12] Dominique Unruh. "Quantum Proofs of Knowledge." In: *EUROCRYPT*. Vol. 7237. Lecture Notes in Computer Science. Springer, 2012, pp. 135–152 (Cited on page 42).

[Unr17] Dominique Unruh. "Post-Quantum Security of Fiat-Shamir." In: *ASIACRYPT*. Vol. 10624. Lecture Notes in Computer Science. Springer, 2017, pp. 65–95 (Cited on page 50).

[Wes19] Benjamin Wesolowski. "Efficient Verifiable Delay Functions." In: *EUROCRYPT*. Vol. 11478. Lecture Notes in Computer Science. Springer, 2019, pp. 379–407 (Cited on pages 36, 136).

[Wig19] Avi Wigderson. *Mathematics and Computation*. Princeton University Press, 2019. ISBN: 978-0-691-18913-0 (Cited on page 11).

[Wik18] Douglas Wikström. "Special Soundness Revisited." In: *IACR Cryptology ePrint Archive* (2018). IACR ePrint: 2018/1157 (Cited on pages 150, 190).

[Wik21] Douglas Wikström. "Special Soundness in the Random Oracle Model." In: *IACR Cryptology ePrint Archive* (2021). IACR ePrint: 2021/1265 (Cited on page 190).

[Yao82] Andrew Chi-Chih Yao. "Protocols for Secure Computations (Extended Abstract)." In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 1982, pp. 160–164 (Cited on page 7).

[Yao86] Andrew Chi-Chih Yao. "How to Generate and Exchange Secrets (Extended Abstract)." In: *IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 1986, pp. 162–167 (Cited on page 7).

[YMR+19] Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan-Gueta, and Ittai Abraham. "HotStuff: BFT Consensus with Linearity and Responsiveness." In: *ACM Symposium on Principles of Distributed Computing (PODC)*. ACM, 2019, pp. 347–356 (Cited on page 218).

SUMMARY

Summary

The field of *(probabilistic) proof systems* has developed into a flourishing subfield of cryptology and computer science. In analogy to mathematical proofs, the goal of a proof system is for a prover to convince a verifier of the correctness of a claim. However, by contrast, probabilistic proofs allow the verifier to make mistakes, i.e., to accept false claims (soundness error) or reject true claims (completeness error). In many occasions, the error probability can be made negligibly small by repetition, causing only a minor loss in efficiency, which is sufficient for most practical applications. Further, probabilistic proofs may have multiple rounds of interaction between the prover and the verifier, in which case they are also referred to as *interactive proofs*. These two relaxations, due to Babai, Goldwasser, Micali and Rackoff [Bab85; GMR85], revolutionized the theory of proofs. For instance, by trading absolute certainty for high probability and allowing interaction, it is possible to prove claims without revealing anything beyond their correctness, i.e., in *zero-knowledge*. Nowadays, zero-knowledge proofs are widely deployed; they are for instance essential in the public-key infrastructures (PKIs) that manage digital identities and secure communication channels on the internet.

Especially the theory of Σ-protocols [Cra96] now provides a well-understood basis for the modular design of zero-knowledge proof systems in a wide variety of application domains. A Σ-protocol is an interactive proof with three rounds; the prover first sends a message to the verifier, who replies with a challenge sampled uniformly at random from some finite set, and after receiving the prover's response the verifier decides whether to accept or reject the prover's claim. The theory of Σ-protocols stands out in its *modularity*; basic Σ-protocols are elegant and easy to analyze, and complex application scenarios are handled by appropriately combining these basic building blocks. This includes proving the satisfiability of an arithmetic circuit $C\colon \mathbb{Z}_q^n \to \mathbb{Z}_q$ [CD98], where $\mathbb{Z}_q$ denotes the ring of integers modulo q. More precisely, it includes proving that C admits a satisfiable input $\mathbf{x} \in \mathbb{Z}_q^n$ such that $C(\mathbf{x}) = 0$. In fact, Σ-protocols even offer a stronger functionality; they allow provers to not only prove that a circuit admits a satisfiable input, but also that they know one. This property is referred to as *knowledge soundness*, and knowledge sound interactive proofs are called *proofs of knowledge*. The circuit satisfiability problem is NP-complete, i.e., every problem for which solutions are efficiently verifiable can be written as a circuit satisfiability problem. Therefore, by means of a Σ-protocol, every efficiently verifiable claim can be proven in zero-knowledge. However, due to the modularity of Σ-protocol theory, there are often more direct and more efficient solutions that avoid the oftentimes cumbersome reduction to a circuit satisfiability problem.

Probabilistic proofs have various performance metrics, indicating for instance the (computational) complexity of generating or verifying a proof. The communi-

cation costs define another important performance metric, i.e., the number of bits communicated between the prover and the verifier. Unfortunately, for many application scenarios, the communication costs of standard Σ-protocols grow *linearly* in the size of the problem instance. For instance, the communication complexity of a Σ-protocol for the circuit satisfiability problem is linear in the size of the arithmetic circuit. More recently, a folding technique was introduced to reduce the communication complexity from linear down to logarithmic in the size of the problem instance [BCC+16; BBB+18]. The resulting protocols are referred to as Bulletproofs. Bulletproofs were introduced as a "drop-in replacement" for Σ-protocols in several applications, such as zero-knowledge proofs for arithmetic circuit satisfiability.

In this dissertation, we reconcile Bulletproofs' folding technique with the established theory of Σ-protocols. We show that the folding technique can be cast as a significant *strengthening*, rather than a replacement, of Σ-protocols. Our starting point is a basic Σ-protocol for proving knowledge of a preimage of a group homomorphism $\Psi\colon \mathbb{G}^n \to \mathbb{H}$. More precisely, this Σ-protocol allows a prover to prove knowledge of a secret input vector $\mathbf{x} \in \mathbb{G}^n$ such that $\Psi(\mathbf{x}) = P$ for some public $P \in \mathbb{H}$, with communication complexity linear in $n \in \mathbb{N}$. Subsequently, we show that, by an appropriate adaptation of Bulletproofs' folding technique, the communication complexity can be reduced down to logarithmic in n (or polylogarithmic depending on the concrete instantiation). In line with Bulletproofs, this reduction comes at the expense of a logarithmic number of rounds, instead of constant. Since the compression mechanism is cast as an extension of a basic Σ-protocol, many techniques well known from Σ-protocol theory directly carry over to this new *compressed* Σ-protocol theory.

Further, we enhance compressed Σ-protocol theory with two higher level functionalities. First, by an arithmetic secret-sharing based technique, we show how to prove the correctness of m multiplication triples $(\alpha_i, \beta_i, \gamma_i = \alpha_i \cdot \beta_i) \in \mathbb{Z}_q^3$ for $1 \leq i \leq m$. More precisely, proving correctness of multiplication triples is reduced to proving knowledge of a homomorphism preimage, i.e., the nonlinear multiplication triple relation is linearized. This approach is known from Σ-protocol theory [CDM00; CDP12] and inspired by secure multiparty computation [CDN15], however, some adaptations are required to make it amenable for compression. By an appropriate and efficient reduction, we show that this functionality enhancement is sufficient for proving the satisfiability of an arithmetic circuit in (poly)logarithmic communication. As a second functionality enhancement, we construct a novel k-out-of-n proof of partial knowledge, allowing to prove knowledge of k-out-of-n homomorphism preimages without revealing which preimages the prover knows. Proofs of partial knowledge, especially 1-out-of-n, have seen myriad applications during the last decades, e.g., in electronic voting, ring signatures, and confidential transaction systems. Our construction shows how to reduce their communication complexity from linear down to (poly)logarithmic in k and n. We avoid the use of generic circuit satisfiability machinery and identify regimes of practical relevance where our approach achieves asymptotic and concrete performance improvements.

Compressed Σ-protocol theory is presented in a simple and abstract language, allowing for instantiations in a variety of cryptographic platforms. In particular,

we show how to instantiate compressed Σ-protocols from the discrete logarithm assumption, resulting in a logarithmic communication complexity. Moreover, we show how to extend this instantiation to bilinear pairing based platforms. Based on the knowledge of exponent assumption, the communication complexity can be reduced further down to constant. Finally, we present strong-RSA and lattice-based instantiations, the latter plausibly providing post-quantum security. Strong-RSA and lattice-based instantiations are subject to a so-called *soundness slack*. This warrants larger protocol parameters and causes the resulting communication complexity to be polylogarithmic rather than logarithmic or constant.

Additionally, we identify and close three gaps in the general theory of multi-round interactive proofs, with particular relevance to Bulletproofs and compressed Σ-protocols. More precisely, it is generally nontrivial to show that an interactive proof is knowledge sound and to find a tight bound on the knowledge error, i.e., the success probability of a dishonest prover. Therefore, in the context of Σ-protocols, the more convenient notion *special-soundness* was introduced [Cra96]. It is well known that special-soundness, or more precisely 2-out-of-N special-soundness, implies knowledge soundness with knowledge error $1/N$, where N is the size of the verifier's challenge set. More generally, k-out-of-N special-soundness implies knowledge soundness with knowledge error $(k-1)/N$. Bulletproofs and compressed Σ-protocols have rendered natural *multi-round* generalizations of special-soundness relevant.

The first open problem that we address is the lack of a *tight* knowledge soundness analysis for special-sound multi-round interactive proofs. Non-tight bounds on the knowledge error warrant the use of overly conservative protocol parameters, possibly rendering concrete instantiations inefficient. We provide the first tight knowledge soundness analysis for the broad class of special-sound multi-round interactive proofs.

The second open problem questions the effect of parallel repetition on the knowledge error. In many occasions, the knowledge error κ is not small enough, and thus needs to be reduced. This can be done generically by repeating the interactive proof in parallel. The effect of parallel repetition on 2-out-of-N special-sound Σ-protocols is well known, but the situation becomes significantly more complicated when considering k-out-of-N special-soundness for $k > 2$, let alone its multi-round generalizations. More precisely, the t-fold parallel repetition of a 2-out-of-N special-sound interactive proof is easily seen to be 2-out-of-N^t special-sound, and thus has knowledge error $1/N^t$. A similar result does not hold for the (multi-round) generalizations of special-soundness. We solve the state-of-affairs by proving that, for all special-sound interactive proofs, t-fold parallel repetition optimally reduces the knowledge error from κ down to κ^t.

Third, we analyze the Fiat-Shamir transformation of special-sound multi-round interactive proofs. The Fiat-Shamir transformation is a commonly used heuristic that renders a public-coin[1] interactive proof non-interactive by replacing the verifier's messages by certain hash function evaluations. Unfortunately, the Fiat-Shamir transformation comes with a security loss; in general, the security loss is *exponential* in the number of rounds of the interactive proof. For multi-round

[1] An interactive proof is said to be public-coin if the verifier publishes all its randomness during a protocol execution.

interactive proofs, this is a very unfortunate situation when it comes to choosing concrete security parameters. If one wants to rely on the proven security reduction, one needs to choose a large security parameter for the interactive proof, in order to compensate for the exponential security loss, affecting its efficiency. Alternatively, one has to give up on proven security and simply assume that the security loss is much milder than what the general bound suggests – indeed, for many interactive proofs, the known attacks do not feature such a large security loss. The latter, of simply assuming the loss to be milder, has become common practice. In this dissertation, we show that for special-sound interactive proofs the security loss is *independent* of the number of rounds. One can now rely on proven security without choosing overly conservative, and hence inefficient, protocol parameters.

Finally, as an application of compressed Σ-protocol theory, we construct a novel k-out-of-N threshold signature scheme (TSS). The TSS is succinct since a threshold signature has size sublinear in k and n, and in contrast to other succinct TSSs, our TSS does not require a trusted setup and is therefore transparent. We believe that, by the modular nature of compressed Σ-protocol theory, many more application scenarios can be handled in an intuitive and efficient manner.

SAMENVATTING

Samenvatting

Het vakgebied van de *(probabilistische) bewijssystemen* heeft zich ontwikkeld tot een bloeiend deelgebied binnen de cryptologie en informatica. In analogie met wiskundige bewijzen, is het doel van een bewijssysteem dat een bewijzer een verificateur kan overtuigen van de juistheid van een bewering. Probabilistische bewijzen laten daarentegen toe dat de verificateur fouten maakt, dat wil zeggen onjuiste beweringen accepteert (*degelijkheidsfout*) of correcte beweringen verwerpt (*volledigheidsfout*). In veel gevallen kan de foutkans door herhaling verwaarloosbaar klein gemaakt worden zonder veel aan efficiëntie in te leveren. Dit is voor de meeste praktische toepassingen voldoende. Verder kunnen probabilistische bewijzen meerdere interactierondes tussen de bewijzer en de verificateur hebben. In dit geval worden probabilistische bewijzen ook wel *interactieve bewijzen* genoemd. Deze veralgemening, geïntroduceerd door Babai, Goldwasser, Micali en Rackoff [Bab85; GMR85], zorgde voor een revolutie in de bewijstheorie. Door absolute zekerheid in te ruilen voor hoge waarschijnlijkheid en interactie toe te staan, is het bijvoorbeeld mogelijk beweringen te bewijzen zonder meer te onthullen dan hun juistheid. Deze eigenschap wordt *nul-kennis* (zero-knowledge) genoemd. Tegenwoordig worden nul-kennis bewijzen op grote schaal ingezet; ze zijn bijvoorbeeld essentieel in de publieke sleutel infrastructuren die digitale identiteiten en beveiligde communicatiekanalen op het internet beheren.

In het bijzonder biedt de theorie van de Σ-protocollen [Cra96] nu een sterke basis voor het modulair ontwerpen van nul-kennis bewijssystemen in een breed scala aan toepassingsdomeinen. Een Σ-protocol is een interactief bewijs met drie rondes; de bewijzer stuurt eerst een bericht naar de verificateur, die antwoordt met een *challenge* die uniform willekeurig is gekozen uit een eindige verzameling, en na ontvangst van een antwoord van de bewijzer beslist de verificateur om de bewering van de bewijzer te accepteren of af te wijzen. De theorie van de Σ-protocollen onderscheidt zich door haar *modulariteit*; elementaire Σ-protocollen zijn elegant en gemakkelijk te analyseren, en complexe toepassingsscenario's worden afgehandeld door deze basisbouwstenen op de juiste manier te combineren. Op deze manier kan bijvoorbeeld de *vervulbaarheid* van een aritmetisch circuit $C\colon \mathbb{Z}_q^n \to \mathbb{Z}_q$ bewezen worden [CD98], waar $\mathbb{Z}_q$ de ring van gehele getallen modulo q is. Preciezer gezegd, met behulp van de juiste Σ-protocollen kan een bewijzer laten zien dat C een input $\mathbf{x} \in \mathbb{Z}_q^n$ heeft waarvoor geldt dat $C(\mathbf{x}) = 0$. Sterker nog, Σ-protocollen bieden een krachtigere functionaliteit; ze stellen bewijzers in staat om niet alleen te bewijzen dat een circuit vervulbaar is, maar ook dat ze een bijbehorende oplossing $\mathbf{x} \in \mathbb{Z}_q^n$ kennen. Deze eigenschap wordt *kennisdegelijkheid* (knowledge soundness) genoemd, en interactieve bewijzen met deze eigenschap worden ook wel *bewijzen van kennis* genoemd. Het circuit-vervulbaarheidsprobleem is NP-compleet, wat betekent dat elk probleem waarvoor oplossingen efficiënt verifieerbaar zijn, kan

worden geschreven als een circuit-vervulbaarheidsprobleem. Daarom kan door middel van een Σ-protocol elke efficiënt verifieerbare bewering in nul-kennis bewezen worden. Vanwege de modulariteit van de Σ-protocoltheorie zijn er voor veel toepassingsscenario's echter directere en efficiëntere oplossingen die de vaak omslachtige reductie tot een circuit-vervulbaarheidsprobleem vermijden.

Probabilistische bewijzen hebben verschillende prestatiemetrieken, die bijvoorbeeld de (rekenkundige) complexiteit aangeven van het genereren of verifiëren van een bewijs. De communicatiekosten vormen een andere belangrijke metriek; het aantal bits dat wordt gecommuniceerd tussen de bewijzer en de verificateur. Helaas groeien voor veel toepassingsscenario's de communicatiekosten van standaard Σ-protocollen *lineair* met de omvang van de probleeminstantie. Zo is de communicatiecomplexiteit van een Σ-protocol voor het circuit-vervulbaarheidsprobleem lineair in de grootte van het aritmetische circuit. Meer recentelijk is een vouwtechniek geïntroduceerd om de communicatiecomplexiteit te verminderen van lineair naar logaritmisch in de grootte van de probleeminstantie [BCC+16; BBB+18]. De resulterende protocollen worden Bulletproofs genoemd. Bulletproofs werden geïntroduceerd als een vervanging voor Σ-protocollen in verschillende toepassingen, zoals nul-kennis bewijzen voor circuit-vervulbaarheid.

In dit proefschrift verzoenen we de vouwtechniek van Bulletproofs met de gevestigde Σ-protocoltheorie. We laten zien dat de vouwtechniek kan worden gezien als een significante *versterking*, in plaats van een vervanging, van Σ-protocollen. Ons uitgangspunt is een elementair Σ-protocol voor het bewijzen van kennis van een *origineel* van een publiek element $P \in \mathbb{H}$ in het codomein van een groepshomomorfisme $\Psi\colon \mathbb{G}^n \to \mathbb{H}$. Nauwkeuriger gezegd stelt dit Σ-protocol een bewijzer in staat om kennis van een geheime inputvector $\mathbf{x} \in \mathbb{G}^n$ te bewijzen, waarvoor geldt dat $\Psi(\mathbf{x}) = P$ voor een publieke $P \in \mathbb{H}$. De communicatiekosten van dit Σ-protocol groeien lineair in $n \in \mathbb{N}$. Vervolgens laten we zien dat de communicatiecomplexiteit, door een aanpassing van de vouwtechniek van Bulletproofs, kan worden gereduceerd tot logaritmisch in n (of polylogaritmisch, afhankelijk van de concrete instantiëring). Vergelijkbaar met Bulletproofs gaat deze verbetering ten koste van een logaritmisch, in plaats van een constant, aantal rondes. Omdat dit compressiemechanisme hier wordt beschouwd als een uitbreiding van een elementair Σ-protocol, kunnen veel technieken bekend uit de Σ-protocoltheorie direct worden overgenomen door deze nieuwe *theorie van de gecomprimeerde Σ-protocollen*.

Verder breiden we de theorie van de gecomprimeerde Σ-protocollen uit met twee aanvullende functionaliteiten. Ten eerste, door middel van een techniek gebaseerd op aritmetische *secret-sharing*, laten we zien hoe de juistheid van m vermenigvuldigingsdrietallen (multiplication triples) $(\alpha_i, \beta_i, \gamma_i = \alpha_i \cdot \beta_i) \in \mathbb{Z}_q^3$ kan worden bewezen $(1 \leq i \leq m)$. Nauwkeuriger gezegd wordt het bewijzen van de juistheid van vermenigvuldigingsdrietallen gereduceerd tot het bewijzen van kennis van een origineel van een homomorfisme. In andere woorden, de niet-lineaire vermenigvuldigingsdrietal-relatie wordt gelineariseerd. Deze aanpak is bekend uit de Σ-protocoltheorie [CDM00; CDP12] en is geïnspireerd door secure multiparty computation [CDN15]. Er zijn echter enkele aanpassingen nodig om deze aanpak geschikt te maken voor compressie. Door een gepaste en efficiënte reductie laten we zien dat deze functionaliteitsverbetering voldoende is om de vervulbaarheid van aritmetische circuits in (poly)logaritmische communicatie te bewij-

zen. Als tweede functionaliteitsverbetering construeren we een nieuw k-uit-n bewijs van partiële kennis, waarmee kennis van k-uit-n originelen van een homomorfisme bewezen kan worden zonder te onthullen welke originelen de bewijzer kent. Bewijzen van partiële kennis, met name 1-uit-n, hebben de afgelopen decennia talloze toepassingen gevonden, bijvoorbeeld in elektronisch stemmen, digitale (ring)handtekeningen en vertrouwelijke transactiesystemen. Onze constructie laat zien hoe de communicatiecomplexiteit kan worden teruggebracht van lineair naar (poly)logaritmisch in k en n. We vermijden het gebruik van generieke reducties naar circuit-vervulbaarheid en identificeren praktische toepassingsscenario's waarbij onze aanpak asymptotische en concrete prestatieverbeteringen oplevert.

De theorie van de gecomprimeerde Σ-protocollen wordt gepresenteerd in een eenvoudige en abstracte taal, waardoor instantiëringen in diverse cryptografische platforms mogelijk zijn. In het bijzonder laten we zien hoe gecomprimeerde Σ-protocollen geïnstantieerd kunnen worden op basis van de discrete logaritme aanname, resulterend in een logaritmische communicatiecomplexiteit. Vervolgens laten we zien hoe deze instantiëring kan worden uitgebreid naar platforms gebaseerd op bilineaire *pairings*. Op basis van de *kennis van de exponent* (knowledge of exponent) aanname kan de communicatiecomplexiteit verder worden teruggebracht naar een constante hoeveelheid. Ten slotte presenteren we strong-RSA en roostergebaseerde instantiëringen, waarbij het aannemelijk is dat de laatste aanname post-quantum veiligheid biedt. Strong-RSA en op roosters gebaseerde instantiëringen zijn onderhevig aan een zogenaamde *degelijkheidsmarge* (soundness slack). Omgaan met een degelijkheidsmarge vereist grotere protocolparameters en zorgt ervoor dat de resulterende communicatiecomplexiteit polylogaritmisch is in plaats van logaritmisch of constant.

Verder identificeren en dichten we drie hiaten in de algemene theorie van interactieve bewijzen met meerdere rondes. Deze resultaten zijn in het bijzonder relevant voor Bulletproofs en gecomprimeerde Σ-protocollen. Het is over het algemeen namelijk niet triviaal om aan te tonen dat een interactief bewijs *kennisdegelijk*, en dus een bewijs van kennis, is en om een goede bovengrens te vinden voor de *kennisfout*, die de kans op succes van een oneerlijke bewijzer aangeeft. Daarom werd in de context van Σ-protocollen de meer handteerbare notie *speciale-degelijkheid* (special-soundness) geïntroduceerd [Cra96]. Het is bekend dat speciale-degelijkheid, of nauwkeuriger gezegd 2-uit-N speciale-degelijkheid, kennisdegelijkheid met kennisfout $1/N$ impliceert, waarbij N de grootte van de challenge-verzameling van de verificateur is. Algemener impliceert k-uit-N speciale-degelijkheid kennisdegelijkheid met kennisfout $(k-1)/N$. Bulletproofs en gecomprimeerde Σ-protocollen hebben natuurlijke generalisaties van speciale-degelijkheid, voor interactieve bewijzen met meerdere rondes, relevant gemaakt.

Het eerste open probleem dat we aanpakken, is het ontbreken van een kennisdegelijkheidsanalyse voor speciaal-degelijke interactieve bewijzen met meerdere rondes. Als de gevonden bovengrens van de kennisfout niet minimaal is, moeten er conservatieve protocolparameters gebruikt worden. Dit maakt concrete instantiëringen onnodig inefficiënt. Wij bieden de eerste analyse voor de brede klasse van speciaal-degelijke interactieve bewijzen met meerdere rondes die resulteert in een minimale bovengrens voor de kennisfout.

Het tweede open probleem onderzoekt het effect van parallelle herhaling op de

kennisfout. In veel gevallen is de kennisfout κ niet klein genoeg en moet deze dus worden verkleind. Dit kan worden gedaan door het interactieve bewijs parallel te herhalen. Het effect van parallelle herhaling op 2-uit-N speciaal-degelijke Σ-protocollen is bekend, maar de situatie wordt aanzienlijk ingewikkelder als we kijken naar k-uit-N speciale-degelijkheid voor $k > 2$. De situatie wordt al helemaal complex wanneer we de generalisaties van speciale-degelijkheid voor interactieve bewijzen met meerdere rondes beschouwen. Het is namelijk gemakkelijk in te zien dat de t-voudige parallelle herhaling van een 2-uit-N speciaal-degelijk interactief bewijs 2-uit-N^t speciaal-degelijk is. Deze parallelle herhaling heeft dus kennisfout $1/N^t$. Een soortgelijk resultaat geldt niet voor de generalisaties van speciale-degelijkheid. We lossen dit probleem op door te bewijzen dat, voor alle interactieve bewijzen die deze generaliseerde speciale-degelijkheid eigenschap bezitten, t-voudige parallelle herhaling de kennisfout optimaal reduceert van κ tot κ^t.

Ten derde analyseren we de Fiat-Shamir transformatie van speciaal-degelijke interactieve bewijzen met meerdere rondes. De Fiat-Shamir transformatie is een veelgebruikte heuristiek die een public-coin[1] interactief bewijs niet-interactief maakt door de berichten van de verificateur te vervangen door bepaalde hashfunctie-evaluaties. Helaas gaat de Fiat-Shamir transformatie gepaard met een gereduceerde veiligheid van het protocol. Dit verlies kan zelfs *exponentieel* in het aantal rondes van het interactieve bewijs zijn, wat een negatief effect heeft op het kiezen van concrete protocolparameters. Als men wil vertrouwen op bewezen veiligheid, moet men grote parameters kiezen voor het interactieve bewijs om het exponentiële verlies te compenseren. Dit beïnvloedt de efficiëntie op een negatieve manier. Als alternatief kan de bewezen veiligheid opgegeven worden en simpelweg aangenomen worden dat het verlies in veiligheid veel milder is dan wat de algemene (exponentiële) grens suggereert. Het is inderdaad zo dat voor veel interactieve bewijzen de bekende aanvallen geen exponentieel verlies vertonen. Aannemen dat het verlies milder is, is een gangbare praktijk geworden. In dit proefschrift laten we zien dat voor interactieve bewijzen met speciale-degelijkheid het veiligheidsverlies *onafhankelijk* is van het aantal rondes. Men kan nu vertrouwen op bewezen veiligheid zonder al te conservatieve en dus inefficiënte protocolparameters te kiezen.

Ten slotte construeren we, als toepassing van gecomprimeerde Σ-protocollen, een nieuw k-uit-N *Threshold Signature Scheme* (TSS). De TSS is compact omdat een threshold signature een grootte heeft die sublineair is in k en n. Verder vereist onze TSS, in tegenstelling tot andere compacte TSS'en, geen vertrouwde partij om de publieke protocolparameters te genereren. Een TSS met deze eigenschap wordt transparant genoemd. Door de modulaire aard van de theorie van de gecomprimeerde Σ-protocollen verwachten wij dat veel meer toepassingsscenario's op een intuïtieve en efficiënte manier benaderd kunnen worden.

[1] Een interactief bewijs wordt *public-coin* genoemd als de verificateur al zijn willekeur (randomness) publiek maakt gedurende een protocol executie.

ACKNOWLEDGMENTS

Acknowledgments

With writing these acknowledgments I realize how many have contributed, directly or indirectly, to this dissertation. I am finishing a project that would not have been possible without the support of family, friends and colleagues. Therefore, a sincere thanks to everyone. Also, or perhaps especially, to the ones I do not address explicitly below.

First, I would like to thank Ronald Cramer, my promotor and daily supervisor. In our collaboration, he taught me to create my own "map of the cryptographic landscape" and aim to position novel ideas and techniques within this map. We have had many interesting discussions resulting in new research directions, not rarely in one of Amsterdam's pubs enjoying a "Kopstootje." Ronald, thanks for everything!

Also to Thijs Veugen I owe my gratitude. Thijs has helped me arrange this part-time PhD construction with CWI and TNO. He helped me acquire support from both organizations and find a supervisor. As such Thijs has been involved from the very start. I could always rely on his incredibly careful reviews, allowing me to apply the much needed finishing touches. Thijs, I hope our collaborations to continue far beyond this PhD.

Further, I would like to thank Serge Fehr. Working with Serge has been extremely educational. His ability to spot even the most subtle mistakes and ambiguities forced me to be very careful and precise. Often enough he sent me back to the drawing board, after I had incorrectly convinced myself that a problem had been solved.

Moreover, I had the pleasure to collaborate with some fantastic co-authors. Especially, I would like to thank Lisa Kohl, Michael Klooß and Matthieu Rambaud, who directly contributed to the results presented in this dissertation. I very much enjoyed the discussions that carried us away.

Next, this PhD project would not have been possible without the support from my employer TNO. In particular, I am extremely grateful for the many research managers that have helped me along the way: Christophe Hoegaerts, Paul de Jager, Daniëlle Keus, Milena Kooij-Janic, Annemieke Kips, Adri Krabbendam and Dick van Smirren. Annemieke put a lot of effort into the contract negotiations prior to the start of this PhD. During the PhD, Daniëlle gave me the freedom to work on topics with yet to be proven practical relevance. And Dick, even after leaving his position as my manager, was always available for a cup of coffee, ready to reflect on my ambitions and personal development.

I would also like to thank my other TNO colleagues. Vincent Dunning, Maran van Heesch, Michiel Marcus, Niels Neumann, Frank Phillipson, Alex Sangers, Ward van der Schoot, Gabriele Spini, Carolien van der Vliet-Hameeteman, Daniël Worm and many others kept inspiring me with new research directions motivated

by all sorts of practical applications. They acquired and led new projects, allowing me to focus on the content. Altogether they have made this PhD a joyful experience.

Moreover, the Cryptology Group of CWI has given me a very warm welcome. Their deep understanding of cryptology was occasionally intimidating, but the doors were always open, for answering my questions and sharing their knowledge.

Further, I could not have finished this dissertation without the everlasting patience of my wife Fieke. All those evenings that I was occupied trying to solve open problems, she was there to support me in the best way imaginable.

Finally, I would like to thank my family and friends. Throughout the years, my parents Jelle and José have supported and encouraged me to pursue my dreams, whatever these might be. My sister Maud has always had my back; she seems to be available day-and-night to help me with anything I need. Also my friends, Stefan, Mark, Norbert, Rick, Jordy and Leon, have been incredibly supportive. I am urged to write that they appreciated all my monologues about cryptology, but often enough they would kindly ask me to change the topic. Their friendship offered me indispensable distractions allowing me to completely recharge whenever I needed to.

Thanks for everything!

ABOUT THE AUTHOR

About the Author

Thomas Attema was born in Amersfoort, the Netherlands, on July 27, 1990. In 2008, he completed his secondary education at Het Nieuwe Eemland College in Amersfoort. He then continued to study Mathematical Sciences at Utrecht University, receiving a bachelor's degree in 2011 and a master's degree in 2013. Under the supervision of professor Frits Beukers, he wrote the master's thesis titled Super Congruences.

Subsequently, in 2013, Thomas started as a researcher at the Netherlands Organisation for Applied Scientific Research (TNO), where he applied mathematical techniques to solve network related problems in a variety of application domains. In 2016, his research interests started to shift towards (applied) cryptography. In 2018, Thomas obtained a part-time PhD position in the Cryptology Group of Centrum Wiskunde & Informatica (CWI), under the supervision of professor Ronald Cramer (CWI & Leiden University) and professor Serge Fehr (CWI & Leiden University).

Thomas currently holds a position as a senior researcher in the Applied Cryptography and Quantum Algorithms department of TNO, combined with a part-time deployment as a senior staff member in the Cryptology Group of CWI.

THE MATHEMATICIAN'S PATTERNS, LIKE THE PAINTER'S OR THE POET'S MUST BE BEAUTIFUL; THE IDEAS, LIKE THE COLOURS OR THE WORDS MUST FIT TOGETHER IN A HARMONIOUS WAY. BEAUTY IS THE FIRST TEST: THERE IS NO PERMANENT PLACE IN THIS WORLD FOR UGLY MATHEMATICS.

— **G.H. Hardy,** *A Mathematician's Apology (1940)*