Assumptions, Efficiency and Trust in Non-Interactive Zero-Knowledge Proofs

Arne Tobias Ødegaard

Thesis for the degree of Philosophiae Doctor (PhD)
University of Bergen, Norway
2022

UNIVERSITY OF BERGEN

Assumptions, Efficiency and Trust in Non-Interactive Zero-Knowledge Proofs

Arne Tobias Ødegaard

Thesis for the degree of Philosophiae Doctor (PhD)
at the University of Bergen

Date of defense: 04.11.2022

Year: 2022

Title: Assumptions, Efficiency and Trust in Non-Interactive Zero-Knowledge Proofs

Name: Arne Tobias Ødegaard

Print: Skipnes Kommunikasjon / University of Bergen

Acknowledgements

I am grateful to my main supervisor, Helger Lipmaa, for introducing me to the field of zero-knowledge, and for providing a guiding hand through this process. I am also grateful to my other supervisor, Øyvind Ytrehus, for all his help.

I have been very lucky to have the opportunity to work with many brilliant people throughout these three years, and I am particularly grateful to my co-authors Geoffroy Couteau, Prastudy Fauzi, Roberto Parisella, Janno Siim and Michał Zając for their great ideas and discussions. I am also grateful that Martha Norberg Hovd, Janno Siim and Morten Øygarden took time out of their busy day to read and provide incredibly helpful feedback on parts of this thesis.

For combining a strong work environment and a strong social environment, I am very thankful to all my colleagues at Simula UiB. They have made my time here thoroughly enjoyable, and have made me a more rounded person in the process. They have also shown remarkable patience for my bad jokes, which I am eternally thankful for.

For helping me rediscover the joys of cycling, I particularly wish to thank Albin Severinson, Alessandro Melloni and Yauhen Yakimenka.

I would finally like to thank my parents who have supported me throughout my life, in all my endeavours.

Abstract

We live in a digital world. A significant part of our lives happens online, and we use the internet for incredibly many different purposes and we rely on increasingly advanced technology. It therefore is important to protect against malicious actors who may try to exploit this reliance for their own gain.

Cryptography is a key part of the answer to protecting internet users. Historically, cryptography has mainly been focused on maintaining the confidentiality of communication, ensuring that no one can read private messages sent between people. In recent decades, cryptography has become concerned with creating protocols which guarantee privacy even as they support more complex actions.

A crucial cryptographic tool to ensure that these protocols are indeed followed is the zero-knowledge proof. A zero-knowledge proof is a process where two parties, a prover and a verifier, exchange messages to convince the verifier that the prover followed the protocol correctly (if indeed the prover did so) without revealing any private information to the verifier.

It is often desirable to create a non-interactive zero-knowledge proof (NIZK), where the prover only sends one message to the verifier. NIZKs have found a number of different applications, which makes them an attractive object of study. A NIZK has a variety of different properties, and improving any of these aspects advances our collective cryptographic knowledge.

In the first paper in this thesis, we construct a new non-interactive zero-knowledge proof for languages based on algebraic sets. This paper is based on work by Couteau and Hartmann (Crypto 2020), which showed how to convert a particular interactive zero-knowledge proof to a NIZK. We follow their approach, but we start with a different interactive zero-knowledge proof. This leads to an improvement compared to their work in several ways, in particular in terms of both assumptions and efficiency.

In the second paper in this thesis, we study the property of subversion zero-knowledge in non-interactive zero-knowledge proofs. It is impossible to create a NIZK without relying on a common reference string (CRS) generated by a trusted party. However, a NIZK with the subversion zero-knowledge property guarantees that no one learns any private information from the proof even if the CRS was generated dishonestly. In this paper, we create a new cryptographic primitive (verifiably-extractable one-way functions) and show how this primitive relates to NIZKs with subversion zero-knowledge.

Sammendrag

Vi lever i en digital verden. En betydelig del av livene våre skjer på nettet, og vi bruker internett for stadig flere formål og er avhengig av stadig mer avansert teknologi. Det er derfor viktig å beskytte seg mot ondsinnede aktører som kan forsøke å utnytte denne avhengigheten for egen vinning.

Kryptografi er en sentral del av svaret på hvordan man kan beskytte internettbrukere. Historisk sett har kryptografi hovedsakelig vært opptatt av konfidensiell kommunikasjon, altså at ingen kan lese private meldinger sendt mellom to personer. I de siste tiårene har kryptografi blitt mer opptatt av å lage protokoller som garanterer personvern selv om man kan gjennomføre komplekse handlinger.

Et viktig kryptografisk verktøy for å sikre at disse protokollene faktisk følges er kunnskapsløse bevis. Et kunnskapsløst bevis er en prosess hvor to parter, en bevisfører og en attestant, utveksler meldinger for å overbevise attestanten om at bevisføreren fulgte protokollen riktig (hvis dette faktisk er tilfelle) uten å avsløre privat informasjon til attestanten.

For de fleste anvendelser er det ønskelig å lage et ikke-interaktivt kunnskapsløst bevis (IIK-bevis), der bevisføreren kun sender én melding til attestanten. IIK-bevis har en rekke ulike bruksområder, som gjør de til attraktive studieobjekter. Et IIK-bevis har en rekke ulike egenskaper og forbedring av noen av disse fremmer vår kollektive kryptografiske kunnskap.

I den første artikkelen i denne avhandlingen konstruerer vi et nytt ikke-interaktivt kunnskapsløst bevis for språk basert på algebraiske mengder. Denne artikkelen er basert på arbeid av Couteau og Hartmann (Crypto 2020), som viste hvordan man omformer et bestemt interaktivt kunnskapsløst bevis til et IIK-bevis. Vi følger deres tilnærming, men vi bruker et annet interaktivt kunnskapsløst bevis. Dette fører til en forbedring sammenlignet med arbeidet deres på flere områder, spesielt når det gjelder både formodninger og effektivitet.

I den andre artikkelen i denne avhandlingen studerer vi egenskapene til ikke-interaktive kunnskapsløse bevis som er motstandsdyktige mot undergraving. Det er umulig å lage et IIK-bevis uten å stole på en felles referansestreng (FRS) generert av en pålitelig tredjepart. Men det finnes eksempler på IIK-bevis der ingen lærer noe privat informasjon fra beviset selv om den felles referansestrengen ble skapt på en uredelig måte. I denne artikkelen lager vi en ny kryptografisk primitiv (verifiserbart-uttrekkbare enveisfunksjoner) og viser hvordan denne primitiven er relatert til IIK-bevis med den ovennevnte egenskapen.

List of publications

1. Geoffroy Couteau, Helger Lipmaa, Roberto Parisella, and Arne Tobias Ødegaard. "Efficient NIZKs for Algebraic Sets". In: *ASIACRYPT 2021, Part III*. ed. by Mehdi Tibouchi and Huaxiong Wang. Vol. 13092. LNCS. Springer, Heidelberg, Dec. 2021, pp. 128–158. DOI: 10.1007/978-3-030-92078-4_5

2. Prastudy Fauzi, Helger Lipmaa, Janno Siim, Michal Zajac, and Arne Tobias Ødegaard. "Verifiably-Extractable OWFs and Their Applications to Subversion Zero-Knowledge". In: *ASIACRYPT 2021, Part IV*. ed. by Mehdi Tibouchi and Huaxiong Wang. Vol. 13093. LNCS. Springer, Heidelberg, Dec. 2021, pp. 618–649. DOI: 10.1007/978-3-030-92068-5_21

Included in this thesis are the extended versions of the published papers which are taken from the *Cryptology ePrint Archive*, with permission from Springer.

Contents

Chapter 1

Introduction

1.1 Cryptography

The Cambridge Dictionary defines cryptography as *the practice of creating and understanding codes that keep information secret*. The field of cryptography concerns, in its modern form, a vast variety of methods for achieving a large number of tasks without revealing secret information to unwanted parties, called adversaries. While the field contains many different subfields with advanced applications, which we will discuss later, historically the most important aspect of cryptography has been *secure communication*, where one party, Alice, wants to communicate privately with a second party, Bob. The goal of secure communication is that an eavesdropper, Eve, should not be able to learn anything about the content of the message Alice sends to Bob even if Eve can see what is being sent by Alice.

The typical approach to achieving secure communication is that Alice starts out with a *plaintext*, the message she wants to send to Bob. She then applies an *encryption algorithm* to scramble the plaintext and render it in an unintelligible form called a *ciphertext*, which she then transmits to Bob. Bob, upon receiving the ciphertext, applies a *decryption algorithm* to unscramble the ciphertext and recover the original plaintext.

Secure communication has been vital at various times throughout history, particularly during times of armed conflict. If a general wants to send orders to their subordinates, it is of utmost importance that the enemy cannot intercept these orders. Thus, advancements in cryptography became crucial war assets. If one side were able to secure their own communication but managed to read the communication of the enemy, this could prove a decisive advantage. Indeed, during World War II, due to work by a large team at Bletchley Park, the Allies were able to read top-secret Axis communication. Their efforts are estimated to have shortened the war by several years and saved millions of lives [Sin99].

Until the 1970s, a plethora of different techniques were used to create the algorithms used for encryption and decryption, but they all followed a similar template. Alice and Bob would first agree on an algorithm and would share a secret key. Using this shared algorithm and key, Alice would compute the ciphertext from her plaintext and send it to Bob, and Bob would recover the plaintext from this ciphertext.

The drawback of this approach is that Alice and Bob need to agree on a key. This

is a surmountable task if Alice and Bob can meet up in person, but if Alice and Bob want to communicate securely over the internet, and they are on opposite sides of the Earth, this suddenly becomes a serious challenge. The advent of public-key cryptography provided the solution to this problem, and in doing so helped revolutionize the field. Diffie and Hellman [DH76], based on earlier work by Merkle [Mer78], defined and created a method for *key exchange*, where two different parties can, by only exchanging messages in public, agree on a shared secret key known only to them. Around the same time, Rivest, Shamir and Adleman [RSA78] created a method for *public-key encryption*, where one party can encrypt the message using a public encryption key, but only the intended recipient can decrypt the ciphertext using their private decryption key. Both of these methods relied on a separation between public and private keys satisfying a specific mathematical relationship. Public-key cryptography was discovered earlier at the British signals intelligence agency GCHQ, but the fact of their discovery was kept secret until several decades later [Sin99].

During the latter half of the twentieth century, alongside the new development of public-key cryptography, cryptography as a whole started moving in a more scientific direction, and away from the various ad-hoc approaches used previously. Provable security became a core part of the field, where security properties were formally defined, and constructions were proven secure. Only certain cryptographic constructions can be proven secure unconditionally, and the remaining constructions rely on unproven, but usually well-motivated, complexity-theoretic assumptions, such as the supposed difficulty of factoring an integer into its prime factors.

As information technology has grown more mature and advanced, the number of possible applications of various kinds of cryptography has increased drastically. These days, cryptographers are interested in a lot more than secure communication. Some of the major topics in vogue at the moment are: *homomorphic encryption*, where one can compute on encrypted data without decrypting it first, which could allow someone to run tests for you based on your private medical data without revealing anything about you; *secure multi-party computation*, where a group of people can compute a function of their inputs while keeping their inputs secret, which could be used to run an auction where the highest bidder wins but the bids of everyone else remain secret; and the topic of this thesis, *zero-knowledge proofs*, where someone can prove properties about secret data without revealing it, which Bob could use to prove that his cryptocurrency transaction is legitimate without revealing the sender, recipient or the amount of money being sent. For more applications of zero-knowledge proofs, see Section 1.3.

To explain what zero-knowledge proofs are, we need to first take a detour into the nature of proofs in mathematics.

1.2 Mathematical proofs

The notion of proof is a big part of what makes mathematics a unique discipline. For a statement to be accepted as true in mathematics, it needs to come with a proof that establishes the logical necessity of the statement.

The recorded history of proofs in mathematics goes back more than 2500 years, to at least Thales of Miletus (624-546 BCE). However, Euclid (ca. 300 BCE) is the person credited with creating the notion of proof that mathematicians are familiar with today. Euclid wrote an incredibly influential textbook, the *Elements*, on the subjects of plane geometry and number theory. The mathematical content in this textbook was significant on its own, but of most impact was the way statements were proved, and how those proofs were presented. Euclid started with a small number of axioms, which he took as pre-established truths. He then used logical derivation rules to derive new truths from these axioms and would continue this process until he had derived the statement he wanted to prove. This *proof system*, where the person creating the proof finds a list of logical deductions which form the proof, and the person checking the proof simply verifies that these logical deductions are indeed correct, remains ubiquitous in mathematics to this very day.

Interactive proofs

The classical proof system has its strengths and weaknesses. It is very straightforward to check if a properly written proof is indeed correct, and if the proof is correct, then the statement must necessarily be true. However, such proofs can become very large, which makes them time-consuming to both write down and verify. A natural question then becomes whether there are other proof systems with more desirable properties. To create a new proof system, a natural place to start is to figure out what the key components of a proof system are, and how some of them can be modified.

Any proof system has two main roles, that of the *prover* and the *verifier*. The prover has discovered the truth of some statement and wants to convince the verifier of this fact. It sends a proof to the verifier, and the verifier checks whether the proof is correct. A proof system needs to satisfy *completeness*, meaning that if the prover knows that a statement is true and why this is the case, the prover can create a proof that the verifier will accept. Additionally, the proof system needs to satisfy *soundness*, meaning that if the verifier accepts the proof, then the statement must be true. Equivalently, it should not be possible to create an acceptable proof of a false statement.

By cleverly modifying some of the properties of the classical proof system, Goldwasser, Micali and Rackoff [GMR85] defined the *interactive proof*. The first defining aspect of an interactive proof is that the prover and verifier can interact with each other. Instead of the prover computing the whole proof on their own, the prover and verifier can send

messages back and forth, and this interaction forms the proof. The second key aspect of an interactive proof is the ability of the verifier to make random decisions, and ask the prover random questions. Along with this reliance on randomness comes a slightly relaxed soundness criterion. The new definition of soundness requires that the probability that a cheating prover convinces the verifier to accept a false statement is sufficiently small, but it can be non-zero, unlike in a classical proof.

A key advantage of interactive proofs is that they allow for the creation of much smaller proofs. Shamir [Sha92] showed that, under a widely believed conjecture in computer science, there are statements that can be proved with a small interactive proof, but where any traditional proof would be so long that the universe would end before it could be verified. Both the interactive nature of the proof, and the random nature of the verifier, are crucial to establishing this advantage over traditional proofs. The invention of interactive proofs has had a massive impact on computer science, and Goldwasser, Micali and Rackoff were awarded the 1993 Gödel Prize for their work, along with Babai and Moran.

1.3 Zero-knowledge proofs

Another benefit of interactive proofs becomes apparent when considering other applications than mathematics. While a proof in mathematics is intended to illuminate why a particular statement is true, there are circumstances where you only want to show that something is true, but not why. Suppose you have created a great sudoku puzzle for your friend to solve, but your friend is not convinced that this puzzle actually has a solution. A perfectly valid proof would be to simply reveal a solution, but this could ruin the enjoyment they would get from solving it. For this reason, you might desire additional properties of a proof beyond just completeness and soundness.

In the same paper where they define interactive proofs, Goldwasser, Micali and Rackoff [GMR85] also define what it means for a proof to be a *zero-knowledge* (ZK) proof. A zero-knowledge proof is a proof where the verifier only learns that the statement is true, but nothing else beyond this fact, particularly no secret information the prover used to create the proof. Returning to our sudoku example, a ZK proof showing that a sudoku puzzle is valid would establish solely this fact, and would not spoil anything about how to solve the puzzle. Ben-Or et al. [Ben+90] showed that every interactive proof can be transformed into a zero-knowledge interactive proof.

Zero-knowledge proofs are where cryptography and mathematical proofs join hands because they are proofs that can be used to protect someone's privacy, as in the following example. An online casino will only let in users who are 18 years or older. If a user wants to gain access to this casino, they could transmit their age to the casino, but if the user wants to keep their age secret, this is a bad idea. The user and website can instead carry

out a zero-knowledge proof to check that the user's age is at least 18. The ZK property guarantees that the casino learns nothing about the age of the user beyond the fact that the user is over 18 years old, and thus the privacy of the user is maintained.

To show the validity of a specific construction of a ZK proof system, one needs to provide a (classical) proof that the construction satisfies completeness, soundness and zero-knowledge. These proofs often rely on a cryptographic assumption, and then they only show that the property is secure against efficient adversaries. For example, a construction might only satisfy *computational soundness*, meaning that an adversary which has unlimited time and resources could find a valid proof of a false statement, but any adversary which can only take some reasonable amount of time would not be able to. Zero-knowledge proofs which only satisfy computational soundness are called zero-knowledge *arguments*.

Non-interactive zero-knowledge proofs

While zero-knowledge proofs were originally formulated in the interactive setting, this turns out to have significant drawbacks for certain applications. Verifying the same statement multiple times becomes complicated as there must be an interaction process every time a statement needs to be checked. Additionally, it places additional demands on the prover, as they need to be continuously available to interact with the verifier and defend their statement. To overcome these drawbacks, it would be convenient to combine the zero-knowledge property with the non-interactive nature of traditional proofs. Sadly, Goldreich and Oren [GO94] showed this to be impossible in its basic form: One cannot hope to achieve zero-knowledge proofs for non-trivial statements if the prover only sends one message to the verifier.

Fortunately, this is not the end of the story. Blum, Feldman and Micali [BFM88] showed that one can construct *non-interactive zero-knowledge proofs* (NIZKs) in the *common reference string* (CRS) model. In this setting, there needs to be a setup phase ahead of time, run by a third party trusted by both the prover and the verifier, which generates some common information shared between the prover and verifier. After this setup has been done, the prover only needs to send one additional message to the verifier to create a proof.

The initial work of Blum, Feldman and Micali was a (non-interactive) proof of concept and had significant drawbacks. Most importantly, their NIZK was very inefficient, in the sense that computing and verifying a proof would take too much time for any real-life applications. An early alternative was the Fiat–Shamir transform, created by Fiat and Shamir [FS87], in which one transforms an interactive ZK proof (with certain specific properties) into a non-interactive ZK proof by letting the prover compute the verifier's messages on their own, in such a way that the prover is not able to cheat. This approach

can lead to very efficient NIZKs and does not rely on a CRS, but the security of such a NIZK is only proven in the random oracle model. This suggests that the NIZK is secure, but provides no concrete proof, as there are protocols proven secure in the random oracle model which are actually insecure [CGH98].

An important historical milestone to mention is the creation by Groth and Sahai [GS08] of Groth–Sahai proofs, provably secure NIZKs based on a very standard and well-known assumption, which do not rely on the random oracle model. Their approach led to proofs which were efficient for a limited set of useful statements. While one could use Groth–Sahai proofs to prove any statement, this would be very slow and the proof would be very big.

zk-SNARKs

While the work of Groth and Sahai presented a significant step forward, there were still further improvements to be made with regard to efficiency. For a number of practical use cases, where thousands of proofs need to be verified every second, and millions of proofs need to be transferred and stored, the size of the proofs, as well as how quickly they can be verified, become crucial, and the previous approaches were simply not sufficient.

It thus became desirable to construct a *succinct non-interactive argument* (SNARG), where succinct means that the size of the proof stays small even as the statement becomes more complex. Kilian [Kil92] used probabilistically checkable proofs to create a four-round argument with succinct communication, and Micali [Mic94] showed how to turn this protocol into a SNARG using the Fiat–Shamir transform to create a non-interactive argument in the random oracle model. Di Crescenzo and Lipmaa [DL08] and Bitansky et. al [Bit+17] followed a similar approach but removed the random oracle model by relying on very strong assumptions. A sequence of works starting with Groth in 2010 [Gro10], continued by Lipmaa [Lip12], Gennaro et al. [Gen+13] along with several others, showed how to construct SNARGs using a mathematical structure called pairings. This line of research culminated in the extremely efficient Groth16 SNARG [Gro16]. Using pairings in this manner one again avoids the random oracle model but has to rely on very strong assumptions.

All the works mentioned since 2010 were in fact *succinct non-interactive arguments of knowledge* (SNARKs). The difference between a SNARG and a SNARK is that the latter provides a stronger notion of soundness known as knowledge-soundness. If a proof is knowledge-sound, this guarantees not only that a prover cannot cheat, but also that any prover which can create a valid proof must know a witness which certifies that the statement is true. For example, using a SNARK the verifier can be convinced that not only is the statement “There exists a password which matches this public account information” true, but the prover must also know such a password. SNARKs that also

satisfy zero-knowledge are referred to as zk-SNARKs, and all previously mentioned SNARKs are of this type.

Applications

Zero-knowledge proofs, and in particular NIZKs and zk-SNARKs, have found a number of applications, both on their own and as a building block for other cryptographic constructions.

The typical application for a zero-knowledge proof concerns a system where users have some private data that would allow them to perform specific actions. Examples of such private data could be their age, their location or their account password. The user can simply reveal their private data, but this is often undesirable. A better solution is to let the user compute a zero-knowledge proof showing that their private data satisfies certain conditions. The soundness property of the proof guarantees that users cannot perform actions they are not allowed to, and the zero-knowledge property guarantees that the user's private data is indeed kept private. Both interactive and non-interactive zero-knowledge proofs could be appropriate, depending on the application.

One such natural application is *anonymous credentials*, introduced by Chaum [Cha85]. Anonymous credentials allow a user to get a credential from some service that contains private data about themselves, such as their age, name, city and ID number. When the user wishes to access a website that restricts access to users based on their attributes, such as an online casino, they can use this credential to compute a zero-knowledge proof which establishes that they satisfy those attributes.

A feature of zero-knowledge proofs which is the basis of a number of applications is that they allow you to prove that you executed a computation correctly. This makes it possible to verify that everyone followed an agreed-upon protocol. An example application where verification comes into play is the area of *verifiable computation*, formalized by Gennaro, Gentry and Parno [GGP10]. Suppose someone wishes to perform a massive computation they cannot do on their own. Verifiable computation would allow them to offload this computation to some cloud service, which would do the computation and return the output. In order to verify that the cloud service did not simply provide some correct-looking answer without actually doing the computation, they would include a zk-SNARK proof that they performed the computation.

NIZKs can provide crucial guarantees in electronic voting systems. A key part of most electronic voting systems is a way to shuffle all the votes to ensure anonymity, a digital analogue of shaking the ballot box, an idea introduced by Chaum [Cha81]. To ensure that the participants behaved honestly, an accompanying NIZK proof is produced to verify that the shuffle was executed correctly, first demonstrated by Sako and Kilian [SK95].

A notable application of zk-SNARKs is found in cryptocurrencies, online currencies

based on a decentralized ledger. In the most famous example, Zcash [Ben+14], zk-SNARKs are used to provide privacy for transactions. A transaction in a cryptocurrency contains a lot of information that needs to be correct, such as which accounts are involved, that the sender has sufficient money in their account, and that the user creating this transaction has access to the sender's account. In Zcash, a user which performs a transaction produces a zk-SNARK proof that everything in their transaction is correct using the secret information about their account. In these last two examples, there are a number of other components that come together to form the whole system, but a zero-knowledge proof is at the heart of both systems to protect users against misbehaving adversaries.

1.4 Assumptions in cryptography

While some parts of cryptography can be proven secure from basic principles, most modern cryptography relies on a vast web of cryptographic assumptions. Assumptions are unproven, but hopefully well-motivated, statements cryptographers use to prove that a construction satisfies some property. Proofs can establish that, as long as the given assumption is true, then the construction does indeed satisfy this property. For example, one could prove that an encryption scheme remains secure as long as the assumption that it is hard to factor integers is true.

Cryptographic assumptions are a necessary underpinning of modern cryptography. However, proving that any of these assumptions are true is a Sisyphean task, as such a proof would typically settle the long-standing problem of P vs. NP, a problem which is beyond our current understanding of mathematics and computation. While we cannot prove that a certain assumption holds, we would like to make the assumptions we rely on as weak as possible because weaker assumptions are more likely to be true. At the very least we want to understand something about which assumptions we rely on, and whether they are necessary to prove the security properties we desire.

One important division of assumptions is into falsifiable and non-falsifiable assumptions. Falsifiable assumptions are assumptions where the fact that they have been broken can be efficiently verified. The integer factoring assumption is an example of a falsifiable assumption because if someone produces two supposed factors of a number, it is easy to multiply them together to see if they indeed are the factors. Because it is hard to verify whether a non-falsifiable assumption has been broken, these assumptions are hard to reason about, and falsifiable assumptions are preferred in general, but it is not always possible to rely on falsifiable assumptions. Related to non-interactive zero-knowledge, Gentry and Wichs [GW11] showed that the soundness of a zk-SNARK can never be based on falsifiable assumptions using most normal proof techniques. However, some less efficient NIZKs can be shown both sound and zero-knowledge based on falsifiable

assumptions, so this represents a trade-off between efficiency and relying on desirable assumptions.

1.5 Research on NIZKs

The field of non-interactive zero-knowledge, despite having seen a large flurry of activity in the past decades, is still a field where a number of open questions remain, and new results are consistently being found. One can make progress by creating a NIZK with improved efficiency, be it the running time of the prover or verifier, or the size of the proof or CRS. Additionally, one can create NIZKs whose security is based on weaker assumptions than previously existing constructions. Since there is a trade-off between assumptions and efficiency, such as any succinct NIZK requiring non-falsifiable assumptions [GW11], another way to make progress is to present a NIZK with a different such trade-off. One can also construct NIZKs that satisfy some additional properties, which might be needed for specific applications. And finally, while it is necessary to have some trust in the third party generating the CRS because there are no NIZKs without a CRS [GO94], it is desirable to reduce the trust required of the third party generating the CRS.

1.6 The results in this thesis

This thesis consists of two papers, both on the topic of non-interactive zero-knowledge.

Paper I

The first paper, *Efficient NIZKs for Algebraic Sets* [Cou+21b], was published at Asiacrypt 2021. It was written with co-authors Geoffroy Couteau, Helger Lipmaa and Roberto Parisella. The version included in this thesis is a full version uploaded to the *Cryptography ePrint archive* [Cou+21a], which fixes some minor errors and adds lengthy appendices which were not included in the published version due to page limits. The main contribution of this paper is a construction of a NIZK for proving statements about roots of polynomials.

The starting point of our construction is a work by Couteau and Hartmann [CH20], which provides a technique by which one can transform a specific interactive zero-knowledge proof into a NIZK. Unlike the Fiat–Shamir transform, this technique ensures that the security of the resulting NIZK can be proven without relying on the random oracle model. We start by constructing a new interactive zero-knowledge proof, which is then transformed into a NIZK using their technique. We define a novel way to represent a polynomial using a matrix, which forms the basis of our zero-knowledge proof. The security of our NIZK relies on a weaker version of the assumption used by Couteau and

Hartmann, and we additionally provide evidence for why the assumption we rely on is reasonable.

Paper II

The second paper, *Verifiably-Extractable OWFs and Their Applications to Subversion Zero-Knowledge* [Fau+21b], was published at Asiacrypt 2021. It was written with co-authors Prastudy Fauzi, Helger Lipmaa, Janno Siim and Michał Zając. The version included in this thesis is a full version uploaded to the *Cryptography ePrint archive* [Fau+21a], which adds some material and fixes some minor errors. The paper studies the assumptions required to achieve zero-knowledge in a NIZK without trusting the CRS creator, and its main contribution is the definition of a new cryptographic primitive which relates to this.

In the CRS model, it is crucial that both the prover and the verifier trust the third party which computes the common reference string because a malicious creator of the CRS could break either the soundness or zero-knowledge properties. It is impossible for neither party to trust the CRS creator because then the prover could simply create the CRS on its own, and this would give a non-interactive zero-knowledge proof without a CRS, which is known to be impossible. However, there are constructions where the verifier still needs to trust the CRS creator to achieve soundness, but the prover can be convinced that zero-knowledge is achieved regardless of what the CRS creator does. This is called subversion zero-knowledge, meaning that the NIZK has the zero-knowledge property even if the CRS is subverted.

In this paper, we study which assumptions are needed to achieve subversion zero-knowledge NIZKs. We define a new primitive called a verifiably-extractable one-way function (VEOWF). A VEOWF is a one-way function where, while it is difficult to compute a preimage given just an image of the function, it should be possible to extract a preimage from any machine outputting an image of the function. Additionally, it must be possible to efficiently verify if some value belongs to the image of the function. We show how a VEOWF can be used to add the subversion zero-knowledge property to certain existing NIZKs. Additionally, we show that any subversion zero-knowledge NIZK fulfilling some additional criteria can be used to construct a VEOWF, hence showing a strong connection between VEOWFs and subversion zero-knowledge NIZKs.

Chapter 2

Preliminaries

This chapter contains key background material, starting with some basic concepts from mathematics and cryptography. It then gives a flavour of assumptions used in cryptography, and how one can classify them. Finally, we define both interactive and non-interactive zero-knowledge proofs and give some example constructions.

2.1 Mathematics

Groups and pairings

A lot of modern cryptography is based on mathematical groups, whose structure provides a good balance between flexibility and security. A large class of groups which find their use in cryptography, and are of special interest when it comes to this thesis, are cyclic groups $\mathbb{G}$ whose order is some large prime p. We simply rely on these properties (as well as certain efficiency and hardness properties) in a black-box way, but it is worth pointing out that the recommended way to implement these groups is by using groups which are elliptic curves over finite fields. For an introduction to elliptic curves in cryptography, see [BSS00].

We consider cyclic groups which come endowed with a group element $\mathfrak{g}$ which generates the whole group $\mathbb{G}$, i.e. it is a generator of $\mathbb{G}$. We will write our groups additively, and we introduce the bracket notation of [Esc+13]. We fix a generator $\mathfrak{g}$, and write $[a]$ to mean $a\mathfrak{g}$, where a is an integer modulo p. (We let $\mathbb{Z}_p$ denote the set of integers modulo p) This notation is linear, i.e. $[a] + [b] = [a + b]$ and $n[a] = [na]$ for all $a, b, n \in \mathbb{Z}_p$. We can also extend the bracket notation to vectors and matrices, by applying the bracket pointwise. For a vector $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{Z}_p^n$, $[\boldsymbol{v}] = ([v_1], \ldots, [v_n]) \in \mathbb{G}^n$, and we will often write $[v_1, \ldots, v_n]$ to mean $[(v_1, \ldots, v_n)]$. Similarly $[\boldsymbol{A}] \in \mathbb{G}^{m \times n}$ for a matrix $\boldsymbol{A} \in \mathbb{Z}_p^{m \times n}$.

For certain applications, like the ones we consider in this thesis, one requires groups with additional structure, namely that of a bilinear pairing. One starts with three groups $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, all of the same order p, and where $\mathfrak{g}_1$ and $\mathfrak{g}_2$ generate $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. A pairing is a function $\hat{e} \colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ which satisfies these three properties:

Bilinearity: $\forall a, b \in \mathbb{Z}_p \colon \hat{e}(a\mathfrak{g}_1, b\mathfrak{g}_2) = ab \cdot \hat{e}(\mathfrak{g}_1, \mathfrak{g}_2)$,

Non-degeneracy: The element $\mathfrak{g}_T = \hat{e}(\mathfrak{g}_1, \mathfrak{g}_2)$ is a generator of $\mathbb{G}_T$,

Efficiency: There is an efficient algorithm to compute $\hat{e}$.

The benefits of the bracket notation become clear in the presence of pairings. We write $[a]_\iota = a\mathfrak{g}_\iota$ for $\iota \in \{1, 2, T\}$. We define $[a]_1 \bullet [b]_2 = \hat{e}([a]_1, [b]_2)$, and the bilinearity requirement then reads that $[a]_1 \bullet [b]_2 = [ab]_T$. We can extend this to vectors and matrices as well, for example for two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of appropriate dimensions, it is the case that $[\boldsymbol{A}]_1 \bullet [\boldsymbol{B}]_2 = [\boldsymbol{AB}]_T$. For pairings to be applicable in cryptography, they must satisfy certain cryptographic hardness assumptions. While there exist many bilinear maps, the only ones known to be suitable for cryptographical applications are in the setting of elliptic curves.

While groups on their own provide for limitless addition and scalar multiplication operations of group elements, pairings allow for one multiplication of group elements, which makes them very flexible for the design of cryptographic protocols. The use of pairings for this purpose was pioneered by Joux [Jou00], and developed further by several others, such as Boneh and Franklin [BF01] and Boneh, Lynn and Shacham [BLS04], and is now a standard tool in the cryptographic toolbox. For an introduction to how pairings can be used in cryptography, see [Men09].

One can classify pairings into three main types depending on the relationship between $\mathbb{G}_1$ and $\mathbb{G}_2$, as done by Galbraith, Paterson and Smart [GPS06].

- If $\mathbb{G}_1 = \mathbb{G}_2$, this is called a type-1 pairing.
- If $\mathbb{G}_1 \neq \mathbb{G}_2$ and there is an efficiently computable isomorphism between the groups, this is called a type-2 pairing.
- If $\mathbb{G}_1 \neq \mathbb{G}_2$ and there is *no* efficiently computable isomorphism between the groups, this is called a type-3 pairing.

Type-1 pairings are also called *symmetric* pairings, and type-2 and type-3 are both referred to as *asymmetric* pairings.

Complexity theory

Languages and complexity classes

In computer science, everything is typically encoded as finite strings of bits, i.e. members of the set $\{0,1\}^*$. If we are discussing the set of prime numbers, for example, we implicitly choose an of the natural numbers (e.g. their binary expansion), and the set which is really under discussion is the set of encodings. Of fundamental interest in computer science are *languages*, where a language $\mathcal{L}$ is simply a subset of $\{0,1\}^*$.

One can classify languages into *complexity classes* based on how they can be recognised, typically by which Turing machines can recognise them. A Turing machine is a mathematical model of computation, essentially describing an abstract form of a computer. For an introduction to the subject, as well as a formal definition, see [Sip13]. Some of the fundamental complexity classes in theoretical computer science and cryptography are:

P: P consists of any language $\mathcal{L}$ where deciding whether $\mathtt{x} \in \mathcal{L}$ can be achieved by a deterministic Turing machine which runs in time polynomial in the size of $\mathtt{x}$.

BPP: BPP consists of any language $\mathcal{L}$ where deciding whether $\mathtt{x} \in \mathcal{L}$ can be achieved by a probabilistic Turing machine which runs in time polynomial in the size of $\mathtt{x}$, and which makes mistakes at most $1/3$ of the time.

NP: NP consists of any language $\mathcal{L}$ where all $\mathtt{x} \in \mathcal{L}$ have a witness $\mathtt{w}$ certifying this fact, and it can be checked in time polynomial in the size of $\mathtt{x}$ whether the certificate is valid. This can be re-formulated as: $\mathcal{L}$ is in NP if there exists a relation $\mathcal{R} = \{(\mathtt{x}, \mathtt{w})\}$ where membership can be decided in polynomial time, and $\mathcal{L} = \{\mathtt{x} \mid \exists \mathtt{w} : (\mathtt{x}, \mathtt{w}) \in \mathcal{R}\}$.

PSPACE: PSPACE consists of any language $\mathcal{L}$ where deciding whether $\mathtt{x} \in \mathcal{L}$ can be achieved by a deterministic Turing machine which requires an amount of space which is polynomial in the size of $\mathtt{x}$.

It is quite easily shown that $\mathsf{P} \subseteq \mathsf{BPP} \subseteq \mathsf{NP} \subseteq \mathsf{PSPACE}$, but it is not known whether any of these inclusions are strict. In fact, the matter of deciding whether $\mathsf{P} = \mathsf{NP}$ is one of the Millenium Prize problems, and solving it comes with a 1 million dollar prize.

Reductions

The fundamental tool of the cryptographer working with provable security is the *reduction*. One wants to prove that a security property P of a certain construction holds as long as some assumption A holds as well, that is one wants to reduce property P to assumption A. The traditional approach is to prove the contrapositive, namely one starts out by assuming that the construction does not have property P. This means that there is some adversary $\mathcal{A}$ which is able to break P. The security reduction then uses this adversary $\mathcal{A}$ to build a new adversary $\mathcal{B}$ which breaks assumption A. One has therefore established that if the construction does not have property P, then assumption A does not hold. It must then be the case that if assumption A holds, no adversary breaks it, and therefore there does not exist an adversary which breaks P, and therefore the construction must indeed have property P.

2.2 Cryptography

Fundamental notions

A key definitional question of cryptography is what it means for a cryptographic scheme to be *secure*. A cryptographic scheme typically consists of some algorithms, which together must satisfy certain security properties, all of which must hold for the scheme to be secure. While it is desirable that a security property can never be broken by any adversary, this is often impossible. The approach usually taken is therefore to define that a security property holds if no reasonable adversary can succeed with more than negligible probability, which leaves open the question of defining the allowable adversaries and probabilities.

We follow the standard approach where we parametrise our schemes by a security parameter λ which can be chosen upon setup to reach the desired security level. Intuitively, the security parameter should represent the number of bits of security one achieves, that means by choosing the security parameter λ, it should take 2^λ time to break the scheme. For technical reasons, algorithms are given the security parameter in the form 1^λ, i.e. λ written in unary. We then define "reasonable adversary" as any probabilistic (i.e. one which can make random decisions) adversary which takes an amount of time which is bounded by $p(\lambda)$, where p is a polynomial. We call such adversaries PPT (probabilistic polynomial-time) adversaries. The probability of success being negligible means that this probability, as a function of λ, goes to 0 faster than the inverse of any (positive-valued) polynomial. Formally, a function $f\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ is said to be negligible if, for all polynomials $p\colon \mathbb{N} \to \mathbb{R}_{>0}$, there exists N such that, if $n \geq N$, then $f(n) < \frac{1}{p(n)}$. We will typically define that a security property holds if all PPT adversaries have a negligible probability of succeeding in breaking the property.

The above definition does not provide, on its own, any concrete security guarantees. Even if a scheme is proven secure given this definition, it could be unclear how large λ needs to be to achieve security against actual adversaries. Providing concrete security bounds that state how much power an adversary needs to break the scheme for specific parameters requires further analysis which is beyond the scope of this thesis.

Group generators

Since our schemes depend on a security parameter, we require algorithms which generate groups and bilinear pairings for use in our schemes. We will postulate the existence of a group generator GpGen which takes in a security parameter 1^λ and outputs a description $\mathsf{gp} = (\mathbb{G}, p, [1])$ consisting of the group itself, its order, and a generator of the group. We additionally postulate the existence of a bilinear pairing generator BGGen which takes in a security parameter 1^λ, and outputs a description $\mathsf{bp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, [1]_1, [1]_2, \hat{e})$,

consisting of the three groups, their order, generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, and the pairing itself. In practice, one often uses a fixed security parameter and a fixed group/pairing, and when that is the case one tends to use an elliptic curve such that the value p is a prime which is approximately 2^{256}, resulting in 128-bit security. Choosing an appropriate curve is an important question, but is outside the scope of this thesis.

Notation

We write $\{0,1\}^n$ to denote all bitstrings of length n. We write $x \leftarrow\!\!\$\, S$ to denote that x is sampled uniformly at random from the set S. If $\mathcal{D}$ is a probability distribution, we write $x \leftarrow\!\!\$\, \mathcal{D}$ to denote that we sample according to $\mathcal{D}$. We write $y \leftarrow \mathcal{A}(x)$ to denote that the adversary $\mathcal{A}$ on input x outputs y. Note that if $\mathcal{A}$ is a randomized algorithm, then y has a probability distribution. In the case where $\mathcal{A}$ is a randomized algorithm, we let $\mathsf{RND}_\lambda(\mathcal{A})$ denote the random coins used by $\mathcal{A}$.

Fundamental primitives

Our work relies on several basic building blocks called *cryptographic primitives*. These are algorithms with a limited scope, designed to perform one simple task. We here define some primitives we refer to in the thesis.

One-way functions

A foundational concept of cryptography is the one-way function, a function which is easy to compute, but hard to reverse. Given just an output of the function, it should be hard to find an input which produces that output. A typical example candidate of a one-way function is multiplication of large prime numbers. While it seems hard to find two prime numbers which multiply to 3233, it is easy to verify that $53 \cdot 61 = 3233$.

Definition 2.2.1 (One-way functions [KL14, p. 332])**.** A polynomial-time computable function $f\colon \{0,1\}^* \to \{0,1\}^*$ is a *one-way* function if for any PPT adversary $\mathcal{A}$, this is negligible:

$$\Pr\left[f(x') = f(x) \mid x' \leftarrow \mathcal{A}(f(x)), x \leftarrow\!\!\$\, \{0,1\}^\lambda\right] \quad .$$

Public-key encryption

Public-key encryption is an encryption method with two different keys, one public key for encryption and one private key for decryption. Such a scheme needs to be correct, namely that decrypting an encrypted message with the right keys gets back the initial message. One also requires some form of security, and here there are a number of flavours, which vary based on the goal of the attacker, as well as their powers. We will here state the basic property of IND-CPA security (indistinguishability under chosen-plaintext attacks),

which requires that the attacker is not able to distinguish encryptions of two different messages of its choice, even though it can encrypt any plaintexts.

Definition 2.2.2 (Public-key encryption [KL14, pp. 378–380])**.** A *public-key encryption* scheme PKE is a tuple of three algorithms $(\mathsf{Kgen}, \mathsf{Enc}, \mathsf{Dec})$ where

- Kgen takes a security parameter 1^λ as input and outputs a keypair $(\mathsf{pk}, \mathsf{sk})$,
- Enc takes a public key pk and a message m as input and outputs a ciphertext c,
- Dec takes a private key sk and a ciphertext c as input and outputs a message m.

The two algorithms Kgen and Enc are randomized algorithms, so they additionally get access to random coin flips as input, but we will not make this explicit unless required for clarity. These algorithms together must satisfy the following properties:

Correctness: For any keypair $(\mathsf{pk}, \mathsf{sk})$ generated by Kgen,

$$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{pk}, m)) = m \ .$$

IND-CPA security: For any PPT adversary $\mathcal{A}$, this is negligible:

$$\left| \Pr\left[b' = b \;\middle|\; \begin{array}{c} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Kgen}(1^\lambda), (m_0, m_1) \leftarrow \mathcal{A}(\mathsf{pk}) \\ b \leftarrow\!\!\$\, \{0,1\}, b' \leftarrow \mathcal{A}(\mathsf{Enc}(m_b)) \end{array} \right] - \frac{1}{2} \right| \ .$$

Note that an adversary which outputs b' at random succeeds with probability $\frac{1}{2}$, so this states that no adversary can do noticeably better than guessing at random.

One example of a public-key encryption scheme is the Elgamal encryption scheme for group elements. The key generation algorithm Kgen outputs a group description $\mathsf{gp} = (\mathbb{G}, p, [1])$ plus a group element $[x]$ as the public key, and the *discrete logarithm* x as the secret key. Encryption is defined as $\mathsf{Enc}(\mathsf{pk}, [m]) = ([r], [m] + r[x])$, where r is a random integer. Decryption is defined as $\mathsf{Dec}(\mathsf{sk}, [c_1, c_2]) = [c_2] - x[c_1]$. It is straightforward to verify the correctness of this scheme, and the IND-CPA security of the scheme can be proven based on the decisional Diffie–Hellman assumption, see Section 2.3.

Commitment schemes

A commitment scheme is a cryptographic primitive which allows someone to, essentially, write down a value in a sealed envelope and later reveal its contents. In a commitment scheme, the user can take some message m and some randomness r and create a commitment c. Later on, the user can reveal their message and randomness, to show that this is what they committed to. There are two key security properties such a scheme must satisfy. A commitment scheme must be *hiding*, no adversary should be able to tell anything about the message based on the commitment. It must also be *binding*, the

commitment must tie the user to a specific value, i.e. the user can not open a commitment to two different values.

Definition 2.2.3 (Commitment schemes [KL14, pp. 187–188])**.** A *commitment scheme* C is a tuple of two algorithms $(\mathsf{Kgen}, \mathsf{Com})$ where

- Kgen takes as input a security parameter λ and outputs a commitment key ck;
- Com takes as input a commitment key ck, a message m and some randomness r and outputs a commitment com.

These algorithms must satisfy:

Hiding: For any PPT adversary $\mathcal{A}$, this is negligible:

$$\left| \Pr\left[b' = b \;\middle|\; \begin{array}{c} \mathsf{ck} \leftarrow \mathsf{Kgen}(1^\lambda), (m_0, m_1) \leftarrow \mathcal{A}(\mathsf{ck}) \\ b \leftarrow\!\!\$\, \{0,1\}, b' \leftarrow \mathcal{A}(\mathsf{ck}, \mathsf{Com}(m_b)) \end{array} \right] - \frac{1}{2} \right| .$$

If the probability is $\frac{1}{2}$ even for any unbounded $\mathcal{A}$, this is called *perfect hiding*. Note that an adversary which outputs b' at random succeeds with probability $\frac{1}{2}$, so this is saying that no adversary can do noticeably better than guessing at random.

Binding: For any PPT adversary $\mathcal{A}$, this is negligible:

$$\Pr\left[\mathsf{Com}(\mathsf{ck}, m, r) = \mathsf{Com}(\mathsf{ck}, m', r') \;\middle|\; \begin{array}{c} \mathsf{ck} \leftarrow \mathsf{Kgen}(1^\lambda) \\ (m, r, m', r') \leftarrow \mathcal{A}(\mathsf{ck}) \end{array} \right] .$$

If the probability is 0 even for any unbounded $\mathcal{A}$, this is called *perfect binding*.

No commitment scheme can be both perfectly binding and perfectly hiding at the same time, one needs to choose at most one of these properties. Any public-key encryption scheme, such as Elgamal, can be seen as a perfectly binding commitment scheme. The Pedersen commitment [Ped92], where the commitment key is a group element $[a]$ and the commitment is $\mathsf{com}([a], m, r) = [m] + r[a]$, is a perfectly hiding commitment scheme that is binding under the discrete logarithm assumption, see Section 2.3.

Hash functions

A hash function is a function which takes in a message and compresses it into a short *digest* of a fixed length. For cryptographic purposes, a hash function is typically required to be collision-resistant, i.e. it must be hard to find two messages which produce the same digest. For certain applications, different properties might be required.

For a full definition, including certain technical nuances dealing with the security parameter, see [KL14, pp. 153–155].

2.3 Assumptions in cryptography

We will give examples of the types of assumptions there are in cryptography, particularly the ones we use in this thesis.

1. A very basic assumption is the assumption which asserts the existence of a one-way function.

2. The *discrete logarithm* assumption asserts the existence of a specific one-way function, namely group exponentiation. (Since the definition of one-way functions we gave in Definition 2.2.1 is about bitstrings, there are some technical nuances we skip here.) Formally, for a group generator GpGen, the assumption states that for any PPT adversary $\mathcal{A}$, this is negligible:

$$\Pr\left[[x'] = [x] \mid \mathsf{gp} \leftarrow \mathsf{GpGen}(1^\lambda), x \leftarrow\!\!\$\, \mathbb{Z}_p, x' \leftarrow \mathcal{A}(\mathsf{gp}, [x])\right] .$$

3. The *decisional Diffie–Hellman* (DDH) assumption states that it is difficult to decide whether a quadruple of group elements forms a Diffie–Hellman tuple, i.e. it is of the form $([1], [x], [y], [xy])$ or if the last element of the tuple is simply a random group element $[z]$. For a group generator GpGen, the assumption states that for any PPT adversary $\mathcal{A}$, this is negligible:

$$\left|\Pr\left[b' = b \;\middle|\; \begin{array}{c} \mathsf{gp} \leftarrow \mathsf{GpGen}(1^\lambda), b \leftarrow\!\!\$\, \{0,1\}, x, y \leftarrow\!\!\$\, \mathbb{Z}_p \\ \text{if } b = 0 \text{ then } z \leftarrow xy \text{ else } z \leftarrow\!\!\$\, \mathbb{Z}_p \\ b' \leftarrow \mathcal{A}(\mathbb{G}, [1], [x], [y], [z]) \end{array}\right] - \frac{1}{2}\right| .$$

4. The *Kernel Matrix Diffie–Hellman* (KerMDH) assumption [MRV16], given here for bilinear pairings, states that given a matrix $[\boldsymbol{A}]_1$ of elements in $\mathbb{G}_1$, it should be hard to find a non-zero vector $\boldsymbol{x}$ of elements in $\mathbb{G}_2$ which belongs to the kernel of $\boldsymbol{A}^T$. Let $\mathcal{D}$ be a distribution of matrices over $\mathbb{Z}_p$ of a fixed size. For a bilinear pairing generator BGGen, the assumption states that for any PPT adversary $\mathcal{A}$, this is negligible:

$$\Pr\left[\boldsymbol{A}^T\boldsymbol{x} = 0 \wedge \boldsymbol{x} \neq \boldsymbol{0} \;\middle|\; \begin{array}{c} \mathsf{bp} \leftarrow \mathsf{BGGen}(1^\lambda), \boldsymbol{A} \leftarrow\!\!\$\, \mathcal{D} \\ [\boldsymbol{x}]_2 \leftarrow \mathcal{A}(\mathsf{bp}, [\boldsymbol{A}]_1) \end{array}\right] .$$

5. The *extended-kernel Matrix Diffie–Hellman* (ExtKerMDH) assumption [CH20], is an extension of the KerMDH assumption. The adversary is given a matrix $[\boldsymbol{D}]_2$ in a group, and is tasked to find a matrix $[\boldsymbol{E}]_2$ and a matrix $[\boldsymbol{F}]_1$ such that $\boldsymbol{F}$ spans the kernel of $\boldsymbol{D}||\boldsymbol{E}$. Let $\mathcal{D}$ be a distribution of matrices over $\mathbb{Z}_p$ of a fixed size. For a bilinear pairing generator BGGen, the assumption states that for any PPT adversary $\mathcal{A}$, this is negligible:

$$\Pr\left[\boldsymbol{F}\left(\begin{smallmatrix}\boldsymbol{D}\\ \boldsymbol{E}\end{smallmatrix}\right) = \boldsymbol{0} \wedge \boldsymbol{F} \text{ has full rank} \;\middle|\; \begin{array}{c} \mathsf{bp} \leftarrow \mathsf{BGGen}(1^\lambda), \boldsymbol{D} \leftarrow\!\!\$\, \mathcal{D}, \\ ([\boldsymbol{F}]_1, [\boldsymbol{E}]_2) \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{D}]_2) \end{array}\right] .$$

6. A very different type of assumption is exemplified by Damgård's knowledge-of-exponent assumption [Dam92]. The assumption states that any adversary which, on input $[1, \alpha]$, can output a pair $[x, x\alpha]$, must know some method for obtaining this x. This is formalized by postulating that for any adversary outputting such a pair, there exists an extractor which has access to the randomness used by the adversary, and which extracts this x. For a group generator GpGen, the assumption states that for any PPT adversary $\mathcal{A}$, there exists a PPT extractor Ext such that this is negligible:

$$\Pr\left[x \neq x' \;\middle|\; \begin{array}{c} \mathsf{gp} \leftarrow \mathsf{GpGen}(\lambda), r \leftarrow\!\!\$\, \mathsf{RND}_\lambda(\mathcal{A}), [\alpha] \leftarrow\!\!\$\, \mathbb{G} \\ [x, x\alpha] \leftarrow \mathcal{A}(\mathsf{gp}, [1, \alpha]; r), x' \leftarrow \mathsf{Ext}(\mathsf{gp}, [1, \alpha]; r) \end{array} \right] .$$

Classifying assumptions

Since there is a massive variety of assumptions used in cryptography, all with their own names and definitions, there is utility in classifying assumptions, so that one can more easily understand what type of assumption one is dealing with.

Perhaps the most natural classification deals with the strength of the assumptions. Naor [Nao03] first presented the idea of classifying assumptions into falsifiable and non-falsifiable assumptions. This notion has been later refined by Gentry and Wichs [GW11], and the current understanding of a falsifiable assumption is an assumption which is a game between an efficient challenger and an adversary, where the challenger can efficiently check whether the adversary broke the assumption. Assumptions 1–4 above are falsifiable. A non-falsifiable assumption is simply any assumption which is not falsifiable, and thus can not be written in this form. Falsifiable assumptions tend to be preferable to non-falsifiable assumptions, as it is hard to reason about non-falsifiable assumptions.

There are several reasons why an assumption might not be falsifiable, and this gives rise to further classification of assumptions, see [Pas13]. One reason could be that the assumption can be written as a game between a challenger and an adversary, but the challenger can not check efficiently whether the adversary succeeded, as is the case for assumption 5. There, the challenger can not efficiently check whether $\boldsymbol{F}$ has full rank given just $[\boldsymbol{F}]_1$, which it would need to do to determine if the adversary succeeded. Further still, some assumptions can not be written as a game at all, such as assumption 6. Unlike most cryptographic assumptions, which are assumptions stating that something is hard, this assumption states that something is rather easy, and thus it is completely different from most cryptographic assumptions.

It is also natural to divide assumptions into concrete or generic assumptions. Concrete assumptions are assumptions that state something about a specific problem, like the DDH assumption. A generic assumption states that some primitive exists, like the assumption that there exists a one-way function. When you create an actual protocol, you need to rely

on concrete assumptions, but generic assumptions can provide theoretical understanding, showing which assumptions are necessary and/or sufficient for certain purposes.

2.4 Zero-knowledge proofs

An interactive proof system consists of two algorithms, the prover P and the verifier V. These algorithms share a common input string $\mathtt{x}$ which may or may not belong to a language $\mathcal{L}$. The algorithms send messages to each other and perform their own computations. Eventually, the verifier outputs either accept or reject. The list of messages sent between the parties is called the *transcript*.

An interactive zero-knowledge proof is an interactive proof that satisfies the following properties:

Completeness: If $\mathtt{x} \in \mathcal{L}$, then the verifier will accept with high probability.

Soundness: If $\mathtt{x} \notin \mathcal{L}$, then the verifier will reject with high probability.

Zero-knowledge: For every verifier V, there exists a simulator which, based only on the statement $\mathtt{x}$, as well as any randomness used by V, can create a simulated transcript which is indistinguishable from a real transcript.

For the sake of simplicity, we are skipping several technical details. These details can be specified, and they do matter when proving relations between different variants of zero-knowledge proofs, or when proving impossibility results.

Σ-protocols

A Σ-protocol is a specific type of interactive proof, which was precisely defined in its full generality by Cramer, Damgård and Schoenmakers [CDS94]. A Σ-protocol is defined for a language $\mathcal{L}$ with a relation $\mathcal{R}$ such that $\mathcal{L} = \{\mathtt{x} \mid \exists \mathtt{w} : (\mathtt{x}, \mathtt{w}) \in \mathcal{R}\}$. It is a three-message interactive proof, where the prover sends an initial message a, the verifier responds with a random element $e \leftarrow\!\!\$\, \mathbb{Z}_p$ or $e \leftarrow\!\!\$\, \{0,1\}^t$, and the prover concludes with a message z. Finally, the verifier either accepts or rejects.

A Σ-protocol must satisfy slightly modified versions of the properties for general zero-knowledge proofs. Completeness is not changed, but soundness and zero-knowledge are replaced with the following:

Special soundness: For any $\mathtt{x}$ and pair of accepting transcripts (a, e, z) and (a, e', z') with $e \neq e'$, one can efficiently compute $\mathtt{w}$ such that $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$.

Special honest-verifier zero-knowledge: There exists a simulator Sim which, given any $\mathtt{x}$ and e, can output an accepting transcript (a, e, z), and the distribution of

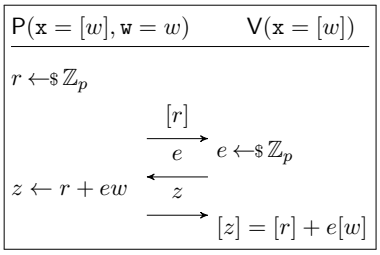


Figure 2.1: Schnorr's Σ-protocol to prove knowledge of a discrete logarithm

this transcript is exactly the same as the distribution of transcripts in an honest execution of the protocol.

These properties are different from the typical requirements of a zero-knowledge proof, but there are ways to build a standard interactive zero-knowledge proof from a Σ-protocol satisfying these special properties, see [Dam00].

Every Σ-protocol is also a *proof of knowledge*. A proof of knowledge for a relation $\mathcal{R}$ is a proof system such that any prover which computes a valid proof for $\mathtt{x}$ must indeed know $\mathtt{w}$ such that $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$. This is formalized by stating that there exists an extractor Ext which obtains this $\mathtt{w}$ from the prover.

Example construction

In Figure 2.1 we show Schnorr's protocol for proving knowledge of a discrete logarithm [Sch90]. The prover knows the discrete logarithm of some public group element $[w]$ and wishes to convince the verifier of this fact. It is important to use a proof of knowledge here because a regular proof could only show that $[w]$ is a group element, which is not helpful. The special soundness property follows because, given two transcripts $([r], e, z)$ and $([r], e', z')$ with $e \neq e'$, one can compute $w = \frac{z-z'}{e-e'}$. Special honest-verifier zero-knowledge follows because the simulator can choose a random z and construct the transcript $([z] - e[w], e, z)$, which is identically distributed to a real transcript.

2.5 Non-interactive zero-knowledge proofs

We now give a definition of non-interactive zero-knowledge (NIZK) in the common reference string (CRS) model. Recall that Goldreich and Oren [GO94] proved that it is impossible to have a NIZK in the plain model, and using a CRS is the standard way of bypassing this impossibility result. The idea behind the CRS is that it comes with a simulation trapdoor only given to the simulator, which allows the simulator to create simulated proofs. This simulator is only a mathematical construct to demonstrate zero-knowledge and does not exist in real life. It is important in practical applications

that the simulation trapdoor is kept hidden, which is why the CRS creator needs to be trusted.

Definition 2.5.1 (Non-interactive zero-knowledge)**.** A non-interactive zero-knowledge proof Π for a relation $\mathcal{R}$ is a tuple $\Pi = (\mathsf{Pgen}, \mathsf{Kgen}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$ of algorithms:

- The parameter generator Pgen takes as input a security parameter 1^λ and outputs some parameters p. For pairing-based applications, this is the description of the bilinear pairing using BGGen. We will not explicitly write the parameters in our security definitions.
- The CRS generator Kgen which takes as input a security parameter 1^λ and some parameters p and outputs a CRS crs and a simulation trapdoor td.
- The prover P which takes as input a CRS crs and a pair $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$ and outputs a proof π.
- The verifier V which takes as input a CRS crs, a statement $\mathtt{x}$ and a proof π and outputs either 0 or 1, respectively rejecting or accepting the proof.
- The zero-knowledge simulator Sim takes as input the simulation trapdoor td and a statement $\mathtt{x}$ and outputs a simulated proof π.

These algorithms must satisfy the following properties:

Completeness: An honest verifier should accept a proof from an honest prover. For all $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$, this is negligible:

$$\Pr\left[\mathsf{V}(\mathsf{crs}, \mathtt{x}, \pi) = 0 \mid \mathsf{crs} \leftarrow \mathsf{Kgen}(1^\lambda), \pi \leftarrow \mathsf{P}(\mathsf{crs}, (\mathtt{x}, \mathtt{w}))\right] \ .$$

If the probability is precisely 0, i.e. the verifier always accepts, this is called *perfect completeness*.

Soundness: No dishonest prover should be able to convince a verifier of the truth of a statement not in the language. There are several flavours of soundness depending on how powerful the adversary is allowed to be and what its goal is. This is *adaptive soundness*, where the adversary can choose the statement it creates a false proof for.

For any adversary $\mathcal{A}$, this is negligible:

$$\Pr\left[\mathsf{V}(\mathsf{crs}, \mathtt{x}, \pi) = 1 \mid \mathsf{crs} \leftarrow \mathsf{Kgen}(1^\lambda), (\mathtt{x}, \pi) \leftarrow \mathcal{A}(\mathsf{crs})\right] \ .$$

If the adversary is restricted to be a PPT adversary, this is *computational soundness*. In this case, we call Π an *argument*. If the adversary can be of unbounded computational power, this is *statistical soundness*. If the probability is 0, even when the adversary has unbounded computational power, this is *perfect soundness*.

Zero-knowledge: The simulator's proof of a true statement must be indistinguishable from an honest proof of that statement. For any adversary $\mathcal{A}$, this is negligible:

$$\left| \Pr\left[b' = b \;\middle|\; \begin{array}{c} (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Kgen}(1^\lambda), (\mathtt{x}, \mathtt{w}) \leftarrow \mathcal{A}(\mathsf{crs}), b \leftarrow\!\!\$\, \{0,1\} \\ \text{if } b = 0 \text{ then } \pi \leftarrow \mathsf{P}(\mathsf{crs}, (\mathtt{x}, \mathtt{w})) \text{ else } \pi \leftarrow \mathsf{Sim}(\mathsf{td}, \mathtt{x}) \\ b' \leftarrow \mathcal{A}(\mathsf{crs}, \mathtt{x}, \mathtt{w}, \pi) \end{array} \right] - \frac{1}{2} \right| .$$

If this holds for PPT adversaries, this is *computational zero-knowledge*. If this holds for unbounded adversaries, this is *statistical zero-knowledge*. If the distributions of the simulated and honest proofs are the same, this is *perfect zero-knowledge*.

No NIZK which is complete can achieve perfect soundness and perfect zero-knowledge at the same time.

The Fiat–Shamir transform

The Fiat–Shamir transform [FS87] is a generic method for transforming a Σ-protocol into a non-interactive zero-knowledge argument. Recall that in a Σ-protocol, after the prover's first message a, the verifier responds with a uniformly random value e, which the prover uses to compute its final message z. The idea of the Fiat–Shamir transform is to let the prover compute e on its own, but in such a way that it cannot choose an e which lets it cheat. In the Fiat–Shamir transform, a hash function H is specified ahead of time, and the prover computes e as $H(x||a)$, hashing the statement concatenated with the first message. It then computes the third message using this e as in the original protocol and sends (a, z) as the proof. The verifier computes $e = H(x||a)$, and checks if the transcript (a, e, z) is valid.

One can prove that the resulting NIZK is sound and zero-knowledge if the hash function is modelled as a random oracle, meaning that on any input, the output is a uniformly random value, independent of any other inputs or outputs, which is not the case for any real hash function. This proof method using a random oracle was pioneered by Bellare and Rogaway [BR93]. It has been prevalent in the field ever since as a way to gain confidence in practical protocols. However, there are results which provide reasons to be sceptical about relying on the random oracle model. Canetti, Goldreich and Halevi [CGH98] showed examples of encryption schemes which are secure in the random oracle model but are insecure when instantiated with any hash function, and Goldwasser and Kalai [GK03] did the same for the Fiat–Shamir transform. While this by no means implies that any NIZK using the Fiat–Shamir transform is insecure, it gives reasonable room for doubt and provides ample motivation to study alternative approaches.

Chapter 3

Overview of Paper I

3.1 Motivation

The Fiat–Shamir transform [FS87] is a method of creating non-interactive zero-knowledge arguments from Σ-protocols, but it produces NIZKs which are only secure in the random oracle model, which is not desirable, see Section 2.5. Couteau and Hartmann [CH20] provided a different method of compiling a Σ-protocol into a NIZK, which did result in a provably secure NIZK in the CRS model. The approach they presented was able to strike a good balance between the strength of assumptions and the efficiency of the NIZK. However, there remained several open questions, which we answer in the affirmative.

- Their compilation worked for a very specific Σ-protocol, but are there other Σ-protocols it would be fruitful to transform to NIZKs, perhaps with improved efficiency?
- Could one base the NIZK on a weaker assumption, perhaps with better evidence of its security?
- Their NIZK supported algebraic languages, would it be possible to support a broader class of languages?
- Their NIZK required a description of the language in a very specific form, could this be achieved with less expertise?

3.2 Previous work

There are a number of NIZKs in the literature, each with its own benefits and drawbacks. For example, there are several constructions of zk-SNARKs in the literature [Gro10; Lip12; Gen+13; Gro16]. Some of them are very efficient, but they all rely on non-falsifiable knowledge-of-exponent assumptions. On the other hand, Groth and Sahai [GS08] constructed NIZKs from a standard falsifiable assumption. However, the efficiency often remains unsatisfying, and moreover building an optimized Groth–Sahai proof requires significant expertise.

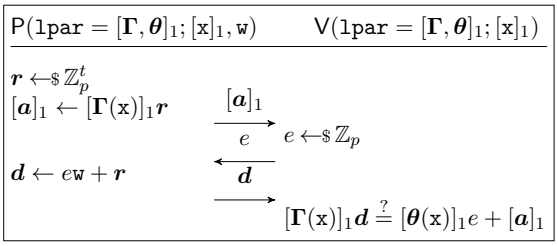


Figure 3.1: Maurer's Σ-protocol for algebraic languages $\mathcal{L}_{\boldsymbol{\Gamma},\boldsymbol{\theta}}$.

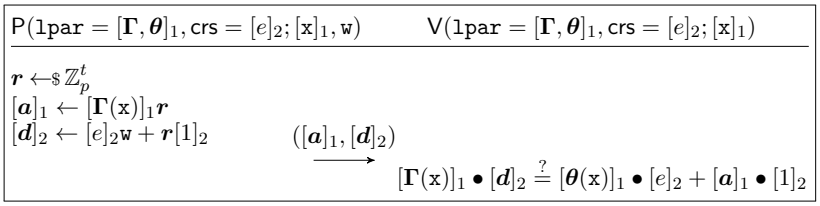


Figure 3.2: Couteau and Hartmann's NIZK for algebraic languages $\mathcal{L}_{\boldsymbol{\Gamma},\boldsymbol{\theta}}$.

A generic method to create NIZKs is to use the Fiat–Shamir transform [FS87], where one compiles a Σ-protocol into a NIZK by letting the prover compute the verifier's messages in a transparent manner. This can yield very efficient NIZKs, but their security is only proven in the random oracle model, which can be undesirable, see Section 2.5.

Couteau and Hartmann [CH20] created a new paradigm to create NIZKs, which created a different set of trade-offs between assumptions and efficiency, landing between zk-SNARKs and Groth–Sahai proofs in both of these categories. Their NIZK was defined for algebraic languages, where an algebraic language is parametrised by two linear maps $\boldsymbol{\Gamma}$, $\boldsymbol{\theta}$, and the language is defined as $\mathcal{L}_{\boldsymbol{\Gamma},\boldsymbol{\theta}} = \{\mathtt{x} : \exists \mathtt{w}, \boldsymbol{\Gamma}(\mathtt{x}) \cdot \mathtt{w} = \boldsymbol{\theta}(\mathtt{x})\}$. This is a generalization of the set of linear languages, where the maps $\boldsymbol{\Gamma}$ and $\boldsymbol{\theta}$ are constant. Of particular relevance, algebraic languages capture the language $\mathcal{L}_{\{0,1\}}$ which consists of Elgamal encryptions of 0 or 1. NIZKs for this language have found a variety of applications.

Couteau and Hartmann started with a bilinear pairing $\mathsf{bp} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, [1]_1, [1]_2, \hat{e})$, and Maurer's Σ-protocol for algebraic languages Figure 3.1 in $\mathbb{G}_1$. By putting the verifier's challenge as $[e]_2$ in the CRS, one can transform the prover's final message and the verification equation by using pairings, and one gets the NIZK in Figure 3.2. The soundness of the NIZK is based on the ExtKerMDH assumption, introduced by Couteau and Hartmann in their paper. See Section 2.3 for its definition.

3.3 Our solution

The main idea of this paper is to use the approach of Couteau and Hartmann, but apply it with a different Σ-protocol for different languages. We construct a NIZK for languages

which are Elgamal encryptions of members of algebraic sets. An algebraic set $\mathcal{A}(\mathcal{F}) \subseteq \mathbb{Z}_p^n$, defined by a set of polynomials $\mathcal{F} \subseteq \mathbb{Z}_p[X]$, consists of all points which are zeros of all polynomials $F \in \mathcal{F}$. In symbols,

$$\mathcal{A}(\mathcal{F}) \coloneqq \{\boldsymbol{\chi} \in \mathbb{Z}_p^n \mid (\forall F \in \mathcal{F})[F(\boldsymbol{\chi}) = 0]\}.$$

Algebraic sets are basic objects of study in algebraic geometry, see [ALO15] for an introduction.

The languages we are concerned about are parameterised by a public key pk and an algebraic set $\mathcal{A}(\mathcal{F})$, in symbols

$$\mathcal{L}_{\mathsf{pk},\mathcal{A}(\mathcal{F})} = \{[\mathbf{ct}]_1 \mid [\mathbf{ct}]_\mathbf{1} = \mathsf{Enc}_{\mathsf{pk}}([\boldsymbol{\chi}]_1) \wedge \boldsymbol{\chi} \in \mathcal{A}(\mathcal{F})\}.$$

To prove that $[\mathbf{ct}]_1 \in \mathcal{L}_{\mathsf{pk},\mathcal{A}(\mathcal{F})}$, we provide proofs that $F(\boldsymbol{\chi}) = 0$ for all $F \in \mathcal{F}$, where $[\boldsymbol{\chi}]_1 = \mathsf{Dec}([\mathbf{ct}]_1)$. The verifier will only accept that $[\mathbf{ct}]_1 \in \mathcal{L}_{\mathsf{pk},\mathcal{A}(\mathcal{F})}$ if all these subproofs are valid. To prove that $F(\boldsymbol{\chi}) = 0$, we do the following:

1. We construct a matrix representation $\boldsymbol{C}(\boldsymbol{X})$ of affine maps which satisfies that $\det(\boldsymbol{C}(\boldsymbol{X})) = F(\boldsymbol{X})$.

2. We create a Σ-protocol to prove that $\det(\boldsymbol{C}(\boldsymbol{\chi})) = 0$, and use the Couteau-Hartmann approach to convert this Σ-protocol to a NIZK.

Matrix representation of polynomials

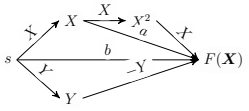
We define a *quasideterminantal representation* (QDR) of a polynomial $F(\boldsymbol{X})$ as a matrix $\boldsymbol{C}(\boldsymbol{X})$ of polynomials which satisfies

i) All entries of $\boldsymbol{C}$ are affine maps, meaning that each entry is of the form $a_0 + \sum_{i=1}^n a_i X_i$ where $\boldsymbol{X} = (X_1, \ldots, X_n)$,

ii) $\det(\boldsymbol{C}(\boldsymbol{X})) = F(\boldsymbol{X})$,

iii) For all $\boldsymbol{\chi} \in \mathbb{Z}_p^n$ such that $F(\boldsymbol{\chi}) = 0$, the first column of $\boldsymbol{C}(\boldsymbol{\chi})$ is in the span of the remaining columns of $\boldsymbol{C}(\boldsymbol{\chi})$.

This is a specialization of *determinantal representations* (DR), which arise in algebraic geometry. A determinantal representation of a polynomial $F(\boldsymbol{X})$ is a matrix $\boldsymbol{C}(\boldsymbol{X})$ which satisfies properties i and ii above. The *determinantal complexity* of a polynomial F is the size of the smallest DR of F, and it is a question of interest in mathematics to compute the determinantal complexity of polynomials. In fact, an algebraic version of the P vs. NP problem (the VP vs. VNP problem, which also remains unsolved) is equivalent to determining the determinantal complexity of a certain class of polynomials. See [SY10] for an introduction to this field.



$$\mathsf{IK}(X,Y) = \begin{pmatrix} X & -1 & 0 & 0 \\ 0 & X & -1 & 0 \\ Y & 0 & 0 & -1 \\ b & a & X & -Y \end{pmatrix}$$

Figure 3.3: ABP example for $F(X,Y) = X^3 + aX + b - Y^2$.

To motivate these properties, we mention what purpose they serve in the resulting Σ-protocol. The first property guarantees that the prover and verifier can compute an encryption of each entry of $\boldsymbol{C}$ based solely on the statement $[\mathbf{ct}]_1$, without knowing the corresponding plaintext values, which works because of the homomorphic properties of Elgamal encryption. The second property guarantees that, when we prove that $\det(\boldsymbol{C}(\boldsymbol{X})) = 0$, this actually corresponds to proving the same statement about F. The motivation for the final property is of a more technical nature, but it is both a necessary and sufficient condition to ensure that the prover will be efficient and to ensure that the zero-knowledge property will hold.

Constructing a QDR from a polynomial is a non-trivial task that highly depends on F. We provide a general framework to construct such QDRs from *algebraic branching programs* (ABPs [Nis91]). An algebraic branching program computes a function $F\colon \mathbb{Z}_p^n \to \mathbb{Z}_p$. It is defined by a directed acyclic graph (V, E) with two special vertices $s, t \in V$ and where each edge is labelled by an affine or constant function in the input variables. The value $F(\boldsymbol{X})$ is the sum over all paths from s to t of the product of the values along the path. Ishai and Kushilevitz [IK00; IK02] found a method to construct a matrix IK from the ABP, and we show that this matrix is a QDR. In Figure 3.3, we show an example of an ABP for the function $F(X,Y) = X^3 + aX + b - Y^2$ and its corresponding QDR.

Σ-protocol and NIZK

To explain our Σ-protocol, we start with the assumption we rely on to prove that the NIZK is sound. We use the CED assumption, which is a slightly weaker version of the $\mathsf{ExtKerMDH}$ assumption of [CH20], here written with different notation. The assumption concerns bilinear pairings, and states that given $[e]_2$, it should be hard to compute $[\boldsymbol{\gamma}]_1 \in \mathbb{G}_1^\ell, [\boldsymbol{C}]_1 \in \mathbb{G}_1^{\ell\times\ell}, [\boldsymbol{\delta}]_2 \in \mathbb{G}_2^{\ell-1}$ such that $\mathrm{rk}(\boldsymbol{C}) = \ell$ and

$$\boldsymbol{\gamma} + \boldsymbol{C}\left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right) = \mathbf{0}. \tag{3.1}$$

The CED assumption simply changes the requirement that $\mathrm{rk}(\boldsymbol{\gamma} \| \boldsymbol{C}) = \ell$ with one where $\mathrm{rk}(\boldsymbol{C}) = \ell$, so every adversary which breaks the CED assumption also breaks the $\mathsf{ExtKerMDH}$ assumption, so CED is a weaker assumption.

In our protocol, we will verify an equivalent version of Equation (3.1) using ciphertexts. The matrix $\boldsymbol{C}$ in Equation (3.1) will be $\boldsymbol{C}(\boldsymbol{\chi})$, where $\boldsymbol{C}$ is a quasideterminantal

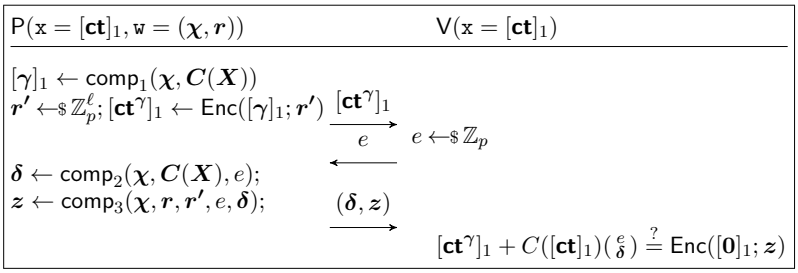


Figure 3.4: The new Σ-protocol from this paper.

representation of F. The idea is that if the prover is cheating, then $F(\boldsymbol{\chi}) \neq 0$, and since $\det(\boldsymbol{C}(\boldsymbol{\chi})) = F(\boldsymbol{\chi})$, then $\det(\boldsymbol{C}(\boldsymbol{\chi})) \neq 0$. By basic linear algebra, this ensures that $\mathrm{rk}(\boldsymbol{C}(\boldsymbol{\chi})) = \ell$, and the cheating prover has broken the assumption.

The Σ-protocol is pictured in Figure 3.4, in a slightly simplified version. The functions comp_i have concrete definitions we omit here for simplicity. In the NIZK, the verifier's challenge is put in $\mathbb{G}_2$ as $[e]_2$, and the proof consists of the unchanged first message, as well as $[\boldsymbol{\delta}]_2$ and $[\boldsymbol{z}]_2$ which are computed from $[e]_2$.

Properties

The efficiency of this NIZK depends on the representation $\boldsymbol{C}$ of F. The size of the proof increases linearly with the size of the matrix $\boldsymbol{C}$. The running times of the prover and the verifier also increase with the size of the matrix, but they additionally depend on the structure of the matrix. Essentially, smaller and simpler matrices lead to more efficient proofs.

The NIZK is proven to be sound based on the novel CED assumption, which is a weaker assumption than the ExtKerMDH assumption used in [CH20]. Additionally, we show in this paper that the CED assumption can be reduced to a very natural gap assumption, essentially stating that having knowledge of the structure in $\mathbb{G}_1$ does not help with solving a certain hard problem in $\mathbb{G}_2$. The CED assumption is in general not falsifiable, but it is for certain specific cases, importantly when F has a very small number of roots.

3.4 Comparison with previous work

A very straightforward application of our general framework is to create a set membership proof. To prove that a ciphertext encrypts a value in the set $\{a_1, \ldots, a_n\}$, we can apply our framework with the polynomial $F(X) = \prod_{i=1}^{n}(X - a_i)$. Our framework yields a set membership proof which has proof size $2n$ elements in $\mathbb{G}_1$ and $2n - 1$ elements in $\mathbb{G}_2$. This represents an approximately 14% decrease in the size of the proof one can get by

optimizing Groth–Sahai proofs [Ràf15] as well as an approximately 33% decrease in the size of the proof one can get by applying Couteau and Hartmann's approach [CH20]. For this application the CED assumption is falsifiable.

We can also use our framework to construct a NIZK for NP. It suffices to create a NIZK for proving that a boolean circuit is satisfiable. We use the well-known technique of Groth, Ostrovsky and Sahai [GOS06], where we encrypt all inputs and intermediate wires in the circuit, and then for each gate in the circuit, use our framework to prove that the gate is computed correctly. Finally, one proves that the output of the final gate is 1. Compared to the existing optimized Groth–Sahai proof for Boolean circuits by Ghadafi et al. [GSW09], our arguments are 20% shorter for the AES circuit described in [GSW09], and this application also relies on a falsifiable version of the CED assumption.

Chapter 4

Overview of Paper II

4.1 Motivation

Non-interactive zero-knowledge arguments are typically defined in a setting where the CRS is generated by a trusted third party, and this CRS generator needs to behave honestly in order for the completeness, soundness and zero-knowledge properties to hold. While there are approaches which reduce the amount of trust required (by designing protocols where one only needs to trust one out of the n parties generating the CRS), it is desirable to have provable security guarantees without needing to trust anyone.

Attempting to reduce trust in central entities is a crucial part of *subversion-resistant* cryptography. Several cryptographic constructions rely on pre-shared parameters set up ahead of time, which need to be chosen by some entity. One example is the choice of an elliptic curve group for various schemes, and another is the CRS for a public NIZK. If someone could subvert these parameters in a specific way, they might be able to break the cryptographic schemes. An infamous example of this is the NIST `Dual_EC_DRBG` curve [BLN15], which did contain a backdoor which allowed someone who knew certain secret values to break security. The goal of subversion-resistant cryptography is to make such subversion attempts impossible, or at least to detect when they are attempted.

A natural question is whether one can create subversion-resistant NIZKs, and which properties can be made subversion-resistant at the same time. There are theoretical limits to what is possible to achieve. Importantly, it is not possible to achieve all three of completeness, soundness and zero-knowledge without some trust in the CRS generator. If this was the case, the prover could generate the CRS on its own, and one would have a non-interactive zero-knowledge proof without a CRS, which is impossible. Bellare, Fuchsbauer and Scafuro [BFS16] additionally showed that it is not possible to achieve zero-knowledge if both the soundness and completeness properties are resistant to subversion. However, they did give a construction of a NIZK where the completeness and zero-knowledge properties are resistant to subversion, and the NIZK satisfies soundness if the CRS generator behaves honestly, which is called a Sub-ZK NIZK.

Our paper sets out to partially answer which assumptions are needed to create Sub-ZK NIZKs.

4.2 Previous work

There are different constructions of Sub-ZK NIZKs in the literature. Bellare, Fuchsbauer and Scafuro [BFS16] gave the first construction of a Sub-ZK NIZK, and both Abdolmaleki et al. [Abd+17] and Fuchsbauer [Fuc18] gave constructions of Sub-ZK SNARKs.

We also build on work surrounding extractable one-way functions, first introduced by Canetti and Dakdouk [CD08], and generalized by Bitansky et al. [Bit+16]. Work on extractable collision-resistant hash functions and their relationship with zk-SNARKs [Bit+17] also inspired this paper.

An extractable one-way function (EOWF) is a one-way function which comes with an extractability property. While it should be hard to produce a preimage of the function given just an output, it should be possible to produce such a preimage if one is given access to a machine computing the function. Essentially, it must be possible to extract the input from a machine computing the function, because such a machine must know a way to find a preimage. This definition formalizes the notion of a knowledge assumption, of which Damgård's Knowledge-of-Exponent assumption is an example.

The definition of the subversion zero-knowledge property is that for any CRS creator which produces a valid CRS there must exist an extractor which computes a simulation trapdoor, which can be used to create simulated proofs. It is important that checking whether the CRS is valid can be done efficiently and by a public algorithm. The previous approaches by [BFS16; Abd+17; Fuc18] to create Sub-ZK NIZKs all followed a similar structure. They ensured that the CRS contained the right elements which would make it possible to verify with a pairing that it was a valid CRS, and used a specific knowledge-of-exponent assumption to extract the simulation trapdoor.

4.3 Our solution

We define a primitive we call verifiably-extractable one-way functions (VEOWF). This is an extractable one-way function, where it additionally must be possible to verify whether some value is in the image of the function. More formally, a function f is a VEOWF if it is:

One-way: Given a value $y = f(x)$, it should be hard to output x' such that $f(x') = y$,

Extractable: For any adversary $\mathcal{A}$ outputting an image y of f, there exists an extractor Ext which outputs a preimage of y,

Verifiable: There exists an efficient algorithm ImV which, given a value y in the codomain of f can decide if y is in the image of f.

The intuition for why this definition has utility compared to an EOWF is that there are circumstances, in particular related to Sub-ZK NIZKs, where we want to check in public whether extraction would be successful, but without actually performing the extraction.

An example of a VEOWF comes from Damgård's Knowledge-of-Exponent assumption. Let bp be a symmetric bilinear pairing, meaning that $\mathbb{G}_1 = \mathbb{G}_2$, and let $[\alpha]_1$ be a randomly chosen group element. The function $f(x) = [x, x\alpha]_1$ is a VEOWF where the one-way property follows from the discrete logarithm assumption, the extractability follows from Damgård's Knowledge-of-Exponent assumption, see Section 2.3 for their definition, and the verifiability follows from the use of pairings. To verify whether an element $[y, z]_1$ belongs to the image of f, one simply checks whether $[y]_1 \bullet [\alpha]_1 = [1]_1 \bullet [z]_1$.

We also define verifiably-extractable generalized one-way functions (VEGOWF), where the function f comes with a relation $\mathbf{RG}$, and a set $Y_{\mathsf{Ext}} \supseteq \mathrm{im}(f)$, and must satisfy:

RG-hardness: Given a value $y = f(x)$, it should be hard to output z such that $\mathbf{RG}(y, z) = 1$,

Verifiability: Given y one can efficiently verify whether $y \in Y_{\mathsf{Ext}}$,

Extractability: For any adversary $\mathcal{A}$ which outputs $y \in Y_{\mathsf{Ext}}$ there exists an extractor $\mathsf{Ext}_{\mathcal{A}}$ which extracts z such that $\mathbf{RG}(y, z) = 1$.

Note that extraction should work even if $y \in Y_{\mathsf{Ext}} \setminus \mathrm{im}(\mathsf{g_e})$, and in general, it might be hard to decide if $y \in \mathrm{im}(\mathsf{g_e})$. Note that all VEOWFs are VEGOWFs with $\mathbf{RG}(y, z) = 1 \iff y = f(z)$ and $Y_{\mathsf{Ext}} = \mathrm{im}(f)$, but not necessarily vice versa. There is in particular one construction from the literature by Bitansky et al. [Bit+16] which results in a VEGOWF but not a VEOWF.

To demonstrate the plausibility of our new primitives, we show how they can be instantiated from a number of established assumptions in the literature.

In addition to the VEOWF from Damgård's Knowledge-of-Exponent assumption, a family of VEGOWFs come from knowledge-of-exponent style assumptions in groups with pairings, examples of which we provide in the paper. The one-way property is based on a standard hardness assumption about groups, the extractability property comes from a knowledge-of-exponent assumption, and finally, the pairing is used to construct ImV.

We show that the construction by Bitansky et al. [Bit+16] of a GEOWF from delegation schemes in a restricted model is also a VEGOWF. Delegation schemes are SNARGs, but only for P. They allow you to delegate a polynomial-time computation and check whether the computation was performed correctly in significantly less time than it takes to perform the computation itself. Unlike SNARGs, delegation schemes can be based on falsifiable assumptions.

Additionally, we show that any knowledge-sound NIZK gives rise to a VEGOWF. Because of the zero-knowledge property, it is hard to compute a witness from the statement

and a proof of that statement. However, because of the knowledge-sound property, it is possible to extract a witness from any prover convincing the verifier. Finally, the basic soundness property of the NIZK ensures verifiability.

We provide two generic constructions where we use a VEGOWF as a core ingredient to transform a proof system Π with certain properties into a Sub-ZK NIZK. We give a brief overview of the construction for the case of VEOWFs, but the same construction works for VEGOWFs as well. The essential idea is to apply the well-known FLS [FLS90] approach with a VEOWF f. The CRS generator generates the CRS of Π, and also picks a value x, and includes $y = f(x)$ in the new CRS. To prove that a statement $\mathtt{x}$ is true, the prover provides a proof using Π that either the prover knows a witness for $\mathtt{x}$, or the prover knows a preimage of y. Since an honest prover does not know a preimage of y, and such a preimage is hard to compute, the only way the prover can create an accepting proof is if it actually knows a witness for $\mathtt{x}$, and thus the statement is true. To demonstrate subversion zero-knowledge, the simulator extracts a preimage of f from the CRS generator and uses this to create a simulated proof. By the properties of the proof system, this simulated proof looks indistinguishable from a real proof. The two constructions differ based on the requirements of the underlying proof system and the details of the techniques we use.

Our final contribution is to show that for any Sub-ZK NIZK with certain additional properties, the algorithm used to generate the CRS is in fact a VEGOWF.

4.4 Comparison with previous work

Prior to our work, it was known how to obtain Sub-ZK NIZKs using concrete knowledge assumptions. In this work we show how subversion zero-knowledge can be obtained by using a generic assumption instead, deepening the theoretical understanding of subversion zero-knowledge.

One particular application of our work is that we show how, from any candidate VEGOWF, one can construct a Sub-ZK SNARK where the subversion zero-knowledge property solely relies on the security of the VEGOWF.

Bibliography

[Abd+17] Behzad Abdolmaleki, Karim Baghery, Helger Lipmaa, and Michal Zajac. "A Subversion-Resistant SNARK". In: *ASIACRYPT 2017, Part III*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10626. LNCS. Springer, Heidelberg, Dec. 2017, pp. 3–33. DOI: 10.1007/978-3-319-70700-6_1.

[ALO15] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. 4th ed. Undergraduate Texts in Mathematics. Springer, May 2015, p. 662.

[Ben+14] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. "Zerocash: Decentralized Anonymous Payments from Bitcoin". In: *2014 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2014, pp. 459–474. DOI: 10.1109/SP.2014.36.

[Ben+90] Michael Ben-Or, Oded Goldreich, Shafi Goldwasser, Johan Håstad, Joe Kilian, Silvio Micali, and Phillip Rogaway. "Everything Provable is Provable in Zero-Knowledge". In: *CRYPTO'88*. Ed. by Shafi Goldwasser. Vol. 403. LNCS. Springer, Heidelberg, Aug. 1990, pp. 37–56. DOI: 10.1007/0-387-34799-2_4.

[BF01] Dan Boneh and Matthew K. Franklin. "Identity-Based Encryption from the Weil Pairing". In: *CRYPTO 2001*. Ed. by Joe Kilian. Vol. 2139. LNCS. Springer, Heidelberg, Aug. 2001, pp. 213–229. DOI: 10.1007/3-540-44647-8_13.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. "Non-Interactive Zero-Knowledge and Its Applications (Extended Abstract)". In: *20th ACM STOC*. ACM Press, May 1988, pp. 103–112. DOI: 10.1145/62212.62222.

[BFS16] Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. "NIZKs with an Untrusted CRS: Security in the Face of Parameter Subversion". In: *ASIACRYPT 2016, Part II*. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10032. LNCS. Springer, Heidelberg, Dec. 2016, pp. 777–804. DOI: 10.1007/978-3-662-53890-6_26.

[Bit+16] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. "On the Existence of Extractable One-Way Functions". In: *SIAM J. Comput.* 45.5 (2016), pp. 1910–1952.

[Bit+17] Nir Bitansky, Ran Canetti, Alessandro Chiesa, Shafi Goldwasser, Huijia Lin, Aviad Rubinstein, and Eran Tromer. "The Hunting of the SNARK". In: *Journal of Cryptology* 30.4 (Oct. 2017), pp. 989–1066. DOI: 10.1007/s00145-016-9241-9.

[BLN15] Daniel J. Bernstein, Tanja Lange, and Ruben Niederhagen. *Dual EC: A Standardized Back Door*. Cryptology ePrint Archive, Report 2015/767. https://eprint.iacr.org/2015/767. 2015.

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. "Short Signatures from the Weil Pairing". In: *Journal of Cryptology* 17.4 (Sept. 2004), pp. 297–319. DOI: 10.1007/s00145-004-0314-9.

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

[BSS00] Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart. *Elliptic Curves in Cryptography*. ISBN: 0521653746. Cambridge Univ Pr, Jan. 2000, p. 204.

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. "Extractable Perfectly One-Way Functions". In: *ICALP 2008, Part II*. Ed. by Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz. Vol. 5126. LNCS. Springer, Heidelberg, July 2008, pp. 449–460. DOI: 10.1007/978-3-540-70583-3_37.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. "Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols". In: *CRYPTO'94*. Ed. by Yvo Desmedt. Vol. 839. LNCS. Springer, Heidelberg, Aug. 1994, pp. 174–187. DOI: 10.1007/3-540-48658-5_19.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. "The Random Oracle Methodology, Revisited (Preliminary Version)". In: *30th ACM STOC*. ACM Press, May 1998, pp. 209–218. DOI: 10.1145/276698.276741.

[CH20] Geoffroy Couteau and Dominik Hartmann. "Shorter Non-interactive Zero-Knowledge Arguments and ZAPs for Algebraic Languages". In: *CRYPTO 2020, Part III*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12172. LNCS. Springer, Heidelberg, Aug. 2020, pp. 768–798. DOI: 10.1007/978-3-030-56877-1_27.

[Cha81] David Chaum. "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms". In: *Communications of the ACM* 24.2 (1981), pp. 84–88.

[Cha85] David Chaum. "Security Without Identification: Transaction Systems to Make Big Brother Obsolete". In: *Communications of the ACM* 28.10 (Oct. 1985), pp. 1030–1044.

[Cou+21a] Geoffroy Couteau, Helger Lipmaa, Roberto Parisella, and Arne Tobias Ødegaard. *Efficient NIZKs for Algebraic Sets*. Cryptology ePrint Archive, Report 2021/1251. https://eprint.iacr.org/2021/1251. 2021.

[Cou+21b] Geoffroy Couteau, Helger Lipmaa, Roberto Parisella, and Arne Tobias Ødegaard. "Efficient NIZKs for Algebraic Sets". In: *ASIACRYPT 2021, Part III*. Ed. by Mehdi Tibouchi and Huaxiong Wang. Vol. 13092. LNCS. Springer, Heidelberg, Dec. 2021, pp. 128–158. DOI: 10.1007/978-3-030-92078-4_5.

[Dam00] Ivan Damgård. "Efficient Concurrent Zero-Knowledge in the Auxiliary String Model". In: *EUROCRYPT 2000*. Ed. by Bart Preneel. Vol. 1807. LNCS. Springer, Heidelberg, May 2000, pp. 418–430. DOI: 10.1007/3-540-45539-6_30.

[Dam92] Ivan Damgård. "Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks". In: *CRYPTO'91*. Ed. by Joan Feigenbaum. Vol. 576. LNCS. Springer, Heidelberg, Aug. 1992, pp. 445–456. DOI: 10.1007/3-540-46766-1_36.

[DH76] Whitfield Diffie and Martin E. Hellman. "New Directions in Cryptography". In: *IEEE Trans. Inf. Theory* IT-22 (Nov. 1976), pp. 644–654.

[DL08] Giovanni Di Crescenzo and Helger Lipmaa. "Succinct NP Proofs from an Extractability Assumption". In: *Computability in Europe, CIE 2008*. Ed. by Arnold Beckmann, Costas Dimitracopoulos, and Benedikt Löwe. Vol. 5028. LNCS. Athens, Greece: Springer, Heidelberg, June 2008, pp. 175–185.

[Esc+13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. "An Algebraic Framework for Diffie-Hellman Assumptions". In: *CRYPTO 2013, Part II*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8043. LNCS. Springer, Heidelberg, Aug. 2013, pp. 129–147. DOI: 10.1007/978-3-642-40084-1_8.

[Fau+21a] Prastudy Fauzi, Helger Lipmaa, Janno Siim, Michal Zajac, and Arne Tobias Ødegaard. *Verifiably-Extractable OWFs and Their Applications to Subversion Zero-Knowledge*. Cryptology ePrint Archive, Report 2021/1264. https://eprint.iacr.org/2021/1264. 2021.

[Fau+21b] Prastudy Fauzi, Helger Lipmaa, Janno Siim, Michal Zajac, and Arne Tobias Ødegaard. "Verifiably-Extractable OWFs and Their Applications to Subversion Zero-Knowledge". In: *ASIACRYPT 2021, Part IV*. Ed. by Mehdi

Tibouchi and Huaxiong Wang. Vol. 13093. LNCS. Springer, Heidelberg, Dec. 2021, pp. 618–649. DOI: 10.1007/978-3-030-92068-5_21.

[FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir. "Multiple Non-Interactive Zero Knowledge Proofs Based on a Single Random String (Extended Abstract)". In: *31st FOCS*. IEEE Computer Society Press, Oct. 1990, pp. 308–317. DOI: 10.1109/FSCS.1990.89549.

[FS87] Amos Fiat and Adi Shamir. "How to Prove Yourself: Practical Solutions to Identification and Signature Problems". In: *CRYPTO'86*. Ed. by Andrew M. Odlyzko. Vol. 263. LNCS. Springer, Heidelberg, Aug. 1987, pp. 186–194. DOI: 10.1007/3-540-47721-7_12.

[Fuc18] Georg Fuchsbauer. "Subversion-Zero-Knowledge SNARKs". In: *PKC 2018, Part I*. Ed. by Michel Abdalla and Ricardo Dahab. Vol. 10769. LNCS. Springer, Heidelberg, Mar. 2018, pp. 315–347. DOI: 10.1007/978-3-319-76578-5_11.

[Gen+13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. "Quadratic Span Programs and Succinct NIZKs without PCPs". In: *EUROCRYPT 2013*. Ed. by Thomas Johansson and Phong Q. Nguyen. Vol. 7881. LNCS. Springer, Heidelberg, May 2013, pp. 626–645. DOI: 10.1007/978-3-642-38348-9_37.

[GGP10] Rosario Gennaro, Craig Gentry, and Bryan Parno. "Non-interactive Verifiable Computing: Outsourcing Computation to Untrusted Workers". In: *CRYPTO 2010*. Ed. by Tal Rabin. Vol. 6223. LNCS. Springer, Heidelberg, Aug. 2010, pp. 465–482. DOI: 10.1007/978-3-642-14623-7_25.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. "On the (In)security of the Fiat-Shamir Paradigm". In: *44th FOCS*. IEEE Computer Society Press, Oct. 2003, pp. 102–115. DOI: 10.1109/SFCS.2003.1238185.

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

[GO94] Oded Goldreich and Yair Oren. "Definitions and Properties of Zero-Knowledge Proof Systems". In: *Journal of Cryptology* 7.1 (Dec. 1994), pp. 1–32. DOI: 10.1007/BF00195207.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. "Non-interactive Zaps and New Techniques for NIZK". In: *CRYPTO 2006*. Ed. by Cynthia Dwork. Vol. 4117. LNCS. Springer, Heidelberg, Aug. 2006, pp. 97–111. DOI: 10.1007/11818175_6.

[GPS06] S.D. Galbraith, K.G. Paterson, and N.P. Smart. *Pairings for Cryptographers*. Cryptology ePrint Archive, Report 2006/165. https://eprint.iacr.org/2006/165. 2006.

[Gro10] Jens Groth. "Short Pairing-Based Non-interactive Zero-Knowledge Arguments". In: *ASIACRYPT 2010*. Ed. by Masayuki Abe. Vol. 6477. LNCS. Springer, Heidelberg, Dec. 2010, pp. 321–340. DOI: 10.1007/978-3-642-17373-8_19.

[Gro16] Jens Groth. "On the Size of Pairing-Based Non-interactive Arguments". In: *EUROCRYPT 2016, Part II*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9666. LNCS. Springer, Heidelberg, May 2016, pp. 305–326. DOI: 10.1007/978-3-662-49896-5_11.

[GS08] Jens Groth and Amit Sahai. "Efficient Non-interactive Proof Systems for Bilinear Groups". In: *EUROCRYPT 2008*. Ed. by Nigel P. Smart. Vol. 4965. LNCS. Springer, Heidelberg, Apr. 2008, pp. 415–432. DOI: 10.1007/978-3-540-78967-3_24.

[GSW09] Essam Ghadafi, Nigel P. Smart, and Bogdan Warinschi. "Practical Zero-Knowledge Proofs for Circuit Evaluation". In: *12th IMA International Conference on Cryptography and Coding*. Ed. by Matthew G. Parker. Vol. 5921. LNCS. Springer, Heidelberg, Dec. 2009, pp. 469–494.

[GW11] Craig Gentry and Daniel Wichs. "Separating succinct non-interactive arguments from all falsifiable assumptions". In: *43rd ACM STOC*. Ed. by Lance Fortnow and Salil P. Vadhan. ACM Press, June 2011, pp. 99–108. DOI: 10.1145/1993636.1993651.

[IK00] Yuval Ishai and Eyal Kushilevitz. "Randomizing Polynomials: A New Representation with Applications to Round-Efficient Secure Computation". In: *41st FOCS*. IEEE Computer Society Press, Nov. 2000, pp. 294–304. DOI: 10.1109/SFCS.2000.892118.

[IK02] Yuval Ishai and Eyal Kushilevitz. "Perfect Constant-Round Secure Computation via Perfect Randomizing Polynomials". In: *ICALP 2002*. Ed. by Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan Eidenbenz, and Ricardo Conejo. Vol. 2380. LNCS. Springer, Heidelberg, July 2002, pp. 244–256. DOI: 10.1007/3-540-45465-9_22.

[Jou00] Antoine Joux. "A One-Round Protocol for Tripartite Diffie-Hellman". In: *ANTS 2000*. Ed. by Wieb Bosma. Vol. 1838. LNCS. Leiden, The Netherlands: Springer, Heidelberg, Feb. 2000, pp. 385–394.

[Kil92] Joe Kilian. "A Note on Efficient Zero-Knowledge Proofs and Arguments (Extended Abstract)". In: *24th ACM STOC*. ACM Press, May 1992, pp. 723–732. DOI: 10.1145/129712.129782.

[KL14] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. 2nd. Chapman & Hall/CRC, 2014. ISBN: 1466570261.

[Lip12] Helger Lipmaa. "Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments". In: *TCC 2012*. Ed. by Ronald Cramer. Vol. 7194. LNCS. Springer, Heidelberg, Mar. 2012, pp. 169–189. DOI: 10.1007/978-3-642-28914-9_10.

[Men09] Alfred Menezes. "An introduction to pairing-based cryptography". In: *Recent trends in cryptography* 477 (2009), pp. 47–65.

[Mer78] Ralph C. Merkle. "Secure Communications over Insecure Channels". In: *Commun. ACM* 21.4 (Apr. 1978), pp. 294–299. ISSN: 0001-0782. DOI: 10.1145/359460.359473. URL: https://doi.org/10.1145/359460.359473.

[Mic94] Silvio Micali. "CS Proofs (Extended Abstracts)". In: *35th FOCS*. IEEE Computer Society Press, Nov. 1994, pp. 436–453. DOI: 10.1109/SFCS.1994.365746.

[MRV16] Paz Morillo, Carla Ràfols, and Jorge Luis Villar. "The Kernel Matrix Diffie-Hellman Assumption". In: *ASIACRYPT 2016, Part I*. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10031. LNCS. Springer, Heidelberg, Dec. 2016, pp. 729–758. DOI: 10.1007/978-3-662-53887-6_27.

[Nao03] Moni Naor. "On Cryptographic Assumptions and Challenges (Invited Talk)". In: *CRYPTO 2003*. Ed. by Dan Boneh. Vol. 2729. LNCS. Springer, Heidelberg, Aug. 2003, pp. 96–109. DOI: 10.1007/978-3-540-45146-4_6.

[Nis91] Noam Nisan. "Lower Bounds for Non-Commutative Computation (Extended Abstract)". In: *23rd ACM STOC*. ACM Press, May 1991, pp. 410–418. DOI: 10.1145/103418.103462.

[Pas13] Rafael Pass. "Unprovable Security of Perfect NIZK and Non-interactive Non-malleable Commitments". In: *TCC 2013*. Ed. by Amit Sahai. Vol. 7785. LNCS. Springer, Heidelberg, Mar. 2013, pp. 334–354. DOI: 10.1007/978-3-642-36594-2_19.

[Ped92] Torben P. Pedersen. "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing". In: *CRYPTO'91*. Ed. by Joan Feigenbaum. Vol. 576. LNCS. Springer, Heidelberg, Aug. 1992, pp. 129–140. DOI: 10.1007/3-540-46766-1_9.

[Ràf15] Carla Ràfols. "Stretching Groth-Sahai: NIZK Proofs of Partial Satisfiability". In: *TCC 2015, Part II*. Ed. by Yevgeniy Dodis and Jesper Buus Nielsen. Vol. 9015. LNCS. Springer, Heidelberg, Mar. 2015, pp. 247–276. DOI: 10.1007/978-3-662-46497-7_10.

[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 (Feb. 1978), pp. 120–126.

[Sch90] Claus-Peter Schnorr. "Efficient Identification and Signatures for Smart Cards". In: *CRYPTO'89*. Ed. by Gilles Brassard. Vol. 435. LNCS. Springer, Heidelberg, Aug. 1990, pp. 239–252. DOI: 10.1007/0-387-34805-0_22.

[Sha92] Adi Shamir. "IP = PSPACE". In: *J. ACM* 39.4 (Oct. 1992), pp. 869–877. ISSN: 0004-5411. DOI: 10.1145/146585.146609. URL: https://doi.org/10.1145/146585.146609.

[Sin99] Simon Singh. *The Code Book: The Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography*. 1st. USA: Doubleday, 1999. ISBN: 0385495315.

[Sip13] Michael Sipser. *Introduction to the Theory of Computation*. Third. Boston, MA: Course Technology, 2013. ISBN: 113318779X.

[SK95] Kazue Sako and Joe Kilian. "Receipt-Free Mix-Type Voting Scheme - A Practical Solution to the Implementation of a Voting Booth". In: *EUROCRYPT'95*. Ed. by Louis C. Guillou and Jean-Jacques Quisquater. Vol. 921. LNCS. Springer, Heidelberg, May 1995, pp. 393–403. DOI: 10.1007/3-540-49264-X_32.

[SY10] Amir Shpilka and Amir Yehudayoff. "Arithmetic Circuits: A Survey of Recent Results and Open Questions". In: *Foundations and Trends® in Theoretical Computer Science* 5.3–4 (2010), pp. 207–388. ISSN: 1551-305X. DOI: 10.1561/0400000039. URL: http://dx.doi.org/10.1561/0400000039.

Chapter 5

Paper I

Efficient NIZKs for Algebraic Sets

Geoffroy Couteau, Helger Lipmaa, Roberto Parisella and Arne Tobias Ødegaard

ASIACRYPT, December 2021.

Abstract

Significantly extending the framework of (Couteau and Hartmann, Crypto 2020), we propose a general methodology to construct NIZKs for showing that an encrypted vector $\boldsymbol{\chi}$ belongs to an algebraic set, i.e., is in the zero locus of an ideal $\mathcal{I}$ of a polynomial ring. In the case where $\mathcal{I}$ is principal, i.e., generated by a single polynomial F, we first construct a matrix that is a "quasideterminantal representation" of F and then a NIZK argument to show that $F(\boldsymbol{\chi}) = 0$. This leads to compact NIZKs for general computational structures, such as polynomial-size algebraic branching programs. We extend the framework to the case where $\mathcal{I}$ is non-principal, obtaining efficient NIZKs for R1CS, arithmetic constraint satisfaction systems, and thus for NP. As an independent result, we explicitly describe the corresponding language of ciphertexts as an algebraic language, with smaller parameters than in previous constructions that were based on the disjunction of algebraic languages. This results in an efficient GL-SPHF for algebraic branching programs.

5.1 Introduction

Zero-knowledge arguments [GMR89] are fundamental cryptographic primitives allowing one to convince a verifier of the truth of a statement while concealing all further information. A particularly appealing type of zero-knowledge arguments, with a wide variety of applications in cryptography, are *non-interactive zero-knowledge arguments* (NIZKs) [BFM88] with a single flow from the prover to the verifier.

Early feasibility results from the 90's established the existence of NIZKs for all NP languages (in the common reference string model) under standard cryptographic assumptions. However, these early constructions were inefficient. In the past decades, a major effort of the cryptographic community has been directed towards obtaining *efficient* and *conceptually simple* NIZK argument systems for many languages of interest. Among the celebrated successes of this line of work are the Fiat-Shamir (FS) transform [FS87], which provides simple and efficient NIZKs but only offers heuristic security guarantees[1], and pairing-based NIZKs such as the Groth-Sahai proof system [GS08] (and its follow-ups).

The quest for efficient and conceptually simple NIZKs. The Groth-Sahai NIZK proof system was a major breakthrough in this line of work, providing the first provably secure (under standard pairing assumptions) and reasonably efficient NIZK for a large class of languages, capturing many concrete languages of interest. This proof system initiated a wide variety of cryptographic applications, and its efficiency was refined in a sequence of works [Bla+10; EG14; Ràf15; Daz+19]. Unfortunately, the efficiency of Groth-Sahai proofs often remains unsatisfying (typically much worse than NIZKs

[1]There have been recent developments towards provably secure Fiat-Shamir NIZKs [CLW18].

obtained with Fiat-Shamir), and building an optimized Groth-Sahai proof for a specific problem is an often tedious process that requires considerable expertise. This lack of conceptual simplicity inhibits the potential for large-scale deployment of this proof system. Therefore, we view it as one of the major open problems in this line of work to obtain an efficient proof system where constructing an optimized proof for a given statement does not require dedicated expertise. The Fiat-Shamir transform offers such a candidate – and as a consequence, it has seen widescale adoption in real-world protocols – but lacks a formal proof of security. The recent line of work on quasi-adaptive NIZKs [JR13; KW15; Abd+20] offers simultaneously simple, efficient, and provably secure proof systems, but these are restricted to a small class of languages – namely, linear languages. Some recent SNARK proof systems also offer generic and efficient methods to handle a large class of languages given by their high-level description; however, they all rely on very strong knowledge-of-exponent style assumptions.

The Couteau-Hartmann argument system. Very recently, Couteau and Hartmann put forth a new framework for constructing pairing based NIZKs [CH20]. At a high level, their approach compiles a specific interactive zero-knowledge proof into a NIZK (as does Fiat-Shamir), by embedding the challenge in the exponent of a group equipped with an asymmetric pairing. The CH argument system enjoys several interesting features:

- It generates compact proofs, with efficiency comparable to Fiat-Shamir arguments, with ultra-short common reference strings (a single group element);
- It has a conceptually simple structure, since it compiles a well-known and simple interactive proof;
- It handles a relatively large class of *algebraic languages* [Ben+13; CC18], which are parameterized languages of the shape $\mathcal{L}_{\mathbf{\Gamma},\boldsymbol{\theta}} = \{\mathtt{x} : \exists \mathtt{w}, \mathbf{\Gamma}(\mathtt{x}) \cdot \mathtt{w} = \boldsymbol{\theta}(\mathtt{x})\}$, where $\mathtt{x}$ is the input, $\mathtt{w}$ is the witness, $\mathbf{\Gamma}$ and $\boldsymbol{\theta}$ are affine maps, such that $\mathtt{x}$ and $\boldsymbol{\theta}(\mathtt{x})$ are vectors and $\mathbf{\Gamma}(\mathtt{x})$ is a matrix. We call $(\boldsymbol{\theta}, \mathbf{\Gamma})$ the *matrix description* of the language $\mathcal{L}$. Since any NP language can be embedded into an algebraic language[2], this gives a proof system for all of NP.

These features make the CH argument system a competitive alternative to Fiat-Shamir and Groth-Sahai in settings where efficiency and conceptual simplicity are desirable while maintaining provable security under a plausible, albeit new, assumption over pairing groups. In a sense, Couteau-Hartmann achieves a sweet spot between efficiency, generality, and underlying assumption.

[2]The classical approach to do so for circuit satisfiability uses algebraic commitments to all values on the wire of the circuit; then the statement "all committed values are consistent and the output is 1" is an algebraic language.

Limitations of the CH argument system. The CH transformation offers attractive efficiency features, but its core advantage is (arguably) its conceptual simplicity. As many previous works pointed out (see e.g. [E K+15]), what "real-world" protocol designers need is a method that can easily take a high-level description of a language, and "automatically" generate a NIZK for this language without going through a tedious and complex process requiring dedicated expertise. Ideally, both the process of generating the NIZK description from the high-level language and the NIZK itself should be efficient.

With this in mind, CH provides an important step in the right direction, where producing the NIZK for any algebraic language is a straightforward generic transformation applied to its matrix description. However, it falls short of fully achieving the desired goal for two reasons.

First, it does not entirely remove the need for dedicated expertise from the NIZK construction; rather, it pushes the complexity of *building the NIZK* to that of *finding its matrix description* given a higher-level description of an algebraic language. However, it does not provide a characterization of which languages, given via a common higher-level description, are algebraic, neither does it give a method to construct their matrix description[3].

Second, the CH-compilation produces NIZKs whose soundness reduces to an instance of the novel ExtKerMDH family of assumptions. However, the particular assumption will only be falsifiable in the much more restricted setting of *witness-samplable* algebraic languages, which essentially seem to capture disjunctions of linear languages. Couteau and Hartmann focused on NIZKs based on the falsifiable variant, which severely limits the class of languages captured by the framework. It is much more desirable to base the security of all NIZKs produced by this framework on a single, plausible, well-supported assumption: this would avoid protocol designers the hurdle of precisely assessing the security of the specific flavor of the ExtKerMDH assumption their particular instance requires.

Our Contribution

We overcome the main limitations of the CH argument system. Our new approach, which significantly departs from the CH methodology, allows us to produce compact NIZKs for a variety of languages, with several appealing features.

A general framework. We provide a generic method to compute, for several important families of languages, a different matrix description of the languages. We then construct a NIZK. We implicitly use the CH-compiler but in a way, different from [CH20]. We focus on the important setting of commit-and-prove NIZK argument systems [Lip16; KOS18;

[3]While we can always embed any language in an algebraic language, this can be inefficient; the CH proof system is efficient when the language is "natively" algebraic.

Kiy20], i.e. languages of the form $\{\mathsf{Com}(x_1), \ldots, \mathsf{Com}(x_n) \mid R(x_1, \ldots, x_n)\}$, where R is some efficiently computable relation. Our method allows us to automatically obtain a compact matrix description for many types of high-level relations.

New NIZKs: improved efficiency or generality. As a first byproduct, we obtain improved NIZKs for some important statements, such as set membership (see Table 5.1) or the language of commitments to points on an elliptic curve[4], as well as new NIZKs for very general classes of statements, such as R1CS, arithmetic constraint satisfaction systems (and thus for NP).

A weaker unified assumption. As the second byproduct of our formal approach, we manage to base all NIZKs in our framework on a slightly weaker form of the extended Kernel Diffie-Hellman assumption, which we call the CED (family of) assumption(s) (for *Computational Extended Determinant* assumption). This turns out to have an important consequence: we show that all instances of our assumption can be based on a single plausible *gap assumption*, which states that solving the kernel Diffie-Hellman assumption in a group $\mathbb{G}_2$ (a well-known search assumption implied in particular by DDH) remains hard, even given a CDH oracle in a *different* group $\mathbb{G}_1$. On top of it, several of our NIZKs (like the one for Boolean Circuit-SAT) are based on a falsifiable CED assumption, while we also show that a slight modification of the NIZK for arithmetic circuits can be also based on a falsifiable variant of CED.

New SPHFs. Eventually, as another byproduct of our methodology, we obtain constructions of Smooth Projective Hash Functions (SPHFs) [GL03] for new languages (SPHFs were the original motivation for introducing the notion of algebraic language, and [Ben+13] gives a generic construction of SPHFs given the matrix description of an algebraic language), including languages describable by efficient algebraic branching programs.

Efficiency, Generality, and Security of our NIZKs

The argument of Couteau and Hartmann [CH20] improves over (even optimized variants of) the standard Groth-Sahai approach on essentially all known algebraic languages. Couteau and Hartmann illustrated this by providing shorter proofs for linear languages (Diffie-Hellman tuples, membership in a linear subspace) and OR proofs (and more generally, membership in t out of n possibly different linear languages), two settings with numerous important applications (to structure-preserving signatures, tightly-secure simulation-sound NIZKs, tightly-mCCA-secure cryptosystems, ring signatures...). Our framework builds upon the Couteau-Hartmann framework, provides a clean mathematical approach to overcoming its main downside (which is that the matrix description of

[4]NIZKs for this type of languages have recently found important applications in blockchain applications, such as the zcash cryptocurrency, see [E K+15] and `https://z.cash/technology/jubjub/`.

Table 5.1: Comparison of set-membership proofs, i..e., NIZKs for $\mathcal{L}_{\mathsf{pk},F}$, where $F(X)$ is univariate, as in Lemmas 5.6.1, 5.9.3 and 5.F.5. The verifier's computation is given in pairings. The Groth-Sahai computation figures are not published and based on our own estimation; hence, we have omitted the computation cost. Note that $|\mathbb{G}_2| = 2|\mathbb{G}_1|$ in common settings. In CHM and new NIZK, $|\mathsf{crs}| = |\mathbb{G}_2|$.

Argument	$\lvert\pi\rvert$	P comp.	V comp.
Previous works			
Optimized GS [Ràf15]	$d\lvert\mathbb{G}_1\rvert + (3d+2)\lvert\mathbb{G}_2\rvert$	-	-
CHM NIZK + [CH20] $(\boldsymbol{\Gamma}, \boldsymbol{\theta})$, Lemma 5.F.5	$(3d-1)\lvert\mathbb{G}_1\rvert + (3d-2)\lvert\mathbb{G}_2\rvert$	$(7d-4)\mathfrak{e}_1 + (3d-1)\mathfrak{e}_2$	$9d-2$
New solutions			
CHM NIZK + new $\boldsymbol{\Gamma}, \boldsymbol{\theta}$, Lemma 5.9.3	$2d\lvert\mathbb{G}_1\rvert + (2d-1)\lvert\mathbb{G}_2\rvert$	$(5d-3)\mathfrak{e}_1 + 4d\mathfrak{e}_2$	$7d-1$
New NIZK, Lemma 5.6.1	$2d\lvert\mathbb{G}_1\rvert + (2d-1)\lvert\mathbb{G}_2\rvert$	$\leq 3d\mathfrak{e}_1 + (4d-2)\mathfrak{e}_2$	$7d-1$

"algebraic languages" must be manually found), and significantly generalizes it. Our framework enjoys most of the benefits of the Couteau-Hartmann framework, such as its ultra-short common random string (a single random group element).

Efficiency. Our framework shines especially as soon as the target language becomes slightly too complex to directly "see" from its description an appropriate and compatible matrix description $\boldsymbol{C}$ of the language; then, we get significant efficiency improvements. We illustrate this on a natural and useful example: set membership proofs for ElGamal ciphertext over $\mathbb{G}_1$ (i.e., the language of ElGamal encryptions of $m \in S$ for some public set S of size d), see Table 5.1. It depicts the complexity of optimized Groth-Sahai proofs, the generic Couteau-Hartmann compilation of Maurer's protocol (denoted CHM) by using the language parameters $(\boldsymbol{\Gamma}, \boldsymbol{\theta})$ provided in [CH20], CHM NIZK for $(\boldsymbol{\Gamma}, \boldsymbol{\theta})$ automatically derived in the current paper from the matrix description $\boldsymbol{C}$, and our new NIZK. On the other hand, our modular approach provides significantly shorter proofs. Taking e.g. $d = 5$, we get a proof about 25% shorter compared to Groth-Sahai. Our approach also significantly improves in terms of computational efficiency. Moreover, since in our approach, we need to only encrypt the data in a single group, as opposed in two groups in the case of (asymmetric-pairing-based) Groth-Sahai, we have three times shorter commitments. In Section 5.8, we also discuss the case of multi-dimensional set membership proofs (where, depending on the structure of the set, our framework can lead to even more significant improvements).

Generality. Our framework also goes way beyond the class of languages naturally handled by Couteau-Hartmann. In particular, we show that our framework directly encompasses *arithmetic constraint satisfaction systems* (aCSPs), i.e., collections of functions $F_1, \ldots, F_\tau$ (called *constraints*) such that each function F_i depends on at most q of its input locations.[5] In particular, this efficiently captures arithmetic circuits, hence all NP languages.[6]

[5]That is, for every $j \in [1, \tau]$ there exist $i_1, \ldots, i_q \in [1, n]$ and $f : \mathbb{F}^q \to \mathbb{F}$ such that $\forall \boldsymbol{\chi} \in \mathbb{F}^n, F_j(\boldsymbol{\chi}) = f(\chi_{i_1}, \ldots, \chi_{i_q})$. Then F is satisfiable if $\forall j, F_j(\boldsymbol{\chi}) = 0$.

[6]Technically, one could always take aCSPs, write them as a circuit satisfiability problem, and embed that into an agebraic language to capture it with the Couteau-Hartmann framework; the point of our

Rank-1 constraints systems (R1CS) are well-known to be powerful, since they capture *compactly* many languages of interest [Gen+13]. They have been widely used in the construction of SNARKs. aCSPs directly extend these simple constraints to arbitrary low-degree polynomial relations. Moving away from R1CS to more expressive constraint systems can potentially be very useful: in many applications of NIZKs with complex languages, an important work is dedicated to finding the "best" R1CS to represent the language. The increased flexibility of being allowed to handle more general constraints can typically allow to achieve a significantly more efficient solution. While systematically revisiting existing works and demonstrating that their R1CS system could be improved using aCSPs would be out of the scope of this paper, we point out that this generalization approach was successfully applied in the past: the work of [HKR19] described a method to go beyond R1CS in "Bulletproof style" random-oracle-based NIZKs (this setting is incomparable to ours, as we focus on NIZKs in the standard model). They show how to handle general quadratic constraints, and demonstrate that this leads to efficiency improvements over Bulletproof on aggregate range proofs. Since aCSPs are even more general, handling any low-degree polynomials, we expect that this representation could lead to significant optimizations for many applications of NIZKs that rely on R1CS representations. However, we are aware of no previous random-oracle-less NIZKs that can handle aCSPs natively.

Furthermore, even in scenarios where R1CS does indeed provide the best possible representation, our framework leads to proofs more compact than Groth-Sahai. We illustrate this on Table 5.2 for the case of general boolean circuits. Here, the standard GOS approach [GOS06] reduces checking each gate of the circuit to checking R1CS equations. When comparing the cost obtained with our framework to the cost achieved by a Groth-Sahai proof (using the optimized variant of [GSW09]), we find that our framework leads to three times smaller commitments, 20% shorter argument, and almost a factor two reduction in computation.

On the non-falsifiability of our assumption. When the algebraic branching program representation of the relation is multivariate, the corresponding matrix description may lead to a NIZK under a non-falsifiable assumption. This might appear at first sight to significantly restrict the interest of our framework: while our NIZKs are typically more efficient than Groth-Sahai, they are usually larger than SNARKs since they grow linearly with (the algebraic branching program representation of) the relation, while SNARKs have size independent of both the relation and the witness. Hence, if we allow non-falsifiable assumptions, wouldn't SNARKs provide a better solution?

We discuss this apparent issue in Section 5.10. First, we identify a large class of important cases where the underlying assumption becomes falsifiable; this includes

framework is that, by capturing this powerful model directly, we can obtain much better efficiency on aCSPs.

Boolean circuits (and thus NP). Second, we provide a general approach to transform *any* NIZK from our framework into NIZKs under a falsifiable assumption, by replacing the underlying commitment scheme by a DLIN-based encryption scheme and double-encrypting certain values. This comes at the cost of increasing the commitment and argument size. Third, we argue that the gap assumption [OP01] underlying our framework is, despite its non-falsifiability, a very natural and plausible assumption; see Section 5.10 for more details. In particular, gap assumptions are generally recognized as much more desirable than knowledge of exponent assumptions. In essence, our assumption says that uncovering structural weaknesses in a group $\mathbb{G}_1$ does not necessarily imply the existence of structural weaknesses in another group $\mathbb{G}_2$; in particular, this assumption trivially holds in the generic bilinear group model (where a CDH oracle in $\mathbb{G}_1$ provides no useful information for breaking any assumption in $\mathbb{G}_2$).

Overall, we view our framework as providing a desirable middle ground between Groth-Sahai (which leads to less efficient NIZKs, but under the standard SXDH assumption) and SNARKs (which lead to more efficient NIZKs in general but require highly non-standard knowledge of exponent assumptions).

Technical Overview

Intuitive overview. At a high level, the Couteau-Hartmann methodology compiles a Σ-protocol for languages of the form $\{\boldsymbol{x} : \exists \boldsymbol{w}, \boldsymbol{\Gamma}(\boldsymbol{x}) \cdot \boldsymbol{w} = \boldsymbol{\theta}(\boldsymbol{x})\}$, where $(\boldsymbol{\Gamma}, \boldsymbol{\theta})$ are linear maps, into a NIZK. This leaves open, however, the tasks of characterizing which languages admit such a representation, *finding* such a representation, and when multiple representations are possible optimizing the choice of the representation. We provide a blueprint for these tasks.

We focus on commit-and-prove languages, a large and useful class of languages. At the heart of our techniques is a general method to convert a set of low-degree polynomial equations $F_i(\boldsymbol{X})$ into a set of "optimized" matrices $\boldsymbol{C_i}(\boldsymbol{X})$ such that $\det(\boldsymbol{C}_i(\boldsymbol{X})) = F_i(\boldsymbol{X})$ with a specific additional structure. We call this matrix a *quasideterminantal (QDR) representation* of the polynomial. Then, we directly construct a compact NIZK proof system for a QDR, using a variant of the Couteau-Hartmann methodology. We prove that the resulting proof system is sound under a CED assumption. Whenever F_i has a polynomial number of roots (e.g., univariate), the corresponding CED assumption is always falsifiable.

Constructing a QDR from a polynomial is a non-trivial task that highly depends on the representation of F_i. We provide a general framework to construct such QDRs from the *algebraic branching program* (ABP [Nis91]) representation of F_i; hence, our framework is especially suited whenever the polynomials have a compact ABP representation. ABP is a powerful model of computation, capturing in particular all log-depth circuits, boolean

branching programs, boolean formulas, logspace circuits, and many more.

Background. The rest of the technical overview requires understanding of some minimal background from algebraic geometry, see [ALO15] for more. Let $\mathbb{F} = \mathbb{Z}_p$ and $\boldsymbol{X} = (X_1, \ldots, X_\nu)$. For a set $\mathcal{F}$ of polynomials in $\mathbb{F}[\boldsymbol{X}]$, let

$$\mathcal{A}(\mathcal{F}) := \{\boldsymbol{\chi} \in \mathbb{F}^\nu : f(\boldsymbol{\chi}) = 0 \text{ for all } f \in \mathcal{F}\}$$

be the *algebraic set defined by* $\mathcal{F}$. A subset $\mathcal{A} \subseteq \mathbb{F}^\nu$ is an *algebraic set* if $\mathcal{A} = \mathcal{A}(\mathcal{F})$ for some $\mathcal{F}$. Given a subset $\mathcal{A}$ of $\mathbb{F}^\nu$, let $\mathcal{I}(\mathcal{A})$ be the ideal of all polynomial functions vanishing on $\mathcal{A}$,

$$\mathcal{I}(\mathcal{A}) := \{f \in \mathbb{F}[\boldsymbol{X}] : f(\boldsymbol{\chi}) = 0 \text{ for all } \boldsymbol{\chi} \in \mathcal{A}\} \ .$$

Since each ideal of $\mathcal{F}[\boldsymbol{X}]$ is finitely generated [ALO15], then so is $\mathcal{I}(\mathcal{A})$, and thus $\mathcal{I}(\mathcal{A}) = \langle F_1, \ldots, F_\tau \rangle$ for some F_i. $\mathcal{I}$ is principal if it is generated by a single polynomial. All univariate ideals are principal. For an ideal $\mathcal{I}$ with generating set $\{F_i\}$, $\mathcal{A}(\mathcal{I}) := \mathcal{A}(\{F_i\})$. We also define $\mathcal{Z}(F) := \mathcal{A}(\{F\})$.

Commit-and-prove NIZKs for algebraic sets. For the sake of concreteness, we focus on commit-and-prove languages where the underlying commitment scheme is the ElGamal encryption scheme; it is easy to extend this approach to any additively homomorphic and perfectly binding algebraic commitment scheme. Let pk be an Elgamal public key and let $\mathcal{A}$ be an algebraic set. We provide a general methodology of constructing a NIZK argument for the language

$$\mathcal{L}_{\mathsf{pk},\mathcal{A}} = \{[\mathbf{ct}]_1 : \exists \boldsymbol{\chi} \text{ such that } \mathsf{Dec}([\mathbf{ct}]_1) = [\boldsymbol{\chi}]_1 \wedge \boldsymbol{\chi} \in \mathcal{A}\}$$

of Elgamal-encryptions of elements of $\mathcal{A}$. We define $\mathcal{L}_{\mathsf{pk},F} := \mathcal{L}_{\mathsf{pk},\mathcal{Z}(F)}$ when we are working with a single polynomial. Assuming $\mathcal{I}(\mathcal{A}) = \langle F_1, \ldots, F_\tau \rangle$, we prove that $\boldsymbol{\chi} \in \mathcal{A}$ by proving that $F_i(\boldsymbol{\chi}) = 0$ for each F_i. The resulting argument system is efficient (probabilistic polynomial-time), assuming that there is

(i) an efficient algorithm (to be run only once) that finds a small generating set $(F_1, \ldots, F_\tau)$ for $\mathcal{I}(\mathcal{A})$ where $\tau = \mathsf{poly}(\lambda)$, and

(ii) an efficient NIZK argument system to show that $F_i(\boldsymbol{\chi}) = 0$ for each F_i.

Note that the NIZK for showing that $F_i(\boldsymbol{\chi}) = 0$ for each i is a simple conjunction of NIZKs for showing for each i that $F_i(\boldsymbol{\chi}) = 0$.

Now, i is a non-cryptographic problem from computational commutative algebra. The classical Buchberger-Möller algorithm [MB82] can find efficiently a finite Gröbner basis $\{F_i\}$ for all algebraic sets $\mathcal{A}$ that have a finite Gröbner basis. Other methods exist, and we will only mention a few. Most importantly, one can relate i to finding efficient arithmetic

circuits and arithmetic constraint satisfaction systems (aCSPs), see Section 5.8.[7] The main technical contribution of our work (on top of the general framework) is to propose an efficient solution to ii.

Constructing a compact proof system for $F(\boldsymbol{\chi}) = 0$. Here, we follow the next blueprint: we construct

(iii) a small matrix $\boldsymbol{C}(\boldsymbol{X})$ (that satisfies some additional properties) of affine maps, such that $\det(\boldsymbol{C}(\boldsymbol{X})) = F(\boldsymbol{X})$, and

(iv) an efficient NIZK argument system for showing that $\det(\boldsymbol{C}(\boldsymbol{\chi})) = 0$ for committed $\boldsymbol{\chi}$.

To solve iv, we build upon the new computational extended determinant assumption (CED). The CED assumption is a relaxation of the ExtKerMDH assumption from [CH20], which itself is a natural generalization of the Kernel Diffie-Hellman assumption. At a high level, CED says that given a matrix in a group $\mathbb{G}_2$, it is hard to find an *extension* of this matrix over $\mathbb{G}_2$, together with a large enough set of linearly independent vectors in $\mathbb{G}_1$ in the kernel of the extended matrix (where $(\mathbb{G}_1, \mathbb{G}_2)$ are groups equipped with an asymmetric pairing). While CED is not falsifiable in general, it can be reduced to a natural gap assumption. The latter reduction does not work with the ExtKerMDH assumption.

Our reduction to the CED assumption proceeds by identifying the matrix $\boldsymbol{C}$, returned by the CED adversary, with the matrix $\boldsymbol{C}(\boldsymbol{X})$ from iii. Intuitively, we construct a reduction that, knowing the Elgamal secret key sk, extracts $[(\boldsymbol{\gamma}\|\boldsymbol{C})(\boldsymbol{\chi})]_1$, where $[\boldsymbol{\chi}]_1 = \mathsf{Dec}_{\mathsf{sk}}([\mathbf{ct}]_1)$, such that $\boldsymbol{C}(\boldsymbol{\chi})$ has full rank iff the soundness adversary cheated, i.e., $F(\boldsymbol{\chi}) \neq 0$. In that case, the reduction can obviously break the CED assumption.

To ensure that the NIZK argument can be constructed, we require that $\boldsymbol{C}$ satisfies two additional properties. Briefly,

(1) $\boldsymbol{C}(\boldsymbol{X})$ is a matrix of affine maps, (to ensure that the matrix is computable from the statement) and

(2) the first column of $\boldsymbol{C}(\boldsymbol{\chi})$ is in the linear span of the remaining columns of the matrix for any $\boldsymbol{\chi} \in \mathcal{Z}(F)$ (a technical condition which ensures that an honest prover can compute the argument).

We say that then $\boldsymbol{C}(\boldsymbol{X})$ is a *quasideterminantal representation (QDR)* of F. We also give some conditions which make it easier to check whether a given matrix is a QDR of F.

[7] There are ample examples of sets $\mathcal{A}$ that have small generating sets (and even small Gröbner bases), which can be found using a variety of standard tricks and methods (e.g. increasing the dimension of the affine space from some $\mathbb{F}^n$ to $\mathbb{F}^{n'}$, $n' > n$, such that the new $n' - n$ "helper variables" make it possible to construct a small Gröbner basis that consists of only small-degree polynomials). We will use such tricks in some of our illustrations and applications.

Building NIZKs from QDRs. Assuming $\boldsymbol{C}(\boldsymbol{X})$ is a QDR of F, we propose a linear-algebraic NIZK argument $\boldsymbol{\Pi}_{\mathsf{nizk}}$ for showing that $\mathbf{x} \in \mathcal{L}_{\mathsf{pk},F}$. We prove that $\boldsymbol{\Pi}_{\mathsf{nizk}}$ is sound under a CED assumption. Importantly, CED is falsifiable if $\mathcal{A} = \mathcal{A}(F)$ has a polynomial number of elements. Otherwise, CED is in general non-falsifiable (except in some relevant cases, see Section 5.10), but belongs to the class of "inefficient-challenger" assumptions (usually considered more realistic than knowledge assumptions, see [Pas13]). Furthermore, CED can be reduced to a single, natural *gap assumption*: the hardness of breaking DDH in a group $\mathbb{G}_2$ given a CDH oracle in a different group $\mathbb{G}_1$. We refer to 5.10 for more details.

Constructing QDRs. The remaining, *highly non-trivial*, problem is to construct a QDR of F, such that the constructed NIZK argument is efficient. In the rest of the paper, we study this problem.

First, we propose a general framework to construct NIZK arguments for $\mathcal{L}_{\mathsf{pk},F}$ where $F(\boldsymbol{\chi})$ can be computed by an efficient *algebraic branching program*. Let Π be an ABP that computes F, with the node set V and the edge set E, and let $\ell = |V| - 1$. Given the methodology of [IK00; IK02], one can represent Π as an $\ell \times \ell$ matrix $\mathsf{IK}(\boldsymbol{X})$, such that $\det(\mathsf{IK}(\boldsymbol{X}))$ is equal to the output of the ABP. We show that such $\mathsf{IK}(\boldsymbol{X})$ is a QDR. Thus, we obtain an efficient computationally-sound NIZK for $\mathcal{L}_{\mathsf{pk},F}$ under a CED assumption.

Applications. We consider several natural applications of our framework.

Univariate polynomials. Given a univariate polynomial $F(X) = \prod(X - \xi_i)$ of degree-d, for different roots ξ_i, we construct a simple matrix $\boldsymbol{C}(X)$. The resulting NIZK argument is about 30% shorter and 20% more computationally efficient than the set membership proof that stems from [CH20, Section C]; see the comparison in Table 5.1.

Commitments to points on an elliptic curve. We construct a NIZK argument to prove that the committed point (X, Y) belongs to the given elliptic curve $Y^2 = X^3 + aX + b$. Such NIZK proofs are popular in cryptocurrency applications, [Ben+14]. The construction of $\boldsymbol{C}(X, Y)$ is motivated by a classical algebraic-geometric (possibility) result that for any homogeneous cubic surface $F(X, Y, Z)$, there exists a 3×3 matrix of affine maps that has $F(X, Y, Z)$ as its determinant [Dic21; Bea00].

OR proofs. In Section 5.6, we look at the special case of OR proofs and study three instantiations of our general protocol to OR arguments. We discuss the advantages and downsides of each.

Non-Principal Ideals. Importantly, in Section 5.8, we capture the very general scenario where $\mathcal{I}(\mathcal{A})$ has a "nice-looking" generating set $(F_1, \ldots, F_\tau)$ (i.e. τ is small and each polynomial has a small degree). Some cryptographically important examples include arithmetic circuits, R1CS, Boolean circuits, and arithmetic constraint satisfaction systems. Thus, we obtain efficient NIZKs for NP.

5.2 Preliminaries

For a matrix $\boldsymbol{A} \in \mathbb{Z}_p^{n\times n}$ and $i \in [1,n]$, let $\boldsymbol{C}_{(i,1)}$ be the submatrix obtained from $\boldsymbol{C}$ by removing the ith row and the first column.

Cryptography. A bilinear group generator $\mathsf{Pgen}(1^\lambda)$ returns $\mathsf{p} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2)$, where $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ are three additive cyclic groups of prime order p, $[1]_\iota$ is a generator of $\mathbb{G}_\iota$ for $\iota \in \{1,2,T\}$ with $[1]_T = \hat{e}([1]_1,[1]_2)$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate efficiently computable bilinear pairing. We require the bilinear pairing to be Type-3 [DGP08], that is, we assume that there is no efficient isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. We use the additive implicit notation of [Esc+13], that is, we write $[a]_\iota$ to denote $a[1]_\iota$ for $\iota \in \{1,2,T\}$. We denote $\hat{e}([a]_1,[b]_2)$ by $[a]_1 \bullet [b]_2$. Thus, $[a]_1 \bullet [b]_2 = [ab]_T$. We freely use the bracket notation together with matrix notation; for example, if $\boldsymbol{AB} = \boldsymbol{C}$ then $[\boldsymbol{A}]_1 \bullet [\boldsymbol{B}]_2 = [\boldsymbol{C}]_T$. We also define

$$[\boldsymbol{A}]_2 \bullet [\boldsymbol{B}]_1 := ([\boldsymbol{B}]_1^\top \bullet [\boldsymbol{A}]_2^\top)^\top = [\boldsymbol{AB}]_T \ .$$

Let $\mathcal{P}_\nu := \{[a_0]_1 + \sum_{i=0}^{\nu} [a_i]_1 X_i : a_i \in \mathbb{Z}_p \text{ for } i \in [0,\nu]\} \subset \mathbb{G}_1[\boldsymbol{X}]$ be the set of linear multivariate polynomials over $\mathbb{G}_1$ in ν variables.

Algebraic languages [CC18; CH20] are parameterized languages of the shape $\mathcal{L}_{\boldsymbol{\Gamma},\boldsymbol{\theta}} = \{\mathtt{x} : \exists \mathtt{w}, \boldsymbol{\Gamma}(\mathtt{x}) \cdot \mathtt{w} = \boldsymbol{\theta}(\mathtt{x})\}$, where $\mathtt{x}$ is the input, $\mathtt{w}$ is the witness, $\boldsymbol{\Gamma}$ and $\boldsymbol{\theta}$ are affine maps, such that $\mathtt{x}$ and $\boldsymbol{\theta}(\mathtt{x})$ are vectors, and $\boldsymbol{\Gamma}(\mathtt{x})$ is a matrix. One can construct Gennaro-Lindell smooth projective hash functions (GL-SPHFs [GL03; Ben+13; Ben16]) for all algebraic languages.

Let $\mathsf{k} \in \{1,2,\ldots\}$ be a small parameter related to the matrix distribution. In the case of asymmetric pairings, usually $\mathsf{k} = 1$. Let $\mathcal{D}_{\ell\mathsf{k}}$ be a probability distribution over $\mathbb{Z}_p^{\ell\times\mathsf{k}}$, where $\ell > \mathsf{k}$. We denote $\mathcal{D}_{\mathsf{k}+1,\mathsf{k}}$ by $\mathcal{D}_\mathsf{k}$. We use the matrix distribution, $\mathcal{L}_1$, defined as the distribution over matrices $\left(\begin{smallmatrix}1\\a\end{smallmatrix}\right)$, where $a \leftarrow\!\!\$\, \mathbb{Z}_p$.

In the Elgamal encryption scheme [ElG84], the public key is $\mathsf{pk} = [1\|\mathsf{sk}]_1$, and

$$\mathsf{Enc}_{\mathsf{pk}}(m;r) = (r[1]_1 \| m[1]_1 + r[\mathsf{sk}]_1) \ .$$

To decrypt, one computes $[m]_1 = \mathsf{Dec}_{\mathsf{sk}}([\boldsymbol{c}]_1) \leftarrow -\mathsf{sk}[c_1]_1 + [c_2]_1$. In what follows, we denote $[\boldsymbol{c}]_1 = \mathsf{Enc}(m;r)$ for a fixed public key $\mathsf{pk} = [1\|\mathsf{sk}]_1$. Elgamal's IND-CPA security is based on $\mathcal{L}_1$-KerMDH, that is, DDH.

The DLIN cryptosystem [BBS04] is less efficient than Elgamal, with the ciphertext consisting of 3 group elements instead of 2. However, it remains secure in the case of symmetric pairings. Its IND-CPA security is based on $\mathcal{L}_2$-KerMDH, that is, DLIN [BBS04]. Briefly,

$$[\boldsymbol{c}]_\iota \leftarrow \mathsf{Enc}_\iota(\chi;r_1,r_2) := (\chi\|r_1\|r_2) \begin{bmatrix} 0 & 0 & 1 \\ \mathsf{sk}_1 & 0 & 1 \\ 0 & \mathsf{sk}_2 & 1 \end{bmatrix}_\iota = [r_1\mathsf{sk}_1 \| r_2\mathsf{sk}_2 \| \chi + r_1 + r_2]_\iota \in \mathbb{G}_\iota^3$$

for public key $\mathsf{pk}_\iota = [1\|\mathsf{sk}_1\|\mathsf{sk}_2]_\iota$ and randomiser (r_1,r_2). The decryption formula is $[\chi]_\iota \leftarrow -1/\mathsf{sk}_1 \cdot [c_1]_\iota - 1/\mathsf{sk}_2 \cdot [c_2]_\iota + [c_3]_\iota$.

The following Extended Kernel Diffie-Hellman assumption ExtKerMDH [CH20] generalizes the well-known KerMDH assumption [MRV16]. (Section 5.A defines KerMDH.) We also define in parallel a new, slightly weaker version of this assumption, CED (*computational extended determinant*).

Definition 5.2.1 ($\mathcal{D}_\mathsf{k}$-$(\ell-1)$-ExtKerMDH)**.** Let $\ell, \mathsf{k} \in \mathbb{N}$, and $\mathcal{D}_\mathsf{k}$ be a matrix distribution. The $\mathcal{D}_\mathsf{k}$-$(\ell-1)$-ExtKerMDH assumption holds in $\mathbb{G}_\iota$ relative to Pgen, if for all PPT adversaries $\mathcal{A}$, the following probability is negligible:

$$\Pr\left[\begin{array}{c|c} \boldsymbol{\delta} \in \mathbb{Z}_p^{(\ell-1)\times\mathsf{k}} \wedge \boldsymbol{\gamma} \in \mathbb{Z}_p^{\ell\times\mathsf{k}} \wedge \boldsymbol{C} \in \mathbb{Z}_p^{\ell\times\ell} \wedge & \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda), [\boldsymbol{D}]_\iota \leftarrow\!\!\$\, \mathcal{D}_\mathsf{k}, \\ (\boldsymbol{\gamma}\|\boldsymbol{C})\left(\begin{smallmatrix}\boldsymbol{D}\\ \boldsymbol{\delta}\end{smallmatrix}\right) = \mathbf{0} \wedge \mathrm{rk}(\boldsymbol{\gamma}\|\boldsymbol{C}) \geq \ell & ([\boldsymbol{\gamma}\|\boldsymbol{C}]_{3-\iota}, [\boldsymbol{\delta}]_\iota) \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{D}]_\iota) \end{array}\right].$$

We define $\mathcal{D}_\mathsf{k}$-$(\ell-1)$-CED analogously, except that we change the condition $\mathrm{rk}(\boldsymbol{\gamma}\|\boldsymbol{C}) \geq \ell$ to $\mathrm{rk}(\boldsymbol{C}) = \ell$.

CED is *weaker* than ExtKerMDH since a successful adversary has to satisfy a stronger condition ($\mathrm{rk}(\boldsymbol{C}) \geq \ell$ instead of $\mathrm{rk}(\boldsymbol{\gamma}\|\boldsymbol{C}) \geq \ell$). Formally:

Lemma 5.2.2. *Let ℓ, k, and $\mathcal{D}_\mathsf{k}$ be as in Definition 5.2.1. If $\mathcal{D}_\mathsf{k}$-$(\ell-1)$-ExtKerMDH holds, then $\mathcal{D}_\mathsf{k}$-$(\ell-1)$-CED holds.*

Proof. Let $\mathcal{A}$ be an adversary that breaks $\mathcal{D}_\mathsf{k}$-$(\ell-1)$-CED with probability ε. We construct the following adversary $\mathcal{B}$ that breaks $\mathcal{D}_\mathsf{k}$-$(\ell-1)$-ExtKerMDH:

$\mathcal{B}(\mathsf{p}, [\boldsymbol{D}]_\iota)$
$([\boldsymbol{\gamma}\|\boldsymbol{C}]_{3-\iota}, [\boldsymbol{\delta}]_\iota) \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{D}]_\iota)$;
return $([\boldsymbol{\gamma}\|\boldsymbol{C}]_{3-\iota}, [\boldsymbol{\delta}]_\iota)$;

If $\mathcal{A}$ succeeds, then by Definition 5.2.1, $(\boldsymbol{\gamma}\|\boldsymbol{C})\left(\begin{smallmatrix}\boldsymbol{D}\\ \boldsymbol{\delta}\end{smallmatrix}\right) = \mathbf{0}$ and $\mathrm{rk}(\boldsymbol{\gamma}) \geq \ell$. However, if $\mathrm{rk}(\boldsymbol{\gamma}) \geq \ell$ then also clearly $\mathrm{rk}(\boldsymbol{\gamma}\|\boldsymbol{C}) \geq \ell$. Thus, $\mathcal{B}$ succeeds with probability $\geq \varepsilon$. □

CED suffices for the security of all NIZK arguments of the current paper. Moreover, in Section 5.10, we reduce CED to a gap assumption. It seems that ExtKerMDH cannot be reduced to the same assumption. Finally, CED is a natural assumption since we always care about $\mathrm{rk}(\boldsymbol{C})$ and not $\mathrm{rk}(\boldsymbol{\gamma}\|\boldsymbol{C}) \geq \ell$.

Despite the general definition, in the rest of the paper (following [CH20]), we will be only concerned with the case $\mathsf{k} = 1$ and $\mathcal{D}_\mathsf{k} = \mathcal{L}_1$.

NIZK Arguments. An adaptive NIZK Π for a family of language distribution $\{\mathcal{D}_\mathsf{p}\}_\mathsf{p}$ consists of five probabilistic algorithms:

(1) $\mathsf{Pgen}(1^\lambda)$: generates public parameters p that fix a distribution $\mathcal{D}_\mathsf{p}$.

(2) $\mathsf{Kgen}(\mathsf{p})$: generates a CRS crs and a trapdoor $\mathtt{td}$. For simplicity of notation, we assume that any group parameters are implicitly included in the CRS. We often denote the sequence "$\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$; $(\mathsf{crs}, \mathtt{td}) \leftarrow \mathsf{Kgen}(\mathsf{p})$" by $(\mathsf{p}, \mathsf{crs}, \mathtt{td}) \leftarrow \mathsf{Kgen}(1^\lambda)$.

(3) $\mathsf{P}(\mathsf{crs}, \mathtt{lpar}, \mathtt{x}, \mathtt{w})$: given a language description $\mathtt{lpar} \in \mathcal{D}_{\mathsf{p}}$ and a statement $\mathtt{x}$ with witness $\mathtt{w}$, outputs a proof π for $\mathtt{x} \in \mathcal{L}_{\mathtt{lpar}}$.

(4) $\mathsf{V}(\mathsf{crs}, \mathtt{lpar}, \mathtt{x}, \pi)$. On input of a CRS, a language description $\mathtt{lpar} \in \mathcal{D}_{\mathsf{p}}$, a statement and a proof, accepts or rejects the proof.

(5) $\mathsf{Sim}(\mathsf{crs}, \mathtt{td}, \mathtt{lpar}, \mathtt{x})$. Given a CRS, the trapdoor $\mathtt{td}$, $\mathtt{lpar} \in \mathcal{D}_{\mathsf{p}}$, and a statement $\mathtt{x}$, outputs a simulated proof for the statement $\mathtt{x} \in \mathcal{L}_{\mathtt{lpar}}$.

Note that the CRS does not depend on the language distribution or language parameters, i.e. we define fully adaptive NIZKs for language distributions. The following properties need to hold for a NIZK argument.

A proof system Π for $\{\mathcal{D}_{\mathsf{p}}\}_{\mathsf{p}}$ is *perfectly complete*, if

$$\Pr\left[\mathsf{V}(\mathsf{crs}, \mathtt{lpar}, \mathtt{x}, \pi) = 1 \;\middle|\; \begin{array}{c}(\mathsf{p}, \mathsf{crs}, \mathtt{td}) \leftarrow\!\$\, \mathsf{K}_{\mathsf{crs}}(1^\lambda); \mathtt{lpar} \in \mathrm{Supp}(\mathcal{D}_{\mathsf{p}});\\ (\mathtt{x}, \mathtt{w}) \in \mathcal{R}_{\mathtt{lpar}}; \pi \leftarrow\!\$\, \mathsf{P}(\mathsf{crs}, \mathtt{lpar}, \mathtt{x}, \mathtt{w})\end{array}\right] = 1$$

A proof system Π for $\{\mathcal{D}_{\mathsf{p}}\}_{\mathsf{p}}$ is *computationally sound*, if for every efficient $\mathcal{A}$,

$$\Pr\left[\begin{array}{c}\mathsf{V}(\mathsf{crs}, \mathtt{lpar}, \mathtt{x}, \pi) = 1\\ \wedge \mathtt{x} \notin \mathcal{L}_{\mathtt{lpar}}\end{array} \;\middle|\; \begin{array}{c}(\mathsf{p}, \mathsf{crs}, \mathtt{td}) \leftarrow\!\$\, \mathsf{K}_{\mathsf{crs}}(1^\lambda);\\ \mathtt{lpar} \in \mathrm{Supp}(\mathcal{D}_{\mathsf{p}}); (\mathtt{x}, \pi) \leftarrow \mathcal{A}(\mathsf{crs}, \mathtt{lpar})\end{array}\right] \approx 0$$

with the probability taken over $\mathsf{K}_{\mathsf{crs}}$.

Π for $\{\mathcal{D}_{\mathsf{p}}\}_{\mathsf{p}}$ is *perfectly zero-knowledge*, if for all λ, all $(\mathsf{p}, \mathsf{crs}, \mathtt{td}) \in \mathrm{Supp}(\mathsf{K}_{\mathsf{crs}}(1^\lambda))$, all $\mathtt{lpar} \in \mathrm{Supp}(\mathcal{D}_{\mathsf{p}})$ and all $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}_{\mathtt{lpar}}$, the distributions $\mathsf{P}(\mathsf{crs}, \mathtt{lpar}, \mathtt{x}, \mathtt{w})$ and $\mathsf{Sim}(\mathsf{crs}, \mathtt{td}, \mathtt{lpar}, \mathtt{x})$ are identical.

Σ-Protocols. A Σ-protocol [CDS94] is a public-coin, three-move interactive proof between a prover P and a verifier V for a relation $\mathcal{R}$, where the prover sends an initial message a, the verifier responds with a random $e \leftarrow\!\$\, \mathbb{Z}_p$ and the prover concludes with a message z. Lastly, the verifier outputs 1, if it accepts and 0 otherwise. In this work we are concerned with three properties of a Σ-protocol: completeness, optimal soundness and honest-verifier zero-knowledge.

CH compilation. Couteau and Hartmann [CH20] compile Σ-protocols to NIZKs in the CRS model for algebraic languages by letting $[e]_2$ be the CRS. The basic Couteau and Hartmann compilation is for a Σ-protocol, inspired by [Mau09], for algebraic languages. We will describe it in Section 5.9.

5.3 Quasideterminantal Representations

Next, we define quasideterminantal representations (QDRs) $\boldsymbol{C}(\boldsymbol{X})$ of a polynomial $F(\boldsymbol{X})$. We prove a technical lemma in Section 5.3 which shows how one can check whether a concrete matrix $\boldsymbol{C}(\boldsymbol{X})$ is a QDR of F. We use this definition in Section 5.4, where, given

a QDR $\boldsymbol{C}(\boldsymbol{X})$, we define the NIZK argument for the associated language $\mathcal{L}_{\mathsf{pk},F}$ (defined in Equation (5.1)), and prove its security.

We first define the class of languages we are interested in. Initially, we are interested in the case where $\mathcal{A} = \mathcal{A}(\{F\})$ for a single polynomial F. Fix $\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$. For a fixed Elgamal public key pk, let $\mathtt{lpar} := (\mathsf{pk}, F)$. (Implicitly, $\mathtt{lpar}$ also contains p.) Let $[\mathbf{ct}]_1 = \mathsf{Enc}([\boldsymbol{\chi}]_1; \boldsymbol{r}) = (\mathsf{Enc}([\chi_i]_1; r_i))_i$. We use freely the notation $F(\mathsf{Dec}([\mathbf{ct}]_1)) = F([\boldsymbol{\chi}]_1) = [F(\boldsymbol{\chi})]_1$. In Section 5.4, we describe a general technique that results both in efficient NIZK arguments for languages

$$\mathcal{L}_{\mathsf{pk},F} = \{[\mathbf{ct}]_1 : \exists \boldsymbol{\chi} \text{ such that } \mathsf{Dec}([\mathbf{ct}]_1) = [\boldsymbol{\chi}]_1 \wedge \boldsymbol{\chi} \in \mathcal{Z}(F)\} \ . \tag{5.1}$$

For example, if $F(X) = X^2 - X$, then $\mathcal{L}_{\mathsf{pk},F}$ corresponds to the language of all Elgamal encryptions of Boolean values under the fixed public key pk.

Intuition. To motivate the definition of QDRs, we first explain the intuition behind the new NIZK argument. Recall from Definition 5.2.1 that an adversary breaks the $\mathcal{L}_1$-$(\ell-1)$-CED assumption if, given $[\boldsymbol{D}]_2 = \left[\begin{smallmatrix}1\\ e\end{smallmatrix}\right]_2 \leftarrow\!\!\$\, \mathcal{L}_1$ (i.e., $e \leftarrow\!\!\$\, \mathbb{Z}_p$), he returns $([\boldsymbol{\gamma}\|\boldsymbol{C}]_1 \in \mathbb{G}_1^{\ell\times(\ell+1)}, [\boldsymbol{\delta}]_2 \in \mathbb{G}_2^{(\ell-1)\times 1})$, such that $\mathrm{rk}(\boldsymbol{C}) \geq \ell$ and

$$\boldsymbol{\gamma} + \boldsymbol{C}\left(\begin{smallmatrix}e\\ \boldsymbol{\delta}\end{smallmatrix}\right) = \mathbf{0}. \tag{5.2}$$

Following [CH20], in our arguments $[e]_2$ (i.e., $[\boldsymbol{D}]_2$) is given in the CRS and $[\boldsymbol{\delta}]_2$ is chosen by the prover. More precisely, the prover sends $\mathsf{Enc}([\boldsymbol{\gamma}\|\boldsymbol{C}]_1)$ and $[\boldsymbol{\delta}]_2$ (together with some elements that make it possible to verify that Equation (5.2) holds using encrypted values) to the verifier.

The matrix $\boldsymbol{C}$ must have full rank whenever the prover cheats, i.e. $F(\boldsymbol{\chi}) \neq 0$. We achieve this by requiring that $\det(\boldsymbol{C}(\boldsymbol{X})) = F(\boldsymbol{X})$. Then, $\mathrm{rk}(\boldsymbol{C}) = \ell$.

We guarantee that $\boldsymbol{C}$ is efficiently computable by requiring that $\boldsymbol{C}(\boldsymbol{X})$ is a matrix of affine maps, and $[\boldsymbol{C}]_1 = [\boldsymbol{C}(\boldsymbol{\chi})]_1$ for $[\boldsymbol{\chi}]_1 = \mathsf{Dec}([\mathsf{ct}]_1)$. This also minimizes communication since each element of $\mathsf{Enc}([\boldsymbol{C}(\boldsymbol{\chi})]_1)$ can be recomputed from $\mathsf{Enc}([\boldsymbol{\chi}]_1)$ by using the homomorphic properties of Elgamal.

On the other hand, assume that the prover is not honest (i.e., $\det(\boldsymbol{C}(\boldsymbol{\chi})) = F(\boldsymbol{\chi}) \neq 0$) but managed to compute $\mathsf{Enc}([\boldsymbol{\gamma}]_1)$ and $[\boldsymbol{\delta}]_2$ accepted by the verifier. Assume that the reduction knows sk (the language trapdoor). Then, the reduction obtains $[\boldsymbol{\chi}]_1$ by decryption and recomputes $[\boldsymbol{C}(\boldsymbol{\chi})]_1$. Since $\det(\boldsymbol{C}(\boldsymbol{\chi})) \neq 0$ but the verifier accepts (i.e., Equation (5.2)), then one can break the CED assumption by returning $[(\boldsymbol{\gamma}\|\boldsymbol{C})(\boldsymbol{\chi})]_1$ and $[\boldsymbol{\delta}]_2$.

Definition

We now define quasideterminantal representations (QDRs) $\boldsymbol{C}(\boldsymbol{X})$ of polynomial F. QDRs are related to the well-known notion of determinantal representation from algebraic geometry, see Section 5.B for a discussion.

Definition 5.3.1 (Quasideterminantal Representation (QDR))**.** Let $F(\boldsymbol{X}) \in \mathbb{Z}_p[\boldsymbol{X}]$ be a ν-variate polynomial. Let $\ell \geq 1$ be an integer. A matrix $\boldsymbol{C}(\boldsymbol{X}) = (C_{ij}(\boldsymbol{X})) \in \mathbb{Z}_p[\boldsymbol{X}]^{\ell \times \ell}$ is *a QDR of* F, if the following requirements hold. Here, $\boldsymbol{C}(\boldsymbol{X}) = (\boldsymbol{h} \| \boldsymbol{T})(\boldsymbol{X})$, where $\boldsymbol{h}(\boldsymbol{X})$ is a column vector.

Affine map: For each i and j, $C_{ij}(\boldsymbol{X}) = \sum_{k=1}^{\nu} P_{kij} X_k + Q_{ij}$, for public $P_{kij}, Q_{ij} \in \mathbb{Z}_p$, is an affine map.

F-rank: $\det(\boldsymbol{C}(\boldsymbol{X})) = F(\boldsymbol{X})$.

First column dependence: For any $\boldsymbol{\chi} \in \mathcal{Z}(F)$, $\boldsymbol{h}(\boldsymbol{\chi}) \in \mathrm{colspace}(\boldsymbol{T}(\boldsymbol{\chi}))$.

The quasideterminantal complexity $\mathsf{qdc}(F)$ of F is the smallest QDR size of F. (Clearly, $\mathsf{qdc}(F) \geq \deg(F)$.)

For example, $\boldsymbol{C}(X) = \left(\begin{smallmatrix} 0 & X \\ X-1 & 1-X \end{smallmatrix}\right)$ is a QDR of $F(X) = X(X-1)$. The first column dependence property follows since $\left(\begin{smallmatrix} 0 \\ \chi-1 \end{smallmatrix}\right) = \left(\begin{smallmatrix} \chi \\ 1-\chi \end{smallmatrix}\right) w$ iff $(\chi, w) = (0, -1)$ or $(\chi, w) = (1, 0)$, i.e., $\chi \in \mathcal{Z}(F)$. On the other hand, $\boldsymbol{C}(X) = \left(\begin{smallmatrix} X & 0 \\ 0 & X-1 \end{smallmatrix}\right)$ is not a QDR (of the same F) since $\left(\begin{smallmatrix} \chi \\ 0 \end{smallmatrix}\right) = \left(\begin{smallmatrix} 0 \\ \chi-1 \end{smallmatrix}\right) w$ iff $(\chi, w) = (0, 0)$.

The first column dependence property is nicely connected to a computational requirement we need for our NIZK. However, it can be difficult to check whether a given matrix satisfies this condition. We now give two alternative conditions that imply the first column dependence property, and which are easier to check.

Lemma 5.3.2. *Suppose a matrix $\boldsymbol{C}$ satisfies the affine map and F-rank properties. If it in addition satisfies one of the following properties, it also satisfies the first column dependence property.*

(1) High right rank: For any $\boldsymbol{\chi} \in \mathbb{Z}_p^{\nu}$, $\mathrm{rk}(\boldsymbol{T}(\boldsymbol{\chi})) = \ell - 1$.

(2) Invertible right-submatrix: there exists i, s.t. $\det(\boldsymbol{C}_{(i,1)}(\boldsymbol{\chi})) \neq 0$ for any $\boldsymbol{\chi}$.

Proof. **(1).** Consider any $\boldsymbol{\chi} \in \mathcal{Z}(F)$. By the F-rank property, $\det(\boldsymbol{C}(\boldsymbol{\chi})) = 0$ and thus $\mathrm{rk}(\boldsymbol{C}(\boldsymbol{\chi})) \leq \ell - 1$. Suppose $\boldsymbol{h}(\boldsymbol{\chi}) \notin \mathrm{colspace}(\boldsymbol{T}(\boldsymbol{\chi}))$. Then $\mathrm{rk}(\boldsymbol{C}(\boldsymbol{\chi})) > \mathrm{rk}(\boldsymbol{T}(\boldsymbol{\chi}))$. By the high right rank property, $\ell - 1 \geq \mathrm{rk}(\boldsymbol{C}(\boldsymbol{\chi})) > \mathrm{rk}(\boldsymbol{T}(\boldsymbol{\chi})) = \ell - 1$, which is a contradiction. Thus, $\boldsymbol{h}(\boldsymbol{\chi}) \in \mathrm{colspace}(\boldsymbol{T}(\boldsymbol{\chi}))$.

(2). From the invertible right-submatrix property, $\mathrm{rk}(\boldsymbol{C}_{(i,1)}(\boldsymbol{\chi})) = \ell - 1$, and thus $\mathrm{rk}(\boldsymbol{T}(\boldsymbol{\chi})) = \ell - 1$. □

E.g., any matrix $\boldsymbol{C}(\boldsymbol{X})$ that contains non-zero elements on its upper 1-diagonal and only 0's above the upper 1-diagonal is automatically a QDR of $F(\boldsymbol{X}) := \det(\boldsymbol{C}(\boldsymbol{X}))$. See Sections 5.5 and 5.6 for more.

Corollaries

The affine map property is needed since we use a homomorphic cryptosystem which makes it possible to compute

$$\mathsf{Enc}([C_{ij}(\boldsymbol{\chi})]_1) = \sum_{k=1}^{\nu} P_{kij}\mathsf{Enc}([\chi_k]_1) + Q_{ij}\mathsf{Enc}([1]_1)$$

given only $\mathsf{Enc}([\boldsymbol{\chi}]_1)$. The F-rank property follows directly from the definition of CED. The first column dependence property, guarantees that the QDR $\boldsymbol{C}(\boldsymbol{X})$ satisfies the following two properties, required later:

Efficient prover: There exist two PPT algorithms that we later explicitly use in the new NIZK argument (see Figure 5.2) for $\mathcal{L}_{\mathsf{pk},F}$. First, $\mathsf{comp}_1(\mathsf{p}, \boldsymbol{\chi}, \boldsymbol{C}(\boldsymbol{X}))$, that computes $[\boldsymbol{\gamma}]_1$ and a state st. Second, $\mathsf{comp}_2(st, [e]_2)$, that computes $[\boldsymbol{\delta}]_2$. We require that if $F(\boldsymbol{\chi}) = 0$, then $([\boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$ satisfy Equation (5.2). We denote the sequential process $([\boldsymbol{\gamma}]_1, st) \leftarrow \mathsf{comp}_1(\mathsf{p}, \boldsymbol{\chi}, \boldsymbol{C}(\boldsymbol{X}))$, $[\boldsymbol{\delta}]_2 \leftarrow \mathsf{comp}_2(st, [e]_2)$ by $([\boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2) \leftarrow \mathsf{comp}(\mathsf{p}, [e]_2, \boldsymbol{\chi}, \boldsymbol{C}(\boldsymbol{X}))$.

Zero-knowledge: For $([\boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2) \leftarrow \mathsf{comp}(\mathsf{p}, [e]_2, \boldsymbol{\chi}, \boldsymbol{C}(\boldsymbol{X}))$, $\boldsymbol{\delta}$ is uniformly random. This requirement is needed for the zero-knowledge property of the resulting NIZK argument.

To be able to construct an efficient Σ-protocol for $\mathcal{L}_{\mathsf{pk},F}$, we need to replace the efficient prover assumption with the following assumption.

Efficient prover over integers: as the "efficient prover" requirement, but one uses e everywhere instead of $[e]_2$, and $\boldsymbol{\delta}$ instead of $[\boldsymbol{\delta}]_2$.

In all our instantiations, the two variations of comp are related as follows: $\mathsf{comp}(\mathsf{p}, [e]_2, \boldsymbol{\chi}, \boldsymbol{C}(\boldsymbol{X}))$ is the same as $\mathsf{comp}(\mathsf{p}, e, \boldsymbol{\chi}, \boldsymbol{C}(\boldsymbol{X}))$ but applies an additional $[\cdot]_2$ to some of the variables.

Remark. We will explicitly need the independence of $[\boldsymbol{\gamma}]_1$ from $[e]_2$ for Σ-protocols and thus for CH-compilation. It is not a priori clear if it is needed for NIZK arguments in general. However, if $\boldsymbol{\gamma} = f(e)$ for some non-constant affine map f, then one cannot efficiently compute $[\boldsymbol{\gamma}]_1$ given only $[e]_2$, since we rely on type-III pairings and those two values belong to different source groups. Thus, independence of $[\boldsymbol{\gamma}]_1$ from $[e]_2$ seems inherent in the case of type-III pairings.

Lemma 5.3.3. *Assume F is as in Definition 5.3.1 and that $\boldsymbol{C}(\boldsymbol{X})$ is a QDR of F. Then*

(1) $\boldsymbol{C}$ has the efficient-prover property.

(2) $\boldsymbol{C}$ has the zero-knowledge property.

$\mathsf{comp}_1(\mathsf{p}, \boldsymbol{\chi}, \boldsymbol{C}(\boldsymbol{X}))$:	$\mathsf{comp}_2(st, \psi(e))$:
Write $\boldsymbol{C}(\boldsymbol{\chi}) = (\boldsymbol{h}\|\boldsymbol{T})(\boldsymbol{\chi})$; $\boldsymbol{y} \leftarrow\$ \mathbb{Z}_p^{\ell-1}$; $\boldsymbol{\gamma} \leftarrow \boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{y}$; $st \leftarrow (\mathsf{p}, \boldsymbol{\chi}, \boldsymbol{C}(\boldsymbol{X}); \boldsymbol{y})$; **return** $([\boldsymbol{\gamma}]_1, st)$;	Write $\boldsymbol{C}(\boldsymbol{\chi}) = (\boldsymbol{h}\|\boldsymbol{T})(\boldsymbol{\chi})$; Compute $\boldsymbol{w}$ such that $\boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{w} = \boldsymbol{h}(\boldsymbol{\chi})$; $\psi(\boldsymbol{\delta}) \leftarrow -(\boldsymbol{w}\psi(e) + \psi(\boldsymbol{y}))$; **return** $\psi(\boldsymbol{\delta})$;

Figure 5.1: comp_i algorithms assuming $\boldsymbol{h}(\boldsymbol{\chi}) \in \mathrm{colspace}(\boldsymbol{T}(\boldsymbol{\chi}))$. Here, $\psi = id$ in the case of the Σ-protocol, and $\psi = [\cdot]_2$ in the case of the NIZK argument.

Proof. Recalling $\boldsymbol{C}(\boldsymbol{X}) = (\boldsymbol{h}\|\boldsymbol{T})(\boldsymbol{X})$, we rewrite Equation (5.2) as

$$\boldsymbol{\gamma} + \boldsymbol{h}(\boldsymbol{X})e + \boldsymbol{T}(\boldsymbol{X})\boldsymbol{\delta} = \boldsymbol{0} \ . \tag{5.3}$$

Assume $\boldsymbol{C}(\boldsymbol{X})$ is a QDR of F. From the first column dependence property, we get that for any $\boldsymbol{\chi} \in \mathcal{Z}(F)$, there exists a $\boldsymbol{w}$, such that $\boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{w} = \boldsymbol{h}(\boldsymbol{\chi})$. Thus for such $\boldsymbol{\chi}$, Equation (5.3) holds iff

$$\boldsymbol{\gamma} + \boldsymbol{T}(\boldsymbol{\chi})(\boldsymbol{w}e + \boldsymbol{\delta}) = \boldsymbol{\gamma} + \boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{w}e + \boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{\delta} = \boldsymbol{0} \ .$$

This gives rise to the following algorithm to compute $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$. In comp_1, one samples $\boldsymbol{y} \leftarrow\$ \mathbb{Z}_p^{\ell-1}$, and outputs $\boldsymbol{\gamma} \leftarrow \boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{y}$. In comp_2, one solves $\boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{w} = \boldsymbol{h}(\boldsymbol{\chi})$ for $\boldsymbol{w}$, and sets $\boldsymbol{\delta} \leftarrow -(\boldsymbol{w}e + \boldsymbol{y})$. Clearly, $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ satisfy Equation (5.2), and $\boldsymbol{\gamma}$ is computed independently of e. Thus, the efficient prover property holds. Since $\boldsymbol{y}$ is uniformly random, so is $\boldsymbol{\delta} = -(\boldsymbol{w}e + \boldsymbol{y})$. Hence, the zero-knowledge property is satisfied. We depict the algorithms in Figure 5.1. □

Finally, we show that any matrix which satisfies the efficient prover property as well as the affine map and F-rank properties must satisfy the first column dependence property. Thus, the latter property is actually needed.

Lemma 5.3.4. *Let $\boldsymbol{C}(\boldsymbol{X})$ be a matrix that satisfies the affine map, F-rank and efficient prover properties. Then $\boldsymbol{C}$ satisfies the first column dependence property.*

Proof. Fix $\mathsf{p}, \boldsymbol{\chi}$, and $\boldsymbol{C}(\boldsymbol{X}) = (\boldsymbol{h}\|\boldsymbol{T})(\boldsymbol{X})$, and let comp_i be any (potentially inefficient) algorithms that output $([\boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$, such that $[\boldsymbol{\gamma}]_1$ does not depend on e. Consider any $([\boldsymbol{\gamma}]_1, st) \leftarrow \mathsf{comp}_1(\mathsf{p}, \boldsymbol{\chi}, \boldsymbol{C}(\boldsymbol{X}))$. For any e and the given st, let $[\boldsymbol{\delta}_e]_2 \leftarrow \mathsf{comp}_2(st; [e]_2)$. Suppose that $\boldsymbol{\gamma}$ does not depend on e. Fix any $e \neq e'$. Since Equation (5.2) and thus Equation (5.3) holds for both e (and thus $\boldsymbol{\delta} = \boldsymbol{\delta}_e$) and e' (and thus $\boldsymbol{\delta} = \boldsymbol{\delta}_{e'}$),

$$\boldsymbol{h}(\boldsymbol{\chi})(e - e') + \boldsymbol{T}(\boldsymbol{\chi})(\boldsymbol{\delta}_e - \boldsymbol{\delta}_{e'}) = 0 \ .$$

Thus, $\boldsymbol{h}(\boldsymbol{\chi}) = \boldsymbol{T}(\boldsymbol{\chi})((\boldsymbol{\delta}_e - \boldsymbol{\delta}_{e'})/(e' - e))$, and thus $\boldsymbol{h}(\boldsymbol{\chi}) \in \mathrm{colspace}(\boldsymbol{T}(\boldsymbol{\chi}))$. □

5.4 Argument for Algebraic Set of Principal Ideal

Fix $\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$ and define $\mathcal{D}_\mathsf{p} := \{\mathtt{lpar} = (\mathsf{pk}, F)\}$, where

(1) pk is an Elgamal public key for encrypting in $\mathbb{G}_1$, and

(2) F is a polynomial with $\mathsf{qdc}(F) = \mathsf{poly}(\lambda)$, i.e., there exists a $\mathsf{poly}(\lambda)$-size QDR $\boldsymbol{C}(\boldsymbol{X})$ of F. (In Sections 5.5 and 5.6, we will show that such QDRs exist for many F-s.)

Before going on, recall that $C_{ij}(\boldsymbol{X}) = \sum_{k=1}^{\nu} P_{kij} X_k + Q_{ij}$ for public P_{kij} and Q_{ij}. To simplify notation, we will use vector/matrix format, by writing

$$\boldsymbol{C}(\boldsymbol{X}) = \sum_{k=1}^{\nu} \boldsymbol{P}_k X_k + \boldsymbol{Q} \ .$$

As always, we denote $\mathsf{Enc}([\boldsymbol{a}]_1; \boldsymbol{r}) := (\mathsf{Enc}([a_i]_1; r_i))_i$. We often omit $\boldsymbol{\chi}$ in notation like $[\boldsymbol{C}(\boldsymbol{\chi})]_1$, and just write $[\boldsymbol{C}]_1$.

Protocol Description

Let $\mathcal{L}_{\mathsf{pk},F}$ be defined as in Equation (5.1). The new Σ-protocol and NIZK argument for $\mathcal{L}_{\mathsf{pk},F}$ are based on the same underlying idea. Since the new NIZK is a CH-compilation of the Σ-protocol, it suffices to describe intuition behind the NIZK.

In the new NIZK argument (see Figure 5.2), P uses comp_1 to compute $[\boldsymbol{\gamma}]_1$ (together with state st), encrypts $[\boldsymbol{\gamma}]_1$ by using fresh randomness $\boldsymbol{\varrho}$, and then uses comp_2 (given $\mathsf{crs} = [e]_2$) to compute $[\boldsymbol{\delta}]_2$. If P is honest, then by the definition of QDRs of F, Equation (5.2) holds, i.e., $\boldsymbol{\gamma} + \boldsymbol{C}(\boldsymbol{\chi})\left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right) = \boldsymbol{0}$. The latter is equivalent to $\boldsymbol{\gamma} + (\sum_k \boldsymbol{P}_k \chi_k)\left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right) = -\boldsymbol{Q}\left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right)$. V needs to be able to check that the last equation holds, while given only an encryption of $[\boldsymbol{\gamma}]_1$. To help V to do that, P sends a vector of randomizers $[\boldsymbol{z}]_2$ to V as helper elements that help to "cancel out" the randomizers used by the prover to encrypt $[\boldsymbol{\gamma}]_1$ and $[\boldsymbol{\chi}]_1$.

The new NIZK argument is given in Figure 5.2.

Efficiency

Next, we estimate the efficiency of the NIZK argument. Note that if we use the comp algorithm given in Figure 5.1, we see that the algorithm computes $\boldsymbol{w}$ and $\boldsymbol{y}$ such that $[\boldsymbol{\delta}]_2 = -(\boldsymbol{w}[e]_2 + \boldsymbol{y}[1]_2)$. This lets us write $\left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 = \left(\begin{smallmatrix} 1 \\ -\boldsymbol{w} \end{smallmatrix}\right)[e]_2 + \left(\begin{smallmatrix} 0 \\ -\boldsymbol{y} \end{smallmatrix}\right)[1]_2$. This allows us to compute $[\boldsymbol{z}]_2$ as $(\sum_{k=1}^{\nu} r_k \boldsymbol{P}_k)\left(\begin{smallmatrix} 1 \\ -\boldsymbol{w} \end{smallmatrix}\right)[e]_2 + (\boldsymbol{\varrho} + \sum_{k=1}^{\nu} r_k \boldsymbol{P}_k)\left(\begin{smallmatrix} 0 \\ -\boldsymbol{y} \end{smallmatrix}\right)[1]_2$, which can be done with 2ℓ exponentiations in $\mathbb{G}_2$. This leads to the following lemma. Its proof follows by direct observation.

Lemma 5.4.1. *Consider $\boldsymbol{\Pi}_{\mathsf{nizk}}$ with QDR $\boldsymbol{C}$. Define $T_P(\boldsymbol{C}) := |\{(i,j) : \exists k, P_{kij} \neq 0\}|$, and $T_Q(\boldsymbol{C}) := |\{(i,j) : Q_{ij} \neq 0\}|$. Let $\mathfrak{c}$ be the time needed to run comp, $\mathfrak{e}_\iota$ is the time of an exponentiation in $\mathbb{G}_\iota$, and $\mathfrak{p}$ is the time of a pairing. Then*

$\mathsf{Kgen}(\mathsf{p}, \mathtt{lpar})$: $e \leftarrow\!\!\$\, \mathbb{Z}_p$; return $(\mathsf{crs}, \mathtt{td}) \leftarrow ([e]_2, e)$;

$\mathsf{P}(\mathsf{crs}, \mathtt{lpar}, \mathtt{x} = [\mathbf{ct}]_1, \mathtt{w} = (\boldsymbol{\chi}, \boldsymbol{r}))$: $([\boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2) \leftarrow \mathsf{comp}(\mathsf{p}, [e]_2, \boldsymbol{\chi}, \boldsymbol{C}(\boldsymbol{X}))$;
$\boldsymbol{\varrho} \leftarrow\!\!\$\, \mathbb{Z}_p^\ell$; $[\mathbf{ct}^\gamma]_1 \leftarrow \mathsf{Enc}([\boldsymbol{\gamma}]_1; \boldsymbol{\varrho}) \in \mathbb{G}_1^{\ell\times 2}$;
$[\boldsymbol{z}]_2 \leftarrow \boldsymbol{\varrho}[1]_2 + (\sum_{k=1}^{\nu} r_k \boldsymbol{P}_k)\,[\,{}^{e}_{\boldsymbol{\delta}}\,]_2 \in \mathbb{G}_2^\ell$.
Return $\pi \leftarrow ([\mathbf{ct}^\gamma]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2) \in \mathbb{G}_1^{\ell\times 2} \times \mathbb{G}_2^{2\ell-1}$.

$\mathsf{V}(\mathsf{crs}, \mathtt{lpar}, \mathtt{x} = [\mathbf{ct}]_1, \pi)$: check $[\boldsymbol{I}_\ell]_2 \bullet [\mathbf{ct}^\gamma]_1 + \sum_{k=1}^{\nu} (\boldsymbol{P}_k\,[\,{}^{e}_{\boldsymbol{\delta}}\,]_2 \bullet [\mathbf{ct}_k]_1) \stackrel{?}{=} (-\boldsymbol{Q}\,[\,{}^{e}_{\boldsymbol{\delta}}\,]_2) \bullet [0\|1]_1 + [\boldsymbol{z}]_2 \bullet \mathsf{pk}$.

$\mathsf{Sim}(\mathsf{crs}, \mathtt{td}, \mathtt{lpar}, \mathtt{x} = [\mathbf{ct}]_1)$: $\boldsymbol{\delta} \leftarrow\!\!\$\, \mathbb{Z}_p^{\ell-1}$;
$\boldsymbol{z} \leftarrow\!\!\$\, \mathbb{Z}_p^\ell$; $[\mathbf{ct}^\gamma]_1 \leftarrow \mathsf{Enc}(-\boldsymbol{Q}(\,{}^{e}_{\boldsymbol{\delta}})[1]_1; \boldsymbol{z}) - \sum_{k=1}^{\nu} \boldsymbol{P}_k(\,{}^{e}_{\boldsymbol{\delta}})[\mathbf{ct}_k]_1$;
Return $\pi \leftarrow ([\mathbf{ct}^\gamma]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2) \in \mathbb{G}_1^{\ell\times 2} \times \mathbb{G}_2^{2\ell-1}$.

Figure 5.2: The new NIZK argument $\boldsymbol{\Pi}_{\mathsf{nizk}}$ for $\mathcal{L}_{\mathsf{pk},F}$.

(1) the prover's computation is dominated by $\mathfrak{c} + 2\ell \cdot \mathfrak{e}_1 + 2\ell \cdot \mathfrak{e}_2$,

(2) the verifier's computation is dominated by $(T_P(\boldsymbol{C}) + T_Q(\boldsymbol{C})) \cdot \mathfrak{e}_2 + 2(2+\nu)\ell \cdot \mathfrak{p}$,

(3) the communication is 2ℓ *elements of* $\mathbb{G}_1$ *and* $2\ell - 1$ *elements of* $\mathbb{G}_2$.

For the argument to be efficient, we need comp to be efficient (according to Section 5.3, it must be efficient to solve the system $\boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{w} = \boldsymbol{h}(\boldsymbol{\chi})$ for $\boldsymbol{w}$, where $\boldsymbol{C}(\boldsymbol{X}) = (\boldsymbol{h}\|\boldsymbol{T})(\boldsymbol{X})$), and the matrices $\boldsymbol{P}_k$ and $\boldsymbol{Q}$ have to be sparse.

In Section 5.5, we propose a way to construct $\boldsymbol{C}(\boldsymbol{X})$ that satisfies these restrictions for any $F(\boldsymbol{X})$ that can be computed by a polynomial-size ABP. In Section 5.6, we study other interesting cases.

The estimate in Lemma 5.4.1 is often over-conservative. For example, let $\boldsymbol{\delta}' = (\,{}^{e}_{\boldsymbol{\delta}})$. If $P_{kij_1} = P_{kij_2} =: P'$ for $j_1 \neq j_2$, then the verifier has to perform one exponentiation $P'([\delta'_{j_1}]_2 + [\delta'_{j_2}]_2)$ instead of two. The same holds when $Q_{ij_1} = Q_{ij_2}$ for some $j_1 \neq j_2$. Moreover, when the exponent is a small constant (in the extreme case, 1 or -1), then one does not have to perform a full-exponentiation.

Security of the NIZK Argument

Theorem 5.4.2. *Let* $\{\mathcal{D}_\mathsf{p}\}_\mathsf{p}$ *be the family of language distributions, where* $\mathcal{D}_\mathsf{p} = \{\mathtt{lpar} = (\mathsf{pk}, F)\}$ *as before. Here,* $F(\boldsymbol{X})$ *is a* ν*-variate polynomial of degree* d*, where* $\nu, d \in \mathsf{poly}(\lambda)$*. Let* $\boldsymbol{C}(\boldsymbol{X}) \in \mathbb{Z}_p[\boldsymbol{X}]^{\ell\times\ell}$ *be a QDR of* F*. The NIZK argument* $\boldsymbol{\Pi}_{\mathsf{nizk}}$ *for* $\{\mathcal{D}_\mathsf{p}\}_\mathsf{p}$ *from Figure 5.2 is perfectly complete and perfectly zero-knowledge. It is computationally (adaptive) sound under the* $\mathcal{L}_1$*-*$(\ell-1)$*-*CED *assumption in* $\mathbb{G}_2$ *relative to* Pgen.

Proof. **Completeness:** To see that the NIZK argument is complete, transform the verification equation as follows:

$$
\begin{aligned}
[\boldsymbol{I}_\ell]_2 \bullet [\mathbf{ct}^\gamma]_1 + \sum_{k=1}^{\nu} (\boldsymbol{P}_k \left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 \bullet [\mathbf{ct}_k]_1) &=^? (-\boldsymbol{Q} \left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2) \bullet [0\|1]_1 + [\boldsymbol{z}]_2 \bullet \mathsf{pk} \iff \\
[\mathbf{ct}^\gamma]_1 + \sum_{k=1}^{\nu} \boldsymbol{P}_k \left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right) [\mathbf{ct}_k]_1 &=^? \mathsf{Enc}([-\boldsymbol{Q} \left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right)]_1; \boldsymbol{z}) \iff \\
\mathsf{Enc}([\boldsymbol{\gamma}]_1; \boldsymbol{\varrho}) + \sum_{k=1}^{\nu} \boldsymbol{P}_k \left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right) \mathsf{Enc}([\chi_k]_1; r_k) &=^? \mathsf{Enc}([-\boldsymbol{Q} \left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right)]_1; \boldsymbol{z}) \iff \\
\mathsf{Enc}\left([\boldsymbol{\gamma} + \boldsymbol{C}(\boldsymbol{\chi}) \left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right)]_1; \boldsymbol{\varrho} + \left(\sum_{k=1}^{\nu} r_k \boldsymbol{P}_k\right) \left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right) - \boldsymbol{z}\right) &=^? \mathsf{Enc}([\mathbf{0}]_1; \mathbf{0})
\end{aligned}
$$

which holds since the prover is honest and due to the definition of $\boldsymbol{z}$.

Perfect zero-knowledge: Fix any λ, $(\mathsf{p}, \mathtt{td}) \in \mathrm{Supp}(\mathsf{K}_{\mathsf{crs}}(1^\lambda))$ and compute $\mathsf{crs} = [\mathtt{td}]_2$. Then fix $\mathtt{lpar} \in \mathrm{Supp}(\mathcal{D}_{\mathsf{p}})$ and $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}_{\mathtt{lpar}}$. In the honest prover's algorithm, since $\boldsymbol{\varrho}$ is uniformly random, then also $\boldsymbol{z}$ is uniformly random. By the zero-knowledge property (see Section 5.3), $\boldsymbol{\delta}$ output by an honest prover is uniformly random. On the other hand, Sim (see Figure 5.2) also samples uniformly random $\boldsymbol{\delta}$ and $\boldsymbol{z}$. Finally, in both the prover's and simulator's case, $[\mathbf{ct}^\gamma]_1$ is the unique value that makes the verifier accept the argument π. Hence, the distributions of the prover and the simulator are perfectly indistinguishable.

Computational soundness. Let $\mathcal{A}$ be a soundness adversary that, for honestly generated crs and any $\mathtt{lpar} \in \mathrm{Supp}(\mathcal{D}_{\mathsf{p}})$ (including $\boldsymbol{C}$), breaks $\boldsymbol{\Pi}_{\mathsf{nizk}}$ in time τ and with probability ε. We construct the following $\mathcal{L}_1$-$(\ell-1)$-CED adversary $\mathcal{B}$. (See Definition 5.2.1 for the definition of CED.)

The CED challenger creates $\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$, $[\boldsymbol{D}]_2 = \left[\begin{smallmatrix} 1 \\ e \end{smallmatrix}\right]_2 \leftarrow\!\!\$\, \mathcal{L}_1$ and sends $(\mathsf{p}, [\boldsymbol{D}]_2)$ to $\mathcal{B}$. $\mathcal{B}$ runs $(\mathsf{crs}, \mathtt{td}) \leftarrow \mathsf{K}_{\mathsf{crs}}(\mathsf{p})$. $\mathcal{B}$ runs the setup algorithm of Elgamal to compute a random secret key sk and public key pk from the correct distribution. $\mathcal{B}$ fixes any F such that $\mathtt{lpar} = (\mathsf{pk}, F) \in \mathrm{Supp}(\mathcal{D}_{\mathsf{p}})$, and sends $\mathsf{crs} = [e]_2$ and $\mathtt{lpar}$ to $\mathcal{A}$. Let $\boldsymbol{C}$ be a fixed $\mathsf{poly}(\lambda)$-size QDR of F.

Assume that $\mathcal{A}$ returns an accepting input-argument pair $(\mathtt{x} = [\mathbf{ct}]_1, \pi)$, such that $\mathtt{x} \notin \mathcal{L}_{\mathtt{lpar}}$, i.e., $[\boldsymbol{\chi}]_1 \leftarrow \mathsf{Dec}([\mathbf{ct}]_1)$ is such that $F(\boldsymbol{\chi}) \neq 0$. $\mathcal{B}$ uses sk to decrypt $[\mathbf{ct}]_1$ to $[\boldsymbol{\chi}]_1$ and $[\mathbf{ct}^\gamma]_1$ to $[\boldsymbol{\gamma}]_1$. $\mathcal{B}$ recomputes $[\boldsymbol{C}(\boldsymbol{\chi})]_1 \leftarrow \sum \boldsymbol{P}_k[\chi_k]_1 + \boldsymbol{Q}$. $\mathcal{B}$ returns $[\boldsymbol{\gamma}\|\boldsymbol{C}(\boldsymbol{\chi})]_1$ and $[\boldsymbol{\delta}]_2$ to the CED challenger.

Since $\mathcal{A}$ is successful, the verification equation in Figure 5.2 holds, and thus also the following "decryption" of the verification equation holds:

$$
[\boldsymbol{I}_\ell]_2 \bullet [\boldsymbol{\gamma}]_1 + \sum_{k=1}^{\nu} (\boldsymbol{P}_k \left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 \bullet [\chi_k]_1) = (-\boldsymbol{Q} \left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2) \bullet [1]_1 \ .
$$

Thus, $\boldsymbol{\gamma} + \boldsymbol{C}(\boldsymbol{\chi}) \left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right) = \mathbf{0}$, i.e., Equation (5.2) holds. Since $\det(\boldsymbol{C}(\boldsymbol{\chi})) = F(\boldsymbol{\chi}) \neq 0$, $\boldsymbol{C}$ has full rank. Thus, $\mathcal{B}$ breaks CED. □

5.5 Efficient Instantiation Based on ABP

In this section we construct QDRs, that we denote by $\mathsf{IK}(\boldsymbol{X})$, for any polynomial F that can be efficiently computed by algebraic branching programs (ABPs, [Nis91; BG99]). This results in NIZKs for the class of languages $\mathcal{L}_{\mathsf{pk},F}$, where F is only restricted to have a small ABP. However, in many cases, the resulting matrix $\mathsf{IK}(\boldsymbol{X})$ is not optimal, and this will be seen in Section 5.7. Thus, following sections consider alternative construction techniques of such matrices.

Preliminaries: Algebraic Branching Programs

A branching program is defined by a directed acyclic graph (V, E), two special vertices $s, t \in V$, and a labeling function ϕ. An algebraic branching program (ABP, [Nis91; BG99]) over a finite field $\mathbb{F}_p$ computes a function $F : \mathbb{F}_p^\nu \to \mathbb{F}_p$. Here, ϕ assigns to each edge in E a fixed affine (possibly, constant) function in input variables, and $F(\boldsymbol{X})$ is the sum over all $s-t$ paths (i.e., paths from s to t) of the product of all the values along the path.

Algebraic branching programs capture a large class of functions, including in particular all log-depth circuits, boolean branching programs, boolean formulas, logspace circuits, and many more. For some type of computations, they are known to provide a relatively compact representation, which makes them especially useful. See [IK00; IK02; IW14] and the references therein.

Ishai and Kushilevitz [IK00; IK02] related ABPs to matrix determinants as follows.

Proposition 5.5.1. *[IK02, Lemma 1] Given an ABP* $\mathsf{abp} = (V, E, s, t, \phi)$ *computing* $F : \mathbb{F}_p^\nu \to \mathbb{F}_p$*, we can efficiently (and deterministically) compute a function* $\mathsf{IK}(\boldsymbol{\chi})$ *mapping an input* $\boldsymbol{\chi} \in \mathbb{F}_p^\nu$ *to a matrix from* $\mathbb{F}_p^{\ell \times \ell}$*, where* $\ell = |V| - 1$*, such that:*

1. $\det(\mathsf{IK}(\boldsymbol{\chi})) = F(\boldsymbol{\chi})$*,*
2. *each entry of* $\mathsf{IK}(\boldsymbol{\chi})$ *is an affine map in a single variable* χ_i*,*
3. $\mathsf{IK}(\boldsymbol{\chi})$ *contains only* -1*'s in the upper* 1*-diagonal (the diagonal above the main diagonal) and* 0*'s above the upper* 1*-diagonal.*

Specifically, IK *is obtained by transposing the matrix you get by removing the column corresponding to* s *and the row corresponding to* t *in the matrix* $\mathsf{adj}(\boldsymbol{X}) - \boldsymbol{I}$*, where* $\mathsf{adj}(\boldsymbol{X})$ *is the adjacency matrix for* abp*.*

Note that the matrix IK is transposed compared to what is found in [IK02, Lemma 1], to ensure consistency with the notation from the CED assumption.

NIZK for Algebraic Branching Programs

Lemma 5.5.2. *Let* $\mathsf{abp} = (V, E, s, t, \phi)$ *be an ABP that computes a* ν*-variate polynomial* $F(\boldsymbol{X})$*. Then* $\mathsf{IK}(\boldsymbol{X})$ *is a QDR of* F *with* $\ell = |V| - 1$*.*

Proof. Items 1 and 2 of Proposition 5.5.1 state directly that the affine map and reducibility properties of Definition 5.3.1 hold. From 3 of Proposition 5.5.1, it follows that $\mathsf{IK}(\boldsymbol{X})_{(\ell,1)}$ is an upper triangular matrix where the diagonal which only consists of -1's. Clearly, $\det(\mathsf{IK}(\boldsymbol{\chi})_{(\ell,1)}) \neq 0$ for any $\boldsymbol{\chi}$; thus, it follows from Lemma 5.3.2 that the first column dependence property is also satisfied. The claim $\ell = |V| - 1$ is obvious. □

In particular, $\mathsf{qdc}(F) \leq |V| - 1$.

Efficiency of comp. We next specialize the general comp_i algorithms given in Figure 5.1 to ABP. For this, we just have to write down how to efficiently do the next two steps:

(1) Compute $\boldsymbol{\gamma} = \boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{y}$. Due to the shape of $\mathsf{IK}(\boldsymbol{\chi})$ and thus of $\boldsymbol{T}(\boldsymbol{\chi})$, one can clearly compute $\boldsymbol{\gamma}$ as $\gamma_i \leftarrow \sum_{j=1}^{i-1} T_{ij}(\boldsymbol{\chi}) y_{j-1} - y_i$ for each $i \in [1, \ell]$.

(2) Solve $\boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{w} = \boldsymbol{h}(\boldsymbol{\chi})$ for $\boldsymbol{w}$. Let $\boldsymbol{T}^*$ be the matrix obtained from $\boldsymbol{T}(\boldsymbol{\chi})$ by omitting its last row, and similarly let $\boldsymbol{h}^*$ be the vector obtained from $\boldsymbol{h}(\boldsymbol{\chi})$ by omitting its last element. One finds $\boldsymbol{w}$ by solving $\boldsymbol{T}^*\boldsymbol{w} = \boldsymbol{h}^*$ by forward substitution, as follows: $w_i \leftarrow \sum_{j=1}^{i-1} T_{ij}(\boldsymbol{\chi}) w_j - h_i(\boldsymbol{\chi})$ for each $i \in [1, \ell - 1]$.

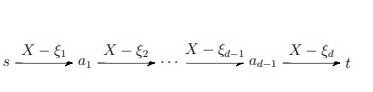
Lemma 5.5.3. *Let* $N(v)$ *be the neighbourhood of a node* v *in the underlying ABP. Assuming* $\boldsymbol{C}(\boldsymbol{X}) = \mathsf{IK}(\boldsymbol{X})$*, the computational complexity of* comp *is dominated by* $2(|E| - |N(s)|) - |N(t)|$ *field multiplications,* ℓ *exponentiations in* $\mathbb{G}_1$*, and* $2(\ell - 1)$ *exponentiations in* $\mathbb{G}_2$*.*

Proof. Clearly, computing $\boldsymbol{\gamma}$ requires at most $|E| - |N(s)|$ field multiplications, and computing $\boldsymbol{w}$ requires at most $|E| - |N(s)| - |N(t)|$ field multiplications. Finally, in the case of the NIZK argument, computing $[\boldsymbol{\gamma}]_1$ requires ℓ exponentiations in $\mathbb{G}_1$, and computing $[\delta]_2$ requires $2(\ell - 1)$ exponentations in $\mathbb{G}_2$. □

5.6 Applications

Univariate F (Set-Membership Proof)

Consider an algebraic set $\mathcal{A} \in \mathbb{Z}_p$ of size $\mathsf{poly}(\lambda)$, generated by τ univariate polynomials $F_1, \ldots, F_\tau \in \mathbb{Z}_p[X]$. As before, we aim to prove that an Elgamal-encrypted χ satisfies $\chi \in \mathcal{A}$, i.e., $F_i(\chi) = 0$ for all i. In the univariate case, all ideals are principal [ALO15, Section 1.5], and thus any ideal can be written as $\mathfrak{I} = \langle F \rangle$ for some F. Thus, $\mathcal{A} = \mathcal{A}(F)$ for $F \leftarrow \gcd(F_1, \ldots, F_\tau)$ [ALO15, Section 1.5].



$$\mathsf{IK}_{path}(X) = \begin{pmatrix} X-\xi_1 & -1 & 0 & \dots & 0 \\ 0 & X-\xi_2 & -1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -1 \\ 0 & 0 & 0 & \dots & X-\xi_d \end{pmatrix}$$

Figure 5.3: The ABP $\mathsf{abp}^d_{\mathsf{path}}(X, \boldsymbol{\xi})$ for $F(X) = \prod_{i=1}^{d}(X - \xi_i)$ and $\mathsf{IK}_{path}(X)$

$\mathsf{Kgen}(\mathsf{p}, \mathtt{lpar})$: $e \leftarrow\!\!\$\, \mathbb{Z}_p$; return $(\mathsf{crs}, \mathsf{td}) \leftarrow ([e]_2, e)$;

$\mathsf{P}(\mathsf{crs}, \mathtt{lpar}, \mathtt{x} = [\mathbf{ct}]_1, \mathtt{w} = (\chi, r))$: $([\boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2) \leftarrow \mathsf{comp}(\mathsf{p}, [e]_2, \chi, \boldsymbol{C}(\boldsymbol{X}))$;
$\boldsymbol{\varrho} \leftarrow\!\!\$\, \mathbb{Z}_p^d$; $[\mathbf{ct}^\gamma]_1 \leftarrow \mathsf{Enc}([\boldsymbol{\gamma}]_1; \boldsymbol{\varrho}) \in \mathbb{G}_1^{d\times 2}$; $[\boldsymbol{z}]_2 \leftarrow \boldsymbol{\varrho}[1]_2 + r\,[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}]_2 \in \mathbb{G}_2^d$;
return $\pi \leftarrow ([\mathbf{ct}^\gamma]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2)$.

$\mathsf{V}(\mathsf{crs}, \mathtt{lpar}, \mathtt{x} = [\mathbf{ct}]_1, \pi)$: check $[\boldsymbol{I}_d]_2 \bullet [\mathbf{ct}^\gamma]_1 + [\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}]_2 \bullet [\mathbf{ct}]_1 + \boldsymbol{Q}\,[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}]_2 \bullet [0\|1]_1 =^? [\boldsymbol{z}]_2 \bullet \mathsf{pk}$.

$\mathsf{Sim}(\mathsf{crs}, \mathsf{td}, \mathtt{lpar}, \mathtt{x} = [\mathbf{ct}]_1)$: $\boldsymbol{\delta} \leftarrow\!\!\$\, \mathbb{Z}_p^{d-1}$; $\boldsymbol{z} \leftarrow\!\!\$\, \mathbb{Z}_p^d$; $[\mathbf{ct}^\gamma]_1 \leftarrow \mathsf{Enc}(-\boldsymbol{Q}(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix})[1]_1; \boldsymbol{z}) - (\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}) \cdot [\mathbf{ct}]_1$; return $\pi \leftarrow ([\mathbf{ct}^\gamma]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2)$.

Figure 5.4: The NIZK argument for $\mathcal{L}_{\mathsf{pk},F}$, where $F(X)$ is a monic univariate polynomial with $\mathsf{qdc}(F) = d$.

Moreover, $\mathcal{I}(\mathcal{A}(F)) = \mathcal{I}(F_{red})$ [ALO15, Section 1.5], where F_{red} has the same roots as F but all with multiplicity one. That is, if $F(X) = \prod(X - \xi_i)^{b_i}$, for $b_i \geq 1$ and mutually different ξ_i, then $F_{red} = \prod(X - \xi_i)$. This *reduced polynomial* F_{red} can be efficiently computed as $F_{red} = F/\gcd(F, F')$, [ALO15, Section 1.5]. Since we are constructiong NIZKs for algebraic sets, in this section, we will assume that $F(X) = F_{red}(X) = \prod(X-\xi_i)$ for mutually different roots ξ_i. (This will be the case if we assume $\mathcal{A} = \{\xi_i\}$ for polynomially many ξ_i.) Thus, it suffices to prove that $F(\chi) = 0$, where F is a reduced polynomial. As before, for efficiency reasons, we assume that F has degree $\mathsf{poly}(\lambda)$.

We now apply the ABP-based protocol to a univariate reduced polynomial F. We depict the ABP $\mathsf{abp}^d_{\mathsf{path}}(X, \boldsymbol{\xi})$ in Figure 5.3. The ABP consists of a single path of length d with edges labelled by values $X - \xi_i$. Clearly, $\mathsf{abp}^d_{\mathsf{path}}(X, \boldsymbol{\xi})$ computes $F(X)$. The corresponding matrix $\mathsf{IK}_{path}(X)$ is also given in Figure 5.3.

Figure 5.4 depicts the resulting set-membership NIZK argument that $X \in \{\xi_i\}$.

Lemma 5.6.1. *Let $F(X)$ be a univariate reduced polynomial. The ABP-based NIZK argument for $\mathcal{L}_{\mathsf{pk},F}$ has prover's computation of at most $3d$ exponentiations in $\mathbb{G}_1$ and $4d-2$ exponentiations in $\mathbb{G}_2$, verifier's computation of $7d-1$ pairings and at most d exponentiations in $\mathbb{G}_2$, and communication of $2d$ elements of $\mathbb{G}_1$ and $2d-1$ elements of $\mathbb{G}_2$.*

Proof. **Prover:** First, we write down the concrete formulas for the comp algorithm from Figure 5.1.

1. Computation of $\boldsymbol{\gamma} = \boldsymbol{T}(\chi)\boldsymbol{y}$: one sets $\gamma_1 \leftarrow -y_1$, $\gamma_i \leftarrow (\chi - \xi_i)y_{i-1} - y_i$ for $i \in [2, d-1]$, and $\gamma_d \leftarrow (\chi - \xi_d)y_{d-1}$. ($d-1$ field operations.)

   $[\boldsymbol{\gamma}]_1$ can then be computed by using at most d exponentiations in $\mathbb{G}_1$. However, if either (a) $\chi = \xi_d$ or (b) $\chi - \xi_i$ is small for all i, then $d-1$ exponentiations suffice.

2. Solving $\boldsymbol{T}(\chi)\boldsymbol{w} = \boldsymbol{h}(\chi)$ for $\boldsymbol{w}$: $w_i \leftarrow -\prod_{j=1}^{i}(\chi - \xi_j)$ for $i \in [1, d-1]$.

   This allows us compute $[\boldsymbol{\delta}]_2$ in the following way: Define $[a_i]_2 := w_i[e]_2$. We can recursively compute $[a_i]_2$ as $[a_1]_2 = (\chi - \xi_1)[e]_2$ and $[a_i]_2 = (\chi - \xi_i)[a_{i-1}]_2$, and so computing each $[a_i]_2$ requires at most 1 exponentiation. Note that if $\chi = \xi_j$, then $[a_j]_2 = [0]_2$ and thus requires no exponentiations. Further, each $[a_i]_2 = [0]_2$ for each $i \geq j$, which then also do not require exponentiations.

   We finally compute $[\delta_i]_2 = [a_i]_2 + [y_i]_2$, which gives us a total of at most $2d-2$ exponentiations in $\mathbb{G}_2$, and we only achieve this bound if $\chi = \xi_d$,

Since field operations are cheap, comp is dominated by at most d exponentiations in $\mathbb{G}_1$ to compute $[\boldsymbol{\gamma}]_1$ and $2d-2$ exponentiations in $\mathbb{G}_2$ (up to $d-2$ of which can have a small exponent $\chi - \xi_i$) to compute $[\boldsymbol{\delta}]_2$. In addition, the prover performs $2d$ exponentiations in $\mathbb{G}_1$ to compute $[\mathbf{ct}^{\gamma}]_1$ and $2d$ exponentiations in $\mathbb{G}_2$ to compute $[\boldsymbol{z}]_2$. Thus, the prover performs $3d$ ($3d-1$ if $\chi = \xi_d$) in $\mathbb{G}_1$ and $4d-2$ exponentiations in $\mathbb{G}_2$.

Verifier: We first note that $\boldsymbol{Q}\left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 = -\boldsymbol{\xi} \circ \left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 - \left[\begin{smallmatrix} \boldsymbol{\delta} \\ 0 \end{smallmatrix}\right]_2 \in \mathbb{G}_2^d$. Thus,

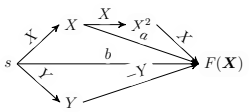
$$\left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 \bullet [\mathbf{ct}]_1 + \boldsymbol{Q}\left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 \bullet [0\|1]_1 = \left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 \bullet [\mathbf{ct}]_1 - (\boldsymbol{\xi} \circ \left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 + \left[\begin{smallmatrix} \boldsymbol{\delta} \\ 0 \end{smallmatrix}\right]_2) \bullet [0\|1]_1 = [\boldsymbol{\kappa}]_T - \left[\begin{smallmatrix} \boldsymbol{\delta} \\ 0 \end{smallmatrix}\right]_2 \bullet [0\|1]_1 \ ,$$

where $[\kappa_i]_T = [(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix})_i]_2 \bullet ([\mathbf{ct}]_1 - \xi_i \circ [0\|1]_1)$. Here, $(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix})_i$ is the ith coefficient of the vector $(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix})$. Thus, $\boldsymbol{Q}\left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2$ can be computed in $3d-1$ pairings. Thus, the verifier's total computation is $7d-1$ pairings. Note that the verifier executes at most d exponentiations; however, this number is smaller if the exponents are small. Moreover, one can usually precompute all values $[\xi_i]_1$.

Communication: $2d$ group elements to transfer the ciphertexts $[\mathbf{ct}^{\gamma}]_1$, $d-1$ group elements to transfer $[\boldsymbol{\delta}]_2$, and d group elements to transfer the randomizers $[\boldsymbol{z}]_2$, $4d-1$ group elements in total. □

Special Case: OR Arguments

In an OR argument, the language is $\mathcal{L}_{\mathsf{pk},X(X-1)}$, that we will just denote by $\mathcal{L}_{\{0,1\}}$, assuming that pk is understood from the context. The case of OR arguments is of particular interest because of its wide applications in many different scenarios. Indeed, one of the most direct applications of [CH20] is a new OR proof with the argument consisting of 7 group elements. Due to the importance of $\mathcal{L}_{\{0,1\}}$, in Section 5.C, we will detail three example NIZK arguments that are all based on CED-matrices. The



$$\mathsf{IK}(X,Y) = \begin{pmatrix} X & -1 & 0 & 0 \\ 0 & X & -1 & 0 \\ Y & 0 & 0 & -1 \\ b & a & X & -Y \end{pmatrix}$$

Figure 5.5: ABP example for $F(X,Y) = X^3 + aX + b - Y^2$.

first argument is based on $\mathsf{abp}^2_{\mathsf{path}}$, and the other two arguments are based on known Σ-protocols from the literature. Interestingly, the third example is not based on ABPs; the added discussion clarifies some benefits of using the ABP-based approach.

Elliptic Curve Points

In Figure 5.5, we depict an ABP and $\mathsf{IK}(X,Y)$ for the bivariate function $F(X,Y) = X^3+aX+b-Y^2$ (i.e., one checks if (X,Y) belongs to the elliptic curve $Y^2 = X^3+aX+b$). In Section 5.7, we will propose a non-ABP-based QDR for the same task. ABPs for hyperelliptic curves $Y^2 + H(X)Y = f(X)$ (where $\deg(H) \leq g$ and $\deg f = 2g+1$) of genus g can be constructed analogously.

NIZK arguments that committed (X,Y) belongs to the curve are interesting in practice since one often needs to prove in zero-knowledge that a verifier of some pairing-based protocol accepts. Such a situation was studied in [Ben+14], who proposed to use cycles of elliptic curves, such that the number of points on one curve is equal to the size of the field of definition of the next, in a cyclic way. Using the NIZK, resulting from the example of the current subsection, one can use a bilinear group with group order p to prove that the encrypted coordinates belong to an elliptic curve where the finite field has size p.

Different normal form. Motivated by [PSV12], we also consider the following less common normal form for an elliptic curve, $F(X,Y) = (X+aY)(X+bY)(X+cY) - X$, for mutually different a, b, c. Then, one can construct the following ABP-based 3×3 QDR:

$$\begin{pmatrix} X+aY & -1 & 0 \\ 0 & X+bY & -1 \\ -X & 0 & X+cY \end{pmatrix} .$$

5.7 On Bivariate Case

Dickson [Dic21] proved that for any degree-d bivariate polynomial $F(\boldsymbol{X})$, there exists a $d \times d$ matrix $\boldsymbol{C}(\boldsymbol{X})$ of affine maps that has $F(\boldsymbol{X})$ as its determinant. Plaumann *et al.* [PSV12] described efficient algorithms for finding $\boldsymbol{C}(\boldsymbol{X})$ for some families of polynomials F; in their case, $\boldsymbol{C}(\boldsymbol{X})$ is usually symmetric and can satisfy some other additional requirement like semidefiniteness. Since the ABP-based approach often blow ups the dimension of the matrix, we will next use the results of [Dic21; PSV12] to

construct a $d \times d$ matrix $\boldsymbol{C}(\boldsymbol{X})$. However, the resulting matrix is usually not a QDR, which results in additional complications. We provide several concrete examples in the case $F(X, Y)$ describes an elliptic curve. Plaumann *et al.* [PSV12] provided also examples for the case $d \in \{4, 5\}$, noting however that finding a determinantal representation of F becomes very time-consuming for $d \geq 5$. In Section 5.D, we will provide an example for $d = 5$. We refer to [PSV12] for algorithms and general discussion.

Optimized Solutions for Elliptic Curves

Let $F(X, Y) = X^3 + aX + b - Y^2$ be a polynomial that describes an elliptic curve. In Section 5.6, we described a small ABP for checking that $(X, Y) \in E(\mathbb{Z}_p)$, where $E(\mathbb{Z}_p) : F(X, Y) = 0$. However, this resulted in a 4×4 matrix $\mathsf{IK}(X, Y)$. Next, we construct 3×3 matrices, of correct determinant, for two different choices of F. In general, there are several inequivalent linear symmetric determinantal representations of F, [PSV12]. In both cases, we chose the matrix by inspection.

Case $F(X, Y) = X^3 + aX + b - Y^2$ for $a \neq 0$. In Section 5.D, we show that in case there exists a 3×3 determinantal representation that is not a QDR, and discuss the possible issues that arise when one tries to use our NIZK argument in such a case.

Case $F(X, Y) = X^3 + b - Y^2$. We will tackle this case in Section 5.D.

5.8 Handling Non-Principal Ideals

Next, we extend the new framework to constructing a NIZK argument that an Elgamal-encrypted $\boldsymbol{\chi}$ satisfies $\boldsymbol{\chi} \in \mathcal{A}$ for any algebraic set $\mathcal{A} = \mathcal{A}(\mathcal{I})$. Namely, assume that $\mathcal{I}(\mathcal{A})$ has a known generating set $(F_1, \ldots, F_\tau)$ for some τ. We prove that $\boldsymbol{\chi} \in \mathcal{A}$ by proving that $F_i(\boldsymbol{\chi}) = 0$ for each F_i. Thus, $\mathcal{D}_{\mathsf{p}} = \{(\mathsf{pk}, \mathcal{A})\}$, where $\mathcal{I}(\mathcal{A}) = \langle F_1, \ldots, F_\tau \rangle$ and each F_i has $\mathsf{qdc}(F_i) = \mathsf{poly}(\lambda)$.

The argument system can be implemented in polynomial time and space, assuming that (1) we know a generating set with small $\tau = \mathsf{poly}(\lambda)$ and with small-degree polynomials, (2) for each F_i, we know a small QDR $\boldsymbol{C}_i(\boldsymbol{X})$ of F_i, and (3) we can construct an efficient NIZK argument system for showing that $\det(\boldsymbol{C}_i(\boldsymbol{X})) = 0$. The previous sections already tackled the last two issues. In this section, we study issue (1). However, the issues are related. In particular, steps (2) and (3) are most efficient for specific type of polynomials F_i, and when solving (1), we have to take this into account.

NIZK for NP

Next, we use the described methodology to implement arithmetic circuits, and then extend it to R1CS (a linear-algebraic version of QAP [Gen+13]) and aCSPs (*arithmetic constraint*

satisfaction systems), i.e, constraint systems where each constraint is a small-degree constant that depends on some small number of inputs. We also show how to directly use our techniques to implement the Groth-Sahai-Ostrovsky constraint system [GOS06] that have efficient reductions to corresponding circuits. Interestingly, this seems to result in the first known pairing-based (random-oracle-less) NIZK for general aCSPs; although see [Sze20] for a recent use of aCSPs to construct SNARKs.

Arithmetic circuits. Let $\mathfrak{C}$ be an arithmetic circuit over $\mathbb{Z}_p$, with n gates (including input gates) and m wires. We construct an algebraic set $\mathcal{A}_{\mathfrak{C}} = (\chi_1, \ldots, \chi_n) \in \mathbb{Z}_p^n$, such that $\boldsymbol{\chi} \in \mathcal{A}_{\mathfrak{C}}$ iff $\mathfrak{C}(\boldsymbol{\chi}) = 0$, as follows. First, $\boldsymbol{\chi}$ corresponds to the vector of wire values. As in the case of QAP [Gen+13], we assume that each gate is a weighted multiplication gate that computes

$$F_i : \left(\sum_j u_{ij}\chi_{i_j}\right)\left(\sum_j v_{ij}\chi_{i_j}\right) \mapsto \chi_i$$

for public u_{ij}, v_{ij}, and i_j, where for the sake of efficiency, the sum is taken over a constant number of values.

1. First, each χ_i corresponds to the value of the output wire of ith gate, with χ_j, $j \leq \mathsf{m}_0$ corresponding to the inputs of the circuit. We also assume that the last few wire values correspond to the output values of the circuit.

2. Second, for each gate $i > \mathsf{m}_0$, we introduce the polynomial $F_i(\boldsymbol{\chi}) = \chi_i - (\sum u_{ij}\chi_{i_j})(\sum v_{ij}\chi_{i_j})$.

Then $\mathcal{A}_{\mathfrak{C}} = \{(\chi_1, \ldots, \chi_m) : F_i(\boldsymbol{\chi}) = 0 \text{ for all } i > \mathsf{m}_0\}$. To construct a NIZK for showing $\boldsymbol{\chi} \in \mathcal{A}_{\mathfrak{C}}$, we do as before:

(1) We let the prover Elgamal-encrypt $\boldsymbol{\chi}$.

(2) We show that $F_i(\boldsymbol{\chi}) = 0$ for all i by using the NIZK argument from Section 5.4.

Note that each polynomial in this case is quadratic, and thus one can construct a 2×2 QDR

$$\boldsymbol{C}(\boldsymbol{\chi}) = \begin{pmatrix} \sum u_{ij}\chi_{i_j} & -1 \\ -\chi_i & \sum v_{ij}\chi_{i_j} \end{pmatrix} .$$

According to [GS08], the Groth-Sahai proof for this task has commitment length $(2m+1)(|\mathbb{G}_1|+|\mathbb{G}_2|)$ and argument length $(2m+2n+2)(|\mathbb{G}_1|+|\mathbb{G}_2|)$. The new NIZK has commitment length $2m|\mathbb{G}_1|$ and argument length $n(4|\mathbb{G}_1|+3|\mathbb{G}_2|)$. Assuming $m \approx n$ and $|\mathbb{G}_2| = 2|\mathbb{G}_1|$, the new NIZK has 3 times shorter commitments/encrypts and 20% shorter proofs. The new NIZK has approximately 1.5–2 times smaller prover's and verifier's computation. Since the computation in [GS08] can probably be optimized, we have not included complete comparison.

Extension: R1CS. In R1CS (*rank-1 constraint system* [Gen+13]), one has n constraints $(\sum u_{ij}\chi_i)(\sum v_{ij}\chi_i) = \sum w_{ij}\chi_i$ in m variables χ_i, for arbitrary public matrices $U = (u_{ij})$,

Table 5.2: Comparison of falsifiable NIZKs for Boolean circuit satisfiability: the Groth-Sahai proof, as optimized by Ghadafi *et al.* [GSW09], and the new NIZK from Section 5.8. Here, $|\mathbb{G}_\iota|$ is the length of one element from $\mathbb{G}_\iota$

Protocol	$\|\mathsf{crs}\|$	$\|\mathsf{com}\|$	$\|\pi\|$	P comp.	V comp.
Groth-Sahai [GSW09]	$4(\|\mathbb{G}_1\|+\|\mathbb{G}_2\|)$	$2(m+1)(\|\mathbb{G}_1\|+\|\mathbb{G}_2\|)$	$(6m+2n+2)(\|\mathbb{G}_1\|+\|\mathbb{G}_2\|)$	$(12m+4n+4)(\mathfrak{e}_1+\mathfrak{e}_2)$	$16(2m+n)\mathfrak{p}$
New, Section 5.8	$\|\mathbb{G}_2\|$	$2m\cdot\|\mathbb{G}_1\|$	$(m+n)(4\|\mathbb{G}_1\|+3\|\mathbb{G}_2\|)$	$(m+n)(5\mathfrak{e}_1+4\mathfrak{e}_2)$	$13(m+n)\mathfrak{p}$

$V=(v_{ij})$, and $W=(w_{ij})$. There is clearly a simple reduction from arithmetic circuits to R1CS. The described solution for arithmetic circuits can be used to construct a NIZK argument system for R1CS, by defining $F_i(\boldsymbol{\chi})=(\sum u_{ij}\chi_i)(\sum v_{ij}\chi_i)-\sum w_{ij}\chi_i$ and

$$\boldsymbol{C}(\boldsymbol{\chi})=\begin{pmatrix}\sum u_{ij}\chi_{i_j} & -1\\ -\sum w_{ij}\chi_{i_j} & \sum v_{ij}\chi_{i_j}\end{pmatrix}.$$

Extension: Arithmetic Constraint Satisfaction Problems (aCSPs). Fix $\mathbb{F}=\mathbb{Z}_q$. Recall that for a $q\geq 1$, a q-aCSP instance F over $\mathbb{F}$ is a collection of functions $F_1,\ldots,F_\tau$ (called *constraints*) such that each function F_i depends on at most q of its input locations. That is, for every $j\in[1,\tau]$ there exist $i_1,\ldots,i_q\in[1,n]$ and $f:\mathbb{F}^q\to\mathbb{F}$ such that $F_j(\boldsymbol{\chi})=f(\chi_{i_1},\ldots,\chi_{i_q})$ for every $\boldsymbol{\chi}\in\mathbb{F}^n$. Then F is satisfiable if $F_j(\boldsymbol{\chi})=0$ for each j.

One can extend R1CS to q-aCSP for small constant q, assuming that F_j are (small-degree) polynomials for which one can construct poly-size QDRs. Intuitively, F is the generating set for some polynomial ideal $\mathcal{I}=\mathcal{I}(\mathcal{A})$, and thus the examples of this subsection fall under our general methodology. One can possibly use some general techniques (see Section 5.8 for some examples) to minimize the generating sets so as to obtain more efficient NIZKs.

Specialization: Boolean Circuits. By using techniques from [GOS06], one can construct a NIZK for any Boolean circuit that, w.l.o.g., consists of only NAND gates. Intuitively, one does this by showing that each wire value is Boolean, and then showing that each NAND gate is followed correctly. The latter can be shown by showing that a certain linear combination of the input and output wires of the NAND gate is Boolean. Thus, here one only uses polynomials of type $f_i(\boldsymbol{\chi})=A(\boldsymbol{\chi})^2-A(\boldsymbol{\chi})$, where $A(\boldsymbol{\chi})=\sum a_{ij}\chi_j$ for some coefficients a_{ij}.

In Table 5.2, we compare the resulting NIZK with the optimized Groth-Sahai proof for Boolean circuits by Ghadafi *et al.* [GSW09]. Here, m is the number of wires and n is the number of gates. In the case of the AES circuit described in [GSW09], $m=33880$ and $n=34136$. Assuming $|\mathbb{G}_2|=2|\mathbb{G}_1|$ and $\mathfrak{e}_2=2\mathfrak{e}_1$, we get that the NIZK of [GSW09] has commitment length $203283|\mathbb{G}_1|$, argument length $814662|\mathbb{G}_1|$, prover's computation $1629324\mathfrak{e}_1$, and verifier's computation $1630336\mathfrak{p}$. The new NIZK has commitment length $67760|\mathbb{G}_1|$, argument length $680160|\mathbb{G}_1|$, and prover's computation $884208\mathfrak{e}_1$, and verifier's computation $884208\mathfrak{p}$. Hence, the new NIZK has 3 times shorter commitments, 20% shorter arguments, and 1.84 times smaller prover's and verifier's computation.

Various Examples

Next, we give very generic background on generating sets and after that, we give some examples of the cases when it pays off directly to work with aCSPs (and not just arithmetic circuits) and then use the described methodology to construct the NIZK. We emphasize that one does not need a Gröbner basis and thus sometimes there exist smaller generating sets. In fact, there exist many alternative methods for constructing efficient aCSPs not directly related to generating sets at all; and the Gröbner basis technique is just one of them — albeit one that is strongly related to our general emphasis on polynomial ideals. As we see from the examples, the efficiency of NIZK depends on a delicate balance between the size of the generating set and the degree of the polynomials in that set. Really, it follows from Lemma 5.4.1 that if the generating set contains polynomials F_i for which QDRs have sizes ℓ_i, then the resulting NIZK has communication complexity $(2\sum \ell_i)(|\mathbb{G}_1| + |\mathbb{G}_2|) - \tau|\mathbb{G}_2|$.

Basic Background on Generating Sets. Generating sets of an ideal can have vastly different cardinality. For example, $\mathbb{Z}$ is generated by either $\{1\}$ or by the set of all primes. Since a Gröbner basis [Buc65] is, in particular, a generating set, one convenient way of finding a generating set is by using a Gröbner basis algorithm; however, such algorithms assume that one already knows a generating set. Fortunately, the Buchberger-Möller algorithm [MB82] (as say implemented by `CoCoA`[8]) can compute a Gröbner basis for $\mathcal{I}(\mathcal{A})$, given any finite set $\mathcal{A}$.

Worst-Case Multi-Dimensional Set-Membership Proof. We performed an exhaustive computer search to come up with an example of a 3-dimensional set of five points that has the least efficient NIZK argument in our framework. One of the examples we found[9] is

$$\mathcal{A} = \{(2,5,1),(2,4,2),(2,5,3),(1,2,4),(3,1,5)\} \ .$$

In this case, we found a reduced degree-lexicographic Gröbner basis

$$\left\{ \begin{array}{l} (y-z-2)(y+z-6), \frac{1}{18}(6x(3y-5)-37y+(z-4)z+68), \\ \frac{1}{9}\left(9x^2-33x+y-(z-4)z+22\right), \frac{1}{3}(-12x+5y+z(z(3z-23)+53)-34) \end{array} \right\}$$

that consists of three quadratic and one cubic polynomials. Clearly, here, each degree-d polynomial has an optimal-size $d \times d$ QDR. In the only non-trivial case (the cubic polynomial), one can use the matrix

$$\boldsymbol{C}_4(x,y,z) = \begin{pmatrix} \overset{z}{53/3} & \overset{1}{23/3-z} & \overset{0}{-4} \\ x-5y/12+17/6 & 0 & -z \end{pmatrix} \ .$$

[8] `http://cocoa.dima.unige.it/`

[9] In the case of many other sets, the NIZK will be much more efficient. We will provide one concrete example in Section 5.E.

Thus, one can construct a NIZK argument with communication of $2(2+2+2+3) = 18$ elements of $\mathbb{G}_1$ and $18-4 = 14$ elements of $\mathbb{G}_2$. Since, usually, elements of $\mathbb{G}_2$ are twice as long as elements of $\mathbb{G}_1$, it means that, in the worst case, such a NIZK argument will only be 4.6 times longer than a single OR proof. This is also the upper bound on the NIZK communication according to our exhaustive search, further discussion would be outside the scope of the current paper.

The most efficient known alternative seems to add (structure-preserving) signatures (SPSs) of 5 points to the CRS, letting the prover encrypt a signature of the chosen point, and then proving that the encrypted value is a valid signature of some point. (See, e.g., [RKP09].) This alternative has both a much larger CRS and worse concrete complexity compared to our NIZK argument. Moreover, it assumes that the underlying signature scheme is unforgeable.

Range proofs. In Section 5.C, we will show how to use our techniques to construct range proofs, i.e., proofs that the committed value χ belongs to some interval $[0, N]$. Couteau and Hartmann's approach can be used to propose range proofs of efficiency $\Theta(\log N)$ by using the binary decomposition of χ. In Section 5.C, we note that the use of the NIZK from Section 5.6 helps us to obtain a NIZK with better verifier's computation.

5.9 Back to Algebraic Languages

The well-known methodology of diverse vector spaces (DVSs, [Ben+13; Ben16]) has been used to successfully create efficient smooth projective hash functions (SPHFs) for algebraic languages. Moreover, by now several constructions of NIZKs based on such SPHFs are known, [ABP15; CH20]. For all such constructions, the first step is to construct language parameters $\boldsymbol{\Gamma}$ and $\boldsymbol{\theta}$ (see Section 5.2). Unfortunately, existing constructions of the language parameters are all somewhat ad hoc.

Next, we improve on the situation by proposing a methodology to construct $(\boldsymbol{\Gamma}, \boldsymbol{\theta})$ for any $\mathcal{L}_{\mathsf{pk},\mathcal{A}}$, where $\mathcal{A}$ is any algebraic set for which Section 5.8 results in an efficient NIZK. We start the process from a QDR $\boldsymbol{C}_i$ of F_i, where $\langle F_1, \ldots, F_\tau \rangle$ is some generating set of $\mathcal{I}(\mathcal{A})$, and output concrete parameters $(\boldsymbol{\Gamma}, \boldsymbol{\theta})$. The problem of constructing such $\boldsymbol{C}_i$ was already tackled in the current paper, with many examples (including the case when $\boldsymbol{C}_i$ is based on an ABP). As the end result, we construct explicit language parameters $(\boldsymbol{\Gamma}, \boldsymbol{\theta})$ for a variety of languages where no such small parameters were known before. Moreover, even in the simple case of univariate polynomials, where previous solutions were known [Ben+13; CH20], the new parameters are smaller than before.

We consider various NIZKs that one can construct for given $(\boldsymbol{\Gamma}, \boldsymbol{\theta})$. For every fixed $(\boldsymbol{\Gamma}, \boldsymbol{\theta})$, the NIZK from Section 5.4 is more efficient than the QA-NIZK of [ABP15] and usually more efficient than the CHM NIZK of [CH20]. Finally, we briefly discuss resulting GL-SPHFs [GL03] based on the new language parameters.

Preliminaries. We describe the CHM (Couteau-Hartmann-Maurer) Σ-protocol and the resulting NIZK in Section 5.F. There, we will also state the efficiency of their construction as a function of $(\boldsymbol{\Gamma}, \boldsymbol{\theta})$. We also restate Theorem 18 from [CH20] about the security of the CHM NIZK.

On Algebraic Languages for Elgamal Ciphertexts

Next, we derive language parameters $\boldsymbol{\Gamma}$ and $\boldsymbol{\theta}$ for an arbitrary $\mathcal{L}_{\mathsf{pk},F}$, such that $\boldsymbol{\theta}(\mathtt{x}) \in \operatorname{colspace} \boldsymbol{\Gamma}(\mathtt{x})$ iff $\mathtt{x} \in \mathcal{L}_{\mathsf{pk},F}$. In the case where $\mathcal{I}(\mathcal{A}) = \langle F_1, \ldots, F_\tau \rangle$ is not a principal ideal, one can then "concatenate" all τ parameters $\boldsymbol{\Gamma}(\mathtt{x})$ and $\boldsymbol{\theta}(\mathtt{x})$.

We start the derivation from the equation $\boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{w} = \boldsymbol{h}(\boldsymbol{\chi})$ in Figure 5.1. To simplify notation, let $\mathcal{E}(\chi; r) := \mathsf{Enc}([\chi]_1; r)^\top \in \mathbb{G}_1^2$ be a transposed ciphertext. Let $\mathcal{E}(\boldsymbol{T}(\boldsymbol{\chi}))$ (resp., $\mathcal{E}(\boldsymbol{h}(\boldsymbol{\chi}))$) denote an element-wise (transposed) encryption of $\boldsymbol{T}(\boldsymbol{\chi})$ (resp., $\boldsymbol{h}(\boldsymbol{\chi})$), where χ_i is encrypted by using randomizer r_i (that is, χ_i is "replaced" by $[\mathsf{ct}_i]_1^\top$) and constants are encrypted by using the randomizer 0. We define $[\boldsymbol{\Gamma}(\mathtt{x})]_1$ and $[\boldsymbol{\theta}(\mathtt{x})]_1$ as follows:

$$[\boldsymbol{\Gamma}(\mathtt{x})]_1 = (\mathcal{E}(\boldsymbol{T}(\boldsymbol{\chi})) \| \mathcal{E}(\mathbf{0}_{d\times d}; \boldsymbol{I}_d)) \in \mathbb{G}_2^{2d\times(2d-1)} \ , \ [\boldsymbol{\theta}(\mathtt{x})]_1 = \mathcal{E}(\boldsymbol{h}(\boldsymbol{\chi})) \in \mathbb{G}_2^{2d} \ . \tag{5.4}$$

Thus, $[\boldsymbol{\Gamma}]_1 \boldsymbol{w}^* = [\boldsymbol{\theta}]_1$ is an "encrypted" version of $\boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{w} = \boldsymbol{h}(\boldsymbol{\chi})$, where $[\boldsymbol{\Gamma}]_1$ contains additional columns and $\boldsymbol{w}^*$ contains additional rows (compared to $\boldsymbol{w}$) to take into account the randomizers used to encrypt χ_i. Note that $\mathcal{E}(\boldsymbol{C}(\boldsymbol{\chi})) = \mathcal{E}(\sum \boldsymbol{P}_k \chi_k + \boldsymbol{Q}; \sum \boldsymbol{P}_k r_k)$.

Example 5.9.1. Let $F(X) = (X-0)(X-1)$, and thus $d = 2$. Recall that then $\boldsymbol{C}(\chi) = \left(\begin{smallmatrix} \chi & -1 \\ 0 & \chi-1 \end{smallmatrix}\right)$ and thus $\boldsymbol{T}(\chi) = \left(\begin{smallmatrix} -1 \\ \chi-1 \end{smallmatrix}\right)$ and $\boldsymbol{h}(\chi) = \left(\begin{smallmatrix} \chi \\ 0 \end{smallmatrix}\right)$. Since $\mathsf{Enc}([0]_1; 1) = [1 \| \mathsf{sk}]_1$ and $\mathsf{Enc}([0]_1; 0) = [0 \| 0]_1$, Equation (5.4) results in

$$[\boldsymbol{\Gamma}]_1 = \left(\begin{array}{c||cc} \mathcal{E}(-1;0) & \mathcal{E}(0;1) & \mathcal{E}(0;0) \\ \mathcal{E}(\chi-1;r) & \mathcal{E}(0;0) & \mathcal{E}(0;1) \end{array}\right) = \left[\begin{array}{c||cc} 0 & 1 & 0 \\ -1 & \mathsf{sk} & 0 \\ \mathsf{ct}_1 & 0 & 1 \\ \mathsf{ct}_2 - 1 & 0 & \mathsf{sk} \end{array}\right]_1 \in \mathbb{G}_1^{4\times 3} \ , \quad [\boldsymbol{\theta}]_1 = \left[\begin{smallmatrix} \mathsf{ct}_1 \\ \mathsf{ct}_2 \\ 0 \\ 0 \end{smallmatrix}\right]_1 \ .$$

A variation of this $[\boldsymbol{\Gamma}, \boldsymbol{\theta}]_1$ was given in [Ben+13; CH20]. To motivate Theorem 5.9.2, note that $w_1^* = w = -\chi$ is a solution of $\boldsymbol{T}(\chi) w_1^* = \boldsymbol{h}(\chi)$. Setting $\hat{\boldsymbol{w}} := (w_2^* \| w_3^*)^\top = r\left(\begin{smallmatrix} 1 \\ -w_1^* \end{smallmatrix}\right) = r\left(\begin{smallmatrix} 1 \\ \chi \end{smallmatrix}\right)$ results in $\boldsymbol{\Gamma}\boldsymbol{w}^* - \boldsymbol{\theta} = (0 \| 0 \| 0 \| -\chi(\chi-1))^\top$, which is equal to $\mathbf{0}_4$ iff $\chi \in \{0, 1\}$.

Theorem 5.9.2. $\mathcal{L}_{\mathsf{pk},F} = \mathcal{L}_{\boldsymbol{\Gamma},\boldsymbol{\theta}}$.

Proof. **(1)** Assume $\mathtt{x} = \mathsf{Enc}(\boldsymbol{\chi}) \in \mathcal{L}_{\mathsf{pk},F}$. By the first column dependence property of Definition 5.3.1, there exists $\boldsymbol{w}$ such that $\boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{w} = \boldsymbol{h}(\boldsymbol{\chi})$, i.e., $\boldsymbol{C}(\boldsymbol{\chi})\left(\begin{smallmatrix} 1 \\ -\boldsymbol{w} \end{smallmatrix}\right) = \mathbf{0}$. To show that $\mathtt{x} \in \mathcal{L}_{\boldsymbol{\Gamma},\boldsymbol{\theta}}$, we need to construct $\boldsymbol{w}^*$ such that $\boldsymbol{\theta} = \boldsymbol{\Gamma}\boldsymbol{w}^*$. First, we set $w_i^* \leftarrow w_i$ for $i \le d-1$. This guarantees that $\mathsf{Dec}([\boldsymbol{\theta}]_1) = \mathsf{Dec}([\boldsymbol{\Gamma}]_1)\boldsymbol{w}^*$. Next, we have to set the remaining coefficients of w_i^* so that also the randomizers in $(\mathcal{E}(\boldsymbol{T}) \| \mathcal{E}(\mathbf{0}_{d\times d}; \boldsymbol{I}_d))\boldsymbol{w}^* = \mathcal{E}(\boldsymbol{h})$

match. Denoting $\hat{\boldsymbol{w}} = (w_d^*, \ldots, w_{2d-1}^*)^\top$, this is achieved by setting $\hat{\boldsymbol{w}} \leftarrow (\sum \boldsymbol{P}_k r_k)(\begin{smallmatrix}1\\ -\boldsymbol{w}\end{smallmatrix})$. Really, then

$$\begin{aligned}(\mathcal{E}(\boldsymbol{T})\|\mathcal{E}(\boldsymbol{0}_{d\times d};\boldsymbol{I}_d))\boldsymbol{w}^* - \mathcal{E}(\boldsymbol{h}(\boldsymbol{\chi})) =& \mathcal{E}\left(\boldsymbol{C}\right)\left(\begin{smallmatrix}-1\\ \boldsymbol{w}\end{smallmatrix}\right) + \mathcal{E}(\boldsymbol{0}_{d\times d};\boldsymbol{I}_d)\hat{\boldsymbol{w}}\\ =& \mathcal{E}\left(\boldsymbol{C};\sum \boldsymbol{P}_k r_k\right)\left(\begin{smallmatrix}-1\\ \boldsymbol{w}\end{smallmatrix}\right) + \mathcal{E}(\boldsymbol{0}_d;\hat{\boldsymbol{w}})\\ =& \mathcal{E}\left(\boldsymbol{0}_d;\left(\sum \boldsymbol{P}_k r_k\right)\left(\begin{smallmatrix}-1\\ \boldsymbol{w}\end{smallmatrix}\right) + \left(\sum \boldsymbol{P}_k r_k\right)\left(\begin{smallmatrix}1\\ -\boldsymbol{w}\end{smallmatrix}\right)\right)\\ =& \mathcal{E}(\boldsymbol{0}_d;\boldsymbol{0}_d)\ .\end{aligned}$$

(2) Assume that $\mathsf{x} = \mathsf{Enc}(\boldsymbol{\chi}) \in \mathcal{L}_{\boldsymbol{\Gamma},\boldsymbol{\theta}}$, and thus $[\boldsymbol{\theta}]_1 \in \mathrm{colspace}([\boldsymbol{\Gamma}]_1)$. Let $\boldsymbol{w}^*$ be such that $\boldsymbol{\theta} = \boldsymbol{\Gamma}\boldsymbol{w}^*$. After entry-wise decrypting, we get $\boldsymbol{\Gamma}^* = (\boldsymbol{T}(\boldsymbol{\chi})\|\boldsymbol{0})\boldsymbol{w}^* = \boldsymbol{h}(\boldsymbol{\chi})$. Let $\boldsymbol{w} = (w_1^*, \ldots, w_d^*)^\top$. Hence, $\boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{w} = \boldsymbol{h}(\boldsymbol{\chi})$, which means that $\boldsymbol{C}(\boldsymbol{\chi})(\begin{smallmatrix}-1\\ \boldsymbol{w}\end{smallmatrix}) = \boldsymbol{0}$. If $\mathsf{x} \notin \mathcal{L}_{\mathsf{pk},F}$ then $\det(\boldsymbol{C}(\boldsymbol{\chi})) \neq 0$. Since -1 is non-zero, this is a contradiction. □

In Section 5.F, we will give two more (lengthy) examples to illustrate how $\boldsymbol{w}^*$ is chosen.

Handling Non-Principal Ideals. Assume $\mathcal{I}(\mathcal{A})$ has a generating set $(F_1, \ldots, F_\tau)$ for $\tau > 1$, and that for each F_i, we have constructed the language parameter $\boldsymbol{\Gamma}_i, \boldsymbol{\theta}_i$. We can then construct the language parameter for $\mathcal{L}_{\mathsf{pk},\mathcal{A}}$ by using the well-known concatenation operation, setting

$$\boldsymbol{\Gamma} = \begin{pmatrix}\boldsymbol{\Gamma}_1 & \cdots & 0\\ \cdots & \cdots & \cdots\\ 0 & \cdots & \boldsymbol{\Gamma}_\tau\end{pmatrix} \text{ and } \boldsymbol{\theta} = \begin{pmatrix}\boldsymbol{\theta}_1\\ \cdots\\ \boldsymbol{\theta}_\tau\end{pmatrix}\ .$$

On the Couteau-Hartmann Disjunction. In Section 5.F, we describe the Couteau-Hartmann disjunction that results in $\boldsymbol{\Gamma}$ of size $(3d-1)\times(3d-2)$ and compare it to Equation (5.4). For the sake of completeness, we also reprove the efficiency of the CHM NIZK from [CH20].

Efficiency of Set-Membership NIZKs: Comparisons

In Table 5.1 we give a concrete efficiency comparison in the case of set-membership. This is motivated by the fact that this is probably the most complex language for which [CH20] provides a concrete NIZK with which we can compare our results. Because of the still large dimensions of $\boldsymbol{\Gamma}$, using the CHM Σ-protocol as in [CH20] for $\mathcal{L}_{\boldsymbol{\Gamma},\boldsymbol{\theta}} = \mathcal{L}_{\mathsf{pk},F}$ has quite a big overhead. Thus, the NIZK in Lemma 5.6.1 is quite a bit more efficient. However, it compares favorably to [CH20]. In the following lemma, we state its efficiency.

Lemma 5.9.3. *Let F be a univariate degree-d polynomial and let $\boldsymbol{C}(\boldsymbol{X})$ be the $\mathsf{abp}_{\mathsf{path}}$-based QDR of F from Section 5.6. Let $[\boldsymbol{\Gamma}]_1$ be constructed as in Equation (5.4). Then, the CHM NIZK argument requires $(5d-3)\mathfrak{e}_1 + 4d\mathfrak{e}_2$ from the prover, $7d-1$ pairings from the verifier, and $4d-1$ group elements.*

Proof. In this proof, we use the notation of Lemma 5.4.1. Note that

$$T_\Gamma = \{|(i,j)| : T_{ij} \neq 0\} + \{|(i,j)| \text{ s.t. } j > 1 : P_{kij} \neq 0 \text{ for some } k\} + 2\cdot\ell$$

and

$$T_\theta = \{|(i,j)| : h_{ij} \neq 0\} + \{|i| : P_{ki1} \neq 0 \text{ for some } k\} \ .$$

For a general $\boldsymbol{C}$, the efficiency estimate follows from Proposition 5.F.1 and the above formulas for T_Γ and T_θ. Hence, we only give concrete estimates for the case of univariate F.

The prover can compute $[\boldsymbol{\Gamma}(\mathtt{x})]_1\boldsymbol{r}$ in $T_\Gamma = 5d-3$ exponentiations in $\mathbb{G}_1$, and $[\boldsymbol{d}]_2$ in $2n = 2\cdot 2d = 4d$ exponentiations in $\mathbb{G}_2$. The verifier executes $T_\Gamma = 5d-3$ pairings to compute $[\boldsymbol{\Gamma}]_1 \bullet [\boldsymbol{d}]_2$, $T_\theta = 2$ pairings to compute $[\boldsymbol{\theta}(\mathtt{x})]_1 \bullet [e]_2$, and $n = 2d$ pairings to compute $[\boldsymbol{a}]_1 \bullet [1]_2$, in total $7d-1$ pairings. □

Note that the computation of the language parameters $\boldsymbol{\Gamma}, \boldsymbol{\theta}$ induces some cost. However, this computation is usually done once in advance. It is also not expensive, both in the case of the new NIZK and the CHM NIZK [CH20] requiring one to compute $[\xi_i]_1$ for each root ξ_i.

GL-SPHFs for Algebraic Sets

We give an example of GL-SPHFs (Gennaro-Lindell smooth projective hash functions [GL03]) based on the new $\mathtt{lpar} = (\boldsymbol{\Gamma}, \boldsymbol{\theta})$. We refer the reader to [CS02; Ben+13; Ben16] for a formal definition of GL-SPHFs. Briefly, recall that an SPHF is defined for a language parameter $\mathtt{lpar}$ and associated language $\mathcal{L}_{\mathtt{lpar}}$. A SPHF consists of an algorithm $\mathsf{hashkg}(\mathtt{lpar})$ to generate the private hashing key hk, an algorithm $\mathsf{projkg}(\mathtt{lpar}, \mathsf{hk})$ to generate a public projection key hp from hk, and two different hashing algorithms: $\mathsf{hash}(\mathtt{lpar}, \mathsf{hk}, \mathtt{x})$ that constructs an hash H, given the input $\mathtt{x}$ and hk, and $\mathsf{projhash}(\mathtt{lpar}, \mathsf{hp}, \mathtt{x}, \mathtt{w})$ that constructs a projection hash pH, given the input $\mathtt{x}$ and its witness $\mathtt{w}$. It is required that (1) $\mathsf{H} = \mathsf{pH}$ when $\mathtt{x} \in \mathcal{L}_{\mathtt{lpar}}$, and that (2) H looks random when $\mathtt{x} \notin \mathcal{L}_{\mathtt{lpar}}$, given $(\mathtt{lpar}, \mathsf{hp}, \mathtt{x})$.

In the GL-SPHFs [GL03], $\mathtt{lpar}$ and the projection key hp can depend on $\mathtt{x}$, while in other types of SPHFs, $\mathtt{x}$ is only chosen after $\mathtt{lpar}$ and hp are fixed. In the "DVS-based" constructions of SPHFs of [Ben+13], one starts with $[\boldsymbol{\Gamma}]_1 \in \mathbb{G}_1^{n\times t}$ and $[\boldsymbol{\theta}]_1 \in \mathbb{G}_1^n$ that may or may not depend on $\mathtt{x} = [\boldsymbol{\Gamma}]_1\mathtt{w}$. One samples a random $\mathsf{hk} = \boldsymbol{\alpha} \leftarrow\!\!\$\, \mathbb{Z}_p^n$, and sets $\mathsf{hp} \leftarrow \boldsymbol{\alpha}^\top[\boldsymbol{\Gamma}]_1$. For $\mathtt{x} = [\boldsymbol{\Gamma}]_1\mathtt{w}$, one computes $\mathsf{pH} = \mathsf{projhash}(\mathtt{lpar}, \mathsf{hp}, \mathtt{x}, \mathtt{w}) \leftarrow \mathsf{hp}\cdot\mathtt{w}$ and $\mathsf{H} = \mathsf{hash}(\mathtt{lpar}, \mathsf{hk}, \mathtt{x}) \leftarrow \mathsf{hk}\cdot\mathtt{x}$.

For any $\mathcal{A}(\mathcal{J})$ for which the NIZK of Section 5.4 is efficient, one can also construct an efficient SPHF by constructing $\boldsymbol{\Gamma}$ and $\boldsymbol{\theta}$ as in Equation (5.4).

Example 5.9.4 (GL-SPHF for the language of elliptic curve points.)**.** Let $\mathcal{A} = \{(X,Y) : Y^2 = X^3 + aX + b\}$ as in Section 5.6. Then, one can use $\mathtt{lpar} = (\boldsymbol{\Gamma}, \boldsymbol{\theta})$ from Example 5.F.4

to define $\mathsf{hk} \leftarrow\!\!\$\, \mathbb{Z}_p^8$, $\mathsf{hp} \leftarrow \boldsymbol{\alpha}^\top [\boldsymbol{\Gamma}]_1 =$

$$\begin{pmatrix} \alpha_3\mathsf{ct}_{11} + \alpha_4\mathsf{ct}_{12} + a\alpha_8 - \alpha_2, \alpha_7\mathsf{ct}_{11} + \alpha_8\mathsf{ct}_{12} - \alpha_4, -\alpha_7\mathsf{ct}_{21} - \alpha_8\mathsf{ct}_{22} - \alpha_6, \\ \alpha_1 + \alpha_2\mathsf{sk}, \alpha_3 + \alpha_4\mathsf{sk}, \alpha_5 + \alpha_6\mathsf{sk}, \alpha_7 + \alpha_8\mathsf{sk} \end{pmatrix}^\top,$$

and, in the case $\mathsf{x} \in \mathcal{L}_{\mathtt{lpar}}$, $\mathsf{pH} = \mathsf{H} = [\boldsymbol{\alpha}^\top \boldsymbol{\Gamma} \mathsf{w}]_1 =$

$$\begin{bmatrix} \chi_1 \left(-\alpha_3\mathsf{ct}_{11} - a\alpha_8 + \alpha_4\chi_1 + \alpha_2\right) - \chi_1 \left(\alpha_7\chi_1\mathsf{ct}_{11} + \mathsf{ct}_{12} \left(\alpha_8\chi_1 + \alpha_4\right)\right) + \\ \chi_2 \left(\alpha_7\mathsf{ct}_{21} + \alpha_8\mathsf{ct}_{22} + \alpha_6\right) + r_1 \left(\alpha_1 + \chi_1 \left(\alpha_3 + \chi_1 \left(\alpha_7 + \alpha_8\mathsf{sk}\right) + \alpha_4\mathsf{sk}\right) + \alpha_2\mathsf{sk}\right) + \\ r_2 \left(\alpha_5 - \chi_2 \left(\alpha_7 + \alpha_8\mathsf{sk}\right) + \alpha_6\mathsf{sk}\right) \end{bmatrix}_1.$$

5.10 On Falsifiability of CED

In the current paper, we significantly expand the class of languages for which the Couteau-Hartmann framework allows for the construction of efficient NIZKs. However, for many of these languages, the underlying variant of the CED assumption is not falsifiable in the sense of Naor [Nao03]. At first sight, even though the Couteau-Hartmann framework leads to particularly compact NIZKs, relying on a non-falsifiable assumption seems to limit the interest of the result severely: if one is willing to rely on non-falsifiable in the first place, then there are countless pairing-based SNARGs and SNARKs which will achieve much more compact proofs [Gro10; Lip12; Gen+13] (albeit the prover cost will be much higher in general).

Next, we discuss the falsifiability of the CED assumption. In Section 5.10, we study the falsifiable CED case, by clarifying for which languages there exist (algebraic) polynomial-time algorithms to check $F(\boldsymbol{\chi}) = 0$. In particular, we point out that for many examples of the current paper, the CED assumption is already falsifiable. After that, we concentrate on the cases when this is not so.

In Section 5.10, we show that despite their unfalsifiability, CED assumptions are fundamentally different in nature from knowledge-of-exponent assumptions (which underlie the security of existing SNARK candidates [Gro10; Lip12; Gen+13]). We will prove that CED assumptions are implied by a new but natural *gap assumption* [OP01] that KerMDH stays secure in $\mathbb{G}_2$ even given a CDH oracle in $\mathbb{G}_1$.

In Section 5.10, we modify our NIZKs to make the CED assumption falsifiable by letting the prover additionally encrypt input elements in $\mathbb{G}_2$. If the polynomial F is quadratic, then the soundness reduction can use them to check whether the prover's inputs belong to the language or not, thus making CED falsifiable. Since each gate of an arithmetic circuit is a quadratic polynomial, one can construct a NIZK for arithmetic circuits under a falsifiable assumption. The reason why we do not start with this solution is the added cost. First, the additional elements make the argument longer. Second, as probably expected, one cannot use Elgamal but has to use the less efficient DLIN cryptosystem [BBS04].

Thus, if CED is falsifiable, then one can use an Elgamal-based solution. Otherwise, one has a security-efficiency tradeoff: one can either rely on a non-falsifiable gap-assumption or use a slightly less efficient DLIN-based falsifiable NIZK.

On Languages for Which CED Is Falsifiable

The CED assumption is falsifiable if there exists an efficient verification algorithm $\mathsf{V_f}$, such that given an arbitrary ciphertext tuple $\mathsf{x} = [\mathbf{ct}_1, \ldots, \mathbf{ct}_\nu]_1$ and an sk-dependent trapdoor $\mathbf{T}$, $\mathsf{V_f}(\mathsf{p}, \mathsf{pk}, \mathsf{x}, \mathbf{T})$ can efficiently check whether $\mathsf{Dec}_{\mathsf{sk}}([\mathbf{ct}_1, \ldots, \mathbf{ct}_\nu]_1) \in \mathcal{L}_{\mathsf{pk},F}$. As in the rest of the paper, we take $\mathbf{T} = \mathsf{sk}$. Thus, given a ciphertext tuple $[\mathbf{ct}]_1$, $\mathsf{V_f}$ can use sk to decrypt it and obtain the plaintext $[\boldsymbol{\chi}]_1$. $\mathsf{V_f}$ then forms the QDR $[\boldsymbol{C}(\boldsymbol{\chi})]_1$ from $[\boldsymbol{\chi}]_1$. If $F(\boldsymbol{\chi}) \neq 0$ (that is, $\mathsf{x} \notin \mathcal{L}_{\mathsf{pk},F}$), then $[\boldsymbol{C}(\boldsymbol{\chi})]_1$ has full rank. Otherwise, it has rank $< \ell$. Thus, if $F(\boldsymbol{X})$ is such that it is possible to check efficiently whether $F(\boldsymbol{\chi}) = 0$, given $[\boldsymbol{\chi}]_1$, we can construct an efficient falsifiability check $\mathsf{V_f}$. (Note that this approach is different from Couteau-Hartmann, who required $\mathbf{T}$ to be a matrix.)

First, if $|\mathcal{A}| = \mathsf{poly}(\lambda)$, then $\mathsf{V_f}$ just checks if $[\boldsymbol{\chi}]_1$ is equal to $[\boldsymbol{a}]_1$ for any $\boldsymbol{a} \in \mathcal{A}$. Thus, the NIZK for the univariate case in Section 5.6 and the NIZK for boolean circuits in Section 5.8 rely on a falsifiable CED assumption. (This assumes that all polynomials have degree $\mathsf{poly}(\lambda)$, and the circuits are polynomial-size.) In general, the NIZK in the case of non-principal ideal, Section 5.8, is based on falsifiable CED iff $\mathcal{A}(\mathcal{I})$ has polynomial size.

The outliers are the cases of principal ideals of multivariate polynomials (since then $|\mathcal{A}(\mathcal{I})|$ can be exponential as in the set of points (X, Y) on an elliptic curve) and some instances of non-principal ideals where $|\mathcal{A}(\mathcal{I})|$ is super-polynomial. In the latter case, we can clarify the situation further. Namely, given a generating set $\langle F_1, \ldots, F_\tau \rangle$, by Bézout's theorem, $\mathcal{A}(\mathcal{I})$ has at most size $\prod \deg F_i$. Assuming each $\deg F_i$ is $\mathsf{poly}(\lambda)$, $\prod \deg F_i$ is super-polynomial if $\tau = \omega(1)$. Thus, constant-size set-membership arguments in Section 5.8 or aCSPs for constant-size arithmetic circuits in Section 5.8 are based on falsifiable CED. However, range proofs and superconstant-size arithmetic circuits are based on non-falsifiable CED.

The super-polynomial size of $\mathcal{A}(\mathcal{I})$ does not mean that efficient $\mathsf{V_f}$ does not exist. E.g., assume $F_j(\boldsymbol{X}) = \prod_i (X_i - s_j)$ for each j. The ideal $\langle F_j \rangle$, for a single j, has exponential size. However, given $[\boldsymbol{\chi}]_1$, one can check if $F_j(\boldsymbol{\chi}) = 0$ by checking if $\chi_i = s_j$ for some j. This can be generalized to the case F_j is a product of affine multivariate polynomials $\sum a_{ik} X_k + b_{ik}$. Clearly, $F(\boldsymbol{\chi}) = 0$ iff one of its affine factors is equal to 0. So, $\mathsf{V_f}$ can check if there exists an i such that $\sum a_{ik}[\chi_k]_1 + b_{ik}[1]_1 = [0]_1$. Generalizing this, one can efficiently establish whether $[\boldsymbol{C}]_1$ is full-rank if the Leibniz formula for the determinant, $\det(\boldsymbol{C}) = \sum_{\sigma \in S_n} (\mathrm{sgn}(\sigma) \prod_{i=1}^n C_{i,\sigma_i})$, contains only one non-zero addend.

On the other hand, since $\mathsf{V_f}$ has only access to $[\boldsymbol{\chi}]_1$, there is not much hope that the CED assumption is falsifiable if F is a product of irreducible polynomials, such that at

least one of them has a total degree greater than one, unless we add some additional, carefully chosen, elements to the proof for this purpose. In the general case, this is not efficient, but the number of additional needed elements might not be prohibitive for some applications.

Finally, the falsifiability of CED depends only on the polynomial F and not on the specific $\boldsymbol{C}$. One could find two different CED-matrices $\boldsymbol{C}_i$ for F, such that the first one results in a more efficient NIZK argument, but the second one has a specific structure enabling one to construct efficient V_f.

CED as a Gap Assumption

We show that CED follows from a new gap assumption, which states that given $\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$, even if one finds some structural properties in $\mathbb{G}_1$ that allows breaking CDH over this group, this does in general not guarantee an efficient algorithm for solving KerMDH [MRV16] over the other group $\mathbb{G}_2$. More formally:

Definition 5.10.1. Assume that the (exponential-time) oracle $\mathcal{O}([x,y]_1)$ outputs $[xy]_1$. $\mathcal{D}_{\ell-1,\mathsf{k}}$-$\mathsf{CDH}_{\mathbb{G}_1} \not\Rightarrow \mathsf{KerMDH}_{\mathbb{G}_2}$ holds relative to Pgen, if $\forall$ PPT $\mathcal{A}$,

$$\Pr\left[\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \boldsymbol{D} \leftarrow\!\!\$\, \mathcal{D}_{\ell-1,\mathsf{k}}; [\boldsymbol{c}]_{3-\iota} \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{p}, [\boldsymbol{D}]_\iota) : \boldsymbol{D}^\top \boldsymbol{c} = \boldsymbol{0}_\mathsf{k} \wedge \boldsymbol{c} \neq \boldsymbol{0}_{\ell-1}\right] \approx_\lambda 0 \ .$$

Theorem 5.10.2. *Let $\ell - 1, \mathsf{k} \in \mathbb{N}$. If the $\mathcal{D}_\mathsf{k}$-$\mathsf{CDH}_{\mathbb{G}_1} \not\Rightarrow \mathsf{KerMDH}_{\mathbb{G}_2}$ assumption holds relative to Pgen, then $\mathcal{D}_\mathsf{k}$-$(\ell-1)$-CED holds in $\mathbb{G}_1$ relative to Pgen.*

of Theorem 5.10.2. Let $\mathcal{A}$ be an CED adversary, as in Definition 5.2.1, that succeeds with a non-negligible probability $\varepsilon_\mathcal{A}$. We construct the following $\mathsf{CDH}_{\mathbb{G}_1} \not\Rightarrow \mathsf{KerMDH}_{\mathbb{G}_2}$ adversary $\mathcal{B}$.

$\mathcal{B}$ receives $\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$ and $[\boldsymbol{D}]_2 \leftarrow \mathcal{D}_\mathsf{k}$, and feeds them to $\mathcal{A}$. Assume $\mathcal{A}$ is successful. $\mathcal{B}$ obtains $([\boldsymbol{\gamma}\|\boldsymbol{C}]_1, [\boldsymbol{\delta}]_2) \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{D}]_2)$, where $\boldsymbol{\gamma} \in \mathbb{Z}_p^{\ell\times\mathsf{k}}$, $\boldsymbol{C} \in \mathbb{Z}_p^{\ell\times\ell}$, and $\boldsymbol{\delta} \in \mathbb{Z}_p^{(\ell-1)\times\mathsf{k}}$. Write

$$(\boldsymbol{\gamma}\|\boldsymbol{C}) = \left(\begin{smallmatrix} \boldsymbol{X}_L & \boldsymbol{X}_R \\ \boldsymbol{v}_L & \boldsymbol{v}_R \end{smallmatrix}\right) \ ,$$

where $\boldsymbol{X}_R \in \mathbb{Z}_p^{(\ell-1)\times(\ell-1)}$ and say $\boldsymbol{v}_L \in \mathbb{Z}_p^{1\times(\mathsf{k}+1)}$. Since $\mathcal{A}$ is successful, we get $\mathrm{rk}(\boldsymbol{C}) \geq \ell$ and thus $\boldsymbol{X}_R$ is invertible. Next, $\mathcal{A}$'s winning condition $(\boldsymbol{\gamma}\|\boldsymbol{C})\left(\begin{smallmatrix}\boldsymbol{D}\\ \boldsymbol{\delta}\end{smallmatrix}\right) = \boldsymbol{0}$ rewrites to

$$\boldsymbol{X}_L \cdot \boldsymbol{D} + \boldsymbol{X}_R \cdot \boldsymbol{\delta} = 0 \ , \qquad \boldsymbol{v}_L \cdot \boldsymbol{D} + \boldsymbol{v}_R \cdot \boldsymbol{\delta} = 0 \ ,$$

which gives, when $\boldsymbol{X}_R$ is invertible, $\boldsymbol{D}^\top \boldsymbol{c} = \boldsymbol{0}$, where

$$\boldsymbol{c} \leftarrow (\boldsymbol{u}_L - \boldsymbol{u}_R \cdot \boldsymbol{X}_R^{-1} \cdot \boldsymbol{X}_L)^\top \in \mathbb{Z}_p^{\mathsf{k}+1} \ .$$

Since[10] $\mathrm{rk}(\boldsymbol{C}) \geq \ell$, we get $\boldsymbol{c} \neq \boldsymbol{0}$. Using Gaussian elimination, one can compute $\boldsymbol{c}$ by an arithmetic circuit over $\mathbb{Z}_p$. Thus, $\mathcal{B}$ can compute $[\boldsymbol{c}]_1$ from $[\boldsymbol{\gamma}\|\boldsymbol{C}]_1$ with the help of $\mathcal{O}$

[10] Note that this is the point where we need to use CED instead of $\mathsf{ExtKerMDH}$ since we cannot deduce $\boldsymbol{c} \neq \boldsymbol{0}$ from $\mathrm{rk}(\boldsymbol{\gamma}\|\boldsymbol{C}) \geq \ell$.

that allows it to multiply exponents over $\mathbb{G}_1$. $\mathcal{B}$ returns $[\boldsymbol{c}]_1$ to the challenger. Clearly, $\mathcal{B}$ breaks KerMDH with probability $\varepsilon_{\mathcal{A}}$. □

Note that in particular, this re-proves the result of [CH20] that CED is secure in the generic bilinear group model (since a CDH oracle in $\mathbb{G}_1$ does not help to break any assumption in $\mathbb{G}_2$ in the generic bilinear group model).

DLIN-Based NIZK Based on Falsifiable CED

While constructing a Sub-ZK QA-NIZK, [Abd+20] had to check efficiently if $\boldsymbol{C}$ is invertible, given only $[\boldsymbol{C}]_1$. We will next study whether we can apply their technique. It is not straightforward to apply it since their case is somewhat different: there, $\boldsymbol{C}$ is a $\mathsf{k} \times \mathsf{k}$ (in particular, $\mathsf{k} \in \{1, 2\}$) public matrix sampled from $\mathcal{D}_{\mathsf{k}}$ and then given as a part of the CRS. In our case, $\boldsymbol{C}$ can have an arbitrary $\mathsf{poly}(\lambda)$ dimension, and it is reconstructed from the input to the NIZK argument.

To explain the technique of [Abd+20], consider the case $[\boldsymbol{C}]_1 \in \mathbb{G}_1^{2\times 2}$. [Abd+20] added to the CRS certain additional elements in $\mathbb{G}_2$ (namely, $[C_{11}, C_{12}]_2$), such that it became possible to check publicly (by using pairings) whether $\det \boldsymbol{C} = 0$ by checking whether $[C_{11}]_1 \bullet [1]_2 = [1]_1 \bullet [C_{11}]_2$, $[C_{12}]_1 \bullet [1]_2 = [1]_1 \bullet [C_{12}]_2$, and $[C_{22}]_1 \bullet [C_{11}]_2 = [C_{21}]_1 \bullet [C_{12}]_2$. One cost of publishing the additional elements in [Abd+20] was that it changed the assumption they used from KerMDH to the less standard SKerMDH assumption [GHR15]. As we see next, we have to use the DLIN cryptosystem [BBS04] instead of the Elgamal cryptosystem. However, as a result, we will obtain a NIZK for any F, computable by a poly-size arithmetic circuit, sound under a falsifiable CED assumption. Another benefit of it is to demonstrate that our framework is not restricted to Elgamal encryptions.

Next, we show how to construct a NIZK, based on a falsifiable CED assumption, for the polynomial $F(X, Y) = X^2 - Y$. We ask the prover to also encrypt X in $\mathbb{G}_2$. In the soundness reduction, a CED-adversary uses the latter, after decryption, to check whether $[X]_1 \bullet [X]_2 = [Y]_1 \bullet [1]_2$. We must ensure that the verifier only accepts the proof if $[X]_2$ is correct, i.e., $[X]_1 \bullet [1]_2 = [1]_1 \bullet [X]_2$. Since Elgamal is not secure given symmetric pairings, we cannot use the secret key or the same randomness in both groups. Hence, we use the DLIN encryption scheme. Given $\mathsf{sk} = (\mathsf{sk}_1, \mathsf{sk}_2)$ and $\mathsf{pk}_\iota = [1 \| \mathsf{sk}_1 \| \mathsf{sk}_2]_\iota$, we define $\mathtt{lpar} := (\mathsf{pk}_1, \mathsf{pk}_2, F)$. Then, $\mathcal{L}_{\mathtt{lpar}} := \{([\mathbf{ct}_1, \mathbf{ct}_2]_1, [\mathbf{ct}_1]_2)\}$, where

$$[\mathbf{ct}_1]_\iota = \mathsf{Enc}_\iota(X; r_1, r_2) = [r_1 \mathsf{sk}_1 \| r_2 \mathsf{sk}_2 \| X + r_1 + r_2]_\iota$$

and

$$[\mathbf{ct}_2]_1 = \mathsf{Enc}_1(Y; r_3, r_4) = [r_3 \mathsf{sk}_1 \| r_4 \mathsf{sk}_2 \| Y + r_3 + r_4]_1 \ .$$

We prove that $[\mathbf{ct}_1, \mathbf{ct}_2]_1$ are encryptions of X and Y such that $X^2 = Y$, by using the QDR $\boldsymbol{C}(X, Y) = \left(\begin{smallmatrix} X & -1 \\ -Y & X \end{smallmatrix}\right)$. The use of the DLIN encryption scheme just affects the efficiency

and the communication size of the protocol. In addition, one can check that $[\mathbf{ct}_1]_1$ and $[\mathbf{ct}_1]_2$ encrypt the same X in two different groups by checking that $[\mathbf{ct}_1]_1 \bullet [1]_2 = [1]_1 \bullet [\mathbf{ct}_1]_2$.

Since the DLIN encryption is doubly-homomorphic like Elgamal, then the argument of Section 5.4 stays essentially the same, with Elgamal encryptions replaced by DLIN encryptions, and the dimensions of randomizers and ciphertexts increasing slightly. In the soundness proof, given that the prover also outputs $\mathsf{Enc}_2(X; r_1, r_2)$, the constructed CED adversary obtains plaintexts $[X, Y]_1, [Z]_2$ and, then can efficiently verify if the statement $X^2 = Y$ holds.

Combining this idea with the rest of our framework, we can construct a NIZK for any language of DLIN-encryptions for any F, based on a falsifiable CED assumption. This is since one can check that $F = 0$ by checking that an arithmetic circuit evaluates to 0, and each gate of an arithmetic circuit evaluates a quadratic function. For example, to prove that $Y^2 = X^3 + aX + b$, one can encrypt Y, Y', X, X', and X'', and then prove that $Y' = Y^2$, $X' = X^2$, $X'' = XX'$, and $Y' = X'' + aX + b$.

Acknowledgment. Geoffroy Couteau was partially supported by the ANR SCENE.

References

[Abd+20] Behzad Abdolmaleki, Helger Lipmaa, Janno Siim, and Michal Zajac. "On QA-NIZK in the BPK Model". In: *PKC 2020, Part I*. Ed. by Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas. Vol. 12110. LNCS. Springer, Heidelberg, May 2020, pp. 590–620. DOI: 10.1007/978-3-030-45374-9_20.

[ABP15] Michel Abdalla, Fabrice Benhamouda, and David Pointcheval. "Disjunctions for Hash Proof Systems: New Constructions and Applications". In: *EUROCRYPT 2015, Part II*. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9057. LNCS. Springer, Heidelberg, Apr. 2015, pp. 69–100. DOI: 10.1007/978-3-662-46803-6_3.

[ALO15] David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. 4th ed. Undergraduate Texts in Mathematics. Springer, May 2015, p. 662.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. "Short Group Signatures". In: *CRYPTO 2004*. Ed. by Matthew Franklin. Vol. 3152. LNCS. Springer, Heidelberg, Aug. 2004, pp. 41–55. DOI: 10.1007/978-3-540-28628-8_3.

[Bea00] Arnaud Beauville. "Determinantal Hypersurfaces". In: *Michigan Math. J.* 48.1 (2000), pp. 39–64.

[Ben+13] Fabrice Benhamouda, Olivier Blazy, Céline Chevalier, David Pointcheval, and Damien Vergnaud. "New Techniques for SPHFs and Efficient One-Round PAKE Protocols". In: *CRYPTO 2013, Part I*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8042. LNCS. Springer, Heidelberg, Aug. 2013, pp. 449–475. DOI: 10.1007/978-3-642-40041-4_25.

[Ben+14] Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. "Scalable Zero Knowledge via Cycles of Elliptic Curves". In: *CRYPTO 2014, Part II*. Ed. by Juan A. Garay and Rosario Gennaro. Vol. 8617. LNCS. Springer, Heidelberg, Aug. 2014, pp. 276–294. DOI: 10.1007/978-3-662-44381-1_16.

[Ben16] Fabrice Ben Hamouda-Guichoux. "Diverse Modules and Zero-Knowledge". PhD thesis. PSL Research University, 2016.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. "Non-Interactive Zero-Knowledge and Its Applications (Extended Abstract)". In: *20th ACM STOC*. ACM Press, May 1988, pp. 103–112. DOI: 10.1145/62212.62222.

[BG99] Amos Beimel and Anna Gál. "On Arithmetic Branching Programs". In: *J. Comput. Syst. Sci.* 59.2 (1999), pp. 195–220.

[Bla+10] Olivier Blazy, Georg Fuchsbauer, Malika Izabachène, Amandine Jambert, Hervé Sibert, and Damien Vergnaud. "Batch Groth-Sahai". In: *ACNS 10*. Ed. by Jianying Zhou and Moti Yung. Vol. 6123. LNCS. Springer, Heidelberg, June 2010, pp. 218–235. DOI: 10.1007/978-3-642-13708-2_14.

[Bou00] Fabrice Boudot. "Efficient Proofs that a Committed Number Lies in an Interval". In: *EUROCRYPT 2000*. Ed. by Bart Preneel. Vol. 1807. LNCS. Springer, Heidelberg, May 2000, pp. 431–444. DOI: 10.1007/3-540-45539-6_31.

[Buc65] Bruno Buchberger. "An Algorithm for Finding the Basis Elements of the Residue Class Ring of a Zero Dimensional Polynomial Ideal". PhD thesis. University of Innsbruck, 1965.

[CC18] Pyrros Chaidos and Geoffroy Couteau. "Efficient Designated-Verifier Non-interactive Zero-Knowledge Proofs of Knowledge". In: *EUROCRYPT 2018, Part III*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10822. LNCS. Springer, Heidelberg, Apr. 2018, pp. 193–221. DOI: 10.1007/978-3-319-78372-7_7.

[CCs08] Jan Camenisch, Rafik Chaabouni, and abhi shelat. "Efficient Protocols for Set Membership and Range Proofs". In: *ASIACRYPT 2008*. Ed. by Josef Pieprzyk. Vol. 5350. LNCS. Springer, Heidelberg, Dec. 2008, pp. 234–252. DOI: 10.1007/978-3-540-89255-7_15.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. "Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols". In: *CRYPTO'94*. Ed. by Yvo Desmedt. Vol. 839. LNCS. Springer, Heidelberg, Aug. 1994, pp. 174–187. DOI: 10.1007/3-540-48658-5_19.

[CG15] Pyrros Chaidos and Jens Groth. "Making Sigma-Protocols Non-interactive Without Random Oracles". In: *PKC 2015*. Ed. by Jonathan Katz. Vol. 9020. LNCS. Springer, Heidelberg, Mar. 2015, pp. 650–670. DOI: 10.1007/978-3-662-46447-2_29.

[CH20] Geoffroy Couteau and Dominik Hartmann. "Shorter Non-interactive Zero-Knowledge Arguments and ZAPs for Algebraic Languages". In: *CRYPTO 2020, Part III*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12172. LNCS. Springer, Heidelberg, Aug. 2020, pp. 768–798. DOI: 10.1007/978-3-030-56877-1_27.

[CLs10] Rafik Chaabouni, Helger Lipmaa, and abhi shelat. "Additive Combinatorics and Discrete Logarithm Based Range Protocols". In: *ACISP 10*. Ed. by Ron Steinfeld and Philip Hawkes. Vol. 6168. LNCS. Springer, Heidelberg, July 2010, pp. 336–351.

[CLW18] Ran Canetti, Alex Lombardi, and Daniel Wichs. *Fiat-Shamir: From Practice to Theory, Part II (NIZK and Correlation Intractability from Circular-Secure FHE)*. Cryptology ePrint Archive, Report 2018/1248. `https://eprint.iacr.org/2018/1248`. 2018.

[CLZ12] Rafik Chaabouni, Helger Lipmaa, and Bingsheng Zhang. "A Non-interactive Range Proof with Constant Communication". In: *FC 2012*. Ed. by Angelos D. Keromytis. Vol. 7397. LNCS. Springer, Heidelberg, Feb. 2012, pp. 179–199.

[CS02] Ronald Cramer and Victor Shoup. "Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption". In: *EUROCRYPT 2002*. Ed. by Lars R. Knudsen. Vol. 2332. LNCS. Springer, Heidelberg, Apr. 2002, pp. 45–64. DOI: `10.1007/3-540-46035-7_4`.

[Daz+19] Vanesa Daza, Alonso González, Zaira Pindado, Carla Ràfols, and Javier Silva. "Shorter Quadratic QA-NIZK Proofs". In: *PKC 2019, Part I*. Ed. by Dongdai Lin and Kazue Sako. Vol. 11442. LNCS. Springer, Heidelberg, Apr. 2019, pp. 314–343. DOI: `10.1007/978-3-030-17253-4_11`.

[DGP08] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. "Pairings for Cryptographers". In: *Discrete Applied Mathematics* 156.16 (2008), pp. 3113–3121.

[Dic21] Leonard Eugene Dickson. "Determination of All General Homogeneous Polynomials Expressible as Determinants with Linear Elements". In: *Trans. of the American Mathematical Society* 22.2 (Apr. 1921), pp. 167–179.

[E K+15] Ahmed E. Kosba, Zhichao Zhao, Andrew Miller, Yi Qian, T.-H. Hubert Chan, Charalampos Papamanthou, Rafael Pass, Abhi Shelat, and Elaine Shi. *C∅C∅: A Framework for Building Composable Zero-Knowledge Proofs*. Tech. rep. 2015/1093. `https://ia.cr/2015/1093`, last accessed version 9 Apr 2017. IACR, Nov. 2015.

[EG14] Alex Escala and Jens Groth. "Fine-Tuning Groth-Sahai Proofs". In: *PKC 2014*. Ed. by Hugo Krawczyk. Vol. 8383. LNCS. Springer, Heidelberg, Mar. 2014, pp. 630–649. DOI: `10.1007/978-3-642-54631-0_36`.

[ElG84] Taher ElGamal. "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms". In: *CRYPTO'84*. Ed. by G. R. Blakley and David Chaum. Vol. 196. LNCS. Springer, Heidelberg, Aug. 1984, pp. 10–18.

[Esc+13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. "An Algebraic Framework for Diffie-Hellman Assumptions". In: *CRYPTO 2013, Part II*. Ed. by Ran Canetti and Juan A. Garay. Vol. 8043. LNCS. Springer, Heidelberg, Aug. 2013, pp. 129–147. DOI: 10.1007/978-3-642-40084-1_8.

[FS87] Amos Fiat and Adi Shamir. "How to Prove Yourself: Practical Solutions to Identification and Signature Problems". In: *CRYPTO'86*. Ed. by Andrew M. Odlyzko. Vol. 263. LNCS. Springer, Heidelberg, Aug. 1987, pp. 186–194. DOI: 10.1007/3-540-47721-7_12.

[Gen+13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. "Quadratic Span Programs and Succinct NIZKs without PCPs". In: *EUROCRYPT 2013*. Ed. by Thomas Johansson and Phong Q. Nguyen. Vol. 7881. LNCS. Springer, Heidelberg, May 2013, pp. 626–645. DOI: 10.1007/978-3-642-38348-9_37.

[GHR15] Alonso González, Alejandro Hevia, and Carla Ràfols. "QA-NIZK Arguments in Asymmetric Groups: New Tools and New Constructions". In: *ASIACRYPT 2015, Part I*. Ed. by Tetsu Iwata and Jung Hee Cheon. Vol. 9452. LNCS. Springer, Heidelberg, Nov. 2015, pp. 605–629. DOI: 10.1007/978-3-662-48797-6_25.

[GL03] Rosario Gennaro and Yehuda Lindell. "A Framework for Password-Based Authenticated Key Exchange". In: *EUROCRYPT 2003*. Ed. by Eli Biham. Vol. 2656. LNCS. https://eprint.iacr.org/2003/032.ps.gz. Springer, Heidelberg, May 2003, pp. 524–543. DOI: 10.1007/3-540-39200-9_33.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. "The Knowledge Complexity of Interactive Proof Systems". In: *SIAM Journal on Computing* 18.1 (1989), pp. 186–208.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. "Non-interactive Zaps and New Techniques for NIZK". In: *CRYPTO 2006*. Ed. by Cynthia Dwork. Vol. 4117. LNCS. Springer, Heidelberg, Aug. 2006, pp. 97–111. DOI: 10.1007/11818175_6.

[Gro10] Jens Groth. "Short Pairing-Based Non-interactive Zero-Knowledge Arguments". In: *ASIACRYPT 2010*. Ed. by Masayuki Abe. Vol. 6477. LNCS. Springer, Heidelberg, Dec. 2010, pp. 321–340. DOI: 10.1007/978-3-642-17373-8_19.

[GS08] Jens Groth and Amit Sahai. "Efficient Non-interactive Proof Systems for Bilinear Groups". In: *EUROCRYPT 2008*. Ed. by Nigel P. Smart. Vol. 4965. LNCS. Springer, Heidelberg, Apr. 2008, pp. 415–432. DOI: 10.1007/978-3-540-78967-3_24.

[GSW09] Essam Ghadafi, Nigel P. Smart, and Bogdan Warinschi. "Practical Zero-Knowledge Proofs for Circuit Evaluation". In: *12th IMA International Conference on Cryptography and Coding*. Ed. by Matthew G. Parker. Vol. 5921. LNCS. Springer, Heidelberg, Dec. 2009, pp. 469–494.

[Har92] Joe Harris. *Algebraic Geometry: A First Course*. Vol. 133. Graduate Texts in Mathematics. Springer-Verlag, 1992, p. 349. ISBN: 978-1441930996.

[HKR19] Max Hoffmann, Michael Klooß, and Andy Rupp. "Efficient Zero-Knowledge Arguments in the Discrete Log Setting, Revisited". In: *ACM CCS 2019*. Ed. by Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz. ACM Press, Nov. 2019, pp. 2093–2110. DOI: 10.1145/3319535.3354251.

[IK00] Yuval Ishai and Eyal Kushilevitz. "Randomizing Polynomials: A New Representation with Applications to Round-Efficient Secure Computation". In: *41st FOCS*. IEEE Computer Society Press, Nov. 2000, pp. 294–304. DOI: 10.1109/SFCS.2000.892118.

[IK02] Yuval Ishai and Eyal Kushilevitz. "Perfect Constant-Round Secure Computation via Perfect Randomizing Polynomials". In: *ICALP 2002*. Ed. by Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan Eidenbenz, and Ricardo Conejo. Vol. 2380. LNCS. Springer, Heidelberg, July 2002, pp. 244–256. DOI: 10.1007/3-540-45465-9_22.

[IW14] Yuval Ishai and Hoeteck Wee. "Partial Garbling Schemes and Their Applications". In: *ICALP 2014, Part I*. Ed. by Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias. Vol. 8572. LNCS. Springer, Heidelberg, July 2014, pp. 650–662. DOI: 10.1007/978-3-662-43948-7_54.

[JR13] Charanjit S. Jutla and Arnab Roy. "Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces". In: *ASIACRYPT 2013, Part I*. Ed. by Kazue Sako and Palash Sarkar. Vol. 8269. LNCS. Springer, Heidelberg, Dec. 2013, pp. 1–20. DOI: 10.1007/978-3-642-42033-7_1.

[Kiy20] Susumu Kiyoshima. "Round-Optimal Black-Box Commit-and-Prove with Succinct Communication". In: *CRYPTO 2020, Part II*. Ed. by Daniele Micciancio and Thomas Ristenpart. Vol. 12171. LNCS. Springer, Heidelberg, Aug. 2020, pp. 533–561. DOI: 10.1007/978-3-030-56880-1_19.

[KOS18] Dakshita Khurana, Rafail Ostrovsky, and Akshayaram Srinivasan. "Round Optimal Black-Box "Commit-and-Prove"". In: *TCC 2018, Part I*. Ed. by Amos Beimel and Stefan Dziembowski. Vol. 11239. LNCS. Springer, Heidelberg, Nov. 2018, pp. 286–313. DOI: 10.1007/978-3-030-03807-6_11.

[KW15] Eike Kiltz and Hoeteck Wee. "Quasi-Adaptive NIZK for Linear Subspaces Revisited". In: *EUROCRYPT 2015, Part II*. Ed. by Elisabeth Oswald and Marc Fischlin. Vol. 9057. LNCS. Springer, Heidelberg, Apr. 2015, pp. 101–128. DOI: 10.1007/978-3-662-46803-6_4.

[LAN03] Helger Lipmaa, N. Asokan, and Valtteri Niemi. "Secure Vickrey Auctions without Threshold Trust". In: *FC 2002*. Ed. by Matt Blaze. Vol. 2357. LNCS. Springer, Heidelberg, Mar. 2003, pp. 87–101.

[Lip03] Helger Lipmaa. "On Diophantine Complexity and Statistical Zero-Knowledge Arguments". In: *ASIACRYPT 2003*. Ed. by Chi-Sung Laih. Vol. 2894. LNCS. Springer, Heidelberg, Nov. 2003, pp. 398–415. DOI: 10.1007/978-3-540-40061-5_26.

[Lip12] Helger Lipmaa. "Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments". In: *TCC 2012*. Ed. by Ronald Cramer. Vol. 7194. LNCS. Springer, Heidelberg, Mar. 2012, pp. 169–189. DOI: 10.1007/978-3-642-28914-9_10.

[Lip16] Helger Lipmaa. "Prover-Efficient Commit-and-Prove Zero-Knowledge SNARKs". In: *AFRICACRYPT 16*. Ed. by David Pointcheval, Abderrahmane Nitaj, and Tajjeeddine Rachidi. Vol. 9646. LNCS. Springer, Heidelberg, Apr. 2016, pp. 185–206. DOI: 10.1007/978-3-319-31517-1_10.

[Mau09] Ueli M. Maurer. "Unifying Zero-Knowledge Proofs of Knowledge". In: *AFRICACRYPT 09*. Ed. by Bart Preneel. Vol. 5580. LNCS. Springer, Heidelberg, June 2009, pp. 272–286.

[MB82] H. Michael Möller and Bruno Buchberger. "The Construction of Multivariate Polynomials with Preassigned Zeros". In: *EUROCAM 1982*. Ed. by Jacques Calmet. Vol. 144. LNCS. Marseille, France: Springer, May 1982, pp. 24–31. ISBN: 3-540-11607-9.

[MRV16] Paz Morillo, Carla Ràfols, and Jorge Luis Villar. "The Kernel Matrix Diffie-Hellman Assumption". In: *ASIACRYPT 2016, Part I*. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10031. LNCS. Springer, Heidelberg, Dec. 2016, pp. 729–758. DOI: 10.1007/978-3-662-53887-6_27.

[Nao03] Moni Naor. "On Cryptographic Assumptions and Challenges (Invited Talk)". In: *CRYPTO 2003*. Ed. by Dan Boneh. Vol. 2729. LNCS. Springer, Heidelberg, Aug. 2003, pp. 96–109. DOI: 10.1007/978-3-540-45146-4_6.

[Nis91] Noam Nisan. "Lower Bounds for Non-Commutative Computation (Extended Abstract)". In: *23rd ACM STOC*. ACM Press, May 1991, pp. 410–418. DOI: 10.1145/103418.103462.

[OP01] Tatsuaki Okamoto and David Pointcheval. "The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes". In: *PKC 2001*. Ed. by Kwangjo Kim. Vol. 1992. LNCS. Springer, Heidelberg, Feb. 2001, pp. 104–118. DOI: 10.1007/3-540-44586-2_8.

[Pas13] Rafael Pass. "Unprovable Security of Perfect NIZK and Non-interactive Non-malleable Commitments". In: *TCC 2013*. Ed. by Amit Sahai. Vol. 7785. LNCS. Springer, Heidelberg, Mar. 2013, pp. 334–354. DOI: 10.1007/978-3-642-36594-2_19.

[PSV12] Daniel Plaumann, Bernd Sturmfels, and Cynthia Vinzant. "Computing Linear Matrix Representations of Helton-Vinnikov Curves". In: *Mathematical Methods in Systems, Optimization, and Control Operator Theory* 222 (2012), pp. 259–277.

[Ràf15] Carla Ràfols. "Stretching Groth-Sahai: NIZK Proofs of Partial Satisfiability". In: *TCC 2015, Part II*. Ed. by Yevgeniy Dodis and Jesper Buus Nielsen. Vol. 9015. LNCS. Springer, Heidelberg, Mar. 2015, pp. 247–276. DOI: 10.1007/978-3-662-46497-7_10.

[RKP09] Alfredo Rial, Markulf Kohlweiss, and Bart Preneel. "Universally Composable Adaptive Priced Oblivious Transfer". In: *PAIRING 2009*. Ed. by Hovav Shacham and Brent Waters. Vol. 5671. LNCS. Springer, Heidelberg, Aug. 2009, pp. 231–247. DOI: 10.1007/978-3-642-03298-1_15.

[Sze20] Alan Szepieniec. *Polynomial IOPs for Linear Algebra Relations*. Cryptology ePrint Archive, Report 2020/1022. https://eprint.iacr.org/2020/1022. 2020.

[V D10] Igor V. Dolgachev. "Topics in Classical Algebraic Geometry". Sept. 2010. URL: https://www.math.ucsd.edu/~eizadi/207A-14/Dolgachev-topics.pdf.

5.A More on Section 5.2

Matrix Assumptions

The following assumptions are, while relatively recently formalized, very standard. In particular, MDDH generalizes DDH and KerMDH generalizes CDH. See [Esc+13; GHR15; MRV16] for more discussion.

Let $\iota \in \{1,2\}$. $\mathcal{D}_{\ell,\mathsf{k}}$-*MDDH*$_{\mathbb{G}_\iota}$ (Matrix Decisional Diffie-Hellman, [Esc+13]) holds relative to Pgen, if $\forall$ PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{mddh}}_{\mathcal{A},\mathsf{Pgen},\mathbb{G}_\iota,\mathcal{D}_{\ell,\mathsf{k}}}(\lambda) := |\varepsilon^0_{\mathcal{A}}(\lambda) - \varepsilon^1_{\mathcal{A}}(\lambda)| \approx_\lambda 0$, where

$$\varepsilon^b_{\mathcal{A}}(\lambda) := \Pr\left[\mathcal{A}(\mathsf{p}, [\boldsymbol{A}, \boldsymbol{y}]_\iota) = 1 \;\middle|\; \begin{array}{c} \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \boldsymbol{A} \leftarrow\!\!\$\, \mathcal{D}_{\ell,\mathsf{k}}; \mathtt{w} \leftarrow\!\!\$\, \mathbb{Z}_p^{\mathsf{k}}; \\ \textbf{if } b = 0 \textbf{ then } \boldsymbol{y} \leftarrow\!\!\$\, \mathbb{Z}_p^{\ell} \textbf{ else } \boldsymbol{y} \leftarrow \boldsymbol{A}\mathtt{w} \textbf{ fi} \end{array} \right] .$$

$\mathcal{D}_{\ell,\mathsf{k}}$-*KerMDH*$_{\mathbb{G}_\iota}$ (Kernel Diffie-Hellman, [MRV16]) holds relative to Pgen, if $\forall$ PPT $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{kermdh}}_{\mathcal{A},\mathcal{D}_{\ell,\mathsf{k}},\iota,\mathsf{Pgen}}(\lambda) :=$

$$\Pr\left[\boldsymbol{A}^\top \boldsymbol{c} = \boldsymbol{0}_{\mathsf{k}} \wedge \boldsymbol{c} \neq \boldsymbol{0}_\ell \;\middle|\; \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \boldsymbol{A} \leftarrow\!\!\$\, \mathcal{D}_{\ell,\mathsf{k}}; [\boldsymbol{c}]_{3-\iota} \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{A}]_\iota) \right] \approx_\lambda 0 .$$

$\mathcal{D}_{\ell,\mathsf{k}}$-*SKerMDH* (Split Kernel Diffie-Hellman, [GHR15]) holds relative to Pgen, if $\forall$ PPT $\mathcal{A}$,

$$\mathsf{Adv}^{\mathrm{skermdh}}_{\mathcal{A},\mathcal{D}_{\ell,\mathsf{k}},\mathsf{Pgen}}(\lambda) := \Pr\left[\begin{array}{c} \boldsymbol{A}^\top(\boldsymbol{c}_1 - \boldsymbol{c}_2) = \boldsymbol{0}_{\mathsf{k}} \wedge \\ \boldsymbol{c}_1 - \boldsymbol{c}_2 \neq \boldsymbol{0}_\ell \end{array} \;\middle|\; \begin{array}{c} \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \boldsymbol{A} \leftarrow\!\!\$\, \mathcal{D}_{\ell,\mathsf{k}}; \\ ([\boldsymbol{c}_1]_1, [\boldsymbol{c}_2]_2) \leftarrow \mathcal{A}(\mathsf{p}, [\boldsymbol{A}]_1, [\boldsymbol{A}]_2) \end{array} \right] \approx_\lambda 0 .$$

According to Lemma 4 of [MRV16], in a bilinear group, if $\mathcal{D}_{\ell,\mathsf{k}}$-MDDH holds then also $\mathcal{D}_{\ell,\mathsf{k}}$-KerMDH holds. According to Lemma 1 of [GHR15], if $\mathcal{D}_{\ell,\mathsf{k}}$-KerMDH holds in generic symmetric bilinear groups then $\mathcal{D}_{\ell,\mathsf{k}}$-SKerMDH holds in generic asymmetric bilinear groups.

5.B More on Section 5.3

Determinantal Representations

The following problem is well-studied in algebraic geometry, [Har92; V D10]. Given a homogeneous polynomial $f(X_0, \ldots, X_n)$ of degree-d find a $d \times d$ matrix $\boldsymbol{C}(\boldsymbol{X}) = (L_{ij}(\boldsymbol{X}))$ with affine maps as its entries such that

$$f(\boldsymbol{X}) = \det(L_{ij}(\boldsymbol{X})) .$$

The resulting equation $\det(\boldsymbol{C}(\boldsymbol{X})) = F(\boldsymbol{X})$ is known as F's *determinantal representation*.

More generally, one considers $\ell \times \ell$ matrices $\boldsymbol{C}(\boldsymbol{X})$ with the same property. In this case, the determinantal complexity $\mathsf{dc}(F)$ of the polynomial F is the minimal size of any determinantal representation of F. Clearly, $\mathsf{dc}(F) \geq \deg(F)$.

Table 5.3: The efficiency of new NIZK arguments for $\mathcal{L}_{\{0,1\}}$. The communication is given as (g_1, g_2, z), where g_ι is the number of $\mathbb{G}_\iota$ elements ($\iota = 1$ in the Σ-protocols) and z is the number of $\mathbb{Z}_p$ elements. The computation is given as (e_1, e_2, p), where e_ι is the number of exponentiations in $\mathbb{G}_\iota$ and p is the number of pairings.

Scheme	$\lvert\mathsf{crs}\rvert$	$\lvert\pi\rvert$	P comp	V comp	Assumpt.
$\boldsymbol{\Pi}^\vee_{\mathsf{simple}}$, $\boldsymbol{\Pi}^\vee_{\mathsf{cg}}$, $\boldsymbol{\Pi}^\vee_{\mathsf{cds}}$	$(0,1,0)$	$(4,3,0)$	$(5,4,0)$	$(0,0,13)$	CED

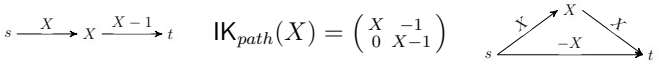


Figure 5.6: The matrices for the ABP-based simple (ABP $\mathsf{abp}^2_{path}(X, \{0,1\})$, left) and the ABP-based Chaidos-Groth (right) argument for $f(X) = X^2 - X = X(X-1)$ and the corresponding matrices.

All plane curves and cubic surfaces have determinantal complexity equal to their degree, [Dic21]. Dickson [Dic21] also proved a general theorem about the impossibility of determinantal representations of size $\deg(F)$ for general polynomials F. See [Dic21; Bea00] for more information. Moreover, efficient algorithms for finding determinantal representations, if they exist, have only been proposed lately [PSV12]; see also Section 5.8.

QDRs, as defined in Definition 5.3.1, additionally have the first column dependence property, which is not required for determinantal representations. Not every determinantal relation is a QDR (see Section 5.7 for some examples) and thus it is plausible that in general, $\mathsf{qdc}(F) > \mathsf{dc}(F)$.

5.C More on Section 5.6

On OR Proofs

$\boldsymbol{\Pi}^\vee_{\mathsf{simple}}$ and $\boldsymbol{\Pi}^\vee_{\mathsf{cg}}$. The NIZK argument $\boldsymbol{\Pi}^\vee_{\mathsf{simple}}$ (see Figure 5.7) for $\mathcal{L}_{\{0,1\}}$ follows from the approach in Section 5.6, by using $\mathsf{abp}^2_{\mathsf{path}}$.

On the other hand, $\boldsymbol{\Pi}^\vee_{\mathsf{cg}}$ (see Figure 5.7) follows from the approach in Section 5.6, given the ABP in Figure 5.6 (right). It is based roughly on the Chaidos-Groth Σ-protocol from [CG15], which itself is based on checking whether $X \cdot X = X$. We depict the ABPs and corresponding matrices $\mathsf{IK}(X)$ in in Figure 5.6. The correctness of both arguments follows from the fact that the solution of $\boldsymbol{T}(\chi)w = \boldsymbol{h}(\chi)$ is $w = -\chi$.

As seen from Figure 5.7, in both $\boldsymbol{\Pi}^\vee_{\mathsf{simple}}$ and $\boldsymbol{\Pi}^\vee_{\mathsf{cg}}$, the prover's computation is dominated by 5 exponentiations in $\mathbb{G}_1$ (to compute $[\boldsymbol{\gamma}]_1$; 5 is sufficient since $\gamma_2 \in \{-\gamma_1, 0, \gamma_1\}$) and 4 exponentiations in $\mathbb{G}_2$ (one to compute $y[1]_2$ as part of the computation of $[\boldsymbol{\delta}]_2$; 3 to compute $[\boldsymbol{z}]_2$ as $\left(\begin{smallmatrix} r \\ r\chi \end{smallmatrix}\right)[e]_2 + (\boldsymbol{\varrho} + \left(\begin{smallmatrix} 0 \\ -ry \end{smallmatrix}\right))[1]_2$). The argument length is 4 elements of $\mathbb{G}_1$

$\mathsf{kgen}(\mathsf{p},\mathtt{lpar})$: $e \leftarrow\!\$\, \mathbb{Z}_p$; return $(\mathsf{crs},\mathtt{td}) \leftarrow ([e]_2, e)$;

$\mathsf{P}(\mathsf{crs},\mathtt{lpar},\mathtt{x},\mathtt{w})$: $y \leftarrow\!\$\, \mathbb{Z}_p$; $\boxed{\boldsymbol{\gamma} \leftarrow \binom{-1}{\chi-1} y}$; $\boxed{\boldsymbol{\gamma} \leftarrow \binom{-1}{\chi} y}$; $\boxed{\boldsymbol{\gamma} \leftarrow \binom{\chi}{1-\chi} y}$;
$\boldsymbol{\varrho} \leftarrow\!\$\, \mathbb{Z}_p^2$; $[\mathbf{ct}^{\boldsymbol{\gamma}}]_1 \leftarrow \mathsf{Enc}([\boldsymbol{\gamma}]_1; \boldsymbol{\varrho})$;
$\boxed{w \leftarrow -\chi}$; $\boxed{w \leftarrow \chi - 1}$; $[\delta]_2 \leftarrow -(w[e]_2 + y[1]_2)$; $[\boldsymbol{z}]_2 \leftarrow \boldsymbol{\varrho}[1]_2 + r\left[\begin{smallmatrix} e \\ \delta \end{smallmatrix}\right]_2$;
return $\pi = ([\mathbf{ct}^{\boldsymbol{\gamma}}]_1, [\delta, \boldsymbol{z}]_2)$.

$\mathsf{V}(\mathsf{crs},\mathtt{lpar},\mathtt{x},\pi)$: check $[\boldsymbol{I}_2]_2 \bullet [\mathbf{ct}^{\boldsymbol{\gamma}}]_1 + \left[\begin{smallmatrix} e \\ \delta \end{smallmatrix}\right]_2 \bullet [\mathbf{ct}]_1 + \boldsymbol{Q}\left[\begin{smallmatrix} e \\ \delta \end{smallmatrix}\right]_2 \bullet [0\|1]_1 \stackrel{?}{=} [\boldsymbol{z}]_2 \bullet \mathsf{pk}$;

Figure 5.7: $\boldsymbol{\Pi}^{\vee}_{\mathsf{simple}}$ (contains boxed entries), $\boldsymbol{\Pi}^{\vee}_{\mathsf{cg}}$ (contains dashed boxed entries), and $\boldsymbol{\Pi}^{\vee}_{\mathsf{cds}}$ (contains dotted boxed entries)

and 3 elements of $\mathbb{G}_2$.

The verifier's computation is dominated by 13 pairings. In the case of $\boldsymbol{\Pi}^{\vee}_{\mathsf{simple}}$, this follows from $\boldsymbol{Q}\left[\begin{smallmatrix} e \\ \delta \end{smallmatrix}\right]_2 = -\left[\begin{smallmatrix} \delta \\ \delta \end{smallmatrix}\right]_2$; thus, $[0\|1]_1 \bullet \boldsymbol{Q}\left[\begin{smallmatrix} e \\ \delta \end{smallmatrix}\right]_2 = -[0\|1]_1 \bullet \left[\begin{smallmatrix} \delta \\ \delta \end{smallmatrix}\right]_2$ can be computed in 1 pairing. In the case of $\boldsymbol{\Pi}^{\vee}_{\mathsf{cg}}$, it follows from $\boldsymbol{Q}\left[\begin{smallmatrix} e \\ \delta \end{smallmatrix}\right]_2 = -\left[\begin{smallmatrix} \delta \\ 0 \end{smallmatrix}\right]_2$; thus, $[0\|1]_1 \bullet \boldsymbol{Q}\left[\begin{smallmatrix} e \\ \delta \end{smallmatrix}\right]_2 = -[0\|1]_1 \bullet \left[\begin{smallmatrix} \delta \\ 0 \end{smallmatrix}\right]_2$ can be computed in 1 pairing.

$\boldsymbol{\Pi}^{\vee}_{\mathsf{cds}}$. From the outset, the famous Cramer-Damgård-Schoenmakers (CDS) Σ-protocol from [CDS94] looks quite different. The idea behind CDS is that to prove that $\chi \in \{0,1\}$, one follows the prover's algorithm in the true branch (resulting in transcript (a_χ, e_χ, z_χ)) and the simulator's algorithm in the other branch (resulting in transcript $(a_{3-\chi}, e_{3-\chi}, z_{3-\chi})$). To make sure that at least one branch is correctly computed, the prover chooses e_i such that $e_1 + e_2 = e$, where e is the verifier's second message. Couteau and Hartmann [CH20] described a CH-compilation of the CDS protocol.

Somewhat unexpectedly, one can use our generic framework also here, by defining the QDR $\boldsymbol{C}_{cds}(X) = \left(\begin{smallmatrix} 0 & X \\ X-1 & 1-X \end{smallmatrix}\right)$. However, $\boldsymbol{C}_{cds}(X)$ does not belong to the class of matrices considered by Ishai and Kushilevitz, [IK00; IK02] and thus not correspond to an ABP.

In Figure 5.7, we also depict the new NIZK argument $\boldsymbol{\Pi}^{\vee}_{\mathsf{cds}}$ that applies Figures 5.1 and 5.2 to $\boldsymbol{C}_{cds}(X)$. The property of CDS that the simulated branch depends on χ carries over since one samples $\gamma_{2-\chi} \leftarrow\!\$\, \mathbb{Z}_p$ and sets $\gamma_{1-\chi} \leftarrow 0$; i.e., the index i of the non-random γ_i depends on χ. Intuitively, the prover simulates the branch $2-\chi$. The reason behind it is that $\det(\boldsymbol{C}_{(1,1)}(\chi)) \neq 0$ if $\chi = 0$ and $\det(\boldsymbol{C}_{(1,2)}(\chi)) \neq 0$ if $\chi = 1$.

As a small optimization, $[\boldsymbol{z}]_2$ can computed as follows:

(1) $[\boldsymbol{z}]_2 = \boldsymbol{\varrho}[1]_2 + r\left[\begin{smallmatrix} e \\ (1-\chi)e - y \end{smallmatrix}\right]_2 = r\binom{\varrho_1}{\varrho_2 - ry}[1]_2 + r\left[\begin{smallmatrix} e \\ e \end{smallmatrix}\right]_2$, if $\chi = 0$,

(2) $[\boldsymbol{z}]_2 = \binom{\varrho_1}{\varrho_2 - ry}[1]_2 + r\left[\begin{smallmatrix} e \\ 0 \end{smallmatrix}\right]_2$, if $\chi = 1$.

In both cases, the prover spends 3 exponentiations in $\mathbb{G}_2$. Thus, the prover's computation is dominated by $5\mathfrak{e}_1 + 4\mathfrak{e}_2$.

To see the verifier accepts note that here $\boldsymbol{Q}\left[\begin{smallmatrix}e\\ \delta\end{smallmatrix}\right]_2 = \left[\begin{smallmatrix}0\\ \delta-e\end{smallmatrix}\right]_2$. In particular, $\left[\begin{smallmatrix}e\\ \delta\end{smallmatrix}\right]_2 \bullet [\mathbf{ct}]_1 + \boldsymbol{Q}\left[\begin{smallmatrix}e\\ \delta\end{smallmatrix}\right]_2 \bullet [0\|1]_1 = \left[\begin{smallmatrix}e\\ \delta\end{smallmatrix}\right]_2 \bullet [\mathbf{ct}]_1 + \left[\begin{smallmatrix}0\\ \delta-e\end{smallmatrix}\right]_2 \bullet [0\|1]_1$ can be computed in 5 pairings. In total, the verifier executes 13 pairings.

Range Proof

The following example both has a long cryptographic pedigree and can be used to simply explain how to expand our framework. In a range proof, the task is to prove that the encrypted value belongs to a fixed range $[0, N]$. Many range proofs have been proposed in the cryptographic literature, [Bou00; LAN03; Lip03; CCs08; RKP09; CLs10; CLZ12; Daz+19], due to their many applications and non-trivial constructions. It is possible that the Couteau-Hartmann compilation works directly with some of the existing Σ-protocol-based range proofs like [LAN03]. We will next show how to use our framework to obtain a proof with $\Theta(\log N)$ communication. Write $\eta = \lfloor \log_2 N \rfloor$. In this case, just setting $\mathcal{A}_N = \{x : 0 \leq x \leq N\}$ results in an inefficient NIZK argument, since $GS(\mathcal{A}_N) = \{\prod_{i=0}^{N}(x-i)\}$ contains a polynomial F of linear-in-N degree $N+1$. (Since F is univariate, one can use the solution of Section 5.6 in this case.)

One can instead use a different generating set of smaller-degree polynomials. Assuming $N = 2^\eta - 1$, a well-known idea in range proofs is to extend x to binary digits x_i, and to prove separately that each x_i is Boolean. In the case $N+1$ is not a power of two, one can use an idea from [LAN03]. Namely, let $b_j := \lfloor (N + 2^j)/2^{j+1} \rfloor$, where $j \in [0, \eta]$. Then, $\chi \in [0, N]$ iff $\chi = \sum_{j=0}^{\eta} b_j \chi_j$ for some $\chi_j \in \{0, 1\}$ [LAN03].

To translate this idea to our framework, we introduce additional indeterminates and write

$$\mathcal{A}'_N = \left\{ (x, x_0, \ldots, x_\eta) : x = \sum_{j=0}^{\eta} b_j x_j \wedge (b_j \in \{0,1\} \text{ for all } j) \right\} .$$

Note that in the terms of algebraic geometry, $\mathcal{A}'_N$ is a variety in the affine space $\mathbb{Z}_p^{\eta+2}$, such that $\mathcal{A}_N$ is its projection to the affine space $\mathbb{Z}_p$.

Clearly,

$$\mathbf{GB}(\mathcal{A}'_N) = \left\{ X_\eta^2 - X_\eta, \ldots, X_0^2 - X_0, X - \sum_{j=0}^{\eta} b_j X_j \right\}$$

is a (lexicographic) Gröbner basis for $\mathcal{A}'_N$ that consists of one linear and η quadratic polynomials. Thus, the resulting NIZK argument has communication complexity $\Theta(\eta) = \Theta(\log N)$. A similar trick is useful in also other settings.

We can base range proofs on d-ary digits, for $d \geq 2$, using an ABP-based univariate NIZK to show that each $X_j \in \{0, \ldots, d-1\}$. One has to execute $\lfloor \log_d N \rfloor$ basic NIZK proofs. The resulting range proof has complexities depicted in Table 5.4. (The complexities are such due to the fact that in this case, all values $\chi - \xi_i$ are small.) In particular, the verifier's computation (which is the most important measure in many applications) is minimized when $d = 3$.

Table 5.4: Complexities in the range proof. Every entry should be multiplied by $\log_2 N$.

	P comp in $(\mathfrak{e}_1, \mathfrak{e}_2)$	V comp in $\mathfrak{p}$	Comm. in $(\lvert\mathbb{G}_1\rvert, \lvert\mathbb{G}_2\rvert)$
General	$(\frac{3d-1}{\log_2 d}, \frac{3d-1}{\log_2 d})$	$\frac{7d-1}{\log_2 d}$	$(\frac{2d}{\log_2 d}, \frac{(2d-1)}{\log_2 d})$
$d = 2$ (also [CH20])	$(5, 5)$	13	$(4, 3)$
$d = 3$	$(5.05, 5.05)$	12.62	$(3.79, 3.15)$

As in the case of the multi-dimensional set-membership proof, an alternative is to use signature-based solutions [RKP09; Daz+19] that offer somewhat better proof size $\Theta(N/\log N)(|\mathbb{G}_1| + |\mathbb{G}_2|)$. However, also here these solutions have a longer CRS size and require that the underlying signature scheme is unforgeable. We leave it as an open question how to combine the protocols of the current paper with signatures.

5.D More on Section 5.7

Elliptic Curve Points, Case $F(X, Y) = X^3 + aX + b - Y^2$ for $a \neq 0$

By inspection, we found the following 3×3 matrix, where[11] $s = \sqrt{-b/a}$:

$$\boldsymbol{C}(X, Y) = \begin{pmatrix} Y & -s & X \\ X & -1 & s \\ a & X & Y \end{pmatrix} . \tag{5.5}$$

Clearly, $\det \boldsymbol{C}(X, Y) = F(X, Y)$. However, $\boldsymbol{C}$ is not a QDR. We will explain next what does it mean in the concrete case.

Solving Equation (5.2) together with $F(X, Y) = 0$ gives us the following formulas to replace into Figure 5.1 depending on which minor of $\boldsymbol{C}$ is non-zero:

$$\boldsymbol{w} \leftarrow \begin{cases} \begin{pmatrix} a(sY - X^2) \\ a(Y - sX) \end{pmatrix} / (aX + b) & \text{if } b + aX \neq 0 \ , \\ \begin{pmatrix} as - XY \\ a + X^2 \end{pmatrix} / (sX + Y) & \text{if } sX + Y \neq 0 \ , \\ \begin{pmatrix} aX - Y^2 \\ as + XY \end{pmatrix} / (sY + X^2) & \text{if } sY + X^2 \neq 0 \ . \end{cases}$$

Since $Y^2 = X^3 + aX + b$, one can use formulas like $X^3 + b = Y^2 - aX$ to modify the expressions. In particular, the three given expressions for $\boldsymbol{w}$ are equivalent if the three denominators $sX + Y = -\det(\boldsymbol{C}_{(1,1)})$, $sY + X^2 = -\det(\boldsymbol{C}_{(2,1)})$, and $aX + b = a \det(\boldsymbol{C}_{(3,1)})$ are all non-zero.

Solving $F(X, Y) = 0$ and $\det(\boldsymbol{C}_{(i,1)}) = 0$ gives that the ith expression for $\boldsymbol{w}$ holds except in either 3, 4, or 2 points. Since there is only one point $(X, Y) = (-b/a, bs/a)$ where all $F(X, Y) = 0$ and $\det(\boldsymbol{C}_{(i,1)}) = 0$ hold, it means one can compute $\boldsymbol{w}$ in all but a single point.

[11] Hence, this assumes that there exists a square root of $-b/a$ modulo p, i.e., that there exists c such that $ac^2 = -b$, which is true for $(p+1)/2$ values of b. If b is not one of those values, one can by inspection find a different matrix. Alternatively, one can use the ABP-based solution from Section 5.6.

Thus, we can construct a NIZK argument, with $\ell = 3$, assuming that there exists a square root of $-b/a$ modulo p. Moreover, it cannot be applied in the special case $(X, Y) = (-b/a, bs/a)$. Thus, strictly speaking, the resulting NIZK is not for $\mathcal{L}_{\mathsf{pk},F}$ but for a different language, and this outlines the need of QDRs. However, the resulting argument could be still interesting in the case when in the honest case, (X, Y) has some restrictions.

Elliptic Curve Points, Case $F(X, Y) = X^3 + b - Y^2$

Consider the following less common normal form for an elliptic curve,

$$F(X, Y) = (X + aY)(X + bY)(X + cY) - X \ ,$$

for mutually different a, b, c; w.l.o.g., let $b \neq 0$. By inspection, we found the following matrix:

$$\boldsymbol{C}^\top(X, Y) = \begin{pmatrix} X & 0 & -1 \\ Y+s & X+s & 1 \\ -sX+Y+s^2 & Y & X \end{pmatrix} \ ,$$

where $s = b^{1/3}$ (assuming b has a cubic root). Then,

$$\boldsymbol{w} \leftarrow \begin{cases} \begin{pmatrix} Y/(s+X)+1 \\ -X \end{pmatrix} & \text{if } s + X \neq 0 \ , \\ \begin{pmatrix} (s^2-sX+X^2+Y)/Y \\ -X \end{pmatrix} & \text{if } Y \neq 0 \ , \\ \begin{pmatrix} -s^2+2sX+(X-1)Y \\ -sX^2+b+Y(X-Y) \end{pmatrix} / (X(s+X) - Y) & \text{if } X(s+X) - Y \neq 0 \ . \end{cases}$$

None of these formulas succeeds if all $F(X, Y) = s + X = Y = X(s + X) - Y = 0$, which can only happen if $(X, Y) = (-s, 0)$.

Fifth-Degree Example

Next, we give a fifth-degree example directly from [PSV12]:

$$\begin{aligned} F(X, Y) =& X^5 + 3X^4Y - 2X^4 - 5X^3Y^2 - 12X^3 - 15X^2Y^3 + 10X^2Y^2 - 28X^2Y + 14X^2 + \\ & 4XY^4 - 6XY^2 - 12XY + 26X + 12Y^5 - 8Y^4 - 32Y^3 + 16Y^2 + 48Y - 24 \ , \end{aligned}$$

and

$$\boldsymbol{C}(X, Y) = \begin{pmatrix} X+Y & 0 & 0 & 0 & 0 \\ 0 & X+2Y & 0 & 0 & 0 \\ 0 & 0 & X-Y & 0 & 0 \\ 0 & 0 & 0 & X-2Y & 0 \\ 0 & 0 & 0 & 0 & X+3Y-2 \end{pmatrix} + \begin{pmatrix} 0 & 2 & 1 & 0 & 0 \\ 2 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 1 \\ 0 & 0 & 2 & 0 & -1 \\ 0 & 1 & 1 & -1 & 0 \end{pmatrix} \ .$$

As noted in [PSV12], this is just one of 33280 possible solutions for the latter (integer) matrix. In this case, one can write 5 different formulas for $\boldsymbol{\chi_1}$, depending on which submatrix $\boldsymbol{C}_{(I,1)}(X, Y)$ has a non-zero determinant. One can check that there are four points for which all these submatrices have a zero determinant.

Note that there is no obvious small-dimensional ABP-based solution in this case.

5.E More on Section 5.8

Another Multi-Dimensional Set-Membership Proof

To demonstrate that one does not always need a set-membership proof of the worst-case size, we will next work out an example for the following set

$$\mathcal{A} = \{(2,1,2),(1,4,2),(3,1,3),(1,2,3)\} \subset \mathbb{Z}_p^3 \ .$$

By using `CoCoA`, we found the following lexicographic Gröbner basis

$$\mathbf{GB}_{lex}(\mathcal{I}) = \left\{(z-3)(z-2), (y-1)(y+2z-8), x+\frac{1}{3}(5y-8)z-3y+3\right\}$$

of size 3. (The corresponding degree-lexicographic and degree-reverse-lexicographic Gröbner bases have size 6.) By following our methodology, to show that $\boldsymbol{\chi} \in \mathcal{A}$, we show that $F_i(\boldsymbol{\chi}) = 0$ for each $F_i \in \mathbf{GB}_{lex}(\mathcal{I})$. More precisely:

- We show that $(z-3)(z-2) = 0$, by using $\boldsymbol{C}_1 = \left(\begin{smallmatrix} z-2 & -1 \\ 0 & z-3 \end{smallmatrix}\right)$.
- We show that $(y-1)(y+2z-8) = 0$, by using $\boldsymbol{C}_2 = \left(\begin{smallmatrix} y-1 & -1 \\ 0 & y+2z-8 \end{smallmatrix}\right)$.
- We show that $3x + y(5z-9) - 8z + 9 = 0$, by using $\boldsymbol{C}_3 = \left(\begin{smallmatrix} y & -1 \\ 3x-8z+9 & 5z-9 \end{smallmatrix}\right)$.

Thus, one needs 3 NIZK arguments for quadratic polynomials ($\ell = 2$). By Lemma 5.4.1, the NIZK argument for $\mathcal{A}$ has thus communication of $3 \cdot 2 \cdot 2 = 12$ elements of $\mathbb{G}_1$ and $3(2 \cdot 2 - 1) = 9$ elements of $\mathbb{G}_2$.

As in all examples in Section 5.8, we used Gröbner-basis techniques to find a small aPCS for $\mathcal{A}$. Clearly, any arithmetic circuit for checking that $\boldsymbol{\chi} \in \mathcal{A}$ has size larger than 3. In particular, in this concrete case, it seems that one needs to use the full power of aPCS.

An alternative generating set, that is not a Gröbner basis, is

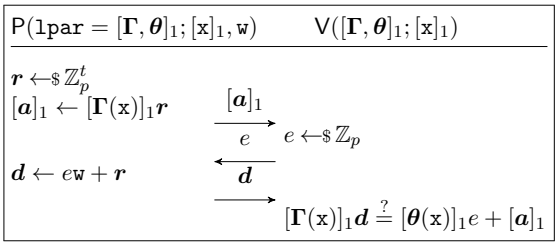
$$GS(\mathcal{I}) = \{(x-1)(y-1), (x-3)(y-2)(z-2), (x-2)(y-4)(z-3)\}$$

of size 3. While GS is tidier, the argument for $GS(\mathcal{A})$ is slightly less efficient since two of the polynomials are cubic. Thus, here, one can construct three QDRs of size 2, 3, and 3. The resulting NIZK has communication of $2 \cdot 2 + 2 \cdot 2 \cdot 3 = 16$ elements of $\mathbb{G}_1$ and $(2 \cdot 2 - 1) + 2 \cdot (2 \cdot 3 - 1) = 13$ elements of $\mathbb{G}_2$.

5.F More on Section 5.9

CHM NIZK

We describe the CHM (Couteau-Hartmann-Maurer) Σ-protocol and the resulting NIZK, see Figure 5.8. For further reference, we state the following results. We refer to Section 5.A and [CH20] for unexplained notions and notation.



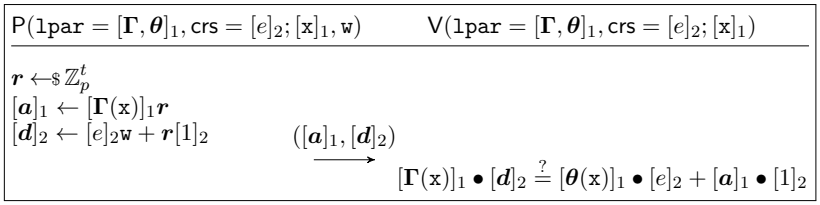


Figure 5.8: The CHM Σ-protocol for algebraic languages $\mathcal{L}_{\boldsymbol{\Gamma},\boldsymbol{\theta}}$ (above) and its Couteau-Hartmann compilation Π_Σ^C (below)

Proposition 5.F.1 (Efficiency of the CHM Σ-Protocol and CH Compilation)**.** *Assume $[\boldsymbol{\Gamma}]_1 \in \mathbb{G}_1^{n\times t}$ and $[\boldsymbol{\theta}]_1 \in \mathbb{G}_1^n$. Let $T_\Gamma := \{|(i,j)| : \Gamma_{ij} \neq 0\}$ and $T_\theta := \{|i| : \theta_i \neq 0\}$. In the CHM Σ-protocol, the prover executes $T_\Gamma \leq nt$ exponentiations and the verifier executes $T_\Gamma + T_\theta + n \leq nt + n$ exponentiations; the communication is n group elements and $t+1$ integers. In the compiled protocol, the prover executes $T_\Gamma \leq nt$ exponentiations in $\mathbb{G}_1$ and $2n$ exponentiations in $\mathbb{G}_2$, and the verifier executes $T_\Gamma + T_\theta + n \leq nt + 2n$ pairings; the communication is $n|\mathbb{G}_1| + t|\mathbb{G}_2|$.*

Proposition 5.F.2 (Couteau-Hartmann)**.** *Consider the NIZK argument Π_Σ^C, described in Figure 5.8, for any algebraic language distribution $\mathcal{D}_{\mathtt{lpar}}$ outputting pairs $\mathtt{lpar} = [\boldsymbol{\Gamma}, \boldsymbol{\theta}]_1 \in \mathcal{P}_\nu^{n\times t} \times \mathcal{P}_\nu^n$.*

1. *It is sound under the $\mathcal{L}_1$-t-CED assumption in $\mathbb{G}_2$ relative to* Pgen.
2. *If the language distribution is witness-sampleable with trapdoors $\mathbf{T}_{\mathtt{lpar}} \in \mathbb{Z}_p^{n\times n}$, then Π_Σ^C is sound under the falsifiable $\mathcal{L}_1$-t-CED assumption in $\mathbb{G}_2$ relative to* Pgen.
3. *If the language distribution is m-trapdoor reducible, then Π_Σ^C is sound under the falsifiable $\mathcal{L}_1$-$(t-m)$-CED assumption in $\mathbb{G}_2$ relative to* Pgen.

Note that [CH20] proved the soundness under KerMDH assumptions, but it is easy to see that the soundness also holds under CED assumptions.

More Examples

To simplify parsing, we have omitted the use of bracket notation in examples, writing say 0 instead of $[0]_1$.

Example 5.F.3. Let $F(X) = \prod_{i=1}^{4}(X - \xi_i)$. Then

$$[\mathbf{\Gamma}]_1 = \left(\begin{array}{ccc|cccc} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & \mathsf{sk} & 0 & 0 & 0 \\ \mathsf{ct}_1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \mathsf{ct}_2 - \xi_2 & -1 & 0 & 0 & \mathsf{sk} & 0 & 0 \\ 0 & \mathsf{ct}_1 & 0 & 0 & 0 & 1 & 0 \\ 0 & \mathsf{ct}_2 - \xi_3 & -1 & 0 & 0 & \mathsf{sk} & 0 \\ 0 & 0 & \mathsf{ct}_1 & 0 & 0 & 0 & 1 \\ 0 & 0 & \mathsf{ct}_2 - \xi_4 & 0 & 0 & 0 & \mathsf{sk} \end{array}\right) \in \mathbb{Z}_p^{8\times 7} \quad , \quad [\boldsymbol{\theta}]_1 = \begin{pmatrix} \mathsf{ct}_1 \\ \mathsf{ct}_2 - \xi_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .$$

In this case, $w_1 = -(\chi - \xi_1)$, $w_2 = -(\chi - \xi_1)(\chi - \xi_2)$, $w_3 = -(\chi - \xi_1)(\chi - \xi_2)(\chi - \xi_3)$, and $\hat{\boldsymbol{w}} = r(\begin{smallmatrix} 1 \\ -\boldsymbol{w} \end{smallmatrix}) = r(1 \| \chi - \xi_1 \| (\chi - \xi_1)(\chi - \xi_2) \| (\chi - \xi_1)(\chi - \xi_2)(\chi - \xi_3))$.

Example 5.F.4 (Elliptic curve.)**.** Let $F(X, Y) = X^3 + aX + b - Y^2$ and

$$\boldsymbol{C}(X, Y) = \begin{pmatrix} X & -1 & 0 & 0 \\ 0 & X & -1 & 0 \\ Y & 0 & 0 & -1 \\ b & a & X & -Y \end{pmatrix}$$

be as in Figure 5.5. Then for $[\mathbf{ct}_1]_1 = \mathsf{Enc}(\chi_1; r_1)$ and $[\mathbf{ct}_2]_1 = \mathsf{Enc}(\chi_2; r_2)$,

$$[\mathbf{\Gamma}]_1 = \left(\begin{array}{ccc|cccc} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & \mathsf{sk} & 0 & 0 & 0 \\ \mathsf{ct}_{11} & 0 & 0 & 0 & 1 & 0 & 0 \\ \mathsf{ct}_{12} & -1 & 0 & 0 & \mathsf{sk} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & \mathsf{sk} & 0 \\ 0 & \mathsf{ct}_{11} & -\mathsf{ct}_{21} & 0 & 0 & 0 & 1 \\ a & \mathsf{ct}_{12} & -\mathsf{ct}_{22} & 0 & 0 & 0 & \mathsf{sk} \end{array}\right) \quad , \quad [\boldsymbol{\theta}]_1 = \begin{pmatrix} \mathsf{ct}_{11} \\ \mathsf{ct}_{12} \\ 0 \\ 0 \\ \mathsf{ct}_{21} \\ \mathsf{ct}_{21} \\ 0 \\ b \end{pmatrix} .$$

In this case, $\boldsymbol{w}^\top = (w_1^* \| \ldots \| w_3^*) = (-\chi_1 \| -\chi_1^2 \| -\chi_2)$, and

$$\begin{aligned} \hat{\boldsymbol{w}} = \begin{pmatrix} w_4^* \\ \cdots \\ w_7^* \end{pmatrix} &= \left(\begin{pmatrix} 1&0&0&0 \\ 0&1&0&0 \\ 0&0&0&0 \\ 0&0&1&0 \end{pmatrix} \cdot r_1 + \begin{pmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 1&0&0&0 \\ 0&0&0&-1 \end{pmatrix} \cdot r_2 \right) \cdot \begin{pmatrix} 1 \\ -\boldsymbol{w} \end{pmatrix} \\ &= \begin{pmatrix} r_1&0&0&0 \\ 0&r_1&0&0 \\ r_2&0&0&0 \\ 0&0&r_1&-r_2 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ \chi_1 \\ \chi_1^2 \\ \chi_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_1\chi_1 \\ r_2 \\ r_1\chi_1^2 - r_2\chi_2 \end{pmatrix} . \end{aligned}$$

Clearly,

$$\begin{aligned} \mathbf{\Gamma} \cdot \boldsymbol{w}^* &= \begin{pmatrix} -\chi_1\mathcal{E}(-1;0) + r_1\mathcal{E}(0;1) \\ -\chi_1\mathcal{E}(\chi_1;r_1) - \chi_1^2\mathcal{E}(-1;0) + r_1\chi_1\mathcal{E}(0;1) \\ -\chi_2\mathcal{E}(-1;0) + r_2\mathcal{E}(0;1) \\ -\chi_1\mathcal{E}(a;0) - \chi_1^2\mathcal{E}(\chi_1;r_1) - \chi_2\mathcal{E}(-\chi_2;-r_2) + (r_1\chi_1^2 - r_2\chi_2)\mathcal{E}(0;1) \end{pmatrix} \\ &= \begin{pmatrix} \mathcal{E}(\chi_1;r_1) \\ \mathcal{E}(-\chi_1^2;-r_1\chi_1) + \mathcal{E}(\chi_1^2;0) + \mathcal{E}(0;r_1\chi_1) \\ \mathcal{E}(\chi_2;0) + \mathcal{E}(0;r_2) \\ \mathcal{E}(-a\chi_1;0) + \mathcal{E}(-\chi_1^3;-r_1\chi_1^2) + \mathcal{E}(\chi_2^2;r_2\chi_2) + \mathcal{E}(0;r_1\chi_1^2 - r_2\chi_2) \end{pmatrix} \\ &= \begin{pmatrix} \mathcal{E}(\chi_1;r_1) \\ \mathcal{E}(0;0) \\ \mathcal{E}(\chi_2;r_2) \\ \mathcal{E}(\chi_2^2 - a\chi_1 - \chi_1^3;0) \end{pmatrix} \overset{(*)}{=} \begin{pmatrix} \mathcal{E}(\chi_1;r_1) \\ \mathcal{E}(0;0) \\ \mathcal{E}(\chi_2;r_2) \\ \mathcal{E}(b;0) \end{pmatrix} = \boldsymbol{\mathcal{E}}(\boldsymbol{h}(\boldsymbol{\chi})) \ , \end{aligned}$$

where $(*)$ holds iff $F(\boldsymbol{\chi}) = 0$.

CHM NIZK based on Couteau-Hartmann Disjunction

On the Couteau-Hartmann Disjunction. Next, we describe the Couteau-Hartmann disjunction that results in $\boldsymbol{\Gamma}$ of size $(3d-1)\times(3d-2)$ and compare it to Equation (5.4).

In Appendix C of [CH20], the authors describe a method of constructing the parameters $[\boldsymbol{\Gamma}]_1$ and $[\boldsymbol{\theta}]_1$ of $\mathcal{L}_{\boldsymbol{\Gamma},\boldsymbol{\theta}}$ for the disjunction of two algebraic languages $\mathcal{L}_{\boldsymbol{\Gamma}_i,\boldsymbol{\theta}_i}$, $i\in\{0,1\}$. That is, $\mathtt{x}\in\mathcal{L}_{\boldsymbol{\Gamma},\boldsymbol{\theta}}$ iff $\mathcal{L}_{\boldsymbol{\Gamma}_i,\boldsymbol{\theta}_i}$ for at least one i. Briefly, they define

$$\boldsymbol{\Gamma} := \begin{pmatrix} \mathbf{0}_{1\times M_1} & 1 & \mathbf{0}_{1\times M_0} & 1 \\ \mathbf{0}_{N_0\times M_1} & \mathbf{0}_{N_0} & \boldsymbol{\Gamma}_0 & \boldsymbol{\theta}_0 \\ \boldsymbol{\Gamma}_1 & \boldsymbol{\theta}_1 & \mathbf{0}_{N_1\times M_0} & \mathbf{0}_{N_1} \end{pmatrix} \quad , \quad \boldsymbol{\theta} := \begin{pmatrix} -1 \\ \mathbf{0}_{N_0+N_1} \end{pmatrix} \tag{5.6}$$

Thus, a disjunction from matrices $[\boldsymbol{\Gamma}_i]_1$ of size $N_i\times M_i$ ends up with a matrix $[\boldsymbol{\Gamma}]_1$ of size $(N_1+N_2+1)\times(M_1+M_2+2)$. In the honest case, a valid witness is either $(\boldsymbol{w}_0^\top,-1,0,0)^\top$ or $(0,0,\boldsymbol{w}_1^\top,-1)^\top$, where $\boldsymbol{w}_i$ is a valid witness corresponding to the ith disjunct.

We will demonstrate how it differs from our parametrization for the two examples given above.

First, when $F(X)=X-\xi$ and thus $[\mathsf{ct}]_1=[r[1]_1\|r[\mathsf{sk}]_1+\xi[1]_1]_1$, then $\boldsymbol{C}=(\chi-\xi)$ and thus

$$[\boldsymbol{\Gamma}]_1 = \begin{pmatrix} [1]_1 \\ [\mathsf{sk}]_1 \end{pmatrix} \in \mathbb{G}_1^{2\times 1} \quad , \quad [\boldsymbol{\theta}]_1 = \begin{pmatrix} \mathsf{ct}_1 \\ \mathsf{ct}_2-[\xi]_1 \end{pmatrix} ,$$

with $w=r$. Applying the disjunction of Equation (5.6) to it for two different values of ξ_i and ciphertexts $[\mathsf{ct}_i]_1$, $i\in\{1,2\}$, we get (omitting the bracket notation)

$$\boldsymbol{\Gamma} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & \mathsf{ct}_{1,1} \\ 0 & 0 & \mathsf{sk} & \mathsf{ct}_{1,2}-\xi_1 \\ 1 & \mathsf{ct}_{1,1} & 0 & 0 \\ \mathsf{sk} & \mathsf{ct}_{1,2}-\xi_2 & 0 & 0 \end{pmatrix} \in \mathbb{Z}_p^{5\times 4} \ , \boldsymbol{\theta} = \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} ,$$

with $w=(r_1,-1,0,0)^\top$ or $w=(0,0,r_2,-1)^\top$. This should be compared with 4×3 matrix $\boldsymbol{\Gamma}$ of [Ben+13] (see also Example 5.9.1). Going one step forward, for $d=4$, the Couteau-Hartmann disjunction results in a matrix of size $(2\cdot 5+1)\times(2\cdot 4+2)=11\times 10$, which should compared with the matrix $\boldsymbol{\Gamma}$ of Example 5.F.3 that has size 8×7. In the general case $d=2^c$ for some $c\geq 1$, the resulting matrix has dimensions

$$(3d-1)\times(3d-2) \ .$$

As noted before, the new solution results in matrices of size $2d\times(2d-1)$.

Efficiency. For the sake of completeness, we reprove the following lemma, also given in [CH20]. Note that $\mathtt{w}$ has zero elements which means that the computation of $[\boldsymbol{d}]_2$ by the prover is more efficient than by the general result Proposition 5.F.1.

Lemma 5.F.5. *Let $d=2^c$, and assume in recursion $\boldsymbol{\Gamma}_0$ and $\boldsymbol{\Gamma}_1$ always have equal dimensions. The CH compiled NIZK argument, as in Figure 5.8, corresponding to $\boldsymbol{\Gamma}$ of this subsection as in Equation (5.6), requires $(7d-4)\mathfrak{e}_1+(3d-1)\mathfrak{e}_2$ from the prover, $(9d-2)\mathfrak{p}$ from the verifier, and the communication is $(3d-1)|\mathbb{G}_1|+(3d-2)|\mathbb{G}_2|$.*

Proof. **Prover's computation.** The prover needs to compute $[\mathbf{\Gamma}(\mathtt{x})]_1\boldsymbol{r}$ and $[e]_2\mathtt{w}+\boldsymbol{r}[1]_2$.

If $d=1$ then the multiplication $[\mathbf{\Gamma}(\mathtt{x})]_1\boldsymbol{r}$ can be executed in $T_1=2$ exponentiations. If $d=2$ then it takes $T_2=10$ exponentiations. Assume that for fixed $d\geq 2$, the multiplication takes T_d exponentiations. Then, T_{2d} can be executed in $2T_d+4$ exponentiations. Solving this recurrence relation gives that $T_d=7d-4$ in $\mathbb{G}_1$.

On top of this, the prover computes $[\boldsymbol{d}]_2 \leftarrow [e]_2\mathtt{w}+\boldsymbol{r}[1]_2$. If $d=1$ then this can be executed in 2 exponentiations. At each recursion step, $\mathtt{w}$ will still have one non-small element and $\boldsymbol{r}$ will have dimension $3d-2$. Thus, this takes $1+(3d-2)=3d-1$ exponentiations in $\mathbb{G}_2$.

Verifier's computation. Since $\mathbf{\Gamma}$ has $6d-2$ non-zero elements, the verifier has to execute $6d-2$ pairings to compute $[\mathbf{\Gamma}]_1 \bullet [\boldsymbol{d}]_2$. In addition, she has to execute 1 pairing to compute $[\boldsymbol{\theta}(\mathtt{x})]_1 \bullet [e]_2$, and $n=3d-1$ pairings to compute $[\boldsymbol{a}]_1 \bullet [1]_2$, in total $9d-2$ pairings.

Communication. $n|\mathbb{G}_1|+t|\mathbb{G}_2| = (3d-1)|\mathbb{G}_1|+(3d-2)|\mathbb{G}_2|$. □

Chapter 6

Paper II

Verifiably-Extractable OWFs and Their Applications to Subversion Zero-Knowledge

Prastudy Fauzi, Helger Lipmaa, Janno Siim, Michał Zając and Arne Tobias Ødegaard

ASIACRYPT, December 2021.

Abstract

An extractable one-way function (EOWF), introduced by Canetti and Dakdouk (ICALP 2008) and generalized by Bitansky et al. (SIAM Journal on Computing vol. 45), is an OWF that allows for efficient extraction of a preimage for the function. We study (generalized) EOWFs that have a public image verification algorithm. We call such OWFs verifiably-extractable and show that several previously known constructions satisfy this notion. We study how such OWFs relate to subversion zero-knowledge (Sub-ZK) NIZKs by using them to generically construct a Sub-ZK NIZK from a NIZK satisfying certain additional properties, and conversely show how to obtain them from any Sub-ZK NIZK. Prior to our work, the Sub-ZK property of NIZKs was achieved using concrete knowledge assumptions.

6.1 Introduction

Extractability is a way to formalize what an algorithm *knows*. It is a notion essential to modern cryptography which dates back to the works of Goldwasser et al. [GMR85] who proposed *proofs of knowledge*, and later formalized for interactive proofs by Bellare and Goldreich [BG93].[1] For non-interactive proofs, Damgård [Dam92] proposed knowledge-of-exponent assumptions, which are non-falsifiable assumptions[2] saying that any efficient algorithm that produces group elements that satisfy a specific relation must know their discrete logarithms.

Investigating extractable primitives, Canetti and Dakdouk [CD08] introduced the notion of extractable one-way functions (EOWFs). These are one-way functions f such that any adversary who produces an image of f must "know" its preimage. One formalizes this by saying that for every adversary $\mathcal{A}$ that outputs a value $y \in \mathrm{im}(f)$, there exists an extractor Ext that, given $\mathcal{A}$'s auxiliary input and randomness, can output a preimage for y under f. In the case of black-box (resp., non-black-box [Bar01; Bar+01]) extractability, Ext is universal and has no access (resp., has access) to $\mathcal{A}$'s code.

Until the work of Bitansky *et al.* in [Bit+16], EOWFs were only known under very strong knowledge-of-exponent assumptions [Bit+12], making little attempt to justify how extraction would work. Bitansky *et al.* defined generalized extractable one-way functions (GEOWFs) and constructed a GEOWF based on sub-exponential learning with errors (or, alternatively, any delegation scheme) and non-black-box extraction, given that the auxiliary input of the adversary is bounded. They also prove that GEOWFs secure against auxiliary input of polynomially unbounded length do not exist assuming

[1] Extractability in interactive protocols is well-studied and involves a technique called *rewinding*. In this paper we focus on extractability for non-interactive protocols.

[2] Essentially, one cannot efficiently check if an adversary breaks the assumption.

indistinguishability obfuscation (which seems an increasingly plausible assumption given recent progress [JLS20; WW20]).

Extractability and SNARKs. Extractability assumptions are widely used in various flavors of non-interactive zero-knowledge (NIZK) protocols, which are useful tools in ensuring privacy and correctness of cryptographic protocols. Succinct non-interactive zero-knowledge arguments of knowledge (zk-SNARKs, [Gro10; Lip12; Gen+13; Gro16]) are NIZKs that have sublinear-length proofs and are knowledge-sound (for any valid proof, the prover must "know" a witness). The knowledge-soundness property of a SNARK relies on being able to extract the witness from an adversary that outputs a valid argument. SNARKs are extremely popular due to practical applications such as verifiable computation and privacy-preserving cryptocurrencies (e.g., Zcash [Ben+14]).

An interesting question is which assumptions are necessary for SNARKs. Due to the impossibility result of Gentry and Wichs [GW11], any adaptively sound SNARK must rely on non-falsifiable assumptions. However, while non-falsifiable assumptions are necessary, they need not be knowledge assumptions. In fact, Bitansky et al. [Bit+12] showed that extractable collision-resistant hash functions (ECRHs) are necessary and sufficient to construct a SNARK that is adaptively sound and only privately verifiable. More precisely, they construct a designated verifier SNARK for NP from an ECRH and (an appropriate) private information retrieval, and construct a (specific variant of) ECRH from a designated verifier SNARK and a CRH. They also showed that ECRH implies EOWF.

Extractability and Subversion Zero-knowledge. Efficient SNARKs are typically defined in the common reference string (CRS) model, where one assumes that the prover and the verifier have access to a CRS generated by a trusted third party. However, in practice, such a party usually does not exist; this is important since a malicious CRS generator may cooperate with the prover to break soundness, or with the verifier to break zero-knowledge. Thus, it is preferable to construct SNARKs, and NIZKs in general, in weaker trust models than the CRS model.

The general notion of parameter subversion has been studied in [Rus+16]. Bellare et al. [BFS16] defined subversion zero-knowledge (Sub-ZK), where zero-knowledge holds even in the case of a dishonestly generated CRS, and constructed a Sub-ZK NIZK argument. Subsequently, [Abd+17; Fuc18; Abd+20b] constructed Sub-ZK SNARKs and [Abd+20a] constructed succinct Sub-ZK quasi-adaptive NIZKs [JR13]. As noted in [Abd+20a], Sub-ZK in the CRS model is equivalent to zero-knowledge in the minimal bare public key (BPK, [Can+00]) model where the authority is only trusted to store the public key of each party. Since auxiliary-string non-black-box NIZK is impossible in the BPK model [GO94], one needs to use non-auxiliary-string non-black-box techniques to achieve Sub-ZK [Abd+20a]. Existing Sub-ZK NIZKs extract a CRS trapdoor from the (possibly malicious) CRS generator, and then use the CRS trapdoor to simulate the NIZK

argument. Prior to our work, extraction in Sub-ZK NIZKs was done using a concrete knowledge-of-exponent assumption.

As previously mentioned, the work of Bitansky et al. [Bit+12] established that extractable collision-resistant hash functions are necessary to obtain adaptive soundness of SNARKs. A natural extension of this question is then to ask:

> Which assumptions are necessary to obtain Sub-ZK for NIZKs and SNARKs? Are those assumptions stronger than the ones required to obtain adaptive soundness of SNARKs?

Our Contributions

Inspired by (G)EOWFs, we propose a new *generic assumption*[3]: the existence of verifiably-extractable (generalized) OWFs (VE(G)OWFs). We argue that VEGOWFs are a natural extension of GEOWFs introduced by Bitansky et al. [Bit+16], and show that in fact their GEOWF construction can easily be turned into a VEGOWF. Moreover, while Bitansky et al. [Bit+16] showed that a GEOWF can be transformed into a EOWF under certain assumptions, we similarly show that any VEGOWF can be transformed into a VEOWF with no further assumptions. To circumvent the impossibility result that EOWF and similar primitives do not exist assuming indistinguishability obfuscation, our definitions include non-black-box extractability as in [Bit+16] and assume a benign distribution of auxiliary inputs as suggested in [BP15b].

Answering the first research question, we show that VEGOWFs are vital in understanding subversion zero-knowledge. Firstly, we show that VEGOWFs allow for the transformation of any perfect NIZK with a publicly verifiable CRS into a Sub-ZK NIZK. Secondly, we show the necessity of VEGOWFs by showing that the existence of a Sub-ZK NIZK with certain properties implies that the NIZK's CRS generation algorithm must be a VEOWF. We also prove that if a NIZK has perfect zero-knowledge and well-formedness of the CRS can be efficiently verified, then we automatically obtain a statistical two-message private-coin witness-indistinguishable argument. Obtaining statistical two-message witness-indistinguishable arguments (either public or private coin) was an open question until recently [Bad+20; Goy+20; LVW20]. Similar observations were previously made about specific Sub-ZK SNARKs in[Fuc18].

We answer the second research question by showing that the assumption corresponding to this primitive seems weaker than that of extractable collision-resistant hash functions. In particular, we show that VEGOWFs can be built either from knowledge assumption or knowledge-sound NIZKs, and we also propose candidate VEGOWFs from various signature schemes.

[3]Generic assumptions postulate the existence of a cryptographic primitive, such as OWFs and one-way permutations. Meanwhile, concrete assumptions are used for concrete constructions, such as the RSA assumption [RSA78] for the RSA cryptosystem.

By showing connections to Sub-ZK NIZK, our work further demonstrates the importance of extractable OWFs as an independent primitive. This tool, which has not been thoroughly studied, seems to lead the way to protocols that are otherwise difficult to achieve. We encourage further study into extractable functions under weaker (or different) assumptions as there are significant differences between various non-black-box techniques.

6.2 Technical Overview

Extending the notions of EOWF [CD08] and GEOWF [Bit+16], we define *Verifiably-Extractable Generalized One-Way Functions* (VEGOWFs), show several instantiations of these and show how it is related to subversion resistant zero-knowledge. Intuitively, an EOWF f is a one-way function such that for any PPT adversary $\mathcal{A}$, there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$, such that if $\mathcal{A}$ outputs $y \in \mathrm{im}(f)$, then $\mathsf{Ext}_{\mathcal{A}}$ (given access to $\mathcal{A}$'s auxiliary input) retrieves x such that $f(x) = y$. Meanwhile, a GEOWF g generalizes EOWFs by introducing a relation $\mathbf{RG}$ such that for every PPT $\mathcal{A}$, there exists an extractor $\mathsf{Ext}_{\mathcal{A}}$, such that if $\mathcal{A}$ outputs $y \in \mathrm{im}(g)$, then $\mathsf{Ext}_{\mathcal{A}}$ (given access to $\mathcal{A}$'s auxiliary input) returns z such that $(y, z) \in \mathbf{RG}$. It is required that it is difficult for any adversary who is only given y to compute such z, i.e., $\mathbf{RG}$ is a hard relation.

Verifiably-Extractable (Generalized) OWFs

A *Verifiably-Extractable Generalized OWF* (VEGOWF) $\mathcal{G} = \{\mathsf{g}_{\mathsf{e}}\}_{\mathsf{e}}$ is a GEOWF which additionally allows one to efficiently check whether extraction will succeed for a given value y. More precisely, we define a relation $\mathbf{RG}_{\mathsf{e}}$ and a set $Y_{\mathsf{Ext}} \supseteq \mathrm{im}(\mathsf{g}_{\mathsf{e}})$ such that

(i) given y one can efficiently verify whether $y \in Y_{\mathsf{Ext}}$ and

(ii) if $y \in Y_{\mathsf{Ext}}$ then there exists an extractor $\mathsf{Ext}_{\mathcal{A}}$ that given non-black-box access to $\mathcal{A}$ extracts z such that $(y, z) \in \mathbf{RG}_{\mathsf{e}}$.

Note that extraction should work even if $y \in Y_{\mathsf{Ext}} \setminus \mathrm{im}(\mathsf{g}_{\mathsf{e}})$, and in general, it might be hard to decide if $y \in \mathrm{im}(\mathsf{g}_{\mathsf{e}})$. We say that a VEGOWF is keyless if e is the security parameter λ; in this case we write $\mathbf{RG}$ instead of $\mathbf{RG}_{\mathsf{e}}$. The formal definition of VEGOWFs can be found in Section 6.4.

We denote both properties together as $\mathbf{RG}$-*verifiable-extractability*. The requirements for $\mathbf{RG}$-hardness remain the same as for GEOWFs. We introduce *verifiably-extractable OWFs* (VEOWF) as a special case of VEGOWFs where the corresponding relation is $\mathbf{RG}_{\mathsf{e}} = \{(\mathsf{g}_{\mathsf{e}}(x), x)\}$.

Generic transformations. We show that any VEGOWF can be transformed to a VEOWF with a simple technique that was first mentioned in [Bit+16], in a slightly different context. However, since the transformation incurs some efficiency loss, we still consider VEGOWFs to be a weaker primitive and base our subversion zero-knowledge

application on VEGOWFs. We also give a construction of a VEGOWF from any GEOWF by evaluating the GEOWF on two different inputs and attaching a NIWI proof (in the plain model) that at least one of the functions was evaluated correctly. Together they give a surprising result that any GEOWF can be transformed to a VEOWF under the relatively mild assumptions (e.g., decisional linear assumption) required by the NIWI. We note that similar techniques have been previously used in specific applications. For example, [Bit+17] uses similar idea to obtain a 3-round zero-knowledge argument from any (non-verifiable) EOWF. We believe it is valuable to point out that this technique works as a general transformation. See Section 6.4 for more details.

Robust Combiners. We show that n VEGOWFs can be combined to a new VEGOWF, which is secure if any $t > n/2$ of the initial functions is secure. A robust combiner [Har+05; Her05; FLP08] for VEGOWFs is useful since many of the proposed VEGOWFs rely on strong assumptions. With combining we only need to trust that some of those strong assumptions hold without knowing which. Details are provided in Section 6.4.

We show several VEGOWFs and VEOWFs under various assumptions like bounded auxiliary input size, knowledge assumptions, and the random oracle.

VEGOWF from the BCPR construction. In the first construction, we show that the keyless GEOWF $\mathcal{G}$ from [Bit+16, Fig. 4] is, in fact, a VEGOWF against any adversary with bounded auxiliary input if we assume that the used delegation scheme has efficient public CRS-verifiability. We recall that a delegation scheme DS [Aie+00] allows one to prove statements of the form "a machine $\mathcal{M}$ outputs y on input x in time t". A delegation proof π_{DS} must be faster to verify than the statement itself. The CRS-verifiability means that one can efficiently check if the DS CRS $\mathsf{crs}_{\mathsf{DS}}$ is a valid CRS.

In the BCPR construction, each function g_{e} computes a CRS $\mathsf{crs}_{\mathsf{DS}}$ for a delegation scheme DS, and then evaluates a PRG on a random value. The relation $\mathbf{RG}(y, z)$ holds for $y = (\mathsf{crs}_{\mathsf{DS}}, v)$ and $z = (\mathcal{A}, \pi_{\mathsf{DS}}, \mathsf{pad})$, if π_{DS} is a DS-proof, using $\mathsf{crs}_{\mathsf{DS}}$ as the CRS, for the statement that $\mathcal{A}$ on input 1^λ outputs v. (pad is a padding.) The proof of $\mathbf{RG}$-hardness is as in [Bit+16], and follows from the security of the PRG together with an argument about Kolmogorov complexity. The $\mathbf{RG}$-verifiable-extractability follows from the CRS-verifiability and completeness of the delegation scheme. See Section 6.4 for more details.

We note that even if the delegation scheme is not CRS-verifiable, one could still make the BCPR EOWF a VEGOWF using the generic transformation presented in Section 6.4.

VEGOWFs from knowledge-of-exponent assumptions. Secondly, we show that many knowledge-of-exponent assumptions naturally imply VEGOWFs. For these VEGOWFs, the input key e consists of a bilinear group description and possibly some additional information.

We first construct of a VEOWF based on the Bilinear Diffie–Hellman Knowledge-of-Exponent (BDH-KE) assumption from [Abd+17] which states that if an adversary on input p (the asymmetric bilinear group description) outputs $([x]_1, [x]_2)$ for some x then he knows x.[4] Here, $\mathsf{e} = \mathsf{p}$ and $\mathsf{g}_\mathsf{p}(x) = ([x]_1, [x]_2)$. See Section 6.C for more details.

We also construct a VEGOWF based on the Diffie–Hellman Knowledge of Exponent (DH-KE) assumption introduced in [BFS16]. The key is a description p of a symmetric bilinear group, and $\mathsf{g}_\mathsf{p}(x, y) = [x, y, xy]_1$. The DH-KE assumption states that is is possible to extract at least one of x and y. This results in a VEGOWF with respect to the relation $\mathbf{RG}_\mathsf{p}([x, y, xy]_1, z) = 1$ iff $z = x$ or $z = y$.

We discuss these and other similar VE(G)OWF constructions in Section 6.4.

VEGOWFs from knowledge sound NIZKs. Thirdly, inspired by [Dak09; Lep02], we build VEGOWFs using knowledge-sound NIZKs. Suppose that we have a knowledge-sound NIZK Π for a relation $\mathcal{R}$ and that $\mathcal{R}$ has an efficient sampling algorithm $\mathcal{S}$ which produces instances that are hard on average. We define $\mathsf{g}_\mathsf{e}(r_\mathcal{S}, r_\pi)$ such that it samples $(\mathtt{x}, \mathtt{w}) \leftarrow \mathcal{S}(r_\mathcal{S})$, uses r_π as random coins to generate a proof π for $\mathtt{x}$, and outputs $(\mathtt{x}, \pi)$. The input e is either the CRS or a description of a hash function (in the random oracle model). We define $\mathbf{RG}_\mathsf{e}((\mathtt{x}, \pi), \mathtt{w}) = 1$ iff π satisfies NIZK verification and $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$. Since Π is knowledge-sound, we obtain **RG**-verifiable-extractability by using Π's verification on $(\mathtt{x}, \pi)$. **RG**-hardness is satisfied since π is simulatable and $\mathcal{S}$ produces hard instances on average.

As an interesting instantiation, if we let $\mathcal{S}$ output $([x], x)$ for a random x and use Schnorr's Σ-protocol together with the Fiat-Shamir heuristic as a NIZK, we obtain a very efficient VEOWF $\mathsf{g}_\mathsf{e}(x, r) := (\mathtt{x} = [x], a = [r], z = H([x], [r]) \cdot x + r)$ where H is a hash function and verification works by asserting that $H(\mathtt{x}, a)\mathtt{x} + a = [z]$. See Section 6.4 for more details.

VEGOWFs from signature schemes. Finally, we propose a novel heuristic for coming up with new VEGOWFs and knowledge-type assumptions in general. The intuition behind signature schemes is that only the one with (at least some) knowledge of the signing key sk can sign a message. Thus, it gives a very simple formula for looking for new VEGOWFs. Let $\Sigma = (\mathsf{Kgen}, \mathsf{Sign}, \mathsf{Vf})$ be a digital signature scheme. Then, $\mathsf{g}_\mathsf{p}(\mathsf{sk}) = (\mathsf{vk} = \mathsf{Kgen}(\mathsf{sk}), \sigma = \mathsf{Sign}(\mathsf{sk}, m = 0))$ is a candidate for a VEGOWF where p is some parameter for the signature scheme, in particular when $\mathsf{vk} \in \mathsf{Kgen}$ can be efficiently tested. Of course, this is just a heuristic since at least the standard notion of existential unforgeability does not require that the signer knows the secret key.

We then proceed by going over many concrete signatures schemes and investigate the security of the corresponding VEGOWF candidate. We see that in some cases the VEGOWF is insecure (e.g., Lamport's one-time signature [Lam79] and RSA signature),

[4] We use the additive notation for bilinear groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ where $[x]_i$ denotes xg_i using the fixed generator g_i of $\mathbb{G}_i$ described in p. A bilinear map $\bullet$ allows us to compute $[x]_1 \bullet [y]_2 = [xy]_T$.

in some cases it gives a VEGOWF that we already considered before (e.g., Schnorr's signature scheme [Sch90] and Boneh-Boyen signature [BB04]) and in some cases we obtain (plausibly secure) VEGOWFs that have not been considered before. In the latter set is for example the DSA signature which gives quite a unique function in a non-pairing-based group and (and a slight modification of) the hash-and-sign lattice based signature scheme of [GPV08], which gives the first lattice based VEGOWF candidate.

Constructing Sub-ZK NIZK from VEGOWF

We propose two generic constructions of a Sub-ZK NIZK. The first construction produces a knowledge-sound Sub-ZK NIZK from any knowledge-sound Sub-WI NIWI[5] and keyless VEGOWF. The second construction produces a sound Sub-ZK NIZK from a sound Sub-WI NIWI, a keyless extractable commitment, and a VEGOWF.

Knowledge-sound Sub-ZK NIZK. For the first construction, we propose a knowledge-sound Sub-ZK NIZK for any NP-relation $\mathcal{R}$ using a variant of the well-known FLS disjunctive approach [FLS90]. Namely, we use a knowledge-sound Sub-WI NIWI Π_{wi} for the composite relation $\mathcal{R}'$, where $((\mathtt{x}, \widehat{y}), (\mathtt{w}, \widehat{z})) \in \mathcal{R}'$ iff either $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$ or $(\widehat{y}, \widehat{z}) \in \mathbf{RG}$. Here $\mathcal{G} = \{\mathtt{g_e}\}$ is a keyless VEGOWF with respect to $\mathbf{RG}$ and $\widehat{y} \in Y_{\mathsf{Ext}}$ being added to Π_{wi}'s CRS. Knowledge-soundness of the new protocol will follow from the knowledge-soundness of Π_{wi} together with the $\mathbf{RG}$-hardness of $\mathcal{G}$, and subversion zero-knowledge follows from the verifiable-extractability of $\mathcal{G}$ and the Sub-WI property of Π_{wi}. This construction preserves succinctness, and thus we obtain a Sub-ZK SNARK from a keyless VEGOWF and a Sub-WI SNARK. We later note that any perfectly zero-knowledge SNARK with efficient CRS verification is automatically a Sub-WI SNARK. See Section 6.5 for the full details of the construction.

Sub-ZK NIZK. Secondly, we construct a Sub-ZK NIZK Π for any NP-relation $\mathcal{R}$. It similarly uses the FLS approach with a keyless VEGOWF, but additionally uses a commitment to a trapdoor. Specifically, Π implements a Sub-WI NIWI Π_{wi} for the relation $\mathcal{R}'$, where $((\mathtt{x}, c, \widehat{y}), (\mathtt{w}, \widehat{z}, \widehat{r})) \in \mathcal{R}'$ iff $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$ or $c = \mathsf{C.com}(\widehat{z}, \widehat{r})$ such that $\mathbf{RG}(\widehat{y}, \widehat{z}) = 1$, where $\mathcal{G}$ is a keyless VEGOWF with respect to $\mathbf{RG}$ and $\mathsf{C} = (\mathsf{com}, \mathsf{Open}, \mathsf{Vf})$ is a keyless extractable commitment scheme.

A proof in Π consists of a commitment c and a proof in Π_{wi}, so this construction is less efficient than the previous one. However, this does not rely on Π_{wi} being knowledge-sound, so the construction is still of interest. The soundness of Π follows from the soundness of Π_{wi} together with the $\mathbf{RG}$-hardness of $\mathcal{G}$ and the extractability of C. Note that Π_{wi} will already guarantee that c is a valid commitment. Therefore, we do not need the commitment itself to have an efficient image verification procedure and can obtain

[5] Although in the literature NIWI often refers to the plain model, in this context we allow for a CRS. A Sub-WI NIWI needs to remain witness indistinguishable even if the CRS is subverted. We note that any CRS-less NIWI is trivially a Sub-WI NIWI.

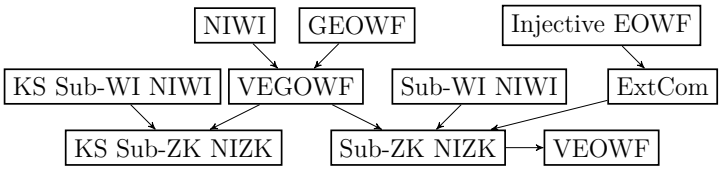


Figure 6.1: Relations between argument systems and extractable functions. Multiple arrows pointing to the same node means that each source node is required to construct the destination node. KS denotes knowledge-sound.

it from any (even non-verifiable) injective EOWF. Sub-ZK follows from the verifiable-extractability of $\mathcal{G}$, the Sub-WI property of Π_{wi} and the hiding property of C. See Section 6.5 for the full details of the construction.

Statistical ZAPRs with adaptive soundness. We observe that if a NIZK has perfect zero-knowledge and CRS-verifiability, then we immediately obtain a statistical two-message private-coin witness-indistinguishable argument. Obtaining statistical two-message witness-indistinguishable arguments that are public-coin (ZAP) or private-coin (ZAPR) was considered a significant open problem, until recent breakthroughs [Bad+20; Goy+20; LVW20]. Note that existing Sub-ZK SNARKs [Abd+17; Fuc18] are already statistical ZAPRs with adaptive soundness. Compared to previous statistical ZAP/ZAPR constructions, the soundness of SNARKs is based on less standard assumptions, but they have much better efficiency. Similar observations about Sub-ZK SNARKs were previously made by Fuchsbauer in [Fuc18].

Instantiations. The relations between our primitives are summarized in Figure 6.1.

Table 6.1 shows a selection of instantiations for our generic constructions and compares them to previous work. We can achieve a keyless extractable commitment from any keyless injective VEOWF (or even from keyless injective EOWF if the commitment does not have to be image verifiable). In particular, this includes a VEOWF based on the symmetric discrete logarithm (SDL) assumption and the BDH-KE assumption, and a VEOWF based on the security of a non-interactive version of Schnorr's protocol.

We can construct a Sub-ZK NIZK by combining a keyless extractable commitment, a VEGOWF, and a Sub-WI NIWI. For example, we may use the Sub-WI NIWI of [GOS06] based on DLIN or [BP15a] based on iO and OWF. In comparison, [BFS16] proposed a Sub-ZK NIZK which is based on the DLIN and DH-KE assumptions. We can obtain a KS Sub-ZK NIZK by combining a KS Sub-WI NIWI with a VEGOWF. In Table 6.1, we consider the case where we use [FO18] as the KS Sub-WI NIWI component, together with a VEGOWF which holds under the same assumptions. In Section 6.5, we also show that existing Sub-ZK SNARKs [Abd+17; Fuc18] can be slightly modified to achieve Sub-ZK from any VEGOWF rather than a specific knowledge-of-exponent assumption.

	Soundness	Knowledge Soundness	Sub-ZK
[BFS16]	DH-KE + CDH	×	DH-KE + DLIN
Sec. 6.5	injective VEOWF	×	injective VEOWF + DLIN
Sec. 6.5	injective VEOWF	×	injective VEOWF + iO
[Abd+17]	GGM	GGM	BDH-KE
[Fuc18, Sec. 4]	q_1-PDH + q_2-PKE	q_1-PDH + q_2-PKE	SKE
[Fuc18, Sec. 5]	q_1-PDH + q_2-PKE + q_3-SDH	q_1-PDH + q_2-PKE + q_3-TSDH	SKE
[Fuc18, Sec. 6]	GGM	GGM	SKE
Sec. 6.5	DH-KE + DL	DH-KE + DL	DH-KE + DLIN

Table 6.1: Instantiations of our generic constructions in comparison to previous work. SKE denotes the Square Knowledge-of-Exponent assumption, GGM denotes the generic group model, PDH denotes the Power Diffie-Hellman assumption, PKE denotes the Power Knowledge-of-Exponent assumption, and TSDH denotes the Target Strong Diffie-Hellman assumption.

Constructing VEOWF from Sub-ZK NIZK

It turns out that not only can Sub-ZK NIZK be constructed with the help of VEGOWF, but (under certain restrictions) Sub-ZK NIZK also implies a VE(G)OWF. In that sense, VEGOWF is both a necessary and a sufficient condition for achieving Sub-ZK NIZKs, similar to how ECRH (also, under certain restrictions) is a necessary and a sufficient condition for achieving a SNARK.

More technically, we consider a CRS generation function $\mathsf{Kgen}_{\mathcal{R},\mathsf{p}}$ of a Sub-ZK NIZK that takes as an input a randomly sampled trapdoor td and outputs a crs. We show that this function has to be one-way if the NIZK is both computationally sound and computationally zero-knowledge. Intuitively, if one-wayness would not hold, the soundness adversary could recover td and use the simulator to construct a proof for a false statement. We additionally require that $\mathsf{Kgen}_{\mathcal{R},\mathsf{p}}$ is injective to avoid the situation where one-wayness adversary computes td is which is particularly bad for simulation among all the possible preimages of crs. Verifiable-extractability property follows straightforwardly from the Sub-ZK property of the NIZK since it requires that td must be extractable. However, here we also need to make some slight restrictions. Namely, the Sub-ZK extractor should be able to extract the complete td, not only some part of it, which might still be sufficient for simulating the proof.

6.3 Preliminaries

Let PPT denote probabilistic polynomial-time. Let $\lambda \in \mathbb{N}$ be the security parameter. All adversaries are stateful. For an algorithm $\mathcal{A}$, let $\mathrm{im}(\mathcal{A})$ be the image of $\mathcal{A}$ (the set of valid outputs of $\mathcal{A}$), let $\mathsf{RND}_\lambda(\mathcal{A})$ denote the random tape of $\mathcal{A}$, and let $r \leftarrow\!\$\, \mathsf{RND}_\lambda(\mathcal{A})$ denote the random choice of values from $\mathsf{RND}_\lambda(\mathcal{A})$. We write that $y \in \mathrm{range}(\mathcal{A}(x))$ if there is non-zero probability that the algorithm $\mathcal{A}$ outputs a value y given the input x. We denote by $\mathsf{negl}(\lambda)$ an arbitrary negligible function and by $\mathsf{poly}(\lambda)$ an arbitrary

polynomial function. We write $a(\lambda) \approx_\lambda b(\lambda)$ if $|a(\lambda) - b(\lambda)| = \mathsf{negl}(\lambda)$. For an NP-relation $\mathcal{R} = \{(\mathtt{x}, \mathtt{w})\}$, let $\mathcal{L}_\mathcal{R} := \{\mathtt{x} : \exists \mathtt{w}, (\mathtt{x}, \mathtt{w}) \in \mathcal{R}\}$ be the corresponding language.

In the pairing-based setting, we use the standard bracket notation together with additive notation, i.e., we write $[a]_\iota$ to denote ag_ι where g_ι is a fixed generator of $\mathbb{G}_\iota$ and $a \in \mathbb{Z}_p$ for some prime p. Intuitively, pairings $\bullet : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ are efficient (one-way) functions that map $([a]_1, [b]_2)$ to $[a]_1 \bullet [b]_2 = [ab]_T$.

Let $A = \{A_\lambda\}_{\lambda\in\mathbb{N}}, B = \{B_\lambda\}_{\lambda\in\mathbb{N}}$ be collections of efficiently sampleable sets, such that $|B_\lambda| > |A_\lambda|$ for each $\lambda \in \mathbb{N}$. A polynomial-time function $\mathsf{PRG}\colon A_\lambda \to B_\lambda$ is a pseudorandom generator (PRG) if its output is computationally indistinguishable from a truly random one.

(Generalized) Extractable OWF

An extractable one-way function (EOWF, [CD08]) g is an OWF with the property that if $\mathcal{A}$ outputs a value in the image of g, then one can extract its preimage. A generalized EOWF (GEOWF, [Bit+16]) is a function g with an associated hard relation $\mathbf{RG}$, such that given $\mathsf{g}(x)$, it is intractable to compute z such that $\mathbf{RG}(\mathsf{g}(x), z) = 1$. However, given a machine (and its auxiliary input) that computes $\mathsf{g}(x)$, it is possible to extract z such that $\mathbf{RG}(\mathsf{g}(x), z) = 1$. One obtains an EOWF when $\mathbf{RG} = \{(\mathsf{g}(x), z) : \mathsf{g}(z) = \mathsf{g}(x)\}$. Unless stated otherwise, we assume that $\mathbf{RG}$ is efficiently checkable.

Bitansky *et al.* [Bit+16] show that, assuming the existence of indistinguishability obfuscation, there do not exist EOWFs or GEOWFs with common auxiliary-input of unbounded polynomial length. However, the result does not rule out their existence when the common auxiliary input comes from some natural distribution, such as the uniform distribution. Thus, nowadays zk-SNARKs explicitly assume that the auxiliary input is benign, i.e., with overwhelming probability it does not encode a malicious obfuscation. We also make the same assumption: if no bound for the auxiliary input is given, then we assume that it is taken from a benign distribution.

We present a slight modification of the GEOWF definition of [Bit+16]. Note that hardness is required to hold even against poly-length auxiliary inputs.

Definition 6.3.1 (GEOWFs)**.** Let $\mathcal{X} = \{X_\lambda\}_\lambda$, $\mathcal{Y} = \{Y_\lambda\}_\lambda$, $\mathcal{Z} = \{Z_\lambda\}_\lambda$ and $\mathcal{K} = \{K_\lambda\}_\lambda$ be collections of sets indexed by $\lambda \in \mathbb{N}$. An efficiently computable family of functions $\mathcal{G} = \{\mathsf{g}_\mathsf{e}\colon X_\lambda \to Y_\lambda \mid \mathsf{e} \in K_\lambda, \lambda \in \mathbb{N}\}$ associated with an efficient (probabilistic) key sampler $\mathsf{KeySamp}$, is a *GEOWF* with respect to a relation $\mathbf{RG}_\mathsf{e}(y, z)$ on triples $(\mathsf{e}, y, z) \in K_\lambda \times Y_\lambda \times Z_\lambda$ if it is:

RG-hard: for any PPT adversary $\mathcal{A}$ and any aux sampled from a benign distribution of $\mathsf{poly}(\lambda)$-bit strings

$$\Pr_{\substack{\mathsf{e}\leftarrow\mathsf{KeySamp}(1^\lambda)\\ x \xleftarrow{\$} X_\lambda}} [z \leftarrow \mathcal{A}(\mathsf{e}, \mathsf{g}_\mathsf{e}(x), \mathsf{aux}) : \mathbf{RG}_\mathsf{e}(\mathsf{g}_\mathsf{e}(x), z) = 1] \leq \mathsf{negl}(\lambda) \ .$$

RG-extractable: For any PPT adversary $\mathcal{A}$, there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$, s.t. for any benign distribution $\mathcal{D}_\lambda$ of $\mathsf{poly}(\lambda)$-bit strings,

$$\Pr_{\substack{\mathsf{e}\leftarrow\mathsf{KeySamp}(1^\lambda)\\ \mathsf{aux}\leftarrow\mathcal{D}_\lambda}}\begin{bmatrix} y\leftarrow\mathcal{A}(\mathsf{e};\mathsf{aux}), z\leftarrow\mathsf{Ext}_{\mathcal{A}}(\mathsf{e};\mathsf{aux}) : \\ y\in\mathrm{im}(\mathsf{g}_\mathsf{e})\wedge\mathbf{RG}_\mathsf{e}(y,z)\neq 1\end{bmatrix}\leq\mathsf{negl}(\lambda)\ .$$

The function is *publicly verifiable* if there exists a polynomial-time tester $\mathcal{T}$ such that for any (e,x,z), $\mathbf{RG}_\mathsf{e}(\mathsf{g}_\mathsf{e}(x),z)=\mathcal{T}(\mathsf{e},\mathsf{g}_\mathsf{e}(x),z)$.

We say that a GEOWF is keyless if, for each security parameter λ, there is only one key $\mathsf{e}=1^\lambda$. For ease of notation, we simply write g_λ and $\mathbf{RG}$ in this case. A GEOWF is an EOWF if $\mathbf{RG}_\mathsf{e}(\mathsf{g}_\mathsf{e}(x),z)=\{(\mathsf{e},\mathsf{g}_\mathsf{e}(x),z):\mathsf{g}_\mathsf{e}(x)=\mathsf{g}_\mathsf{e}(z)\}$.

Bounded auxiliary input. We also consider GEOWFs where the auxiliary input in $\mathbf{RG}$-extractability holds for any $\mathsf{aux}\in\{0,1\}^{\mathfrak{b}(\lambda)}$ (not just for a benign distribution) for some fixed polynomial $\mathfrak{b}$. We call these $\mathfrak{b}$-bounded GEOWFs.

BCPR GEOWF and EOWF

Bitansky *et al.* [Bit+16] show that if the common auxilliary string of the adversary and the extractor has an a priori bounded length $\mathfrak{b}(\lambda)$, then one can implement extractable one-way functions (EOWF) based on a pseudorandom generator and a universal delegation scheme [KRR14; KPY19]. In a universal delegation scheme (cf. Section 6.A), one delegates computation of some circuit M on input x to a prover, who must compute $M(x)$ and provide a proof π that he computed it correctly; any verifier that is given $(M,x,M(x),\pi)$ must be able to verify the proof in less time than computing $M(x)$ itself. One can construct universal delegation schemes under the subexponential learning with errors assumption [KRR14] and even falsifiable assumptions [KPY19] for languages in BPP.

BCPR GEOWF. We briefly describe the construction from [Bit+16] of a GEOWF secure against an adversary with $(\mathfrak{b}(\lambda)-\omega(1))$-bounded auxiliary input.

Fix a polynomial $\mathfrak{b}(\lambda)$. Let $\mathsf{PRG}\colon\{0,1\}^\lambda\to\{0,1\}^{\mathfrak{b}(\lambda)+\lambda}$ be a PRG. Let DS be a *universal* delegation scheme that consists of a CRS generator $\mathsf{DS.K}$, a prover $\mathsf{DS.P}$, and a verifier $\mathsf{DS.V}$. We assume that using DS, one can construct a succinct proof π_DS of length $\mathsf{DS.plen}(\lambda)$ that a Turing machine M on input 1^λ outputs some value v in time $T(\lambda)$, where $T(\lambda)\in(2^{\omega(\log\lambda)},2^{\mathsf{poly}(\lambda)})$ is some superpolynomial function. DS must satisfy that the proof verification complexity is linear in M's size and polylogarithmic in M's execution time T.

We define the function $\mathsf{g}_\lambda\colon(s,r)\mapsto(\mathsf{crs}_\mathsf{DS},v)$ and the corresponding relation $\mathbf{RG}(y,z)$ as in Figure 6.2, where $y=(\mathsf{crs}_\mathsf{DS},v)$ and $z=(\mathsf{M},\pi_\mathsf{DS},\mathsf{pad})$ with $|z|=\mathtt{lpar}(\lambda)$.

Proposition 6.3.2 ([Bit+16, Theorem 14])**.** *$\mathcal{G}=\{\mathsf{g}_\lambda\}_{\lambda\in\mathbb{N}}$, depicted in Figure 6.2, is a GEOWF with respect to $\mathbf{RG}$, against $(\mathfrak{b}(\lambda)-\omega(1))$-bounded auxiliary input.*

$\mathsf{g}_\lambda(s, r)$

$(\mathsf{crs}_{\mathsf{DS}}, \tau) \leftarrow \mathsf{DS.K}(1^\lambda; r)$; // the generator for universal delegation
return $(\mathsf{crs}_{\mathsf{DS}}, v \leftarrow \mathsf{PRG}(s))$;

$\mathbf{RG}(y, z)$

parse $y = (\mathsf{crs}_{\mathsf{DS}}, v), z = (\mathsf{M}, \pi_{\mathsf{DS}}, \mathsf{pad})$;
// $|\mathsf{M}| = \mathfrak{b}(\lambda), |\pi_{\mathsf{DS}}| = \mathsf{DS.plen}(\lambda), |\mathsf{pad}| = \mathtt{lpar}(\lambda) - \mathfrak{b}(\lambda) - \mathsf{DS.plen}(\lambda)$;
find the verification state τ corresponding to the reference string $\mathsf{crs}_{\mathsf{DS}}$;
verify the statement "$\mathsf{M}(1^\lambda)$ outputs v in $T(\lambda)$ steps" by using π_{DS} (DS proof);
return 1 iff the DS verifier accepts π_{DS};

Figure 6.2: BCPR GEOWF $\mathcal{G}$ (above) and the relation $\mathbf{RG}(y, z)$ (below).

This proposition relies on the security of DS and PRG. In addition, it uses a Barak-type [Bar01] extractability paradigm (namely, the machine M is the adversary who outputs y). It is worth noting that a similar approach with a number of extra steps [Bit+16, Theorem 14] also allows one to construct a function family which is an EOWF against $(\mathfrak{b}(\lambda) - \omega(1))$-bounded auxiliary-input. We will see an adaptation of this approach in Section 6.4.

NIZK and NIWI Arguments

We recall the definition of NIZK and NIWI arguments and their security properties. We assume that RGen is a relation generator that output an NP relation $\mathcal{R}$ and a parameter p (e.g., the group description). An argument system Ψ is a tuple of PPT algorithms $(\mathsf{K}, \mathsf{P}, \mathsf{V})$. The CRS generation algorithm K takes in $(\mathcal{R}, \mathsf{p})$ and outputs a crs and a trapdoor td (which may be $\perp$ if the argument does not have zero-knowledge). The prover algorithm P takes in $\mathcal{R}, \mathsf{p}, \mathsf{crs}$ and $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$ and outputs a proof π. The verifier algorithm V takes in $(\mathcal{R}, \mathsf{p}, \mathsf{crs}, \mathtt{x}, \pi)$ and outputs either 0 (rejecting the proof) or 1 (accepting the proof). A NIZK argument system will additionally have a simulator Sim that takes in $(\mathcal{R}, \mathsf{p}, \mathsf{crs}, \mathsf{td}, \mathtt{x})$ and outputs a simulated proof π for the statement $\mathtt{x}$. Furthermore, a subversion resistant argument will have a CRS verification algorithm CV that take in $(\mathcal{R}, \mathsf{p}, \mathsf{crs})$ and output either 0 (by rejecting the CRS) or 1 (by accepting the CRS).

Definition 6.3.3 (Perfect Completeness [Gro16])**.** A non-interactive argument Ψ is *perfectly complete for* RGen, if for all λ, all $(\mathcal{R}, \mathsf{p}) \in \mathrm{range}(\mathsf{RGen}(1^\lambda))$, and $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$,

$$\Pr\left[\mathsf{crs} \leftarrow \mathsf{K}(\mathcal{R}, \mathsf{p}) : \mathsf{V}(\mathcal{R}, \mathsf{p}, \mathsf{crs}, \mathtt{x}, \mathsf{P}(\mathcal{R}, \mathsf{p}, \mathsf{crs}, \mathtt{x}, \mathtt{w})) = 1\right] = 1 \ .$$

Definition 6.3.4 (Perfect CRS Verifiability)**.** A non-interactive (subversion-resistant) argument Ψ is *perfectly CRS-verifiable for* RGen, if for all λ and all $(\mathcal{R}, \mathsf{p}) \in \mathrm{range}(\mathsf{RGen}(1^\lambda))$, $\Pr\left[(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{K}(\mathcal{R}, \mathsf{p}) : \mathsf{CV}(\mathcal{R}, \mathsf{p}, \mathsf{crs}) = 1\right] = 1$.

Definition 6.3.5 (Computational Soundness)**.** Ψ is computationally (adaptively) *sound for* RGen, if for every PPT $\mathcal{A}$,

$$\Pr\left[\begin{array}{l}(\mathcal{R},\mathsf{p})\leftarrow\mathsf{RGen}(1^\lambda),(\mathsf{crs},\mathsf{td})\leftarrow\mathsf{K}(\mathcal{R},\mathsf{p}),(\mathtt{x},\pi)\leftarrow\mathcal{A}(\mathcal{R},\mathsf{p},\mathsf{crs}):\\ \mathtt{x}\notin\mathcal{L}_{\mathcal{R}}\wedge\mathsf{V}(\mathcal{R},\mathsf{p},\mathsf{crs},\mathtt{x},\pi)=1\end{array}\right]\leq\mathsf{negl}(\lambda)\ .$$

Definition 6.3.6 (Computational Knowledge Soundness)**.** Ψ is computationally (adaptively) *knowledge-sound for* RGen, if for every PPT $\mathcal{A}$, there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$, such that

$$\Pr\left[\begin{array}{l}(\mathcal{R},\mathsf{p})\leftarrow\mathsf{RGen}(1^\lambda),(\mathsf{crs},\mathsf{td})\leftarrow\mathsf{K}(\mathcal{R},\mathsf{p}),r\leftarrow\!\!\$\,\mathsf{RND}_\lambda(\mathcal{A}),\\ (\mathtt{x},\pi)\leftarrow\mathcal{A}(\mathcal{R},\mathsf{p},\mathsf{crs};r),\mathtt{w}\leftarrow\mathsf{Ext}_{\mathcal{A}}(\mathcal{R},\mathsf{p},\mathsf{crs};r):\\ (\mathtt{x},\mathtt{w})\notin\mathcal{R}\wedge\mathsf{V}(\mathcal{R},\mathsf{p},\mathsf{crs},\mathtt{x},\pi)=1\end{array}\right]\leq\mathsf{negl}(\lambda)\ .$$

Above we assume that the input $(\mathcal{R},\mathsf{p},\mathsf{crs};r)$ comes from a benign distribution and thus avoids the impossibility result of [Bit+16].

Definition 6.3.7 (Statistically Composable ZK [Gro06])**.** Ψ is *statistically composable zero-knowledge for* RGen, if for all $(\mathcal{R},\mathsf{p})\in\mathrm{range}(\mathsf{RGen}(1^\lambda))$, and all computationally unbounded $\mathcal{A}$, $\varepsilon_0^{comp}\approx_\lambda\varepsilon_1^{comp}$, where $\varepsilon_b^{comp}=$

$$\Pr\left[\begin{array}{l}(\mathsf{crs},\mathsf{td})\leftarrow\mathsf{K}(\mathcal{R},\mathsf{p}),(\mathtt{x},\mathtt{w})\leftarrow\mathcal{A}(\mathcal{R},\mathsf{p},\mathsf{crs},\mathsf{td});\pi_0\leftarrow\mathsf{P}(\mathcal{R},\mathsf{p},\mathsf{crs},\mathtt{x},\mathtt{w});\\ \pi_1\leftarrow\mathsf{Sim}(\mathcal{R},\mathsf{p},\mathsf{crs},\mathsf{td},\mathtt{x}):(\mathtt{x},\mathtt{w})\in\mathcal{R}\wedge\mathcal{A}(\pi_b)=1\end{array}\right]\ .$$

Ψ is *perfectly composable ZK for* RGen if one requires that $\varepsilon_0^{comp}=\varepsilon_1^{comp}$. In Theorem 6.6.3 we also consider a computational version of this definition, that is $\mathcal{A}$ is a PPT adversary and the input td is not given as input to $\mathcal{A}$.

Definition 6.3.8 (Statistically Composable Sub-ZK [Abd+17])**.** Ψ is *statistically composable subversion ZK (Sub-ZK) for* RGen, if for any PPT subverter $\mathcal{Z}$ there exists a PPT $\mathsf{Ext}_{\mathcal{Z}}$, such that for all $\mathcal{R}\in\mathrm{range}(\mathsf{RGen}(1^\lambda))$, and all computationally unbounded $\mathcal{A}$, $\varepsilon_0^{comp}\approx_\lambda\varepsilon_1^{comp}$, where $\varepsilon_b^{comp}=$

$$\Pr\left[\begin{array}{l}r\leftarrow\!\!\$\,\mathsf{RND}_\lambda(\mathcal{Z}),(\mathsf{crs},\mathsf{aux}_{\mathcal{Z}})\leftarrow\mathcal{Z}(\mathcal{R},\mathsf{p};r),\mathsf{td}\leftarrow\mathsf{Ext}_{\mathcal{Z}}(\mathcal{R},\mathsf{p};r)\\ (\mathtt{x},\mathtt{w})\leftarrow\mathcal{A}(\mathcal{R},\mathsf{p},\mathsf{crs},\mathsf{td},\mathsf{aux}_{\mathcal{Z}}),\pi_0\leftarrow\mathsf{P}(\mathcal{R},\mathsf{p},\mathsf{crs},\mathtt{x},\mathtt{w});\\ \pi_1\leftarrow\mathsf{Sim}(\mathcal{R},\mathsf{p},\mathsf{crs},\mathsf{td},\mathtt{x}):(\mathtt{x},\mathtt{w})\in\mathcal{R}\wedge\mathsf{CV}(\mathcal{R},\mathsf{p},\mathsf{crs})=1\wedge\mathcal{A}(\pi_b,\mathsf{aux}_{\mathcal{Z}})=1\end{array}\right].$$

Ψ is *perfectly composable Sub-ZK for* RGen if one requires that $\varepsilon_0^{comp}=\varepsilon_1^{comp}$.

Definition 6.3.9 (Witness Indistinguishability)**.** Ψ is *computationally witness indistinguishable (WI) for* RGen, if for any PPT $\mathcal{A}$, $\varepsilon_0^{wi}\approx_\lambda\varepsilon_1^{wi}$, where $\varepsilon_b^{wi}=$

$$\Pr\left[\begin{array}{l}(\mathsf{crs},\mathsf{td})\leftarrow\mathsf{K}(\mathcal{R},\mathsf{p}),(\mathtt{x},\mathtt{w}_0,\mathtt{w}_1)\leftarrow\mathcal{A}(\mathcal{R},\mathsf{p},\mathsf{crs}),\pi_b\leftarrow\mathsf{P}(\mathcal{R},\mathsf{p},\mathsf{crs},\mathtt{x},\mathtt{w}_b):\\ (\mathtt{x},\mathtt{w}_0)\in\mathcal{R}\wedge(\mathtt{x},\mathtt{w}_1)\in\mathcal{R}\wedge\mathcal{A}(\pi_b)=1\end{array}\right].$$

Ψ is *perfectly WI for* RGen if one requires that $\varepsilon_0^{wi}=\varepsilon_1^{wi}$ for unbounded $\mathcal{A}$. Note that td above might be $\perp$ if Ψ is not zero-knowledge.

Definition 6.3.10 (Sub-WI [BFS16])**.** Ψ is *computationally Sub-WI for* RGen, if for any PPT subverter $\mathcal{Z}$, $\varepsilon_0^{wi} \approx_\lambda \varepsilon_1^{wi}$, where $\varepsilon_b^{wi} =$

$$\Pr\left[\begin{matrix}(\mathsf{crs}, \mathtt{x}, \mathtt{w}_0, \mathtt{w}_1, \mathsf{aux}_{\mathcal{Z}}) \leftarrow \mathcal{Z}(\mathcal{R}, \mathsf{p}), \pi_b \leftarrow \mathsf{P}(\mathcal{R}, \mathsf{p}, \mathsf{crs}, \mathtt{x}, \mathtt{w}_b) : \\ (\mathtt{x}, \mathtt{w}_0) \in \mathcal{R} \wedge (\mathtt{x}, \mathtt{w}_1) \in \mathcal{R} \wedge \mathsf{CV}(\mathcal{R}, \mathsf{p}, \mathsf{crs}) = 1 \wedge \mathcal{Z}(\pi_b, \mathsf{aux}_{\mathcal{Z}}) = 1\end{matrix}\right].$$

Ψ is *perfectly Sub-WI for* RGen if one requires that $\varepsilon_0^{wi} = \varepsilon_1^{wi}$ for an unbounded $\mathcal{Z}$. In case Ψ does not utilise any common string we assume $\mathsf{CV}(\mathcal{R}, \mathsf{p}, \varepsilon) = 1$.

6.4 Verifiably-Extractable Generalized OWFs

Definition

We study GEOWFs $\mathcal{G} = \{\mathsf{g}_\mathsf{e}\}$ that come with an efficient (public) algorithm that decides whether or not extraction is going to be successful. That is, we require that there exists an extraction verification algorithm EV, such that $\mathsf{EV}(\mathsf{e}, y)$ decides whether $y \in Y_{\mathsf{Ext}} \supseteq \mathrm{im}(\mathsf{g}_\mathsf{e})$, where extraction succeeds for any $y \in Y_{\mathsf{Ext}}$. We also require that, with overwhelming probability, extraction is successful for any adversary which outputs a value in Y_{Ext}. (Extraction *may* succeed even if $y \notin Y_{\mathsf{Ext}}$.) We call GEOWFs with such properties *Verifiably-Extractable Generalized OWFs* (VEGOWFs).

Although for some VEGOWFs it may hold that $Y_{\mathsf{Ext}} = \mathrm{im}(\mathsf{g}_\mathsf{e})$, it is not necessarily the case. For example in the BCPR GEOWF, one is not able to decide if $y \in \mathrm{im}(\mathsf{g}_\lambda)$, because any such algorithm can be used to decide membership in $\mathrm{im}(\mathsf{PRG})$ which contradicts the security of PRG. However, as we will show, extraction is successful for any $y = (\mathsf{crs}_{\mathsf{DS}}, v)$, where $\mathsf{crs}_{\mathsf{DS}}$ is a valid DS CRS and v is *any* string output by an adversary with bounded auxiliary input.

Define VEGOWFs as GEOWFs where the **RG**-extractability property has been substituted with the following, stronger one. (It makes an implicit assumption that EV exists.)

RG-verifiably-extractable with respect to Y_{Ext}**:** Let $\mathrm{im}(\mathsf{g}_\mathsf{e}) \subseteq Y_{\mathsf{Ext}} \subseteq Y_\lambda$, and let EV be an efficient algorithm such that $\mathsf{EV}(\mathsf{e}; y) = 1$ iff $y \in Y_{\mathsf{Ext}}$. For any PPT adversary $\mathcal{A}$, there exists a PPT extractor Ext, s.t. for any benign distribution $\mathcal{D}_\lambda$ of $\mathsf{poly}(\lambda)$-bit strings,

$$\Pr_{\substack{\mathsf{e} \leftarrow \mathsf{KeySamp}(1^\lambda) \\ \mathsf{aux} \leftarrow \mathcal{D}_\lambda}}\left[\begin{matrix}y \leftarrow \mathcal{A}(\mathsf{e}; \mathsf{aux}), z \leftarrow \mathsf{Ext}(\mathsf{e}; \mathsf{aux}) : \\ y \in Y_{\mathsf{Ext}} \wedge (y, z) \notin \mathbf{RG}_\mathsf{e}\end{matrix}\right] \leq \mathsf{negl}(\lambda) \ .$$

If this definition holds for adversaries with auxiliary input length bounded by some polynomial $\mathfrak{b}(\lambda)$, we say that that the GEOWF is **RG**-*verifiably-extractable against* $\mathfrak{b}$-*bounded adversaries with respect to* Y_{Ext}.

$\mathsf{f}_\mathsf{e}(i \in \{0,1\}^\lambda, x \in X_\lambda, y \in Y_\lambda, z \in X_\lambda)$	$\mathsf{ImV}_\mathsf{f}(\mathsf{e}; y)$
if $i \neq 0^\lambda$ **then return** $\mathsf{g}_\mathsf{e}(x)$; **elseif** $(y,z) \in \mathbf{RG}_\mathsf{e} \wedge \mathsf{EV}_\mathsf{g}(\mathsf{e}; y)$ **then return** y; **else return** $\perp$;	**return** $\mathsf{EV}_\mathsf{g}(\mathsf{e}; y) \vee y = \perp$;

Figure 6.3: Transformation from a VEGOWF $\mathcal{G} = \{\mathsf{g}_\mathsf{e}\}_\mathsf{e}$ to a VEOWF $\mathcal{F} = \{\mathsf{f}_\mathsf{e}\}_\mathsf{e}$.

We also require that there is a PPT algorithm t, such that for any $x \in X_\lambda$, $(\mathsf{g}_\mathsf{e}(x), t(x)) \in \mathbf{RG}_\mathsf{e}$, that is, given x, t computes the "witness" for $\mathsf{g}_\mathsf{e}(x)$ in $\mathbf{RG}$.

If there exists an algorithm ImV that decides membership in $\mathrm{im}(\mathsf{g}_\mathsf{e})$, then the GEOWF is *image-verifiable*. Clearly, any image-verifiable GEOWF is also verifiably-extractable with respect to $Y_{\mathsf{Ext}} = \mathrm{im}(\mathsf{g}_\mathsf{e})$. Furthermore, for an EOWF, $\mathbf{RG}_\mathsf{e}$ only consists of pairs $(\mathsf{g}_\mathsf{e}(x), x)$ so extraction is not possible if one is given $y \notin \mathrm{im}(\mathsf{g}_\mathsf{e})$. Hence, for an EOWF, verifiable-extractability is the same as image-verifiability.

Generic transformations

VEGOWF ⇒ VEOWF. Surprisingly, any VEGOWF can be transformed to a VEOWF with the transformation in Figure 6.3 that adds very little overhead. The idea is to include to a VEGOWF g_e a branch input $i \in \{0,1\}^\lambda$. If $i \neq 0^\lambda$, which happens with an overwhelming probability, then g_e works as usual and outputs $\mathsf{g}_\mathsf{e}(x)$. However, on a trapdoor branch $i = 0^\lambda$, the function uses its two extra inputs y and z. If y satisfies $\mathsf{EV}_\mathsf{g}(\mathsf{e}; y)$ and $(y,z) \in \mathbf{RG}_\mathsf{e}$, it outputs y (or $\perp$ if the condition is not met). One-wayness follows since with overwhelming probability the function outputs $y \in \mathrm{im}(\mathsf{g}_\mathsf{e})$ and the preimage has to contain either x such that $\mathsf{g}_\mathsf{e}(x) = y$ or z such that $(y,z) \in \mathbf{RG}_\mathsf{e}$. By outputting either $t(x)$ (in the first case) or z (in the other case), one breaks $\mathbf{RG}$-hardness. On the other hand, the VEOWF extractor can use the VEGOWF extractor to recover z from y when $\mathsf{EV}_\mathsf{g}(\mathsf{e}; y) = 1$ and then return a preimage $(0^\lambda, \perp, y, z)$.

A similar transformation was introduced in [Bit+16] to obtain EOWFs from GEOWFs. However, they observed that an adversary can pick as input $(0^\lambda, \perp, y, z)$ with $(y,z) \in \mathbf{RG}_\mathsf{e}$, but $y \notin \mathrm{im}(\mathsf{g}_\mathsf{e})$. This makes the extraction impossible. Our construction does not run into this issue since we assume that extraction is possible when $\mathsf{EV}(\mathsf{e}; y) = 1$.

Theorem 6.4.1. *If $\mathcal{G} = \{\mathsf{g}_\mathsf{e}\}_\mathsf{e}$ is $\mathbf{RG}$-hard and $\mathbf{RG}$-verifiably-extractable, then $\mathcal{F} = \{\mathsf{f}_\mathsf{e}\}_\mathsf{e}$ in Figure 6.3 is a VEOWF.*

GEOWFs ⇒ VEGOWF. We now consider a generic transformation from a GEOWF to a VEGOWF. One approach is to simply append a NIZK proof π which proves that the given value was computed correctly. A problem with this approach is that it would require a CRS computed by a trusted third party, which might not be desirable in a number of settings. We therefore give a modification of this approach, where we instead rely on a

$\mathsf{g}_\mathsf{e}(x_1, x_2, r)$	$\mathsf{EV}(\mathsf{e}; (y_1, y_2, \pi))$
$y_1 \leftarrow \mathsf{f}_\mathsf{e}(x_1); y_2 \leftarrow \mathsf{f}_\mathsf{e}(x_2);$ $\pi \leftarrow \mathsf{P}\left(\mathcal{R}_\mathsf{e}, (\mathsf{f}_\mathsf{e}(x_1), \mathsf{f}_\mathsf{e}(x_2)), (x_1, x_2); r\right);$ **else return** (y_1, y_2, π);	**return** $\mathsf{V}(\mathcal{R}_\mathsf{e}, (y_1, y_2), \pi)$;

Figure 6.4: Transformation from a GEOWF $\mathcal{F} = \{\mathsf{f}_\mathsf{e}\}_\mathsf{e}$ to a VEGOWF $\mathcal{G} = \{\mathsf{g}_\mathsf{e}\}_\mathsf{e}$.

NIWI, which are known to exist in the plain model under various assumptions [BOV03; GOS06; BP15a].

The intuition is that we create a new function $g(x, y, r) = (f(x), f(y), \pi)$ where π is a NIWI proof (created using randomness r) showing that either $f(x)$ or $f(y)$ belongs to the image of f (in which case extraction will be possible). Verifiable-extractability follows from extractability of the GEOWF as well as perfect soundness of the NIWI, and hardness will follow from the hardness of f and witness-indistinguishability of the NIWI.

Consider a GEOWF $\mathcal{F} = \{\mathsf{f}_\mathsf{e}\}_\mathsf{e}$ with an associated relation $\mathbf{RG}$. Let $\Pi = (\mathsf{P}, \mathsf{V})$ be a perfectly sound NIWI, and let the relation $\mathcal{R}_\mathsf{e}((y_1, y_2), (x_1', x_2'))$ hold iff $y_1 = \mathsf{f}_\mathsf{e}(x_1')$ or $y_2 = \mathsf{f}_\mathsf{e}(x_2')$. We define a VEGOWF $\mathcal{G} = \{\mathsf{g}_\mathsf{e}\}_\mathsf{e}$ with an extraction verification algorithm EV in Figure 6.4 and define the hardness relation:

$$\mathbf{RG}'_\mathsf{e}((y_1, y_2, \pi), (z_1, z_2)) := \mathbf{RG}_\mathsf{e}(y_1, z_1) \vee \mathbf{RG}_\mathsf{e}(y_2, z_2).$$

Similar techniques have been used before in conjunction with EOWFs (e.g, 3-round ZK in[Bit+17]) but not, up to our knowledge, as a generic transformation.

Theorem 6.4.2. *If $\mathcal{F}$ is a GEOWF with respect to $\mathbf{RG}$, then $\mathcal{G}$ in Figure 6.4 is a VEGOWF with respect to $\mathbf{RG}'$.*

Proof. **Verifiable-extractability:** Suppose an adversary $\mathcal{A}$ outputs $y = (y_1, y_2, \pi)$ such that $\mathsf{EV}(\mathsf{e}; (y_1, y_2, \pi)) = 1$. This means that the verifier V accepts the proof, and by perfect soundness this shows that $y_1 \in \mathrm{im}(\mathsf{f}_\mathsf{e})$ or $y_2 \in \mathrm{im}(\mathsf{f}_\mathsf{e})$. Based on $\mathcal{A}$, we can create adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$, which output y_1 and y_2, respectively. WLOG, assume that $y_1 \in \mathrm{im}(\mathsf{f}_\mathsf{e})$. By the definition of GEOWFs, there exists an extractor $\mathsf{Ext}_{\mathcal{A}_1}$ which is able to extract z_1 such that $\mathbf{RG}_\mathsf{e}(y_1, z_1)$ holds, and therefore we are able to extract $z = (z_1, z_2)$ such that $\mathbf{RG}'_\mathsf{e}(y, z)$ holds, which establishes verifiable-extractability.

$\mathbf{RG}'$-hardness: Suppose there exists an adversary $\mathcal{A}$ against $\mathbf{RG}'$-hardness. Therefore, given $(\mathsf{f}_\mathsf{e}(x_1), \mathsf{f}_\mathsf{e}(x_2), \pi = \mathsf{P}(\mathcal{R}_\mathsf{e}, (\mathsf{f}_\mathsf{e}(x_1), \mathsf{f}_\mathsf{e}(x_2)), (x_1, x_2); r)$ with x, y, r chosen uniformly at random, it outputs (z_1, z_2) such that, with noticeable probability $\varepsilon(\lambda)$, $\mathbf{RG}'_\mathsf{e}((\mathsf{f}_\mathsf{e}(x_1), \mathsf{f}_\mathsf{e}(x_2), \pi), (z_1, z_2))$ holds.

We will now use $\mathcal{A}$ to break $\mathbf{RG}$-hardness of f_e. Suppose we are given as input $\mathsf{f}_\mathsf{e}(x)$ for a randomly chosen x, and we wish to find z such that $\mathbf{RG}_\mathsf{e}(\mathsf{f}_\mathsf{e}(x), z) = 1$. To do this, we choose x' and r at random, and run $\mathcal{A}$ on one of the two inputs, chosen at random

with probability 1/2:

$$(\mathsf{f}_\mathsf{e}(x), \mathsf{f}_\mathsf{e}(x'), \mathsf{P}(\mathcal{R}_\mathsf{e}, (\mathsf{f}_\mathsf{e}(x), \mathsf{f}_\mathsf{e}(x')), (\bot, x'); r),$$
$$(\mathsf{f}_\mathsf{e}(x'), \mathsf{f}_\mathsf{e}(x), \mathsf{P}(\mathcal{R}_\mathsf{e}, (\mathsf{f}_\mathsf{e}(x'), \mathsf{f}_\mathsf{e}(x)), (x', \bot); r).$$

Here $\bot$ is some arbitrary element of the input space. By the witness-indistinguishability of Π, these two inputs are indistinguishable from honestly generated values $\mathsf{g}_\mathsf{e}(x, x', r)$ and $\mathsf{g}_\mathsf{e}(x', x, r)$, respectively, and these two values are equally distributed. Therefore, $\mathcal{A}$ will output z such that $\mathbf{RG}(f(x), z)$ with probability $\varepsilon(\lambda)/2 - \mathsf{negl}(\lambda)$, which is a noticeable probability, and therefore contradicts the assumption that $\mathcal{F}$ is $\mathbf{RG}$-hard. □

A robust combiner. Additionally, a simple robust combiner is possible for VEGOWFs (or even for GEOWFs). Let us suppose that $\mathcal{G} = \{\mathsf{g}_{\mathsf{e}_1}\}_{\mathsf{e}_1}$, $\mathcal{F} = \{\mathsf{f}_{\mathsf{e}_2}\}_{\mathsf{e}_2}$, and $\mathcal{H} = \{\mathsf{h}_{\mathsf{e}_3}\}_{\mathsf{e}_3}$ are candidate VEGOWFs for the respective relations $\mathbf{RG}^\mathsf{g}$, $\mathbf{RG}^\mathsf{f}$, and $\mathbf{RG}^\mathsf{h}$. We do assume that the associated extraction verification algorithm always accepts when given a value in the image of each candidate, but we make no other assumption about the security of the candidates.

We define a new VEGOWF $\mathcal{T} = \{\mathsf{t}_\mathsf{e}\}_\mathsf{e}$ by $\mathsf{t}_\mathsf{e}(x, y, z) := (\mathsf{g}_{\mathsf{e}_1}(x), \mathsf{f}_{\mathsf{e}_2}(y), \mathsf{h}_{\mathsf{e}_3}(z))$ where $\mathsf{e} = (\mathsf{e}_1, \mathsf{e}_2, \mathsf{e}_3)$ and the relation $\mathbf{RG}_\mathsf{e}$ is

$$\left\{ \begin{array}{l} \Big((a,b,c),(z_1,z_2)\Big) : \Big((a,z_1) \in \mathbf{RG}^\mathsf{g}_{\mathsf{e}_1} \wedge (b,z_2) \in \mathbf{RG}^\mathsf{f}_{\mathsf{e}_2}\Big) \vee \\ \Big((a,z_1) \in \mathbf{RG}^\mathsf{g}_{\mathsf{e}_1} \wedge (c,z_2) \in \mathbf{RG}^\mathsf{h}_{\mathsf{e}_3}\Big) \vee \Big((b,z_1) \in \mathbf{RG}^\mathsf{f}_{\mathsf{e}_2} \wedge (c,z_2) \in \mathbf{RG}^\mathsf{h}_{\mathsf{e}_3}\Big) \end{array} \right\}.$$

We define the new extraction verification algorithm to accept when all individual extraction verification algorithms accept.

If any two of the candidates are hard for their respective relations, then $\mathcal{T}$ is $\mathbf{RG}$-hard. Similarly, if any two are extractable, then $\mathcal{T}$ is $\mathbf{RG}$-extractable. The idea can be generalized to n VEGOWFs for an arbitrary constant n, where it is sufficient that more than $n/2$ are secure. An interesting open question is to construct a robust combiner where fewer functions have to be secure.

The BCPR GEOWF is Verifiably-Extractable

We show that if a delegation scheme DS is CRS-verifiable, then the BCPR GEOWF from Figure 6.2 is verifiably-extractable with respect to $Y_{\mathsf{Ext}} = \mathrm{im}(\mathsf{DS.K}(1^\lambda)) \times \{0,1\}^{\mathfrak{b}(\lambda)+\lambda}$. That is, z contains the code of an adversary and the DS argument, independently of whether or not $y \in \mathrm{im}(\mathsf{g}_\lambda)$.

The proof of the following theorem is very similar to the proof of Theorem 14 from [Bit+16]; we have reproduced it for the sake of completeness.

Theorem 6.4.3. *Let* DS *be a delegation scheme that has publicly verifiable proofs and CRS, and let* $\mathsf{PRG} : \{0,1\}^\lambda \to \{0,1\}^{\mathfrak{b}(\lambda)+\lambda}$ *be a PRG. Let* $\mathcal{G} = \{\mathsf{g}_\lambda\}_{\lambda\in\mathbb{N}}$ *and* $\mathbf{RG}$ *be as in*

Figure 6.2. $\mathcal{G}$ is a VEGOWF for **RG** *with respect to* $Y_{\mathsf{Ext}} = \mathrm{im}(\mathsf{DS.K}(1^\lambda)) \times \{0,1\}^{\mathfrak{b}(\lambda)+\lambda}$ *and* $(\mathfrak{b}(\lambda) - \omega(1))$*-bounded* aux*.*

Proof. **RG-hardness.** Identical to the proof of Theorem 14 in [Bit+16].

RG-verifiable-extractability. Since DS is CRS-verifiable, there exists an algorithm CV which decides if $\mathsf{crs}_{\mathsf{DS}} \in \mathrm{im}(\mathsf{DS.K}(1^\lambda))$. On input $y = (\mathsf{crs}_{\mathsf{DS}}, v)$, the new extraction verification algorithm EV returns 1 if $\mathsf{CV}(\mathsf{crs}_{\mathsf{DS}}) = 1$ and $|v| = \mathfrak{b}(\lambda) + \lambda$.

We show that there is one universal PPT extractor Ext that can handle any PPT adversary $\mathcal{A}$ with advice of size at most $\mathfrak{b}(\lambda) - \omega(1)$. For an adversary $\mathcal{A}$ (a Turing machine) and advice $\mathsf{aux} \in \{0,1\}^{\mathfrak{b}(\lambda)-\omega(1)}$, denote by $\mathcal{A}_{\mathsf{aux}}$ the machine that, on input 1^λ, runs $\mathcal{A}(1^\lambda; \mathsf{aux})$. Assume that (i) $\mathcal{A}_{\mathsf{aux}}$ has description size at most $\mathfrak{b}(\lambda)$ and that (i) on input 1^λ, after at most $t_{\mathcal{A}} < T(\lambda)$ steps, it outputs $\mathcal{A}_{\mathsf{aux}}(1^\lambda) := y = (\mathsf{crs}_{\mathsf{DS}}, v) \in \{0,1\}^{\mathtt{lpar}'(\lambda)}$. (Recall $Y_{\mathsf{Ext}} \subseteq \{0,1\}^{\mathtt{lpar}'(\lambda)}$.) The extractor $\mathsf{Ext}(\mathcal{A}, \mathsf{aux}, 1^{t_{\mathcal{A}}})$ works as follows:

$\mathsf{Ext}(\mathcal{A}, \mathsf{aux}, 1^{t_{\mathcal{A}}})$

Construct $\mathcal{A}_{\mathsf{aux}}$;
$(\mathsf{crs}_{\mathsf{DS}}, v) \leftarrow \mathcal{A}_{\mathsf{aux}}(1^\lambda)$; **if** $\mathsf{EV}((\mathsf{crs}_{\mathsf{DS}}, v)) = 0$ **then return** $\perp$; **fi** ;
Compute a DS-argument π_{DS} for the fact that "$\mathcal{A}_{\mathsf{aux}}(1^\lambda) = (\mathsf{crs}_{\mathsf{DS}}, v)$";
return $z \leftarrow (\mathcal{A}_{\mathsf{aux}}, \pi_{\mathsf{DS}}, \mathsf{pad})$;

It follows directly from the perfect completeness of DS that $\mathbf{RG}(y, z) = 1$. Since this holds for any $(\mathsf{crs}_{\mathsf{DS}}, v) \in Y_{\mathsf{Ext}}$ output by an adversary with $(\mathfrak{b}(\lambda) - \omega(1))$-bounded auxiliary input, we get **RG**-verifiable-extractability. By the relative prover efficiency of the delegation scheme, the extractor's running time is polynomial in the running time $t_{\mathcal{A}}$ of the adversary. □

To instantiate the construction, we need a delegation scheme with public CRS and proof verification. Firstly, SNARKs in [Mic00; Abd+17; Fuc18] satisfy both properties and have succinct proofs. All of them are based on non-falsifiable assumptions, however, here it is only needed that they are sound for the class P. Thus, even a tautological security assumption (the corresponding SNARK is sound for BPP) would be falsifiable. Such an assumption about [Mic00] was made say in [CLP13]. Secondly, some recent suggestions for delegation schemes [KPY19; Kat+19] with public proof-verification are based on non-tautological falsifiable assumptions. Unfortunately, it is not immediately evident if those schemes also have CRS-verifiability. We leave the latter as an important open problem.

VEGOWFs from Knowledge-of-Exponent Assumptions

Next, we construct VEGOWFs based on knowledge-of-exponent (KE) assumptions, a logical direction partially motivated by [Dak09, Section 3.3.1.1]. In each case, the key is a description p of an asymmetric or symmetric (in the latter case, we state it explicitly)

bilinear group generated by a group generator algorithm $\mathsf{Pgen}(1^\lambda)$. Note that if the group generator Pgen is deterministic, i.e., each security parameter corresponds to a unique group, this is a keyless EOWF.

The ABLZ VEOWF from BDH-KE. The ABLZ VEOWF is based on an idea from Abdolmaleki *et al.* [Abd+17]. We define $\mathsf{g}_\mathsf{p}(x) := ([x]_1, [x]_2)$. The one-way property of the ABLZ EOWF is equivalent to the Symmetric Discrete Logarithm (SDL) [Bic+10] assumption, and extractability is equivalent to the BDH-KE assumption introduced in [Abd+17]. Finally, one can verify if $([x]_1, [y]_2) \in \mathrm{im}(\mathsf{g}_\mathsf{p})$ by checking that $[x]_1 \bullet [1]_1 = [1]_1 \bullet [y]_2$. We give a formal proof that this is a VEOWF in Section 6.C. Note that this VEOWF is injective.

VEGOWF from DH-KE. Some KE assumptions from the literature lead to VEG-OWFs rather than VEOWFs. The Diffie-Hellman KE (DH-KE) assumption introduced in [BFS16] states that any adversary which produces a DDH triple $[x, y, xy]_1$ must know at least one of x and y. Given a symmetric bilinear group, this gives rise to the following VEGOWF. Define $\mathsf{g}_\mathsf{p}(x, y) := [x, y, xy]_1$ and the relation $\mathbf{RG}_\mathsf{p}([x, y, xy]_1, z) = 1$ iff $z = x$ or $z = y$. We can verify if $[x, y, w]_1 \in \mathrm{im}(\mathsf{g}_\mathsf{p})$ by checking that $[x]_1 \bullet [y]_1 = [w]_1 \bullet [1]_1$. This function is **RG**-hard if the discrete logarithm problem is hard and is verifiably-extractable if the DH-KE assumption holds.

Further examples. There are also a number of other knowledge of exponent assumptions in the literature, and these give rise to the following verifiably-extractable injective OWFs:

- $\mathsf{g}_{(\mathsf{p},[1,\alpha]_1)}(x) := [x, x\alpha]_1$ is a OWF under the discrete logarithm assumption and verifiably-extractable for symmetric pairings under the knowledge-of-exponent assumption [Dam92].
- $\mathsf{g}_\mathsf{p}(x) = ([1, x, \ldots, x^q]_1, [1, x, \ldots, x^q]_2)$ is a OWF under the q-PDL assumption [Lip12] and verifiably-extractable under the q-PKE assumption [Dan+14].
- $\mathsf{g}_\mathsf{p}(x) = ([x, x^2]_1, [x]_2)$ is a OWF under a well-known variant of the discrete logarithm assumption and verifiably-extractable under the square knowledge of exponent assumption [Fuc18].
- $\mathsf{g}_\mathsf{p}(x) = ([x]_1, [1/x]_2)$ is a OWF under the inverse-exponent assumption [SS01] and verifiably-extractable under the tautological assumption, which we call *inverse-KE*, that it is hard to compute $[x]_1$, $[1/x]_2$ without knowing x.

VEGOWFs from Knowledge-Sound NIZK

Dakdouk [Dak09, Section 3.3.3.2] observed that EOWFs can be constructed from the proof of knowledge (PoK) assumption of Lepinski [Lep02] which states that a specific

non-interactive Σ-protocol described in [Lep02] is secure. We generalize this idea, and show how to use knowledge-sound NIZKs to build VEGOWFs.

Suppose that $\mathcal{R}$ is an NP relation with a sampler $\mathcal{S}_{\mathcal{R},\mathsf{p}}$ that outputs $(\mathtt{x},\mathtt{w})$, such that (i) it is efficient to verify that $(\mathtt{x},\mathtt{w})$ is a possible output of $\mathcal{S}_{\mathcal{R},\mathsf{p}}$, and (ii) with an overwhelming probability it is computationally hard to guess $\mathtt{w}$ given $\mathtt{x}$. Then we say that this relation is $\mathcal{S}_{\mathcal{R},\mathsf{p}}$-hard. Such samplers (and relations) are common in cryptography, e.g., the discrete logarithm problem ($\mathtt{x} = [x]_1, \mathtt{w} = x$ for a uniformly random x) and the short integer solution problem ($\mathtt{x} = A$ is a random matrix and $\mathtt{w} = \boldsymbol{x}$ is a short integer vector such that $A\boldsymbol{x} = 0$).

Consider a knowledge-sound NIZK $\Pi = (\mathsf{Kgen}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$ for a $\mathcal{S}_{\mathcal{R},\mathsf{p}}$-hard relation $\mathcal{R}$, where $\mathsf{P}, \mathsf{V}, \mathsf{Sim}$ are the prover, the verifier, and the simulator. Kgen is the "key" generation algorithm, such that $\mathsf{Kgen}(\mathcal{R},\mathsf{p})$ produces an auxiliary input aux_Π, provided to P, V and Sim. If the NIZK uses a random oracle, then aux_Π may contain the description of a hash function instantiating the random oracle. If the NIZK is CRS-based, then aux_Π contains the CRS. The following theorem shows how to construct a VEGOWF given a knowledge-sound NIZK.

Theorem 6.4.4. *Define* $\mathcal{G} := \{\mathsf{g}_{\mathcal{R},\mathsf{p},\mathsf{aux}_\Pi}\}_{\mathcal{R}\in\mathsf{RGen}(1^\lambda),\mathsf{p}\leftarrow\mathsf{Pgen}(1^\lambda),\mathsf{aux}_\Pi\in\mathsf{Kgen}(\mathcal{R},\mathsf{p})}$*, where* $\mathsf{g}_{\mathcal{R},\mathsf{p},\mathsf{aux}_\Pi}(r_\mathcal{S}, r_\Pi)$ *sets* $(\mathtt{x},\mathtt{w}) \leftarrow \mathcal{S}_\mathcal{R}(r_\mathcal{S})$*,* π *produced by* Π*'s prover* P *for* $\mathtt{x},\mathtt{w}$*, and then outputs* $(\mathtt{x},\pi)$*. Define the corresponding relation as* $\mathbf{RG}_{\mathcal{R},\mathsf{p},\mathsf{aux}_\Pi} :=$

$$\{(\widehat{y},\widehat{z}) : \widehat{y} = (\mathtt{x},\pi) \wedge \widehat{z} = \mathtt{w} \wedge \Pi.\mathsf{V} \text{ accepts } \pi \wedge (\mathtt{x},\mathtt{w}) \in \mathcal{R}\}\,. \tag{6.1}$$

If $\mathcal{R}$ *is* $\mathcal{S}_\mathcal{R}$*-hard and* Π *is zero-knowledge, then* $\mathcal{G}$ *is* $\mathbf{RG}$*-hard. If* Π *is a proof of knowledge, then* $\mathcal{G}$ *is* $\mathbf{RG}$*-verifiably-extractable.*

Proof. **RG-hardness:** Let $\mathcal{B}$ be an adversary that given $\widehat{y} = (\mathtt{x},\pi)$, where π is a proof for $(\mathtt{x},\mathtt{w})$ returns $\widehat{z}$, such that $\mathbf{RG}_{\mathcal{R},\mathsf{p},\mathsf{aux}_\Pi}(\widehat{y},\widehat{z})$ holds with non-negligible probability. We construct an adversary $\mathcal{B}$ that breaks $\mathcal{S}_\mathcal{R}$-hardness. On input $(\mathcal{R},\mathtt{x})$, $\mathcal{B}$ sets $\mathsf{aux}_\Pi \leftarrow \mathsf{Kgen}(\mathcal{R},\mathsf{p})$, runs the simulator Sim and gets a simulated proof π_{Sim}. Since Π is zero-knowledge, $\mathcal{B}$ outputs the same $\widehat{z} = \mathtt{w}$ (with overwhelming probability) when run on $\widehat{y} = (\mathtt{x},\pi)$ and $\widehat{y} = (\mathtt{x},\pi_{\mathsf{Sim}})$. Thus, $\mathcal{B}$ breaks the $\mathcal{S}_{\mathcal{R},\mathsf{p}}$-hardness of $\mathcal{R}$ with non-negligible probability.

RG-verifiable-extractability: Clearly, one can verify that $\widehat{y} \in \mathrm{im}(\mathsf{g}_{\mathcal{R},\mathsf{p},\mathsf{aux}_\Pi})$ by checking that the NIZK verifier accepts $\widehat{y} = (\mathtt{x},\pi)$, i.e., Π's verifier accepts. We use the knowledge-soundness extractor Ext from Π to build a $\mathcal{G}$ extractor $\mathsf{Ext}_\mathcal{G}$. Let $\mathcal{A}_{\mathsf{ext}}$ be an algorithm that on input $(\mathcal{R},\mathsf{p},\mathsf{aux}_\Pi)$ outputs $\widehat{y} \in \mathrm{im}(\mathsf{g}_{\mathcal{R},\mathsf{p},\mathsf{aux}_\Pi})$. Since $\widehat{y} \in \mathrm{im}(\mathsf{g}_{\mathcal{R},\mathsf{p},\mathsf{aux}_\Pi})$, then $\widehat{y} = (\mathtt{x},\pi)$ and Π's verifier accepts. $\mathsf{Ext}_\mathcal{G}$ runs Ext on the same input $(\mathcal{R},\mathsf{p},\mathsf{aux}_\Pi)$ given to $\mathcal{A}_{\mathsf{ext}}$. By knowledge-soundness, with an overwhelming probability, the Π-extractor Ext outputs $\mathtt{w}$, such that $(\mathtt{x},\mathtt{w}) \in \mathcal{R}$. $\mathsf{Ext}_\mathcal{G}$ sets $\widehat{z} \leftarrow \mathtt{w}$, and succeeds with the same probability as Ext. □

For the sake of concreteness we instantiate the above result as follows. Let Σ be the non-interactive version (e.g., by using the Fiat-Shamir transform) of the well-known Schnorr's protocol for proving the knowledge of the discrete logarithm of $\mathsf{x} = [x]_1$. Let the VEGOWF key be $\mathsf{e} = (\mathcal{R}, \mathsf{p}, \mathsf{aux}_\Pi = H)$, where p is the system parameters (group description). Define $\mathsf{g}_\mathsf{e}(x, r) := ([x]_1, a = [r]_1, z = cx + r) = \widehat{y}$, where $c = H([x]_1, [r]_1)$. The verifier recomputes c and accepts if $[z]_1 = c\mathsf{x} + a$ and $c = H(\mathsf{x}, a)$. Then $\mathbf{RG}_\mathsf{e}$-verifiable-extractability holds since Σ is knowledge-sound in the random oracle model and the algebraic group model [FPS20]. If Σ is zero-knowledge in the random oracle model and the discrete logarithm problem is hard, g_e is also $\mathbf{RG}_\mathsf{e}$-hard. Moreover, Σ can be used to get an injective VEOWF since after the extractor extracts the witness x, it can also compute $r \leftarrow z - cx$.

VEGOWFs from Signature Schemes

We propose the following heuristic approach for finding new candidates for VEGOWFs. Suppose that $\Sigma = (\mathsf{Kgen}, \mathsf{Sign}, \mathsf{Vf})$ is a digital signature scheme. If an adversary outputs (vk, σ) such that $\mathsf{vk} \in \mathsf{Kgen}$ and $\mathsf{Vf}(\mathsf{vk}, \sigma, m = 0) = 1$, then there exists an extractor that can recover (some part of) sk. In other words, we follow the intuition that if someone can sign a message (say $m = 0$ for simplicity), then she must possess the secret key. Moreover, if $\mathsf{vk} \in \mathsf{Kgen}$ can be efficiently verified, then we might be able obtain a VEGOWF.

Remark. Note that unforgeability of a signature scheme does not require that the signer *knows* the secret key. It is only important that the adversary cannot produce valid signatures for previously unsigned messages. A stronger notion of knowledge has been formalized by signatures of knowledge [CL06], where the signer can sign messages under any statement $\mathsf{x} \in \mathcal{L}$ if it knows the corresponding witness. In general this is a very strong notion and implies, e.g., simulation-extractable NIZKs. Therefore, we will not focus on those constructions here.

There are signature schemes which do give believable VEGOWF candidates, but there are also cases where it clearly fails.

Negative example: RSA signatures. Let H be a hash function, $\mathsf{sk} = d$ be the secret key and $\mathsf{vk} = (n, e)$ be a public key such that $de \equiv 1 \pmod{n}$. A signature of an integer m is then $\sigma = H(m)^d \mod n$, and a signature σ of a message m is valid if $\sigma^e \equiv H(m) \pmod{n}$. However, RSA signatures are also not good candidates for a VEGOWF. The adversary could easily compute $\mathsf{vk} = (n, 3)$ such that $H(0) \mod n$ is a perfect cube, then output $(\mathsf{vk}, (H(0) \mod n)^{1/3})$.

Positive example: Boneh-Boyen signatures. Boneh-Boyen [BB04] is a pairing-based signature scheme where $\mathsf{vk} = [x]_2$ and $\mathsf{sk} = x \leftarrow\!\!\$\, \mathbb{Z}_p$ and $\mathsf{Sign}(\mathsf{sk}, m) = [1/(x + m)]_1$. In fact, $\mathsf{g}_\mathsf{p}(x) = (\mathsf{vk}, \mathsf{Sign}(0)) = ([x]_2, [1/x]_1)$ is an asymmetric version of a VEOWF

candidate mentioned in Section 6.4. In particular, it is verifiably-extractable under a similar tautological assumption.

Positive example: BLS signatures. BLS [BLS04] is another pairing-based signature scheme where $\mathsf{vk} = [x]_2$, $\mathsf{sk} = x \leftarrow\!\!\$\, \mathbb{Z}_p$, and $\mathsf{Sign}(\mathsf{sk}, m) = xH(m) = [\sigma]_1$ where H hashes into $\mathbb{G}_1$. Verification is done by checking that $[\sigma]_1[1]_2 = H(m)[x]_2$. This gives us a VEOWF candidate $\mathsf{g}_\mathsf{p}(x) = ([x]_2, xH(0))$.

Positive example: DSA. In the DSA signature scheme,[6] we again have some discrete logarithm secure group $\mathsf{p} = (\mathbb{G}, p, g)$. The verification key is $\mathsf{vk} = g^x$ for $\mathsf{sk} = x \leftarrow\!\!\$\, \mathbb{Z}_p$, $\sigma = \mathsf{Sign}(\mathsf{sk}, M \in \{0,1\}^*; r) = (u = g^r \bmod p, v = r^{-1}(H_K(m) + xu) \bmod p)$, and the verifier checks that $0 < u, v < p$ and $u = (g^{H_K(M)}\mathsf{vk}^u)^{v^{-1}} \bmod p$. DSA results in a candidate VEOWF $\mathsf{g}_{\mathsf{p},K}(x, r) = (g^x, g^r \bmod p, r^{-1}(H_K(m) + xu) \bmod p)$.

Hash-and-sign lattice signatures. We recall hash-and-sign lattice-based signatures introduced by Gentry et al. [GPV08], which relies on the hardness of the short integer solution problem. Let p be a prime, H be a hash function, and let $A \in \mathbb{Z}_p^{m \times n}$ be a randomly generated matrix. Define $L_p^\perp(A) := \{y | Ay = 0 \mod p\}$, and let T be a basis of $L_p^\perp(A)$ with short vectors. The trapdoor can be used to compute short vectors s s.t. $As = b$, for any vector b. Set $\mathsf{vk} = A$ and $\mathsf{sk} = T$.

To sign a message m, one first computes $b = H(m)$, then outputs a short $s = \sigma_A(b)$ such that $As = b$. A signature σ of a message m is valid if it is short and if $A\sigma = H(m)$. However, this does not work as a VEGOWF. The adversary could easily compute s with a nice structure (e.g., a unit vector), then choose A such that $As = H(\mathbf{0})$. An easy fix is to set $b = H(A, m)$ to prevent choosing A after setting s. This results in a candidate VEOWF $\mathsf{g}_p(x) = (A, \sigma_A(H(A, \mathbf{0})))$, where x is a short basis of $L_p^\perp(A)$.

Negative example: Lamport's one-time signature. We briefly remind the idea of Lamport's signature scheme [Lam79]. Let $f : X \to Y$ be a one-way function and suppose we want to sign n-bit messages. The secret key sk is a $2 \times n$ matrix where $\mathsf{sk}_{b,i} \leftarrow\!\!\$\, X$ and the public key vk is also a $2 \times n$ matrix where $\mathsf{vk}_{b,i} = f(\mathsf{sk}_{b,i})$. In order to sign a message $m \in \{0,1\}^n$, signer reveals $\mathsf{sk}_{m_i+1,i}$ for $i = 1, \ldots, n$. That is, the signer reveals half of the secret key that corresponds to the bit-representation of m.

However, this one does not seem to be a good candidate for a VEGOWF. An adversary could easily compute vk that contains a valid preimage for secret keys that correspond to $\mathsf{Sign}(0)$ and rest of the vk is computed obliviously, for example, by hashing into a group in case $f(x) = g^x$.

Schnorr's signature. We recall the signature scheme from [Sch90]. Let $H_K : \{0,1\}^* \to \mathbb{Z}_p$ be a collision-resistant hash function with a key K and let $\mathsf{p} = (\mathbb{G}, p, g)$ be a description of a prime p order group with a generator g. We assume that the discrete logarithm assumption holds in $\mathbb{G}$. The verification key $\mathsf{vk} = g^x$ where $\mathsf{sk} = x \leftarrow\!\!\$\, \mathbb{Z}_p$ and $\mathsf{Sign}(\mathsf{sk}, M \in$

[6] `https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf`

$\{0,1\}^*) = (g^y, z = ex + y) = \sigma$ where $e = H_K(g^y, M)$ and $y \leftarrow\!\$\, \mathbb{Z}_p$. Verification is done by checking that $g^\sigma = (g^x)^e \cdot g^y$. The candidate VEOWF we obtain is $\mathsf{g}_{\mathsf{p},K}(x,y) = (g^x, g^y, H_K(g^y, 0)x + y)$.

Due to a well-known weakness of the Fiat-Shamir paradigm, $\mathsf{g}_{\mathsf{p},K}(x,y)$ is insecure. The adversary can sample $g^y \leftarrow\!\$\, \mathbb{G}$ without necessarily knowing y, sample $z \leftarrow \mathbb{Z}_p$, and then compute $g^x = (g^z/g^y)^{1/(H_K(g^y,0))}$. Hence the verification holds by definition but the adversary may not know x, y. We can easily fix this by instead computing $H_K(g^x, g^y, 0)$, which will give the VEOWF mentioned in Section 6.4.

6.5 Sub-ZK NIZKs Based on VEGOWFs

We give a generic construction of a knowledge-sound Sub-ZK NIZK from any VEGOWF and any knowledge-sound Sub-WI NIWI in the CRS model. We also give a generic construction of a sound Sub-ZK NIZK from any VEGOWF, any keyless extractable commitment and any Sub-WI NIWI in the CRS model. Later, we show some interesting instantiations of these constructions.

Constructing Knowledge-sound Sub-ZK NIZK

Let $\mathcal{G} = \{\mathsf{g}_\lambda : X_\lambda \to Y_\lambda \mid \lambda \in \mathbb{N}\}$ be a keyless VEGOWF with respect to a publicly testable relation $\mathbf{RG}$ on triples $(1^\lambda, \widehat{y}, \widehat{z})$. We construct a knowledge-sound Sub-ZK NIZK Π by using a knowledge-sound Sub-WI NIWI Π_{wi} and $\mathcal{G}$. To prove that $\mathtt{x} \in \mathcal{L}$, we use Π_{wi} to prove that $(\mathtt{x}, \widehat{y}) \in \mathcal{L}'$, where $\widehat{y} \in Y_{\mathsf{Ext}}$ is a new element in the CRS for Π, and

$$\mathcal{R}' := \{(\mathtt{x}_{\mathcal{R}'} = (\mathtt{x}, \widehat{y}), \mathtt{w}_{\mathcal{R}'} = (\mathtt{w}, \widehat{z})) : (\mathtt{x}, \mathtt{w}) \in \mathcal{R} \vee (\widehat{y}, \widehat{z}) \in \mathbf{RG}\}$$

where $\mathcal{L} = \{\mathtt{x} \mid \exists \mathtt{w} : (\mathtt{x}, \mathtt{w}) \in \mathcal{R}\}$ and $\mathcal{L}' = \{\mathtt{x}_{\mathcal{R}'} \mid \exists \mathtt{w}_{\mathcal{R}'} : (\mathtt{x}_{\mathcal{R}'}, \mathtt{w}_{\mathcal{R}'}) \in \mathcal{R}'\}$. We assume that $\mathcal{R}$ is generated by a relation generator $\mathsf{RGen}(1^\lambda)$. The full construction of Π can be found in Figure 6.5.

The construction yields a knowledge-sound Sub-ZK NIZK, where knowledge-soundness follows from the $\mathbf{RG}$-hardness of $\mathcal{G}$ and the knowledge-soundness of Π_{wi}, and subversion zero-knowledge is achieved by the $\mathbf{RG}$-verifiable-extractability of $\mathcal{G}$ as well as the subversion witness-indistinguishability of Π_{wi}.

Note that if $\mathcal{R}$ is implemented by a circuit of size k and $\mathbf{RG}$ is implemented by a circuit of size l, then the efficiency of Π is the same as the efficiency of Π' for the modified circuit of size $k + l$. Note also that l is independent of $\mathcal{R}$.

Theorem 6.5.1 (Knowledge-sound Sub-WI NIWI + VEGOWF $\implies$ Knowledge-sound Sub-ZK NIZK)**.** *Let Π_{wi} be a non-interactive argument for $\mathcal{R}'$ and let $\mathcal{G} = \{\mathsf{g}_\lambda\}_{\lambda\in\mathbb{N}}$ be a keyless function family with a corresponding publicly testable relation $\mathbf{RG}$.*
(1) If Π_{wi} is complete then Π is complete.

$\mathsf{K}(\mathcal{R})$

$\widehat{x} \leftarrow\!\!\$\ X_\lambda$;
$\widehat{y} \leftarrow \mathsf{g}_\lambda(\widehat{x})$
$\mathsf{crs}' \leftarrow \mathsf{K}'(\mathcal{R})$
$\mathsf{crs} \leftarrow (\mathsf{crs}', \widehat{y})$
$\mathsf{td} \leftarrow t(\widehat{x})$
return $(\mathsf{crs}, \mathsf{td})$

$\mathsf{CV}(\mathcal{R}, \mathsf{crs})$

parse $\mathsf{crs} = (\mathsf{crs}', \widehat{y})$;
if $\mathsf{CV}(\mathcal{R}', \mathsf{crs}') = 1 \wedge \widehat{y} \in Y_{\mathsf{Ext}}$
then return 1
else return 0

$\mathsf{Sim}(\mathcal{R}, \mathsf{crs}, \mathtt{x}, \mathsf{td})$

parse $\mathsf{crs} = (\mathsf{crs}', \widehat{y})$;
return $\mathsf{P}'(\mathcal{R}', \mathsf{crs}', (\mathtt{x}, \widehat{y}), (\bot, \mathsf{td}))$

$\mathsf{P}(\mathcal{R}, \mathsf{crs}, \mathtt{x}, \mathtt{w})$

parse $\mathsf{crs} = (\mathsf{crs}', \widehat{y})$
return $\pi \leftarrow \mathsf{P}'(\mathcal{R}', \mathsf{crs}', (\mathtt{x}, \widehat{y}), (\mathtt{w}, \bot))$;

$\mathsf{V}(\mathcal{R}, \mathsf{crs}, \mathtt{x}, \pi)$

parse $\mathsf{crs} = (\mathsf{crs}', \widehat{y})$;
return $\mathsf{V}'(\mathcal{R}', \mathsf{crs}', (\mathtt{x}, \widehat{y}), \pi)$

Figure 6.5: The Sub-ZK KS NIZK $\Pi = (\mathsf{K}, \mathsf{CV}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$, where $\Pi_{wi} = (\mathsf{K}', \mathsf{CV}', \mathsf{P}', \mathsf{V}')$ is a Sub-WI KS argument, and $\mathcal{G} = \{\mathsf{g}_\lambda\}_{\lambda \in \mathbb{N}}$ is a VEGOWF. Recall that t computes the "witness" for $\mathsf{g}_\lambda(\widehat{x})$ in **RG**.

$\mathcal{B}(\mathcal{R}', \mathsf{crs}'; r)$

$\widehat{x} \leftarrow\!\!\$\ X_\lambda; \widehat{y} \leftarrow \mathsf{g}_\lambda(\widehat{x})$;
$\mathsf{crs} \leftarrow (\mathsf{crs}', \widehat{y})$;
$(\mathtt{x}, \pi) \leftarrow \mathcal{A}(\mathcal{R}, \mathsf{crs}; r)$;
return $((\mathtt{x}, \widehat{y}), \pi)$;

$\mathcal{C}(\mathcal{R}', \widehat{y}, \mathsf{aux}; r)$

$\mathsf{crs}' \leftarrow \mathsf{K}'(\mathcal{R}')$;
$(\mathtt{w}, \widehat{z}) \leftarrow \mathsf{Ext}^{wi}_{\mathcal{B}}(\mathcal{R}', \mathsf{crs}'; r)$;
return $\widehat{z}$;

Figure 6.6: $\mathcal{B}$, $\mathcal{C}$ in the knowledge-soundness proof of Theorem 6.5.1.

(2) If Π_{wi} is knowledge-sound for $\mathcal{R}'$ and $\mathcal{G}$ is **RG***-hard then Π is knowledge-sound for $\mathcal{R}$.*

(3) If Π_{wi} is Sub-WI for $\mathcal{R}'$ and $\mathcal{G}$ is **RG***-verifiably-extractable, then Π is Sub-ZK for $\mathcal{R}$.*

(4) If Π_{wi} is a Sub-WI SNARK and $\mathcal{G}$ is a VEGOWF with respect to a relation **RG** *which takes inputs of polynomial size, then Π is a Sub-ZK SNARK.*

Proof. **Completeness:** Straightforward.

Knowledge Soundness: Since Π_{wi} is knowledge-sound, for every Π_{wi}-adversary $\mathcal{B}$ there exists a knowledge-soundness extractor $\mathsf{Ext}^{wi}_{\mathcal{B}}$ such that if $\mathcal{B}(\mathcal{R}', \mathsf{crs}'; r)$ returns an acceptable instance–proof pair $(\mathtt{x}_{\mathcal{R}'} = (\mathtt{x}, \widehat{y}), \pi)$ then $\mathsf{Ext}^{wi}_{\mathcal{B}}(\mathcal{R}', \mathsf{crs}'; r)$ returns $\mathtt{w}_{\mathcal{R}'} = (\mathtt{w}, \widehat{z})$ which satisfies $(\mathtt{x}_{\mathcal{R}'}, \mathtt{w}_{\mathcal{R}'}) \in \mathcal{R}'$ with all but negligible probability ε_{ksnd}.

Since $\mathcal{G}$ is **RG**-hard, for every $\mathcal{G}$-adversary $\mathcal{C}$, the probability that for random $\widehat{x}$, $\mathcal{C}(1^\lambda, \widehat{y} = \mathsf{g}_\lambda(\widehat{x}), \mathsf{aux}) = \widehat{z}$ such that $(\widehat{y}, \widehat{z}) \in \mathbf{RG}$ is bounded by some negligible ε_{hard}.

Let $\mathcal{A}$ be an adversary for the knowledge-soundness of Π that succeeds with probability $\varepsilon_{\mathcal{A}}$. That is, for $r \leftarrow\!\!\$\ \mathsf{RND}_\lambda(\mathcal{A})$, $\mathcal{A}(\mathcal{R}, \mathsf{crs}; r)$ returns with probability $\varepsilon_{\mathcal{A}}$ an instance $\mathtt{x}$ and proof π such that the Π-verifier V accepts but no extractor equipped with the code and randomness of $\mathcal{A}$ returns $\mathtt{w}$ such that $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$.

We construct the two adversaries $\mathcal{B}$ (against the knowledge-soundness of Π_{wi}) and $\mathcal{C}$ (against the **RG**-hardness of $\mathcal{G}$), see Figure 6.6. We denote by bad the event that $\mathsf{Ext}_{\mathcal{B}}^{wi}$ fails, i.e., for $\mathtt{x}_{\mathcal{R}'}$ output by $\mathcal{B}$ and $\mathtt{w}_{\mathcal{R}'}$ output by $\mathsf{Ext}_{\mathcal{B}}^{wi}$, $(\mathtt{x}_{\mathcal{R}'}, \mathtt{w}_{\mathcal{R}'}) \notin \mathcal{R}'$. Clearly, $\mathcal{B}$ succeeds if (i) $\mathcal{A}$ succeeds (from which it follows that $\mathcal{A}$ returns $(\mathtt{x}, \pi)$ such that V accepts $(\mathtt{x}, \pi)$ and thus V' accepts $(\mathtt{x}_{\mathcal{R}'}, \pi)$) and (ii) bad holds, since then the Π_{wi}-verifier V accepts a statement for which an extractor cannot extract a witness. Thus, $\Pr[\mathcal{B} \text{ succeeds}] \geq \Pr[\mathcal{A} \text{ succeeds} \wedge \mathsf{bad}]$.

On the other hand, $\mathcal{C}$ succeeds if (i) $\mathcal{A}$ succeeds (i.e., $\mathcal{A}$ returns $(\mathtt{x}, \pi)$ such that V accepts $(\mathtt{x}, \pi)$) and for $\mathtt{w}$ returned by any $\mathsf{Ext}_{\mathcal{A}}$, $(\mathtt{x}, \mathtt{w}) \notin \mathcal{R}$, and (ii) bad does not hold (i.e., for $\mathtt{w}_{\mathcal{R}'}$ extracted by $\mathsf{Ext}_{\mathcal{B}}^{wi}$, $(\mathtt{x}_{\mathcal{R}'}, \mathtt{w}_{\mathcal{R}'}) \in \mathcal{R}'$, which means that, since $(\mathtt{x}, \mathtt{w}) \notin \mathcal{R}$, $(\widehat{y}, \widehat{z}) \in \mathbf{RG}$), and therefore $\mathcal{C}$ has output $\widehat{z}$ such that $(\widehat{y}, \widehat{z}) \in \mathbf{RG}$. Thus, $\Pr[\mathcal{C} \text{ succeeds}] \geq \Pr\left[\mathcal{A} \text{ succeeds} \wedge \overline{\mathsf{bad}}\right]$. Clearly,

$$\begin{aligned}
\varepsilon_{\mathcal{A}} = \Pr[\mathcal{A} \text{ succeeds}] &= \Pr[\mathcal{A} \text{ succeeds} \wedge \mathsf{bad}] + \Pr\left[\mathcal{A} \text{ succeeds} \wedge \overline{\mathsf{bad}}\right] \\
&\leq \Pr[\mathcal{B} \text{ succeeds}] + \Pr[\mathcal{C} \text{ succeeds}] \\
&\leq \varepsilon_{ksnd} + \varepsilon_{hard} \ .
\end{aligned}$$

Hence any adversary against the knowledge-soundness of Π has a negligible probability of succeeding, so Π is knowledge-sound.

Sub-ZK: Let $\mathcal{Z}$ be any subverter that, on input $(\mathcal{R}'; r_{\mathcal{Z}})$, creates $\mathsf{crs} = (\mathsf{crs}', \widehat{y})$ such that $\mathsf{CV}(\mathcal{R}, \mathsf{crs}) = 1$, along with some auxiliary information $\mathsf{aux}_{\mathcal{Z}}$. We construct the following adversary $\mathcal{B}$:

$$\frac{\mathcal{B}(\mathcal{R}'; r_{\mathcal{Z}})}{(\mathsf{crs}', \widehat{y}), \mathsf{aux}_{\mathcal{Z}} \leftarrow \mathcal{Z}(\mathcal{R}'; r_{\mathcal{Z}}); \mathbf{return}\ \widehat{y};}$$

Since $\mathsf{CV}(\mathcal{R}, \mathsf{crs}) = 1$, we have that $\widehat{y} \in Y_{\mathsf{Ext}}$. Since $\mathcal{G}$ is **RG**-verifiably-extractable, there exists an extractor $\mathsf{Ext}_{\mathcal{B}}$ such that, with overwhelming probability $1-\varepsilon_{ext}$, $\widehat{z} \leftarrow \mathsf{Ext}_{\mathcal{B}}(\mathcal{R}; r_{\mathcal{Z}})$ where $(\widehat{y}, \widehat{z}) \in \mathbf{RG}$.

The extractor $\mathsf{Ext}_{\mathcal{Z}}$ for $\mathcal{Z}$ works as follows: Given $\mathcal{Z}$, construct $\mathcal{B}$ as above, and output the result of $\mathsf{Ext}_{\mathcal{B}}$. Clearly, $\mathsf{Ext}_{\mathcal{Z}}$ succeeds precisely when $\mathsf{Ext}_{\mathcal{B}}$ does. Let bad denote the event where extraction fails, i.e. the $\widehat{z}$ returned by $\mathsf{Ext}_{\mathcal{B}}(\mathcal{R}; r_{\mathcal{Z}})$ does not satisfy $(\widehat{y}, \widehat{z}) \in \mathbf{RG}$. We then have that $\Pr[\mathsf{bad}] = \varepsilon_{ext}$.

Consider an adversary $\mathcal{A}_{sub\text{-}zk}$ against the Sub-ZK property of Π. Based on $\mathcal{Z}$ and $\mathcal{A}_{sub\text{-}zk}$, we construct an adversary $\mathcal{A}_{sub\text{-}wi}$ which succeeds precisely when $\mathcal{A}_{sub\text{-}zk}$ succeeds and extraction is successful.

$\mathcal{A}_{sub\text{-}wi}(\mathcal{R}', r_{\mathcal{Z}})$

$((\mathsf{crs}', \widehat{y}), \mathsf{aux}_{\mathcal{Z}}; \widehat{z}) \leftarrow (\mathcal{Z}; \mathsf{Ext}_{\mathcal{Z}})(\mathcal{R}', r_{\mathcal{Z}})$;

$(\mathtt{x}, \mathtt{w}) \leftarrow \mathcal{A}_{sub\text{-}zk}(\mathcal{R}', (\mathsf{crs}', \widehat{y}), \mathsf{aux}_{\mathcal{Z}})$

$\pi_b \leftarrow \mathsf{O}_{wi}(\mathsf{crs}', (\mathtt{x}, \widehat{y}), (\mathtt{w}, \bot), (\bot, \widehat{z}), \mathsf{aux}_{\mathcal{Z}})$;

return $\mathcal{A}_{sub\text{-}zk}(\pi_b, \mathsf{aux}_{\mathcal{Z}})$;

Here O_{wi} is the challenger of the Sub-WI game that takes a NIWI instance x along with two witnesses w_0, w_1 and returns a proof π_b which uses w_b for a randomly sampled $b \leftarrow\!\!\$\, \{0,1\}$.

$\mathcal{A}_{sub\text{-}wi}$ succeeds in the Sub-WI game when both i) $(\bot, \widehat{z})$ is a valid witness (this happens precisely when $\mathsf{Ext}_{\mathcal{Z}}$ succeeds, i.e. when $\overline{\mathsf{bad}}$ holds) and ii) it distinguishes $(\mathsf{P}'(\mathcal{R}', \mathsf{crs}', (\mathtt{x}, \widehat{y}), (\mathtt{w}, \bot)), \mathsf{aux}_{\mathcal{Z}})$ and $(\mathsf{P}'(\mathcal{R}', \mathsf{crs}', (\mathtt{x}, \widehat{y}), (\bot, \widehat{z})), \mathsf{aux}_{\mathcal{Z}})$. By the definition of P and Sim, these distributions equal $(\mathsf{P}(\mathcal{R}, \mathsf{crs}, \mathtt{x}, \mathtt{w}), \mathsf{aux}_{\mathcal{Z}})$ and $(\mathsf{Sim}(\mathcal{R}, \mathsf{crs}, \widehat{z}, \mathtt{x}), \mathsf{aux}_{\mathcal{Z}})$, respectively. Thus $\mathcal{A}_{sub\text{-}wi}$ succeeds when both $\overline{\mathsf{bad}}$ holds and $\mathcal{A}_{sub\text{-}zk}$ succeeds, so $\Pr[\mathcal{A}_{sub\text{-}wi} \text{ succeeds}] \geq \Pr[\mathcal{A}_{sub\text{-}zk} \text{ succeeds} \wedge \overline{\mathsf{bad}}]$.

Let $\varepsilon_{sub\text{-}wi}$ be the maximal advantage any adversary has in breaking Sub-WI of Π_{wi}. Then

$$\begin{aligned}\Pr[\mathcal{A}_{sub\text{-}zk} \text{ succeeds}] &= \Pr[\mathcal{A}_{sub\text{-}zk} \text{ succeeds} \wedge \mathsf{bad}] + \Pr[\mathcal{A}_{sub\text{-}zk} \text{ succeeds} \wedge \overline{\mathsf{bad}}] \\ &\leq \Pr[\mathsf{bad}] + \Pr[\mathcal{A}_{sub\text{-}wi} \text{ succeeds}] \\ &\leq \varepsilon_{ext} + \varepsilon_{sub\text{-}wi}\end{aligned}$$

Note that while computational (respectively, statistical) Sub-WI implies computational (respectively, statistical) Sub-ZK, perfect Sub-WI implies perfect Sub-ZK only if there is no chance of extraction failure for the VEGOWF, otherwise we get statistical Sub-ZK.

Sub-ZK SNARK: Suppose Π_{wi} is succinct for $\mathcal{R}'$, then there is some polynomial poly and some $c < 1$ such that the size of any proof π_{wi} from Π_{wi} is bounded by $\mathsf{poly}(\lambda) \cdot (|(\mathtt{x}, \widehat{y})| + |(\mathtt{w}, \widehat{z})|)^c = \mathsf{poly}(\lambda) \cdot (|\mathtt{x}| + |\mathtt{w}| + |\widehat{y}| + |\widehat{z}|)^c \leq \mathsf{poly}'' \cdot (\lambda)(|\mathtt{x}| + |\mathtt{w}|)^c$, for some polynomial poly'' since $|\widehat{y}|, |\widehat{z}|$ are bounded by a polynomial. Since any proof in Π' for $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$ is a proof in Π_{wi} for $((\mathtt{x}, \widehat{y}), (\mathtt{w}, \widehat{z})) \in \mathcal{R}'$, this shows that any proof in Π' is succinct. The remaining properties of being a Sub-ZK SNARK have been proved above. □

Constructing Sub-ZK NIZK

Next, we propose a Sub-ZK NIZK Π which only relies on Π_{wi} being sound, not knowledge-sound, but Π will also not be knowledge-sound. As part of this construction, we rely on a keyless extractable commitment scheme. We now give the definition of a keyless extractable commitment scheme, and in Section 6.B we show how this can be constructed based on injective EOWFs.

Definition 6.5.2. We say that $\mathsf{com}_\lambda \colon \mathcal{M}_\lambda \times \mathcal{R}_\lambda \to \mathcal{C}_\lambda$ is a keyless extractable commitment if it satisfies the following properties.

Computational hiding: For any PPT adversary $\mathcal{A}$, $\varepsilon_0 \approx_\lambda \varepsilon_1$, where

$$\varepsilon_b := \Pr\begin{bmatrix}(m_1, m_2) \leftarrow \mathcal{A}(1^\lambda), r \leftarrow\!\!\$\, \mathcal{R}_\lambda, c \leftarrow \mathsf{com}_\lambda(m_b; r) : \\ m_1, m_2 \in \mathcal{M}_\lambda \wedge \mathcal{A}(c) = 1\end{bmatrix} .$$

Perfect binding: For any adversary $\mathcal{A}$ and $\lambda \in \mathbb{N}$,

$$\Pr\begin{bmatrix}(m_1, r_1, m_2, r_2) \leftarrow \mathcal{A}(1^\lambda) : \\ \mathsf{com}_\lambda(m_1; r_1) = \mathsf{com}_\lambda(m_2; r_2) \wedge m_1 \neq m_2\end{bmatrix} = 0 .$$

Non-black-box extractability: Let $\mathcal{D}$ be a family $\{D_\lambda\}_\lambda$ of efficiently sampleable distributions. We say that $\mathsf{com}_\lambda \colon \mathcal{M}_\lambda \times \mathcal{R}_\lambda \to \mathcal{C}_\lambda$ is non-black-box extractable with respect to auxiliary distribution $\mathcal{D}$ if for any PPT adversary $\mathcal{A}$, there exists a PPT extractor $\mathsf{Ext}_\mathcal{A}$ such that,

$$\Pr\begin{bmatrix}\mathsf{aux} \leftarrow\!\!\$\, D_\lambda, c \leftarrow \mathcal{A}(1^\lambda, \mathsf{aux}), m \leftarrow \mathsf{Ext}_\mathcal{A}(1^\lambda, \mathsf{aux}), \\ c \in \mathrm{im}(\mathsf{com}_\lambda) : c = \mathsf{com}_\lambda(m; r) \text{ for some } r \in \mathcal{R}_\lambda;\end{bmatrix} \geq 1 - \mathsf{negl}(\lambda) .$$

In some cases, we may have an efficient commitment verification function ComV_λ that outputs 1 on input c if and only if $c \in \mathrm{im}(\mathsf{com}_\lambda)$.

Let $\mathcal{G} = \{\mathsf{g}_\lambda\}_{\lambda\in\mathbb{N}}$ be a function family with associated relation $\mathbf{RG}$. Let $\mathsf{C} = (\mathsf{com}, \mathsf{Open}, \mathsf{Vf})$ be an extractable commitment scheme. Let Π_{wi} be a NIWI argument for the relation $\mathcal{R}' = \left\{((\mathtt{x}, c, \widehat{y}), (\mathtt{w}, \widehat{z}, \widehat{r})) : \ ((\mathtt{x}, \mathtt{w}) \in \mathcal{R}) \vee (c = \mathsf{com}(\widehat{z}; \widehat{r}) \wedge (\widehat{y}, \widehat{z}) \in \mathbf{RG})\right\}$.

$$\mathcal{R}' = \left\{((\mathtt{x}, c, \widehat{y}), (\mathtt{w}, \widehat{z}, \widehat{r})) : \ ((\mathtt{x}, \mathtt{w}) \in \mathcal{R}) \vee (c = \mathsf{com}(\widehat{z}; \widehat{r}) \wedge (\widehat{y}, \widehat{z}) \in \mathbf{RG})\right\} .$$

We set $\mathsf{crs} = (\mathsf{crs}', \widehat{y})$, where crs' is the CRS of the underlying NIWI Π_{wi} for $\mathcal{R}'$ and crs is the CRS of the NIZK for $\mathcal{R}$. The argument consists of the commitment c and the Π_{wi}-argument π; see Figure 6.7.

Theorem 6.5.3 (Sub-WI NIWI + VEGOWF + ExtCom $\implies$ Sub-ZK NIZK)**.** *Let Π_{wi} be a non-interactive argument, C be a commitment scheme, and $\mathcal{G}$ be a function family with associated publicly testable relation $\mathbf{RG}$.*

(1) If Π_{wi} is perfectly complete then Π is perfectly complete.

(2) If Π_{wi} is sound, C is keyless and extractable, and $\mathcal{G}$ is $\mathbf{RG}$-hard then Π is sound.

(3) If Π_{wi} is Sub-WI, $\mathcal{G}$ is $\mathbf{RG}$-verifiably-extractable, and C is keyless and hiding, then Π is Sub-ZK.

Proof. **Perfect completeness:** obvious.

Soundness: Consider an adversary $\mathcal{A}$ against the soundness of Π, which gets as input a relation $\mathcal{R}$ as well as an honestly computed CRS crs, and outputs (x, π) such that $x \notin \mathcal{L}$ but

$\mathsf{K}(\mathcal{R})$

$\widehat{x} \leftarrow\!\$\ X_\lambda$;
$\widehat{y} \leftarrow \mathsf{g}_\lambda(\widehat{x})$;
$\mathsf{crs}' \leftarrow \mathsf{K}'(\mathcal{R}')$;
$\mathsf{crs} \leftarrow (\mathsf{crs}', \widehat{y})$;
$\mathsf{td} \leftarrow t(\widehat{x})$;
return $(\mathsf{crs}, \mathsf{td})$;

$\mathsf{CV}(\mathcal{R}, \mathsf{crs})$

parse $\mathsf{crs} = (\mathsf{crs}', \widehat{y})$;
if $\mathsf{CV}'(\mathcal{R}', \mathsf{crs}') = 1 \wedge y \in Y_{\mathsf{Ext}}$;
  then return 1
  else return 0

$\mathsf{Sim}(\mathcal{R}, \mathsf{crs}, \mathsf{td} = \widehat{z}, \mathtt{x})$

parse $\mathsf{crs} = (\mathsf{crs}', \widehat{y})$;
$r \leftarrow\!\$\ \mathsf{RND}_\lambda(\mathsf{com})$;
$c \leftarrow \mathsf{com}(\widehat{z}; r)$;
$\pi' \leftarrow \mathsf{P}(\mathcal{R}', \mathsf{crs}', (\mathtt{x}, c, \widehat{y}), (\bot, \widehat{z}, r))$;
return $\pi \leftarrow (c, \pi')$

$\mathsf{P}(\mathcal{R}, \mathsf{crs}, \mathtt{x}, \mathtt{w})$

parse $\mathsf{crs} = (\mathsf{crs}', \widehat{y})$;
$r \leftarrow \mathsf{RND}_\lambda(\mathsf{com})$;
$c \leftarrow \mathsf{com}(x_\lambda; r)$ where $x_\lambda \leftarrow\!\$\ X_\lambda$;
$\pi' \leftarrow \mathsf{P}'(\mathcal{R}', \mathsf{crs}', (\mathtt{x}, c, \widehat{y}), (\mathtt{w}, x_\lambda, r))$;
return $\pi \leftarrow (c, \pi')$;

$\mathsf{V}(\mathcal{R}, \mathsf{crs}, \mathtt{x}, \pi)$

parse $\pi = (c, \pi')$;
parse $\mathsf{crs} = (\mathsf{crs}', \widehat{y})$;
return $\mathsf{V}'(\mathcal{R}', \mathsf{crs}', (\mathtt{x}, c, \widehat{y}), \pi')$;

Figure 6.7: The Sub-ZK NIZK $\Pi = (\mathsf{K}, \mathsf{CV}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$, where $\Pi_{wi} = (\mathsf{K}', \mathsf{CV}', \mathsf{P}', \mathsf{V}')$ is a Sub-WI NIWI, C is an extractable commitment scheme, and $\mathcal{G} = \{\mathsf{g}_\lambda\}_{\lambda \in \mathbb{N}}$ is a GEOWF.

$\mathcal{A}_{snd}(\mathcal{R}', \mathsf{crs}')$

$\widehat{x} \leftarrow\!\$\ X_\lambda; \widehat{y} \leftarrow \mathsf{g}_\lambda(\widehat{x})$;
$\mathsf{crs} \leftarrow (\mathsf{crs}', \widehat{y})$;
$(\mathtt{x}, (c, \pi')) \leftarrow \mathcal{A}(\mathcal{R}, \mathsf{crs})$;
return $((\mathtt{x}, c, \widehat{y}), \pi')$;

$\mathcal{C}(\mathcal{R}, \mathsf{crs})$

$(\mathtt{x}, (c, \pi)) \leftarrow \mathcal{A}(\mathcal{R}, \mathsf{crs})$;
return c;

$\mathcal{A}_{hard}(\mathcal{R}', \widehat{y})$

$\mathsf{crs}' \leftarrow \mathsf{K}(\mathcal{R}')$;
$\mathsf{crs} \leftarrow (\mathsf{crs}', \widehat{y})$;
$\widehat{z} \leftarrow \mathsf{Ext}_{\mathcal{C}}(\mathcal{R}, \mathsf{crs})$;
return $\widehat{z}$;

Figure 6.8: $\mathcal{A}_{snd}$, $\mathcal{C}$, $\mathcal{A}_{hard}$ in the soundness proof of Theorem 6.5.3.

π is accepted by the verifier V. This means that $\mathsf{V}(\mathcal{R}, \mathsf{crs}, x, \pi) = \mathsf{V}'(\mathcal{R}', \mathsf{crs}', (x, c, \widehat{y}), \pi')$ returns 1. We use $\mathcal{A}$ to construct two adversaries, one against the soundness of Π_{wi} and one against the **RG**-hardness of $\mathcal{G}$, see Figure 6.8. Let in denote the event that $(x, c, \widehat{y}) \in \mathcal{L}'$. Let ext denote the event that $\mathsf{Ext}_{\mathcal{C}}$ succeeds in extraction.

Note that, if $\mathcal{A}$ succeeds in breaking the soundness of Π by returning $(\mathtt{x}, (c, \pi'))$ such that $(\mathtt{x}, c, \widehat{y}) \notin \mathcal{L}'$, then $\mathcal{A}_{snd}$ succeeds in breaking the soundness of Π_{wi}. Therefore $\Pr[\mathcal{A} \text{ succeeds} \wedge \overline{\mathsf{in}}] \leq \Pr[\mathcal{A}_{snd} \text{ succeeds}]$.

We also see that, if $\mathcal{A}$ succeeds in breaking the soundness of Π by returning $(\mathtt{x}, (c, \pi'))$ such that $(\mathtt{x}, c, \widehat{y}) \in \mathcal{L}'$ and the extractor $\mathsf{Ext}_{\mathcal{C}}$ is successful, then $\mathcal{A}_{hard}$ succeeds in breaking the **RG**-hardness of $\mathcal{G}$. Hence $\Pr[\mathcal{A} \text{ succeeds} \wedge \mathsf{in} \wedge \mathsf{ext}] \leq \Pr[\mathcal{A}_{hard} \text{ succeeds}]$.

We have that

$$\begin{aligned}\Pr[\mathcal{A} \text{ succeeds}] &\leq \Pr[\mathcal{A} \text{ succeeds} \wedge \overline{\mathsf{in}}] + \Pr[\mathcal{A} \text{ succeeds} \wedge \mathsf{in} \wedge \mathsf{ext}] + \Pr[\overline{\mathsf{ext}}] \\ &\leq \Pr[\mathcal{A}_{snd} \text{ succeeds}] + \Pr[\mathcal{A}_{hard} \text{ succeeds}] + \Pr[\overline{\mathsf{ext}}]\end{aligned}$$

By the soundness of Π, the **RG**-hardness of $\mathcal{G}$ and the extractability of C, respectively, we have that$\Pr[\mathcal{A}_{snd} \text{ succeeds}]$, $\Pr[\mathcal{A}_{hard} \text{ succeeds}]$ and $\Pr[\overline{\mathsf{ext}}]$ are all negligible. Hence $\Pr[\mathcal{A} \text{ succeeds}]$ is negligible so Π is sound.

Sub-ZK: Let $\mathcal{Z}(\mathcal{R}; r_{\mathcal{Z}})$ be a subverter that outputs $\mathsf{crs} = (\mathsf{crs}', \widehat{y})$ accepted by CV, as well as some auxiliary information $\mathsf{aux}_{\mathcal{Z}}$. We first construct a PPT algorithm $\mathcal{A}(\mathcal{R}; r_{\mathcal{Z}})$ which invokes $\mathcal{Z}(\mathcal{R}; r_{\mathcal{Z}})$, obtains crs, and then outputs $\widehat{y}$. Since $\mathsf{CV}(\mathsf{crs}) = 1$, $\mathcal{A}$ outputs a value in Y_{Ext}. Since $\mathcal{G}$ is **RG**-extractable, there exists an extractor $\mathsf{Ext}_{\mathcal{A}}(\mathcal{R}; r_{\mathcal{Z}})$ that extracts $\widehat{z}$ such that $\mathbf{RG}(\widehat{y}, \widehat{z}) = 1$ with overwhelming probability $1 - \varepsilon_{ext}$. This $\widehat{z}$ is given to the simulator Sim, which acts as described in Figure 6.7.

Let $\mathcal{A}$ be a PPT adversary which, on input $\mathcal{R}, \mathsf{crs}, \widehat{z}$ and $\mathsf{aux}_{\mathcal{Z}}$, outputs $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$. We prove that $\mathcal{A}$ cannot distinguish between an honest proof generated with $\mathtt{w}$ and a simulated proof using $\widehat{z}$ by providing a sequence of indistinguishable games. Let r_{Sim} and r_{P} be the randomness used by Sim and P, respectively, and let r be some independent randomness. Let x_λ be a random element of X_λ. Let ε_{sw} be the maximal probability of breaking Sub-WI of $\Pi_{wi} = (\mathsf{K}', \mathsf{CV}', \mathsf{P}', \mathsf{V}')$, and let ε_{hide} be the maximal probability of breaking the hiding property of C (see Definition 6.5.2). By the Sub-WI property of Π_{wi} and the hiding property of C, ε_{sw} and ε_{hide} are negligible.

Game 1: This is the output of the simulator Sim in the Sub-ZK game:
$(c_{\mathsf{Sim}}, \mathsf{P}'(\mathcal{R}', \mathsf{crs}', (\mathtt{x}, c_{\mathsf{Sim}}, \widehat{y}), (\bot, \widehat{z}, r_{\mathsf{Sim}})), \mathsf{aux})$.

Game 2: We change the witness used by the prover P' to use an actual witness for $\mathtt{x}$, and output $(c_{\mathsf{Sim}}, \mathsf{P}'(\mathcal{R}', \mathsf{crs}', (\mathtt{x}, c_{\mathsf{Sim}}, \widehat{y}), (\mathtt{w}, x_\lambda, r)), \mathsf{aux})$. Because Π_{wi} is Sub-WI, this is distinguishable from Game 1 with probability at most ε_{sw}.

Game 3: We now replace c_{Sim} with c_{P} in Game 2 and output
$(c_{\mathsf{P}}, \mathsf{P}'(\mathcal{R}', \mathsf{crs}', (\mathtt{x}, c_{\mathsf{P}}, \widehat{y}), (\mathtt{w}, x_\lambda, r)), \mathsf{aux})$. By the hiding property of C, this is distinguishable from Game 2 with probability at most ε_{hide}.

Game 4: We now simply switch from using randomness r in the witness, to using randomness r_{P}, which gives us the output of the honest prover in the Sub-ZK game: $(c_{\mathsf{P}}, \mathsf{P}'(\mathcal{R}', \mathsf{crs}', (\mathtt{x}, c_{\mathsf{P}}, \widehat{y}), (\mathtt{w}, x_\lambda, r_{\mathsf{P}})), \mathsf{aux})$. By the Sub-WI property of Π_{wi}, this is distinguishable from Game 3 with probability at most ε_{sw}.

Hence Game 1 and Game 4, which are the simulator and prover's output in the Sub-ZK game, respectively, are distinguishable with probability at most $2\varepsilon_{sw} + \varepsilon_{hide}$ which is negligible, hence Π is Sub-ZK. □

Instantiations and Statistical ZAPR

We show some interesting instantiations of the above construction and also make a simple, but significant, connection between Sub-ZK NIZK and ZAPs with private random coin (ZAPRs).

Firstly, we argue that there is a knowledge-sound Sub-ZK NIZK based on the DLin and DH-KE assumptions. To the best of our knowledge, the only known knowledge-sound Sub-ZK NIZKs are Sub-ZK SNARKs. Our construction therefore relies arguably on weaker assumptions.

Proposition 6.5.4. *There exists a knowledge-sound Sub-ZK NIZK based on the DLin and DH-KE assumptions with 3 group elements as the CRS and with a proof size of $\mathcal{O}\lambda(k+l)$ where k is the circuit size and l is size of a circuit verifying the image of the DH-KE GEOWF.*

Proof. In [FO18] it is proven that there exists a knowledge-sound NIWI in the plain model based on the DLin and DH-KE assumptions. Since it has no CRS, it is also Sub-WI. From Section 6.4, there exists a VEGOWF based on the DH-KE and discrete logarithm (DL) assumptions (note that DLIN implies DL). We now apply our construction in Figure 6.5 using the knowledge-sound NIWI from [FO18] and the VEGOWF from Section 6.4. It then follows from Theorem 6.5.1 that the resulting protocol is a knowledge-sound Sub-ZK NIZK. □

Let us next prove a helpful lemma that shows when NIWI is Sub-WI. The corollary follows since perfect zero knowledge implies perfect WI.

Lemma 6.5.5. *Suppose Ψ is perfectly WI for relation $\mathcal{R}$ and there exists an efficient CRS validation algorithm CV. Then Ψ is Sub-WI.*

Proof. Definition 6.3.9 for perfect WI states that for all honestly generated CRS crs (i.e., CRS in the image of $\mathsf{K}(\mathcal{R})$), instances $\mathtt{x}$, and corresponding witnesses $\mathtt{w}_0, \mathtt{w}_1$, no unbounded adversary can distinguish a proof generated using either $(\mathsf{crs}, \mathtt{x}, \mathtt{w}_0)$ or $(\mathsf{crs}, \mathtt{x}, \mathtt{w}_1)$. Note that if a subverter can create a valid crs such that $\mathcal{A}$ breaks Sub-WI with probability at least $\varepsilon > 0$, the same $\mathcal{A}$ can break WI with probability at least $\varepsilon/(|\mathsf{crs}| + |\mathsf{aux}_{\mathcal{Z}}|) > 0$ by simply guessing crs and $\mathsf{aux}_{\mathcal{Z}}$. Hence assuming perfect WI, verifying that a subverter-generated CRS crs is in fact in the image of $\mathsf{K}(\mathcal{R})$ is enough to assure that perfect subversion WI holds. □

Corollary 6.5.6. *If Ψ is perfectly zero-knowledge and there exist an efficient CRS validation algorithm, then Ψ is Sub-WI.*

Therefore, the efficient SNARK constructions in [Abd+17; Fuc18], the updatable SNARKs in [Gro+18; Mal+19], and the shuffle argument in [Agg+20] are all Sub-WI. The same observation about Sub-ZK SNARKs was already made by Fuchsbauer in[Fuc18]. These arguments have a CRS validation algorithm and were already known to be Sub-ZK under a knowledge assumption. However, the above result shows that they are perfect Sub-WI *without any assumptions*. Moreover, any NIWI without a CRS is trivially Sub-WI.

Firstly, it means that [Abd+17; Fuc18] are statistical ZAPRs with adaptive soundness. The only other pairing-based ZAPR is [LVW20] which is less efficient and uses much more advanced tools, but relies on weaker assumptions for soundness. Secondly, if we use the SNARKs of [Abd+17; Fuc18] in Figure 6.5, we have Sub-ZK SNARKs from any VEGOWF rather than from a specific knowledge assumption.

Proposition 6.5.7. *Suppose there exists a perfectly zero-knowledge SNARK with an efficient CRS validation algorithm* CV *and there exists a VEGOWF. Then there exists a Sub-ZK SNARK.*

Proof. Since the given SNARK Π is perfectly ZK and has a CV algorithm, it follows from Corollary 6.5.6 that it is perfectly Sub-WI. Applying our construction in Section 6.5 to Π and the VEGOWF $\mathcal{G}$ to construct a new SNARK Π', it then follows from part (4) of Theorem 6.5.1 that Π' is a Sub-ZK SNARK, as desired. □

6.6 Characterising Sub-ZK NIZKs

We show that the CRS generation algorithm K of a NIZK is a VEOWF if and only if the NIZK is Sub-ZK. Let RGen be a relation generator, and let $\Pi = (\mathsf{K}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$ be a NIZK argument for RGen. We define a family of functions $\mathcal{G}_{\mathsf{K}} = \left\{\mathsf{K}_{\mathcal{R},\mathsf{p}} \colon \{\mathsf{td}\} \to \{\mathsf{crs}\} \mid (\mathcal{R}, \mathsf{p}) \in \mathsf{RGen}(1^\lambda), \lambda \in \mathbb{N}\right\}$ where $\mathsf{K}_{\mathcal{R},\mathsf{p}}$ takes in a uniformly sampled trapdoor td and maps it deterministically to a crs. We assume that the distribution $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{Kgen}(\mathcal{R}, \mathsf{p})$ is the same as $(\mathsf{crs} \leftarrow \mathsf{K}_{\mathcal{R},\mathsf{p}}(\mathsf{td}), \mathsf{td} \leftarrow\!\!\$\, \{\mathsf{td}\})$. We use both notations interchangeably in this section.

Let us start by establishing the following straightforward connection.

Theorem 6.6.1 (VEOWF $\mathcal{G}_{\mathsf{K}} \implies$ Sub-ZK)**.** *Suppose* $\Pi = (\mathsf{K}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$ *is a perfect NIZK argument. If* $\mathcal{G}_{\mathsf{K}}$ *is a VEOWF with image verification algorithm* ImV*, then* Π *is statistically composable Sub-ZK with respect to the CRS verification algorithm* $\mathsf{CV} = \mathsf{ImV}$*.*

Proof. Consider a subverter $\mathcal{Z}$ which outputs a CRS crs. We only need to consider the case where $\mathsf{CV}(\mathsf{crs}) = 1$ and thus $\mathsf{crs} \in \mathrm{im}(\mathsf{K}_{\mathcal{R},\mathsf{p}})$. Since $\mathsf{K}_{\mathcal{R},\mathsf{p}}$ is a VEOWF and the subverter $\mathcal{Z}$ outputs an image of $\mathsf{K}_{\mathcal{R},\mathsf{p}}$, we know that there exists an extractor $\mathsf{Ext}_{\mathcal{Z}}$ which with overwhelming probability outputs a simulation trapdoor td. Since Π is perfect zero-knowledge, proofs $\pi_0 \leftarrow \mathsf{Sim}(\mathcal{R}, \mathsf{p}, \mathsf{td}, \mathsf{crs}, \mathtt{x})$ and $\pi_1 \leftarrow \mathsf{P}(\mathcal{R}, \mathsf{p}, \mathsf{crs}, \mathtt{x}, \mathtt{w})$ are identically distributed. □

Remark. The same result does not hold for statistical (or computational) NIZK since there might be a negligible number of CRSs where td does not allow simulation, which the subverter could output.

Following [Gro16], we say that the relation generator RGen *has a* $\varepsilon_{\mathcal{S}}$*-hard decisional problem* if there exist two samplers $\mathcal{S}$ and $\mathcal{S}'$ such that for $(\mathcal{R}, \mathsf{p}) \leftarrow \mathsf{RGen}(1^\lambda)$ (1) sampler $\mathcal{S}(\mathcal{R}, \mathsf{p})$ produces $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$, and (2) $\mathcal{S}'(\mathcal{R}, \mathsf{p})$ produces $\mathtt{x} \notin \mathcal{L}_{\mathcal{R}}$. Furthermore, for some negligible $\varepsilon_{\mathcal{S}}$, it holds for all PPT adversaries $\mathcal{A}$ that $|\varepsilon_0 - \varepsilon_1| \leq \varepsilon_{\mathcal{S}}$, where $\varepsilon_b = \Pr\left[(\mathcal{R}, \mathsf{p}) \leftarrow \mathsf{RGen}(1^\lambda), (\mathtt{x}_0, \mathtt{w}_0) \leftarrow \mathcal{S}(\mathcal{R}, \mathsf{p}), \mathtt{x}_1 \leftarrow \mathcal{S}'(\mathcal{R}, \mathsf{p}) : \mathcal{A}(\mathcal{R}, \mathsf{p}, \mathtt{x}_b) = 1\right]$.

A simple example of this is the language of Diffie-Hellman tuples where $\mathsf{p} = (\mathbb{G}, g, p) \leftarrow \mathsf{RGen}(1^\lambda)$ is a group description, $\mathcal{S}$ outputs $(\mathtt{x} = (g^x, g^y, g^{xy}), \mathtt{w} = (x, y))$ for random $x, y \leftarrow\!\!\$\, \mathbb{Z}_p$, and $\mathcal{S}'$ outputs g^x, g^y, g^z for random $x, y \leftarrow\!\!\$\, \mathbb{Z}_p$ and $z \leftarrow\!\!\$\, \mathbb{Z}_p \setminus \{xy\}$.

Now let us establish the opposite connection between VEOWF and Sub-ZK. In general, the extractor in subversion zero-knowledge definition does not need to extract the whole preimage of the CRS function. It just needs to extract something which allows for simulation of proofs. For example, this could be only a small part of the full trapdoor. Due to this, we restrict ourselves slightly and lend the following notion from [Abd+20b].

Definition 6.6.2 (Trapdoor-Extractability [Abd+20b])**.** A subversion-resistant argument Ψ for a relation RGen has trapdoor-extractability if for any PPT subverter $\mathcal{Z}$ there exists a PPT extractor $\mathsf{Ext}_{\mathcal{Z}}$, s.t. for all λ and $(\mathcal{R}, \mathsf{p}) \in \mathsf{RGen}(1^\lambda)$,

$$\Pr\left[\begin{array}{l} r \leftarrow\!\!\$\, \mathsf{RND}_\lambda(\mathcal{Z}), \mathsf{crs} \leftarrow \mathcal{Z}(\mathcal{R}, \mathsf{p}; r), \mathsf{td} \leftarrow \mathsf{Ext}_{\mathcal{Z}}(\mathcal{R}, \mathsf{p}; r) : \\ \mathsf{CV}(\mathcal{R}, \mathsf{p}, \mathsf{crs}) = 1 \wedge \mathsf{K}_{\mathcal{R},\mathsf{p}}(\mathsf{td}) \neq \mathsf{crs} \end{array}\right] \leq \mathsf{negl}(\lambda) \ .$$

Theorem 6.6.3 (Sub-ZK $\implies$ VEOWF $\mathcal{G}_{\mathsf{K}}$)**.** *Assume Π is a NIZK argument for RGen, which has $\varepsilon_{\mathcal{S}}$-hard decisional problems. Let $\mathcal{G}_{\mathsf{K}}$ be as defined above. Assume the distribution $\mathcal{D}_\lambda$ is benign. Then*

1. *if (i) $\Pi = (\mathsf{K}, \mathsf{P}, \mathsf{V}, \mathsf{Sim})$ is perfectly complete, computationally sound, and computationally zero-knowledge, and (ii) $\mathsf{K}_{\mathcal{R},\mathsf{p}}$ is injective, then $\mathcal{G}_{\mathsf{K}}$ is a one-way function;*
2. *if $\Pi = (\mathsf{K}, \mathsf{P}, \mathsf{V}, \mathsf{Sim}, \mathsf{CV})$ is a statistically composable Sub-ZK argument with trapdoor extractability, then $\mathcal{G}_{\mathsf{K}}$ is verifiably-extractable with $\mathcal{G}_{\mathsf{K}}.\mathsf{ImV} = \Pi.\mathsf{CV}$ respect to auxiliary inputs $(\mathcal{R}, \mathsf{p}, r)$ where $(\mathcal{R}, \mathsf{p}) \leftarrow \mathsf{RGen}(1^\lambda)$, $r \leftarrow\!\!\$\, \{0,1\}^{\mathsf{poly}(\lambda)}$.*

Proof. **Soundness + ZK $\implies$ One-Wayness.** Suppose there exists a PPT adversary $\mathcal{A}$ that breaks one-wayness of $\mathcal{G}_{\mathsf{K}}$ with probability ε_{owf}. That is, for a random $(\mathcal{R}, \mathsf{p}) \leftarrow \mathsf{KeySamp}_{\mathcal{G}}(1^\lambda)$, $\mathsf{td} \leftarrow\!\!\$\, \{\mathsf{td}\}, \mathsf{aux} \leftarrow\!\!\$\, \mathcal{D}_\lambda$, the $\mathcal{A}(\mathcal{R}, \mathsf{p}, \mathsf{crs} = \mathsf{K}_{\mathcal{R},\mathsf{p}}(\mathsf{td}), \mathsf{aux})$ outputs td' such that $\mathsf{K}_{\mathcal{R},\mathsf{p}}(\mathsf{td}') = \mathsf{crs}$ with probability ε_{owf}.

We are going to construct a PPT adversary $\mathcal{B}$ that internally runs $\mathcal{A}$ together with an auxiliary input aux. We build the soundness adversary $\mathcal{B}$ as follows:

1. $\mathcal{B}$ gets $(\mathcal{R}, \mathsf{p}, \mathsf{crs})$ as an input;
2. $\mathcal{B}$ samples $\mathsf{aux}' \leftarrow\!\!\$\, \mathcal{D}_\lambda$ and computes $\mathsf{td}' \leftarrow \mathcal{A}(\mathcal{R}, \mathsf{p}, \mathsf{crs}, \mathsf{aux}')$;
3. $\mathcal{B}$ outputs $\mathtt{x}$ such that $\mathtt{x} \leftarrow \mathcal{S}'(\mathcal{R}, \mathsf{p})$ (i.e. $\mathtt{x} \notin \mathcal{L}_{\mathcal{R}}$) along with a simulated proof $\pi \leftarrow \mathsf{Sim}(\mathcal{R}, \mathsf{p}, \mathsf{crs}, \mathsf{td}', \mathtt{x})$.

Since $\mathtt{x} \notin \mathcal{L}_{\mathcal{R}}$ by definition, it means that $\mathcal{B}$ wins the soundness game if $\mathsf{V}(\mathcal{R}, \mathsf{p}, \mathsf{crs}, \mathtt{x}, \pi) = 1$. We use games in Figure 6.9 to quantify the probability that $\mathsf{V}(\mathcal{R}, \mathsf{p}, \mathsf{crs}, \mathtt{x}, \pi) = 1$ in the soundness game.

Game 0: This is the original soundness game without the condition $\mathtt{x} \notin \mathcal{L}_{\mathcal{R}}$ with the adversary $\mathcal{B}$ inlined. The winning condition is just $\mathsf{V}(\mathcal{R}, \mathsf{p}, \mathsf{crs}, \mathtt{x}, \pi) = 1$.

Game 1: We change Game 0 such that $\mathcal{B}$ samples a true statement-witness pair $(\mathtt{x}, \mathtt{w}) \leftarrow \mathcal{S}(\mathcal{R}, \mathsf{p})$ instead.

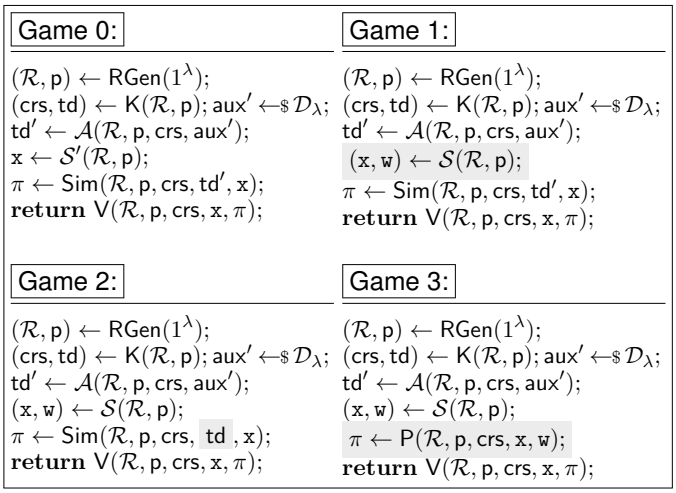


Figure 6.9: Security games for Theorem 6.6.3.

Game 2: We modify Game 1 such that the simulator gets the real trapdoor td as an input rather than the trapdoor td' extracted by $\mathcal{A}$.

Game 3: Finally, instead of simulating the proof π, we use the witness $\mathtt{w}$ to create an honest proof.

Let us denote the probability of Game i outputting 1 by ε_i. Firstly, it is clear that ε_0 is the probability of $\mathcal{B}$ winning (that is, outputting 1) in the soundness game since, although, we do not check the condition $\mathtt{x} \notin \mathcal{L}_\mathcal{R}$, it always holds for the adversary $\mathcal{B}$. We now prove that distinguishing Game 0 and Game 1 succeeds with probability at most $\varepsilon_\mathcal{S}$.

Lemma 6.6.4. *For the probabilities ε_0 and ε_1 defined as above, $|\varepsilon_0 - \varepsilon_1| \leq \varepsilon_\mathcal{S}$.*

Proof. Consider the following adversary $\mathcal{C}$ against the $\varepsilon_\mathcal{S}$-hardness. Firstly, $\mathcal{C}$ gets as an input $(\mathcal{R}, \mathsf{p}, \mathtt{x}_b)$ where $\mathtt{x}_1$ is generated by $\mathcal{S}$ and $\mathtt{x}_0$ is generated by $\mathcal{S}'$. Then, $\mathcal{C}$ samples $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{K}(\mathcal{R}, \mathsf{p})$ and $\mathsf{aux}' \leftarrow\$ \mathcal{D}_\lambda$, computes $\mathsf{td}' \leftarrow \mathcal{A}(\mathcal{R}, \mathsf{p}, \mathsf{crs}, \mathsf{aux}')$, and simulates the proof $\pi \leftarrow \mathsf{Sim}(\mathcal{R}, \mathsf{p}, \mathsf{crs}, \mathsf{td}', \mathtt{x})$. It returns the answer of $\mathsf{V}(\mathcal{R}, \mathsf{p}, \mathsf{crs}, \mathtt{x}, \pi)$.

By construction, the probability that $\mathcal{C}$ outputs 1 given $\mathtt{x}_0$ is ε_0 and given $\mathtt{x}_1$ is ε_1. It thus follows that $|\varepsilon_0 - \varepsilon_1| \leq \varepsilon_\mathcal{S}$. □

Lemma 6.6.5. *Assuming that $\mathsf{K}_{\mathcal{R},\mathsf{p}}$ is injective, $|\varepsilon_1 - \varepsilon_2| \leq 1 - \varepsilon_{owf}$.*

Proof. The only difference between Game 1 and Game 2 is that one uses td' for simulation and the other uses td. If $\mathcal{A}$ is successful in breaking one-wayness, then $\mathsf{td} = \mathsf{td}'$ (since $\mathsf{K}_{\mathcal{R},\mathsf{p}}$ is injective) and output distributions of both games are the same. That happens with probability ε_{owf}. Outputs distributions of games can differ only when $\mathcal{A}$ fails in

breaking one-wayness, which happens at most with the probability $1-\varepsilon_{owf}$. We conclude that $|\varepsilon_1 - \varepsilon_2| \leq 1 - \varepsilon_{owf}$. □

The final game transition is based on the zero-knowledge property.

Lemma 6.6.6. *Let ε_{zk} denote the maximum advantage that any PPT adversary wins in the zero-knowledge game. Then, $|\varepsilon_2 - \varepsilon_3| \leq \varepsilon_{zk}$.*

Proof. Consider the verifier V as the adversary in the zero-knowledge game. From this perspective Game 2 is the zero-knowledge game with the simulator and Game 3 is the zero-knowledge game with the honest prover given that we ignore the line $\mathsf{td}' \leftarrow \mathcal{A}(\mathcal{R}, \mathsf{p}, \mathsf{crs}, \mathsf{aux})$. It follows that $|\varepsilon_2 - \varepsilon_3| \leq \varepsilon_{zk}$. □

Using the triangle inequality, we now get that $|\varepsilon_0 - \varepsilon_3| \leq \varepsilon_{\mathcal{S}} + (1-\varepsilon_{owf}) + \varepsilon_{zk}$. Since the argument system is perfectly complete, $\varepsilon_3 = 1$ and therefore $|\varepsilon_0 - \varepsilon_3| = |\varepsilon_0 - 1| = 1 - \varepsilon_0$. Putting equations together, we get $1 - \varepsilon_0 \leq \varepsilon_{\mathcal{S}} + (1 - \varepsilon_{owf}) + \varepsilon_{zk}$, which can be simplified to $\varepsilon_{owf} \leq \varepsilon_0 + \varepsilon_{\mathcal{S}} + \varepsilon_{zk}$, which is negligible.

Sub-ZK $\implies$ verifiable-extractability. This part of the proof is essentially tautological. Let $\mathcal{A}$ be an adversary in the verifiable extractability game and let $\mathsf{aux} = (\mathcal{R}, \mathsf{p}, r)$ where $(\mathcal{R}, \mathsf{p}) \leftarrow \mathsf{RGen}(1^\lambda)$ and $r \leftarrow\!\!\$\, \{0,1\}^{\mathsf{poly}(\lambda)}$. Suppose that $\mathcal{A}$ is Sub-ZK subverter that outputs crs such that $\mathsf{CV}(\mathcal{R}, \mathsf{p}, \mathsf{crs}) = 1$. Then according to the trapdoor extractability property, there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$ that on input aux, outputs with an overwhelming td such that $\mathsf{K}_{\mathcal{R},\mathsf{p}}(\mathsf{td}) = \mathsf{crs}$. Thus, verifiable extractability holds. □

Acknowledgements. Janno Siim and Helger Lipmaa were partially supported by the Estonian Research Council grant (PRG49).

References

[Abd+17] Behzad Abdolmaleki, Karim Baghery, Helger Lipmaa, and Michal Zajac. "A Subversion-Resistant SNARK". In: *ASIACRYPT 2017, Part III*. Ed. by Tsuyoshi Takagi and Thomas Peyrin. Vol. 10626. LNCS. Springer, Heidelberg, Dec. 2017, pp. 3–33. DOI: 10.1007/978-3-319-70700-6_1.

[Abd+20a] Behzad Abdolmaleki, Helger Lipmaa, Janno Siim, and Michal Zajac. "On QA-NIZK in the BPK Model". In: *PKC 2020, Part I*. Ed. by Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas. Vol. 12110. LNCS. Springer, Heidelberg, May 2020, pp. 590–620. DOI: 10.1007/978-3-030-45374-9_20.

[Abd+20b] Behzad Abdolmaleki, Helger Lipmaa, Janno Siim, and Michał Zając. *On Subversion-Resistant SNARKs*. Cryptology ePrint Archive, Report 2020/668. https://eprint.iacr.org/2020/668. 2020.

[Agg+20] Antonis Aggelakis, Prastudy Fauzi, Georgios Korfiatis, Panos Louridas, Foteinos Mergoupis-Anagnou, Janno Siim, and Michal Zajac. "A Non-interactive Shuffle Argument with Low Trust Assumptions". In: *CT-RSA 2020*. Ed. by Stanislaw Jarecki. Vol. 12006. LNCS. Springer, Heidelberg, Feb. 2020, pp. 667–692. DOI: 10.1007/978-3-030-40186-3_28.

[Aie+00] William Aiello, Sandeep N. Bhatt, Rafail Ostrovsky, and Sivaramakrishnan Rajagopalan. "Fast Verification of Any Remote Procedure Call: Short Witness-Indistinguishable One-Round Proofs for NP". In: *ICALP 2000*. Ed. by Ugo Montanari, José D. P. Rolim, and Emo Welzl. Vol. 1853. LNCS. Springer, Heidelberg, July 2000, pp. 463–474. DOI: 10.1007/3-540-45022-X_39.

[Bad+20] Saikrishna Badrinarayanan, Rex Fernando, Aayush Jain, Dakshita Khurana, and Amit Sahai. "Statistical ZAP Arguments". In: *EUROCRYPT 2020, Part III*. Ed. by Anne Canteaut and Yuval Ishai. Vol. 12107. LNCS. Springer, Heidelberg, May 2020, pp. 642–667. DOI: 10.1007/978-3-030-45727-3_22.

[Bar+01] Boaz Barak, Oded Goldreich, Shafi Goldwasser, and Yehuda Lindell. "Resettably-Sound Zero-Knowledge and its Applications". In: *42nd FOCS*. IEEE Computer Society Press, Oct. 2001, pp. 116–125. DOI: 10.1109/SFCS.2001.959886.

[Bar01] Boaz Barak. "How to Go Beyond the Black-Box Simulation Barrier". In: *42nd FOCS*. IEEE Computer Society Press, Oct. 2001, pp. 106–115. DOI: 10.1109/SFCS.2001.959885.

[BB04] Dan Boneh and Xavier Boyen. "Short Signatures Without Random Oracles". In: *EUROCRYPT 2004*. Ed. by Christian Cachin and Jan Camenisch. Vol. 3027. LNCS. Springer, Heidelberg, May 2004, pp. 56–73. DOI: 10.1007/978-3-540-24676-3_4.

[Ben+14] Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. "Zerocash: Decentralized Anonymous Payments from Bitcoin". In: *2014 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2014, pp. 459–474. DOI: 10.1109/SP.2014.36.

[Ben+15] Eli Ben-Sasson, Alessandro Chiesa, Matthew Green, Eran Tromer, and Madars Virza. "Secure Sampling of Public Parameters for Succinct Zero Knowledge Proofs". In: *2015 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2015, pp. 287–304. DOI: 10.1109/SP.2015.25.

[BFS16] Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. "NIZKs with an Untrusted CRS: Security in the Face of Parameter Subversion". In: *ASIACRYPT 2016, Part II*. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10032. LNCS. Springer, Heidelberg, Dec. 2016, pp. 777–804. DOI: 10.1007/978-3-662-53890-6_26.

[BG93] Mihir Bellare and Oded Goldreich. "On Defining Proofs of Knowledge". In: *CRYPTO'92*. Ed. by Ernest F. Brickell. Vol. 740. LNCS. Springer, Heidelberg, Aug. 1993, pp. 390–420. DOI: 10.1007/3-540-48071-4_28.

[Bic+10] Patrik Bichsel, Jan Camenisch, Gregory Neven, Nigel P. Smart, and Bogdan Warinschi. "Get Shorty via Group Signatures without Encryption". In: *SCN 10*. Ed. by Juan A. Garay and Roberto De Prisco. Vol. 6280. LNCS. Springer, Heidelberg, Sept. 2010, pp. 381–398. DOI: 10.1007/978-3-642-15317-4_24.

[Bit+12] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. "From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again". In: *ITCS 2012*. Ed. by Shafi Goldwasser. ACM, Jan. 2012, pp. 326–349. DOI: 10.1145/2090236.2090263.

[Bit+16] Nir Bitansky, Ran Canetti, Omer Paneth, and Alon Rosen. "On the Existence of Extractable One-Way Functions". In: *SIAM J. Comput.* 45.5 (2016), pp. 1910–1952.

[Bit+17] Nir Bitansky, Ran Canetti, Alessandro Chiesa, Shafi Goldwasser, Huijia Lin, Aviad Rubinstein, and Eran Tromer. "The Hunting of the SNARK". In: *Journal of Cryptology* 30.4 (Oct. 2017), pp. 989–1066. DOI: 10.1007/s00145-016-9241-9.

[BLS04] Dan Boneh, Ben Lynn, and Hovav Shacham. "Short Signatures from the Weil Pairing". In: *Journal of Cryptology* 17.4 (Sept. 2004), pp. 297–319. DOI: 10.1007/s00145-004-0314-9.

[BOV03] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. "Derandomization in Cryptography". In: *CRYPTO 2003*. Ed. by Dan Boneh. Vol. 2729. LNCS. Springer, Heidelberg, Aug. 2003, pp. 299–315. DOI: 10.1007/978-3-540-45146-4_18.

[BP15a] Nir Bitansky and Omer Paneth. "ZAPs and Non-Interactive Witness Indistinguishability from Indistinguishability Obfuscation". In: *TCC 2015, Part II*. Ed. by Yevgeniy Dodis and Jesper Buus Nielsen. Vol. 9015. LNCS. Springer, Heidelberg, Mar. 2015, pp. 401–427. DOI: 10.1007/978-3-662-46497-7_16.

[BP15b] Elette Boyle and Rafael Pass. "Limits of Extractability Assumptions with Distributional Auxiliary Input". In: *ASIACRYPT 2015, Part II*. Ed. by Tetsu Iwata and Jung Hee Cheon. Vol. 9453. LNCS. Springer, Heidelberg, Nov. 2015, pp. 236–261. DOI: 10.1007/978-3-662-48800-3_10.

[Can+00] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. "Resettable zero-knowledge (extended abstract)". In: *32nd ACM STOC*. ACM Press, May 2000, pp. 235–244. DOI: 10.1145/335305.335334.

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. "Extractable Perfectly One-Way Functions". In: *ICALP 2008, Part II*. Ed. by Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz. Vol. 5126. LNCS. Springer, Heidelberg, July 2008, pp. 449–460. DOI: 10.1007/978-3-540-70583-3_37.

[CD09] Ran Canetti and Ronny Ramzi Dakdouk. "Towards a Theory of Extractable Functions". In: *TCC 2009*. Ed. by Omer Reingold. Vol. 5444. LNCS. Springer, Heidelberg, Mar. 2009, pp. 595–613. DOI: 10.1007/978-3-642-00457-5_35.

[CL06] Melissa Chase and Anna Lysyanskaya. "On Signatures of Knowledge". In: *CRYPTO 2006*. Ed. by Cynthia Dwork. Vol. 4117. LNCS. Springer, Heidelberg, Aug. 2006, pp. 78–96. DOI: 10.1007/11818175_5.

[CLP13] Kai-Min Chung, Huijia Lin, and Rafael Pass. "Constant-Round Concurrent Zero Knowledge from P-Certificates". In: *54th FOCS*. IEEE Computer Society Press, Oct. 2013, pp. 50–59. DOI: 10.1109/FOCS.2013.14.

[Dak09] Ramzi Ronny Dakdouk. "Theory and Application of Extractable Functions". PhD thesis. Yale University, 2009, p. 229.

[Dam92] Ivan Damgård. "Towards Practical Public Key Systems Secure Against Chosen Ciphertext Attacks". In: *CRYPTO'91*. Ed. by Joan Feigenbaum. Vol. 576. LNCS. Springer, Heidelberg, Aug. 1992, pp. 445–456. DOI: 10.1007/3-540-46766-1_36.

[Dan+14] George Danezis, Cédric Fournet, Jens Groth, and Markulf Kohlweiss. "Square Span Programs with Applications to Succinct NIZK Arguments". In: *ASIACRYPT 2014, Part I*. Ed. by Palash Sarkar and Tetsu Iwata. Vol. 8873. LNCS. Springer, Heidelberg, Dec. 2014, pp. 532–550. DOI: 10.1007/978-3-662-45611-8_28.

[FLP08] Marc Fischlin, Anja Lehmann, and Krzysztof Pietrzak. "Robust Multi-property Combiners for Hash Functions Revisited". In: *ICALP 2008, Part II*. Ed. by Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz. Vol. 5126. LNCS. Springer, Heidelberg, July 2008, pp. 655–666. DOI: 10.1007/978-3-540-70583-3_53.

[FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir. "Multiple Non-Interactive Zero Knowledge Proofs Based on a Single Random String (Extended Abstract)". In: *31st FOCS*. IEEE Computer Society Press, Oct. 1990, pp. 308–317. DOI: 10.1109/FSCS.1990.89549.

[FO18] Georg Fuchsbauer and Michele Orrù. "Non-interactive Zaps of Knowledge". In: *ACNS 18*. Ed. by Bart Preneel and Frederik Vercauteren. Vol. 10892. LNCS. Springer, Heidelberg, July 2018, pp. 44–62. DOI: 10.1007/978-3-319-93387-0_3.

[FPS20] Georg Fuchsbauer, Antoine Plouviez, and Yannick Seurin. "Blind Schnorr Signatures and Signed ElGamal Encryption in the Algebraic Group Model". In: *EUROCRYPT 2020, Part II*. Ed. by Anne Canteaut and Yuval Ishai. Vol. 12106. LNCS. Springer, Heidelberg, May 2020, pp. 63–95. DOI: 10.1007/978-3-030-45724-2_3.

[Fuc18] Georg Fuchsbauer. "Subversion-Zero-Knowledge SNARKs". In: *PKC 2018, Part I*. Ed. by Michel Abdalla and Ricardo Dahab. Vol. 10769. LNCS. Springer, Heidelberg, Mar. 2018, pp. 315–347. DOI: 10.1007/978-3-319-76578-5_11.

[Gen+13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. "Quadratic Span Programs and Succinct NIZKs without PCPs". In: *EUROCRYPT 2013*. Ed. by Thomas Johansson and Phong Q. Nguyen. Vol. 7881. LNCS. Springer, Heidelberg, May 2013, pp. 626–645. DOI: 10.1007/978-3-642-38348-9_37.

[GL89] Oded Goldreich and Leonid A. Levin. "A Hard-Core Predicate for all One-Way Functions". In: *21st ACM STOC*. ACM Press, May 1989, pp. 25–32. DOI: 10.1145/73007.73010.

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

[GO94] Oded Goldreich and Yair Oren. "Definitions and Properties of Zero-Knowledge Proof Systems". In: *Journal of Cryptology* 7.1 (Dec. 1994), pp. 1–32. DOI: 10.1007/BF00195207.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. "Non-interactive Zaps and New Techniques for NIZK". In: *CRYPTO 2006*. Ed. by Cynthia Dwork. Vol. 4117. LNCS. Springer, Heidelberg, Aug. 2006, pp. 97–111. DOI: 10.1007/11818175_6.

[Goy+20] Vipul Goyal, Abhishek Jain, Zhengzhong Jin, and Giulio Malavolta. "Statistical Zaps and New Oblivious Transfer Protocols". In: *EUROCRYPT 2020, Part III*. Ed. by Anne Canteaut and Yuval Ishai. Vol. 12107. LNCS. Springer, Heidelberg, May 2020, pp. 668–699. DOI: 10.1007/978-3-030-45727-3_23.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. "Trapdoors for hard lattices and new cryptographic constructions". In: *40th ACM STOC*. Ed. by Richard E. Ladner and Cynthia Dwork. ACM Press, May 2008, pp. 197–206. DOI: 10.1145/1374376.1374407.

[Gro+18] Jens Groth, Markulf Kohlweiss, Mary Maller, Sarah Meiklejohn, and Ian Miers. "Updatable and Universal Common Reference Strings with Applications to zk-SNARKs". In: *CRYPTO 2018, Part III*. Ed. by Hovav Shacham and Alexandra Boldyreva. Vol. 10993. LNCS. Springer, Heidelberg, Aug. 2018, pp. 698–728. DOI: 10.1007/978-3-319-96878-0_24.

[Gro06] Jens Groth. "Simulation-Sound NIZK Proofs for a Practical Language and Constant Size Group Signatures". In: *ASIACRYPT 2006*. Ed. by Xuejia Lai and Kefei Chen. Vol. 4284. LNCS. Springer, Heidelberg, Dec. 2006, pp. 444–459. DOI: 10.1007/11935230_29.

[Gro10] Jens Groth. "Short Pairing-Based Non-interactive Zero-Knowledge Arguments". In: *ASIACRYPT 2010*. Ed. by Masayuki Abe. Vol. 6477. LNCS. Springer, Heidelberg, Dec. 2010, pp. 321–340. DOI: 10.1007/978-3-642-17373-8_19.

[Gro16] Jens Groth. "On the Size of Pairing-Based Non-interactive Arguments". In: *EUROCRYPT 2016, Part II*. Ed. by Marc Fischlin and Jean-Sébastien Coron. Vol. 9666. LNCS. Springer, Heidelberg, May 2016, pp. 305–326. DOI: 10.1007/978-3-662-49896-5_11.

[GW11] Craig Gentry and Daniel Wichs. "Separating succinct non-interactive arguments from all falsifiable assumptions". In: *43rd ACM STOC*. Ed. by Lance Fortnow and Salil P. Vadhan. ACM Press, June 2011, pp. 99–108. DOI: 10.1145/1993636.1993651.

[Har+05] Danny Harnik, Joe Kilian, Moni Naor, Omer Reingold, and Alon Rosen. "On Robust Combiners for Oblivious Transfer and Other Primitives". In: *EUROCRYPT 2005*. Ed. by Ronald Cramer. Vol. 3494. LNCS. Springer, Heidelberg, May 2005, pp. 96–113. DOI: 10.1007/11426639_6.

[Her05] Amir Herzberg. "On Tolerant Cryptographic Constructions". In: *CT-RSA 2005*. Ed. by Alfred Menezes. Vol. 3376. LNCS. Springer, Heidelberg, Feb. 2005, pp. 172–190. DOI: 10.1007/978-3-540-30574-3_13.

[JLS20] Aayush Jain, Huijia Lin, and Amit Sahai. *Indistinguishability Obfuscation from Well-Founded Assumptions*. Cryptology ePrint Archive, Report 2020/1003. https://eprint.iacr.org/2020/1003. 2020.

[JR13] Charanjit S. Jutla and Arnab Roy. "Shorter Quasi-Adaptive NIZK Proofs for Linear Subspaces". In: *ASIACRYPT 2013, Part I*. Ed. by Kazue Sako and Palash Sarkar. Vol. 8269. LNCS. Springer, Heidelberg, Dec. 2013, pp. 1–20. DOI: 10.1007/978-3-642-42033-7_1.

[Kat+19] Shuichi Katsumata, Ryo Nishimaki, Shota Yamada, and Takashi Yamakawa. "Exploring Constructions of Compact NIZKs from Various Assumptions". In: *CRYPTO 2019, Part III*. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11694. LNCS. Springer, Heidelberg, Aug. 2019, pp. 639–669. DOI: 10.1007/978-3-030-26954-8_21.

[KPY19] Yael Tauman Kalai, Omer Paneth, and Lisa Yang. "How to delegate computations publicly". In: *51st ACM STOC*. Ed. by Moses Charikar and Edith Cohen. ACM Press, June 2019, pp. 1115–1124. DOI: 10.1145/3313276.3316411.

[KRR14] Yael Tauman Kalai, Ran Raz, and Ron D. Rothblum. "How to delegate computations: the power of no-signaling proofs". In: *46th ACM STOC*. Ed. by David B. Shmoys. ACM Press, May 2014, pp. 485–494. DOI: 10.1145/2591796.2591809.

[Lam79] Leslie Lamport. *Constructing Digital Signatures from a One-way Function*. Technical Report SRI-CSL-98. SRI International Computer Science Laboratory, Oct. 1979.

[Lep02] Matthew Lepinski. "On the existence of 3-round zero-knowledge proofs". MA thesis. MIT, USA, 2002.

[Lip12] Helger Lipmaa. "Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments". In: *TCC 2012*. Ed. by Ronald Cramer. Vol. 7194. LNCS. Springer, Heidelberg, Mar. 2012, pp. 169–189. DOI: 10.1007/978-3-642-28914-9_10.

[LVW20] Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. "Statistical ZAPR Arguments from Bilinear Maps". In: *EUROCRYPT 2020, Part III*. Ed. by Anne Canteaut and Yuval Ishai. Vol. 12107. LNCS. Springer, Heidelberg, May 2020, pp. 620–641. DOI: 10.1007/978-3-030-45727-3_21.

[Mal+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 CCS 2019*. Ed. by Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz. ACM Press, Nov. 2019, pp. 2111–2128. DOI: 10.1145/3319535.3339817.

[Mau05] Ueli M. Maurer. "Abstract Models of Computation in Cryptography (Invited Paper)". In: *10th IMA International Conference on Cryptography and Coding*. Ed. by Nigel P. Smart. Vol. 3796. LNCS. Springer, Heidelberg, Dec. 2005, pp. 1–12.

[Mic00] Silvio Micali. "Computationally Sound Proofs". In: *SIAM J. Comput.* 30.4 (2000), pp. 1253–1298.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". In: *Communications of the Association for Computing Machinery* 21.2 (1978), pp. 120–126.

[Rus+16] Alexander Russell, Qiang Tang, Moti Yung, and Hong-Sheng Zhou. "Cliptography: Clipping the Power of Kleptographic Attacks". In: *ASIACRYPT 2016, Part II*. Ed. by Jung Hee Cheon and Tsuyoshi Takagi. Vol. 10032. LNCS. Springer, Heidelberg, Dec. 2016, pp. 34–64. DOI: 10.1007/978-3-662-53890-6_2.

[Sch90] Claus-Peter Schnorr. "Efficient Identification and Signatures for Smart Cards". In: *CRYPTO'89*. Ed. by Gilles Brassard. Vol. 435. LNCS. Springer, Heidelberg, Aug. 1990, pp. 239–252. DOI: 10.1007/0-387-34805-0_22.

[SS01] Ahmad-Reza Sadeghi and Michael Steiner. "Assumptions Related to Discrete Logarithms: Why Subtleties Make a Real Difference". In: *EUROCRYPT 2001*. Ed. by Birgit Pfitzmann. Vol. 2045. LNCS. Springer, Heidelberg, May 2001, pp. 244–261. DOI: 10.1007/3-540-44987-6_16.

[WW20] Hoeteck Wee and Daniel Wichs. *Candidate Obfuscation via Oblivious LWE Sampling*. Cryptology ePrint Archive, Report 2020/1042. https://eprint.iacr.org/2020/1042. 2020.

6.A Universal Delegation for Deterministic Computations

The following definition is taken from [Bit+16].

Let $\mathcal{L}_U$ be the universal language consisting of all tuples (M, x, t), such that machine M accepts x within time t. Let $\mathcal{L}_U(T)$ be the set of all pairs (M, x), such that $(M, x, T) \in \mathcal{L}_U$.

Let $T(\lambda) \in (2^{\omega(\log \lambda)}, 2^{\mathsf{poly}(\lambda)})$ be a computable superpolynomial function. A universal delegation system for $\mathbf{DTIME}(T)$ consists of three algorithms $(\mathsf{K}, \mathsf{P}, \mathsf{V})$, where

- The probabilistic generator K, given 1^λ, outputs a CRS $\mathsf{crs}_{\mathsf{DS}}$ and corresponding verification state τ. K is independent of any statement, proven later.
- The honest prover $\mathsf{P}(\mathsf{crs}_{\mathsf{DS}}; M, x)$ produces a certificate π that $(M, x) \in \mathcal{L}_U(T)$.
- The verifier $\mathsf{V}(\mathsf{crs}_{\mathsf{DS}}, \tau; M, x; \pi)$ verifies the validity of $(M, x) \in \mathcal{L}_U(T)$.

A universal delegation system $(\mathsf{K}, \mathsf{P}, \mathsf{V})$ for $\mathbf{DTIME}(T)$ is secure if it satisfies the following conditions:

Perfect completeness: for any λ, $(M, x) \in \mathcal{L}_u(T(\lambda))$, $(\mathsf{crs}_{\mathsf{DS}}, \tau) \leftarrow \mathsf{K}(1^\lambda)$, $\pi \leftarrow \mathsf{P}(\mathsf{crs}_{\mathsf{DS}}; M, x)$, it holds that $\mathsf{V}(\mathsf{crs}_{\mathsf{DS}}, \tau; M, x; \pi) = 1$.

Adaptive soundness for a bounded number of statements: there is a polynomial $\mathfrak{b}$, such that for any poly-size P^*, and any set of at most $2^{\mathfrak{b}(\lambda)}$ false statements $S \subseteq \{0,1\}^{\mathsf{poly}(\lambda)} \setminus \mathcal{L}_U(T(\lambda))$,

$$\Pr\left[\begin{array}{l}(\mathsf{crs}_{\mathsf{DS}}, \tau) \leftarrow \mathsf{K}(1^\lambda), (M, x, \pi) \leftarrow \mathsf{P}^*(\mathsf{crs}_{\mathsf{DS}}) \text{ s.t. } (M, x) \in S : \\ \mathsf{V}(\mathsf{crs}_{\mathsf{DS}}, \tau; M, x; \pi) = 1\end{array}\right] \leq \mathsf{negl}(\lambda) \ .$$

Efficiency: there exists a polynomial p, such that for every λ, $t \leq T(\lambda)$, and $(M, x) \in \mathcal{L}_U(t)$,

- K runs in time $p(\lambda)$,
- V runs in time $p(\lambda + |M| + |x|)$,
- P runs in time $p(\lambda + |M| + |x| + t)$.

The scheme is *publicly verifiable* if soundness is maintained when the malicious prover is also given the verification state τ. In this case, we will assume that τ appears in the clear in the reference string $\mathsf{crs}_{\mathsf{DS}}$.

For example, [KPY19] is a delegation scheme based on falsifiable assumptions.

$\mathsf{com}_\lambda(b) \quad /\!/ \quad b \in \{0,1\}$	$\mathsf{ComV}_\lambda(c)$
$r \leftarrow\!\!\$\, \{0,1\}^{2\ell}$; **return** $(\mathsf{g}'_\lambda(r), B(r) \oplus b)$	**parse** $c = (\widehat{y}, \widehat{b})$; **return** $\widehat{y} \in \mathrm{im}(\mathsf{g}'_\lambda) \wedge \widehat{b} \in \{0,1\}$

Figure 6.10: Keyless extractable bit commitment.

6.B Keyless Non-Black-Box Extractable Commitments

Inspired by Canetti and Dakdouk [CD09], we show how to construct a non-interactive keyless extractable commitment from an injective (verifiably-)extractable keyless OWF g_λ. Clearly, black-box extraction is impossible since there is no secret key for the extractor to use. Thus, we consider a non-black-box extractability definition where the extractor can depend on the adversary.

Let us observe that an EOWF is essentially a trapdoor function where the trapdoor is the adversary's code and the auxiliary input is aux. There are well-known constructions for obtaining public key encryption from any trapdoor function and a hard-core predicate. We are going to use a similar approach to obtain a keyless extractable commitment from a (verifiably-)extractable OWF and a hard-core predicate. In particular, if the EOWF is verifiable, then the commitment will have an efficient verification algorithm ComV_λ. We recall the definition of a hard-core predicate of a one-way function.

Definition 6.B.1. We say that an OWF $\mathsf{g}\colon X \to Y$ has a hard-core predicate $B\colon X \to \{0,1\}$, if for any PPT adversary $\mathcal{A}$, $\varepsilon_0 \approx_\lambda \varepsilon_1$, where

$$\varepsilon_b := \Pr[r \leftarrow\!\!\$\, X, u_0 = B(r), u_1 \leftarrow\!\!\$\, \{0,1\} : \mathcal{A}(\mathsf{g}(r), u_b) = 1] \ .$$

Let $\mathsf{g}_\lambda\colon \{0,1\}^\ell \to \{0,1\}^{\ell'}$ be a verifiably-extractable injective OWF. (See Section 6.4 for examples of such functions.) Using the Goldreich–Levin construction [GL89], we can obtain a verifiably-extractable OWF g'_λ with a hard-core predicate B. The Goldreich–Levin construction defines $\mathsf{g}'_\lambda(x,y) := (\mathsf{g}_\lambda(x), y)$ where $|x| = |y| = \ell$ and $B(x,y) = \sum_{i=1}^{\ell} x_i \cdot y_i \bmod 2$. Note that g'_λ is still a verifiably-extractable OWF since x can be extracted from $\mathsf{g}_\lambda(x)$ and y is public. We use g'_λ and B to construct a simple non-black-box extractable bit commitment scheme that is shown in Figure 6.10.

Theorem 6.B.2. *The bit commitment scheme in Figure 6.10 has the following properties:*

1. *If g'_λ is injective, then it is perfectly binding.*
2. *If g'_λ is an OWF and B is its hard-core predicate, then g'_λ is computationally hiding.*
3. *If g'_λ is verifiably-extractable with respect to an auxiliary distribution $\mathcal{D}$ and injective, then it is non-black-box extractable with respect to the same distribution.*

Proof. **Perfect Binding:** Suppose an adversary outputs two valid openings (r_1, b_1) and (r_2, b_2) to the same commitment, i.e., $(\mathsf{g}'_\lambda(r_1); B(r_1) \oplus b_1) = (\mathsf{g}'_\lambda(r_2); B(r_2) \oplus b_2)$. Since g'_λ is injective, then $\mathsf{g}'_\lambda(r_1) = \mathsf{g}'_\lambda(r_2)$ implies $r_1 = r_2$. Then $B(r_1) = B(r_2)$ so $b_1 = b_2$.

Computational hiding: Let $\mathcal{A}$ be a PPT adversary that tries to break computational hiding. Suppose $\mathcal{A}$ outputs messages $m_0, m_1 \in \{0,1\}$. Let ε_i denote the probability that $\mathcal{A}$ wins in Game i.

Game 0: This is the original computational hiding game where $c = \mathsf{com}(m_0; r) = (\mathsf{g}'_\lambda(r); m_0 \oplus B(r))$ for $r \leftarrow\!\!\$\, \{0,1\}^{2\ell}$.

Game 1: Now instead of using $B(r)$, we sample a random bit $u \leftarrow\!\!\$\, \{0,1\}$ and compute the commitment as $c = (\mathsf{g}'_\lambda(r); m_0 \oplus u)$. Clearly, $|\varepsilon_0 - \varepsilon_1| \leq \varepsilon_{hc}$ where ε_{hc} denotes the advantage a PPT adversary has in distinguishing $B(r)$ from a uniform bit.

Game 2: We change the previous game and commit to m_1 instead, that is, $c = (\mathsf{g}'_\lambda(r); m_1 \oplus u)$. Since u is uniformly random, we have that $\varepsilon_1 = \varepsilon_2$.

Game 3: We change c back to a real commitment, i.e., $c = (\mathsf{g}'_\lambda(r); m_1 \oplus B(r))$. Clearly again, $|\varepsilon_2 - \varepsilon_3| \leq \varepsilon_{hc}$.

Using the triangle inequality, we get that the advantage $\mathcal{A}$ has in breaking computational hiding is bounded by $2\varepsilon_{hc} \approx_\lambda 0$.

Non-black-box extractability: Let $\mathcal{A}(1^\lambda, \mathsf{aux})$ be a PPT adversary that outputs a commitment $c = (\widehat{y}, \widehat{b})$ such that $\mathsf{ComV}_\lambda(c) = 1$. Then $\widehat{y} \in \mathrm{im}(\mathsf{g}'_\lambda)$ and $\widehat{b} \in \{0,1\}$. According to the definition of verifiable-extractability, there exists an extractor $\mathsf{Ext}_{\mathcal{A}}^{\mathsf{g}'_\lambda}$ that given $\mathcal{A}$'s auxiliary input aux, outputs r such that $\widehat{y} = \mathsf{g}'_\lambda(r)$ with overwhelming probability. Thus we can construct an extractor $\mathsf{Ext}_{\mathcal{A}}(1^\lambda, \mathsf{aux})$ that runs $\mathsf{Ext}_{\mathcal{A}}^{\mathsf{g}'_\lambda}(1^\lambda, \mathsf{aux})$ to recover a unique r, then returns $\widehat{b} \oplus B(r)$. Extractor $\mathsf{Ext}_{\mathcal{A}}$ succeeds with the same probability as $\mathsf{Ext}_{\mathcal{A}}^{\mathsf{g}'_\lambda}$. □

Remark. Note that if we have the usual notion of extractability, then ComV_λ might be inefficient.

The bit commitment scheme above can be extended to arbitrary-length messages by simply committing to each bit of the message. However, this would be very inefficient. Instead of the hard-core predicate B one can use a hard-core function f, i.e., a function which produces t hard-core bits and thus allows to commit to messages of length t. Goldreich and Levin proposed the following construction. Let $\mathsf{g}_\lambda \colon \{0,1\}^\ell \to \{0,1\}^{\ell'}$ be a one-way function, then so is $\mathsf{g}^*_\lambda(x, y) := (\mathsf{g}(x), y)$ where $|y| = \ell + t(\ell) - 1$ and $t := t(\ell) := \min\{\ell, c \cdot \lceil \log_2 \ell \rceil\}$ for any constant $c > 0$. In general, then $t(\ell) = \mathcal{O}\log_2 \ell$. They define the hard-core function as $f(x, y) := (b_1, \ldots, b_t)$ where $b_i = B(x, (y_i, \ldots, y_{i+\ell-1})) = \sum_{j=1}^{\ell} x_j y_{i+j-1} \bmod 2$. The construction of the multi-bit commitment scheme is given in Figure 6.11. The security proof is similar to the bit commitment case.

Theorem 6.B.3. *The commitment scheme in Figure 6.11 has the following properties:*

1. *If g^*_λ is injective, then it is perfectly binding.*

$\mathsf{com}(m)$ // $m \in \{0,1\}^{tk}$	$\mathsf{ComV}(c)$
parse $m = m_1 \| \ldots \| m_k$ // $m_i \in \{0,1\}^t$; **for** $i = 1, \ldots, k$: $r_i \leftarrow\$ \{0,1\}^{2\ell+t-1}$; $c_i = (\mathsf{g}^*_\lambda(r_i), f(r_i) \oplus m_i)$; **return** $(c_1, \ldots, c_k)$;	**parse** $c = (c_1, \ldots, c_k)$ **for** $i \in [1,k]$ **parse** $c_i = (\widehat{y}_i, \widehat{s}_i)$; **if** $\widehat{y}_i \notin \mathrm{im}(\mathsf{g}^*_\lambda) \vee \widehat{s}_i \notin \{0,1\}^t$ **then** **return** 0 **else return** 1

Figure 6.11: Keyless extractable commitment for bit strings.

2. *If g^*_λ is a OWF and f is its hard-core function, then it is computationally hiding.*
3. *If g^*_λ is verifiably-extractable respect to an auxiliary distribution $\mathcal{D}$ and injective, then it is non-black-box extractable respect to the same distribution*

6.C ABLZ EOWF

We recall that the ABLZ EOWF (based on [Abd+17]) is defined as $\mathsf{g}_\mathsf{p}(x) := ([x]_1, [x]_2)$, where p is an asymmetric bilinear group (esp. we assume that $([1]_1, [1]_2) \in \mathsf{p}$), and we recall the definitions of the SDL and BDH-KE assumptions.

The one-way property of the ABLZ EOWF relies on the *Symmetric Discrete Logarithm (SDL) [Bic+10] assumption* which holds relative to Pgen, if for any PPT $\mathcal{A}$,

$$\Pr\left[\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); x \leftarrow\$ \mathbb{Z}_p : \mathcal{A}(\mathsf{p}, [x]_1, [x]_2) = x\right] \leq \mathsf{negl}(\lambda) \ .$$

The *BDH-KE* assumption [Abd+17] states that for any PPT adversary $\mathcal{A}$, there exists a PPT extractor $\mathsf{Ext}_\mathcal{A}$, such that

$$\Pr\begin{bmatrix}\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); r \leftarrow \mathsf{RND}_\lambda(\mathcal{A}); ([a]_1, [a']_2) \leftarrow \mathcal{A}(\mathsf{p}; r);\\ a'' \leftarrow \mathsf{Ext}_\mathcal{A}(\mathsf{p}; r) : [a]_1 \bullet [1]_2 = [1]_1 \bullet [a']_2 \wedge a \neq a''\end{bmatrix} \leq \mathsf{negl}(\lambda) \ .$$

Lemma 6.C.1. *Let $\mathcal{G} = \{\mathsf{g}_\mathsf{p}\}_{\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda), \lambda \in \mathbb{N}}$ be the ABLZ EOWF. The family $\mathcal{G}$ is a VEOWF.*

Proof. **One-wayness:** Assume that $\mathcal{B}$ is an adversary that given $([x]_1, [x]_2)$, for a random $x \leftarrow\$ \mathbb{Z}_p$, outputs x with non-negligible probability. Then $\mathcal{B}$ breaks the SDL assumption.

Verifiable-extractability with respect to aux of length $\mathfrak{b}(\lambda)$: Let $\mathcal{A}_{ve}$ be an adversary that breaks verifiable-extractability with some probability upper-bounded by ε_{ve}. We use security of the BDH-KE assumption to show that ε_{ve} is negligible. Let ε_{bdh} be the upper-bound for any PPT adversary $\mathcal{A}_{bdh}$ (that does not take any advice) to break the BDH-KE assumption. That is, an extractor $\mathsf{Ext}_{bdh}(\mathsf{p})$ fails with probability at most ε_{bdh}. Assume $\mathcal{A}_{bdh}$ proceeds as follows.

First it samples the auxiliary input $\mathsf{aux} \leftarrow \mathcal{D}_\lambda$ then runs $\mathcal{A}_{ve}(\mathsf{p}, \mathsf{aux})$ and gets $([x]_1, [x]_2)$ which it outputs. Since with probability ε_{ve} no extractor can reveal x we get that $\varepsilon_{bdh} \geq \varepsilon_{ve}$. Since ε_{ve} is non-negligible, ε_{bdh} is also non-negligible. □

Remark (Securing BDH-KE assumption against malicious aux). The limiting restriction on the adversary's auxiliary input can be lifted if we assume that it is set prior to the choice of the groups generators. That is, first the adversary gets its advice, then the generators are picked. Intuitively, in this setting no auxiliary input could given an advantage in breaking the assumption, even if $\mathcal{A}$ gets $\mathbf{g}^a \in \mathbb{G}_1, \mathbf{h}^a \in \mathbb{G}_2$. Since the parameters are fixed afterwards and group generators are picked randomly we have $[1]_1 = \mathbf{g}^\alpha, [1]_2 = \mathbf{h}^\beta$, for some random α, β; furthermore $\mathbf{g}^a = [a/\alpha]_1$ and $\mathbf{h}^a = [a/\beta]_2$. Thus with overwhelming probability, $[a/\alpha]_1 \bullet [1]_2 \neq [1]_1 \bullet [a/\beta]_2$.

Note that the generic (bilinear) group model (G(B)GM) allows for such a setup. As proposed by Maurer in [Mau05], in G(B)GM an adversary $\mathcal{A}$ is given access to an oracle O that performs group operations on behalf of $\mathcal{A}$, who does not see any information about the structure nor binary representation of group elements at all. That is, the only information $\mathcal{A}$ gets about the group elements is in which cells of memory of O the group elements are stored. Unfortunately, in real life assuming that the adversary knows nothing about (say) a group generator before the protocol starts may be unreasonable.

From ABLZ to a more general class of functions. In general, one can replace the ABLZ function $\mathcal{G}$ with any similar function from integers to group elements, where one can argue in the generic group model extractability, and where one can efficiently verify whether some element belongs to the image of $\mathcal{G}$ or not. In the pairing-based setting, similar functions have been studied in say [Ben+15] (in the context of generating SNARK CRSs by using MPC; however, here extractability is not needed) and [Abd+17] (in the context of Sub-ZK SNARKs).

One can have $\mathcal{G} = \{\mathsf{g}_\mathsf{e}\}_\lambda$ from the following class $\mathsf{BDHClass}$ of function families, where x_i are secret trapdoors and $\mathsf{e} = \mathsf{p}$. Here, we require that

$$\mathsf{g}_\mathsf{e}(\boldsymbol{x}) = ([\varrho(\boldsymbol{x}) : \varrho \in \mathcal{R}]_1, [\sigma(\boldsymbol{x}) : \sigma \in \mathcal{S}]_2, [\tau(\boldsymbol{x}) : \tau \in \mathcal{T}]_T)$$

for three sets of polynomials $\mathcal{R}$, $\mathcal{S}$, $\mathcal{T}$ from $\mathbb{Z}_p[\boldsymbol{X}]$, such that

(i) $X_i \in \mathcal{R} \cap \mathcal{S}$ for each i,

(ii) If $f(\boldsymbol{X}) \in \mathcal{R} \cup \mathcal{S} \cup \mathcal{T}$ has degree $d > 0$, then there exist polynomials ϱ, σ of degree $< d$, such that $f(\boldsymbol{X}) = \varrho(\boldsymbol{X})\sigma(\boldsymbol{X})$ and $\varrho(\boldsymbol{X}) \in \mathcal{R}, \sigma(\boldsymbol{X}) \in \mathcal{S}$.

Here, Item i guarantees that one can use the BDH-KE assumption for extractability, and Item ii guarantees verifiability. One can replace Item i with some other bootstrapping requirement if one wishes to use a different knowledge assumption. Most generally, one can omit Item i and rely on a tautological knowledge assumption (that then has to be proven secure in the generic group model).

Graphic design: Communication Division, UiB / Print: Skipnes Kommunikasjon AS

uib.no

ISBN: 9788230851784 (print)
9788230855041 (PDF)