On a New, Efficient Framework for Falsifiable Non-interactive Zero-Knowledge Arguments

Roberto Parisella

Thesis for the degree of Philosophiae Doctor (PhD)
University of Bergen, Norway
2023

UNIVERSITY OF BERGEN

On a New, Efficient Framework for Falsifiable Non-interactive Zero-Knowledge Arguments

Roberto Parisella

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

Date of defense: 09.06.2023

Year: 2023

Title: On a New, Efficient Framework for Falsifiable Non-interactive Zero-Knowledge Arguments

Name: Roberto Parisella

Print: Skipnes Kommunikasjon / University of Bergen

Acknowledgments

A big and heartfelt thank you goes to Helger Lipmaa, my main supervisor. Not only has he been a great academic mentor, providing guidance and support while always encouraging my independence. He has also been a great friend, and I am confident he will be in the future. I am also grateful to my other supervisor Øyvind Ytrehus, for all his help.

I want to thank all my co-authors: Arne Tobias Ødegaard, Geoffroy Couteau, Chaya Ganesh, and Hamidreza Khoshakhlagh. It was a great pleasure for me to collaborate with them; I feel grateful for everything I have learned from them and for all the inspiring conversations we had.

A special thank goes to all the people in Simula UiB for creating and maintaining a lovely working environment where I felt welcomed and appreciated more than anywhere else. Particularly, I want to thank Arne Tobias Ødegaard, Martha Norberg Hovd, and Janno Siim for reading my work and providing many helpful comments.

I also want to thank my family, who supported me, and my education, even if it took a bit longer than expected.

Lastly, the biggest thank you goes to my wife, Maryla, for turning my life upside down in a way I could not even imagine. She joined me during this journey with enthralling enthusiasm. She always supported and encouraged me to pursue my dreams and aspirations, showing me how proud and happy she was with my successes.

Abstract

A zero-knowledge proof is a protocol between a prover, and a verifier. The prover aims to convince the verifier of the truth of some statement, such as possessing credentials for a valid credit card, without revealing any private information, such as the credentials themselves. In many applications, it is desirable to use NIZKs (*Non-Interactive Zero-Knowledge*) proofs, where the prover sends outputs only a single message that can be verified by many verifiers.

As a drawback, secure NIZKs for non-trivial languages can only exist in the presence of a trusted third party that computes a common reference string and makes it available to both the prover and verifier. When no such party exists, one sometimes relies on non-interactive witness indistinguishability (NIWI), a weaker notion of privacy. The study of efficient and secure NIZKs is a crucial part of cryptography that has been thriving recently due to blockchain applications.

In the first paper, we construct a new NIZK for the language of common zeros of a finite set of polynomials over a finite field. We demonstrate its usefulness by giving a large number of example applications. Notably, it is possible to go from a high-level language description to the definition of the NIZK almost automatically, lessening the need for dedicated cryptographic expertise. In the second paper, we construct a NIWI using a new compiler. We explore the notion of *Knowledge Soundness* (a security notion stronger than soundness) of some NIZK constructions. In the third paper, we extended the first paper's work by constructing a new set (non-)membership NIZK that allows us to prove that an element belongs or does not belong to the given set.

Many new constructions have better efficiency compared to already-known constructions.

Sammendrag

Et kunnskapsløst bevis er en protokoll mellom en bevisfører og en attestant. Bevisføreren har som mål å overbevise attestanten om at visse utsagn er korrekte, som besittelse av kortnummeret til et gyldig kredittkort, uten å avsløre noen private opplysninger, som for eksempel kortnummeret selv. I mange anvendelser er det ønskelig å bruke IIK-bevis (*Ikke-interaktive kunnskapsløse bevis*), der bevisføreren produserer kun en enkelt melding som kan bekreftes av mange attestanter.

En ulempe er at sikre IIK-bevis for ikke-trivielle språk kun kan eksistere ved tilstedeværelsen av en pålitelig tredjepart som beregner en felles referansestreng som blir gjort tilgjengelig for både bevisføreren og attestanten. Når ingen slik part eksisterer liter man av og til på ikke-interaktiv vitne-uskillbarhet, en svakere form for personvern. Studiet av effektive og sikre IIK-bevis er en kritisk del av kryptografi som har blomstret opp i det siste grunnet anvendelser i blokkjeder.

I den første artikkelen konstruerer vi et nytt IIK-bevis for språkene som består av alle felles nullpunkter for en endelig mengde polynomer over en endelig kropp. Vi demonstrerer nytteverdien av beviset ved flerfoldige eksempler på anvendelser. Særlig verdt å merke seg er at det er mulig å gå nesten automatisk fra en beskrivelse av et språk på et høyt nivå til definisjonen av IIK-beviset, som minsker behovet for dedikert kryptografisk eskpertise. I den andre artikkelen konstruerer vi et IIV-bevis ved å bruke en ny kompilator. Vi utforsker begrepet *Kunnskapslydighet* (et sterkere sikkerhetsbegrep enn lydighet) for noen konstruksjoner av IIK-bevis. I den tredje artikkelen utvider vi arbeidet fra den første artikkelen ved å konstruere et nytt IIK-bevis for mengde-medlemskap som lar oss bevise at et element ligger, eller ikke ligger, i den gitte mengden.

Flere nye konstruksjoner har bedre effektivitet sammenlignet med allerede kjente konstruksjoner.

List of Publications

Efficient NIZKs for Algebraic Sets

Geoffroy Couteau, Helger Lipmaa, Roberto Parisella, and Arne Tobias Ødegaard. In *ASIACRYPT 2021, Part III.* ed. by Mehdi Tibouchi and Huaxiong Wang. Vol. 13092. LNCS. Springer, Heidelberg, Dec. 2021, pp. 128–158.

`https://doi.org/10.1007/978-3-030-92078-4_5`

NIWI and New Notions of Extraction for Algebraic Languages

Chaya Ganesh, Hamidreza Khoshakhlagh and Roberto Parisella.

In *International Conference on Security and Cryptography for Networks, SCN 2022* ed. by Clemente Galbi and Stanislaw Jarecki. Lecture Notes in Computer Science, vol 13409. Springer, Cham.

`https://doi.org/10.1007/978-3-031-14791-3_30`

Set (Non-)Membership NIZKs from Determinantal Accumulators

Helger Lipmaa and Roberto Parisella.

Cryptology ePrint Archive 2022, Paper Report 2022/1570,

`https://eprint.iacr.org/2022/1570`

Contents

Chapter 1

Introduction

1.1 Cryptographic Primitives

In the last few decades, we have seen exponential growth in digital communication. We now live in a time where digital services have replaced many in-person interactions. For instance, today it is common practice to handle even bureaucratic issues online, while the internet was just a resource for the military and academia some decades ago. Alternatively, we can compare how often our credit card data are used online to how often they were used for the same purpose at the beginning of this century. Consequently, a massive amount of sensitive data is now exchanged and stored online. It should not come as a surprise then that the need for efficient ways to communicate privately and confidentially has become more and more crucial over the past few years.

Cryptography is what should come to mind when it comes to privacy and confidentiality. When the average person thinks about cryptography, they most likely think of the classical problem of two parties that want to achieve confidentiality in their communication against a (possibly malicious) eavesdropping third party. Traditionally in the cryptographic research literature, the two parties that want to communicate are identified as Alice and Bob.

It is known how to achieve such private communication, supposing that Alice and Bob agreed on a secret key representing secret information in advance. Alice can then use a publically known encryption algorithm to encrypt a message, having the secret key as additional input, and send the ciphertext to Bob. Bob can use another publically known algorithm to decrypt the ciphertext and recover the message. Only Bob, who knows the secret key Alice used to encrypt, can correctly compute the original message from the ciphertext. Cryptographic primitives with the functionality we have just described are called *symmetric encryption schemes*. In the context of cryptography, the word symmetric indicates that the same secret key is shared among parties.

Thanks to decades of formalization of cryptography and, more generally, computer science, we now have several cryptographic primitives to ensure confidentiality in many scenarios. For instance, we can think of public-key encryption schemes: cryptographic primitives that aim to solve the problem of confidential communication between two parties that do not share a secret key. In this setting, Bob can generate a pair of correlated public and secret keys and publish the first while keeping the second for himself. If Alice wants to send a confidential message to Bob, she can encrypt the message using his public key and send the ciphertext to Bob. Only Bob, who knows the secret key, can correctly decrypt the ciphertext and read the message. In other words, instead of achieving confidentiality under the assumption that the parties can share a secret in advance, we achieve it assuming that it is impossible to recover the secret key from the corresponding public key. Diffie and Hellman [DH76] proposed the public-key paradigm in 1976, revolutionizing the entire field of cryptography. If one thinks about how often we need to communicate privately with a party we have had no real-life contact with, it is easy to argue how important it is to develop efficient and secure public-key encryption schemes.

Today cryptography is used to achieve privacy and confidentiality in a wide range of scenarios. The current work is focused on the cryptographic primitive known as *zero-knowledge proof*. A zero-knowledge proof is used in the following scenario. Suppose Alice knows a secret and Bob needs to be sure that she indeed knows the secret. However, Alice does not want to reveal the secret; she only needs to convince Bob that she knows it. In other words, Alice wants to show knowledge about a specific statement without revealing this knowledge. For example, Alice wants to show that she is registered in a list that grants certain privileges, without revealing which pseudonym in the list corresponds to her own. Of course, this would be trivial if Alice revealed her pseudonym to Bob: he can just check if Alice's pseudonym is on the list. However, in this case, Bob would receive Alice's pseudonym, compromising her privacy. While in reality, for many applications, it would be sufficient for Bob to know merely that Alice's pseudonym is on the list.

1.2 Zero-knowledge Protocols

Let us now recall some useful basic mathematical concepts. Given two sets A and B, a binary *relation* $\mathscr{R}$ is a subset of the cartesian product $A \times B$. Since we only deal with binary relations in the current thesis, we omit the term binary. Elements in A are called *statements*. For each statement $\mathtt{x} \in A$, a *witness* for $\mathtt{x}$ is an element $\mathtt{w} \in B$, such that $(\mathtt{x},\mathtt{w}) \in \mathscr{R}$. We can naturally associate a *language* $\mathscr{L}_{\mathscr{R}}$ to each relation as the set of elements in A for which a witness exists.

$$\mathscr{L}_{\mathscr{R}} = \{\mathtt{x} \in A : \exists \mathtt{w} \in B, (\mathtt{x},\mathtt{w}) \in \mathscr{R}\}.$$

Elements in a language are called *true statements*, and all other statements are called *false statements*. We call PPT (Probabilistic Polynomial-Time) Turing machines that run in time bounded by a polynomial function in the input length and are allowed to use random coins to perform their computation. In the context of this thesis, we say informally that a language is *hard*, if no PPT can compute a witness, given a statement. An NP-complete language is a language such that: (i) there exists a deterministic polynomial-time Turing machine that on input $(\mathtt{x},\mathtt{w})$ outputs 1 if and only if $(\mathtt{x},\mathtt{w}) \in \mathscr{R}$ and (ii) if a procedure to decide if $\mathtt{x} \in \mathscr{L}_{\mathscr{R}}$ exists, then it could be used to decide membership of every language for which (i) holds. NP stands for Non-deterministic Polynomial-time, and it means that there exists a non-deterministic Turing machine, with running time bounded by a polynomial function in the input length, that can decide membership of $\mathscr{L}_{\mathscr{R}}$. Intuitively, this non-deterministic Turing machine is defined by trying all the possible $\mathtt{w}$ involved in (i). In contrast, languages in P, are languages that a deterministic, polynomial-time Turing machine can decide. Whether $\mathsf{P} \neq \mathsf{NP}$ is one of the most important open problems in mathematics; it is one of the Millennium Prize problems. It is common practice in theoretical computer science to conjecture that indeed $\mathsf{NP} \neq \mathsf{P}$, despite the lack of a proof. As an example, a classic NP-complete problem consists of assigning a colour, amongst three, to each vertex of a given graph, such that two adjacent vertices do not have the same colour. We can define a relation considering a graph as a statement and a three-colourability over the graph as a witness.

Zero-knowledge proofs were defined in 1985 by Goldwasser, Micali and Rackoff [GMR85]. As an acknowledgement of the importance of their foundational work, they got awarded the Gödel prize in 1993. In a zero-knowledge proof, a statement is given as a public input to two parties, the *prover* and the *verifier*. As additional secret input, the prover receives a witness proving that the statement is in a certain language: typically, a hard language. Informally, the prover aims to convince the verifier that the statement is true, without revealing any additional information about the witness. Thus, the two parties start exchanging messages in accordance with what the protocol prescribes. At the end of the interaction, the verifier performs additional computation, having the set of exchanged messages—also called *transcript* or *proof*—as additional input. According to the result of this computation, the verifier either outputs *accept* or *reject*, and it outputs accept if and only if it is convinced that a witness for the statement exists.

We require zero-knowledge proofs to have three security properties.

Completeness: the honest prover always convinces the verifier.

Soundness: the verifier can always catch a malicious prover and outputs reject for every false statement.

Zero-knowledge: at the end of the interaction, a (potentially malicious) verifier learns nothing about the witness besides the fact that a witness exists for the public statement.

In principle, zero-knowledge proofs can be used to ensure the honesty of parties involved in protocols without compromising their privacy. One can achieve the previous task, requiring any party to send a zero-knowledge proof of the correctness of its computation, along with any messages. By the completeness and soundness properties of the proof, we know that the execution has been done correctly. Concurrently, the zero-knowledge property guarantees no private information is leaked during the protocols' execution. For example, zero-knowledge proofs are necessary for electronic voting systems, where they are used to check that a voter is registered as having the right to vote and to collect his vote anonymously. Zero-knowledge proofs are also a crucial building block of anonymous authentications when a party needs to authenticate itself without revealing its identity. As a third noteworthy example, zero-knowledge proofs are used in digital currencies, such as Zcash, and blockchains, such as Algorand, to carry out tasks involving proving properties on confidential data. For instance, zero-knowledge proofs are used to prove possession of enough unspent coins to complete a transaction without revealing other information, like the total amount of owned coins. Or they are used to achieve anonymity in a digital payment procedure, guaranteeing privacy about details such as addresses, transactions type and quantity, buyer and seller identities, and many more.

Many zero-knowledge proofs require online interaction between the parties. Interaction is a not desirable feature. In many applications, it is necessary to allow verification when the prover is offline and not available for interaction. For example, for electronic voting, it is crucial that any verifier can check the validity of the election, even long after it has taken place. Moreover, zero-knowledge protocols are used to define other cryptographic primitives, such as digital signatures. The need for interaction would severely compromise the usability of such schemes, sometimes irretrievably.

Blum, Feldman, and Micali [BFM88] introduced the concept of *non-interactive* zero-knowledge proofs, which are abbreviated as NIZKs. A NIZK prover computes the proof and sends it in one message to the verifier, who, without any interaction, only has to output accept or reject. The first NIZK constructions were defined in the so-called *CRS model*. That is, we suppose the existence of a trusted third party that computes a *common reference string* (CRS) and makes it available to both the prover and the verifier. In contrast, we refer to the *plain model* as the setting where no trusted setup is required. Unfortunately, Goldreich and Oren [GO94] showed that it is impossible to define NIZKs in the plain model, enjoying completeness, soundness, and zero-knowledge, for non-trivial languages. Thus, NIZKs are only defined in the CRS model.

1.3 Cryptographic Assumptions

As it is often the case in complexity theory, in cryptography, many results are proven under various cryptographic assumptions: unproven statements assumed to be true and used as a starting point for formal mathematical proofs. A cryptographic assumption is often a statement about the hardness of a certain problem. Throughout this thesis, we will state many different cryptographic assumptions. A different but closely related concept is a conjecture: a statement believed to be true amongst the vast majority of the scientific community, despite the lack of a formal proof. A well-established computer science practice is to take conjectures and state them as assumptions. Arguably the most remarkable example of this practice is the case of the “P versus NP” problem, mentioned in Section 1.2. The unproven statement that P is different from NP is widely used as an assumption in many branches of theoretical computer science. Proving the security of any cryptographic assumption formulated from one of these hardness conjectures would imply a major breakthrough in complexity theory. Therefore, it is not considered likely to happen soon. Since the number and nature of assumptions used in cryptography are vast and diverse, it should not be surprising that a significant and flourishing research direction is to study relations between these assumptions and establish a hierarchy of trust among them [Nao03, Pas13, GK16].

The first important distinction for cryptographic assumptions is between computational and non-computational. A *computational* assumption is an assumption whose validity can be verified using a Turing machine, traditionally called a challenger. More precisely, a computational assumption follows the following pattern: an *adversary*, defined as a Turing machine, with certain properties, that solves a given problem does not exist. Typically, the property we require from the adversaries is that they should be PPT. Computational assumptions are called in this way because verifying if one of them is false is a computational task. Suppose that we have an adversary that can efficiently solve a given problem. Then we can define a Turing machine, called *challenger*, which samples valid inputs, calls the adversary and finally checks if the solution is correct. Thus, one can exhibit an adversary that solves the given problem to show that a computational assumption is false. Anyone else can run the challenger to verify whether the adversary is successful. Note that the word computational has been used with different meanings in the context of cryptographic assumptions. In this thesis, we use the meaning we have just described, following how the word computational is used by [Nao03] in the context of cryptographic assumptions classification.

Among computational assumptions, the most desirable ones are those for which it is possible to define an efficient (PPT) challenger. Following Naor's classification [Nao03], we call the assumptions in this category *falsifiable* and any other (not necessarily computational) assumption *non-falsifiable*. An example of a falsifiable assumption is the hardness of computing the discrete logarithm in certain finite groups. Let g be a generator of an abelian group $\mathbb{G}$. Informally, the *discrete logarithm* assumption for the group $\mathbb{G}$ states that there does not exist a PPT adversary which computes x, on input g^x. It is possible to verify if the previous assumption is false with a PPT challenger defined as follows: sample a uniformly random group element, give it to a discrete logarithm adversary which returns an exponent, and verify if the exponent returned is the correct one. Another falsifiable assumption, widely used in cryptography, is the CDH-assumption (*Computational Diffie-Hellmann*) [den90], defined in Section 2.1.

Falsifiable assumptions in which adversaries must distinguish between two distributions are referred to as *decisional assumptions*[1]. An adversary of a decisional assumption receives inputs from one out of two distributions, and it must output a bit. We require the output to be consistent to which distribution the adversary receives the input from: it should output 0 if and only if it receives the input from the first distribution and 1 otherwise. A famous example of a decisional assumption is the DDH-assumption (*Decisional Diffie-Hellmann*), according to which no PPT adversaries can distinguish between the distributions (g^x, g^y, g^{xy}) and (g^x, g^y, g^z) where x, y, z are uniformly random exponents. The challenger of the DDH-assumption is defined as follows: (i) toss a coin $b \in \{0, 1\}$ and sample uniform x, y, z, (ii) compute b' with the candidate DDH adversary, on input (g^x, g^y, g^{xy+bz}), and (iii) check if $b = b'$.

[1] The word computational is widely used with the meaning of "non-decisional". We remark we are using the word computational here with a different meaning.

Let us describe *computational non-falsifiable* assumptions: computational assumptions with a challenger defined as a Turing machine that runs in superpolynomial time. A widely used assumption in this category is the one-more discrete logarithm assumption [BNPS03, PV05, BMV08, BFP21]. The *one-more discrete logarithm* assumption states that given $l+1$ group elements, it is hard to return the discrete logarithm of all of them, even if we have access to an oracle that can compute up to l discrete logarithms. The challenger for this assumption has to emulate the discrete logarithm oracle. Under the hardness of computing the discrete logarithm, it seems unlikely to find a polynomial time challenger for the one-more discrete logarithm assumption. In this class we also find many gap assumptions [OP01]. A *gap assumption* is an assumption of the form “a problem B is hard even if another problem A is easy”. Tipically, gap assumptions are formulated taking both A and B as supposedly hard problems. A gap assumption challenger has to emulate an oracle that solves “problem A”, resulting in a superpolynomial time challenger.

Because of the impossibility of running a superpolynomial time challenger in practice, assumptions in this class are less desirable than falsifiable assumptions. In fact, paraphrasing [Nao03] a security proof of a cryptographic primitive, under an assumption, can be seen as a proof for the statement “the construction is secure or the assumption is false”. If the assumption is falsifiable, it is easier to check the veracity of the second clause in the previous formula. For this reason, security proofs under falsifiable assumptions are considered a stronger security condition than proofs under non-falsifiable assumptions. When we want to prove the security of a cryptographic scheme, it is therefore better to do it under any falsifiable assumption. Thus, an important research direction is to define new cryptographic primitives and improve the efficiency and functionality of existing ones, with the constraint of proving security under falsifiable assumptions.

Moving toward less desirable assumptions, cryptographers commonly use *knowledge assumptions*, such as the *knowledge-of-exponent* assumption [Dam92, BP04]. A knowledge assumption is a non-computational assumption stating that if an efficient Turing machine can reliably compute a certain output from an input distribution, then it must know specific intermediate values. This knowledge requirement is formalized by postulating the existence of an efficient Turing machine called *extractor*. The extractor has to compute the intermediate value reliably. To check that a knowledge assumption is false, one must show an adversary for which each possible extractor fails to compute the intermediate value reliably. We refer to [BP04], for an extensive discussion on falsifying a knowledge assumption. However, since knowledge assumptions are much more difficult to falsify than any computational assumption [BP04, Nao03], they are considered even less desirable than any computational (possibly non-falsifiable) assumptions.

1.4 Idealized Models

Security proofs that rely only on assumptions and limit the adversaries' computational resources are referred to as proofs in the *standard model*. In many proofs, however, security is proven using *idealized models*. In cryptography, we talk about idealized models when a concrete primitive is replaced with an idealized version, in a security proof.

Probably, the most popular idealized model used in cryptography is the *random oracle model*, or ROM for short. In the ROM, any party has access to a black-box oracle that implements a truly uniformly random function. In other words, the random oracle is supposed to secretly sample a function from the uniform distribution in a finite function space and black-box evaluate the secret function for all the parties. ROM is widely used in the context of zero-knowledge [BR93].

Other idealized models are also used in zero-knowledge protocols security proofs, such as the GGM (*Generic Group Model*) [Sho97, Mau05]. In GGM we suppose the existence of a "perfect unstructured" cryptographic group, where group elements are perfectly random encodings of their exponent. Thus, adversaries can only perform generic group operations in GGM.

Fuchsbauer et al. proposed the AGM (*Algebraic Group Model*) [FKL18]: a different idealized way of modelling cryptographic groups. A PPT is said to be *algebraic* if it must know a representation of each group element it outputs in the form of a linear combination of the group elements it received as input. Initially, Fuchsbauer et al. required the algebraic PPT to output the coefficients of the linear representation as well, as additional output. A different formalization requires that for each algebraic adversary, an adversary-dependent PPT extractor exists that computes the coefficients of the linear representation. In this thesis, we use this second formalization. One can think of the AGM as a "generalized knowledge assumption", where we require adversaries to explain their computation through linear combinations of the inputs. For this reason, AGM extractors are adversary-dependent, and receive adversary's random coins as additional input: the same is true for extractors of knowledge assumptions. The AGM is considered more realistic and closer to the standard model than the GGM. This is because AGM adversaries receive proper group elements as input, not abstract labels, as in the case of GGM. Thus, they can see the group structure and try to exploit it. However, different from the standard model, AGM adversaries must explain their computation in terms of linear combinations of the inputs.

Idealized models have been very useful in proving the security of efficient and complex primitives. Moreover, no successful attacks have been found so far against concrete and used protocols with a security proof in an idealized model. Nevertheless, security proofs in idealized model should still be considered as heuristics. In fact, attacks on (contrived) artificial protocols, or un-instantiablity result [GK03, Den02, Zha22] have shown that proofs in idealized models do not translate automatically to proofs in the standard model.

Having security proofs in the standard model and under computational and preferably falsifiable assumptions is most desirable. However, proposing new computational (hopefully falsifiable) assumptions and proving the security of the assumption in an idealized model has become more and more popular, mostly as a way to prove security of more efficient primitives. Specifically, a new, "standard looking", falsifiable assumption, tailored to prove the security of a specific primitive can be defined. Then, a good validation for the new assumption is to reduce it to a more standard one, such as the hardness of discrete logarithm, relying on an idealized model.

1.5 Different Levels of Soundness

We can sometimes prove that valid proofs for false statements cannot exist. Or, equivalently, the soundness of some proof systems holds even against computationally unbounded adversaries. Following standard terminology, from now on, we refer to *zero-knowledge proofs* as proof systems with completeness, zero-knowledge, and soundness that holds against any (potentially unbounded adversaries). Zero-knowledge proof systems with soundness that holds only against PPT adversaries are called *zero-knowledge arguments*.

Moreover, the standard definition of soundness only guarantees that it is intractable to compute valid proofs for false statements. However, for some applications, a stronger definition of soundness is required.

Knowledge soundness: if a prover can compute a valid proof for a given statement, then it must know a valid witness.

We can, for instance, think of languages for which false statements do not exist: say that the statement $\mathtt{x}$ is an element of an abelian group $\mathbb{G}$, generated by g, and the witness an exponent $\mathtt{w}$, such that $g^{\mathtt{w}} = \mathtt{x}$.

There exist efficient NIZKs that are knowledge sound under knowledge assumptions or in idealized models. However, security proofs under knowledge assumptions or in idealized models are undesirable, as discussed in Sections 1.3 and 1.4. Alternatively, one can show knowledge soundness, augmenting the statement with a trapdoor for extraction. For instance, a well-known methodology consists of adding a public key of an encryption scheme to the CRS. Then, we let the prover add an encryption of the witness to the proof and prove that the ciphertext is computed correctly. Unfortunately, this approach irreparably compromises efficiency. Therefore, it is not preferred in practice.

A solution to define NIZK proofs (with knowledge soundness) in the standard model, under falsifiable assumptions, is to rely on a weaker definition of knowledge soundness, stating that the prover must only know a function of a valid witness. This property is called partial knowledge soundness [BCKL08].

However, no NIZK are known with: (i) knowledge soundness under falsifiable assumptions, in the standard model, and (ii) no efficiency loss compared with the state-of-the-art (not-knowledge) sound ones.

1.6 NIZKs in the Random Oracle Model

Fiat and Shamir [FS87] defined NIZKs from interactive zero-knowledge proofs, letting the prover generate verifier messages on its own through a cryptographic hash function, through a compiler called *Fiat-Shamir transform*. A *cryptographic hash function* is a function h, such that, given an element y in the image of h, it is hard to find an element x in the domain, such that $h(x) = y$. Alternatively, hash functions are defined by the following stronger property: it is hard to find $x_1 \neq x_2$, such that $h(x_1) = h(x_2)$. The Fiat-Shamir transform was the earliest technique proposed to define secure NIZKs. In terms of efficiency, Fiat-Shamir NIZKs have proof size as big as the prover's communication of the starting interactive proof. Moreover, the hash function evaluation is the only overhead in computational complexity introduced by the Fiat-Shamir compiler, over the complexity of the interactive protocol we started with. Fiat-Shamir NIZKs from very optimized interactive protocols are usually state-of-art in terms of efficiency for many scenarios, and improving on them is often challenging.

To define efficient NIZKs for a given language, a standard and successful pattern is to start designing a Sigma protocol for it [Sch90, CDS94, CCs08, Mau09, ACR21], and then apply the Fiat-Shamir transform. A Sigma protocol is a three-round, zero-knowledge proof, with security often based on mild falsifiable assumptions [Mau09] and verifier interaction consisting only of a uniformly random message sent to the prover in the second round; see Section 2.6 for a formal definition.

Fiat-Shamir NIZKs are only proven secure in the ROM, when we instantiate the cryptographic hash function with a random oracle. Moreover, it is proven that any concrete implementation of the hash function fails to instantiate secure Fiat-Shamir NIZKs unconditionally. Specifically, (contrived) secure interactive zero-knowledge protocols exist, for which the Fiat-Shamir transformation results in insecure NIZKs for any implementation of the hash function [GK03]. For this reason, Fiat-Shamir NIZKs that use concrete implementations for the hash function, such as SHA-256, are considered only heuristically secure.

Although no concrete attacks have been found on any non-contrived Fiat-Shamir NIZKs, it is desirable to define NIZKs secure under computational, and hopefully falsifiable, assumptions, in the standard model, without compromising on efficiency.

1.7 NIZKs in the Standard Model

Groth-Sahai NIZKs. In the seminal work [GS08], Groth and Sahai defined a new class of NIZKs whose security is based on trusted, falsifiable cryptographic assumptions, in the standard model. Their result was recognized from the very beginning as a breakthrough: they developed a framework to define falsifiable NIZKs for a large class of practical languages. Many improvements followed their initial result.

In part, the merit of the method is shown by a rich line of GS NIZKs for specific and interesting applications. Ghadafi et al. [GSW09] construct the first practical NIZKs for circuit satisfiability, based on falsifiable assumptions. Belenkiy et al. [BCKL08], Acar and Nguyen [AN11] and later Daza et al. [DGP$^+$19] define Groth-Sahai *set-membership proofs*, discussed in Section 1.11.

Furthermore, several results show application-independent, further optimizations for the original constructions in many interesting cases. Escala and Groth [EG14] show how to optimize the proof size and the prover computational complexity. Moreover, they show how to redefine Groth-Sahai NIZKs as commit-and-prove NIZKs by letting the prover choose its own CRS and prove that the chosen one still guarantees soundness. Rafols [Ràf15] define efficient Groth-Sahai NIZK arguments of partial satisfiability of sets of equations. Her technique, for instance, can be used to further reduce proof size of NIZKs defined in [CGS07] or [BCKL08].

The Groth-Sahai framework carries some built-in limitations that severely compromise its applicability for real-life applications. The first critical step is efficiency. Regarding both communication and computational complexity for the prover and verifier, a Groth-Sahai NIZK often has quite an efficiency gap compared to Fiat-Shamir NIZKs. In practice, this gap irreparably compromises its use. Moreover, designing and optimizing a Groth-Sahai NIZK for a specific application is an uphill task requiring considerable work by dedicated expertise. The Groth-Sahai framework is, at its core, a way to define a NIZK for a specific class of languages directly: languages generated by a set of PPEs (*Pairing-Product Equations*). Informally, we can think of pairing-product equations as quadratic equations.

The design of Groth-Sahai NIZK follows a precise pattern.

1. Find an efficient (possibly optimal) representation of a given problem as a set of PPEs.
2. Apply the Groth-Sahai framework and any relevant further optimization from follow-up works to prove in zero-knowledge the possessions of elements that satisfy the given set of PPEs.

Many technical and application-dependent choices have to be taken in each step.

Due to these drawbacks, Groth-Sahai NIZKs cannot be considered a satisfying end for the quest for efficient NIZKs, under falsifiable assumptions, in the standard model.

Provably secure Fiat-Shamir NIZKs. Recently, a new line of work showed how to instantiate provable-secure Fiat-Shamir NIZK in the standard model [KRR17, HL18, CCH$^+$19, CLW18, PS19, CPV20]. At their core, these NIZKs are defined using the Fiat-Shamir compiler with a concrete family of seeded hash functions that achieves the *correlation intractability* property. Intuitively, this property means that given a function h, for each relation $\mathscr{R}$ in a given class of relations, it is computationally intractable to find an input x, such that $(x, h(x)) \in \mathscr{R}$. It is known how to prove that many families of hash functions are correlation intractable for any computable relations under different falsifiable assumptions; the existence of a circular-secure fully homomorphic encryption [CLW18], and the learning with errors [PS19] being notable examples. NIZKs of this class have as good communication complexity as the original Fiat-Shamir NIZKs, and they are secure in the standard model under falsifiable assumptions. However, so far, NIZKs from this line of research are only of theoretical interest: in existing constructions, the complexity of evaluating the concrete hash function for both prover and verifier is a critical bottleneck. Correlation-intractable hash functions are only known from computationally expensive primitives [CLW18, PS19]. Thus, the resulting NIZK often does not meet the efficiency requirement for concrete applications, in term of computational complexity. A different issue is that we cannot unconditionally apply the Fiat-Shamir compiler with correlation-intractable hash functions to any Sigma protocol. Recall the impossibility result in [GK03], discussed in Section 1.6, stating that any concrete hash function fails to instantiate the Fiat-Shamir transform unconditionally. Thus, it is not surprising that the class of Sigma protocols must be restricted. A sufficient condition to define a secure NIZK, using correlation intractable hash functions, is that the Sigma protocols is a trapdoor Sigma protocols [CLW18]. [CPV20][2] show a compiler to define a trapdoor Sigma protocol from any Sigma protocols. Unfortunately, this compiler severely compromises even communication complexity.

1.8 Couteau-Hartmann NIZKs

Couteau and Hartmann [CH20] define a novel type of NIZKs. At a very high level, they define a NIZK by compiling a Sigma protocol over an abelian group $\mathbb{G}_1$ into a non-interactive zero-knowledge argument over bilinear groups, by embedding the second message e into a different abelian group $\mathbb{G}_2$ and adding the embedded challenge to the CRS. Security informally relies on the hypothesis that no efficient isomorphisms exist between the two groups $\mathbb{G}_1$ and $\mathbb{G}_2$. See Section 2.3, for the definition and a discussion on cryptographic bilinear groups. The soundness of the resulting NIZK is proven under a novel computational assumption.

Couteau-Hartmann NIZKs represented a significant advance towards practical NIZKs, under falsifiable assumptions. As shown by many tables in [CH20], they significantly improve efficiency over optimized Groth-Sahai constructions, for many interesting applications.

However, the Couteau-Hartmann framework also has some critical limitations. First, unlike the Fiat-Shamir one, their compiler defines secure NIZKs only if applied to one specific Sigma protocol. In particular, they started from a Sigma protocol for a specific class of languages: *algebraic languages*. These are languages of the form $\mathscr{L}_{\Gamma,\theta} = \{\mathtt{x} : \exists \mathtt{w}, \Gamma(\mathtt{x}) \cdot \mathtt{w} = \theta(\mathtt{x})\}$, where $\mathtt{x}$ is the input, $\mathtt{w}$ is the witness, Γ and θ are affine maps, such that $\theta(\mathtt{x})$ is a vector and $\Gamma(\mathtt{x})$ is a matrix. Thus, to define a Couteau-Hartmann NIZK for a given application, one must first find an efficient representation of a given problem as an algebraic language and then apply the compiler to the Sigma protocol for the found language. To go from the description of the algebraic language to the optimized Couteau-Hartmann NIZK is an automatic task. However, finding an efficient representation of a given problem as an algebraic language means finding suitable and possibly short Γ, θ. The latter task has to be manually performed every time. Having to manually find efficient language parameters is an undesirable feature, since it requires

[2] Although the author of this thesis is also a co-author of [CPV20], this article is not included in the thesis.

work from dedicated expertise, as already discussed in Section 1.7 for the Groth-Sahai framework. Moreover, it needs to be clarified how many languages admit an efficient representation as an algebraic language. To define efficient Couteau-Hartmann NIZKs, we must start with algebraic languages with small language parameters. Therefore, even though algebraic languages can be used to express all problems in NP, we are only interested in the cases where a problem can be expressed as an algebraic language with small language parameters. Because algebraic languages can only handle linear equations, the task of finding small parameters for algebraic languages seems to be even harder than finding an efficient representation in terms of PPEs (which can handle quadratic equations).

Another weakness of Couteau-Hartmann NIZKs is represented by their new computational assumption: the *extended kernel matrix Diffie-Hellman* ($\mathsf{ExtKerMDH}$) assumption. To motivate the introduction of the $\mathsf{ExtKerMDH}$ assumption, the authors reduce it to discrete logarithm in AGM [FKL18] and in GGM [Sho97, MW98]. As discussed in Section 1.4, introducing a new falsifiable assumption, providing proof of security in GGM or AGM, is a common practice. However, the $\mathsf{ExtKerMDH}$ assumption is only guaranteed to be falsifiable for some cases, while in general, it is a computational, non-falsifiable assumption, see Section 1.3. Couteau and Hartmann showed that the $\mathsf{ExtKerMDH}$ assumption is falsifiable for a restricted class of algebraic languages. As shown in [CH20], this class includes "disjunctions of linear languages", but it was not determined if any other languages of interest for applications are included. For the general case of security based on a non-falsifiable version of the $\mathsf{ExtKerMDH}$ assumption, a security proof in an idealized model is not considered a good enough motivation for introducing a new assumption.

Couteau and Hartmann's work left us with some important open problems, whose solution would lead to making the best out of the new compiler.

1. Is it possible to apply the Couteau-Hartmann compiler to different Sigma protocols? Hopefully, results in this direction would increase the class of languages efficiently and natively supported by this framework, resulting in the definition of practical NIZKs, secure in the standard model, for a broader class of applications.
2. Can we enlarge the class of languages for which we can define a Couteau-Hartmann NIZK with security under falsifiable assumptions? Alternatively, can we show better security conditions and motivations for the general case of the underlying assumption not being falsifiable?
3. Can we construct small language parameters in an (almost) automatic way, thus lessening the need for dedicated expertise?

1.9 Succinct Arguments of Knowledge

Zero-knowledge arguments are *succinct* if the proof size is sublinear in the witness size. Non-interactive succinct zero-knowledge arguments are called SNARGs (*Succinct Non-interactive ARGuments*). Since SNARGs offer the best performance in terms of proof size, they have become one of the most popular topics in cryptography [Gro10, BCCT12, Lip12a, GGPR13, PHGR13, Lip13, Gro16, BBB$^+$18, GWC19, RZ21, LSZ22]. Succinctness is a crucial requirement in applications where the size of the relations is already much bigger than what can be afforded in terms of communication complexity. For instance, we can think about the relations we deal with in blockchains, cryptocurrencies, or verifiable outsourced computations. Due to the topic's popularity, we now have many different constructions of SNARGs. Therefore, to talk about them as a whole category is a challenging task that carries inevitable generalizations and is out of the scope of the current work. Here, we only mention why we should not compare them to non-succinct NIZKs.

The most significant difference separating SNARGs from non-succinct NIZKs is that the formers' security cannot rely on falsifiable assumptions. Almost every work on succinct zero-knowledge quotes the famous impossibility result proven by Gentry and Wichs, stating that it is impossible to reduce (adaptive) soundness of SNARGs for hard languages to a falsifiable assumption [GW11][3]. Here adaptive soundness means that a malicious prover can first see the generated CRS, and then try to compute a valid proof for an adaptively chosen false statement. The impossibility result in [GW11] alone should be considered enough to separate the study of succinct and non-succinct zero-knowledge. In reality, succinct and non-succinct NIZKs are separated by a wider gap regarding the category of the assumptions. In fact, no (adaptively sound) SNARGs are known even with security under non-falsifiable, yet still computational, assumptions, and this task is still an exciting open question. We know how to prove the soundness of SNARGs only relying on non-computational knowledge assumptions, or in idealized models.

1.10 NIWI in the Plain Model

As stated above, the assumption of a trusted third party that securely computes the CRS is necessary to define sound NIZKs for hard languages. However, it is possible to de-

[3] Gentry-Wichs impossibility is here stated very informally. We refer the reader to [GW11] for precise information.

fine non-interactive sound proofs, without the requirement of a trusted setup, as long as we give up on zero-knowledge, and rely on a weaker notion of privacy. Informally, a protocol is *witness indistinguishable* if it is hard for any (potentially malicious) verifier to distinguish between provers that use different witnesses for any given statement. We call protocols with this property NIWI (*Non-Interactive Witness Indistinguishability*) proofs in the plain model, recalling that in Section 1.2 we used the expression "in the plain model" to indicate that no trusted setup is required. For many interesting applications, witness indistinguishability is a natural requirement. We can, for instance, think of languages of the type "either one knows a trapdoor or a witness for a statement". Algebraic languages, Sections 1.8 and 2.8, and languages defined by PPEs, Section 1.7, are also valid examples.

A general idea to define NIWIs in the plain model is to start from NIZKs that are perfectly sound for some CRS choices. Specifically, the idea is to let the prover choose many CRSs by itself, with the restriction that at least one ensures perfect soundness. The prover will compute a proof for each of the chosen CRSs and send all the pairs of CRS and proof to the verifier. Then, we must also equip the verifier with a polynomial-time, decision algorithm to check that the prover chooses the CRSs, in such a way that perfect soundness holds for at least one of them. Since, here and in the rest of this work, we only deal with NIWI defined without trusted setup, sometimes we omit the expression "in the plain model".

The first NIWI construction was proposed by Barak et al. [BOV03], obtained by derandomizing a specific class of NIZKs. This approach has drawbacks that make it unsuitable for applications. First, it is secure under a non-standard complexity-theoretic assumption. Moreover, the prover must send a logarithmic number of pairs CRS, proof, to the verifier, which result in a very inefficient construction, even starting from efficient NIZKs.

Later Groth et al. [GOS06] (and the journal version [GOS12]) proposed a much more efficient methodology to define a NIWI. Their idea is based on the fact that given two CRSs of a Groth-Sahai proof [GS08] it is possible for the verifier to efficiently check if at least one of them guarantees perfect soundness. Thus the NIWI is defined by letting the prover choose two Groth-Sahai CRS by itself, compute relative Groth-Sahai proofs and send them to the verifier. The latter checks the correctness of the proof and the CRSs. This construction solved the issues the previous one suffered: it is secure under standard, falsifiable, cryptographic assumptions and relatively efficient. Notably, the efficiency overhead is constant (in the security parameter, but not in the language size) compared to the corresponding NIZK in the CRS model: the proof consists of two Groth-Sahai proofs. However, this NIWI construction inherits all limitations of the Groth-Sahai framework, described in Section 1.7. Particularly, even if the efficiency is much better compared to the construction of Barak et al. twice as bad as Groth-Sahai is still not acceptable for many applications.

From previous work, the problem of defining a secure NIWI that is just as efficient as the corresponding NIZK in the CRS model was left as an open question.

Bitansky and Paneth [BP15] also defined NIWI under indistinguishability obfuscation and one-way permutations. However, their construction is impractical for applications, and is therefore left out of the current discussion.

1.11 Set-Membership Proofs

Set (non-)membership NIZKs are zero-knowledge proof systems to argue that a given element χ is (is not) in a public set S. We can define a set-membership proof using a digital *signature scheme* and a general NIZK framework (typical choices are Fiat-Shamir or Groth-Sahai). The CRS generator samples a pair of public and secret keys for the signature scheme, and computes a signature for each element in the set S. Then, it publishes the set-membership CRS composed by the public key, signatures of elements in S and a CRS for the used NIZK framework. The signature secret key is kept secret to ensure that it is impossible to compute valid signatures for elements not in S. The prover, on input an element in S, picks the corresponding signature in the CRS and encrypts the tuple element-signature. Then, it proves that the two ciphertexts are encryptions of elements which satisfy the signature verification.

Belenkiy et al. [BCKL08] defined a set-membership NIZK using a Structure-Preserving Signature scheme and a Groth-Sahai NIZK, with security based on falsifiable assumptions. Daza et al. [DGP$^+$19] improved on efficiency over [BCKL08], defined set-membership NIZK using the more efficient weak Boneh-Boyen signature scheme [BB04] and a Groth-Sahai NIZK. Signature-based set-membership NIZKs have two critical downsides. First, assuming S of polynomial size, its complement would be of exponential size. Consequently, defining a set non-membership argument in this setting seems impossible. Moreover, the CRS depends on the set S. Thus, one must securely compute a new CRS every time the set S changes.

A different approach is to define set-membership NIZKs, using a primitive called accumulator [Bd94]. An *accumulator* is a cryptographic primitive used to prove (non)-membership of an element in a set without zero-knowledge. To define a set membership NIZK, we can add the zero-knowledge property to an accumulator using a suitable compiler. Accumulator-based set membership NIZKs do not inherently suffer from the issues we have just described in the case of signature-based ones. It is possible to define universal (both membership and non-membership can be proven) accumulator-based set-membership NIZKs, and with CRS independent from the set S[4]. Being universal is a necessary property for some applications, for instance, anonymous credential systems. Acar and Nguyen [AN11] defined a universal set (non-)membership NIZK, using the Groth-Sahai framework as a compiler to add zero-knowledge to the accumulator from [Ngu05]. [AN11] NIZKs have security based on falsifiable assumptions and a CRS that only depends on the size of S.

Compared to best set (non-)membership NIZKs in the ROM [CCs08, VB20], all the mentioned solutions, secure under falsifiable assumptions, in the standard model, performs much worse in term of efficiency. Thus, it is an interesting problem to define more efficient set (non-)membership NIZKs, with security under falsifiable assumptions, and possibly CRS that only depends on the size of S.

Note that [DGP^{+}19] also proposed a different solution, based on the weak Boneh-Boyen signature scheme and a QA-NIZK (Quasi-Adaptive NIZK) [KW15], with security based on falsifiable assumptions. This QA-NIZK-based set membership has aggregation as an exciting feature (we can prove membership of many elements in a single proof). Moreover, it has better communication complexity compared to any Groth-Sahai set-membership NIZKs. However, as well as suffering from the issues it shares with other signature-based solutions, [DGP^{+}19] QA-NIZK set membership has verifier with computational complexity linear in the size of S. All the other solutions described here have verifier's computational complexity independent of the size of S. This independency is an essential requirement, since the size of S is significant in many applications, and the verifier's computational power is limited. Therefore, even acknowledging the better performance in proof size, we do not compare [DGP^{+}19] QA-NIZK set membership with competitors with constant time verifier.

1.12 Results of the Current Thesis

This thesis is a collection of three papers, with the common purpose of defining better and more efficient non-interactive protocols, under computational and, possibly falsifiable, assumptions, in the standard model. It is worth mentioning that each of the three papers results from a joint work. Therefore, a list of personal contributions is included in this section. However, it should be taken into account that sometimes it is not clear

[4] The CRS will depend only on the size of S.

to point out what can be considered a personal achievement in a joint work. In collaborative research, results often follow from ideas, rounds of feedback and interactions among authors. The list of personal contributions should then be considered as a list of arguments where the author of this thesis has contributed the most, in terms of leading the discussion as well as writing the final result. The full version of each article is included in the current thesis.

Efficient NIZKs for Algebraic Sets.

The first article [CLPØ21a], full version [CLPØ21b], is a joint work with Geoffroy Couteau, Helger Lipmaa, and Arne Tobias Ødegaard. In this paper, we propose a new methodology to define NIZKs whose security is based on computational cryptographic assumptions, building on the work of Couteau and Hartmann [CH20]. The core idea is a new and efficient Sigma protocol for showing that an encrypted vector decrypts to an element belonging to an algebraic set. An *algebraic set* is a set of elements that are common roots of a finite set of polynomials. Then, we apply the CH-compiler (Couteau Hartmann) to the new Sigma protocol, showing that this procedure leads to secure NIZKs. We significantly improve over [CH20] in terms of both security and expressivity.

The security of the resulting NIZK is shown under a new assumption: the CED (*Computational Extended Determinant*) assumption. The CED assumption is a weaker form of a specific ExtKerMDH assumption used in [CH20]; here weaker means that any adversary that breaks CED, is also a successful adversary against the ExtKerMDH assumption. As per the ExtKerMDH, the CED assumption is a computational assumption, although not always falsifiable. However, we were able to show the security of the new NIZK, under a falsifiable version of CED, for many more significant cases than [CH20]. Moreover, for the general case when CED is used in its non-falsifiable variant, we show that it can be reduced to a single, very plausible gap assumption [OP01]. It is unclear if the same holds for the more general ExtKerMDH assumption, for which only reductions in idealized models are known.

However, the most significant angle of improvement is in terms of expressivity. We show in our paper, how finding an efficient and natural representation of problems as algebraic sets is more straightforward compared to doing the same for other languages popular in pairing-based cryptography, such as algebraic languages (used in [CH20]) or PPEs (used in [GS08]). Consequently, our work improves efficiency over the previous state-of-the-art, for many interesting cases, as shown in the paper by comparison tables (see [CLPØ21a] 1, Table 1 and 2). Moreover, the new framework has the novel and appealing feature of allowing one to go from a high-level (non-cryptographic) description of the problem to the optimized NIZK almost automatically. Remarkably, the previous task does not require work from dedicated cryptographic experts.

Despite many years of research, we remark that the new framework for NIZKs defined in [CH20] and in this work represents the only known improvements in efficiency over the Groth-Sahai framework. The latter is achieved at the cost of relying on less standard, yet plausible, assumptions.

Author's contributions in this work included optimizing the efficiency of prover computational complexity, exploring the cases based on a falsifiable version of CED, partially writing soundness and zero-knowledge proofs and an extensive literature search and comparison with previous solutions.

NIWI and New Notions of Extraction for Algebraic Languages.

The second paper [GKP22b], full version [GKP22a], is a joint work with Chaya Ganesh, and Hamidreza Khoshakhlagh. In this work, we give a new construction of a NIWI in the plain model. Our NIWI (in the plain model) construction is based on, and is as efficient as, [CH20] NIZK proof (in the CRS model). We construct the NIWI, letting the prover pick its own CRS and output some auxiliary elements to prove the correctness of the choice. Therefore, the verifier can use the auxiliary elements to check that perfect soundness holds. Witness indistinguishability is proven under a new decisional (falsifiable) assumption. To motivate the new assumption, we prove that it holds in the AGM. The result is a new NIWI for algebraic languages, which is more efficient than the state-of-the-art competitor [GOS12], at the cost of relying on a less standard, but importantly falsifiable, assumption.

Furthermore, we explore the knowledge soundness of Couteau-Hartmann NIZK systems. We define the notion of strong partial knowledge soundness, and we prove that Couteau-Hartmann NIZK proof achieves this property. Then, we define the notion of semantic knowledge soundness. We investigate the relationship between semantic knowledge soundness, and different existing notions of knowledge soundness. We prove that semantic knowledge soundness is a general definition that recovers existing notions of knowledge soundness as special cases. Lastly, we show that Couteau-Hartmann NIZKs cannot satisfy semantic extraction in the standard model under the hardness of discrete logarithm.

Author's contributions in this work includes writing the AGM proof for the new assumption, writing the proof of strong partial extractability for Couteau-Hartmann NIZK proofs, formulating the definition of semantic knowledge soundness, partially writing the proof of relationship between semantic knowledge soundness, and different existing notions of knowledge soundness, and writing the proof for the impossibility result about knowledge soundness of Couteau-Hartmann NIZK arguments.

Set (Non-)Membership NIZKs from Determinantal Accumulators.

The last article [LP22] is a joint work with Helger Lipmaa. In this work, we construct a new set (non-)membership NIZK. To the best of our knowledge, the NIZK presented in [LP22], improves over the previous version from many angles. First, it achieves the best communication complexity and verifier computational complexity among all previous falsifiable set membership NIZKs: [BCKL08, AN11] and Groth-Sahai based NIZK from [DGP$^+$19]. Moreover, since it is an accumulator-based NIZK, it inherits all the advantages of accumulator-based solutions: it has a CRS that depends only on the size of the set S, and it is universal, supporting non-membership proofs. Since [LP22] set (non-)membership NIZK has constant time verifier, CRS independent from the set S, and supports non-membership proof, it should not be compared to the QA-NIZK set-membership from [DGP$^+$19]. Nevertheless, we note that our NIZK performs better in terms of proof size even compared to [DGP$^+$19] QA-NIZK set-membership.

The set (non-)membership NIZK defined in this work is based on the novel concept of determinantal accumulator and the CLPØ framework for NIZKs. We first define a determinantal accumulator, then compile it into a NIZK (adding zero-knowledge) using the CLPØ NIZK system. We can informally think of determinantal primitives as those that are “friendly” with the framework of [CH20, CLPØ21a]. Here the term friendly is used in the sense that the zero-knowledge compilation adds minimal overhead to the resulting NIZK, similar to how Structure-Preserving primitives are defined as “friendly” with the Groth-Sahai framework [BCKL08, AFG$^+$16]. More generally, we develop a straightforward and modular technique to define efficient NIZKs in the standard model. First, we construct an algebraic representation of a given problem. Then, we prove the security of a (not zero-knowledge) determinantal primitive. Finally, we compile it into a NIZK using CLPØ framework. We emphasize that the technique we have just described is an essential contribution of this article.

The security of the new determinantal accumulator is proven under new falsifiable assumptions. Once again, to justify introducing new assumptions, we prove they are secure in the AGM. The AGM security proof of the new assumptions is technically challenging and an important contribution of this work.

Since the pairing-based setting is nowadays a well-studied and established topic, it is not easy to come up with consistent advancement in efficiency for simple problems such as set (non-)membership NIZKs. This work shows that the CLPØ framework allows for improvements over Groth-Sahai solutions. We leave it as an open question to explore whether achieving the same advancements for other popular problems is possible.

As another significant contribution, we extend the framework defined in [CLPØ21a], by proposing a general methodology to prove non-membership in an algebraic set with minimal complexity overhead.

Author's contributions in this work included an extensive literature search and comparison with previous solutions, optimizing the efficiency of the accumulator scheme, partially writing the AGM proof for the new assumptions, proposing how to define the new assumptions in such a way they are falsifiable, and write the proof for the ZK compiler.

Chapter 2

Preliminaries

2.1 Notations and Basic Concepts

For any positive integer n, $[n]$ denotes the set $\{1,\ldots,n\}$. Let $\lambda \in \mathbb{N}$ be the security parameter. In the second paper, we denote the secure parameter with k. A function is negligible if it is definitely smaller than the inverse of any polynomial. Let $\mathsf{negl}(\lambda)$ be an arbitrary negligible function. We write $a(\lambda) \approx_{\lambda} b(\lambda)$ if $|a-b| \leq \mathsf{negl}(\lambda)$ for an arbitrary negligible function. Note that a is a negligible function if $a \approx_{\lambda} 0$. When a function can be expressed in the form $1-\mathsf{negl}(\lambda)$, we say that it is overwhelming in λ. Given two distritutions D_1, D_2 over the same support, we write $D_1 \equiv D_2$ to indicate that the two distributions are equal.

Let X, Y be two sets. A relation $\mathscr{R}$ is a subset of the cartesian product $X \times Y$. We call elements in the first set X statements. A witness for a statement $\mathtt{x} \in X$ is an element $\mathtt{w} \in Y$ such that $(\mathtt{x},\mathtt{w}) \in \mathscr{R}$. We associate to each relation the language $\mathscr{L}_{\mathscr{R}} = \{\mathtt{x} \in X : \exists \mathtt{w}, (\mathtt{x},\mathtt{w}) \in \mathscr{R}\}$. True statements are elements in the language $\mathscr{L}_{\mathscr{R}}$ and false statements elements in X but not in $\mathscr{L}_{\mathscr{R}}$. Some classes of languages are parametrized by a public language parameter $\mathtt{lpar}$ sampled from a certain distribution. See Section 2.8 for an example of such a class of languages. When this is the case, we indicate the language as $\mathscr{L}_{lpar,\mathscr{R}}$.

We use DPT (resp. PPT) to mean a deterministic (resp. probabilistic) polynomial time algorithm. We write $Y \leftarrow \mathscr{A}(X)$ to denote an algorithm with input X and output Y. $\mathsf{RND}_{\lambda}(\mathscr{A})$ denotes the random tape of $\mathscr{A}$ (for given λ), and $r \leftarrow_{\$} \mathsf{RND}_{\lambda}(\mathscr{A})$ denotes the uniformly random choice of r from $\mathsf{RND}_{\lambda}(\mathscr{A})$. We write $Y \leftarrow \mathscr{A}(X;r)$ to denote that a probabilistic algorithm outputs Y on input X and random coins r. Further, we write $a \leftarrow_{\$} S$ to denote that a is sampled according to distribution S, or uniformly randomly if S is a set. All adversaries will be stateful.

To represent matrices and vectors, we use bold letters. We use bold upper-case letters for matrices and bold lower-case letters for vectors. Vectors are, by default, column vectors. For a matrix $\mathbf{A}$, $\mathbf{A}_i$ denotes its ith row, $\mathbf{A}^{(j)}$ denotes its jth column, and $\mathbf{A}_{ij}$ denote the element at row i and column j.

An OWF (One-Way Function) f is a function such that there exists a PPT that computes $f(x)$, but given y no PPT can compute with more than negligible probability an x such that $y = f(x)$.

Security through games and reductions. We often prove that properties hold under certain assumptions. That is, we want to prove statements of the form "if the assumption A holds, then property B holds as well". To achieve this goal, we use games and reductions.

Cryptographic properties are frequently defined as games between parties in which one (or rarely more than one) tries to compute outputs with specific properties, receiving inputs through interaction with the other parties involved. For instance, the soundness property of zero-knowledge proofs is defined by a (potentially malicious) prover interacting with an honest verifier, trying to compute a valid proof for a false statement. Security is then proven by contradiction. We suppose the existence of an adversary $\mathscr{A}$ able to win the game that defines the property B. We then define an adversary $\mathscr{B}$ that runs $\mathscr{A}$. Lastly, we prove that $\mathscr{B}$ contradicts assumption A. In this case, we say that $\mathscr{B}$ breaks the assumption A. Since we conjecture that assumption A holds, then such $\mathscr{B}$ must not exist, which implies that the original $\mathscr{A}$ should not exist as well. It follows that property B holds as well, quod erat demonstrandum.

This procedure is called a reduction of property B to assumption A, or equivalently it is said that we have reduced property B to assumption A. More precisely, with a reduction, we prove by contradiction statements of the form "property B holds or assumption A is false".

Cryptographic assumptions. The DL (discrete logarithm) assumption in a group $\mathbb{G}$ of order p states that it is hard to compute the discrete logarithm of a random element in $\mathbb{G}$.

Definition 1 (Discrete logarithm assumption.) *Given a cyclic group $\mathbb{G}$ of order p generated by g, for each PPT adversary $\mathscr{A}$*

$$\Pr\left[\; g^w = h \;\middle|\; h \leftarrow_{\$} \mathbb{G}; w \leftarrow \mathscr{A}(g,h) \;\right] \leq \mathsf{negl}(\lambda).$$

We now state the CDH (Computational Diffie-Hellman) assumption.

Definition 2 (CDH **assumption.**) *Given a cyclic group $\mathbb{G}$ of order p generated by g, for each PPT adversary $\mathscr{A}$*

$$\Pr\left[\ h = g^{xy} \ \middle|\ x, y \leftarrow_{\$} \mathbb{Z}_p; h \leftarrow \mathscr{A}(g, g^x, g^y)\ \right] \leq \mathsf{negl}(\lambda).$$

We also state a decisional version of the Diffie-Hellmann assumption: the DDH assumption.

Definition 3 (DDH **assumption.**) *Given a cyclic group $\mathbb{G}$ of order p generated by g, for each PPT adversary $\mathscr{A}$*

$$\Pr\left[\ b' = b \ \middle|\ x, y, z \leftarrow_{\$} \mathbb{Z}_p; b \leftarrow_{\$} \{0,1\}; b' \leftarrow \mathscr{A}(g, g^x, g^y, g^{xy+bz})\ \right] \leq 1/2 + \mathsf{negl}(\lambda).$$

2.2 Algebraic Branching Programs

In this thesis, we use an algebraic model of computation for polynomials over fields called ABP (Algebraic Branching Program) [Nis91, BG99]. An ABP is a directed acyclic graph with two special vertices s (source vertex) and t (target vertex), and a function that assigns a label to each edge. Each label is an affine multivariate function (a polynomial of total degree up to 1). An ABP computes a polynomial f if f is equal to the sum over all paths from s to t of the products of labels in the path.

Definition 4 (Algebraic Branching Program.) *An algebraic branching program is a tuple (V, E, s, t, ϕ) such that (V, E) is a directed acyclic graph, $s, t \in V$ are two distinct vertices and $\phi : E \to \mathbb{Z}_p[\mathbf{X}]$ is a function that associates each edge to a polynomial. In addition, we require the following conditions*

1. *No edges go out from t or into s. That is, for each $v \in V$ we have $(v, s), (t, v) \notin E$.*
2. *For each $e \in E$, $\phi(e)$ is an affine function (a linear polynomial).*

Let $s - t$ be the set of all possible paths from s to t in the graph. We say that the ABP (V, E, s, t, ϕ) computes a polynomial F if $F(\mathbf{X}) = \sum_{P \in s-t} \prod_{a \in P} \phi(a)$.

In Fig. 2.1, we can see as an example an ABP that computes $F(X, Y) = X^3 + aX + b - Y^2$. Note that there are 4 different paths from s to t. For each of them the product of the labels in the path is equal to a monomial of F. As a model of computation, ABP can be used to represent a wide range of functions, such as log-depth arithmetic circuits and boolean formulas. Moreover, ABPs often provide an efficient and compact representation of the polynomial they compute. See [Val79, SY10] a more precise characterization of the class of functions that admit an efficient ABP representation.

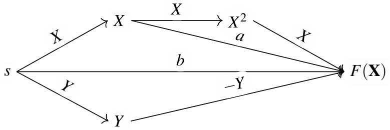


Figure 2.1: ABP example for $F(X,Y) = X^3 + aX + b - Y^2$.

See [IK00, IK02, IW14] and the reference they cite for a more in-depth discussion about ABPs.

2.3 Asymmetric Bilinear Pairings

Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be three different additive, cyclic groups of order a large prime p, generated respectively by g_1, g_2, g_T. Let also $\mathbb{Z}_p$ be the ring of integers, modulo p. We use bracket notation for groups, introduced in [EHK$^+$13]. For $\iota \in \{1,2,T\}$ and each $x, y \in \mathbb{Z}_p$, we denote the generator g_ι with $[1]_\iota$ and we write $[x]_\iota$ for g_ι^x, and $[x]_\iota + [y]_\iota$ for $g_\iota^x g_\iota^y$.

As a common practise in cryptographic literature, starting from [BF01], we require the existence of a bilinear pairing function $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, with the following property.

Non-degeneracy: the element $e([1]_1, [1]_2)$ is a generator of $\mathbb{G}_T$, denoted as $[1]_T$..

Bilinearity: for each $a, b \in \mathbb{Z}_p$, $e(a[1]_1, b[1]_2) = ab[1]_T$.

Efficiently computable: e can be computed by an efficient deterministic Turing machine.

We denote $e([x]_1, [y]_2)$ as $[x]_1 \bullet [y]_2$. We write $[x]_1 \bullet [y]_2 = e([x]_1, [y]_2) = [xy]_T$. For each $x \in \mathbb{Z}_p$ and each group element $[y]_\iota$, we denote the operation $x[y]_\iota = [xy]_\iota$ with scalar multiplication. Since scalar multiplication corresponds to the exponentiation in cyclic multiplicative groups, with a little abuse of notation we also sometimes indicate this operation as exponentiation[1]. To use bilinear groups in cryptography, we require additional hardness properties. We require the hardness of the discrete logarithm problem Definition 1, in all three groups: for $\iota \in \{1,2,T\}$, given a uniformly random group element $[x]_\iota$ no PPT can compute x with more than negligible probability. More precisely, we require that for each $\iota \in \{1,2,T\}$, $[\cdot]_\iota : \mathbb{Z}_p \to \mathbb{G}_\iota$ is an OWF. Lastly, we require that

[1] In the second paper [GKP22b] we always use the term exponentiation for the operation $x[y]_\iota = [xy]_\iota$.

no efficiently computable isomorphism exists between $\mathbb{G}_1$ and $\mathbb{G}_2$. This last property is what characterizes type III bilinear pairings, which are also called asymmetric pairings [GPS06]. In contrast, in type I, we set $\mathbb{G}_1 = \mathbb{G}_2$ and in type II, we require the existence of an efficient isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. In this thesis, we always use type III pairings.

We use bilinear pairings as a black-box structure, assuming the needed hardness property to hold. More precisely, we assume that there exists an efficient algorithm Pgen that takes as input a security parameter and returns the description of the three groups, with a generator and the description of the pairing operation. We also assume that $\mathsf{p} = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, [1]_1, [2]_1, \bullet) \leftarrow \mathsf{Pgen}(1^\lambda)$, is given as input to all the algorithms. However, we sometimes do not list p explicitly as input. For real applications, groups defined by elliptic curves over finite fields are used to implement bilinear cryptographic pairings; see [BSS00, BD19] for details about implementation. We point out that using type III pairings leads to implementations of group elements with shorter bit representation, thus better efficiency. See [BD19] for more information about size and efficiency of type III state-of-the-art pairings. As a rule of thumb, it is important to take in mind that elements in $\mathbb{G}_2$ are about twice longer in bit size than elements in $\mathbb{G}_1$.

Finally, we naturally extend this notation to vectors and matrices, applying bracket operators pointwise, and define the pairing operation as $[\mathbf{A}]_1 \bullet [\mathbf{B}]_2 = [\mathbf{AB}]_T$ for any two matrices with compatible dimensions.

We state some standard assumptions in pairing based cryptography.

Definition 5 (Symmetric discrete logarithm assumption.) *For any PPT adversary $\mathscr{A}$, it holds that*

$$\Pr\left[\ x = x' \ \middle|\ x \leftarrow_{\$} \mathbb{Z}_p; x' \leftarrow \mathscr{A}([x]_1, [x]_2)\ \right] \leq \mathsf{negl}(\lambda).$$

Let $D_{k,l}$ be a distribution of matrices over $\mathbb{Z}_p^{k \times l}$.

Definition 6 (KerMDH-assumption.) *For any PPT adversary $\mathscr{A}$, it holds that for each $\iota \in \{1,2\}$*

$$\Pr\left[\ \mathbf{c}^T \mathbf{A} = 0 \ \middle|\ \mathbf{A} \leftarrow_{\$} D_{k,l}; [\mathbf{c}]_{3-\iota} \leftarrow \mathscr{A}([\mathbf{A}]_\iota)\ \right] \leq \mathsf{negl}(\lambda).$$

2.4 Algebraic Group Model

The AGM (Algebraic Group Model) [FKL18] is an idealized model of computation, where we consider only a restricted class of adversaries.

Let us consider algorithms that receive group elements as inputs. We call algebraic any algorithm that, when it outputs a group element, it must know a representation of the output as a linear combination of its inputs. Such algebraic algorithms were first considered in [BV98] and defined in [PV05]. Differently from how it is done in [FKL18], we formalize algebraic algorithms, by the existence of an extractor that computes the coefficients of such linear combination, given the algorithm code and the random coins it used for the execution. We give the formal definition directly for asymmetric setting, where any output in a given group must be a known linear combination only of inputs in the same group.

Definition 7 *An algorithm $\mathscr{A}$ is algebraic if there exists a PPT extractor $\mathsf{Ext}_{\mathscr{A}}$ such that, for any vector of group elements $\mathbf{X} = ([\mathbf{X_1}]_1, [\mathbf{X_2}]_2)$, we have*

$$\Pr\left[\begin{array}{c|c} \mathbf{Y_1} \neq \alpha_1 \mathbf{X_1} \wedge & \mathsf{p} \leftarrow_{\$} \mathsf{Pgen}(1^{\lambda}); r \leftarrow_{\$} \mathsf{RND}_{\lambda}(\mathscr{A}); \\ \mathbf{Y_2} \neq \alpha_2 \mathbf{X_2} & ([\mathbf{Y_1}]_1, [\mathbf{Y_2}]_2) \leftarrow \mathscr{A}(\mathbf{X}; r); (\alpha_1, \alpha_2) \leftarrow \mathsf{Ext}_{\mathscr{A}}(\mathbf{X}, r) \end{array}\right] \leq \mathsf{negl}(\lambda)$$

See Section 1.4 for a discussion about the choice of defining AGM with extractors and the relation between AGM and knowledge assumptions.

In AGM, we show security by reductions, with the restriction that we consider only algebraic adversaries. Reductions in AGM heavily rely on algebraic extractors.

Therefore, a proof of security in AGM is a proof of statements of the form “property B holds for all algebraic adversaries, or assumption A is false”. It is common to use AGM to justify introducing new assumptions, proving that they hold against algebraic adversaries under well-established and standard assumptions, such as the hardness of discrete logarithm. As stated in Section 1.4, an AGM reduction to a well-established assumption is considered a better proof of security than a proof in GGM [Sho97, Mau05], another idealized model for cryptographic groups.

Recall that decisional assumptions adversaries output bits, which are not group elements. The original AGM [FKL18] lacks a definition of algebraic adversaries for decisional assumptions. Rotem and Segev [RS20] show how to extend the AGM to prove the security of decisional assumptions, defining algebraic distinguishers.

2.5 Public-Key Encryption Scheme

A public-key encryption scheme is a triple of PPT algorithms.

Key generator: on input a security parameter, outputs a pair of correlated public key pk and secret key sk.

Encryption: on input a public key pk and a plaintext m, computes a ciphertext $\mathbf{c} = \mathsf{Enc}_{\mathsf{pk}}(m;r)$ using random coins r.

Decryption: on input a ciphertext $\mathbf{c}$ and a secret key sk, computes a plaintext $m = \mathsf{Dec}_{\mathsf{sk}}(\mathbf{c})$.

In this thesis, we use public-key encryption scheme with the following security properties.

Correctness: for each pair of $(\mathsf{pk},\mathsf{sk})$ computed by the key generator algorithm, for each plaintext m and random coins r, we have $m = \mathsf{Dec}_{\mathsf{sk}}\left(\mathsf{Enc}_{\mathsf{pk}}(m;r)\right)$.

IND-CPA security: for each two-stage PPT algorithm $\mathscr{A}_1, \mathscr{A}_2$ it holds that

$$\Pr\left[\begin{array}{c|c} b = b' & \begin{array}{c}(\mathsf{pk},\mathsf{sk}) \leftarrow_{\$} \mathsf{kgen}(1^\lambda); (m_0, m_1, \mathsf{st}) \leftarrow \mathscr{A}_1(\mathsf{pk}); \\ b \leftarrow_{\$} \{0,1\}; b' \leftarrow \mathscr{A}_2(\mathsf{st}, \mathsf{Enc}_{\mathsf{pk}}(m_b, r))\end{array}\end{array}\right] \approx_\lambda \frac{1}{2},$$

where r is fresh randomness and st is an internal state that the two-stage adversary uses to pass information from one stage to the other.

Public keys of a public-key encryption scheme, are used as language parameters for parametrized languages $\mathscr{L}_{\mathtt{lpar},A}$ described in Section 2.1. Paticularly we can define $\mathtt{lpar} = \mathsf{pk}$ and $LANG_{\mathtt{lpar},\mathscr{R}} = \{\mathbf{c} : \mathbf{c} = \mathsf{Enc}_{\mathsf{pk}}(x;r) \land x \in A\}$.

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

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

To decrypt, one computes $[m]_1 = \mathsf{Dec}_{\mathsf{sk}}([\mathbf{c}]_1) \leftarrow -\mathsf{sk}[c_1]_1 + [c_2]_1$. In what follows, we denote $[\mathbf{c}]_1 = \mathsf{Enc}(m;r)$ for a fixed public key $\mathsf{pk} = [1\|\mathsf{sk}]_1$. Recall the DDH assumption: given $x, y, z \leftarrow_{\$} \mathbb{Z}_p$ and a bit b, no PPT can computes a bit b' on input $([x]_1, [y]_1, [xy + bz]_1)$ such that $b = b'$, with probability greater than $1/2 + \mathsf{negl}(\lambda)$. If the DDH assumption holds Definition 3, then the Elgamal encryption scheme is IND-CPA secure.

Note that Elgamal encryption scheme is linearly homomorphic: given two ciphertexts $[\mathbf{c_1}]_1 = \mathsf{Enc}_{\mathsf{pk}}(m_1;r_1), [\mathbf{c_2}]_1 = \mathsf{Enc}_{\mathsf{pk}}(m_2;r_2)$ we have that $\mathsf{Dec}_{\mathsf{sk}}([\mathbf{c_1}]_1 + [\mathbf{c_2}]_1) = [m_1]_1 + [m_2]_1$. We heavily use the linear homomorphic property of the Elgamal encryption scheme in this thesis.

$$\begin{tikzcd}
\mathsf{P}[\mathtt{x},\mathtt{w}] & & \mathsf{V}(\mathtt{x}) \\
(a,\mathsf{st}) \leftarrow \mathsf{P}(\mathtt{x},\mathtt{w}) \arrow[rr, "{[a]}"] & & {} \\
{} & & e \leftarrow_{\$} \arrow[ll, "e"'] \\
d \leftarrow \mathsf{P}(\mathtt{x},\mathtt{w},\mathsf{st},e) \arrow[rr, "d"] & & {} \\
 & & b \leftarrow \mathsf{V}(\mathtt{x},\pi = (a,e,d))
\end{tikzcd}$$

Figure 2.2: The flow of a Sigma protocol

2.6 Sigma Protocols

A Sigma protocol [CDS94] for a relation $\mathscr{R}$ is a public-coin three-round interactive protocol between a prover P and a verifier V on input a common statement $\mathtt{x}$. The prover has a witness $\mathtt{w}$, such that $(\mathtt{x},\mathtt{w}) \in \mathscr{R}$ as additional secret input. The flow of a Sigma protocol is depicted in Fig. 2.2.

For any pair of interactive algorithms A,B, we denote with $\langle A(a),B\rangle(b)$ the output of B, after an interaction with A, on common input b and A's secret input a. Standard security notions for a Sigma protocol are completeness, special soundness, and special honest verifier zero-knowledge (SHVZK).

Completeness : for any $(\mathtt{x},\mathtt{w}) \in \mathscr{R}$

$$\Pr\left[\ \langle \mathsf{P}(\mathtt{w}),\mathsf{V}\rangle(\mathtt{x}) = 1 \vee\ (\mathtt{x},\mathtt{w}) \notin \mathscr{R}\ \middle|\ (\mathtt{x},\mathtt{w}) \leftarrow \mathscr{A}(1^\lambda)\ \right] = 1$$

Special Soundness: there exists a PPT algorithm Ext that given a statement $\mathtt{x}$ and two accepting transcripts $(a,e,d),(a,e',d')$ with the same first message and $e \neq e'$ outputs a witness $\mathtt{w}$, such that $(\mathtt{x},\mathtt{w}) \in \mathscr{R}$ with overwhelming probability.

Special Honest-Verifier Zero-Knowledge (SHVZK): there exists a PPT simulator Sim such that for any $(\mathtt{x},\mathtt{w}) \in \mathscr{R}$ and $e \in \{0,1\}^k$, the distributions of $\mathsf{Sim}(\mathtt{x},e)$ is identical to the distribution of the 3-move honest transcript obtained when V sends e as challenge and P runs on common input $\mathtt{x}$ and private input $\mathtt{w}$.

Sometimes, optimal soundness [MP03] is achieved, as an alternatively weaker notion of soundness. Roughly speaking, a Sigma protocol is *optimally sound* if given a false statement $\mathtt{x}$ there do not exist two accepting transcripts $(a,e,d),(a,e',d')$ with the same first message and $e \neq e'$.

2.7 Non-Interactive Zero-Knowledge Arguments

A NIZK [BFM88] for a language $\mathscr{L}_{\mathscr{R}}$ consists of four PPT algorithms.

CRS generator kgen**:** on input the security parameter, generates a CRS $\mathtt{crs}$ and a trapdoor $\mathtt{td}$.

Prover P**:** on input a CRS $\mathtt{crs}$, a statement $\mathtt{x}$, and a witness $\mathtt{w}$ outputs a proof π for $\mathtt{x} \in \mathscr{L}_{\mathscr{R}}$, or equivalently $(\mathtt{x}, \mathtt{w}) \in \mathscr{R}$.

Verifier V**:** on input a CRS, a statement and a proof, outputs 1 for accepting or 0 for rejecting the proof.

Simulator Sim**:** on input a couple of $(\mathtt{crs}, \mathtt{td})$ CRS with a relative trapdoor, and a true statement $\mathtt{x}$ computes a simulated proof π.

In addition, the following properties are required.

Perfect completeness: for any pair $(\mathtt{x}, \mathtt{w}) \in \mathscr{R}$, and for any $(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{kgen}(1^\lambda)$ we have

$$\Pr\left[\, \mathsf{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 \,\middle|\, \pi \leftarrow \mathsf{P}(\mathtt{crs}, \mathtt{x}, \mathtt{w}) \,\right] = 1.$$

Computational adaptive soundness: for any PPT adversary $\mathscr{A}$

$$\Pr\left[\begin{array}{c} \mathsf{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 \wedge \\ \mathtt{x} \notin \mathscr{L}_{\mathscr{R}} \end{array} \,\middle|\, \begin{array}{c} (\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{kgen}(1^\lambda); \\ (\mathtt{x}, \pi) \leftarrow \mathscr{A}(\mathtt{crs}) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

If valid proofs for false statement cannot exist, we have *perfect soundness*. Informally, computational non-adaptive soundness is defined by letting the adversary choose the false statement $\mathtt{x}$ before recieving the CRS. In this work we never use non-adaptive soundness, and we implicitly refer to computational soundness as computational adaptive soundness, omitting the term adaptive.

Perfect zero-knowledge: for any $(\mathtt{x}, \mathtt{w}) \in \mathscr{R}$, and for any $(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{kgen}(1^\lambda)$ the following distributions are identical

$$\mathsf{P}(\mathtt{crs}, \mathtt{x}, \mathtt{w}) \equiv \mathsf{Sim}(\mathtt{crs}, \mathtt{td}, \mathtt{x}).$$

If the two distributions are computationally indistinguishable, we have computational zero-knowledge.

Note that the zero-knowledge property is formalized by the existence of an efficient simulator that can compute valid proofs without knowing the witness. This definition aims to capture that, if it is possible to compute proofs distributed as those computed by the honest prover, without knowing the witness, then the proof carries no information about the witness. The simulator can compute valid proofs, without knowing the witness, only because it receives a secret trapdoor $\mathtt{td}$ as additional input. For soundness to hold, it is required that computing the trapdoor $\mathtt{td}$ from the CRS $\mathtt{crs}$ is computationally intractable. Pass showed that it is impossible to define NIZKs with perfect soundness and perfect zero-knowledge [Pas13].

We say that a NIZK is black-box knowledge-sound if there exists a PPT extractor that computes a witness, for a statement, given a CRS with the related trapdoor, and an accepting proof. If the extractor is allowed to depend on the adversary and receives the random coins used by the adversary as additional input, the NIZK achieves white-box knowledge soundness.

Black-box knowledge soundness: there exists an extractor $\mathsf{Ext}_{\mathsf{BB}}$ such that, for any PPT adversary $\mathscr{A}$:

$$\Pr\left[\begin{array}{c|c} \mathsf{V}(\mathtt{crs},\mathtt{x},\pi)=1 & (\mathtt{crs},\mathtt{td}) \leftarrow \mathsf{kgen}(1^{\lambda}); \\ \wedge(\mathtt{x},\mathtt{w}) \notin \mathscr{R} & (\mathtt{x},\pi) \leftarrow \mathscr{A}(\mathtt{crs}); \mathtt{w} \leftarrow \mathsf{Ext}_{\mathsf{BB}}(\mathtt{td},\mathtt{x},\pi) \end{array}\right] \leq \mathsf{negl}(\lambda)$$

White-box knowledge soundness: for any PPT adversary $\mathscr{A}$, there exists an efficient extractor $\mathsf{Ext}_{\mathsf{WB},\mathscr{A}}$ such that:

$$\Pr\left[\begin{array}{c|c} \mathsf{V}(\mathtt{crs},\mathtt{x},\pi)=1 & (\mathtt{crs},\mathtt{td}) \leftarrow \mathsf{kgen}(1^{\lambda}); r \leftarrow_{\$} \mathsf{RND}_{\lambda}(\mathscr{A}); \\ \wedge(\mathtt{x},\mathtt{w}) \notin \mathscr{R} & (\mathtt{x},\pi) \leftarrow \mathscr{A}(\mathtt{crs}; r); \mathtt{w} \leftarrow \mathsf{Ext}_{\mathsf{WB},\mathscr{A}}(\mathtt{td},\mathtt{x},\pi,r) \end{array}\right] \leq \mathsf{negl}(\lambda)$$

where r is the random coins of $\mathscr{A}$.

Lastly, we state the witness indistinguishability definition [FS90] for non-interactive protocols, as used in [GOS12]. Roughly speaking, a protocol is witness indistinguishable if it is impossible to distinguish which witness the prover used to compute a valid proof. Recall that we are interested in non-interactive witness indistinguishable proof systems in the plain model without a trusted setup.

Witness Indistinguishability (WI): for every PPT verifier $(\mathsf{V}_1^*,\mathsf{V}_2^*)$, for all $(\mathtt{x},\mathtt{w}_1,\mathtt{w}_2)$

such that $(\mathtt{x},\mathtt{w}_1) \in \mathscr{R}_{\mathtt{lpar}}, (\mathtt{x},\mathtt{w}_2) \in \mathscr{R}_{\mathtt{lpar}}$, we have

$$\Pr\left[b = b' \;\middle|\; \begin{array}{c} (\mathtt{x},\mathtt{w}_1,\mathtt{w}_2,\mathsf{st}) \leftarrow \mathsf{V}_1^*(\mathtt{lpar}); b \leftarrow_{\$} \{0,1\}; \\ \pi \leftarrow \mathsf{P}(\mathtt{lpar},\mathtt{x},\mathtt{w}_b); b' \leftarrow \mathsf{V}_2^*(\mathsf{st},\pi) \end{array} \right] \approx_\lambda \frac{1}{2}$$

Here st is an internal state that the two-stage adversary uses to pass information from one stage to the other. See Section 1.10 for a discussion about NIWI in the plain model.

2.8 Couteau-Hartmann Framework

An algebraic language is a language described by linear equations over abelian groups. Let $l,k,n \in \mathbb{N}$, $\mathtt{lpar} = (\Gamma,\theta)$ be a pair of linear maps $\Gamma : \mathbb{G}^l \to \mathbb{G}^{n\times k}, \theta : \mathbb{G}^l \to \mathbb{G}^n$.[2] The language $\mathscr{L}_{\mathtt{lpar}}$ is defined as

$$\mathscr{L}_{\mathtt{lpar}} = \{[\mathtt{x}]_1 \in \mathbb{G}^l : \exists \mathtt{w} \in \mathbb{Z}_p^k, [\Gamma(\mathtt{x})]_1\mathtt{w} = [\theta(\mathtt{x})]_1\}.$$

Algebraic languages are as expressive as NP, because we can use them to represent boolean circuits. Linear languages are algebraic languages such that Γ is a constant function. Many languages of interest for applications admit a representation as a linear language. For instance, the language of Elgamal encryption of bits is a linear language. See [CH20] and their references, or the second paper in this thesis [GKP22a], for more information about linear languages.

Couteau and Hartmann [CH20] introduce a new approach to define pairing-based NIZKs for algebraic langauges.

Their methodology consists in obtaining a NIZK from compiling a Sigma protocol for algebraic languages, reported for completeness in Fig. 2.3. See Section 1.8 in the intro, for a discussion on the compiler they used.

The NIZK is depicted in Fig. 2.4. Soundness of the NIZK argument is proven under the $\mathsf{ExtKerMDH}$ assumption. For completeness we report the definition of the $\mathsf{ExtKerMDH}$ assumption, and the theorem about the security of the NIZK argument.

Definition 8 ($\mathsf{ExtKerMDH}$ assumption.) *Let $\mathscr{L}_1$ be the distribution $\left[\begin{smallmatrix}1\\e\end{smallmatrix}\right]_2$, where $e \leftarrow_{\$} \mathbb{Z}_p$. The $\mathscr{L}_1$-(k)-$\mathsf{ExtKerMDH}$ assumption holds in $\mathbb{G}_2$ relative to Pgen, if for all PPT adver-*

[2] Sometimes algebraic languages are defined as maps from the field $\mathbb{Z}_p$ to the group

$$\begin{array}{lcl} \mathsf{P}(\texttt{lpar},[\mathrm{x}],\mathtt{w}) & & \mathsf{V}(\texttt{lpar},[\mathrm{x}]) \\ \mathbf{r} \leftarrow_{\$} \mathbb{Z}_p^k & & \\ [\mathbf{a}] := [\Gamma(\mathrm{x})]\mathbf{r} & & \\ & \xrightarrow{\qquad [\mathbf{a}] \qquad} & \\ & & e \leftarrow_{\$} \mathbb{Z}_p \\ & \xleftarrow{\qquad e \qquad} & \\ \mathbf{d} := \mathtt{w}e + \mathbf{r} & & \\ & \xrightarrow{\qquad \mathbf{d} \qquad} & \\ & & [\Gamma(\mathrm{x})]\mathbf{d} \stackrel{?}{=} [\theta(\mathrm{x})]e + [\mathbf{a}] \end{array}$$

Figure 2.3: Sigma protocol for an algebraic language $\mathscr{L}_{\texttt{lpar}}$ with $\texttt{lpar} = (\Gamma, \theta)$

```
$\mathsf{kgen}(1^\lambda)$
$e \leftarrow_{\$} \mathbb{Z}_p$
$\texttt{crs} := (\mathsf{p}, [e]_2); \texttt{td} := e$
$\textbf{return } (\texttt{crs}, \texttt{td})$

$\mathsf{P}(\texttt{lpar}, \texttt{crs}, [\mathrm{x}]_1, \mathtt{w})$
$\mathbf{r} \leftarrow_{\$} \mathbb{Z}_p^k$
$[\mathbf{a}]_1 := [\Gamma(\mathrm{x})]_1 \mathbf{r}$
$[\mathbf{d}]_2 := \mathtt{w}[e]_2 + [\mathbf{r}]_2$
$\textbf{return } \pi := ([\mathbf{a}]_1, [\mathbf{d}]_2)$

$\mathsf{Sim}(\texttt{lpar}, \texttt{crs}, e, [\mathrm{x}]_1)$
$\mathbf{d} \leftarrow_{\$} \mathbb{Z}_p^l;$
$[\mathbf{A}]_1 = [\Gamma(\mathrm{x})]_1 \mathbf{d} - e[\theta(\mathrm{x})]_1;$
$\textbf{return } \pi := ([\mathbf{a}]_1, [\mathbf{d}]_2)$

$\mathsf{V}(\texttt{lpar}, \texttt{crs}, [\mathrm{x}]_1, \pi = ([\mathbf{a}]_1, [\mathbf{d}]_2))$
$[\Gamma(\mathrm{x})]_1 \bullet [\mathbf{d}]_2 =^? [\theta(\mathrm{x})]_1 \bullet [e]_2 + [\mathbf{a}]_1 \bullet [1]_2$
```

Figure 2.4: NIZK argument for algebraic language $\mathscr{L}_{\texttt{lpar}}$ with $\texttt{lpar} = (\Gamma, \theta)$ [CH20]

saries $\mathscr{A}$, the following probability is negligible:

$$\Pr\left[\begin{array}{c|c} [\mathbf{C}]_1 \in \mathbb{G}_1^{k+1\times k+2} \wedge [\delta]_2 \in \mathbb{G}_2^k \wedge & \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda), [\mathbf{D}]_2 \leftarrow_{\$} \mathscr{L}_1, \\ \mathbf{C}\binom{\mathbf{D}}{\delta} = \mathbf{0} \wedge \mathrm{rk}(\mathbf{C}) \geq k & ([\mathbf{C}]_1, [\delta]_2) \leftarrow \mathscr{A}(\mathsf{p}, [\mathbf{D}]_2) \end{array} \right].$$

Proposition 1 ([CH20]) *Let $\mathscr{L}_{\Gamma,\theta}$ be an algebraic language, with $\Gamma : \mathbb{G}^l \to \mathbb{G}^{n\times k}, \theta : \mathbb{G}^l \to \mathbb{G}^n$. The NIZK argument in Fig. 2.4 for the algebraic language $\mathscr{L}_{\Gamma,\theta}$ is perfect complete, perfect zero-knowledge and computational adaptive sound, under the $\mathscr{L}_1$-(k)-*ExtKerMDH *assumption in $\mathbb{G}_2$.*

Couteau and Hartmann also propose a second compiler that leads to a NIZK proof with perfect soundness and computational zero-knowledge, applied to the same Sigma protocol. We refer to Section 3.2 for an informal description of this second compiler and to [CH20] or the second article in this thesis [GKP22b] for a formal description and the security proof.

2.9 Accumulator

Benaloh and de Mare defined accumulators in [BdM93]. Universal accumulators [BLL00, BLL02, LLX07, Lip12b, DHS15] allow non-membership arguments.

We define accumulators in the CRS model only. Hence, within the context of the current paper, universal accumulators are set (non-)membership arguments, without zero-knowledge, in the case the input χ is public. That is, for $\mathtt{lpar} = \mathscr{S}$, a universal (CRS-model) accumulator is a (non-zk) set (non-)membership non-interactive argument system for the following complementary languages:

$$\mathscr{L}^{\mathsf{acc}}_{\mathtt{lpar}} = \mathscr{S} \quad , \quad \bar{\mathscr{L}}^{\mathsf{acc}}_{\mathtt{lpar}} = \mathscr{D} \setminus \mathscr{S} \quad .$$

Here $\mathscr{D} \subseteq \mathbb{Z}_p$ is the set of elements that can be accumulated. The computation commitment algorithm com corresponds to the accumulator's commitment algorithm that inputs a set $\mathscr{S}$ and outputs its short commitment. A CRS-model accumulator can have a trapdoor. However, since χ is public (and no zero-knowledge is required) then the trapdoor is not used.

As with all argument systems, a universal accumulator must satisfy completeness and soundness properties. Because of historical reasons, the latter is usually known as *collision-resistance*. Full definitions follow.

A universal accumulator ACC must be *perfectly complete*: for $(\mathtt{crs}, \mathtt{td}) \in \mathsf{kgen}(1^\lambda)$, $\chi \in \mathscr{D}$, and $\mathscr{S} \subseteq \mathscr{D}$, $\mathsf{V}(\mathtt{crs}, \mathsf{com}(\mathtt{crs}, \mathscr{S}), \chi, \mathsf{P}(\mathtt{crs}, \mathscr{S}, \chi))$ outputs Member if $\chi \in \mathscr{S}$ and $\mathsf{NotMember}$ if $\chi \notin \mathscr{S}$.

Definition 9 *Let* ACC *be a universal accumulator.* ACC *is* collision-resistant [BP97] *if for all* $N = \mathsf{poly}(\lambda)$ *and PPT adversaries* $\mathscr{A}$*,*

$$\Pr\left[\begin{array}{c|c} \mathscr{S} \in \mathscr{D}^{\leq N} \wedge & \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \\ \left(\begin{array}{c} (\chi \notin \mathscr{S} \wedge v = \mathsf{Member}) \vee \\ (\chi \in \mathscr{S} \wedge v = \mathsf{NotMember}) \end{array}\right) & \begin{array}{c} (\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{kgen}(\mathsf{p}, N); \\ (\mathscr{S}, \chi, \psi) \leftarrow \mathscr{A}(\mathtt{crs}); \end{array} \\ & v \leftarrow \mathsf{V}(\mathtt{crs}, \mathsf{com}(\mathtt{crs}, \mathscr{S}), \chi, \psi) \end{array}\right] \leq \mathsf{negl}(\lambda),$$

where $\mathscr{D}^{\leq N}$ *is a set of elements in* $\mathscr{D}$ *of size up to N.*

Nguyen [Ngu05] proposed a pairing-based CRS-model accumulator with $\mathscr{D} = \mathbb{Z}_p$.

Damgård and Triandopoulos [DT08] and Au et al. [ATSM09] showed independently how to make it universal by adding a non-membership argument.

Sometimes accumulator can satisfy a weaker notion of security called f-collision-resistance, where collision-resistance holds even against adversaries that output $f(\chi)$, instead of χ. In this thesis, we use the $[\cdot]_1$-collision-resistance property. See the third paper in this thesis [LP22] for more about f-collision-resistance.

Chapter 3

Technical overview

3.1 Article I

The main technical contribution in [CLPØ21a] is a new methodology to construct a NIZK for languages defined by encryptions of common roots of a set of polynomials. As explained in Section 1.12, the new NIZK framework is an improvement over in [CH20] in many respects. We show how to go from a high-level description of a given relation, to an efficient NIZK for that relation, in an almost automatic way. Notably, for the NIZK described in this article, the described task does not require dedicated cryptographic expertise. We also show better security conditions for soundness.

More precisely, let ν be a small integer and p be a large prime. Let $\mathscr{M} = \{f\}$ be a finite set of ν-variate polynomials over $\mathbb{Z}_p$. Define the *algebraic set* generated by $\mathscr{M}$ as $A = \{\mathbf{x} \in \mathbb{Z}_p^\nu : \forall f \in \mathscr{M}, f(\mathbf{x}) = 0\}$. $\mathscr{M}$ is called a base of the algebraic set A. We refer to the article [CLPØ21a] for a more precise discussion about algebraic sets and their relations with ideals in polynomial rings. We choose a linearly homomorphic, public-key encryption scheme, for instance, the Elgamal encryption scheme described in Section 2.5. We let pk be an Elgamal public key and define the language of encryption of elements in A as

$$\mathscr{L}_{\mathsf{pk},A} = \{[\mathbf{c}]_1 : \exists (r, \mathbf{x}), \mathsf{Enc}_{\mathsf{pk}}(\mathbf{x}, r) = [\mathbf{c}]_1 \land \mathbf{x} \in A\}.$$

Finally, our main contribution is to define a NIZK for $\mathscr{L}_{\mathsf{pk},A}$.

It is worth mentioning, though, that finding an algebraic set base $\mathscr{M}$, composed by a few polynomials of low degree, is, in general, a non-cryptographic open problem, discussed more in detail in the article, see [CLPØ21a]. The task of defining an efficient NIZK for an algebraic set A is performed in two steps: finding a good base $\mathscr{M}$ (composed by a few polynomials of low degree), and define a NIZK to argue the possession of an element $\mathbf{x}$, root of all polynomials in $\mathscr{M}$. We now describe how we solve this second step, starting by noticing that we can construct our NIZK as a conjunction of NIZKs for the simpler language defined as encryptions of roots of a single polynomial f. Let us show a NIZK for the simpler language

$$\mathscr{L}_{\mathsf{pk},f} = \{[\mathbf{c}]_1 : \exists(r,\mathbf{x}), \mathsf{Enc}_{\mathsf{pk}}(\mathbf{x},r) = [\mathbf{c}]_1 \wedge f(\mathbf{x}) = 0\}. \tag{3.1}$$

Our first step is to find a suitable representation of the given polynomial f, in terms of a matrix C. A *QDR* (*Quasi-Determinantal Representation*) of a ν-variate polynomial f, is a matrix C with the following properties.

Affine map: every entry is an affine map in $C_{ij} : \mathbb{Z}_p^\nu \to \mathbb{Z}_p$. We write $C_{ij}(\mathbf{X}) = \sum_{k=1}^{\nu} P_{kij}X_k + Q_{ij}$, for public constant P_{kij}, and Q_{ij}. To simplify notation, we will use vector/matrix format, by writing

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

Determinantal representation: the determinant of C is equal to the polynomial f: $\det(C(\mathbf{x})) = f(\mathbf{x})$. In particular, this requirement implies that $C(\mathbf{x})$ is singular if and only if $f(\mathbf{x}) = 0$.

First column dependence: Write $C(\mathbf{x}) = [h(\mathbf{x})||T(\mathbf{x})]$, where h is the first column of C. For any root $\mathbf{x}$ of f, $h(\mathbf{x})$ must be in the space generated by the columns of $T(\mathbf{x})$.

If we omit the third condition, we recover the notion of *determinantal representation*: a well-studied notion in algebraic geometry, see [CLPØ21a] appendix B.1 for a more precise discussion. We define our NIZK for the language described in Eq. (3.1), by solving two subtasks: (i) finding a QDR of f, (ii) define a NIZK to prove that a ciphertext decrypts to an element $\mathbf{x}$ such that $\det(C(\mathbf{x})) = 0$ for a given QDR C.

The step (i) can be performed using ABP Definition 4. Given an ABP (V,E) for computing f, we show how to compute a QDR C of f, of size $(|V|-1) \times (|V|-1)$, from the adjacency matrix of the ABP. Recall that the adjacency matrix of a graph (V,E), labeled by ϕ, is a $|V| \times |V|$ matrix, defined as having 0 in entry i,j if $(i,j) \notin E$ and having $\phi(i,j)$ if $i,j \in E$. The computation of C is based on the methodology in [IK00, IK02]. Specifically, a QDR for a polynomial f computed by an ABP (V,E,s,t,ϕ) is obtained by

1. taking the adjacency matrix of the ABP and subtracting the identity matrix from it,
2. removing the row corresponding to t and the column corresponding to s,
3. transposing the matrix you get so far.

See [IK02], Lemma 1, for a proof of why the procedure just described outputs a QDR of f.

For instance, applying this procedure to the ABP in Fig. 2.1, we get $\mathsf{IK}(X,Y)$ QDR of $F(X,Y) = X^3 + aX + b - Y^2$, defined as

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

To solve (ii) we introduced the CED assumption (*Computational Extended Determinant*): a weaker version of the $\mathsf{ExtKerMDH}$ assumption [CH20]. In a nutshell, the CED assumption states that given a uniformly random group element $[e]_2$, it is hard to compute a full-rank $[C]_1 \in \mathbb{G}_1^{l \times l}$ and vectors $[\gamma]_1 \in \mathbb{G}_1^l$ and $[\delta]_2 \in \mathbb{G}_2^{l-1}$ such that

$$[\gamma]_1 \bullet [1]_2 + [C]_1 \bullet [\begin{smallmatrix} e \\ \delta \end{smallmatrix}]_2 = [0]_T. \tag{3.2}$$

Definition 10 (**CED assumption.**) *The* CED *assumption holds in* $\mathbb{G}_2$ *relative to* Pgen, *if for all PPT adversaries* $\mathscr{A}$, *the following probability is negligible:*

$$\Pr\left[\begin{array}{c|c} [C]_1 \in \mathbb{G}_1^{k \times k} \wedge [\gamma]_1 \in \mathbb{G}_1^k \wedge [\delta]_2 \in \mathbb{G}_2^{k-1} \wedge & \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda), e \leftarrow_{\$} \mathbb{Z}_p, \\ \gamma + C(\begin{smallmatrix} e \\ \delta \end{smallmatrix}) = \mathbf{0} \wedge \mathrm{rk}(C) = k & ([\gamma, C]_1, [\delta]_2) \leftarrow \mathscr{A}(\mathsf{p}, [e]_2) \end{array} \right].$$

To see how CED relates to the $\mathsf{ExtKerMDH}$ assumption, Definition 8, just notes that any adversaries that breaks CED can break $\mathsf{ExtKerMDH}$, by setting $[\mathbf{C}]_1 = [\gamma || C]_1$. Note that the vice versa is not true: breaking $\mathsf{ExtKerMDH}$ is not equivalent to breaking CED. An $\mathsf{ExtKerMDH}$ adversary can be successful if $[C]_1$ is not full-rank, but $[\gamma || C]_1$ is. In this case, though, the adversary will not be successful against the CED assumption. Later in this section, we explain the advantages of reducing soundness to CED and not to $\mathsf{ExtKerMDH}$.

If the prover is honest, then $C(\mathbf{x})$ is singular. We show that the prover can efficiently compute $[\gamma]_1, [\delta]_2$ such that Eq. (3.2) is satisfied, relying on the *First Column Dependence* property. Let $[h(\mathbf{x}) || T(\mathbf{x})] = C(\mathbf{x})$ be a $l \times l$ QDR of a polynomial f. For each root $\mathbf{x}$ of f, there exists $\mathbf{w} \in \mathbb{Z}_p^{l-1}$ such that $h(\mathbf{x}) = T(\mathbf{x})\mathbf{w}$. The prover samples a random vector $\mathbf{y} \in \mathbb{Z}_p^{l-1}$ and computes $\gamma = T(\mathbf{x})\mathbf{y}$. Note that γ is computed indipendently from e. Then, the prover computes $\mathbf{w}$ solving the linear system $h(\mathbf{x}) = T(\mathbf{x})\mathbf{w}$. And

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

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

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

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

Figure 3.1: The NIZK for $\mathscr{L}_{\mathsf{pk},f}$, where $C(X)$ is a QDR of f, included in `lpar`.

lastly, it computes $[\delta]_2 = -\mathbf{w}[e]_2 + \mathbf{y}[1]_2$. We denote with comp the algorithm described here, used by the prover to compute $[\gamma]_1, [\delta]_2$, on input $(\mathsf{p}, [e]_2, \mathbf{x}, \mathbf{C}(\mathbf{X}))$.

To preserve zero-knowledge, the prover computes $[\mathsf{ct}^{\gamma}]_1$: encryption of $[\gamma]_1$, under fresh randomness. The verifier can homomorphically check an encrypted version of Eq. (3.2), relying on the homomorphic property of the encryption scheme and the *Affine Map* property of C. The prover additionally sends a vector $[\mathbf{z}]_2$ of elements in $\mathbb{G}_2$. $[\mathbf{z}]_2$ is used by the verifier to annihilate encryption randomizers while checking Eq. (3.2) on ciphertexts. Thus, having the statement $[\mathsf{ct}]_1$, an encryption of $\mathbf{x}$, and the proof $\pi = ([\mathsf{ct}^{\gamma}]_1, [\delta, \mathbf{z}]_2)$, the verifier homomorphically checks that Eq. (3.2) holds. For completeness, the resulting NIZK argument is depicted in Fig. 3.1.

Now, assume by contradiction that a malicious verifier can compute a valid proof for a false statement: an encryption of a vector $\mathbf{x}$ such that $C(\mathbf{x})$ is full rank. A reduction can decrypt the statement and ct^{γ}, and output the full rank $[C(\mathbf{x})]_1$ and vectors $[\gamma]_1, [\delta]_2$, breaking the CED assumption. Therefore, if the CED assumption and an encrypted version of Eq. (3.2) holds, then we can assume that $\det(C(\mathbf{x})) = 0$.

Finally, from the *determinantal representation* property and $\det(C(\mathbf{x})) = 0$, the verifier concludes that $\mathbf{x}$ is a root of f. Or, equivalently, the NIZK argument is computationally sound under the CED assumption.

Because checking the rank of $[C(\mathbf{x})]_1$ is a hard task in general, unless C has some spe-

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

Table 3.1: Comparison of falsifiable NIZKs for Boolean circuit satisfiability: the Groth-Sahai proof, as optimized by Ghadafi et al. [GSW09], and the new NIZK from [CLPØ21a], Section 8.1. Here, $\lvert\mathbb{G}_\iota\rvert$ is the length of one element from $\mathbb{G}_\iota$

cific restrictions, the CED assumption is a computational yet non-falsifiable assumption. In the paper, we show sufficient conditions on C for the underlying CED assumption to be falsifiable, see [CLPØ21a], Section 10. We also show many interesting applications, in the form of NIZK constructions, more efficient than the previous state-of-the-art, whose security is based on falsifiable CED. We show better security conditions for the general case, where the soundness of our NIZK is reduced to a non-falsifiable version of CED. Namely, we reduce (in the standard model) the CED assumption to a very plausible and standard gap assumption [OP01]: "KerMDH is hard in $\mathbb{G}_2$ even if CDH is easy in $\mathbb{G}_1$". We emphasize that, despite being non-falsifiable, our gap assumption is very natural and plausible. Notably, it is much more desirable than knowledge assumptions or the use of idealized models. No reductions in the plain model to any standard assumption are known for the more general ExtKerMDH assumption.

The article shows interesting examples of languages for which we achieve significant efficiency improvements compared to the most optimized variant of Groth-Sahai NIZK. As an example, we report Table 2 from [CLPØ21a], showing how we improve in efficiency for the language of boolean circuit-satisfiability. In Table 3.1, we compare our NIZK for circuit satisfiability with the optimized Groth-Sahai proof for Boolean circuits by Ghadafi et al. [GSW09]. We consider circuits with m wires and n gates. When comparing efficiency, one should consider that the size of an element in $\mathbb{G}_2$ is usually twice the size of an element in $\mathbb{G}_1$. Moreover $\mathfrak{e}_2$, the time to perfom a multiplication in $\mathbb{G}_2$ is about twice $\mathfrak{e}_1$, the time for the multiplication in $\mathbb{G}_1$. We indicate with $\mathfrak{p}$ the time to perform a pairing operation. As shown in the table, the new NIZK, with soundness on falsifiable CED, has 3 times shorter commitments, 20% shorter arguments, and 1.84 times smaller prover's and verifier's computation. Importantly, as already stated in Section 1.12, the framework defined in [CH20], and in the current paper, is so far the only known way of improving in efficiency over Groth-Sahai.

Nevertheless, the framework itself is a significant contribution on its own. We leave it as an open question if there are other languages where our framework outperforms Groth-Sahai NIZK in efficiency. The third article of this thesis [LP22] partially answers to this question positively.

3.2 Article II

We here list technical details about all the contributions in [GKP22b]. See Section 1.12 for discussion and background on the contributions in this article.

A new NIWI in the plain model. Our starting point is the NIZK proof for algebraic languages in [CH20]. Couteau-Hartmann NIZK proofs are defined through a compiler applied to a linear Sigma protocol for algebraic languages, depicted in Fig. 2.3. Let us start by describing the Sigma protocol for the algebraic language

$$\mathscr{L}_{\Gamma,\theta} = \{[\mathtt{x}]_1 \in \mathbb{G}^l : \exists \mathtt{w} \in \mathbb{Z}_p^k : [\Gamma(\mathtt{x})]_1 \mathtt{w} = [\theta(\mathtt{x})]_1\}.$$

The prover samples randomness $\mathbf{r}$ and computes the first message $[\mathbf{a}]_1 = [\Gamma(\mathtt{x})]_1 \mathbf{r}$. The verifier replies with a challenge $e \leftarrow_{\$} \mathbb{Z}_p$. The prover then sends the third message $\mathbf{d} = e\mathtt{w} + \mathbf{r}$. Finally, the verifier checks if $[\Gamma(\mathtt{x})]_1 \mathbf{d} = [\theta(\mathtt{x})]_1 e + [\mathbf{a}]_1$, and outputs 1 if the previous verification equations hold. By the special soundness of the Sigma protocol, it is possible to efficiently compute a valid witness having on input two valid proofs $([\mathbf{a}]_1, e_1, \mathbf{d}_1), ([\mathbf{a}]_1, e_2, \mathbf{d}_2)$ for the same statement $\mathtt{x}$, with the same first message $[\mathbf{a}]_1$, and different challenges $e_1 \neq e_2$.

As prescribed in [CH20], to define a NIZK proof from the Sigma protocol pick two different challenges $e_1 \neq e_2$, two uniformly random elements $s_1, s_2 \leftarrow_{\$} \mathbb{Z}_p$, and publish the CRS $([s_1, s_1 e_1, s_2, s_2 e_2]_2)$. The prover computes a first message $[\mathbf{a}]_1$, as before and two masked third messages in $\mathbb{G}_2$, $[\mathbf{d}_i]_2$ for $i \in \{1, 2\}$, each answering the challenge e_i. The verifier can check both verification equations in $\mathbb{G}_T$, by

$$[\Gamma(\mathtt{x})]_1 [\mathbf{d}_i]_2 \stackrel{?}{=} [\theta(\mathtt{x})]_1 \bullet [s_i e_i]_2 + [\mathbf{a}]_1 \bullet [s_i]_2, \forall i \in 1, 2.$$

[CH20] prove perfect soundness of the NIZK proof from the special soundness of the underlying Sigma protocol.

We note that if the prover picks the CRS on its own, perfect soundness is preserved, as long as he honestly chooses $e_1 \neq e_2$. Thus, we define a NIWI proof in the plain model, modifying the Couteau-Hartmann NIZK proof. We let the prover pick its own CRS $([s_1, s_1 e_1, s_2, s_2 e_2]_2)$, with $e_1 \neq e_2$ and send it with the related proof $\pi = ([\mathbf{a}]_1, [\mathbf{d}_1, \mathbf{d}_2]_2)$ and two auxiliary group elements $([s_1, s_2]_1)$.

The verifier can now use the auxiliary group elements $([s_1, s_2]_1)$, to check that the prover was honest in picking a valid CRS with $e_1 \neq e_2$, and then run the original NIZK proof verifier. Perfect soundness of the NIWI proof follows from the special soundness of the underlying Sigma protocol, using a similar argument to the one in [CH20] for the NIZKs' soundness proof. The witness indistinguishability property is shown under a new decisional assumption. The new assumption is basically a tautological, decisional (falsifiable) assumption for the witness indistinguishability game of our NIWI proof. Following the golden standard in pairing-based cryptography, to validate the new assumption, we show that it holds in AGM, under a standard variation of a discrete logarithm assumption. We note that we used the decisional AGM [RS20] for the reduction.

See Section 1.12, for an informal comparison in efficiency with other NIWI in the plain model constructions.

We left the question of whether it is possible to prove the witness indistinguishability property of our scheme under a more standard assumption open. However, we believe this will likely not be the case. Standard decisional, pairing-based assumptions one can find in literature are all different flavours of the DDH assumption. To the best of our knowledge, they all imply the DDH assumption. Since the latter is used to prove the zero-knowledge property, we are unlikely to be able to use it to prove witness indistinguishability for a construction without a trusted setup. To be more precise, zero-knowledge of the NIZK proof (in the CRS model) in [CH20] is proven letting the simulator generate a CRS with $e_1 = e_2$. Under the DDH assumption, this simulated CRS is indistinguishable from an honestly generated one. To define our NIWI, we disallow the possibility of having valid proof for CRSs with $e_1 = e_2$. This seems to result in the impossibility of a reduction to any flavour of the DDH assumption.

We also left open the question of defining NIWI in the plain model for algebraic sets. One can always write languages generated by algebraic sets as algebraic languages; see section 9 of the first article of this thesis [CLPØ21a]. However, since we use an encryption scheme in the language definition, each statement would only have at most 1 witness. Thus, the witness indistinguishability property seems useless for languages described in [CLPØ21a].

Strong partial extractability. Recall that for a NIZK proof for the language $\mathscr{L}_{\mathscr{R}}$, the standard knowledge soundness is defined by the existence of a PPT extractor Ext that can compute a valid witness $\mathtt{w}$, for a statement $\mathtt{x}$, having a pair of $\mathtt{crs}, \mathtt{td}$ and a valid proof π for $\mathtt{x}$. In pairing-based cryptography, such a powerful extractor is unlikely to be defined under falsifiable assumptions[1]. However, there are cases where some useful, partial information about the witness can still be computed. Let f be a OWF. Balenkiy et al. [BCKL08] define the notion of *f-extractability* by requiring the existence of a

[1] As stated in Section 1.5, adding a bit-by-bit encryption of the witness would irreparably compromis the efficiency of the NIZK. Therefore, this option is left out of the current discussion.

PPT extractor Ext able to compute a function of the witness $\tilde{\mathtt{w}} = f(\mathtt{w})$. Particularly, Ext computes $\tilde{\mathtt{w}}$, for a statement $\mathtt{x}$, having $(\mathtt{crs}, \mathtt{td})$ and a proof π for $\mathtt{x}$. It is required that if the verifier accepts $(\mathtt{crs}, \mathtt{x}, \pi)$, then $\tilde{\mathtt{w}} = f(\mathtt{w})$ and $(\mathtt{x}, \mathtt{w}) \in \mathscr{R}$. In pairing-based cryptography, typically f is the scalar multiplication in one of the groups, as shown in [BCKL08] for Groth-Sahai NIZK proofs. Full-extractability seems out of reach due to the lack of a trapdoor to compute the discrete logarithm[2].

We note that Couteau-Hartmann NIZK proofs for algebraic languages $\mathscr{L}_{\Gamma,\theta}$ are $[\cdot]_2$-extractable. Moreover, we show that, as opposed to Groth-Sahai proofs, the partial witness $[\mathtt{w}]_2$ can be used to check membership, in the case of NIZK proofs in [CH20]. In fact $[\mathtt{x}]_1$ is a true statement if and only if $[\theta(\mathtt{x})]_1 \bullet [1]_2 = [\blacksquare(\mathtt{x})]_1 \bullet [\mathtt{w}]_2$. Lastly, despite its usage in checking membership, the partial witness $\tilde{\mathtt{w}} = [\mathtt{w}]_2$ cannot be considered a full witness. Notably, there does not exist a PPT adversary able to compute a valid proof, on input $(\mathtt{x}, \tilde{\mathtt{w}})$, but not $(\mathtt{w})$. We define *strong f-extractability* as f-extractability, and the possibility of deciding membership, but not computing a proof with $\tilde{\mathtt{w}}$. We show that Couteau-Hartmann NIZK proofs are strong $[\cdot]_2$-extractable, under falsifiable assumptions.

Impossibility of semantic extractability. In the case of full extractability, the extractor is required to compute the witness $\mathtt{w}$. When we consider classic knowledge soundness definitions, depicted in Section 2.7, we note that they differ in two points. For black-box knowledge soundness, the extractor has to be universal: there must exist a single extractor that works well with each PPT prover. In a sense, the black-box extractor only relies on the semantic property of its inputs $(\mathtt{crs}, \mathtt{td}, \mathtt{x}, \pi)$ to be a tuple that makes the verifier accept. On the contrary, for white-box knowledge soundness, one requires the extractor to be dependent on one specific, potentially malicious, prover, represented as a PPT. Thus, for each PPT, potentially malicious, prover $\mathscr{A}$ there exists a prover-dependent extractor $\mathsf{Ext}_{\mathscr{A}}$ that computes the witness, using $\mathscr{A}$'s code as additional information.

Moreover, the two definitions differ in another aspect. As additional input, the white-box extractor receives the entire string of random coins which $\mathscr{A}$ used to compute its output. In the black-box case, the extractor does not see any portion of the string of the random coins.

We define a new notion of extraction for NIZK called *semantic knowledge soundness*. First, we consider the adversary's randomness as an input from a specific distri-

[2] To achieve the extraction of the full witness [AHK20] defined the groups in a way that such a trapdoor for computing discrete logarithm exists. However, no such trapdoors are known in the case of elliptic-curve groups used in real applications.

bution. Once we clarify that, we can associate a function with each adversary. For each Turing machine $\mathscr{A}$, we say that $\mathscr{A}$ implements the function f that associates each tuple of inputs and random string to the related (deterministic) output computed by $\mathscr{A}$. We require that adversaries implementing the same function must have the same extractor. Moreover, we allow the flexibility to split the random coins into two different strings, enabling the semantic extractor to see only one of the two portions. This choice results in a flexible definition, allowing gradually more powerful extractors to be defined as the randomness they are allowed to see grows.

We investigate how this new notion relates to the classic notions of extraction. We show how semantic knowledge soundness is a general definition that recovers the two classic notions of extraction as particular cases. We prove that if we allow the extractor to see all the randomness, semantic extraction is equivalent to white-box extraction. This equivalence suggests that to bound the extractor to the function implemented by the adversary is the right choice, compared to binding it to the specific Turing machine. Although the two notions are equivalent, it is clear that one can use a semantic extractor in a reduction as a Turing machine per se, without being forced to generate its input using the specific adversary the extractor was designed for. This is a crucial property of semantic extraction that we use in our impossibility result. Then, we show that black-box extraction trivially implies semantic extraction. We also prove a slightly weaker implication in the other direction.

Finally, we show that semantic extraction is impossible for the [CH20] NIZK argument, depicted here in Fig. 2.4. More precisely, we show that no semantic extractor that sees only a portion of the random coins can successfully compute a witness. To understand the intuition behind this impossibility result, we start by noticing that statement, proof, and CRS consist only of group elements. At the same time, the witness and the CRS trapdoor are defined as elements in $\mathbb{Z}_p$. Intuitively, soundness relies on the hardness of discrete logarithm in $\mathbb{G}_1$ and $\mathbb{G}_2$ and on the property of asymmetric pairings of not admitting efficient isomorphisms between the two source groups[3].

Let us look at the computation of the NIZK argument proof by the honest prover, focusing on the element $[\mathbf{d}]_2 = \mathtt{w}[e]_2 + \mathbf{r}[1]_2$, where $\mathbf{r}$ is the prover randomness. We focus on $[\mathbf{d}]_2$ because it is the only part where the prover uses the witness. Suppose a semantic extractor can compute $\mathtt{w}$ from $[\mathbf{d}]_2$, the CRS trapdoor e and the other group elements composing the proof and the statement. Now, one can observe that, from a semantic point of view, there is no difference between the honestly computed $[\mathbf{d}]_2$ and

[3] In fact, it is easy to show knowledge soundness in AGM, from this simple assumptions only, as we do in this paper.

the case where the CRS trapdoor e is used to maliciously compute $[\mathbf{d}]_2$ as $[\mathtt{w}]_2 e + \mathbf{r}[1]_2$. Notably, the semantic extractor should be able to recover $\mathtt{w}$ even when the proof is dishonestly computed using $\mathtt{td}$. Lastly, we note that if such a semantic extractor exists, we can exploit it to break the discrete logarithm. After embedding the challenge into $\mathtt{w}$, the reduction would sample e by itself, dishonestly compute $[\mathbf{d}]_2$ and then invoke the semantic extractor to compute $\mathtt{w}$, which contains the discrete logarithm of the challenge.

The reduction sketched above does not work correctly if the semantic extractor can see all the adversary randomness (e.g., when the semantic extractor is equivalent to a classic white-box extractor). Although, as soon as some randomness is hidden from the extractor, the reduction can correctly embed the discrete logarithm challenge in that hidden part of the execution. Thus, we have shown that semantic extraction is impossible unless the extractor sees all the adversary randomness. Particularly, black-box knowledge soundness is impossible. But we argue that our impossibility result rules out many extractors more powerful than the classic black-box ones.

A different interpretation of our impossibility result is that the proof only shows the knowledge of $[\mathtt{w}]_1, [\mathtt{w}]_2$. To go a step further and argue the knowledge of the full witness $\mathtt{w}$, from the knowledge of $[\mathtt{w}]_1, [\mathtt{w}]_2$, one must rely on a knowledge assumption (a suitable assumption is defined in [ABLZ17], or in the journal version [ALSZ21]) or one must use idealized models.

Our result also suggests that computing the witness for many algebraic languages, given a statement $[\mathtt{x}]_1$, is as hard as extracting a witness from the statement and a valid proof under a CRS for which we know the trapdoor. More precisely, note that the hardness of the algebraic languages is based on the hardness of computing the discrete logarithm. We base the impossibility result on the hardness of symmetric discrete logarithm: it is computationally intractable to compute x, on input $[x]_1$ and $[x]_2$. Therefore, our result states that either there is a gap between the hardness of discrete logarithm and the hardness of the symmetric discrete logarithm or computing $\mathtt{w}$ from $[\mathtt{x}]_1$ is as hard as computing $\mathtt{w}$ from $(e, \mathbf{r}, [\mathtt{x}]_1, \pi = ([\mathbf{a}]_1, [\mathbf{d}]_2))$ where π is computed by the honest prover on random coins $\mathbf{r}$ (and $\mathsf{V}([e]_2, [\mathtt{x}]_1, \pi) = 1$).

We believe that the new view of considering semantic adversaries is, by itself, a strong contribution of this article. Note that semantic techniques are already implicitly used in cryptography, everytime a non-black-box technique is used in the standard model. For instance, we can think of the non-black-box zero-knowledge simulator in [BLV03] as a semantic simulator. We left as future work the task of exploring if the idea could be applied to prove results in other branches of cryptography.

3.3 Article III

We describe here the main contribution of the paper [LP22]: a new *set (non-)membership* NIZK argument, see Sections 1.11 and 1.12. Let $\mathscr{S} = \{a_1, \dots, a_m\}$ be a finite set of elements in $\mathbb{Z}_p$. Recall that an *accumulator* is a cryptographic primitive, implementing a non-zero-knowledge proof to show that an element χ is in the public set $\mathscr{S}$, see Section 2.9. Recall also that an accumulator is *universal* if it is possible to prove that χ is not an element in the set $\mathscr{S}$. Following a methodology discussed in Section 1.11, we can define a set (non-)membership NIZK, by adding zero-knowledge to a universal accumulator.

Our approach relies on using the CLPØ, framework presented in the first article of this thesis [CLPØ21a], as a very efficient zero-knowledge compiler for a new accumulator we define. The new accumulator is designed to be "friendly" with the choice of using CLPØ as a compiler. We use the term *determinantal accumulator* to emphasize this friendliness, similarly to how structure-preserving is used in the context of signatures, to emphasize friendliness with the Groth-Sahai framework in [BCKL08, AFG$^+$16, FHS19]. See the article [LP22], for the precise definition of a determinantal accumulator.

Let us discuss the details of our accumulator construction. We focus on the case of set membership, referring to the paper for the non-membership case. Inspired by [Ngu05], given the public set $\mathscr{S}$ and an element $\chi \in \mathscr{S}$, we define the polynomials

$$\mathbf{Z}_{\mathscr{S}}(\Sigma) = \prod_{a \in \mathscr{S}} (\Sigma - a) \quad ; \quad Q(\Sigma) = \prod_{a \in \mathscr{S}, a \neq \chi} (\Sigma - a).$$

We construct the accumulator CRS with group elements $[1, \sigma^1, \dots, \sigma^m]_1$, where $\sigma \leftarrow_{\$} \mathbb{Z}_p$ is a secret trapdoor and we require $|\mathscr{S}| \leq m$. Nguyen's solution was to let the prover output group elements $[\mathsf{q} = Q(\sigma)]_1, [\chi]_2$. The verifier can verify the correctness of the accumulator proof, using pairings, by checking that

$$[\mathsf{q}]_1 \bullet [\sigma - \chi]_2 - [\mathbf{Z}_{\mathscr{S}}(\sigma)]_1 \bullet [1]_2 = [0]_T.$$

We choose a different approach. We define matrices

$$\mathbf{C}_\Sigma(X, Q) := \begin{bmatrix} \Sigma - X & -1 \\ -\mathbf{Z}_{\mathscr{S}}(\Sigma) & Q \end{bmatrix}_1 \quad \text{and} \quad \mathbf{C}_\sigma(\chi, \mathsf{q}) = \begin{bmatrix} \sigma - \chi & -1 \\ -\mathbf{Z}_{\mathscr{S}}(\sigma) & \mathsf{q} \end{bmatrix}_1.$$

We let the prover output $[\chi, \mathsf{q}]_1$. The verifier will accept the proof if $\det(\mathbf{C}_\sigma(\chi, \mathsf{q})) = 0 (= \mathsf{q}(\sigma - \chi) - \mathbf{Z}_{\mathscr{S}}(\sigma))$. Now, the attentive reader will not have missed that checking if $\mathbf{C}_\sigma(\chi, \mathsf{q})$ is singular, given its entries only as group elements, is a computationally intractable task. Therefore, following the framework developed in the first article of this thesis, [CLPØ21a], we let the prover output additional hints $[\gamma]_1, [\delta]_2$ such that the CLPØ style verification equation

$$[\gamma]_1 \bullet [1]_2 + \left[\begin{smallmatrix} \sigma - \chi & -1 \\ -\mathbf{Z}_{\mathscr{S}}(\sigma) & \mathsf{q} \end{smallmatrix}\right]_1 \bullet \left[\begin{smallmatrix} e \\ \delta \end{smallmatrix}\right]_2 = [0]_T,$$

holds. Here $[e]_2$ is added as an additional element to the CRS. Thus, our determinantal accumulator's proof of $\chi \in \mathscr{S}$ contains $[\gamma, \mathsf{q}]_1, [\delta]_2$.

Unfortunately, this construction is not secure: since the prover outputs χ only as a group element, it can compute it as a function of σ and easily forge valid proof for $\chi \notin \mathscr{S}$. A possible solution requires that the prover outputs χ as an integer. However, this solution would not be satisfactory. Remember that we want to use the accumulator to define a CLPØ set (non-)membership NIZK. After decryption, the soundness reduction of such a NIZK argument can only recover $[\chi]_1$ as a group element. Therefore, we need our accumulator to be $[\cdot]_1$-*collision resistant*: no PPT adversary should be able to compute a group element $[\chi]_1$ and a valid accumulator proof for any $\chi \notin \mathscr{S}$. Previous falsifiable constructions [BCKL08, AN11, DGP$^+$19] had the same issue, which they solved by a "knowledge equation". However, introducing a new equation is a major source of inefficiency. See Section 1.11 for a discussion about previous falsifiable set membership NIZKs.

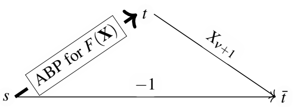
As a novelty, we define a $[\cdot]_1$-collision-resistant accumulator by cleverly using a new trapdoor τ, without resorting to another equation. Given an accumulator proof $([\gamma, \chi, \mathsf{q}]_1, [\delta]_2)$, the accumulator verifier checks if

$$[\delta]_1 \bullet [1]_2 = [\mathbf{C}_{\sigma,\tau}(\chi, \mathsf{q})]_1 \bullet \left[\begin{smallmatrix} e \\ \delta \end{smallmatrix}\right]_2,$$

where $\mathbf{C}_{\sigma,\tau}(\chi, \mathsf{q})$ is still a 2×2 matrix, but slightly different than the $\mathbf{C}_\sigma(\chi, \mathsf{q})$ defined previously. Notably, the $[\cdot]_1$-collision-resistant accumulator is as efficient as the (not $[\cdot]_1$-collision resistant) one described above.

The security of the accumulator scheme is based on new falsifiable assumptions we introduced in the paper, one for the membership case and one for non-membership. To motivate the introduction of new assumptions, we again show that they can be reduced in AGM to a standard flavour of a discrete logarithm assumption. The AGM assumptions' proof of security is one of the most technically involved contribution of the article.

Lastly, we use CLPØ framework for NIZK as a compiler to add zero-knowledge



$$\left(\begin{array}{ccccc|c} \multicolumn{5}{c|}{\mathsf{IK}_F(X_1,\dots,X_\nu)} & \begin{matrix}0\\0\\\vdots\\0\\-1\end{matrix} \\ \hline -1 & 0 & \dots & 0 & 0 & X_{\nu+1}\end{array}\right)$$

Figure 3.2: ABP $\overline{\mathsf{abp}}$ for the $g(\mathbf{X},X_{\nu+1}) = f(\mathbf{X})X_{\nu+1} - 1$ and the QDR matrix $\mathsf{IK}_g(\mathbf{X},X_{\nu+1})$.

to the determinantal accumulator. We use the Elgamal encryption scheme from Section 2.5 to achieve the best efficiency. We aim to define a set membership NIZK, to show that $[\mathsf{ct}]_1$ is an encryption of an element in the public set $\mathscr{S}$. We let the prover encrypt $[\gamma,\mathsf{q}]_1$ under fresh randomness and compute additional elements $[\mathbf{z}]_2$, following the methodology we have explained in the first article of this thesis. The verifier homomorphically checks the accumulator's CLPØ style verification equation, using the elements $[\mathbf{z}]_2$ to annihilate encryption randomizers. Computational adaptive soundness of the NIZK argument follows from the $[\cdot]_1$-collision-resistance of the accumulator scheme. Computational zero-knowledge is shown under the IND-CPA security of the Elgamal encryption scheme.

A general non-membership NIZK. As an independent result, we show how to define a NIZK to show that a given encryption decrypts to an element that is not a root of a polynomial f. Given an ABP to compute the ν-variate polynomial f, we show how to automatically define an ABP to compute a $(\nu+1)$-variate polynomial g such that: (i) if $f(\mathbf{x}) = 0$ then $g(\mathbf{x},x_{\nu+1}) \neq 0$ for each $x_{\nu+1} \in \mathbb{Z}_p$ and (ii) if $f(\mathbf{x}) \neq 0$, then it is easy to find a $x_{\nu+1} \in \mathbb{Z}_p$ such that $g(\mathbf{x},x_{\nu+1}) = 0$. We then use this ABP to define a CLPØ NIZK for the polynomial g, following the methodology of the first article of this thesis. The ABP for the new non-membership NIZK is depicted in Fig. 3.2.

Bibliography

[ABLZ17] Behzad Abdolmaleki, Karim Baghery, Helger Lipmaa, and Michal Zajac. A subversion-resistant SNARK. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part III*, volume 10626 of *LNCS*, pages 3–33. Springer, Heidelberg, December 2017. doi:10.1007/978-3-319-70700-6_1.

[ACR21] Thomas Attema, Ronald Cramer, and Matthieu Rambaud. Compressed Σ-protocols for bilinear group arithmetic circuits and application to logarithmic transparent threshold signatures. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 526–556. Springer, Heidelberg, December 2021. doi:10.1007/978-3-030-92068-5_18.

[AFG$^+$16] Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. *Journal of Cryptology*, 29(2):363–421, April 2016. doi:10.1007/s00145-014-9196-7.

[AHK20] Thomas Agrikola, Dennis Hofheinz, and Julia Kastner. On instantiating the algebraic group model from falsifiable assumptions. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part II*, volume 12106 of *LNCS*, pages 96–126. Springer, Heidelberg, May 2020. doi:10.1007/978-3-030-45724-2_4.

[ALSZ21] Behzad Abdolmaleki, Helger Lipmaa, Janno Siim, and Michal Zajac. On subversion-resistant SNARKs. *Journal of Cryptology*, 34(3):17, July 2021. doi:10.1007/s00145-021-09379-y.

[AN11] Tolga Acar and Lan Nguyen. Revocation for delegatable anonymous credentials. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 423–440.

Springer, Heidelberg, March 2011. doi:10.1007/978-3-642-19379-8_26.

[ATSM09] Man Ho Au, Patrick P. Tsang, Willy Susilo, and Yi Mu. Dynamic universal accumulators for DDH groups and their application to attribute-based anonymous credential systems. In Marc Fischlin, editor, *CT-RSA 2009*, volume 5473 of *LNCS*, pages 295–308. Springer, Heidelberg, April 2009. doi:10.1007/978-3-642-00862-7_20.

[BB04] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 56–73. Springer, Heidelberg, May 2004. doi:10.1007/978-3-540-24676-3_4.

[BBB$^+$18] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society Press, May 2018. doi:10.1109/SP.2018.00020.

[BCCT12] Nir Bitansky, Ran Canetti, Alessandro Chiesa, and Eran Tromer. From extractable collision resistance to succinct non-interactive arguments of knowledge, and back again. In Shafi Goldwasser, editor, *ITCS 2012*, pages 326–349. ACM, January 2012. doi:10.1145/2090236.2090263.

[BCKL08] Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. P-signatures and noninteractive anonymous credentials. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 356–374. Springer, Heidelberg, March 2008. doi:10.1007/978-3-540-78524-8_20.

[Bd94] Josh Cohen Benaloh and Michael de Mare. One-way accumulators: A decentralized alternative to digital sinatures (extended abstract). In Tor Helleseth, editor, *EUROCRYPT'93*, volume 765 of *LNCS*, pages 274–285. Springer, Heidelberg, May 1994. doi:10.1007/3-540-48285-7_24.

[BD19] Razvan Barbulescu and Sylvain Duquesne. Updating key size estimations for pairings. *Journal of Cryptology*, 32(4):1298–1336, October 2019. doi:10.1007/s00145-018-9280-5.

[BdM93] Josh Benaloh and Michael de Mare. One-Way Accumulators: A Decentralized Alternative to Digital Signatures. In Tor Helleseth, editor, *EUROCRYPT 1993*, volume 765 of *LNCS*, pages 274–285, Lofthus, Norway, May 23–27, 1993. Springer, Heidelberg, 1994.

[BF01] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229. Springer, Heidelberg, August 2001. `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*, pages 103–112. ACM Press, May 1988. `doi:10.1145/62212.62222`.

[BFP21] Balthazar Bauer, Georg Fuchsbauer, and Antoine Plouviez. The one-more discrete logarithm assumption in the generic group model. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part IV*, volume 13093 of *LNCS*, pages 587–617. Springer, Heidelberg, December 2021. `doi:10.1007/978-3-030-92068-5_20`.

[BG99] Amos Beimel and Anna Gál. On Arithmetic Branching Programs. *J. Comput. Syst. Sci.*, 59(2):195–220, 1999.

[BLL00] Ahto Buldas, Peeter Laud, and Helger Lipmaa. Accountable certificate management using undeniable attestations. In Dimitris Gritzalis, Sushil Jajodia, and Pierangela Samarati, editors, *ACM CCS 2000*, pages 9–17. ACM Press, November 2000. `doi:10.1145/352600.352604`.

[BLL02] Ahto Buldas, Peeter Laud, and Helger Lipmaa. Eliminating Counterevidence with Applications to Accountable Certificate Management. *Journal of Computer Security*, 10(3):273–296, 2002.

[BLV03] Boaz Barak, Yehuda Lindell, and Salil P. Vadhan. Lower bounds for non-black-box zero knowledge. In *44th FOCS*, pages 384–393. IEEE Computer Society Press, October 2003. `doi:10.1109/SFCS.2003.1238212`.

[BMV08] Emmanuel Bresson, Jean Monnerat, and Damien Vergnaud. Separation results on the “one-more” computational problems. In Tal Malkin, editor, *CT-RSA 2008*, volume 4964 of *LNCS*, pages 71–87. Springer, Heidelberg, April 2008. `doi:10.1007/978-3-540-79263-5_5`.

[BNPS03] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3):185–215, June 2003. doi:10.1007/s00145-002-0120-1.

[BOV03] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 299–315. Springer, Heidelberg, August 2003. doi:10.1007/978-3-540-45146-4_18.

[BP97] Niko Barić and Birgit Pfitzmann. Collision-Free Accumulators and Fail-Stop Signature Schemes without Trees. In Walter Fumy, editor, *EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 480–494, Konstanz, Germany, 11–15 May 1997. Springer, Heidelberg.

[BP04] Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 273–289. Springer, Heidelberg, August 2004. doi:10.1007/978-3-540-28628-8_17.

[BP15] Nir Bitansky and Omer Paneth. ZAPs and non-interactive witness indistinguishability from indistinguishability obfuscation. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 401–427. Springer, Heidelberg, March 2015. doi:10.1007/978-3-662-46497-7_16.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, November 1993. doi:10.1145/168588.168596.

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

[BV98] Dan Boneh and Ramarathnam Venkatesan. Breaking RSA may not be equivalent to factoring. In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 59–71. Springer, Heidelberg, May / June 1998. doi:10.1007/BFb0054117.

[CCH$^+$19] Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-Shamir: from practice to theory. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1082–1090. ACM Press, June 2019. `doi:10.1145/3313276.3316380`.

[CCs08] Jan Camenisch, Rafik Chaabouni, and abhi shelat. Efficient protocols for set membership and range proofs. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 234–252. Springer, Heidelberg, December 2008. `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 Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 174–187. Springer, Heidelberg, August 1994. `doi:10.1007/3-540-48658-5_19`.

[CGS07] Nishanth Chandran, Jens Groth, and Amit Sahai. Ring signatures of sub-linear size without random oracles. In Lars Arge, Christian Cachin, Tomasz Jurdzinski, and Andrzej Tarlecki, editors, *ICALP 2007*, volume 4596 of *LNCS*, pages 423–434. Springer, Heidelberg, July 2007. `doi:10.1007/978-3-540-73420-8_38`.

[CH20] Geoffroy Couteau and Dominik Hartmann. Shorter non-interactive zero-knowledge arguments and ZAPs for algebraic languages. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 768–798. Springer, Heidelberg, August 2020. `doi:10.1007/978-3-030-56877-1_27`.

[CLPØ21a] Geoffroy Couteau, Helger Lipmaa, Roberto Parisella, and Arne Tobias Ødegaard. Efficient NIZKs for algebraic sets. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part III*, volume 13092 of *LNCS*, pages 128–158. Springer, Heidelberg, December 2021. `doi:10.1007/978-3-030-92078-4_5`.

[CLPØ21b] Geoffroy Couteau, Helger Lipmaa, Roberto Parisella, and Arne Tobias Ødegaard. Efficient NIZKs for algebraic sets. Cryptology ePrint Archive, Report 2021/1251, 2021. `https://eprint.iacr.org/2021/1251`.

[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, 2018. https://eprint.iacr.org/2018/1248.

[CPV20] Michele Ciampi, Roberto Parisella, and Daniele Venturi. On adaptive security of delayed-input sigma protocols and fiat-shamir NIZKs. In Clemente Galdi and Vladimir Kolesnikov, editors, *SCN 20*, volume 12238 of *LNCS*, pages 670–690. Springer, Heidelberg, September 2020. doi:10.1007/978-3-030-57990-6_33.

[Dam92] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *CRYPTO'91*, volume 576 of *LNCS*, pages 445–456. Springer, Heidelberg, August 1992. doi:10.1007/3-540-46766-1_36.

[den90] Bert den Boer. Diffie-Hellman is as strong as discrete log for certain primes (rump session). In Shafi Goldwasser, editor, *CRYPTO'88*, volume 403 of *LNCS*, pages 530–539. Springer, Heidelberg, August 1990. doi:10.1007/0-387-34799-2_38.

[Den02] Alexander W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 100–109. Springer, Heidelberg, December 2002. doi:10.1007/3-540-36178-2_6.

[DGP$^+$19] Vanesa Daza, Alonso González, Zaira Pindado, Carla Ràfols, and Javier Silva. Shorter quadratic QA-NIZK proofs. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part I*, volume 11442 of *LNCS*, pages 314–343. Springer, Heidelberg, April 2019. doi:10.1007/978-3-030-17253-4_11.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DHS15] David Derler, Christian Hanser, and Daniel Slamanig. Revisiting cryptographic accumulators, additional properties and relations to other primitives. In Kaisa Nyberg, editor, *CT-RSA 2015*, volume 9048 of *LNCS*, pages 127–144. Springer, Heidelberg, April 2015. doi:10.1007/978-3-319-16715-2_7.

[DT08] Ivan Damgård and Nikos Triandopoulos. Supporting non-membership proofs with bilinear-map accumulators. Cryptology ePrint Archive, Report 2008/538, 2008. https://eprint.iacr.org/2008/538.

[EG14] Alex Escala and Jens Groth. Fine-tuning Groth-Sahai proofs. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 630–649. Springer, Heidelberg, March 2014. doi:10.1007/978-3-642-54631-0_36.

[EHK$^+$13] Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, August 2013. doi:10.1007/978-3-642-40084-1_8.

[ElG84] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18. Springer, Heidelberg, August 1984.

[FHS19] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. *Journal of Cryptology*, 32(2):498–546, April 2019. doi:10.1007/s00145-018-9281-4.

[FKL18] Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Heidelberg, August 2018. doi:10.1007/978-3-319-96881-0_2.

[FS87] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987. doi:10.1007/3-540-47721-7_12.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *22nd ACM STOC*, pages 416–426. ACM Press, May 1990. doi:10.1145/100216.100272.

[GGPR13] Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645. Springer, Heidelberg, May 2013. doi:10.1007/978-3-642-38348-9_37.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *44th FOCS*, pages 102–115. IEEE Computer Society Press, October 2003. doi:10.1109/SFCS.2003.1238185.

[GK16] Shafi Goldwasser and Yael Tauman Kalai. Cryptographic assumptions: A position paper. In Eyal Kushilevitz and Tal Malkin, editors, *TCC 2016-A, Part I*, volume 9562 of *LNCS*, pages 505–522. Springer, Heidelberg, January 2016. doi:10.1007/978-3-662-49096-9_21.

[GKP22a] Chaya Ganesh, Hamidreza Khoshakhlagh, and Roberto Parisella. NIWI and new notions of extraction for algebraic languages. Cryptology ePrint Archive, Report 2022/851, 2022. https://eprint.iacr.org/2022/851.

[GKP22b] Chaya Ganesh, Hamidreza Khoshakhlagh, and Roberto Parisella. Niwi and new notions of extraction for algebraic languages. In Clemente Galdi and Stanislaw Jarecki, editors, *Security and Cryptography for Networks*, pages 687–710, Cham, 2022. Springer International Publishing.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *17th ACM STOC*, pages 291–304. ACM Press, May 1985. doi:10.1145/22145.22178.

[GO94] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, December 1994. doi:10.1007/BF00195207.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 97–111. Springer, Heidelberg, August 2006. doi:10.1007/11818175_6.

[GOS12] Jens Groth, Rafail Ostrovsky, and Amit Sahai. New Techniques for Non-interactive Zero-Knowledge. *Journal of the ACM*, 59(3), 2012.

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

[Gro10] Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of

LNCS, pages 321–340. Springer, Heidelberg, December 2010. doi:10.1007/978-3-642-17373-8_19.

[Gro16] Jens Groth. On the size of pairing-based non-interactive arguments. In Marc Fischlin and Jean-Sébastien Coron, editors, *EUROCRYPT 2016, Part II*, volume 9666 of *LNCS*, pages 305–326. Springer, Heidelberg, May 2016. doi:10.1007/978-3-662-49896-5_11.

[GS08] Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, April 2008. 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 Matthew G. Parker, editor, *12th IMA International Conference on Cryptography and Coding*, volume 5921 of *LNCS*, pages 469–494. Springer, Heidelberg, December 2009.

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

[GWC19] Ariel Gabizon, Zachary J. Williamson, and Oana Ciobotaru. PLONK: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Report 2019/953, 2019. https://eprint.iacr.org/2019/953.

[HL18] Justin Holmgren and Alex Lombardi. Cryptographic hashing from strong one-way functions (or: One-way product functions and their applications). In Mikkel Thorup, editor, *59th FOCS*, pages 850–858. IEEE Computer Society Press, October 2018. doi:10.1109/FOCS.2018.00085.

[IK00] Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st FOCS*, pages 294–304. IEEE Computer Society Press, November 2000. doi:10.1109/SFCS.2000.892118.

[IK02] Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan

Eidenbenz, and Ricardo Conejo, editors, *ICALP 2002*, volume 2380 of *LNCS*, pages 244–256. Springer, Heidelberg, July 2002. doi:10.1007/3-540-45465-9_22.

[IW14] Yuval Ishai and Hoeteck Wee. Partial garbling schemes and their applications. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *ICALP 2014, Part I*, volume 8572 of *LNCS*, pages 650–662. Springer, Heidelberg, July 2014. doi:10.1007/978-3-662-43948-7_54.

[KRR17] Yael Tauman Kalai, Guy N. Rothblum, and Ron D. Rothblum. From obfuscation to the security of Fiat-Shamir for proofs. In Jonathan Katz and Hovav Shacham, editors, *CRYPTO 2017, Part II*, volume 10402 of *LNCS*, pages 224–251. Springer, Heidelberg, August 2017. doi:10.1007/978-3-319-63715-0_8.

[KW15] Eike Kiltz and Hoeteck Wee. Quasi-adaptive NIZK for linear subspaces revisited. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 101–128. Springer, Heidelberg, April 2015. doi:10.1007/978-3-662-46803-6_4.

[Lip12a] Helger Lipmaa. Progression-free sets and sublinear pairing-based non-interactive zero-knowledge arguments. In Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 169–189. Springer, Heidelberg, March 2012. doi:10.1007/978-3-642-28914-9_10.

[Lip12b] Helger Lipmaa. Secure accumulators from euclidean rings without trusted setup. In Feng Bao, Pierangela Samarati, and Jianying Zhou, editors, *ACNS 12*, volume 7341 of *LNCS*, pages 224–240. Springer, Heidelberg, June 2012. doi:10.1007/978-3-642-31284-7_14.

[Lip13] Helger Lipmaa. Succinct non-interactive zero knowledge arguments from span programs and linear error-correcting codes. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 41–60. Springer, Heidelberg, December 2013. doi:10.1007/978-3-642-42033-7_3.

[LLX07] Jiangtao Li, Ninghui Li, and Rui Xue. Universal accumulators with efficient nonmembership proofs. In Jonathan Katz and Moti Yung, editors, *ACNS 07*, volume 4521 of *LNCS*, pages 253–269. Springer, Heidelberg, June 2007. doi:10.1007/978-3-540-72738-5_17.

[LP22] Helger Lipmaa and Roberto Parisella. Set (non-)membership nizks from determinantal accumulators. Cryptology ePrint Archive, Paper 2022/1570, 2022. https://eprint.iacr.org/2022/1570. URL: https://eprint.iacr.org/2022/1570.

[LSZ22] Helger Lipmaa, Janno Siim, and Michal Zajac. Counting vampires: From univariate sumcheck to updatable ZK-SNARK. Cryptology ePrint Archive, Report 2022/406, 2022. https://eprint.iacr.org/2022/406.

[Mau05] Ueli M. Maurer. Abstract models of computation in cryptography (invited paper). In Nigel P. Smart, editor, *10th IMA International Conference on Cryptography and Coding*, volume 3796 of *LNCS*, pages 1–12. Springer, Heidelberg, December 2005.

[Mau09] Ueli M. Maurer. Unifying zero-knowledge proofs of knowledge. In Bart Preneel, editor, *AFRICACRYPT 09*, volume 5580 of *LNCS*, pages 272–286. Springer, Heidelberg, June 2009.

[MP03] Daniele Micciancio and Erez Petrank. Simulatable commitments and efficient concurrent zero-knowledge. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 140–159. Springer, Heidelberg, May 2003. doi:10.1007/3-540-39200-9_9.

[MW98] Ueli M. Maurer and Stefan Wolf. Lower bounds on generic algorithms in groups. In Kaisa Nyberg, editor, *EUROCRYPT'98*, volume 1403 of *LNCS*, pages 72–84. Springer, Heidelberg, May / June 1998. doi:10.1007/BFb0054118.

[Nao03] Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, Heidelberg, August 2003. doi:10.1007/978-3-540-45146-4_6.

[Ngu05] Lan Nguyen. Accumulators from bilinear pairings and applications. In Alfred Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 275–292. Springer, Heidelberg, February 2005. doi:10.1007/978-3-540-30574-3_19.

[Nis91] Noam Nisan. Lower bounds for non-commutative computation (extended abstract). In *23rd ACM STOC*, pages 410–418. ACM Press, May 1991. 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 Kwangjo Kim, editor, *PKC 2001*, volume 1992 of *LNCS*, pages 104–118. Springer, Heidelberg, February 2001. doi:10.1007/3-540-44586-2_8.

[Pas13] Rafael Pass. Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 334–354. Springer, Heidelberg, March 2013. doi:10.1007/978-3-642-36594-2_19.

[PHGR13] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *2013 IEEE Symposium on Security and Privacy*, pages 238–252. IEEE Computer Society Press, May 2013. doi:10.1109/SP.2013.47.

[PS19] Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part I*, volume 11692 of *LNCS*, pages 89–114. Springer, Heidelberg, August 2019. doi:10.1007/978-3-030-26948-7_4.

[PV05] Pascal Paillier and Damien Vergnaud. Discrete-log-based signatures may not be equivalent to discrete log. In Bimal K. Roy, editor, *ASIACRYPT 2005*, volume 3788 of *LNCS*, pages 1–20. Springer, Heidelberg, December 2005. doi:10.1007/11593447_1.

[Ràf15] Carla Ràfols. Stretching groth-sahai: NIZK proofs of partial satisfiability. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 247–276. Springer, Heidelberg, March 2015. doi:10.1007/978-3-662-46497-7_10.

[RS20] Lior Rotem and Gil Segev. Algebraic distinguishers: From discrete logarithms to decisional uber assumptions. In Rafael Pass and Krzysztof Pietrzak, editors, *TCC 2020, Part III*, volume 12552 of *LNCS*, pages 366–389. Springer, Heidelberg, November 2020. doi:10.1007/978-3-030-64381-2_13.

[RZ21] Carla Ràfols and Arantxa Zapico. An algebraic framework for universal and updatable SNARKs. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 774–804, Virtual Event, August 2021. Springer, Heidelberg. doi:10.1007/978-3-030-84242-0_27.

[Sch90] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *CRYPTO'89*, volume 435 of *LNCS*, pages 239–252. Springer, Heidelberg, August 1990. doi:10.1007/0-387-34805-0_22.

[Sho97] Victor Shoup. Lower bounds for discrete logarithms and related problems. In Walter Fumy, editor, *EUROCRYPT'97*, volume 1233 of *LNCS*, pages 256–266. Springer, Heidelberg, May 1997. doi:10.1007/3-540-69053-0_18.

[SY10] Amir Shpilka and Amir Yehudayoff. *Arithmetic Circuits: A Survey of Recent Results and Open Questions*, volume 5 of *Foundations and Trends in Theoretical Computer Science*. Now Publishers Inc, December 2010.

[Val79] Leslie G. Valiant. Completeness Classes in Algebra. In *STOC 1979*, pages 249–261, Atlanta, Georgia, USA, 30 April—2 May 1979.

[VB20] Giuseppe Vitto and Alex Biryukov. Dynamic universal accumulator with batch update over bilinear groups. Cryptology ePrint Archive, Report 2020/777, 2020. https://eprint.iacr.org/2020/777.

[Zha22] Mark Zhandry. To label, or not to label (in generic groups). In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part III*, volume 13509 of *LNCS*, pages 66–96. Springer, Heidelberg, August 2022. doi:10.1007/978-3-031-15982-4_3.

Chapter 4

Efficient NIZKs for Algebraic Sets

Geoffroy Couteau
Helger Lipmaa
Roberto Parisella
Arne Tobias Ødegaard

In *ASIACRYPT 2021, Part III.* ed. by Mehdi Tibouchi and Huaxiong Wang. Vol. 13092. LNCS. Springer, Heidelberg, Dec. 2021, pp. 128–158.
`https://doi.org/10.1007/978-3-030-92078-4_5`

In this thesis, we include the full version published on ePrint. Compared to the conference version, we have corrected some minor issues, notations, inconsistencies and typos. Moreover, there are some additional contents in the appendix.

Efficient NIZKs for Algebraic Sets

Geoffroy Couteau[1], Helger Lipmaa[2], Roberto Parisella[2], and Arne Tobias Ødegaard[2]

[1] CNRS, IRIF, Université de Paris, Paris, France
[2] Simula UiB, Bergen, Norway

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.

Keywords: Algebraic branching programs, algebraic languages, algebraic sets, NIZK, pairing-based cryptography, SPHF, zero knowledge

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[3], 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 [BFI+10,EG14,Ràf15,DGP+19]. Unfortunately, the efficiency of Groth-Sahai proofs often remains unsatisfying (typically much worse than NIZKs 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,ALSZ20] 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.

[3] There have been recent developments towards provably secure Fiat-Shamir NIZKs [CCH+19].

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* [BBC+13,CC18], which 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. We call $(\boldsymbol{\theta}, \boldsymbol{\Gamma})$ the *matrix description* of the language $\mathcal{L}$. Since any NP language can be embedded into an algebraic language[4], 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.

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. [KZM+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[5].

Second, the CH-compilation produces NIZKs whose soundness reduces to an instance of the novel $\mathsf{ExtKerMDH}$ 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 $\mathsf{ExtKerMDH}$ assumption their particular instance requires.

1.1 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,Kiy20], i.e. languages of the form

[4] 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.

[5] 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.

$\{\mathsf{Com}(x_1), \ldots, \mathsf{Com}(x_n) \mid R(x_1, \ldots, x_n)\}$, 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 1) or the language of commitments to points on an elliptic curve[6], 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 [BBC$^+$13] gives a generic construction of SPHFs given the matrix description of an algebraic language), including languages describable by efficient algebraic branching programs.

1.2 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 "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 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 8.2, 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

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

Table 1. Comparison of set-membership proofs, i..e., NIZKs for $\mathcal{L}_{\mathsf{pk},F}$, where $F(X)$ is univariate, as in Lemmas 8, 9 and 10. 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, $|\mathtt{crs}| = |\mathbb{G}_2|$.

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 10	$(3d-1)\lvert\mathbb{G}_1\rvert + (3d-2)\lvert\mathbb{G}_2\rvert$	$(7d-4)\mathfrak{e}_1 + (3d-1)\mathfrak{e}_2$	$9d-2$
New solutions			
CHM NIZK + new $\boldsymbol{\Gamma}, \boldsymbol{\theta}$, Lemma 9	$2d\lvert\mathbb{G}_1\rvert + (2d-1)\lvert\mathbb{G}_2\rvert$	$(5d-3)\mathfrak{e}_1 + 4d\mathfrak{e}_2$	$7d-1$
New NIZK, Lemma 8	$2d\lvert\mathbb{G}_1\rvert + (2d-1)\lvert\mathbb{G}_2\rvert$	$\leq 3d\mathfrak{e}_1 + (4d-2)\mathfrak{e}_2$	$7d-1$

function F_i depends on at most q of its input locations.[7] In particular, this efficiently captures arithmetic circuits, hence all NP languages.[8]

Rank-1 constraints systems (R1CS) are well-known to be powerful, since they capture *compactly* many languages of interest [GGPR13]. 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 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 10. First, we identify a large class of important cases where the underlying assumption becomes falsifiable; this includes 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

[7] 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 $\forall j, F_j(\boldsymbol{\chi}) = 0$.

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

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 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).

1.3 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 [CLO15] 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 $\mathcal{A}$,

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

Since each ideal of $\mathcal{F}[\boldsymbol{X}]$ is finitely generated [CLO15], 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 8.1.[9] 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.

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 $\mathtt{x} \in \mathcal{L}_{\mathsf{pk},F}$. 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 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 10.2 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 $\mathit{\Pi}$ 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 $\mathit{\Pi}$ 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

[9] 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.

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 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, [BCTV14]. 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 6.2, 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 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.

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 [GPS08], 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 [EHK$^+$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 $\mathcal{P}_\nu := \{[a_0]_1 + \sum_{i=1}^{\nu} [a_i]_1 X_i : a_i \in \mathbb{Z}_p \text{ for } i \in [0, \nu]\} \subset \mathbb{G}_1[\boldsymbol{X}]$ 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,BBC$^+$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 $\mathsf{ExtKerMDH}$ [CH20] generalizes the well-known KerMDH assumption [MRV16]. (Appendix A.1 defines KerMDH.) We also define in parallel a new, slightly weaker version of this assumption, CED (*computational extended determinant*).

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

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

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 $\mathsf{ExtKerMDH}$ 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 1. *Let ℓ, k, and $\mathcal{D}_\mathsf{k}$ be as in Definition 1. If $\mathcal{D}_\mathsf{k}$-$(\ell-1)$-$\mathsf{ExtKerMDH}$ 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 $\mathcal{D}_\mathsf{k}$-$(\ell-1)$-$\mathsf{ExtKerMDH}$:

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

If $\mathcal{A}$ succeeds, then by Definition 1, $(\boldsymbol{\gamma}\|\boldsymbol{C})\binom{\boldsymbol{D}}{\boldsymbol{\delta}} = \mathbf{0}$ 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 10.2, we reduce CED to a gap assumption. It seems that $\mathsf{ExtKerMDH}$ 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 $\mathtt{crs}$ 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 "$\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$; $(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{kgen}(\mathsf{p})$" by $(\mathsf{p}, \mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{kgen}(1^\lambda)$.
(3) $\mathsf{P}(\mathtt{crs}, \mathtt{lpar}, \mathtt{x}, \mathtt{w})$: 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) $\mathsf{V}(\mathtt{crs}, \mathtt{lpar}, \mathtt{x}, \pi)$. 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) $\mathsf{Sim}(\mathtt{crs}, \mathtt{td}, \mathtt{lpar}, \mathtt{x})$. 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

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

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

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

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

Π for $\{\mathcal{D}_{\mathsf{p}}\}_{\mathsf{p}}$ is *perfectly zero-knowledge*, if for all λ, all $(\mathsf{p}, \mathtt{crs}, \mathtt{td}) \in \mathrm{Supp}(\mathsf{K}_{\mathtt{crs}}(1^\lambda))$, all $\mathtt{lpar} \in \mathrm{Supp}(\mathcal{D}_{\mathsf{p}})$ and all $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}_{\mathtt{lpar}}$, the distributions $\mathsf{P}(\mathtt{crs}, \mathtt{lpar}, \mathtt{x}, \mathtt{w})$ and $\mathsf{Sim}(\mathtt{crs}, \mathtt{td}, \mathtt{lpar}, \mathtt{x})$ 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 9.

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 3.1 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 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 Eq. (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 4, we describe a general technique that results both in efficient NIZK arguments for languages

$$\mathcal{L}_{\mathsf{pk},F} = \{[\mathbf{ct}]_1 : \exists \boldsymbol{\chi} \text{ such that } \mathsf{Dec}([\mathbf{ct}]_1) = [\boldsymbol{\chi}]_1 \wedge \boldsymbol{\chi} \in \mathcal{Z}(F)\} \ . \tag{1}$$

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 1 that an adversary breaks the $\mathcal{L}_1$-$(\ell-1)$-CED assumption if, given $[\boldsymbol{D}]_2 = [\begin{smallmatrix} 1 \\ e \end{smallmatrix}]_2 \leftarrow_{\$} \mathcal{L}_1$ (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

$$\boldsymbol{\gamma} + \boldsymbol{C}\left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right) = \boldsymbol{0}. \tag{2}$$

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 Eq. (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., Eq. (2)), then one can break the CED assumption by returning $[(\boldsymbol{\gamma} \| \boldsymbol{C})(\boldsymbol{\chi})]_1$ and $[\boldsymbol{\delta}]_2$.

3.1 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 Appendix B.1 for a discussion.

Definition 2 (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 $\boldsymbol{\chi} \in \mathcal{Z}(F)$, $\boldsymbol{h}(\boldsymbol{\chi}) \in \operatorname{colspace}(\boldsymbol{T}(\boldsymbol{\chi}))$.*

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 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 $\boldsymbol{\chi} \in \mathbb{Z}_p^{\nu}$, $\operatorname{rk}(\boldsymbol{T}(\boldsymbol{\chi})) = \ell - 1$.
(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 $\operatorname{rk}(\boldsymbol{C}(\boldsymbol{\chi})) \leq \ell - 1$. Suppose $\boldsymbol{h}(\boldsymbol{\chi}) \notin \operatorname{colspace}(\boldsymbol{T}(\boldsymbol{\chi}))$. Then $\operatorname{rk}(\boldsymbol{C}(\boldsymbol{\chi})) > \operatorname{rk}(\boldsymbol{T}(\boldsymbol{\chi}))$. By the high right rank property, $\ell - 1 \geq \operatorname{rk}(\boldsymbol{C}(\boldsymbol{\chi})) > \operatorname{rk}(\boldsymbol{T}(\boldsymbol{\chi})) = \ell - 1$, which is a contradiction. Thus, $\boldsymbol{h}(\boldsymbol{\chi}) \in \operatorname{colspace}(\boldsymbol{T}(\boldsymbol{\chi}))$.

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

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 and 6 for more.

3.2 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 Fig. 2) for $\boldsymbol{\mathcal{L}}_{\mathsf{pk},F}$. 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 Eq. (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.

```
$\mathsf{comp}_1(\mathsf{p}, \boldsymbol{\chi}, \boldsymbol{C}(\boldsymbol{X}))$:
    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 $([\boldsymbol{\gamma}]_1, st)$;

$\mathsf{comp}_2(st, \psi(e))$:
    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 $\psi(\boldsymbol{\delta})$;
```

Fig. 1. comp_i algorithms assuming $\boldsymbol{h}(\boldsymbol{\chi}) \in \text{colspace}(\boldsymbol{T}(\boldsymbol{\chi}))$. Here, $\psi = id$ in the case of the Σ-protocol, and $\psi = [\cdot]_2$ in the case of the NIZK argument.

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 1. 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 3. *Assume F is as in Definition 2 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.

Proof. Recalling $\boldsymbol{C}(\boldsymbol{X}) = (\boldsymbol{h} \| \boldsymbol{T})(\boldsymbol{X})$, we rewrite Eq. (2) as

$$\boldsymbol{\gamma} + \boldsymbol{h}(\boldsymbol{X})e + \boldsymbol{T}(\boldsymbol{X})\boldsymbol{\delta} = \boldsymbol{0} \ . \tag{3}$$

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 $\boldsymbol{\chi}$, Eq. (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 Eq. (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 Fig. 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 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 Eq. (2) and thus Eq. (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 $\boldsymbol{h}(\boldsymbol{\chi}) \in \text{colspace}(\boldsymbol{T}(\boldsymbol{\chi}))$. □

```
$\mathsf{kgen}(\mathsf{p}, \mathtt{lpar})$: $e \leftarrow_{\$} \mathbb{Z}_p$; return $(\mathtt{crs}, \mathtt{td}) \leftarrow ([e]_2, e)$ ;
$\mathsf{P}(\mathtt{crs}, \mathtt{lpar}, \mathtt{x} = [\mathbf{ct}]_1, \mathtt{w} = (\boldsymbol{\chi}, \boldsymbol{r}))$: $([\boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2) \leftarrow \mathsf{comp}(\mathsf{p}, [e]_2, \boldsymbol{\chi}, \boldsymbol{C}(\boldsymbol{X}))$;
    $\boldsymbol{\varrho} \leftarrow_{\$} \mathbb{Z}_p^{\ell}$; $[\mathbf{ct}^{\gamma}]_1 \leftarrow \mathsf{Enc}([\boldsymbol{\gamma}]_1; \boldsymbol{\varrho}) \in \mathbb{G}_1^{\ell \times 2}$;
    $[\boldsymbol{z}]_2 \leftarrow \boldsymbol{\varrho}[1]_2 + (\sum_{k=1}^{\nu} r_k \boldsymbol{P}_k) \left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 \in \mathbb{G}_2^{\ell}$.
    Return $\pi \leftarrow ([\mathbf{ct}^{\gamma}]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2) \in \mathbb{G}_1^{\ell \times 2} \times \mathbb{G}_2^{2\ell - 1}$.
$\mathsf{V}(\mathtt{crs}, \mathtt{lpar}, \mathtt{x} = [\mathbf{ct}]_1, \pi)$: check $[\boldsymbol{I}_{\ell}]_2 \bullet [\mathbf{ct}^{\gamma}]_1 + \sum_{k=1}^{\nu} (\boldsymbol{P}_k \left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 \bullet [\mathbf{ct}_k]_1) =^? (-\boldsymbol{Q} \left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2) \bullet [0 \| 1]_1 + [\boldsymbol{z}]_2 \bullet$
    $\mathsf{pk}$.
$\mathsf{Sim}(\mathtt{crs}, \mathtt{td}, \mathtt{lpar}, \mathtt{x} = [\mathbf{ct}]_1)$: $\boldsymbol{\delta} \leftarrow_{\$} \mathbb{Z}_p^{\ell - 1}$;
    $\boldsymbol{z} \leftarrow_{\$} \mathbb{Z}_p^{\ell}$; $[\mathbf{ct}^{\gamma}]_1 \leftarrow \mathsf{Enc}(-\boldsymbol{Q}\left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right)[1]_1; \boldsymbol{z}) - \sum_{k=1}^{\nu} \boldsymbol{P}_k \left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right)[\mathbf{ct}_k]_1$;
    Return $\pi \leftarrow ([\mathbf{ct}^{\gamma}]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2) \in \mathbb{G}_1^{\ell \times 2} \times \mathbb{G}_2^{2\ell - 1}$.
```

Fig. 2. The new NIZK argument $\mathbf{\Pi}_{\mathsf{nizk}}$ for $\mathcal{L}_{\mathsf{pk},F}$.

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 and 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

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

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 $[\boldsymbol{C}]_1$.

4.1 Protocol Description

Let $\mathcal{L}_{\mathsf{pk},F}$ be defined as in Eq. (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 Fig. 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 $\mathtt{crs} = [e]_2$) to compute $[\boldsymbol{\delta}]_2$. If P is honest, then by the definition of QDRs of F, Eq. (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 Fig. 2.

4.2 Efficiency

Next, we estimate the efficiency of the NIZK argument. Note that if we use the comp algorithm given in Fig. 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. *Consider $\mathbf{\Pi}_{\mathsf{nizk}}$ 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*

(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 3.1, 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, 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 6, we study other interesting cases.

The estimate in Lemma 5 is often over-conservative. For example, let $\boldsymbol{\delta}' = \left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right)$. 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.

4.3 Security of the NIZK Argument

Theorem 1. *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 Fig. 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, $(\mathsf{p}, \mathtt{td}) \in \operatorname{Supp}(\mathsf{K}_\mathtt{crs}(1^\lambda))$ and compute $\mathtt{crs} = [\mathtt{td}]_2$. Then fix $\mathtt{lpar} \in \operatorname{Supp}(\mathcal{D}_\mathsf{p})$ 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 3.2), $\boldsymbol{\delta}$ output by an honest prover is uniformly random. On the other hand, Sim (see Fig. 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 $\mathtt{crs}$ and any $\mathtt{lpar} \in \operatorname{Supp}(\mathcal{D}_\mathsf{p})$ (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 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 $(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{K}_\mathtt{crs}(\mathsf{p})$. $\mathcal{B}$ 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 $\mathtt{lpar} = (\mathsf{pk}, F) \in \operatorname{Supp}(\mathcal{D}_\mathsf{p})$, and sends $\mathtt{crs} = [e]_2$ 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 $[\gamma]_1$. $\mathcal{B}$ 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 Fig. 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, $\boldsymbol{\gamma} + \boldsymbol{C}(\boldsymbol{\chi})\binom{e}{\boldsymbol{\delta}} = \mathbf{0}$, i.e., Eq. (2) holds. Since $\det(\boldsymbol{C}(\boldsymbol{\chi})) = F(\boldsymbol{\chi}) \neq 0$, $\boldsymbol{C}$ has full rank. Thus, $\mathcal{B}$ breaks CED. □

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 7.1. Thus, following sections consider alternative construction techniques of such matrices.

5.1 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 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.

5.2 NIZK for Algebraic Branching Programs

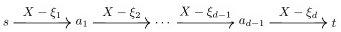
Lemma 6. *Let* $\mathsf{abp} = (V, E, s, t, \phi)$ *be an ABP that computes a* ν*-variate polynomial* $F(\boldsymbol{X})$*. Then* $\mathsf{IK}(\boldsymbol{X})$ *is a QDR of* F *with* $\ell = |V| - 1$*.*

Proof. Items 1 and 2 of Proposition 1 state directly that the affine map and reducibility properties of Definition 2 hold. From 3 of Proposition 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 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 Fig. 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 $i \in [1, \ell]$.



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

Fig. 3. The ABP $\mathsf{abp}^d_{\mathsf{path}}(X, \boldsymbol{\xi})$ for $F(X) = \prod_{i=1}^d (X - \xi_i)$ and $\mathsf{IK}_{path}(X)$

(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 7. *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 $\mathbb{G}_2$. □

6 Applications

6.1 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 $F_1, \dots, F_\tau \in \mathbb{Z}_p[X]$. 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 [CLO15, Section 1.5], and thus any ideal can be written as $\mathcal{I} = \langle F \rangle$ for some F. Thus, $\mathcal{A} = \mathcal{A}(F)$ for $F \leftarrow \gcd(F_1, \dots, F_\tau)$ [CLO15, Section 1.5].

Moreover, $\mathcal{I}(\mathcal{A}(F)) = \mathcal{I}(F_{red})$ [CLO15, 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 $F_{red} = F/\gcd(F, F')$, [CLO15, 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 Fig. 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 Fig. 3.

Fig. 4 depicts the resulting set-membership NIZK argument that $X \in \{\xi_i\}$.

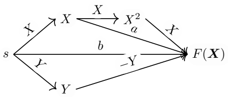
Lemma 8. *Let $F(X)$ be a univariate reduced polynomial. The ABP-based NIZK argument for $\mathcal{L}_{\mathsf{pk},F}$ has prover's computation of at most $3d$ exponentiations in $\mathbb{G}_1$ and $4d - 2$ exponentiations in $\mathbb{G}_2$, verifier's computation of $7d - 1$ pairings and at most d exponentiations 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 Fig. 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 $\chi = \xi_d$,

```
$\mathsf{kgen}(\mathsf{p}, \mathtt{lpar})$: $e \leftarrow_\$ \mathbb{Z}_p$; return $(\mathtt{crs}, \mathtt{td}) \leftarrow ([e]_2, e)$ ;
$\mathsf{P}(\mathtt{crs}, \mathtt{lpar}, \mathtt{x} = [\mathbf{ct}]_1, \mathtt{w} = (\chi, r))$: $([\boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2) \leftarrow \mathsf{comp}(\mathsf{p}, [e]_2, \chi, \boldsymbol{C}(\boldsymbol{X}))$;
    $\boldsymbol{\varrho} \leftarrow_\$ \mathbb{Z}_p^d$; $[\mathbf{ct}^{\boldsymbol{\gamma}}]_1 \leftarrow \mathsf{Enc}([\boldsymbol{\gamma}]_1; \boldsymbol{\varrho}) \in \mathbb{G}_1^{d \times 2}$; $[\boldsymbol{z}]_2 \leftarrow \boldsymbol{\varrho}[1]_2 + r \left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 \in \mathbb{G}_2^d$;
    return $\pi \leftarrow ([\mathbf{ct}^{\boldsymbol{\gamma}}]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2)$.
$\mathsf{V}(\mathtt{crs}, \mathtt{lpar}, \mathtt{x} = [\mathbf{ct}]_1, \pi)$: check $[\boldsymbol{I}_d]_2 \bullet [\mathbf{ct}^{\boldsymbol{\gamma}}]_1 + \left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 \bullet [\mathbf{ct}]_1 + \boldsymbol{Q} \left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 \bullet [0\|1]_1 =^? [\boldsymbol{z}]_2 \bullet \mathsf{pk}$.
$\mathsf{Sim}(\mathtt{crs}, \mathtt{td}, \mathtt{lpar}, \mathtt{x} = [\mathbf{ct}]_1)$: $\boldsymbol{\delta} \leftarrow_\$ \mathbb{Z}_p^{d-1}$; $\boldsymbol{z} \leftarrow_\$ \mathbb{Z}_p^d$; $[\mathbf{ct}^{\boldsymbol{\gamma}}]_1 \leftarrow \mathsf{Enc}(-\boldsymbol{Q}\left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right)[1]_1; \boldsymbol{z}) - \left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right) \cdot [\mathbf{ct}]_1$; return
    $\pi \leftarrow ([\mathbf{ct}^{\boldsymbol{\gamma}}]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2)$.
```

Fig. 4. The NIZK argument for $\mathcal{L}_{\mathsf{pk},F}$, where $F(X)$ is a monic univariate polynomial with $\mathsf{qdc}(F) = d$.



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

Fig. 5. ABP example for $F(X, Y) = X^3 + aX + b - Y^2$.

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 $[\mathbf{ct}^{\boldsymbol{\gamma}}]_1$ 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 $[\kappa_i]_T = [\left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right)_i]_2 \bullet ([\mathbf{ct}]_1 - \xi_i \circ [0\|1]_1)$. Here, $\left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right)_i$ is the ith coefficient of the vector $\left(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right)$. 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 $[\mathbf{ct}^{\boldsymbol{\gamma}}]_1$, $d - 1$ group elements to transfer $[\boldsymbol{\delta}]_2$, and d group elements to transfer the randomizers $[\boldsymbol{z}]_2$, $4d - 1$ group elements in total. □

6.2 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 Appendix C.1, we will detail three example NIZK arguments that are all based on CED-matrices. The 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.

6.3 Elliptic Curve Points

In Fig. 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 7.1, 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 [BCTV14], 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:

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

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 Appendix D.3, we will provide an example for $d = 5$. We refer to [PSV12] for algorithms and general discussion.

7.1 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 6.3, 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 Appendix D.1, 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 Appendix D.2.

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.

8.1 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 [GGPR13]) 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 $\mathcal{A}_{\mathfrak{C}} = (\chi_1, \dots, \chi_n) \in \mathbb{Z}_p^n$, 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 [GGPR13], 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, $j \le \mathsf{m}_0$ 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 $\mathcal{A}_{\mathfrak{C}} = \{(\chi_1, \dots, \chi_m) : F_i(\boldsymbol{\chi}) = 0 \text{ for all } i > \mathsf{m}_0\}$. 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 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* [GGPR13]), 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 $U = (u_{ij})$, $V = (v_{ij})$, 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 $q \ge 1$, a q-aCSP instance F over $\mathbb{F}$ is a collection of functions $F_1, \dots, F_\tau$ (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 $i_1, \dots, i_q \in [1, n]$ and $f : \mathbb{F}^q \to \mathbb{F}$ such that $F_j(\boldsymbol{\chi}) = f(\chi_{i_1}, \dots, \chi_{i_q})$ 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 8.2 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

Table 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 8.1. Here, $|\mathbb{G}_\iota|$ is the length of one element from $\mathbb{G}_\iota$

Protocol	$\lvert\mathsf{crs}\rvert$	$\lvert\mathsf{com}\rvert$	$\lvert\pi\rvert$	P comp.	V comp.
Groth-Sahai [GSW09]	$4(\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert)$	$2(m+1)(\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert)$	$(6m+2n+2)(\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert)$	$(12m+4n+4)(\mathfrak{e}_1+\mathfrak{e}_2)$	$16(2m+n)\mathfrak{p}$
New, Section 8.1	$\lvert\mathbb{G}_2\rvert$	$2m \cdot \lvert\mathbb{G}_1\rvert$	$(m+n)(4\lvert\mathbb{G}_1\rvert + 3\lvert\mathbb{G}_2\rvert)$	$(m+n)(5\mathfrak{e}_1+4\mathfrak{e}_2)$	$13(m+n)\mathfrak{p}$

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 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.

8.2 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 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`[10]) 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[11] is

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

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

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

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

[11] In the case of many other sets, the NIZK will be much more efficient. We will provide one concrete example in Appendix E.1.

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

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

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 Appendix C.2, 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 Appendix C.2, we note that the use of the NIZK from Section 6.1 helps us to obtain a NIZK with better verifier's computation.

9 Back to Algebraic Languages

The well-known methodology of diverse vector spaces (DVSs, [BBC$^+$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 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 8 results in an efficient NIZK. We start the process from a QDR $\boldsymbol{C}_i$ of F_i, where $\langle F_1, \dots, F_\tau \rangle$ 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 [BBC$^+$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 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 Appendix F.1. 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.

9.1 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 $\mathcal{I}(\mathcal{A}) = \langle F_1, \dots, F_\tau \rangle$ 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 Fig. 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 $[\mathbf{ct}_i]_1^\top$) and constants are encrypted by using the randomizer 0. We define $[\boldsymbol{\Gamma}(\mathtt{x})]_1$ and $[\boldsymbol{\theta}(\mathtt{x})]_1$ as follows:

$$[\boldsymbol{\Gamma}(\mathtt{x})]_1 = (\mathcal{E}(\boldsymbol{T}(\boldsymbol{\chi})) \| \mathcal{E}(\boldsymbol{0}_{d \times d}; \boldsymbol{I}_d)) \in \mathbb{G}_2^{2d \times (2d-1)} \quad , \quad [\boldsymbol{\theta}(\mathtt{x})]_1 = \mathcal{E}(\boldsymbol{h}(\boldsymbol{\chi})) \in \mathbb{G}_2^{2d} \quad . \tag{4}$$

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 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 $\mathsf{Enc}([0]_1; 0) = [0 \| 0]_1$, Eq. (4) results in

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

A variation of this $[\boldsymbol{\Gamma}, \boldsymbol{\theta}]_1$ was given in [BBC+13,CH20]. To motivate Theorem 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 $\boldsymbol{0}_4$ iff $\chi \in \{0, 1\}$.

Theorem 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 2, there exists $\boldsymbol{w}$ such that $\boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{w} = \boldsymbol{h}(\boldsymbol{\chi})$, i.e., $\boldsymbol{C}(\boldsymbol{\chi})\left(\begin{smallmatrix} 1 \\ -\boldsymbol{w} \end{smallmatrix} \right) = \boldsymbol{0}$. 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 $i \leq d-1$. 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 $(\mathcal{E}(\boldsymbol{T}) \| \mathcal{E}(\boldsymbol{0}_{d \times d}; \boldsymbol{I}_d))\boldsymbol{w}^* = \mathcal{E}(\boldsymbol{h})$ match. Denoting $\hat{\boldsymbol{w}} = (w_d^*, \ldots, w_{2d-1}^*)^\top$, this is achieved by setting $\hat{\boldsymbol{w}} \leftarrow (\sum \boldsymbol{P}_k r_k)\left(\begin{smallmatrix} 1 \\ -\boldsymbol{w} \end{smallmatrix} \right)$. Really, then

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

(2) Assume that $\mathtt{x} = \mathsf{Enc}(\boldsymbol{\chi}) \in \mathcal{L}_{\boldsymbol{\Gamma},\boldsymbol{\theta}}$, 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 $\boldsymbol{C}(\boldsymbol{\chi})\left(\begin{smallmatrix} -1 \\ \boldsymbol{w} \end{smallmatrix} \right) = \boldsymbol{0}$. If $\mathtt{x} \notin \mathcal{L}_{\mathsf{pk},F}$ then $\det(\boldsymbol{C}(\boldsymbol{\chi})) \neq 0$. Since -1 is non-zero, this is a contradiction. □

In Appendix F.2, 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

$$\boldsymbol{\Gamma} = \left(\begin{smallmatrix} \boldsymbol{\Gamma}_1 & \ldots & 0 \\ \ldots & \ldots & \ldots \\ 0 & \ldots & \boldsymbol{\Gamma}_\tau \end{smallmatrix} \right) \text{ and } \boldsymbol{\theta} = \left(\begin{smallmatrix} \boldsymbol{\theta}_1 \\ \ldots \\ \boldsymbol{\theta}_\tau \end{smallmatrix} \right) \quad .$$

On the Couteau-Hartmann Disjunction. In Appendix F.3, we describe the Couteau-Hartmann disjunction that results in $\boldsymbol{\Gamma}$ of size $(3d-1) \times (3d-2)$ and compare it to Eq. (4). For the sake of completeness, we also reprove the efficiency of the CHM NIZK from [CH20].

9.2 Efficiency of Set-Membership NIZKs: Comparisons

In Table 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 8 is quite a bit more efficient. However, it compares favorably to [CH20]. In the following lemma, we state its efficiency.

Lemma 9. *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 6.1. Let $[\boldsymbol{\Gamma}]_1$ be constructed as in Eq. (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. 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

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

For a general $\boldsymbol{C}$, the efficiency estimate follows from Proposition 2 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.

9.3 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,BBC$^+$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 [BBC$^+$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 $\mathcal{A}(\mathcal{I})$ for which the NIZK of Section 4 is efficient, one can also construct an efficient SPHF by constructing $\boldsymbol{\Gamma}$ and $\boldsymbol{\theta}$ as in Eq. (4).

Example 2 (GL-SPHF for the language of elliptic curve points.). Let $\mathcal{A} = \{(X,Y) : Y^2 = X^3 + aX + b\}$ as in Section 6.3. Then, one can use $\mathtt{lpar} = (\boldsymbol{\Gamma}, \boldsymbol{\theta})$ from Example 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 $\mathtt{x} \in \mathcal{L}_{\mathtt{lpar}}$, $\mathsf{pH} = \mathsf{H} = [\boldsymbol{\alpha}^\top\boldsymbol{\Gamma}\mathtt{w}]_1 =$

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

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,GGPR13] (albeit the prover cost will be much higher in general).

Next, we discuss the falsifiability of the CED assumption. In Section 10.1, 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 10.2, 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,GGPR13]). 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 10.3, 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.

10.1 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 $\mathtt{x} = [\mathbf{ct}_1, \dots, \mathbf{ct}_\nu]_1$ and an sk-dependent trapdoor $\mathbf{T}$, $\mathsf{V_f}(\mathsf{p}, \mathsf{pk}, \mathtt{x}, \mathbf{T})$ can efficiently check whether $\mathsf{Dec}_{\mathsf{sk}}([\mathbf{ct}_1, \dots, \mathbf{ct}_\nu]_1) \in \mathcal{L}_{\mathsf{pk},F}$. 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, $\mathtt{x} \notin \mathcal{L}_{\mathsf{pk},F}$), 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 6.1 and the NIZK for boolean circuits in Section 8.1 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 8, is based on falsifiable CED iff $\mathcal{A}(\mathfrak{I})$ has polynomial size.

The outliers are the cases of principal ideals of multivariate polynomials (since then $|\mathcal{A}(\mathfrak{I})|$ can be exponential as in the set of points (X, Y) on an elliptic curve) and some instances of non-principal ideals where $|\mathcal{A}(\mathfrak{I})|$ is super-polynomial. In the latter case, we can clarify the situation further. Namely, given a generating set $\langle F_1, \dots, F_\tau \rangle$, by Bézout's theorem, $\mathcal{A}(\mathfrak{I})$ 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 8.2 or aCSPs for constant-size arithmetic circuits in Section 8.1 are based on falsifiable CED. However, range proofs and superconstant-size arithmetic circuits are based on non-falsifiable CED.

The super-polynomial size of $\mathcal{A}(\mathfrak{I})$ 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 $\mathsf{V_f}$.

10.2 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 3. *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 $\mathcal{A}$,*

$$\Pr\left[\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \boldsymbol{D} \leftarrow_{\$} \mathcal{D}_{\ell-1,\mathsf{k}}; [\boldsymbol{c}]_1 \leftarrow \mathcal{A}^{\mathcal{O}}(\mathsf{p}, [\boldsymbol{D}]_2) : \boldsymbol{D}^\top \boldsymbol{c} = \boldsymbol{0}_{\mathsf{k}} \wedge \boldsymbol{c} \neq \boldsymbol{0}_{\ell-1}\right] \approx_\lambda 0 \enspace .$$

Theorem 3. *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.*

Proof (of Theorem 3). Let $\mathcal{A}$ be an CED adversary, as in Definition 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

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

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

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

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

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

Since[12] $\mathrm{rk}(\boldsymbol{C}) \geq \ell$, 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 $\mathcal{O}$ 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).

10.3 DLIN-Based NIZK Based on Falsifiable CED

While constructing a Sub-ZK QA-NIZK, [ALSZ20] 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 [ALSZ20], consider the case $[\boldsymbol{C}]_1 \in \mathbb{G}_1^{2\times2}$. [ALSZ20] 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 [ALSZ20] was that

[12] 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 $\mathrm{rk}(\boldsymbol{\gamma}\|\boldsymbol{C}) \geq \ell$.

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

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

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 4.1 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

ABP15. Michel Abdalla, Fabrice Benhamouda, and David Pointcheval. Disjunctions for hash proof systems: New constructions and applications. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 69–100. Springer, Heidelberg, April 2015. doi:10.1007/978-3-662-46803-6_3.

ALSZ20. Behzad Abdolmaleki, Helger Lipmaa, Janno Siim, and Michal Zajac. On QA-NIZK in the BPK model. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 590–620. Springer, Heidelberg, May 2020. doi:10.1007/978-3-030-45374-9_20.

BBC$^+$13. Fabrice Benhamouda, Olivier Blazy, Céline Chevalier, David Pointcheval, and Damien Vergnaud. New techniques for SPHFs and efficient one-round PAKE protocols. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part I*, volume 8042 of *LNCS*, pages 449–475. Springer, Heidelberg, August 2013. doi:10.1007/978-3-642-40041-4_25.

BBS04. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, Heidelberg, August 2004. doi:10.1007/978-3-540-28628-8_3.

BCTV14. Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, and Madars Virza. Scalable zero knowledge via cycles of elliptic curves. In Juan A. Garay and Rosario Gennaro, editors, *CRYPTO 2014, Part II*, volume 8617 of *LNCS*, pages 276–294. Springer, Heidelberg, August 2014. doi:10.1007/978-3-662-44381-1_16.

Bea00. Arnaud Beauville. Determinantal Hypersurfaces. *Michigan Math. J.*, 48(1):39–64, 2000.

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

BFI$^+$10. Olivier Blazy, Georg Fuchsbauer, Malika Izabachène, Amandine Jambert, Hervé Sibert, and Damien Vergnaud. Batch Groth-Sahai. In Jianying Zhou and Moti Yung, editors, *ACNS 10*, volume 6123 of *LNCS*, pages 218–235. Springer, Heidelberg, June 2010. doi:10.1007/978-3-642-13708-2_14.

BFM88. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112. ACM Press, May 1988. doi:10.1145/62212.62222.

BG99. Amos Beimel and Anna Gál. On Arithmetic Branching Programs. *J. Comput. Syst. Sci.*, 59(2):195–220, 1999.

Bou00. Fabrice Boudot. Efficient proofs that a committed number lies in an interval. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 431–444. Springer, Heidelberg, May 2000. 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 Jesper Buus Nielsen and Vincent Rijmen, editors, *EUROCRYPT 2018, Part III*, volume 10822 of *LNCS*, pages 193–221. Springer, Heidelberg, April / May 2018. doi:10.1007/978-3-319-78372-7_7.

CCH$^+$19. Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-Shamir: from practice to theory. In Moses Charikar and Edith Cohen, editors, *51st ACM STOC*, pages 1082–1090. ACM Press, June 2019. doi:10.1145/3313276.3316380.

CCs08. Jan Camenisch, Rafik Chaabouni, and abhi shelat. Efficient protocols for set membership and range proofs. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 234–252. Springer, Heidelberg, December 2008. 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 Yvo Desmedt, editor, *CRYPTO'94*, volume 839 of *LNCS*, pages 174–187. Springer, Heidelberg, August 1994. doi:10.1007/3-540-48658-5_19.

CG15. Pyrros Chaidos and Jens Groth. Making sigma-protocols non-interactive without random oracles. In Jonathan Katz, editor, *PKC 2015*, volume 9020 of *LNCS*, pages 650–670. Springer, Heidelberg, March / April 2015. 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 Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 768–798. Springer, Heidelberg, August 2020. doi:10.1007/978-3-030-56877-1_27.

CLO15. David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra.* Undergraduate Texts in Mathematics. Springer, 4 edition, May 13, 2015.

CLs10. Rafik Chaabouni, Helger Lipmaa, and abhi shelat. Additive combinatorics and discrete logarithm based range protocols. In Ron Steinfeld and Philip Hawkes, editors, *ACISP 10*, volume 6168 of *LNCS*, pages 336–351. Springer, Heidelberg, July 2010.

CLZ12. Rafik Chaabouni, Helger Lipmaa, and Bingsheng Zhang. A non-interactive range proof with constant communication. In Angelos D. Keromytis, editor, *FC 2012*, volume 7397 of *LNCS*, pages 179–199. Springer, Heidelberg, February / March 2012.

CS02. Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 45–64. Springer, Heidelberg, April / May 2002. doi:10.1007/3-540-46035-7_4.

DGP$^+$19. Vanesa Daza, Alonso González, Zaira Pindado, Carla Ràfols, and Javier Silva. Shorter quadratic QA-NIZK proofs. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part I*, volume 11442 of *LNCS*, pages 314–343. Springer, Heidelberg, April 2019. doi:10.1007/978-3-030-17253-4_11.

Dic21. Leonard Eugene Dickson. Determination of All General Homogeneous Polynomials Expressible as Determinants with Linear Elements. *Trans. of the American Mathematical Society*, 22(2):167–179, April 1921.

Dol10. Igor V. Dolgachev. Topics in Classical Algebraic Geometry. September 7, 2010. URL: https://www.math.ucsd.edu/~eizadi/207A-14/Dolgachev-topics.pdf.

EG14. Alex Escala and Jens Groth. Fine-tuning Groth-Sahai proofs. In Hugo Krawczyk, editor, *PKC 2014*, volume 8383 of *LNCS*, pages 630–649. Springer, Heidelberg, March 2014. doi:10.1007/978-3-642-54631-0_36.

EHK$^+$13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013, Part II*, volume 8043 of *LNCS*, pages 129–147. Springer, Heidelberg, August 2013. doi:10.1007/978-3-642-40084-1_8.

ElG84. Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *CRYPTO'84*, volume 196 of *LNCS*, pages 10–18. Springer, Heidelberg, August 1984.

FS87. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *CRYPTO'86*, volume 263 of *LNCS*, pages 186–194. Springer, Heidelberg, August 1987. doi:10.1007/3-540-47721-7_12.

GGPR13. Rosario Gennaro, Craig Gentry, Bryan Parno, and Mariana Raykova. Quadratic span programs and succinct NIZKs without PCPs. In Thomas Johansson and Phong Q. Nguyen, editors, *EUROCRYPT 2013*, volume 7881 of *LNCS*, pages 626–645. Springer, Heidelberg, May 2013. 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 Tetsu Iwata and Jung Hee Cheon, editors, *ASIACRYPT 2015, Part I*, volume 9452 of *LNCS*, pages 605–629. Springer, Heidelberg, November / December 2015. doi:10.1007/978-3-662-48797-6_25.

GL03. Rosario Gennaro and Yehuda Lindell. A framework for password-based authenticated key exchange. In Eli Biham, editor, *EUROCRYPT 2003*, volume 2656 of *LNCS*, pages 524–543. Springer, Heidelberg, May 2003. https://eprint.iacr.org/2003/032.ps.gz. doi:10.1007/3-540-39200-9_33.

GMR89. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

GOS06. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 97–111. Springer, Heidelberg, August 2006. doi:10.1007/11818175_6.

GPS08. Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for Cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

Gro10. Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340. Springer, Heidelberg, December 2010. doi:10.1007/978-3-642-17373-8_19.

GS08. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, April 2008. 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 Matthew G. Parker, editor, *12th IMA International Conference on Cryptography and Coding*, volume 5921 of *LNCS*, pages 469–494. Springer, Heidelberg, December 2009.

Har92. Joe Harris. *Algebraic Geometry: A First Course*, volume 133 of *Graduate Texts in Mathematics*. Springer-Verlag, 1992.

HKR19. Max Hoffmann, Michael Klooß, and Andy Rupp. Efficient zero-knowledge arguments in the discrete log setting, revisited. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *ACM CCS 2019*, pages 2093–2110. ACM Press, November 2019. 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*, pages 294–304. IEEE Computer Society Press, November 2000. doi:10.1109/SFCS.2000.892118.

IK02. Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan Eidenbenz, and Ricardo Conejo, editors, *ICALP 2002*, volume 2380 of *LNCS*, pages 244–256. Springer, Heidelberg, July 2002. doi:10.1007/3-540-45465-9_22.

IW14. Yuval Ishai and Hoeteck Wee. Partial garbling schemes and their applications. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *ICALP 2014, Part I*, volume 8572 of *LNCS*, pages 650–662. Springer, Heidelberg, July 2014. 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 Kazue Sako and Palash Sarkar, editors, *ASIACRYPT 2013, Part I*, volume 8269 of *LNCS*, pages 1–20. Springer, Heidelberg, December 2013. doi:10.1007/978-3-642-42033-7_1.

Kiy20. Susumu Kiyoshima. Round-optimal black-box commit-and-prove with succinct communication. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part II*, volume 12171 of *LNCS*, pages 533–561. Springer, Heidelberg, August 2020. 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 Amos Beimel and Stefan Dziembowski, editors, *TCC 2018, Part I*, volume 11239 of *LNCS*, pages 286–313. Springer, Heidelberg, November 2018. doi:10.1007/978-3-030-03807-6_11.

KW15. Eike Kiltz and Hoeteck Wee. Quasi-adaptive NIZK for linear subspaces revisited. In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 101–128. Springer, Heidelberg, April 2015. doi:10.1007/978-3-662-46803-6_4.

KZM+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. Technical Report 2015/1093, IACR, November 10, 2015. `https://ia.cr/2015/1093`, last accessed version 9 Apr 2017.

LAN03. Helger Lipmaa, N. Asokan, and Valtteri Niemi. Secure Vickrey auctions without threshold trust. In Matt Blaze, editor, *FC 2002*, volume 2357 of *LNCS*, pages 87–101. Springer, Heidelberg, March 2003.

Lip03. Helger Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In Chi-Sung Laih, editor, *ASIACRYPT 2003*, volume 2894 of *LNCS*, pages 398–415. Springer, Heidelberg, November / December 2003. `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 Ronald Cramer, editor, *TCC 2012*, volume 7194 of *LNCS*, pages 169–189. Springer, Heidelberg, March 2012. `doi:10.1007/978-3-642-28914-9_10`.

Lip16. Helger Lipmaa. Prover-efficient commit-and-prove zero-knowledge SNARKs. In David Pointcheval, Abderrahmane Nitaj, and Tajjeeddine Rachidi, editors, *AFRICACRYPT 16*, volume 9646 of *LNCS*, pages 185–206. Springer, Heidelberg, April 2016. `doi:10.1007/978-3-319-31517-1_10`.

Mau09. Ueli M. Maurer. Unifying zero-knowledge proofs of knowledge. In Bart Preneel, editor, *AFRICACRYPT 09*, volume 5580 of *LNCS*, pages 272–286. Springer, Heidelberg, June 2009.

MB82. H. Michael Möller and Bruno Buchberger. The Construction of Multivariate Polynomials with Preassigned Zeros. In Jacques Calmet, editor, *EUROCAM 1982*, volume 144 of *LNCS*, pages 24–31, Marseille, France, 5-7 April 1982. Springer.

MRV16. Paz Morillo, Carla Ràfols, and Jorge Luis Villar. The kernel matrix Diffie-Hellman assumption. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part I*, volume 10031 of *LNCS*, pages 729–758. Springer, Heidelberg, December 2016. `doi:10.1007/978-3-662-53887-6_27`.

Nao03. Moni Naor. On cryptographic assumptions and challenges (invited talk). In Dan Boneh, editor, *CRYPTO 2003*, volume 2729 of *LNCS*, pages 96–109. Springer, Heidelberg, August 2003. `doi:10.1007/978-3-540-45146-4_6`.

Nis91. Noam Nisan. Lower bounds for non-commutative computation (extended abstract). In *23rd ACM STOC*, pages 410–418. ACM Press, May 1991. `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 Kwangjo Kim, editor, *PKC 2001*, volume 1992 of *LNCS*, pages 104–118. Springer, Heidelberg, February 2001. `doi:10.1007/3-540-44586-2_8`.

Pas13. Rafael Pass. Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In Amit Sahai, editor, *TCC 2013*, volume 7785 of *LNCS*, pages 334–354. Springer, Heidelberg, March 2013. `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. *Mathematical Methods in Systems, Optimization, and Control Operator Theory*, 222:259–277, 2012.

Ràf15. Carla Ràfols. Stretching groth-sahai: NIZK proofs of partial satisfiability. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 247–276. Springer, Heidelberg, March 2015. `doi:10.1007/978-3-662-46497-7_10`.

RKP09. Alfredo Rial, Markulf Kohlweiss, and Bart Preneel. Universally composable adaptive priced oblivious transfer. In Hovav Shacham and Brent Waters, editors, *PAIRING 2009*, volume 5671 of *LNCS*, pages 231–247. Springer, Heidelberg, August 2009. `doi:10.1007/978-3-642-03298-1_15`.

Sze20. Alan Szepieniec. Polynomial IOPs for linear algebra relations. Cryptology ePrint Archive, Report 2020/1022, 2020. `https://eprint.iacr.org/2020/1022`.

A More on Section 2

A.1 Matrix Assumptions

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

Let $\iota \in \{1, 2\}$. $\mathcal{D}_{\ell,\mathsf{k}}$-*MDDH*$_{\mathbb{G}_\iota}$ (Matrix Decisional Diffie-Hellman, [EHK+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 $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{kermdh}}_{\mathcal{A},\mathcal{D}_{\ell,\mathsf{k}},\iota,\mathsf{Pgen}}(\lambda) :=$

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

Table 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	$\lvert\mathtt{crs}\rvert$	$\lvert\pi\rvert$	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

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

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

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.

B More on Section 3

B.1 Determinantal Representations

The following problem is well-studied in algebraic geometry, [Har92,Dol10]. Given a homogeneous polynomial $f(X_0, \dots, X_n)$ 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

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

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, $\mathsf{dc}(F) \geq \deg(F)$.

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 8.

QDRs, as defined in Definition 2, additionally have the first column dependence property, which is not required for determinantal representations. Not every determinantal relation is a QDR (see Section 7 for some examples) and thus it is plausible that in general, $\mathsf{qdc}(F) > \mathsf{dc}(F)$.

C More on Section 6

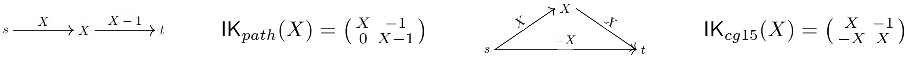
C.1 On OR Proofs

$\boldsymbol{\Pi}^{\vee}_{\mathsf{simple}}$ and $\boldsymbol{\Pi}^{\vee}_{\mathsf{cg}}$. The NIZK argument $\boldsymbol{\Pi}^{\vee}_{\mathsf{simple}}$ (see Fig. 7) for $\mathcal{L}_{\{0,1\}}$ follows from the approach in Section 6.1, by using $\mathsf{abp}^2_{\mathsf{path}}$.

On the other hand, $\boldsymbol{\Pi}^{\vee}_{\mathsf{cg}}$ (see Fig. 7) follows from the approach in Section 6.1, given the ABP in Fig. 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 $\mathsf{lK}(X)$ in in Fig. 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 Fig. 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 $\mathbb{G}_1$ 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 $\boldsymbol{Q}\left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 = -\left[\begin{smallmatrix} \delta \\ \delta \end{smallmatrix}\right]_2$; thus, $[0\|1]_1 \bullet \boldsymbol{Q}\left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 = -[0\|1]_1 \bullet \left[\begin{smallmatrix} \delta \\ \delta \end{smallmatrix}\right]_2$ can be computed in 1 pairing. In the case of $\boldsymbol{\Pi}^{\vee}_{\mathsf{cg}}$, it follows from $\boldsymbol{Q}\left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 = -\left[\begin{smallmatrix} \delta \\ 0 \end{smallmatrix}\right]_2$; thus, $[0\|1]_1 \bullet \boldsymbol{Q}\left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 = -[0\|1]_1 \bullet \left[\begin{smallmatrix} \delta \\ 0 \end{smallmatrix}\right]_2$ can be computed in 1 pairing.



$$\mathsf{IK}_{path}(X) = \left(\begin{smallmatrix} X & -1 \\ 0 & X-1 \end{smallmatrix}\right) \qquad \mathsf{IK}_{cg15}(X) = \left(\begin{smallmatrix} X & -1 \\ -X & X \end{smallmatrix}\right)$$

Fig. 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.

```
$\mathsf{kgen}(\mathsf{p}, \mathtt{lpar})$: $e \leftarrow_\$ \mathbb{Z}_p$; return $(\mathtt{crs}, \mathtt{td}) \leftarrow ([e]_2, e)$ ;
$\mathsf{P}(\mathtt{crs}, \mathtt{lpar}, \mathtt{x}, \mathtt{w})$: $y \leftarrow_\$ \mathbb{Z}_p$; $\boxed{\boldsymbol{\gamma} \leftarrow \left(\begin{smallmatrix} -1 \\ \chi-1 \end{smallmatrix}\right) y}$; $\boxed{\boldsymbol{\gamma} \leftarrow \left(\begin{smallmatrix} -1 \\ \chi \end{smallmatrix}\right) y}$; $\boxed{\boldsymbol{\gamma} \leftarrow \left(\begin{smallmatrix} \chi \\ 1-\chi \end{smallmatrix}\right) y}$;
    $\boldsymbol{\varrho} \leftarrow_\$ \mathbb{Z}_p^2$; $[\mathbf{ct}^{\boldsymbol{\gamma}}]_1 \leftarrow \mathsf{Enc}([\boldsymbol{\gamma}]_1; \boldsymbol{\varrho})$;
    $\boxed{\boxed{w \leftarrow -\chi}}$; $\boxed{w \leftarrow \chi - 1}$; $[\delta]_2 \leftarrow -(w[e]_2 + y[1]_2)$; $[\boldsymbol{z}]_2 \leftarrow \boldsymbol{\varrho}[1]_2 + r \left[\begin{smallmatrix} e \\ \delta \end{smallmatrix}\right]_2$;
    return $\pi = ([\mathbf{ct}^{\boldsymbol{\gamma}}]_1, [\delta, \boldsymbol{z}]_2)$.
$\mathsf{V}(\mathtt{crs}, \mathtt{lpar}, \mathtt{x}, \pi)$: check $[\boldsymbol{I}_2]_2 \bullet [\mathbf{ct}^{\boldsymbol{\gamma}}]_1 + \left[\begin{smallmatrix} e \\ \delta \end{smallmatrix}\right]_2 \bullet [\mathbf{ct}]_1 + \boldsymbol{Q} \left[\begin{smallmatrix} e \\ \delta \end{smallmatrix}\right]_2 \bullet [0\|1]_1 =^? [\boldsymbol{z}]_2 \bullet \mathsf{pk}$;
```

Fig. 7. $\boldsymbol{\Pi}^{\vee}_{\mathsf{simple}}$ (contains $\boxed{\text{boxed}}$ entries), $\boldsymbol{\Pi}^{\vee}_{\mathsf{cg}}$ (contains $\boxed{\text{dashed boxed}}$ entries), and $\boldsymbol{\Pi}^{\vee}_{\mathsf{cds}}$ (contains $\boxed{\text{dotted boxed}}$ entries)

$\boldsymbol{\Pi}^{\vee}_{\mathsf{cds}}$**.** 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 Fig. 7, we also depict the new NIZK argument $\boldsymbol{\Pi}^{\vee}_{\mathsf{cds}}$ that applies Figs. 1 and 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) $[\boldsymbol{z}]_2 = \boldsymbol{\varrho}[1]_2 + r \left[\begin{smallmatrix} e \\ (1-\chi)e-y \end{smallmatrix}\right]_2 = r\left(\begin{smallmatrix} \varrho_1 \\ \varrho_2-ry \end{smallmatrix}\right) [\,1\,]_2 + r \left[\begin{smallmatrix} e \\ e \end{smallmatrix}\right]_2$, if $\chi = 0$,
(2) $[\boldsymbol{z}]_2 = \left(\begin{smallmatrix} \varrho_1 \\ \varrho_2-ry \end{smallmatrix}\right)[1]_2 + r \left[\begin{smallmatrix} e \\ 0 \end{smallmatrix}\right]_2$, 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 $\boldsymbol{Q} \left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 = \left[\begin{smallmatrix} 0 \\ \delta-e \end{smallmatrix}\right]_2$. In particular, $\left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 \bullet [\mathbf{ct}]_1 + \boldsymbol{Q} \left[\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 \bullet [0\|1]_1 = \left[\begin{smallmatrix} e \\ \delta \end{smallmatrix}\right]_2 \bullet [\mathbf{ct}]_1 + \left[\begin{smallmatrix} 0 \\ \delta-e \end{smallmatrix}\right]_2 \bullet [0\|1]_1$ can be computed in 5 pairings. In total, the verifier executes 13 pairings.

C.2 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,DGP$^+$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 6.1 in this case.)

Table 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	$(3.79, 3.15)$

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

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

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,

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

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 $X_j \in \{0, \dots, d-1\}$. One has to execute $\lfloor \log_d N \rfloor$ basic NIZK proofs. The resulting range proof has complexities depicted in Table 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 $d = 3$.

As in the case of the multi-dimensional set-membership proof, an alternative is to use signature-based solutions [RKP09,DGP$^+$19] that offer somewhat better proof size $\Theta(N/\log N)(\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert)$. 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.

D More on Section 7

D.1 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[13] $s = \sqrt{-b/a}$:

$$\boldsymbol{C}(X, Y) = \begin{pmatrix} Y & -s & X \\ X & -1 & s \\ a & X & Y \end{pmatrix} . \tag{5}$$

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.

[13] 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 6.3.

Solving Eq. (2) together with $F(X,Y) = 0$ gives us the following formulas to replace into Fig. 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.

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.

D.2 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 $(X,Y) = (-s,0)$.

D.3 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.

E More on Section 8

E.1 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, 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 8.2, 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 $\mathbb{G}_2$.

F More on Section 9

F.1 CHM NIZK

We describe the CHM (Couteau-Hartmann-Maurer) Σ-protocol and the resulting NIZK, see Fig. 8. For further reference, we state the following results. We refer to Appendix A and [CH20] for unexplained notions and notation.

Proposition 2 (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 3 (Couteau-Hartmann). *Consider the NIZK argument Π_Σ^C, described in Fig. 8, for any algebraic language distribution $\mathcal{D}_{\texttt{lpar}}$ outputting pairs $\texttt{lpar} = [\boldsymbol{\Gamma}, \boldsymbol{\theta}]_1 \in \mathcal{P}_\nu^{n \times t} \times \mathcal{P}_\nu^n$.*

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 $\mathbf{T}_{\texttt{lpar}} \in \mathbb{Z}_p^{n \times n}$, 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.

$$
\boxed{\begin{array}{lcl}
\mathsf{P}(\mathtt{lpar} = [\boldsymbol{\Gamma}, \boldsymbol{\theta}]_1; [\mathtt{x}]_1, \mathtt{w}) & & \mathsf{V}([\boldsymbol{\Gamma}, \boldsymbol{\theta}]_1; [\mathtt{x}]_1) \\
\hline
\boldsymbol{r} \leftarrow_{\$} \mathbb{Z}_p^t & & \\
[\boldsymbol{a}]_1 \leftarrow [\boldsymbol{\Gamma}(\mathtt{x})]_1 \boldsymbol{r} & \xrightarrow{[\boldsymbol{a}]_1} & \\
 & \xleftarrow{e} & e \leftarrow_{\$} \mathbb{Z}_p \\
\boldsymbol{d} \leftarrow e\mathtt{w} + \boldsymbol{r} & \xrightarrow{\boldsymbol{d}} & \\
 & & [\boldsymbol{\Gamma}(\mathtt{x})]_1 \boldsymbol{d} \overset{?}{=} [\boldsymbol{\theta}(\mathtt{x})]_1 e + [\boldsymbol{a}]_1
\end{array}}
$$

$$
\boxed{\begin{array}{lcl}
\mathsf{P}(\mathtt{lpar} = [\boldsymbol{\Gamma}, \boldsymbol{\theta}]_1, \mathtt{crs} = [e]_2; [\mathtt{x}]_1, \mathtt{w}) & & \mathsf{V}(\mathtt{lpar} = [\boldsymbol{\Gamma}, \boldsymbol{\theta}]_1, \mathtt{crs} = [e]_2; [\mathtt{x}]_1) \\
\hline
\boldsymbol{r} \leftarrow_{\$} \mathbb{Z}_p^t & & \\
[\boldsymbol{a}]_1 \leftarrow [\boldsymbol{\Gamma}(\mathtt{x})]_1 \boldsymbol{r} & & \\
[\boldsymbol{d}]_2 \leftarrow [e]_2 \mathtt{w} + \boldsymbol{r}[1]_2 & \xrightarrow{([\boldsymbol{a}]_1, [\boldsymbol{d}]_2)} & \\
 & & [\boldsymbol{\Gamma}(\mathtt{x})]_1 \bullet [\boldsymbol{d}]_2 \overset{?}{=} [\boldsymbol{\theta}(\mathtt{x})]_1 \bullet [e]_2 + [\boldsymbol{a}]_1 \bullet [1]_2
\end{array}}
$$

Fig. 8. The CHM Σ-protocol for algebraic languages $\mathcal{L}_{\boldsymbol{\Gamma},\boldsymbol{\theta}}$ (above) and its Couteau-Hartmann compilation Π_{Σ}^{C} (below)

F.2 More Examples

To simplify parsing, we have omitted the use of bracket notation in examples, writing say 0 instead of $[0]_1$.

Example 3. Let $F(X) = \prod_{i=1}^{4}(X - \xi_i)$. Then

$$
[\boldsymbol{\Gamma}]_1 = \left(\begin{array}{ccc|cccc}
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
-1 & 0 & 0 & \mathsf{sk} & 0 & 0 & 0 \\
\mathsf{ct}_1 & 0 & 0 & 0 & 1 & 0 & 0 \\
\mathsf{ct}_2 - \xi_2 & -1 & 0 & 0 & \mathsf{sk} & 0 & 0 \\
0 & \mathsf{ct}_1 & 0 & 0 & 0 & 1 & 0 \\
0 & \mathsf{ct}_2 - \xi_3 & -1 & 0 & 0 & \mathsf{sk} & 0 \\
0 & 0 & \mathsf{ct}_1 & 0 & 0 & 0 & 1 \\
0 & 0 & \mathsf{ct}_2 - \xi_4 & 0 & 0 & 0 & \mathsf{sk}
\end{array}\right) \in \mathbb{Z}_p^{8\times 7} \quad , \quad [\boldsymbol{\theta}]_1 = \begin{pmatrix} \mathsf{ct}_1 \\ \mathsf{ct}_2 - \xi_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} .
$$

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 $\hat{\boldsymbol{w}} = r\left(\begin{smallmatrix} 1 \\ -\boldsymbol{w} \end{smallmatrix}\right) = r(1 \| \chi - \xi_1 \| (\chi - \xi_1)(\chi - \xi_2) \| (\chi - \xi_1)(\chi - \xi_2)(\chi - \xi_3))$.

Example 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 Fig. 5. Then for $[\mathbf{ct}_1]_1 = \mathsf{Enc}(\chi_1; r_1)$ and $[\mathbf{ct}_2]_1 = \mathsf{Enc}(\chi_2; r_2)$,

$$
[\boldsymbol{\Gamma}]_1 = \left(\begin{array}{ccc|cccc}
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
-1 & 0 & 0 & \mathsf{sk} & 0 & 0 & 0 \\
\mathsf{ct}_{11} & 0 & 0 & 0 & 1 & 0 & 0 \\
\mathsf{ct}_{12} & -1 & 0 & 0 & \mathsf{sk} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & -1 & 0 & 0 & \mathsf{sk} & 0 \\
0 & \mathsf{ct}_{11} & -\mathsf{ct}_{21} & 0 & 0 & 0 & 1 \\
a & \mathsf{ct}_{12} & -\mathsf{ct}_{22} & 0 & 0 & 0 & \mathsf{sk}
\end{array}\right) \quad , \quad [\boldsymbol{\theta}]_1 = \begin{pmatrix} \mathsf{ct}_{11} \\ \mathsf{ct}_{12} \\ 0 \\ 0 \\ \mathsf{ct}_{21} \\ \mathsf{ct}_{21} \\ 0 \\ b \end{pmatrix} .
$$

In this case, $\boldsymbol{w}^\top = (w_1^* \| \dots \| w_3^*) = (-\chi_1 \| -\chi_1^2 \| -\chi_2)$, and

$$
\hat{\boldsymbol{w}} = \begin{pmatrix} w_4^* \\ \dots \\ w_7^* \end{pmatrix} = \left(\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot r_1 + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \cdot r_2 \right) \cdot \left(\begin{smallmatrix} 1 \\ -\boldsymbol{w} \end{smallmatrix}\right)
$$

$$=\begin{pmatrix} r_1 & 0 & 0 & 0 \\ 0 & r_1 & 0 & 0 \\ r_2 & 0 & 0 & 0 \\ 0 & 0 & r_1 & -r_2 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ \chi_1 \\ \chi_1^2 \\ \chi_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_1\chi_1 \\ r_2 \\ r_1\chi_1^2 - r_2\chi_2 \end{pmatrix} \ .$$

Clearly,

$$\begin{aligned} \boldsymbol{\Gamma} \cdot \boldsymbol{w}^* &= \begin{pmatrix} -\chi_1\mathcal{E}(-1;0)+r_1\mathcal{E}(0;1) \\ -\chi_1\mathcal{E}(\chi_1;r_1)-\chi_1^2\mathcal{E}(-1;0)+r_1\chi_1\mathcal{E}(0;1) \\ -\chi_2\mathcal{E}(-1;0)+r_2\mathcal{E}(0;1) \\ -\chi_1\mathcal{E}(a;0)-\chi_1^2\mathcal{E}(\chi_1;r_1)-\chi_2\mathcal{E}(-\chi_2;-r_2)+(r_1\chi_1^2-r_2\chi_2)\mathcal{E}(0;1) \end{pmatrix} \\ &= \begin{pmatrix} \mathcal{E}(\chi_1;r_1) \\ \mathcal{E}(-\chi_1^2;-r_1\chi_1)+\mathcal{E}(\chi_1^2;0)+\mathcal{E}(0;r_1\chi_1) \\ \mathcal{E}(\chi_2;0)+\mathcal{E}(0;r_2) \\ \mathcal{E}(-a\chi_1;0)+\mathcal{E}(-\chi_1^3;-r_1\chi_1^2)+\mathcal{E}(\chi_2^2;r_2\chi_2)+\mathcal{E}(0;r_1\chi_1^2-r_2\chi_2) \end{pmatrix} \\ &= \begin{pmatrix} \mathcal{E}(\chi_1;r_1) \\ \mathcal{E}(0;0) \\ \mathcal{E}(\chi_2;r_2) \\ \mathcal{E}(\chi_2^2-a\chi_1-\chi_1^3;0) \end{pmatrix} \overset{(*)}{=} \begin{pmatrix} \mathcal{E}(\chi_1;r_1) \\ \mathcal{E}(0;0) \\ \mathcal{E}(\chi_2;r_2) \\ \mathcal{E}(b;0) \end{pmatrix} = \mathcal{E}(\boldsymbol{h}(\boldsymbol{\chi})) \ , \end{aligned}$$

where $(*)$ holds iff $F(\boldsymbol{\chi}) = 0$.

F.3 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 Eq. (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

$$\boldsymbol{\Gamma} := \begin{pmatrix} \mathbf{0}_{1\times M_1} & 1 & \mathbf{0}_{1\times M_0} & 1 \\ \mathbf{0}_{N_0\times M_1} & \mathbf{0}_{N_0} & \boldsymbol{\Gamma}_0 & \boldsymbol{\theta}_0 \\ \boldsymbol{\Gamma}_1 & \boldsymbol{\theta}_1 & \mathbf{0}_{N_1\times M_0} & \mathbf{0}_{N_1} \end{pmatrix} \quad , \quad \boldsymbol{\theta} := \begin{pmatrix} -1 \\ \mathbf{0}_{N_0+N_1} \end{pmatrix} \tag{6}$$

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 $[\mathbf{ct}]_1 = [r[1]_1 \| r[\mathsf{sk}]_1 + \xi[1]_1]_1$, 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 Eq. (6) to it for two different values of ξ_i and ciphertexts $[\mathbf{ct}_i]_1$, $i \in \{1, 2\}$, 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 [BBC+13] (see also Example 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 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 2.

Lemma 10. *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 Fig. 8, corresponding to $\boldsymbol{\Gamma}$ of this subsection as in Eq.* (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 $[\boldsymbol{\Gamma}(\mathtt{x})]_1\boldsymbol{r}$ and $[e]_2\mathtt{w} + \boldsymbol{r}[1]_2$.

If $d = 1$ then the multiplication $[\boldsymbol{\Gamma}(\mathtt{x})]_1\boldsymbol{r}$ 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 $\boldsymbol{\Gamma}$ has $6d-2$ non-zero elements, the verifier has to execute $6d-2$ pairings to compute $[\boldsymbol{\Gamma}]_1 \bullet [\boldsymbol{d}]_2$. 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 5

NIWI and New Notions of Extraction for Algebraic Languages

Chaya Ganesh
Hamidreza Khoshakhlagh
Roberto Parisella

In *International Conference on Security and Cryptography for Networks, SCN 2022* ed. by Clemente Galbi and Stanislaw Jarecki. Lecture Notes in Computer Science, vol 13409. Springer, Cham.
https://doi.org/10.1007/978-3-031-14791-3_30

In this thesis we include the extended version published on ePrint. The extended version includes some additional results and all the formal proofs in the appendix.

NIWI and New Notions of Extraction for Algebraic Languages

Chaya Ganesh[1], Hamidreza Khoshakhlagh[2], and Roberto Parisella[3]

[1] Indian Institute of Science
chaya@iisc.ac.in
[2] Aarhus University
hamidreza@cs.au.dk
[3] Simula UiB
robertoparisella@hotmail.it

Abstract. We give an efficient construction of a computational non-interactive witness indistinguishable (NIWI) proof in the plain model, and investigate notions of extraction for NIZKs for algebraic languages. Our starting point is the recent work of Couteau and Hartmann (CRYPTO 2020) who developed a new framework (CH framework) for constructing non-interactive zero-knowledge proofs and arguments under falsifiable assumptions for a large class of languages called algebraic languages. In this paper, we construct an efficient NIWI proof in the plain model for algebraic languages based on the CH framework. In the plain model, our NIWI construction is more efficient for algebraic languages than state-of-the-art Groth-Ostrovsky-Sahai (GOS) NIWI (JACM 2012). Next, we explore knowledge soundness of NIZK systems in the CH framework. We define a notion of strong f-extractability, and show that the CH proof system satisfies this notion.

We then put forth a new definition of knowledge soundness called *semantic extraction*. We explore the relationship of semantic extraction with existing knowledge soundness definitions and show that it is a general definition that recovers black-box and non-black-box definitions as special cases. Finally, we show that NIZKs for algebraic languages in the CH framework cannot satisfy semantic extraction. We extend this impossibility to a class of NIZK arguments over algebraic languages, namely quasi-adaptive NIZK arguments that are constructed from smooth projective hash functions.

1 Introduction

Zero-knowledge proofs, introduced by Goldwasser, Micali and Rackoff [40], are cryptographic primitives that allow a prover to convince a verifier that a statement is true without revealing any other information. Zero-knowledge proof systems have a rich history in cryptography [38,32,12] finding numerous applications in cryptographic constructions such as identification schemes [31], public-key encryption [50], signature schemes [21], anonymous credentials [20], secure multi-party computation [39], and a wide variety of emerging applications.

The notion of zero-knowledge proof was later extended to non-interactive zero-knowledge (NIZK) proofs by Blum, Feldman and Micali [16] where there is a single message sent from the prover to the verifier. NIZKs are particularly useful in low-interaction settings, and feasibility is known for all of NP in the common reference string (CRS) model.

Pairing-based NIZKs. Starting from the work of Groth and Sahai [43], many pairing-based NIZK proof systems have been constructed. These proof systems avoid the need for expensive reductions to NP-complete languages and can directly handle a large class of languages over abelian groups.

Another line of work for constructing pairing-based NIZKs is via a smooth projective hash function (SPHF) [27]. For a language over some abelian group $\mathbb{G}_1$, a secret hashing key is embedded in group $\mathbb{G}_2$, and this NIZK proof can be verified via a pairing operation between $\mathbb{G}_1$ and $\mathbb{G}_2$. The SPHF-based approach leads to very efficient proofs for linear languages. However, they only provide a quasi-adaptive type of soundness, where the CRS can depend on the language.

NIWIs. One can relax the security of a NIZK argument to a Non-Interactive Witness Indistinguishable (NIWI) argument by replacing the zero-knowledge property with a weaker witness indistinguishability (WI) property. Unlike NIZKs for which we know impossibility in the plain model [16], and can therefore only exist in the CRS model, NIWIs are possible in the plain model. Informally, witness indistinguishability means that the verifier at the end of protocol, cannot guess which of the possible witnesses the prover used to compute the proof.

The general idea to construct a NIWI in the plain model, is to start from zero-knowledge proofs that are perfectly sound for some choice of the verifier randomness (or some choice of the CRS). Namely, we let the prover sample the randomness by itself and include additional checks to force the prover to compute at least one proof for such choice of randomness. The first NIWI construction in the plain model was proposed by Barak et al. [8] obtained by derandomizing any two-round zero-knowledge proof (ZAP) [28]. The idea behind the construction is to let the prover send a "high enough" number of proofs, each for a different choice of randomness, such that it is hard to cheat for all of them. There are however drawbacks that make such NIWI schemes unsuitable in practical applications. In the NIWI of [8], the prover has to compute a logarithmic (in the security parameter) number of proofs, which leads to inefficient schemes, both in terms of computation and communication, even starting from efficient, say, linear ZAPs. Also, security is based on a complexity theoretic assumption (namely $\mathbf{E} = \mathbf{DTIME}(2^{O(n)})$ has a function of circuit complexity $2^{\Omega(n)}$) that implies $\mathbf{BPP} = \mathbf{P}$.

Groth, Ostrovsky and Sahai [42] proposed a different framework for NIWI proofs, which leads to more efficient proofs for concrete languages (instead of circuit satisfiability). The key idea in [42] is to force the prover to produce two CRSs, such that at least one of them guarantees perfect soundness. Moreover, the structure of the CRS is such that multiplication of one element can always transform a computationally sound CRS into a perfectly sound CRS. The NIWI proof system can now take advantage of the structure in the CRS as follows: the prover generates a CRS on its own and provides proofs under both the chosen CRS and its transformation. Perfect soundness holds by the fact that at least one of the two CRSs guarantees this property. Some of the issues in the construction of [8] mentioned above are overcome by the NIWI proof system of [42], thanks also to further optimizations [53]. Namely, it is based on well-established assumptions, and the number of proof elements is constant instead of logarithmic in the security parameter. However, for some applications, having communication complexity that is twice the size of a Groth-Sahai (GS) proof is still not practical, particularly considering that GS NIZK, and consequently the NIWI often comes with a drastic efficiency reduction due to the need for reducing the original language to an intermediate language supported by the GS proof system.

In this work, we construct more efficient computational NIWI proofs in the plain model for a larger class of languages.

CH framework. Recently, Couteau and Hartmann [25] developed a new framework (henceforth referred to as the CH framework) for constructing non-interactive zero-knowledge proofs and arguments for a broad class of languages under a falsifiable assumption. They provide several constructions whose efficiency is satisfactory for many applications and enjoy a number of interesting features such as having proofs that are as short as proofs resulting from the Fiat-Shamir transformation applied to Σ-protocols. Their approach, at a very high level, consists of compiling a Σ-protocol over an abelian group $\mathbb{G}_1$ into a non-interactive zero-knowledge argument over Type III pairings by embedding the challenge e into a group $\mathbb{G}_2$ and adding the embedded challenge to the CRS.

The work of [25] also obtains a simple and efficient ZAP argument in the plain model where the WI property holds statistically as opposed to all previous pairing-based constructions that satisfy computational WI. While this ZAP argument can be compiled directly into a non-interactive ZAP using the compiler of [8], the prover, as mentioned above, needs to send logarithmically many proofs, hence decreasing the efficiency of the original scheme.

CH framework with knowledge soundness. All aforementioned proof systems based on CH framework only guarantee soundness meaning that accepting proofs cannot be computed for false statements. Typically, applications require a stronger notion of soundness called *knowledge soundness* which guarantees that the prover *knows* a witness for a statement if it can make the verifier accept. This notion of knowledge soundness is formalized by the existence of an efficient extractor that can extract a valid witness from the prover whenever the prover provides a valid proof. Given that the NIZK systems in [25] only guarantee soundness, we investigate the possibility of knowledge soundness of the CH protocol, and pairing-based arguments in general.

> *Can we construct NIZK proofs in the CH framework with knowledge soundness?*

A naïve solution to provide extractability in the CRS model is to use well-known techniques to augment the statement with a trapdoor for extraction. In particular, given a CRS that contains a public key pk, the most efficient currently known approach is to ask the prover to encrypt the witness under pk and then prove that the ciphertext is computed correctly. The extractor can then use the secret key of pk to recover a valid witness from the proof. This however makes the proof size much larger. On a high level, this is because existing algebraic encryption schemes are not friendly enough with the CH framework, unless we perform the encryption bit-by-bit as in [48,14], which makes the construction undesirable. More importantly, the underlying NP relation is now changed into an augmented relation that should also manage the correctness of ciphertext computations. Our goal is however to study the (im)possibility of extractability for the standard CH framework without changing the underlying relation.

Another solution is to show extractability under knowledge assumptions, or in idealized models such as generic group model (GGM) [56] or algebraic group model (AGM) [34]. Indeed, it is not hard to show that CH NIZKs are knowledge sound in the AGM [4]. Gentry and Wichs [37] show impossibility of a black-box reduction to a falsifiable assumption to prove soundness for succinct arguments, where the proof size is logarithmic in the size of the witness and the statement. However, the use of idealized models or knowledge assumptions to prove knowledge soundness of *non-succinct* proof systems seems to be less justifiable.

At first look, it might seem like knowledge assumptions for extraction are justified since soundness of some CH NIZK is already based on a *non-falsifiable* version of the **extKerMDH** assumption. As per Naor's classification [49], knowledge assumptions are a class of non-falsifiable assumptions. However, since knowledge assumptions stipulate "feasibility" of efficient extraction, they do not fit within a taxonomy of *intractability* assumptions [52]. On the other hand, an assumption such as **extKerMDH**, while non-falsifiable, is still an intractability assumption that can be phrased as a game between an adversary and a challenger, albeit with an inefficient challenger.

1.1 Our Contributions

We study NIZK and NIWI constructions in the pairing-based setting and make the following contributions.

NIWI in the plain model. Different from the aforementioned idea of constructing NIWI in the plain model based on the CH framework [25] using the compiler of [8], we investigate a more efficient strategy inspired by the approach of [42] which allows the verifier to verify if, given a (small) set of CRSs, at least one of them is perfectly binding, without breaking soundness.

Our construction is based on the existence of an efficient algorithm that, given one CRS of the NIZK proof of [25], allows the verifier to check if it is a perfectly binding one without compromising the soundness property. The key idea in constructing such an algorithm is, at a high level, to add two additional group elements to the CRS, chosen such that assuming the existence of Type III pairings, it allows the verifier to (efficiently) check the distribution of the

[4] We show knowledge soundness of the CH argument in the AGM in Appendix D.1.

CRS (with a technique similar to what was done in [2]) while not compromising the WI property. Now, with the verifier equipped with such an algorithm, we construct a non-interactive ZAP by letting the prover compute this CRS and output it together with the proof.

We need additional ideas to prove security of the resulting construction. First, as noted in [25], the soundness of the resulting NIZK proof is based on the special soundness property of the underlying Σ-protocol. Soundness of our NIWI proof follows from the same reasoning and from the correctness of the algorithm that checks the distribution of the CRS. Indeed, if the verifier accepts, then the prover correctly sampled a perfectly binding CRS and thus soundness holds. To show WI, we rely on a new decisional assumption, which we validate in the AGM. The ability of the verifier to check the distribution of the CRS relies on DDH being easy, and therefore it is not possible to rely on DDH for WI.

CH framework with knowledge soundness. The proof and argument systems presented in [25] and our NIWI construction ensure only soundness. As our second contribution, we investigate knowledge soundness of NIZK systems in the CH framework.

f-extractability. We define a notion of *strong f-extractability* that extends related notions of partial extraction used in literature. Informally, an argument system satisfies f-extractability if there exists an efficient extractor that outputs $\widetilde{\mathtt{w}}$ whenever the verifier accepts the proof for statement $\mathtt{x}$, where $\widetilde{\mathtt{w}} = f(\mathtt{w})$ and $\mathtt{w}$ is a valid witness for $\mathtt{x}$. We extend the notion to strong f-extractability where we ask that the partial witness $\widetilde{\mathtt{w}}$ allows for efficiently deciding membership of the statement. We show that the CH proof system satisfies this notion where the extracted value is the encoding of a witness to $\mathbb{G}_2$.

Semantic extraction. We then investigate the possibility of *knowledge soundness* of the CH NIZKs, and pairing-based arguments in general. We show that the CH argument is knowledge sound in the Algebraic Group Model (AGM), and then ask the following question: can we show knowledge soundness in the standard model without relying on knowledge assumptions or show impossibility of extraction? Towards this end, we put forth a notion of extraction called *semantic extraction*, and prove that this notion of extraction is impossible for the CH argument. The intuition behind the definition of semantic extraction is to consider the random coins of the adversary as an input from a certain distribution. This makes it possible to associate a function to each adversary: the function that it computes on certain inputs including its random coins. We then require that adversaries that implement the same function, have the same extractor. We allow the flexibility to split the random coins in two distinct portions, and then allow the extractor to see only one of the two portions. This gives a general definition that, depending on how much randomness we allow the extractor to see, gradually makes the extractor more powerful. We then investigate the relationship between semantic extraction and classic notions of extraction. We show that semantic extraction is a *general* definition, that captures both white-box(n-BB) and black-box(BB) extraction. In particular, BB extraction trivially implies semantic extraction. Also a slightly weaker version of the other direction is true, when we give no randomness to the semantic extractor. Moreover, semantic extraction is equivalent to n-BB extraction, where we give to the extractor all the random coins of the adversary. Finally, we show impossibility of semantic extraction for CH argument: that no extractor that sees only a portion of the adversary's randomness can succeed. We then generalize this impossibility to a class of NIZK arguments over algebraic languages, namely *quasi-adaptive* NIZK arguments based on SPHFs. As a concrete case, we show that the most efficient Quasi-Adaptive NIZK construction of Kiltz and Wee [46] cannot be semantically extractable. While black-box extraction is impossible since the arguments are shorter than the witnesses, the impossibility of semantic extraction is a stronger result. We present this in Appendix D.4.

1.2 Technical Overview

In this section we provide a technical overview of our results. We start with an overview of our NIWI construction in the plain model. Then we discuss our definition of semantic extraction and sketch our impossibility result for semantic extractability of CH NIZKs.

NIWI in the plain model. The starting point of our construction is the NIZK proof for algebraic languages in [25] which is based on a compiler that converts a Σ-protocol with linear answers over a group $\mathbb{G}_1$ into a NIZK argument by embedding the verifier's challenge into a group $\mathbb{G}_2$ in the CRS.

Σ-protocols for linear languages. A linear language with language parameter $[\mathbf{M}]_1 \in \mathbb{G}_1^{n\times k}$ is defined as $\mathcal{L}_{\mathbf{M}} = \left\{[\mathbf{x}]_1 \in \mathbb{G}_1^n | \exists \mathbf{w} \in \mathbb{Z}_p^k : [\mathbf{x}]_1 = [\mathbf{M}]_1 \cdot \mathbf{w}\right\}$. A Σ-protocol for a linear language $\mathcal{L}_{\mathbf{M}}$ with corresponding relation $\mathcal{R}_{\mathbf{M}}$ is a three-move honest-verifier zero-knowledge (HVZK) proof system between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$ with the following syntax. First, $\mathcal{P}$ with an input pair $([\mathbf{x}]_1, \mathbf{w})$ selects $\mathbf{r} \leftarrow \mathbb{Z}_p^k$ and sends a first message $[\mathbf{a}]_1 := [\mathbf{M}]_1 \cdot \mathbf{r} \in \mathbb{G}_1^n$ to $\mathcal{V}$. Next, $\mathcal{V}$ sends a random string $e \in \mathbb{Z}_p$ to $\mathcal{P}$. Finally, $\mathcal{P}$ sends a reply $\mathbf{d} := \mathbf{w}e + \mathbf{r} \in \mathbb{Z}_p^k$ to $\mathcal{V}$, who accepts the proof if $[\mathbf{M}]_1 \cdot \mathbf{d} = [\mathbf{x}]_1 e + [\mathbf{a}]_1$. The *special soundness* property states that for any $[\mathbf{x}]_1$ and any pair of accepting conversations $([\mathbf{a}]_1, e, \mathbf{d}), ([\mathbf{a}]_1, e', \mathbf{d}')$ on $[\mathbf{x}]_1$ where $e \neq e'$, one can efficiently compute a valid witness $\mathbf{w}$ for $[\mathbf{x}]_1$.

CH Compiler. Couteau and Hartmann [25] proposed the following approach to compile a Σ-protocol into a NIZK in the CRS model: the setup algorithm picks a random $e \in \mathbb{Z}_p$ and sets $[e]_2$ as the CRS. The prover computes $[\mathbf{a}]_1$ as in the Σ-protocol, and an embedding of $\mathbf{d}$ in $\mathbb{G}_2$ by computing $[\mathbf{d}]_2 := \mathbf{w} \cdot [e]_2 + \mathbf{r} \cdot [1]_2$. The proof can (publicly) be verified by checking if the pairing equation $[\mathbf{M}]_1[\mathbf{d}]_2 = [\mathbf{x}]_1[e]_2 + [\mathbf{a}]_1[1]_2$ holds. While this leads to an argument system with computational soundness, [25] further shows how to turn the argument into a proof by providing two challenges with two different generators in the CRS and having the prover answer both with the same randomness. The (unconditional) special soundness property of the underlying Σ-protocol now guarantees that a witness exists, resulting in perfect soundness.

The idea behind our NIWI construction is as follows: consider the CRS of the CH NIZK proof $[s_1, s_2, e_1 s_1, e_2 s_2]_2 \in \mathbb{G}_2^4$, where $e_1, e_2, s_1, s_2 \in \mathbb{Z}_p$, and $[e_1, e_2]_2$

play the role of the two challenges (embedded in $\mathbb{G}_2$) in the underlying Σ-protocol. Now, we have the prover pick the CRS and the verifier checks that this CRS computed by a potentially malicious prover is such that $e_1 \neq e_2$, so we can rely on the special soundness of the underlying Σ-protocol. We then prove that the proof is witness-indistinguishable by relying on a new decisional assumption that we show secure in the AGM. This observation leads us to a NIWI proof in the plain model, where we let the prover to choose the "crs" parameters by itself, such that it is verifiable that $e_1 \neq e_2$.

Extractability in the CH framework. We now give an overview of the extractability notions we explore, the new notion of *semantic extraction* we propose, and the impossibility of semantic extraction for CH NIZKs.

The standard definition of knowledge extraction asks for the existence of an efficient algorithm called *extractor* that takes as input a proof π of a statement $\mathtt{x}$ and returns a value $\mathtt{w}'$ such that $\mathtt{w}'$ is a witness for the truth of $\mathtt{x}$, i.e., $(\mathtt{x}, \mathtt{w}') \in \mathcal{R}$. While such *full extractability* captures the fact that the prover must have known the witness, there are instances where the existence of such a powerful extractor is unlikely; however, it is still possible to extract some partial information about the witness. One concrete example is the Groth-Sahai non-interactive proof of knowledge [43] from which one can only extract a one-way function of the witness $f(\mathtt{w})$ where $f : \mathbb{F} \rightarrow \mathbb{G}$ is the encoding of the witness in the underlying group. The barrier to full extractability is the fact that there does not seem to be a trapdoor that can be used to compute, in an efficient way, a witness $\mathtt{w}$ from $f(\mathtt{w})$ (i.e., discrete logarithm problem). To capture this notion of *partial extractability*, Belenkiy et al. [10] formalized the notion of f-extractability by the existence of an efficient algorithm that outputs $\widetilde{\mathtt{w}}$ such that there exists some $\mathtt{w}$ with $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$ and $\widetilde{\mathtt{w}} = f(\mathtt{w})$[5]. In their context of constructing anonymous credentials, f-extractability is used by relaxing the notion of unforgeability to f-unforgeable signatures where the adversary produces $(f(m), \sigma)$ pair (as opposed to (m, σ)) without previously obtaining a signature on m. Since then, f-extractability has been used as a standard property in many privacy-preserving authentication mechanisms [4,18,30,41,44,54].

We begin with this observation that the CH NIZK proof is not only f-extractable for $f := [\cdot]_2$, but the extracted value also allows to decide the membership of the statement via pairing checks. To see this, let $([\mathbf{x}]_1, \mathbf{w})$ be a pair of statement-witness in the linear relation $\mathcal{R}_{\mathbf{M}}$ that returns 1 if $[\mathbf{x}]_1 = [\mathbf{M}]_1 \cdot \mathbf{w}$. One can observe that extracting $[\mathbf{w}]_2$ suffices to decide the membership of $[\mathbf{x}]_1$ by checking if $[\mathbf{M}]_1[\mathbf{w}]_2 = [\mathbf{x}]_1[1]_2$. The primary distinction between partial and full extractability is in the ability to decide membership of the statement being proven via the extracted value. We fill the gap between the two notions by defining a stronger form of partial extractability called *strong f-extractability* which guarantees the existence of an efficient procedure D that for any given statement $\mathtt{x}$ and f-extracted value $\widetilde{\mathtt{w}} := f(\mathtt{w})$, $\mathsf{D}(\mathtt{x}, \widetilde{\mathtt{w}})$ can decide the membership of $\mathtt{x}$. Note that $\widetilde{\mathtt{w}}$ still falls short of being a full witness for the relation; assuming that f is

[5] Note that this a generalization of the standard notion as the identity function $f(\cdot)$ implies full extractability.

one-way, $\widetilde{\mathtt{w}}$ cannot be used to produce a valid proof for $\mathtt{x}$. This is what separates strong f-extractability from full extractability.

Impossibility of Semantic Extraction. We show impossibility of semantic extraction for the CH NIZK argument for algebraic languages. Note that this is a stronger result than ruling out BB extraction. Our impossibility holds only for semantic extraction where there exists a portion of the adversary's randomness that the extractor cannot see.

We now articulate the implications of ruling out semantic extraction for pairing-based arguments. In these systems, a proof consists *only* of group elements, while witnesses are elements of the underlying field[6]. Soundness relies on the hardness of discrete logarithm in order to argue that the exponents of elements in the CRS remain hidden from the prover. As a concrete example, let us consider the CH NIZK argument that essentially compiles a Σ-protocol with three-round messages $([\mathbf{a}], e, \mathbf{d})$ into a NIZK argument in the CRS model in such a way that the CRS includes $[e]_2$ and the proof consists of two (vector of) group elements $([\mathbf{a}]_1, [\mathbf{d}]_2)$. Informally, the security relies on the fact that the prover cannot compute e (or $[e]_1$) and the second component $[\mathbf{d}]_2$ should have been computed as $[\mathbf{d}]_2 = \mathbf{d}_0[1]_2 + \mathbf{d}_1[e]_2$. But now, one can observe that from a *semantic* point of view, there is no distinction between the case that $[\mathbf{d}]_2$ is computed honestly as above and the case where the CRS trapdoor e is used for generating $[\mathbf{d}]_2$ as $\mathbf{d}_0[1]_2 + e[\mathbf{d}_1]_2$. In fact, if an extractor Ext that is limited to being *semantic* is able to extract the witness $\mathbf{d}_1$, then one can invoke Ext to break the discrete logarithm in $\mathbb{G}_2$ by sampling e in the reduction. The above reduction does not go through if Ext is a semantic extractor that has access to all the adversary random coins (we show that such Ext is equivalent to a classic white-box extractor). But as soon as some randomness is hidden from the extractor, we can define an adversary that embeds a DL challenge in this hidden part of the execution, for which no extractor can exist. This means that a valid proof in such argument systems does not prove "knowledge" of $\mathbf{w}$, but only knowledge of $[\mathbf{w}]_1, [\mathbf{w}]_2$, and in order to extract $\mathbf{w}$, one must rely on the hypothesis of asymmetric pairings to conclude that the prover actually knew $\mathbf{w}$ as a field element, which is essentially a knowledge-of-exponent type assumption.

Our results suggest that for most algebraic languages, extracting a witness given the statement $[\mathbf{x}]_1$ is as hard as extracting a witness given $[\mathbf{x}]_1$, a valid proof π together with used randomness r and trapdoor of the CRS e. Thus, if an extractor that is *not* based on knowledge assumption exists, it completely ignores the proof and just recomputes sampling a true statement together with its relative witness. This can also be seen in the following way: consider a language whose hardness relies on the hardness of discrete logarithm. Now, computing the witness from the statement is as hard as discrete logarithm; computing the witness given the statement, a proof, randomness used to compute the proof, and trapdoor is (in the case of CH20) as hard as symmetric discrete logarithm

[6] In structure preserving systems, the witness can be group elements as well, but in this work, we are only interested in proof systems where witnesses are field elements.

(SDL). This implies that either there is a gap between DL and SDL; or computing $\mathbf{w}$ from $[\mathbf{x}]_1$ is as hard as computing $\mathbf{w}$ from $([\mathbf{x}]_1, r, \pi, e)$. In the case of SPHF, both hardness of the language and our result rely on hardness of discrete logarithm, implying that computing $\mathbf{w}$ from $[\mathbf{x}]_1$ is as hard as computing $\mathbf{w}$ from $([\mathbf{x}]_1, \pi, r, \mathtt{td})$. This gives an explanation for why in the pairing-based setting, we have perfect soundness and f-extractability, like we show the CH proof is, while no fully extractable scheme exists under falsifiable assumptions.

2 Preliminaries

Notation. For any positive integer n, $[n]$ denotes the set $\{1, \ldots, n\}$. Let $k \in \mathbb{N}$ be the security parameter. Let $\mathsf{negl}(k)$ be an arbitrary negligible function. We write $a \approx_k b$ if $|a - b| \leq \mathsf{negl}(k)$. Moreover a is a negligible function if $a \approx_k 0$. When a function can be expressed in the form $1 - \mathsf{negl}(k)$, we say that it is overwhelming in k. We use DPT (resp. PPT) to mean a deterministic (resp. probabilistic) polynomial time algorithm. We write $Y \leftarrow \mathsf{F}(X)$ to denote an algorithm with input X and output Y. Further, we write $a \xleftarrow{\$} S$ to denote that a is sampled according to distribution S, or uniformly randomly if S is a set. For two interactive machines $\mathcal{P}$ and $\mathcal{V}$, we denote by $\langle \mathcal{P}(\alpha), \mathcal{V}(\beta) \rangle(\gamma)$ the output of $\mathcal{V}$ after running on private input β with $\mathcal{P}$ using private input α, both having common input γ. All adversaries will be stateful. To represent matrices and vectors, we use bold upper-case and bold lower-case letters, respectively.

2.1 Bilinear Groups

We use additive notation for groups. Throughout the paper we let $\mathcal{G}$ be a bilinear group generator that on input security parameter k returns $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2) \leftarrow \mathcal{G}(1^k)$, where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are groups of prime order p, $[1]_1$ and $[1]_2$ are respectively the generators for $\mathbb{G}_1$ and $\mathbb{G}_2$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate efficiently computable bilinear map such that $\forall [u]_1 \in \mathbb{G}_1, \forall [v]_2 \in \mathbb{G}_2, \forall a, b \in \mathbb{Z}_p : \hat{e}(a[U]_1, b[V]_2) = (ab)\hat{e}([U]_1, [V]_2)$.

We denote $\hat{e}([U]_1, [V]_2)$ as $[U]_1[V]_2$. We consider only type III pairings, where there does not exist an efficient isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$.

2.2 Algebraic Languages

We refer to algebraic languages as the set of languages associated to a relation that can be described by algebraic equations over an abelian group. More precisely, let $\mathtt{gpar} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2) \leftarrow \mathcal{G}(1^k)$. For the rest of the paper, we suppose that these global parameters $\mathtt{gpar}$ are implicitly given as input to each algorithm. Let $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$ be a set of language parameters generated by a polynomial-time algorithm $\mathtt{setup.lpar}$ which takes $\mathtt{gpar}$ as input. Here, $\mathbf{M} : \mathbb{G}^\ell \mapsto \mathbb{G}^{n \times k}$ and $\boldsymbol{\theta} : \mathbb{G}^\ell \mapsto \mathbb{G}^n$ are linear maps such that their different coefficients are not necessarily in the same algebraic structures. Namely, in the most common case, given a bilinear group $\mathtt{gpar} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2)$, they can belong to either $\mathbb{Z}_p$, $\mathbb{G}_1$, $\mathbb{G}_2$, or $\mathbb{G}_T$ as long as the equation $\boldsymbol{\theta}(\mathbf{x}) = \mathbf{M}(\mathbf{x}) \cdot \mathbf{w}$ is "well-consistent". However, in this paper we only use algebraic languages where the statement is defined as elements in $\mathbb{G}_1$. Formally, we define the algebraic language $\mathcal{L}_{\mathtt{lpar}} \subset \mathcal{X}_{\mathtt{lpar}}$ as

$$\mathcal{L}_{\mathtt{lpar}} = \left\{ [\mathbf{x}]_1 \in \mathbb{G}_1^{\ell} | \exists \mathbf{w} \in \mathbb{Z}_p^k : [\boldsymbol{\theta}(\mathbf{x})]_1 = [\mathbf{M}(\mathbf{x})]_1 \cdot \mathbf{w} \right\} \quad . \tag{1}$$

An algebraic language where $\mathbf{M}$ is independent of $\mathbf{x}$ and $\boldsymbol{\theta}$ is the identity is called a *linear language*. We sometimes require algebraic languages to satisfy a property we call 1DL-friendly. Roughly, this is to enable the embedding of a symmetric simple discrete logarithm challenge, which is given as a pair of group elements, into an algebraic statement in the reduction. We give the definition(Definition 13) in Appendix A.4. We note that algebraic languages are as expressive as NP, since every Boolean circuit can be represented by sets of linear equations.

2.3 Non-interactive Zero-knowledge Arguments

A NIZK (non-interactive zero-knowledge) argument Π, for a family of languages $\mathcal{L}_{\mathtt{lpar}}$ consists of four PPT algorithms.

- CRSGen on input a security parameter 1^k generates a pair $(\mathtt{crs}, \mathtt{td})$.
- $\mathcal{P}$ on input a $\mathtt{crs}$, a statement $\mathtt{x}$ and a witness $\mathtt{w}$, computes a proof π.
- $\mathcal{V}$ on input a $\mathtt{crs}$, a statement $\mathtt{x}$ and a proof π outputs 1 (accept) or 0 (reject).
- Sim on input $\mathtt{td}$, a true statement $\mathtt{x}$ computes a simulated proof π.

Here we are implicitly supposing that $\mathtt{lpar}$ is always given as input. We assume that each $\mathtt{td}$ corresponds to only one $\mathtt{crs}$ and also that given $\mathtt{td}$ it is possible to efficiently and deterministically compute the corresponding $\mathtt{crs}$. This is w.l.o.g., since it is always possible to define the trapdoor in a way that the previous property is satisfied. The following properties are required for a NIZK argument:

- *Perfect completeness*: for any pair of true statement $\mathtt{x}$ with a relative witness $\mathtt{w}$, for any $\mathtt{crs}$ computed by CRSGen
$$\Pr\left[\mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 | \pi \leftarrow \mathcal{P}(\mathtt{crs}, \mathtt{x}, \mathtt{w})\right] = 1.$$
- *Computational soundness*: for any PPT adversary $\mathcal{A}$
$$\Pr\left[\begin{array}{c} \mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 \\ \wedge\ \mathtt{x} \notin \mathcal{L}_{\mathtt{lpar}} \end{array} \middle| \begin{array}{c} (\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k); \\ (\mathtt{x}, \pi) \leftarrow \mathcal{A}(\mathtt{crs}) \end{array}\right] \leq \mathsf{negl}(k)$$
- *(Perfect) zero-knowledge*: for any true statement, witness pair $(\mathtt{x}, \mathtt{w})$, for any $(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k)$ the following distributions are identical
$$\mathcal{P}(\mathtt{crs}, \mathtt{x}, \mathtt{w}) \equiv \mathsf{Sim}(\mathtt{crs}, \mathtt{td}, \mathtt{x}).$$

If the zero-knowledge property requires the two distributions to only be computationally insitinguishable, then we get a computational NIZK. If soundness holds even against unbounded adversaries, we say that the protocol is a NIZK proof system, with perfect soundness. We say that Π is black-box knowledge sound if there exists an efficient extractor that computes a witness, given a statement, an accepting proof and the $\mathtt{crs}$ trapdoor.

Definition 1 (BB Knowledge soundness). *Let $\Pi = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$ be a NIZK argument for the relation $\mathcal{R}_{\mathtt{lpar}}$, defined by some language parameter* $\mathtt{lpar}$. *We say that Π is black-box knowledge sound, if there exists an extractor $\mathsf{Ext}_{\mathsf{bb}}$ such that, for any PPT adversary $\mathcal{A}$:*

$$\Pr\left[\begin{array}{l}\mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 \\ \wedge(\mathtt{x}, \mathtt{w}) \notin \mathcal{R}_{\mathtt{lpar}}\end{array}\middle|\begin{array}{c}(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k); \\ (\mathtt{x}, \pi) \leftarrow \mathcal{A}(\mathtt{crs}, \mathtt{lpar}; r); \mathtt{w} \leftarrow \mathsf{Ext}_{\mathsf{bb}}(\mathtt{td}, \mathtt{x}, \pi)\end{array}\right] \leq \mathsf{negl}(k)$$

where r is the random coins of the adversary.

If the extractor is allowed to depend on the adversary and we also give it as additional input, the random coins used by the adversary, we say that Π is white-box knowledge sound.

Definition 2 (n-BB Knowledge soundness). *Let $\Pi = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$ be a NIZK argument for the relation $\mathcal{R}_{\mathtt{lpar}}$, defined by some language parameter* $\mathtt{lpar}$. *We say that Π is white-box knowledge sound, if for any PPT adversary $\mathcal{A}$, there exists an efficient extractor $\mathsf{Ext}_{\mathsf{wb},\mathcal{A}}$ such that:*

$$\Pr\left[\begin{array}{l}\mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 \\ \wedge(\mathtt{x}, \mathtt{w}) \notin \mathcal{R}_{\mathtt{lpar}}\end{array}\middle|\begin{array}{c}(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k); \\ (\mathtt{x}, \pi) \leftarrow \mathcal{A}(\mathtt{crs}, \mathtt{lpar}; r); \mathtt{w} \leftarrow \mathsf{Ext}_{\mathsf{wb},\mathcal{A}}(\mathtt{td}, \mathtt{x}, \pi, r)\end{array}\right] \leq \mathsf{negl}(k)$$

where r is the random coins of $\mathcal{A}$.

We also consider the concrete security variants of the above definitions. Roughly, Π is (t, ϵ)-BB knowledge sound if the extraction property holds with respect to all $t(k)$-time bounded provers (as opposed to all PPT provers), and that the extractor succeeds except with probability ϵ (as opposed to being negligible). We give the formal definitions of the concrete-security versions in Appendix D.2.

Lastly, we state the witness indistinguishability definition for non-interactive protocols. Recall that we are interested in non-interactive witness indistinguishable proof systems in the plain model without a trusted setup.

Definition 3 (Witness Indistinguishability (WI)). *A non-interactive proof system $\Pi = (\mathcal{P}, \mathcal{V})$ for language $\mathcal{L}_{\mathtt{lpar}}$ is WI if for every PPT verifier $(\mathcal{V}_1^*, \mathcal{V}_2^*)$, for all $(\mathtt{x}, \mathtt{w}_1, \mathtt{w}_2)$ such that $(\mathtt{x}, \mathtt{w}_1) \in \mathcal{R}_{\mathtt{lpar}}, (\mathtt{x}, \mathtt{w}_2) \in \mathcal{R}_{\mathtt{lpar}}$, we have*

$$\Pr\left[b \leftarrow \mathcal{V}_2^*(\mathsf{st}, \pi) \middle| (\mathtt{x}, \mathtt{w}_1, \mathtt{w}_2, \mathsf{st}) \leftarrow \mathcal{V}_1^*(\mathtt{lpar}); b \xleftarrow{\$} \{0,1\}; \pi \leftarrow \mathcal{P}(\mathtt{lpar}, \mathtt{x}, \mathtt{w}_b)\right] \approx_k \frac{1}{2}$$

2.4 From Σ-protocols to NIZKs

Recently, Couteau and Hartmann [25] propose a new approach for building pairing-based non-interactive zero-knowledge arguments for algebraic languages. At a high level, their approach is based on compiling a Σ-protocol (see Appendix A.1) into a non-interactive zero-knowledge argument by embedding the challenge in $\mathbb{G}_2$ and publishing it once in the $\mathtt{crs}$. The NIZK argument is depicted in Fig. 2, where we denote as $\mathcal{S}_\Sigma$ the simulator for special honest verifier zero-knowledge property of the Σ-protocol. A variant of their compiler yields NIZK *proofs*, depicted in Fig. 3, based on standard assumptions. We refer to Appendix A.5 for more details.

$$\begin{array}{lcl}
\mathcal{P}(\mathtt{lpar},[\mathbf{x}],\mathbf{w}) & & \mathcal{V}(\mathtt{lpar},[\mathbf{x}]) \\
\mathbf{r} \leftarrow \mathbb{Z}_p^k & & \\
[\mathbf{a}] := [\mathbf{M}(\mathbf{x})]\mathbf{r} & & \\
 & \xrightarrow{\quad [\mathbf{a}] \quad} & \\
 & & e \leftarrow \mathbb{Z}_p \\
 & \xleftarrow{\quad e \quad} & \\
\mathbf{d} := \mathbf{w}e + \mathbf{r} & & \\
 & \xrightarrow{\quad \mathbf{d} \quad} & \\
 & & [\mathbf{M}(\mathbf{x})]\mathbf{d} \stackrel{?}{=} [\boldsymbol{\theta}(\mathbf{x})]e + [\mathbf{a}]
\end{array}$$

Fig. 1: Σ-protocol for algebraic language $\mathcal{L}_{\mathtt{lpar}}$ with $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$

```
$\mathsf{CRSGen}(1^k)$
    $\mathsf{gpar} \leftarrow \mathsf{setup.gpar}(1^k)$
    $e \leftarrow \mathbb{Z}_p$
    $\mathtt{crs} := (\mathsf{gpar}, [e]_2); \mathtt{td} := e$
    return $(\mathtt{crs}, \mathtt{td})$

$\mathcal{P}(\mathtt{lpar}, \mathtt{crs}, [\mathbf{x}]_1, \mathbf{w})$
    $\mathbf{r} \leftarrow \mathbb{Z}_p^k$
    $[\mathbf{a}]_1 := [\mathbf{M}(\mathbf{x})]_1 \mathbf{r}$
    $[\mathbf{d}]_2 := \mathbf{w}[e]_2 + [\mathbf{r}]_2$
    return $\pi := ([\mathbf{a}]_1, [\mathbf{d}]_2)$

$\mathsf{Sim}(\mathtt{lpar}, \mathtt{crs}, e, [\mathbf{x}]_1)$
    $([\mathbf{a}]_1, \mathbf{d}) := \mathcal{S}_\Sigma([\mathbf{x}]_1, e)$
    return $\pi := ([\mathbf{a}]_1, [\mathbf{d}]_2)$

$\mathcal{V}(\mathtt{lpar}, \mathtt{crs}, [\mathbf{x}]_1, \pi = ([\mathbf{a}]_1, [\mathbf{d}]_2))$
    $[\mathbf{M}(\mathbf{x})]_1 \cdot [\mathbf{d}]_2 \stackrel{?}{=} [\boldsymbol{\theta}(\mathbf{x})]_1 \cdot [e]_2 + [\mathbf{a}]_1 \cdot [1]_2$
```

Fig. 2: NIZK argument for algebraic language $\mathcal{L}_{\mathtt{lpar}}$ with $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$ [25]

2.5 Cryptographic Assumptions

The DL (discrete logarithm) assumption in group $\mathbb{G}_\iota$ of order p states that it is hard to compute the discrete logarithm of a random element in $\mathbb{G}_\iota$.

Assumption 1 (Discrete logarithm assumption) *For any PPT adversary $\mathcal{A}$, it holds that:*

$$\Pr\left[\, w[1]_\iota = [x]_\iota \,\middle|\, w \leftarrow \mathcal{A}([1, x]_\iota) \,\right] \leq \mathsf{negl}(k)$$

where x is sampled from the uniform distribution over $\mathbb{Z}_p$.

Assumption 2 (Symmetric discrete logarithm assumption) *For any PPT adversary $\mathcal{A}$, it holds that:*

$$\Pr\left[\, w[1]_\iota = [x]_\iota \ ; \ \iota = 1, 2 \,\middle|\, w \leftarrow \mathcal{A}([1, x]_1, [1, x]_2) \,\right] \leq \mathsf{negl}(k)$$

where x is sampled from the uniform distribution over $\mathbb{Z}_p$.

```
$\mathsf{CRSGen}(1^k)$
    $s_1, s_2, e_1, e_2 \leftarrow \mathbb{Z}_p$
    $\mathtt{crs} := ([s_1, s_2, s_1e_1, s_2e_2]_2)$
    return $\mathtt{crs}$

$\mathcal{P}(\mathtt{lpar}, \mathtt{crs}, [\mathbf{x}]_1, \mathbf{w})$
    $\mathbf{r} \leftarrow \mathbb{Z}_p^k$
    $[\mathbf{a}]_1 := [\mathbf{M}(\mathbf{x})]_1\mathbf{r}$
    $[\mathbf{d}_i]_2 := \mathbf{w}[s_ie_i]_2 + \mathbf{r}[s_i]_2$
    return $\pi := ([\mathbf{a}]_1, [\mathbf{d}_1, \mathbf{d}_2]_2)$

$\mathsf{Sim}(\mathtt{lpar}, [\mathbf{x}]_1)$
    $e, s_1, s_2 \leftarrow \mathbb{Z}_p$
    $([\mathbf{a}]_1, \mathbf{d}) := \mathcal{S}_\Sigma([\mathbf{x}]_1, e)$
    $\mathtt{crs} = ([s_1, s_2, s_1e, s_2e]_2)$
    $\pi := ([\mathbf{a}]_1, [\mathbf{d}s_1, \mathbf{d}s_2]_2)$
    return $(\mathtt{crs}, \pi)$

$\mathcal{V}(\mathtt{lpar}, \mathtt{crs}, [\mathbf{x}]_1, \pi = ([\mathbf{a}]_1, [\mathbf{d}_1, \mathbf{d}_2]_2))$
    for $i \in \{1, 2\}$ check
        $[\mathbf{M}(\mathbf{x})]_1 \cdot [\mathbf{d}_i]_2 \overset{?}{=} [\boldsymbol{\theta}(\mathbf{x})]_1 \cdot [s_ie_i]_2 + [\mathbf{a}]_1 \cdot [s_i]_2$
```

Fig. 3: NIZK proof for algebraic language $\mathcal{L}_{\mathtt{lpar}}$ with $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$ [25]

The co-CDH assumption was first proposed in [17]. Later a modified version of the assumption was proposed in [51] which we adapt as follows.

Assumption 3 (Computational co-Diffie-Hellman (co-CDH) assumption) *For any PPT adversary $\mathcal{A}$, it holds that:*

$$\Pr\left[\, [xy]_2 \leftarrow \mathcal{A}([1, x]_1, [1, x, y]_2) \,\right] \leq \mathsf{negl}(k)$$

where x, y are sampled from the uniform distribution over $\mathbb{Z}_p$.

3 NIWI Proof in the Plain Model

Our NIWI proof system in the plain model is given in Fig. 4. We show that our construction is perfectly sound and computationally WI. To show WI, we rely on a new assumption that we validate in the algebraic group model (AGM) in Appendix B.3. While it might seem like we can show WI by relying on DDH in the second group and then invoking the WI of the underlying sigma protocol, the presence of $[s_2]_1$ in the proof makes this impossible. In fact, we rely on DDH being easy for perfect soundness by enabling the verifier to check that the two challenges are indeed distinct. We show that the new assumption holds in the AGM introduced by Fuchsbauer, Kiltz and Loss [34]. The model is a relaxation of the generic group model [56] that captures adversaries exploiting the representation of the underlying group, and has been shown to be useful in reasoning about security properties of various constructions [47,35,23]. The work of [55] extends this model to handle decisional assumptions by introducing the notion of algebraic distinguishers. We use this model to show the algebraic equivalence

```
$\mathcal{P}(\texttt{lpar}, [\mathbf{x}]_1, \mathbf{w})$
$s_1, s_2, e_1, e_2 \leftarrow \mathbb{Z}_p$ s.t $e_1 \neq e_2$
$\mathbf{r} \leftarrow \mathbb{Z}_p^k$
$[\mathbf{a}]_1 := [\mathbf{M}(\mathbf{x})]_1 \mathbf{r}$
$\mathbf{d}_i := s_i e_i \mathbf{w} + s_i \mathbf{r}$ for $i = 1, 2$
return $\pi := \Big([\mathbf{a}, s_1, s_2]_1,$
    $[s_1, s_2, s_1 e_1, s_2 e_2, \mathbf{d}_1, \mathbf{d}_2]_2\Big)$

$\mathcal{V}(\texttt{lpar}, [\mathbf{x}]_1, \pi)$
parse $\pi$ as $\Big([\mathbf{a}, c_1, c_2]_1, [s_1, s_2, E_1, E_2, \mathbf{d}_1, \mathbf{d}_2]_2\Big)$
accept if all the following checks pass
$[c_i]_1[1]_2 \overset{?}{=} [1]_1[s_i]_2$ for $i \in \{1, 2\}$    (1)
$[c_2]_1[E_1]_2 \overset{?}{\neq} [c_1]_1[E_2]_2$    (2)
for $i \in \{1, 2\}$ :
    $[\mathbf{M}(\mathbf{x})]_1[\mathbf{d}_i]_2 \overset{?}{=} [\boldsymbol{\theta}(\mathbf{x})]_1[E_i]_2 + [\mathbf{a}]_1[s_i]_2$    (3)
```

Fig. 4: NIWI proof for algebraic language $\mathcal{L}_{\texttt{lpar}}$ with $\texttt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$

```
$G_{\mathbf{ADHR},b}(\mathcal{A}, \texttt{lpar})$
$([\mathbf{x}]_1, \mathbf{w}_0, \mathbf{w}_1) \leftarrow \mathcal{A}([1]_1, [2]_2, \texttt{lpar})$;
$s_1, s_2, e_1, e_2 \leftarrow \mathbb{Z}_p; \mathbf{r} \leftarrow \mathbb{Z}_p^k; (e_1 \neq e_2)$;
$\pi = ([\mathbf{M}(\mathbf{x})\mathbf{r}, s_1, s_2]_1, [s_1, s_1 e_1, s_2, s_2 e_2, s_1 e_1 \mathbf{w}_b + s_1 \mathbf{r}, s_2 e_2 \mathbf{w}_b + s_2 \mathbf{r}]_2)$;
$b' \leftarrow \mathcal{A}([\mathbf{M}(\mathbf{x})]_1, \mathbf{w}_0, \mathbf{w}_1, \pi)$;
if $b = b'$ then return 1; else return 0 fi ;
```

Fig. 5: Algebraic decisional hidden range games $G_{\mathbf{ADHR},i}$.

between our assumption and *symmetric power discrete logarithm* (SPDL) assumption. While the assumption we make is a tautological assumption, we hope it will be analysed further and will find other applications, just like the tautological Kiltz-Wee assumption for QA-NIZK [46,3]. We believe it is an interesting open problem to prove the security of our construction under standard decisional assumptions.

Assumption 4 (Algebraic decisional hidden range) *Let* $\texttt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$ *be any pair of language parameter that defines the algebraic language* $\mathcal{L}_{\texttt{lpar}}$*. Let* $G_{\boldsymbol{ADHR},i}$*, for* $i \in \{0, 1\}$ *be the games depicted in Fig. 5. The* $(\mathbf{M}, \boldsymbol{\theta})$*-**ADHR** assumption states that for any PPT adversary* $\mathcal{A}$*,*

$$\mathbf{Adv}_{\mathcal{A},\texttt{lpar}}^{G_{\boldsymbol{ADHR},0,1}} = |\Pr[G_{\boldsymbol{ADHR},0}(\mathcal{A}, \texttt{lpar}) = 1] - \Pr[G_{\boldsymbol{ADHR},1}(\mathcal{A}, \texttt{lpar}) = 1]| \leq \mathsf{negl}(k).$$

Theorem 1. *For any algebraic language* $\mathcal{L}_{\texttt{lpar}}$*, with* $\texttt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$*, the protocol in Fig. 4 is a non-interactive witness indistinguishable proof under the* $(\mathbf{M}, \boldsymbol{\theta})$*-**ADHR** assumption.*

Proof. (Perfect completeness). We show that an honest prover convinces an honest verifier with probability 1. For an honestly generated proof $\pi = ([\mathbf{a}, c_1, c_2]_1, [s_1, s_2, E_1, E_2, \mathbf{d}_1, \mathbf{d}_2]_2)$, by construction, we have that $c_i = s_i^{-1}$, $E_i = s_i e_i$ and $\mathbf{d}_i = s_i(e_i \mathbf{w} + \mathbf{r})$. It is easy to see that all the verifier checks pass.

1. $[c_i]_1[s_i]_2 = [s_i^{-1}]_1[s_i]_2 = [1]_T$.
2. $[c_1]_1[E_1]_2 = [s_1^{-1}]_1[s_1e_1]_2 = [e_1]_T$, and $[c_2]_1[E_2]_2 = [s_2^{-1}]_1[s_2e_2]_2 = [e_2]_T$, and since $e_1 \neq e_2$, we have $[c_1]_1[E_1]_2 \neq [c_2]_1[E_2]_2$.
3. $\mathbf{M}(\mathbf{x})\mathbf{d}_i = s_ie_i\mathbf{M}(\mathbf{x})\mathbf{w} + s_i\mathbf{M}(\mathbf{x})\mathbf{r} = E_i\boldsymbol{\theta}(\mathbf{x}) + \mathbf{a}s_i$.

(Perfect soundness). Let $\mathcal{A}$ be any (possibly unbounded) adversary that breaks the soundness property by outputting a proof $\tilde{\pi} = ([\tilde{a}, \tilde{c}_1, \tilde{c}_2]_1, [\tilde{s}_1, \tilde{s}_2, \tilde{E}_1, \tilde{E}_2, \tilde{d}_1, \tilde{d}_2]_2)$ relative to an (adaptively) chosen statement $\mathtt{x} = [\mathbf{x}]_1 \notin \mathcal{L}_{\mathtt{lpar}}$, such that the NIWI verifier accepts $\tilde{\pi}$. We show that such an accepting proof contradicts with the assumption that $\mathtt{x} \notin \mathcal{L}_{\mathtt{lpar}}$. In what follows, the index i will always be used as for each $i \in \{1, 2\}$.

From the verifier's check (1), it must be that $\tilde{c}_i = \tilde{s}_i$. Moreover, from check (3) we have that $\mathbf{M}(\mathbf{x})\tilde{d}_i = \boldsymbol{\theta}(\mathbf{x})\tilde{E}_i + \tilde{a}\tilde{s}_i$, which means that $\mathbf{M}(\mathbf{x})\tilde{d}_i/\tilde{c}_i = \boldsymbol{\theta}(\mathbf{x})\tilde{E}_i/\tilde{c}_i + \tilde{a}$. Now, since the NIWI verifier accepts the proof, from check (2), we have that $\tilde{c}_2\tilde{E}_1 \neq \tilde{c}_1\tilde{E}_2$. Therefore, there exists a pair of valid transcripts $([\tilde{a}]_1, \tilde{E}_i/\tilde{c}_i, \tilde{d}_i/\tilde{c}_i)$ for $\mathtt{x}$, with the same first message $[\tilde{a}]_1$ and different challenges. From special soundness of the underlying Σ-protocol, there exists an extractor that outputs a witness for $\mathtt{x}$ given two such transcripts. This contradicts the assumption that $\mathtt{x} \notin \mathcal{L}_{\mathtt{lpar}}$.

(Witness indistinguishability). Let $\mathcal{L}_{\mathtt{lpar}}$ be an algebraic language with $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$. Let $\mathcal{A}$ be a PPT adversary that wins the WI game with non-negligible probability ϵ. We build an efficient adversary $\mathcal{B}$ against $(\mathbf{M}, \boldsymbol{\theta})$-**ADHR** assumption as follows: $\mathcal{B}$ first calls $\mathcal{A}$ and obtains $\mathsf{st} = ([\mathbf{x}]_1, \mathbf{w}_0, \mathbf{w}_1)$. It then outputs st and receives π from the challenger. Lastly, $\mathcal{B}$ calls $\mathcal{A}$ on π and returns $\mathcal{A}$'s decision bit. Since the challenger of $G_{\mathbf{ADHR},i}$ computes π exactly as the honest prover of the NIWI in Fig. 4, $\mathcal{B}$ breaks the assumption with the same non-negligible probability ϵ.

□

We discuss the efficiency of our construction and applications of NIWI in the plain model in Appendix B.

4 Partial Extractability for the CH Framework

In this section, we first recall the definition of f-extractability and show the NIZK proof system in Fig. 3 is $[\cdot]_2$-extractable. Next, we strengthen this property by introducing a new notion called *strong f-extractability* where the partial witness $\widetilde{\mathtt{w}}$ can be used by an efficient algorithm to decide membership of the statement. In more detail, here we also require the existence of an efficiently computable decision procedure D such that for $\widetilde{\mathtt{w}} = f(\mathtt{w})$ output by the extractor, $\mathsf{D}(\mathtt{x}, \widetilde{\mathtt{w}})$ decides membership of $\mathtt{x}$ (i.e., $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$ iff $\mathsf{D}(\mathtt{x}, \widetilde{\mathtt{w}}) = 1$). However, $\widetilde{\mathtt{w}}$ falls short of being a witness for the relation; assuming that f is one-way, $\widetilde{\mathtt{w}}$ cannot be used to produce a valid proof for $\mathtt{x}$.

Definition 4 (f-extractability [10]). *Let $\Pi = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$ be a NIZK argument for the relation $\mathcal{R}$, defined by some language parameter* $\mathtt{lpar}$.

Let f be an efficiently computable function. We say that Π is (black-box) f-extractable if there exists a PPT extractor Ext *such that for any PPT adversary that returns an accepting proof π for a statement* $\mathtt{x}$*,* Ext *outputs a value* $\widetilde{\mathtt{w}}$ *for which there exists some* $\mathtt{w}$ *such that* $(\mathtt{x},\mathtt{w})\in\mathcal{R}$ *and* $\widetilde{\mathtt{w}}=f(\mathtt{w})$ *with overwhelming probability. More formally, for any PPT adversary $\mathcal{A}$, we have*

$$\Pr\left[\begin{array}{l|c} \mathcal{V}(\mathtt{crs},\mathtt{x},\pi)=1 & (\mathtt{crs},\mathtt{td})\leftarrow \mathsf{CRSGen}(1^k); \\ \wedge(\mathtt{x},f^{-1}(\widetilde{\mathtt{w}}))\notin\mathcal{R} & (\mathtt{x},\pi)\leftarrow\mathcal{A}(\mathtt{crs},\mathtt{lpar};r);\widetilde{\mathtt{w}}\leftarrow\mathsf{Ext}(\mathtt{td},\mathtt{x},\pi)\end{array}\right]\le\mathsf{negl}(k)$$

where r is the random coins of the adversary.

We show that the CH proof system satisfies f-extractability where $f(x)$ is the encoding of x to $\mathbb{G}_2$. We state the lemma below and give the proof in Appendix C.

Lemma 1. *The NIZK proof system of [25] depicted in Fig. 3 is $[\cdot]_2$-extractable.*

4.1 Strong f-extractability

We now define strong f-extractability as an strengthening of f-extractability where the extracted value further allows to decide membership of the statement (although it cannot be used to produce a valid proof for it).

Definition 5 (Strong f-extractability). *Let* $\Pi=(\mathsf{CRSGen},\mathcal{P},\mathcal{V},\mathsf{Sim})$ *be a NIZK argument for the relation $\mathcal{R}$, defined by some language parameter* $\mathtt{lpar}$*. Let f be an efficiently computable function. We say that Π is strong f-extractable if the following properties hold:*

Extractability. *Π is f-extractable (see Definition 4).*

Decidability. *There exists a DPT algorithm* D*, such that for any statement* $\mathtt{x}$ *and string* $\widetilde{\mathtt{w}}$*, it holds that* $\mathsf{D}(\mathtt{x},\widetilde{\mathtt{w}})=1$ *iff* $(\mathtt{x},\mathtt{w})\in\mathcal{R}$*, where* $\widetilde{\mathtt{w}}=f(\mathtt{w})$*.*

One-wayness. *For any* $(\mathtt{x},\widetilde{\mathtt{w}})$ *sampled uniformly at random s.t* $\mathsf{D}(\mathtt{x},\widetilde{\mathtt{w}})=1$*, if there exists a PPT adversary $\mathcal{A}$ and a polynomial $p(\cdot)$, such that*

$$\Pr\left[\mathcal{V}(\mathtt{crs},\mathtt{x},\pi')=1\middle|\pi'\leftarrow\mathcal{A}(\mathtt{crs},\mathtt{x},\widetilde{\mathtt{w}})\right]\ge\frac{1}{p(k)},$$

there exists a PPT algorithm $\mathcal{I}$, and polynomial $q(\cdot)$ such that

$$\Pr\left[f(\bar{\mathtt{w}})=\widetilde{\mathtt{w}}\middle|\bar{\mathtt{w}}\leftarrow I(\widetilde{\mathtt{w}})\right]\ge\frac{1}{q(k)}.$$

Remark 1. Similar to Definition 4, strong f-extractability is defined without any restriction on f and hence it can recover full extractability for the case when f is the identity function. However, we only focus on strong f-extractability for non-trivial f in this work. Having no restriction on f in Definitions 4 and 5 makes strong f-extractability a middle ground between full and f-extractability.

We show in Appendix C.2 that the proof system in Fig. 3 is strong $[\cdot]_2$-extractable under a standard hardness assumption. We remark that it is not clear whether the argument system in Fig. 2 satisfies strong f-extractability. Intuitively, if it did, then such an algorithm could likely be used to compute the witness $\mathbf{w}$ in the case of the underlying Σ-protocol, given only one transcript, which is impossible by SHVZK.

5 Full Extractability for the CH Framework

The CH argument system from Fig. 2 is knowledge sound in the AGM. (We show in Sec. D.1). Now, we turn to showing limitations of proving knowledge soundness.[7] We begin this section by defining a notion of knowledge soundness called *semantic extraction.* We study the relationship between semantic knowledge soundness and standard notions of black-box (BB) and white-box (n-BB) knowledge soundness. Then, we show impossibility of the existence of semantic extractors for the CH argument system in Fig. 2. The generalization of this impossibility to quasi-adaptive NIZK arguments constructed from SPHFs is in Appendix D.4.

Notation. We introduce some additional notation for this section. We denote by $\mathbf{CRS}$ the set of all possible $\texttt{crs}$'s. We denote by χ the set of the statements $\texttt{x}$ and by $\mathit{\Psi}$ the set of all possible proofs π We also split the randomness of PPT-s into two strings s and t. We denote by $\mathbf{\Gamma_t}$ the set of all possible strings t and by $\mathbf{\Gamma_s}$ the set of all possible strings s. Looking ahead, for adversarial provers, this split, at a high level, is to distinguish between the portion of randomness that is provided to the semantic extractor (t), and the portion that is not (s). Note that, while $\mathbf{CRS}, \chi, \mathit{\Psi}$ are defined by the NIZK construction, the randomness spaces are not fixed by the NIZK. We only assume that s, t have polynomial size.

5.1 Semantic Extractor

We now define our new notion of extraction. Informally, this extractor inverts the "semantic" function implemented by an adversarial prover regardless of *how* the computation was done. The key difference from n-BB notion is that we will not ask for a different extractor for every PPT $\mathcal{A}$, instead, we ask for an extractor associated with a function f; this extractor is universal for all TMs (even unbounded ones) that implement f. We begin by modeling the function implemented by a knowledge soundness adversary. To capture any possible adversarial strategy, we consider functions f and a distribution D from which random coins are sampled for a machine that implements f.

Definition 6 (Knowledge soundness strategy (KSS)). *Consider NIZK* $\mathit{\Pi} = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$. *Let* $f : \mathbf{CRS} \times \mathbf{\Gamma_s} \times \mathbf{\Gamma_t} \to \chi \times \mathit{\Psi}$ *be a function, and* D *be the uniform distribution over* $\mathbf{\Gamma_s} \times \mathbf{\Gamma_t}$. f *is said to be a knowledge soundness strategy for* $\mathit{\Pi}$ *if*

$$\Pr\left[\mathcal{V}(\texttt{crs}, \texttt{x}, \pi) = 1 \,\middle|\, \begin{array}{c} (\texttt{crs}, \texttt{td}) \leftarrow \mathsf{CRSGen}(1^k); (s||t) \leftarrow D \\ \\ f(\texttt{crs}; (s,t)) = (\texttt{x}, \pi) \end{array}\right] = \eta(k)$$

[7] Recently, [5] instantiated AGM under falsifiable assumptions. However, their construction relies on indistinguishability obfuscation. It is inherently inefficient and not a practical group for applications. Here, we focus on feasibility of knowledge soundness of the CH framework as is in the standard model, without compromising on the efficiency.

where $\eta(k)$ is non-negligible. We say that a TM $\mathcal{A}$ implements the knowledge soundness strategy f, if for any $\mathtt{crs} \in \mathbf{CRS}$ and $(s,t) \leftarrow D$, we have $z \leftarrow \mathcal{A}(\mathtt{crs}; s, t)$, where $z = f(\mathtt{crs}, s, t)$. If there exists a PPT $\mathcal{A}$ that implements a knowledge soundness strategy f, we say that f is efficiently implementable.

We now define semantic knowledge soundness for a KSS.

Definition 7 (Semantic knowledge soundness). *Consider a NIZK argument $\mathit{\Pi} = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$. Let D be the uniform distribution over $\mathbf{\Gamma_s} \times \mathbf{\Gamma_t}$. We call $\mathit{\Pi}$ semantic knowledge sound if for every efficiently implementable KSS f, there exists a PPT extractor $\mathsf{Ext} = \mathsf{Ext}_f$, such that, for each (even unbounded) TM $\mathcal{A}^*$ that implements f, we have*

$$\Pr\left[\begin{array}{c} \mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 \\ \wedge(\mathtt{x}, \mathtt{w}) \notin \mathcal{R} \end{array} \middle| \begin{array}{c} (\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k); (s||t) \leftarrow D \\ (\mathtt{x}, \pi) \leftarrow \mathcal{A}^*(\mathtt{crs}; (s,t)); \boxed{\mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi, t)} \end{array}\right] \leq \mathsf{negl}(k)$$

Remark 2. We note that asking for extraction only against provers that implement a KSS is *not* a weakening of the extraction definition, since we only care about extracting from provers that make the verifier accept with non-negligible probability.

Remark 3. Note that this definition is a generalization of the usual knowledge soundness definitions. In particular, if we hide all the randomness from the extractor (that is $\mathbf{\Gamma_t}$ is the set that contains only the empty string), then we recover the usual black-box knowledge soundness. On the other hand, if we give the extractor all the randomness used by the adversary (that is $\mathbf{\Gamma_s}$ is the set that contains only the empty string), then we recover the canonical white-box knowledge soundness. We discuss these connections formally in Appendix D.2. We define semBB and $\mathsf{semn\text{-}BB}$ exactly as in Definition 7 with the boxed part replaced with $\mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi)$, and $\mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi, s||t)$ respectively.

Remark 4 (Canonical knowledge soundness adversary). The usual definition of knowledge soundness naturally handles the existence of an extractor for the honest prover. Our definition handles the case of the honest prover too; we show the honest efficiently implementable KSS for a NIZK $\mathit{\Pi}$ below:

1. Sample uniformly random strings $(s,t) \leftarrow \mathbf{\Gamma_s} \times \mathbf{\Gamma_t}$.
2. Sample a true statement $\mathtt{x}$ together with $\mathtt{w}$, from the uniform distribution over pair of $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$, using random seed s. Note that this can be done efficiently. That is, there exists a PPT $\mathcal{A}$ that computes $(\mathtt{x}, \mathtt{w})$ on random coins s. Let us define the function $g : \mathbf{\Gamma_s} \to \chi \times \{0,1\}^*$ as $g(s) = (\mathtt{x}, \mathtt{w})$.
3. Run the honest prover algorithm on input $(\mathtt{crs}, \mathtt{x}, \mathtt{w})$ and random coins t, to compute a proof π. Define the function $g' : \mathbf{CRS} \times \chi \times \{0,1\}^* \times \mathbf{\Gamma_t} \to \mathit{\Psi}$ as $g'(\mathtt{crs}, \mathtt{x}, \mathtt{w}, t) = \mathcal{P}(\mathtt{crs}, \mathtt{x}, \mathtt{w}; t)$.
4. Define $f : \mathbf{CRS} \times \mathbf{\Gamma_s} \times \mathbf{\Gamma_t} \to \chi \times \mathit{\Psi}$ as $f(\mathtt{crs}, (s,t)) = (\mathtt{x}, \pi)$ where $(\mathtt{x}, \mathtt{w}) = g(s)$ and $\pi = g'(\mathtt{crs}, \mathtt{x}, \mathtt{w}, t)$.

We call this f the canonical knowledge soundness strategy, and a PPT algorithm that implements it the *canonical adversary* of knowledge soundness.

We illustrate the meaningfulness of the new notion by showing relationships of semantic extraction with BB and n-BB extraction definitions in Appendix D.2. Here we point out that the notion of semantic extraction has been implicitly used in other works. For instance, standard Σ-protocols satisfy the semantic extraction notion. By special soundness, given a certain number of accepting transcripts for the same statement, and the same prover's first message, an extractor exists that outputs a valid witness. The extractor, therefore, does not depend on the prover's computation, instead, on a "semantic" function: one that outputs two different accepting transcripts relative to the same statement, and the same first message. One advantage in thinking of an extractor as a semantic one is the possibility to use it in a reduction, without its relative "native" adversary. This is indeed what is done in the proof of soundness for the NIZK proof of [25] described in Fig. 3, which is based on the existence of an (unbounded) TM that computes a valid input for the special soundness extractor, and then relying on the implicit semantic property of the latter.

The non-black-box nature of the semantic definition is limited to making non-black-box use of the malicious prover's randomness, but otherwise the prover's TM is treated as a black-box. There are instances in literature where a n-BB technique in fact corresponds to a semantic technique. Consider the case of simulation – Barak's non-black-box zero-knowledge protocol [7]. Though simulation is defined to make non-black-box use of the verifier's TM, it can be modified to only make non-black-box use of the auxiliary input and running time of the verifier, and not its TM. The property needed to define the simulator is the existence of an efficient (with bounded-length description) adversary. Then in the security proof, the next-message function implemented by the adversary is used, together with the ability to choose its random coins. This means that the security proof works for any adversary (even an unbounded one) that computes the same next-message function. Moreover, the zero-knowledge simulator for each of these adversaries would be exactly the same simulator as the one defined for the efficient adversary. For concreteness, we may think that, given the code of one efficient adversary, we define a simulator that works for each TM that computes the same function, in the sense that we use the code in a black-box way; by just fixing the random coins and taking partial outputs.

5.2 Impossibility of Semantic Knowledge Soundness for CH-NIZK

In this section we focus on semantic knowledge soundness of NIZK argument in Fig. 2 for a large and useful class of algebraic languages. We show in Appendix D.1 that when the adversary is algebraic, knowledge soundness holds in the AGM for this NIZK argument. We ask for knowledge soundness in the standard model, and show that CH NIZK argument cannot be semantic knowledge sound. The impossibility can be interpreted as an adversary explicitly violating AGM rules by hiding some exponent about the statement, and thus making the extractor fail. We refer to Remark 5, for more remarks on the interpretation of this result, while we focus on technical details for the rest of this section.

We now show the impossibility proof of semantic knowledge soundness of CH arguments for linear languages $\mathcal{L}_{\texttt{lpar}}$, where $\texttt{lpar} = [\mathbf{M}]_1$ is a constant matrix. The proof of Theorem 2 for general case of 1DL-friendly languages is deferred to Appendix D.3.

Lemma 2. *Let $\mathcal{L}_{\texttt{lpar}}$ be a linear language defined by constant matrix $\texttt{lpar} := [\mathbf{M}]_1$. The NIZK argument in Fig. 2 cannot be semantic knowledge sound for $\mathcal{L}_{\texttt{lpar}}$ under the SDL assumption.*

Proof. We denote as w_i components of the vector $\mathbf{w}$. The description of the canonical prover adversary on input ($\texttt{crs} = [e]_2$) and random coins (s, t), where $t = (\mathbf{r}, r')$ is given in Fig. 6a. Let Ext_f be the semantic extractor for the function $f([e]_2; (s, t)) = ([\mathbf{x}]_1, \pi)$, with $\pi = ([\mathbf{a}]_1, [\mathbf{d}]_2)$ that is implemented by the canonical prover adversary. By completeness of the NIZK argument, $\mathsf{Ext}_f(e, [\mathbf{x}, \mathbf{a}]_1, [\mathbf{d}]_2, t)$ outputs a valid witness $\mathbf{w}$ for $[\mathbf{x}]_1$ with overwhelming probability. Let us consider the (not polynomial-time) TM $\mathcal{P}^*$ as in Fig. 6b that implements f. $\mathcal{P}^*$ implements the same f of the canonical adversary and therefore its output can be used to feed the same extractor Ext_f.

We now exploit Ext_f to define an adversary $\mathcal{A}$ against SDL assumption. On input an SDL challenge $([w_1]_1, [w_1]_2)$, $\mathcal{A}$ is defined as in Fig. 6c. Since $\mathcal{A}$ computes inputs of Ext_f exactly as $\mathcal{P}^*$ does, they are correctly distributed, and hence $\mathcal{A}$ breaks SDL with the same probability that Ext_f succeeds.

Theorem 2. *Let $\mathcal{L}_{\texttt{lpar}}$ be a 1DL-friendly algebraic language (Definition 13) defined by language parameters $\texttt{lpar} := (\mathbf{M}, \boldsymbol{\theta})$. The NIZK argument in Fig. 2 cannot be semantic knowledge sound for $\mathcal{L}_{\texttt{lpar}}$ under the SDL assumption.*

Remark 5. Since our reduction exploits the knowledge of the trapdoor to compute a proof, (as a typical ZK simulator would do), it might seem like we are arguing about extracting from the simulator. However this is not the case, at least in general. We note that the procedure defined by the SDL adversary is very different from the zero-knowledge simulator. First, the adversary knows something that the simulator does not, which is $[\mathbf{x}]_2$. Moreover, the adversary is able to compute $[\mathbf{a}]_1$ before computing $[\mathbf{d}]_2$ as the honest prover; while the simulator, in order to compute a proof must compute $\mathbf{d}$ before. This can be also seen as the fact that the honest prover and simulator do not implement the same function. In fact, given the language parameter $\mathbf{M} \in \mathbb{Z}_p^{n \times m}$ the prover computes a proof π as a function of $\mathbf{x}, \mathbf{w}, r$ where $r \in \mathbb{Z}_p^{n \times 1}$, while the simulator computes a proof which is a function of random coins $r_{\mathsf{Sim}} \in \mathbb{Z}_p^{m \times 1}$. In order to invoke the semantic extractor associated to the honest prover, we must have a function that defines a relation between the two randomness. This, for instance, can be done (inefficiently) only in some particular cases, like when $\mathbf{M}$ is a square invertible matrix. Finally, the existence of such cases is evidence towards the impossibility of extraction. In fact, given the latter case, since we have perfect zero-knowledge for a relation that defines only true statement, given a proof from the NIZK

1. Sample uniformly random w_1 using random coins s.
2. Sample other components of $\mathbf{w}$, w_i for $i \neq 1$, using random coins r'.
3. Compute $[\mathbf{x}]_1 = [\mathbf{M}]_1\mathbf{w}$.
4. Compute $[\mathbf{a}]_1 = [\mathbf{M}]_1\mathbf{r}$.
5. Compute $[\mathbf{d}]_2 = \mathbf{w}[e]_2 + \mathbf{r}[1]_2$.
6. Output $([\mathbf{x}]_1, \pi = ([\mathbf{a}]_1, [\mathbf{d}]_2))$.

(a) Canonical prover adversary

1. Use random coins s to sample w_1 and compute $[w_1]_1, [w_1]_2$.
2. Compute (inefficiently) e from $[e]_2$.
3. Sample other components of $\mathbf{w}$ using random coins r'.
4. Compute $[\mathbf{x}]_1 = [\mathbf{Mw}]_1$.
5. Compute $[\mathbf{a}]_1 = [\mathbf{Mr}]_1$.
6. Compute $[\mathbf{d}]_2 = e[\mathbf{w}]_2 + \mathbf{r}[1]_2$.

(b) Unbounded adversary

1. Sample $e, \mathbf{r}, r'$.
2. Sample other components of $\mathbf{w}$, w_i for $i \neq 1$, using random coins r'.
3. Compute $[\mathbf{x}]_1 = [\mathbf{Mw}]_1$.
4. Compute $[\mathbf{a}]_1 = [\mathbf{Mr}]_1$.
5. Compute $[\mathbf{d}]_2 = e[\mathbf{w}]_2 + \mathbf{r}[1]_2$.
6. Compute $\mathbf{w} \leftarrow \mathsf{Ext}_f(([\mathbf{a}]_1, [\mathbf{d}]_2), e, [\mathbf{x}]_1, (\mathbf{r}, r'))$.
7. Output w_1.

(c) SDL adversary

Fig. 6: Procedures for Lemma 2

argument, it is impossible to distinguish the case when the prover was honest, from the case when a powerful adversary just computes the discrete logarithm of the CRS and runs the simulator. Furthermore, it is impossible to distinguish the case that adversary had $[\mathbf{w}]_2$ and the trapdoor e, instead of $\mathbf{w}$ without relying on knowledge-type assumptions.

References

1. Abdalla, M., Benhamouda, F., Pointcheval, D.: Disjunctions for hash proof systems: New constructions and applications. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 69–100. Springer, Heidelberg (Apr 2015)
2. Abdolmaleki, B., Baghery, K., Lipmaa, H., Zajac, M.: A subversion-resistant SNARK. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part III. LNCS, vol. 10626, pp. 3–33. Springer, Heidelberg (Dec 2017)
3. Abdolmaleki, B., Lipmaa, H., Siim, J., Zajac, M.: On QA-NIZK in the BPK model. In: Kiayias, A., Kohlweiss, M., Wallden, P., Zikas, V. (eds.) PKC 2020, Part I. LNCS, vol. 12110, pp. 590–620. Springer, Heidelberg (May 2020)
4. Acar, T., Nguyen, L.: Revocation for delegatable anonymous credentials. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 423–440. Springer, Heidelberg (Mar 2011)

5. Agrikola, T., Hofheinz, D., Kastner, J.: On instantiating the algebraic group model from falsifiable assumptions. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part II. LNCS, vol. 12106, pp. 96–126. Springer, Heidelberg (May 2020)
6. Ananth, P., Asharov, G., Dahari, H., Goyal, V.: Towards accountability in crs generation. IACR Eurocrypt 2021, `https://eprint.iacr.org/2021/1090.pdf`
7. Barak, B.: How to go beyond the black-box simulation barrier. In: 42nd FOCS. pp. 106–115. IEEE Computer Society Press (Oct 2001)
8. Barak, B., Ong, S.J., Vadhan, S.P.: Derandomization in cryptography. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 299–315. Springer, Heidelberg, Santa Barbara, USA (Aug 17–21, 2003)
9. Bauer, B., Fuchsbauer, G., Loss, J.: A classification of computational assumptions in the algebraic group model. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part II. LNCS, vol. 12171, pp. 121–151. Springer, Heidelberg (Aug 2020)
10. Belenkiy, M., Chase, M., Kohlweiss, M., Lysyanskaya, A.: P-signatures and noninteractive anonymous credentials. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 356–374. Springer, Heidelberg (Mar 2008)
11. Ben Hamouda-Guichoux, F.: Diverse Modules and Zero-Knowledge. Ph.D. thesis, PSL Research University (2016)
12. Ben-Or, M., Goldreich, O., Goldwasser, S., Håstad, J., Kilian, J., Micali, S., Rogaway, P.: Everything provable is provable in zero-knowledge. In: Goldwasser, S. (ed.) CRYPTO'88. LNCS, vol. 403, pp. 37–56. Springer, Heidelberg (Aug 1990)
13. Benhamouda, F., Blazy, O., Chevalier, C., Pointcheval, D., Vergnaud, D.: New techniques for SPHFs and efficient one-round PAKE protocols. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 449–475. Springer, Heidelberg (Aug 2013)
14. Benhamouda, F., Pointcheval, D.: Trapdoor Smooth Projective Hash Functions. Tech. Rep. 2013/341, IACR (Jun 3 2013), available at `http://eprint.iacr.org/2013/341`, last retrieved version from 27 Aug 2013
15. Bitansky, N.: Verifiable random functions from non-interactive witness-indistinguishable proofs. Cryptology ePrint Archive, Report 2017/018 (2017), `http://eprint.iacr.org/2017/018`
16. Blum, M., Feldman, P., Micali, S.: Non-Interactive Zero-Knowledge and Its Applications. In: STOC 1988. pp. 103–112. ACM Press, Chicago, Illinois, USA (May 2–4, 1988)
17. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (May 2003)
18. Bootle, J., Cerulli, A., Chaidos, P., Ghadafi, E., Groth, J., Petit, C.: Short accountable ring signatures based on DDH. Cryptology ePrint Archive, Report 2015/643 (2015), `http://eprint.iacr.org/2015/643`
19. Boyen, X.: The uber-assumption family (invited talk). In: Galbraith, S.D., Paterson, K.G. (eds.) PAIRING 2008. LNCS, vol. 5209, pp. 39–56. Springer, Heidelberg (Sep 2008)
20. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer, Heidelberg (May 2001)
21. Camenisch, J., Stadler, M.: Efficient group signature schemes for large groups (extended abstract). In: Kaliski Jr., B.S. (ed.) CRYPTO'97. LNCS, vol. 1294, pp. 410–424. Springer, Heidelberg (Aug 1997)

22. Campanelli, M., Fiore, D., Querol, A.: LegoSNARK: Modular design and composition of succinct zero-knowledge proofs. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019. pp. 2075–2092. ACM Press (Nov 2019)
23. Chiesa, A., Hu, Y., Maller, M., Mishra, P., Vesely, N., Ward, N.P.: Marlin: Preprocessing zkSNARKs with universal and updatable SRS. In: Canteaut, A., Ishai, Y. (eds.) EUROCRYPT 2020, Part I. LNCS, vol. 12105, pp. 738–768. Springer, Heidelberg (May 2020)
24. Chung, K.M., Lui, E., Pass, R.: From weak to strong zero-knowledge and applications. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part I. LNCS, vol. 9014, pp. 66–92. Springer, Heidelberg (Mar 2015)
25. Couteau, G., Hartmann, D.: Shorter non-interactive zero-knowledge arguments and ZAPs for algebraic languages. In: Micciancio, D., Ristenpart, T. (eds.) CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 768–798. Springer, Heidelberg (Aug 2020)
26. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: Desmedt, Y. (ed.) CRYPTO'94. LNCS, vol. 839, pp. 174–187. Springer, Heidelberg (Aug 1994)
27. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (Apr / May 2002)
28. Dwork, C., Naor, M.: Zaps and their applications. In: 41st FOCS. pp. 283–293. IEEE Computer Society Press (Nov 2000)
29. Escala, A., Groth, J.: Fine-tuning groth-sahai proofs. Cryptology ePrint Archive, Report 2013/662 (2013), `http://eprint.iacr.org/2013/662`
30. Faonio, A., Fiore, D., Herranz, J., Ràfols, C.: Structure-preserving and rerandomizable RCCA-secure public key encryption and its applications. In: Galbraith, S.D., Moriai, S. (eds.) ASIACRYPT 2019, Part III. LNCS, vol. 11923, pp. 159–190. Springer, Heidelberg (Dec 2019)
31. Feige, U., Fiat, A., Shamir, A.: Zero knowledge proofs of identity. In: Aho, A. (ed.) 19th ACM STOC. pp. 210–217. ACM Press (May 1987)
32. Fortnow, L.: The complexity of perfect zero-knowledge (extended abstract). In: Aho, A. (ed.) 19th ACM STOC. pp. 204–209. ACM Press (May 1987)
33. Freund, Y., Schapire, R.E.: Adaptive game playing using multiplicative weights. Games and Economic Behavior 29(1-2), 79–103 (1999)
34. Fuchsbauer, G., Kiltz, E., Loss, J.: The algebraic group model and its applications. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 33–62. Springer, Heidelberg (Aug 2018)
35. Gabizon, A., Williamson, Z.J., Ciobotaru, O.: PLONK: Permutations over lagrange-bases for oecumenical noninteractive arguments of knowledge. Cryptology ePrint Archive, Report 2019/953 (2019), `https://eprint.iacr.org/2019/953`
36. Garg, S., Ostrovsky, R., Visconti, I., Wadia, A.: Resettable statistical zero knowledge. In: Cramer, R. (ed.) TCC 2012. LNCS, vol. 7194, pp. 494–511. Springer, Heidelberg (Mar 2012)
37. Gentry, C., Wichs, D.: Separating succinct non-interactive arguments from all falsifiable assumptions. In: Fortnow, L., Vadhan, S.P. (eds.) 43rd ACM STOC. pp. 99–108. ACM Press (Jun 2011)
38. Goldreich, O., Micali, S., Wigderson, A.: Proofs that yield nothing but their validity and a methodology of cryptographic protocol design (extended abstract). In: 27th FOCS. pp. 174–187. IEEE Computer Society Press (Oct 1986)

39. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game or A completeness theorem for protocols with honest majority. In: Aho, A. (ed.) 19th ACM STOC. pp. 218–229. ACM Press (May 1987)
40. Goldwasser, S., Micali, S., Rackoff, C.: The Knowledge Complexity of Interactive Proof-Systems. In: Sedgewick, R. (ed.) STOC 1985. pp. 291–304. ACM Press, Providence, Rhode Island, USA (May 6–8, 1985)
41. Green, M., Hohenberger, S.: Practical adaptive oblivious transfer from simple assumptions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 347–363. Springer, Heidelberg (Mar 2011)
42. Groth, J., Ostrovsky, R., Sahai, A.: New Techniques for Noninteractive Zero-Knowledge. Journal of the ACM 59(3) (2012)
43. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (Apr 2008)
44. Izabachène, M., Libert, B., Vergnaud, D.: Block-wise P-signatures and non-interactive anonymous credentials with efficient attributes. In: Chen, L. (ed.) 13th IMA International Conference on Cryptography and Coding. LNCS, vol. 7089, pp. 431–450. Springer, Heidelberg (Dec 2011)
45. Jutla, C.S., Roy, A.: Shorter quasi-adaptive NIZK proofs for linear subspaces. In: Sako, K., Sarkar, P. (eds.) ASIACRYPT 2013, Part I. LNCS, vol. 8269, pp. 1–20. Springer, Heidelberg (Dec 2013)
46. Kiltz, E., Wee, H.: Quasi-adaptive NIZK for linear subspaces revisited. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 101–128. Springer, Heidelberg (Apr 2015)
47. Maller, M., Bowe, S., Kohlweiss, M., Meiklejohn, S.: Sonic: Zero-knowledge SNARKs from linear-size universal and updatable structured reference strings. In: Cavallaro, L., Kinder, J., Wang, X., Katz, J. (eds.) ACM CCS 2019. pp. 2111–2128. ACM Press (Nov 2019)
48. Meiklejohn, S.: An extension of the groth-sahai proof system (2009)
49. Naor, M.: On cryptographic assumptions and challenges (invited talk). In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 96–109. Springer, Heidelberg (Aug 2003)
50. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: 22nd ACM STOC. pp. 427–437. ACM Press (May 1990)
51. Ng, T., Tan, S., Chin, J.: A variant of BLS signature scheme with tight security reduction. In: Mobile Networks and Management - 9th International Conference, MONAMI 2017, Melbourne, Australia, December 13-15, 2017, Proceedings. pp. 150–163 (2017), https://doi.org/10.1007/978-3-319-90775-8_13
52. Pass, R.: Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In: Sahai, A. (ed.) TCC 2013. LNCS, vol. 7785, pp. 334–354. Springer, Heidelberg (Mar 2013)
53. Ràfols, C.: Stretching groth-sahai: NIZK proofs of partial satisfiability. In: Dodis, Y., Nielsen, J.B. (eds.) TCC 2015, Part II. LNCS, vol. 9015, pp. 247–276. Springer, Heidelberg (Mar 2015)
54. Rial, A., Kohlweiss, M., Preneel, B.: Universally composable adaptive priced oblivious transfer. In: Shacham, H., Waters, B. (eds.) PAIRING 2009. LNCS, vol. 5671, pp. 231–247. Springer, Heidelberg (Aug 2009)
55. Rotem, L., Segev, G.: Algebraic distinguishers: From discrete logarithms to decisional uber assumptions. In: Pass, R., Pietrzak, K. (eds.) TCC 2020, Part III. LNCS, vol. 12552, pp. 366–389. Springer, Heidelberg (Nov 2020)

56. Shoup, V.: Lower bounds for discrete logarithms and related problems. In: Fumy, W. (ed.) EUROCRYPT'97. LNCS, vol. 1233, pp. 256–266. Springer, Heidelberg (May 1997)

Appendix

A Additional Preliminaries

A.1 Σ-Protocols

A Σ-protocol is a public-coin three-round interactive protocol between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$. A Σ-protocol should satisfy completeness, special soundness, and special honest verifier zero-knowledge (SHVZK), defined as follows:

Definition 8 (Completeness). *A Σ-protocol is complete for relation $\mathcal{R}$, if for any PPT adversary $\mathcal{A}$, and any honest $\mathcal{P}$ and $\mathcal{V}$,*

$$\Pr\left[\langle\mathcal{P}(\mathtt{w}),\mathcal{V}\rangle(\mathtt{x})=1\vee\ (\mathtt{x},\mathtt{w})\notin\mathcal{R}\middle|(\mathtt{x},\mathtt{w})\leftarrow\mathcal{A}(1^k)\right]=1$$

Definition 9 (Special Soundness). *A Σ-protocol for a relation $\mathcal{R}$ is special sound, if there exists a PPT algorithm* Ext *that given a statement* $\mathtt{x}$ *and two accepting transcripts $(a,e,d),(a,e',d')$ with the same first message and $e\neq e'$ outputs a witness* $\mathtt{w}$*, such that $(\mathtt{x},\mathtt{w})\in\mathcal{R}$ with overwhelming probability.*

Definition 10 (Special Honest-Verifier Zero-Knowledge (SHVZK)). *A Σ-protocol for a relation $\mathcal{R}$ is SHVZK, if there exists a PPT simulator* Sim *such that for $(\mathtt{x},\mathtt{w})\in\mathcal{R}$ and $e\in\{0,1\}^k$, the distributions of* $\mathsf{Sim}(\mathtt{x},e)$ *is identical to the distribution of the 3-move honest transcript obtained when $\mathcal{V}$ sends e as challenge and $\mathcal{P}$ runs on common input* $\mathtt{x}$ *and private input* $\mathtt{w}$ *such that $(\mathtt{x},\mathtt{w})\in\mathcal{R}$.*

For the sake of completeness, we also recall the definition of witness indistinguishability for Σ-protocols. As shown in [26], every Σ-protocol that enjoys Completeness, Special Soundness and perfect Honest Verifier Zero Knowledge (HVZK) is perfect WI.

Definition 11 (Witness Indistinguishability (WI)). *A Σ-protocol for a relation $\mathcal{R}$ is perfect WI* [8] *if for every malicious verifier $\mathcal{V}^*$, for all $\mathsf{st}=(\mathtt{x},\mathtt{w}_1,\mathtt{w}_2)$ such that $(\mathtt{x},\mathtt{w}_1)\in\mathcal{R},(\mathtt{x},\mathtt{w}_2)\in\mathcal{R}$, we have*

$$\Pr\left[\langle\mathcal{P}(\mathtt{w}_1,1^k),\mathcal{V}^*(\mathsf{st})\rangle(\mathtt{x})=1\right]=\Pr\left[\langle\mathcal{P}(\mathtt{w}_2,1^k),\mathcal{V}^*(\mathsf{st})\rangle(\mathtt{x})=1\right]$$

[8] WI is used to mean both “witness indistinguishability” and “witness indistinguishable”.

A.2 Witness Sampleable (WS) Languages

For a witness sampleable language $\mathcal{L}$, the language parameters come together with a trapdoor which allows to check whether $\mathtt{x} \in \mathcal{L}$. In this case, we suppose that $\mathsf{setup}.\mathtt{lpar}$ also outputs a (language) trapdoor $\mathtt{ltrap}$ associated with $\mathtt{lpar}$ and allows to decide whether a given $\mathtt{x} \in \mathcal{X}$ is in $\mathcal{L}$ or not. It is easy to see that for linear languages, this trapdoor is the exponents of all matrix entries. We refer to [25] for formal definition and more details of WS languages.

A.3 Smooth Projective Hash Function (SPHF)

A SPHF is defined as follows (cf. [13]).

Definition 12. *A SPHF for* $\{\mathcal{L}_{\mathtt{lpar}}\}$ *is a tuple of PPT algorithms* $(\mathsf{setup}, \mathsf{hashkg}, \mathsf{projkg}, \mathsf{hash}, \mathsf{projhash})$*, which are defined as follows:*

$\mathsf{setup}(1^k)$**:** *Takes a security parameter k and generates the global parameters* pp *together with the language parameters* $\mathtt{lpar}$ *(we assume that all algorithms have access to* pp*).*

$\mathsf{hashkg}(\mathtt{lpar})$**:** *Takes a language parameter* $\mathtt{lpar}$ *and outputs a hashing key* hk*.*

$\mathsf{projkg}(\mathtt{lpar}; \mathsf{hk}, \mathtt{x})$**:** *Takes a hashing key* hk*,* $\mathtt{lpar}$*, and a statement* $\mathtt{x}$ *and outputs a projection key* hp*, possibly depending on* $\mathtt{x}$*.*

$\mathsf{hash}(\mathtt{lpar}; \mathsf{hk}, \mathtt{x})$**:** *Takes a hashing key* hk*,* $\mathtt{lpar}$*, and a statement* $\mathtt{x}$ *and outputs a hash value* H*.*

$\mathsf{projhash}(\mathtt{lpar}; \mathsf{hp}, \mathtt{x}, \mathtt{w})$**:** *Takes a projection key* hp*,* $\mathtt{lpar}$*, a statement* $\mathtt{x}$*, and a witness* $\mathtt{w}$ *for* $\mathtt{x} \in \mathcal{L}$ *and outputs a hash value* pH*.*

A SPHF needs to satisfy the following properties:

Correctness. It is required that $\mathsf{hash}(\mathtt{lpar}; \mathsf{hk}, \mathtt{x}) = \mathsf{projhash}(\mathtt{lpar}; \mathsf{hp}, \mathtt{x}, \mathtt{w})$ for all $\mathtt{x} \in \mathcal{L}$ and their corresponding witnesses $\mathtt{w}$.

Smoothness. It is required that for any $\mathtt{lpar}$ and any $\mathtt{x} \notin \mathcal{L}$, the following distributions are statistically indistinguishable:

$$\{(\mathsf{hp}, \mathsf{H}) : \mathsf{hk} \leftarrow \mathsf{hashkg}(\mathtt{lpar}), \mathsf{hp} \leftarrow \mathsf{projkg}(\mathtt{lpar}; \mathsf{hk}, \mathtt{x}), \mathsf{H} \leftarrow \mathsf{hash}(\mathtt{lpar}; \mathsf{hk}, \mathtt{x})\}$$
$$\{(\mathsf{hp}, \mathsf{H}) : \mathsf{hk} \leftarrow \mathsf{hashkg}(\mathtt{lpar}), \mathsf{hp} \leftarrow \mathsf{projkg}(\mathtt{lpar}; \mathsf{hk}, \mathtt{x}), \mathsf{H} \leftarrow \Omega\} \quad .$$

where Ω is the set of hash values.

A.4 Construction of SPHF from Diverse Vector Space

A diverse vector space (DVS) [13,1,11] is a representation of a language $\mathcal{L} \subseteq \mathcal{X}$ as a subspace $\hat{\mathcal{L}}$ of some vector space. Let $\mathcal{R} = \{(\mathtt{x}, \mathtt{w})\}$ be a relation with $\mathcal{L} = \{\mathtt{x} : \exists \mathtt{w}, (\mathtt{x}, \mathtt{w}) \in \mathcal{R}\}$. Let pp be system parameters, including say the description of a bilinear group. A (pairing-based) DVS $\mathcal{V}$ is defined as $\mathcal{V} = (\mathsf{pp}, \mathcal{X}, \mathcal{L}, \mathcal{R}, n, k, \mathbf{M}, \boldsymbol{\theta}, \boldsymbol{\lambda})$, where $\mathbf{M}(\mathtt{x})$ is an $n \times k$ matrix, $\boldsymbol{\theta}(\mathtt{x})$ is an n-dimensional vector, and $\boldsymbol{\lambda}(\mathtt{x}, \mathtt{w})$ a k-dimensional vector. The matrix $\mathbf{M}(\mathtt{x})$ can depend on $\mathbf{x}$ (in this case, it is called GL-DVS) or not (KV-DVS). Moreover, different coefficients of $\boldsymbol{\theta}(\mathtt{x})$, $\mathbf{M}(\mathtt{x})$, and $\boldsymbol{\lambda}(\mathtt{x}, \mathtt{w})$ can belong to different algebraic structures as long as the equation $\boldsymbol{\theta}(\mathtt{x}) = \mathbf{M}(\mathtt{x}) \cdot \boldsymbol{\lambda}(\mathtt{x}, \mathtt{w})$ is well-consistent. In the most common case, this means that given a bilinear group $\mathsf{pp} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, g_1, g_2)$, they belong to either $\mathbb{Z}_p$, $\mathbb{G}_1$, $\mathbb{G}_2$, or $\mathbb{G}_T$ as long as the consistency of the above equation is preserved.

A DVS $\mathcal{V}$ satisfies the following properties [11]:

- *coordinate-independence of groups:* the group in which each coordinate of $\boldsymbol{\theta}(\mathtt{x})$ lies is independent of $\mathtt{x}$.
- *perfect completeness:* for any $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$, $\boldsymbol{\theta}(\mathtt{x}) = \mathbf{M}(\mathtt{x}) \cdot \boldsymbol{\lambda}(\mathtt{x}, \mathtt{w})$.
- *statistical ε-soundness:* $\forall \mathtt{x} \in \mathcal{X} \setminus \mathcal{L}$, $\Pr[\boldsymbol{\theta}(\mathtt{x}) \in \mathrm{colspace}(\mathbf{M}(\mathtt{x}))] \leq \varepsilon$.

In this work, we only deal with DVSs where $\boldsymbol{\lambda}$ is the identity function. I.e., $\boldsymbol{\lambda}(\mathtt{x}, \mathtt{w}) = \mathtt{w}$. Given a GL/KV-DVS for $\mathcal{L}$, one can construct an efficient GL/KV-SPHF for $\mathtt{x}' \in \mathcal{L}$, where $\mathtt{w} = \mathtt{w}'$ and $\mathtt{x} = [\boldsymbol{\theta}(\mathtt{x}')]_\iota = [\mathbf{M}(\mathtt{x}')]_\iota \mathtt{w}'$ [13], see Fig. 7. Here, the only possible nonlinear operation is the dependency of $\boldsymbol{\theta}$ and $\mathbf{M}$ on the actual input $\mathtt{x}'$. It is known that if $\mathcal{V}$ is a 0-sound GL-DVS/KV-DVS, then the PHF in Fig. 7 is a perfectly smooth GL/KV-SPHF, see Theorem 3.1.11 in [11].

- $\mathsf{hashkg}(\mathtt{lpar})$: sample $\vec{\alpha} \leftarrow \mathbb{Z}_p^n$, and output $\mathsf{hk} \leftarrow \vec{\alpha}$;
- $\mathsf{projkg}(\mathtt{lpar}; \mathsf{hk}, \mathtt{x} = [\boldsymbol{\theta}(\mathtt{x}')]_\iota)$: $[\boldsymbol{\gamma}]_\iota^\top \leftarrow \vec{\alpha}^\top [\mathbf{M}(\mathtt{x}')]_\iota \in \mathbb{G}_\iota^{1 \times k}$; return $\mathsf{hp} \leftarrow [\boldsymbol{\gamma}]_\iota$;
- $\mathsf{hash}(\mathtt{lpar}; \mathsf{hk}, \mathtt{x})$: return $\mathsf{H} \leftarrow \vec{\alpha}^\top [\boldsymbol{\theta}(\mathtt{x}')]_\iota$;
- $\mathsf{projhash}(\mathtt{lpar}; \mathsf{hp}, \mathtt{x}, \mathtt{w} = \mathtt{w}')$: return $\mathsf{pH} \leftarrow [\boldsymbol{\gamma}]_\iota^\top \mathtt{w}'$;

Fig. 7: DVS-based SPHF construction for $\mathcal{L}_{\mathtt{lpar}}$ with $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$.

We sometimes require algebraic languages to satisfy a property we call 1DL-friendly. The reason we need this property is to enable the embedding of a symmetric simple discrete logarithm challenge, which is given as a pair of group elements, into an algebraic statement in the reduction. We give the definition below.

Definition 13. *An algebraic languages is* 1*DL-friendly if given a uniformly random element, $c \leftarrow \mathbb{Z}_p$, there exist a tuple of functions λ_x, λ_w such that the following procedure can be used to generate a pair of true statement $[\mathbf{x}]_1$ with a relative witness $\mathbf{w}$.*

- *Define $w_1 = c$ and sample uniformly random $w_2, \ldots, w_d$ (independently from w_1). Compute $\mathbf{w} = \lambda_w(w_1, \ldots, w_n)$. We restrict here to functions λ_w that are affine in w_1.*

- *Compute* $[\mathbf{x}]_1 = \lambda_x(w_1, \ldots, w_n)$*, such that* $\mathbf{M}(\mathbf{x})\mathbf{w} = \boldsymbol{\theta}(\mathbf{x})$*. Again, we restrict to functions* λ_x *that are affine in* w_1*.*

Practically, given group elements $[x]_1, [x]_2$, we implicitly define $x = w_1$ and sample $w_2, \ldots, w_n \leftarrow \mathbb{Z}_p$. Then, we compute $[\mathbf{x}]_1 = \lambda_x([x]_1, w_2, \ldots, w_n)$ and a $\mathbb{G}_2$-encoding of the relative witness $[\mathbf{w}]_2 = \lambda_w([x]_2, w_2, \ldots, w_n)$, which are efficiently computable because λ_x, λ_w are required to be affine as a function of w_1. We remark that this condition is only assumed for simplicity, in order to state our formal theorem under simple assumptions. However, our framework could in principle, work for any hard algebraic language, at the cost of using more structured assumptions.

A.5 From Σ-protocols to NIZKs (Extended)

The work of [25] proposed a framework for compiling a Σ-protocol for algebraic languages into a non-interactive zero-knowledge argument by embedding the challenge in $\mathbb{G}_2$ and publishing it once in the `crs`. The soundness of the compiled NIZK is based on a new family of assumptions *extended-kernel Matrix Diffie-Hellman* (**extKerMDH**) that are not necessarily falsifiable [9].

Couteau and Hartmann [25] also showed how to achieve perfect soundness by making use of the unconditional special soundness of the Σ-protocol. More precisely, they proved that the compiled protocol in Fig. 3 is a NIZK proof with computational zero-knowledge if the DDH assumption holds in $\mathbb{G}_2$, and the underlying Σ-protocol is complete, special sound and SHVZK.

We remark that there is no efficient extractor to compute the witness in the latter proof system. In fact the existence of a witness is guaranteed by the special soundness of the underlying Σ-protocol, however, to be able to extract it, we need an unbounded extractor to compute the exponent of group elements. To be more precise, an efficient extractor can compute, in the best case, only exponentiations of the witness in either $\mathbb{G}_1$ or $\mathbb{G}_2$ as shown in Section 4. It is worth mentioning that the soundness proof is based on the existence of this unbounded extractor, to compute a pair of proofs of the underlying Σ-protocol. More precisely, given a valid proof for a false statement and under an honestly generated CRS, we can (inefficiently) compute the field elements $(s_1, e_1, s_2, e_2, d_1, d_2)$ and output two valid proofs for the underlying Σ-protocol with the same first message and different challenges (with overwhelming probability). This contradicts the special soundness property, which states that two such proofs cannot exist for a false statement.

A.6 Algebraic Group Model

Algebraic algorithms. We recall that AGM essentially states that for every efficient algorithm $\mathcal{A}$ that outputs the vector $[\mathbf{y}]_\iota$ of group elements in $\mathbb{G}_\iota$ when given inputs the vector $[\mathbf{x}]_\iota$ of group elements in $\mathbb{G}_\iota$, there exists an efficient

[9] Although the assumption is falsifiable for all witness-sampleable languages (A.2).

extractor $\mathsf{Ext}_{\mathcal{A}}$ that returns a matrix $\mathbf{A}$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}$. In particular, since we are working in the setting of asymmetric bilinear pairings, we require that any outputs in one group must depend only on the inputs it receives in that group.

Algebraic Distinguishers. Here, we briefly recall the notion of algebraic distinguishers and refer the reader to [55] for more details.

A distinguisher is an algorithm that aims to distinguish between 2 games. Particularly, we consider adversaries $\mathcal{A}$ that engage in games with challengers, parametrized by a bit $b \in \{0, 1\}$. We refer to G_b as the game where the bit b is chosen. A distinguisher $\mathcal{A}$ aims to detect if it is playing the game G_0, or G_1. At the end of its interaction with the challenger, it outputs a decisional bit b'. $\mathcal{A}$ wins the game if $b = b'$. Let us denote by $\mathbf{View}_{\mathcal{A}}^{G_b}$ the random variable that describes the view of $\mathcal{A}$ in the game G_b (that is the input it received so far and the internal random tape). Moreover, let $[x_0, \ldots, x_{n_1}]_1, [y_0, \ldots, y_{n_2}]_2$ be $\mathcal{A}$'s input, with $x_0 = y_0 = 1$, and let $\vec{w}$ be a vector indexed by two indices $i \in \{0, \ldots, n_1\}, j \in \{0, \ldots, n_2\}$ such that the component w_{ij} is naturally associated to the pairing of inputs $[x_i]_1$ and $[y_j]_1$, i.e., $[x_i]_1[y_j]_2 = [x_i y_j]_T$. We indicate with $\left[\mathbf{View}_{\mathcal{A}}^{G_b}\right]_{supp(\vec{w})}$ the random variable that is defined by the view $\mathcal{A}$ in the game G_b omitting all group elements whose corresponding entry in $\vec{w}$ is 0. A distinguisher $\mathcal{A}$ participating in an algebraic game G_b, is said to be algebraic if there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$ that computes a vector $\vec{w}$ that explains the decision in an algebraic way, at least with a certain probability.

Definition 14 (Algebraic distinguisher). *A distinguisher $\mathcal{A}$ participating in an algebraic game G_b is said to be algebraic if there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$ that computes a vector of field element $\vec{w}$ such that the following condition holds.*

1. $\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} w_{ij}[x_i y_j]_T = [0]_T$.
2. *Let t be an upper bound over the running time of $\mathcal{A}$ and ϵ be the probability that $\mathcal{A}$ successfully distinguishes between G_0 and G_1. Then*

$$\Pr\left[\left[\mathbf{View}_{\mathcal{A}}^{G_0}\right]_{supp(\vec{w})} \neq \left[\mathbf{View}_{\mathcal{A}}^{G_1}\right]_{supp(\vec{w})}\right] \geq \epsilon/t^2,$$

   where the inequality is intended as distributions and the probability is over the choice of $\vec{w}$ induced by a random execution of $G_b(\mathcal{A})$ and $\mathsf{Ext}_{\mathcal{A}}$.

B NIWI Proof in the Plain Model

B.1 Efficiency of Our NIWI Proof

We give an informal comparison between the Groth-Ostrovsky-Sahai (GOS) NIWI [42] in the plain model and our NIWI in Fig. 4.

Recall that GOS techniques [42] to construct NIWI in the plain model consist of sending two distinct Groth-Sahai proofs, along with two different

$\texttt{crs}$-s chosen by the prover. This results in communication complexity that is two times the size of proof plus the $\texttt{crs}$ [10]. With our technique, a NIWI proof has communication complexity of one CH proof, on top of 6 group elements that are sent as the "crs" that the prover chooses. As noted in [25], proofs in the CH framework has the same size as optimized Groth-Sahai proofs, for many languages of interest, such as disjunctions of linear languages. Hence our NIWI proof, in these cases, has better communication complexity compared to Groth-Sahai NIWI [42].

When proving statements where the statement is augmented with intermediate commitments, our resulting statement size is much shorter, since we only need to commit in $\mathbb{G}_1$, while usually one needs to commit in both groups with GS. This results in better communication complexity in scenarios where we embed a circuit satisfiability problem as an algebraic language and commitments are part of the statement that are sent along with the proof.

Finally, we point out that our NIWI construction is the first that achieves constant overhead for communication and computational complexity (with respect to the language size), compared to the corresponding NIZK proof in the CRS model.

B.2 Applications

There are several works [15,36,6] which show how one can make use of NIWI in the plain model to construct more complex cryptographic primitives. Bitansky et al. [15] showed how to construct verifiable random functions and verifiable function commitment schemes using NIWI in the plain model. In [36], Garg et al. introduced the notion of *Efficiently Extractable Non-Interactive Instance-Dependent Commitment Scheme* and constructed a two-round resettable statistical witness-indistinguishable argument for languages that have such type of commitments. The key idea in their construction is to make use of a NIWI proof system in the plain model to ensure that verifier's challenge in the first round of the argument is well-formed. The fact that the verifier's challenge is a commitment to a random message indicates that the NIWI language is "natively" algebraic, and hence our NIWI can be used to improve the efficiency of the resulting argument in [36], wherein the NIWI is instantiated with [42].

The recent work of Ananth et al. [6] which provides a notion of accountability towards the CRS generation authority employs a NIWI proof system. To this end, the authority is required to include some valid transcript in the CRS and since he is the one who generates the CRS, the idea of using a NIZK proof does not work. The authority instead proves a statement about the transcript using a NIWI proof. In more detail, the authority provides four commitments $(\mathsf{cm} = (\mathsf{cm}_0, \mathsf{cm}_1), \overline{\mathsf{cm}} = (\overline{\mathsf{cm}}_0, \overline{\mathsf{cm}}_1))$ and uses a NIWI in the plain model to prove

[10] The proof size can naturally be improved by using optimized variants of Groth-Sahai proofs like [29,53]

that one of cm or $\overline{\mathsf{cm}}$ are commitments to both bits 0 and 1. Interestingly, the NIWI language corresponding to the statements defined by the commitments is again natively algebraic, for which our NIWI is suitable.

B.3 New Computational Assumption and AGM Proof of Security

Assumption 5 (Symmetric power discrete logarithm (SPDL)) *Let q_1, q_2 be two integers. The (q_1, q_2)-**SPDL** assumption holds if for any PPT adversary $\mathcal{A}$,*

$$\Pr\left[\, y^* = y \middle|\; y^* \leftarrow \mathcal{A}([1, y, y^2, \ldots, y^{q_1}]_1, [1, y, y^2, \ldots, y^{q_2}]_2) \,\right] \leq \mathsf{negl}(k)$$

where y is sampled from the uniform distribution over $\mathbb{Z}_p$.

Following the framework of [55], we prove the security of our new assumption in the AGM. We start by restating the new assumption.

Assumption 6 (Algebraic decisional hidden range) *Let $G_{\boldsymbol{ADHR},i}$, for $i \in \{0, 1\}$ be the games depicted in Fig. 5. Let $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$ be any pair of language parameter that defines the algebraic language $\mathcal{L}_{\mathtt{lpar}}$. The $(\mathbf{M}, \boldsymbol{\theta})$-**ADHR** assumption states that for any PPT adversary $\mathcal{A}$,*

$$\mathbf{Adv}_{\mathcal{A},\mathtt{lpar}}^{G_{\boldsymbol{ADHR},0,1}} = |\Pr[G_{\boldsymbol{ADHR},0}(\mathcal{A}, \mathtt{lpar}) = 1] - \Pr[G_{\boldsymbol{ADHR},1}(\mathcal{A}, \mathtt{lpar}) = 1]| \leq \mathsf{negl}(k).$$

Theorem 3. *If the $(1, 2)$-**SPDL** holds, then for any PPT algebraic distinguisher $\mathcal{A}$, it holds that*

$$\mathbf{Adv}_{\mathcal{A},\mathtt{lpar}}^{G_{\boldsymbol{ADHR},0,1}} \leq \mathsf{negl}(k)$$

for any $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$ that defines the algebraic language $\mathcal{L}_{\mathtt{lpar}}$.

Proof. Let us first consider the case that $\mathbf{M}$ is of dimension $d \times 1$ for any $d > 0$, so $\mathbf{r}$ is a single element. Let $\mathcal{A}$ be an algebraic PPT distinguisher for the games depicted in Fig. 5. In Fig. 8 we show how to exploit $\mathcal{A}$ in order to define an adversary $\mathcal{B}$ to $(1, 2)$-SPDL problem. The reduction proceeds as follows: $\mathcal{B}$ first picks some language parameter $\mathtt{lpar}$ and then runs the first stage of $\mathcal{A}$ in order to obtain $(\mathbf{x}, \mathbf{w}_0, \mathbf{w}_1)$. Note that since $\mathcal{B}$ knows $\mathtt{lpar}$ as field elements, $\mathcal{A}$ is algebraic and $\mathcal{A}$ receives as input only the generators, we can assume that $\mathcal{B}$ knows $\mathbf{x}$ as field elements. Next, $\mathcal{B}$ samples some uniformly random elements $(u_1, u_2, u_r, t_1, t_2, t_r)$ to embed the challenge as elements $u_i y + t_i$. This is a standard procedure frequently used to embed a univariate challenge in a multivariate polynomial [34,9]. Note that elements u_1, u_2, u_r and y are perfectly hidden to $\mathcal{A}$ as they are "one-time padded" with t_i-s. This property will be used later in the proof. Then, $\mathcal{B}$ samples uniformly random trapdoors e_1, e_2, computes π and then runs the second phase of $\mathcal{A}$ in order to obtain the distinguisher bit b'. Note that $\mathcal{B}$ needs $[Y^2]_2$ in order to compute elements of the form $[s_i \mathbf{r}]_2$.

```
$\mathcal{B}^{\mathcal{A}}_{2\text{-}SPDL}([y]_1, [y, y^2]_2)$
Fix any $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$;
$(\mathbf{x}, \mathbf{w}_0, \mathbf{w}_1) \leftarrow \mathcal{A}([1]_1, [1]_2, \mathtt{lpar})$;
$b \leftarrow \{0, 1\}$;
$u_r, u_1, u_2, t_r, t_1, t_2 \leftarrow \mathbb{Z}_p$;
if $u_r[y]_1 + t_r[1]_1 = [0]_1$, then return $- t_r/u_r$; fi ; (1)
if $\exists i \in \{1, 2\}$ such that $u_i[y]_1 + t_i[1]_1 = [0]_1$, then return $- t_i/u_i$; fi ; (2)
$e_1, e_2 \leftarrow \mathbb{Z}_p; (e_1 \neq e_2)$
for $\iota \in \{1, 2\}$   $[\mathbf{r}]_\iota = u_r[y]_\iota + t_r[1]_\iota$;
  for $i \in \{1, 2\}$   $[s_i]_\iota = u_r[y]_\iota + t_r[1]_\iota$;
    $[s_i\mathbf{r}]_2 = u_i u_r[y^2]_2 + (u_i t_r + u_r t_i)[y]_2 + t_i t_r[1]_2$;
    $[d_i]_2 = e_i \mathbf{w}_b [s_i]_2 + [s_i\mathbf{r}]_2$;
    if $[d_i]_2 = [0]_2$ then return $- (t_r + e_i \mathbf{w}_b)/u_r$; fi ; (3)
  endfor
endfor $(*)$
$\pi = ([\mathbf{M}(\mathbf{x})\mathbf{r}, s_1, s_2]_1, [s_1, s_1 e_1, s_2, s_2 e_2, d_1, d_2]_2)$;
$b' \leftarrow \mathcal{A}([\mathbf{M}(\mathbf{x})]_1, \mathbf{w}_0, \mathbf{w}_1, \pi)$;
  $(\vec{\phi}, \vec{\psi}, \vec{\sigma}) \leftarrow \mathsf{Ext}_{\mathcal{A}}$;
  Find all the roots of the univariate polynomial
  $V(Y) = V(u_1 Y + t_1, u_2 Y + t_2, e_1, e_2, u_r Y + t_r, \mathbf{x})$;
  Check if one of the roots $y^*$ is equal to $y$; If yes return $y^*$ else return $\perp$;
```

Fig. 8: SPDL reduction for the new assumption. Polynomials Φ, Σ, Ψ are as defined in Eq. (2) and polynomial V is as defined in Eq. (3).

Let us define the polynomials

$$\begin{aligned}
\Phi(\vec{S}, \vec{E}) &= \sum_{i=1}^{2}[\phi_{0i} S_i + \phi_{1i} S_i E_i] + \sum_{i,j=1}^{2}[\phi_{2ij} S_i S_j + \phi_{3ij} S_i S_j E_j];\\
\Sigma(R, X) &= \sigma_0 + \sigma_1 \mathbf{M}(X) R.\\
\Psi(\vec{S}, \vec{E}, R, X) &= \sum_{i=1}^{2}[\psi_{0i}(S_i E_i \mathbf{w} + S_i R) + \psi_{1i}\mathbf{M}(X) R S_i + \psi_{2i}\mathbf{M}(X) R S_i E_i]\\
&\quad + \sum_{i=1}^{2}[\psi_{3i}\mathbf{M}(X) R (S_i E_i \mathbf{w} + S_i R)] + \sum_{i,j=1}^{2}[\psi_{4ij} S_j (S_i E_i \mathbf{w} + S_i R)];
\end{aligned} \tag{2}$$

Since $\mathcal{A}$ is supposed to be an algebraic distinguisher, there exists a PPT extractor $\mathsf{Ext}_{\mathcal{A}}$ that computes coefficients $(\vec{\psi}, \vec{\phi}, \vec{\sigma})$ such that the following verification polynomial $V(\vec{S}, \vec{E}, R, X) = \Phi(\vec{S}, \vec{E}) + \Psi(\vec{S}, \vec{E}, R, X) + \Sigma(R, X)$ is 0 when evaluated in the point defined by $\mathcal{A}$'s inputs. That is,

$$V(\vec{s}, \vec{e}, \mathbf{r}, \mathbf{x}) = \Phi(\vec{s}, \vec{e}) + \Psi(\vec{s}, \vec{e}, \mathbf{r}, \mathbf{x}) + \Sigma(\mathbf{r}, \mathbf{x}) = 0. \tag{3}$$

It is easy to see that V is the polynomial taking all the possible pairings among $\mathcal{A}$'s inputs. As shown in Fig. 8, $\mathcal{B}$ invokes $\mathsf{Ext}_{\mathcal{A}}$ to compute coefficients $(\vec{\phi}, \vec{\psi}, \vec{\sigma})$ of V.

Recall that by definition of algebraic distinguisher, with high probability, V must have a number of non-zero coefficients, such that the view of $\mathcal{A}$, when restricted to the input corresponding to the non-zero monomials, is distributed differently in the two games. Particularly, in our case this implies that V must explicitly depend on the used witness $\mathbf{w}_b$. Note that the monomials of V in which $\mathbf{w}_b$ is multiplied by $\mathbf{M}(\mathbf{x})$ are the same in both games, since $\mathbf{M}(\mathbf{x})\mathbf{w}_0 = \mathbf{M}(\mathbf{x})\mathbf{w}_1 = \boldsymbol{\theta}(\mathbf{x})$. Thus $[\mathbf{View}^{\mathbf{w}_b}_{\mathcal{A}}]_{supp(\vec{\phi},\vec{\psi},\vec{\sigma})}$ for $b \in \{0, 1\}$ are distributed differently if and only if V has a non-zero coefficient that corresponds to a monomial in which $\mathbf{w}_b$ but not $\mathbf{M}(\mathbf{x})$ appears. Formally, let ϵ be the advantage of $\mathcal{A}$ in distinguishing the two distributions, and t be the running time of $\mathcal{A}$. Let Hit be the event that V explicitly depends on the used witness $\mathbf{w}_b$. Then $\Pr[\mathsf{Hit}] \geq \epsilon/t^2$ by the definition of algebraic distinguishers.

We first observe that $\mathcal{B}$ stops before the point labeled as $(*)$ with negligible probability. This can be concluded by the fact that $(u_1, u_2, u_r, t_1, t_2, t_r)$ and e_1, e_2 are sampled from uniform distributions which implies that the elements $e_i\mathbf{w}_b$ are distributed uniformly at random too.

We now show that $\Pr[\mathcal{B} \text{ wins}] \geq \mathsf{negl}(k) + \Pr[\mathsf{Hit}]$. Note that the variable R appears in Ψ only multiplied by at least one of the variables S_1, S_2. Suppose that $\Psi(\vec{S}, \vec{e}, R, \mathbf{x})$ is a polynomial of degree at least 1 in R. So, there exists a non-zero element $\tilde{\psi}$ of $\vec{\psi}$ that corresponds to a monomial in which the variable R appears. Let $S_1^z S_2^q R J(e_1, e_2, \mathbf{x})$, for $z, q \in \{0, 1, 2\}$ and some J, be this monomial. Since Σ is independent from S_i and Φ is independent from R, then $V(S_1, S_2, \vec{e}, R, \mathbf{x}) = \tilde{\psi} S_1^z S_2^q R J(e_1, e_2, \mathbf{x}) + P(S_1, S_2, \vec{e}, R, \mathbf{x})$ where P is a trivariate polynomial that does not contain a monomial of the type $S_1^z S_2^q R$. Thus, $V(S_1, S_2, \vec{e}, R, \mathbf{x})$ is also a non-zero polynomial of degree at least 2. Suppose now that $\Psi(\vec{S}, \vec{e}, R, \mathbf{x})$ is of degree 0 in R. This can happen if and only if $\Psi = -\sum_{i=1}^{2} \mathbf{M}(\mathbf{x})\mathbf{w}\psi_{1i} S_i E_i$, ($\psi_{0i} = -\psi_{1i}$, and other coefficients of Ψ are equal to 0). Thus Ψ, and also V are independent from $\mathbf{w}$ and the view of $\mathcal{A}$ is the same in the two games. Note in fact that $\mathbf{M}(S_i E_i \mathbf{w}) = \boldsymbol{\theta}(\mathbf{x}) S_i E_i$ for each valid witness. By definition of Hit, this cannot happen. Thus, we have shown that, conditioned on the event Hit, V is a non-zero polynomial of (total) degree at least 2.

We now recall a lemma from [9] that we use in our proof.

Lemma 3 ([9]). *Let $V(X_1, ..., X_m)$ be a non-zero multivariate polynomial in $\mathbb{Z}_p$ of total degree d. For each vectors $\vec{u}, \vec{t}$ of length m, define $V(Y)$ as $V(Y) = P(u_1 Y + t_1, \ldots, u_m Y + t_m)$. Then the coefficient of maximal degree of Q is a polynomial in $u_1, \ldots, u_m$ of degree d.*

By applying this lemma, we have that the coefficient of the term with maximal degree in $V(Y) = V(u_1Y + t_1, u_2Y + t_2, e_1, e_2, u_rY + t_r, \mathbf{x})$ is polynomial in u_1, u_2, u_r of degree at least 2. Let $v(u_1, u_2, u_r)$ be this term. Since u_1, u_2, u_r are perfectly hidden to $\mathcal{A}$, the probability that $v(u_1, u_2, u_r) = 0$ is negligible based on the Schwartz-Zippel lemma.

Summing up, we have $V(y) = 0$ and, conditioned on Hit, $V(Y) \neq 0$ as a polynomial, except with negligible probability. This shows that $\Pr[\mathcal{B} \text{ wins}] \geq \mathsf{negl}(k) + \Pr[\mathsf{Hit}] - \mathsf{negl}(k) \geq \mathsf{negl}(k) + \epsilon/t^2$.

Thus, $\epsilon \leq t^2 \Pr[\mathcal{B} \text{ wins}] + \mathsf{negl}(k)$. The fact that $\Pr[\mathcal{B} \text{ wins}]$ is negligible by assumption concludes the proof for the case of $(d \times 1)$-dimension $\mathbf{M}$.

What is left is to generalize the proof to the case where $\mathbf{M}$ is a $n \times k$ matrix. In this case, $\mathcal{B}$ will sample k different and independently chosen uniformly random u_{ir}, t_{ir} and define each value of $\mathbf{r}$ as $u_{ir}Y + t_{ir}$. Then, instead of having just one verification polynomial V, we have k verification polynomials $\{V_i\}_{i \in [k]}$, one for each line of $\mathbf{M}$. By the definition of AGM distinguisher, at least one of these polynomials, say V_i, must explicitly depend on $\mathbf{w}_b$. Applying the same procedure to V_i as described above completes the proof. □

C Partial Extractability of CH Framework

C.1 f-extractability of CH Proof systems

We show that the CH NIZK proof system satisfies f extractability where $f(x)$ is the encoding of x to $\mathbb{G}_2$.

Lemma 4 (Lemma 1 restated). *The NIZK proof system of [25] depicted in Fig. 3 is* $[\cdot]_2$*-extractable.*

Proof. We show the existence of an efficient extractor Ext that given a trapdoor $\mathtt{td}$ and a valid proof π for any statement $[\mathbf{x}]_1$, outputs partial witness $\widetilde{\mathbf{w}}$. Let $\mathtt{td} = (e_1, e_2, s_1, s_2)$. By relying on the soundness of the NIZK proof and the fact that a valid proof $\pi = ([\mathbf{a}]_1, [\mathbf{d}_1, \mathbf{d}_2]_2)$ must satisfy the verification equations, Ext computes a partial witness $\widetilde{\mathbf{w}}$ as follows:

- $[\mathbf{d}'_i]_2 := [\mathbf{d}_i]_2 s_i^{-1} = \mathbf{w}[e_i]_2 + \mathbf{r}[1]_2$
- $\vec{u} = [\mathbf{d}'_1]_2 - [\mathbf{d}'_2]_2$
- **return** $\widetilde{\mathbf{w}} = \vec{u}(e_1 - e_2)^{-1}$

It is easy to see that $\widetilde{\mathbf{w}} = [\mathbf{w}]_2$ with probability 1.

C.2 Strong Partial Extractability of CH proof systems

To ease exposition we first show the proof for linear languages and then prove the general case for any 1DL-friendly language.

Lemma 5. *Let* $\mathcal{L}_{\mathtt{lpar}}$ *be any linear language defined by* $\mathtt{lpar} = [\mathbf{M}]_1$*. Assuming that co-CDH problem is hard, the NIZK proof system in Fig. 3 for* $\mathcal{L}_{\mathtt{lpar}}$ *is strong* $[\cdot]_2$*-extractable.*

Proof. From Lemma 1, we have that the proof system is $[\cdot]_2$-extractable. This means that for any adversarially generated $([\mathbf{x}]_1, \pi)$ which passes the verification, the extractor can extract $\widetilde{\mathbf{w}} = [\mathbf{w}]_2$. To prove that it satisfies decidability property, we define the algorithm D as follows: $\mathsf{D}([\mathbf{x}]_1, \widetilde{\mathbf{w}} = [\mathbf{w}]_2)$ returns 1 if $[\mathbf{M}]_1[\mathbf{w}]_2 = [\mathbf{x}]_1[1]_2$. It is clear that D is efficient. To show how D decides the membership of $[\mathbf{x}]_1$, note that the pairing equality holds iff $\mathbf{w} = f^{-1}([\mathbf{w}]_2)$ (for $f(x) := [x]_2$) is a valid witness for $[\mathbf{x}]_1$, i.e., $([\mathbf{x}]_1, \mathbf{w}) \in \mathcal{R}_{\mathtt{lpar}}$. We now argue that, compute $\mathbf{w} = f^{-1}([\mathbf{w}]_2)$, is as hard as computing a valid proof π' for $[\mathbf{x}]_1$ given $\widetilde{\mathbf{w}} = [\mathbf{w}]_2$. Clearly, computing $\mathbf{w} = f^{-1}([\mathbf{w}]_2)$ is hard given the hardness of discrete logarithm in $\mathbb{G}_2$. We now show the hardness of computing a valid proof, given a partial witness $\widetilde{\mathbf{w}}$, by a reduction to the co-CDH problem. Recall that co-CDH problem asks to compute $[XY]_2$, given $([1, X, Y]_2) \in \mathbb{G}_2$ and $([1, X]_1) \in \mathbb{G}_1$ as input.

Consider the linear language $\mathcal{L}_{\mathtt{lpar}}$, defined by $\mathtt{lpar} = [\mathbf{M}]_1$, where $\mathbf{M} = (m_{ij}) \in \mathbb{Z}_p^{n \times k}$. W.l.o.g we can assume that the first entry of $\mathbf{M}$ (i.e., m_{11}) is non-zero [11]. Let $\mathcal{A}$ be an efficient algorithm that on input $(\mathtt{lpar}, \mathtt{crs}, [\mathbf{x}]_1, \widetilde{\mathbf{w}})$ computes a valid proof π with non-negligible probability ϵ. We construct an efficient algorithm $\mathcal{B}$ against co-CDH problem so that on input challenge $([1, X]_1, [1, X, Y]_2)$ proceeds as follows:

- Generate the CRS parameters by sampling $s_1, s_2, e_1 \leftarrow \mathbb{Z}_p$ and set $e_2 = [Y]_2$. Let $\mathtt{crs} = ([s_1, s_2, s_1e_1, s_2e_2]_2)$. It is clear that the distribution of $\mathtt{crs}$ is the same as an honestly generated CRS.
- Define $[w_1]_1 = [X]_1$ and sample uniformly random elements $w_2, \dots, w_k \leftarrow \mathbb{Z}_p$. Let $[\mathbf{w}]_1 = [w_1, \dots, w_k]_1$ and compute $[\mathbf{x}]_1 = \mathbf{M}[\mathbf{w}]_1$. Compute also $\widetilde{\mathbf{w}} = [\mathbf{w}]_2$, where $[w_1]_2 = [X]_2$ is from the challenge.
- Run $\mathcal{A}$ on input $(\mathtt{lpar}, \mathtt{crs}, [\mathbf{x}]_1, \widetilde{\mathbf{w}})$ to obtain a proof $\pi = ([\mathbf{a}]_1, [\mathbf{d}_1, \mathbf{d}_2]_2)$.
- Check if π makes the verifier accepts; and abort otherwise.
- Let u be the first entry of the vector $(s_1\mathbf{M}[\mathbf{d}_2]_2 + s_1s_2e_1[\mathbf{x}]_2 - s_2\mathbf{M}[\mathbf{d}_1]_2)/(s_1s_2)$. Return $([u]_2 - (\sum_{i=2}^{d} m_{1i}w_i)[e_2]_2)/m_{11}$.

To see that the output is $[XY]_2$, we note that, if the verifier accepts, then $\mathbf{M}\mathbf{d}_i = \mathbf{x}s_ie_i + s_i\mathbf{a}$ for $i \in 1, 2$. Thus, we have $\mathbf{M}\mathbf{w}Y = \mathbf{x}Y = \mathbf{x}e_2 = (s_1\mathbf{M}\mathbf{d}_2 + s_1s_2e_1\mathbf{x} - s_2\mathbf{M}\mathbf{d}_1)/(s_1s_2)$. To complete the proof we note that, since $X = \mathbf{w}_1$, the first entry of $\mathbf{M}\mathbf{w}Y$ is equal to $m_{11}XY + (\sum_{i=2}^{d} m_{1i}w_i)Y$. This shows that $\mathcal{B}$ returns $[XY]_2$ with at least the same probability ϵ that $\mathcal{A}$ computes a valid proof given only $\widetilde{\mathbf{w}}$ as the witness. □

Strong partial extractability for 1DL-friendly languages. Fix any $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$ such that the defined algebraic language is 1DL-friendly. Thus, there exists two affine functions λ_x, λ_w such that, $\mathbf{M}(\lambda_x(X))\lambda_w(X) = \boldsymbol{\theta}(\lambda_x(X))$. Here, with a little abuse of notation, we implicitly assume that $X = w_1$, $w_2, ..., w_n$ are fixed values and $\lambda_x(X)$ indicates $\lambda_x(X, w_2, ..., w_n)$. Same for λ_w. Since the

[11] This is without loss of generality since columns of $\mathbf{M}$ can be assumed to be linearly independent.

composition of a linear and an affine map is still affine, we have that each entry of $(\mathbf{M}(\lambda_x(X)))_{ij}$ is defined by an affine function $m_{1ij}X + m_{0ij}$. Moreover, each entry of $(\lambda_w(X))_j$ is defined by an affine function $w_{1j}X + w_{0j}$. Note that each $w_2, ..., w_n$ corresponds to a set of coefficients $m_{1ij}, m_{0ij}, w_{1j}, w_{0j}$; and viceversa, i.e., to each set of coefficients, at least one choice of $w_2, ..., w_n$ is corresponded. Thus, we can assume that the reduction below knows $w_2, ..., w_n$. Given, any $e \in \mathbb{Z}_p$, for any $i \in \{1, ..., n\}$ let us define the polynomial

$$g_{iT}(X,Y) = X^2Y\sum_j(m_{1ij}w_{1j}) + XY\sum_j(m_{1ij}w_{0j} + m_{0ij}w_{1j}) - X^2e\sum_j(m_{1ij}w_{1j}) \tag{4}$$

Note that this polynomial is not in the subspace generated by the base $\{1, X, X^2, Y, XY\}$, as long as $\sum_j(m_{1ij}w_{1j}) \neq 0$. Following the framework of Uber-assumptions (see [19]) we define the following assumption.

Assumption 7 *Let g_{iT} be any polynomial as defined in Eq.* (4)*, such that $\sum_j(m_{1ij}w_{1j}) \neq 0$. For any PPT adversary $\mathcal{A}$ it holds that:*

$$\Pr\left[t = g_{iT}(x,y) \middle| x, y \leftarrow \mathbb{Z}_p; [t]_T \leftarrow \mathcal{A}([1,x]_1, [1,x,y]_2)\right] \leq \mathsf{negl}(k)$$

Lemma 6. *If Assumption 7 holds, then the NIZK proof system in [25], depicted in Fig. 3 for any 1DL-friendly language, is strong $[\cdot]_2$-extractable.*

Proof. The first part of the proof is the same as in Lemma 5. We just need to show that, given $\widetilde{\mathbf{w}} = [\mathbf{w}]_2$, computing a proof is hard. We prove the hardness under Assumption 7.

Consider any g_{it} defined by the choice of any 1DL-friendly language and any $e \in \mathbb{Z}_p$. Suppose that $\mathcal{A}$ can efficiently compute a valid proof with high probability, having on input $(\mathtt{lpar}, \mathtt{crs}, [\mathbf{x}]_1, \widetilde{\mathbf{w}} = [\mathbf{w}]_2)$. We show how to define an efficient adversary $\mathcal{B}$ to compute $[g_{it}(x,y)]_T$, given the challenge $([1,x]_1, [1,x,y]_2)$. $\mathcal{B}$ is defined as follows.

- Let e be as in Eq. (4). If $e = y$ then compute $[g_{iT}(x,y)]_T$. Note that in this case $[g_{iT}(x,y)]_T$ can be easily computed having $([1,x,xy]_1, [1,x,y]_2)$.
- Else, generate the CRS parameters by sampling $s_1, s_2 \leftarrow \mathbb{Z}_p$ and set $[e_1]_2 = [y]_2, e_2 = e$. Let $\mathtt{crs} = ([s_1, s_2, s_1e_1, s_2e_2]_2)$. It is clear that the distribution of $\mathtt{crs}$ is the same as one that is generated honestly.
- Define $[\mathbf{x}]_1 = [\lambda_x(x)]_1, [\mathbf{w}]_1 = [\lambda_w(x)]_1, [\mathbf{w}]_2 = [\lambda_w(x)]_2$. Note that $\widetilde{\mathbf{w}} = [\mathbf{w}]_2$.
- Run $\mathcal{A}$ on input $([1,\mathbf{x}]_1, [1, s_1, s_2, s_1e_1, s_2e_2, \mathbf{w}]_2)$ to obtain a proof $\pi = ([\mathbf{a}]_1, [\mathbf{d}_1, \mathbf{d}_2]_2)$.
- Check if π makes the verifier accepts, otherwise $\mathcal{B}$ abort.
- Compute $[\delta^{\mathbf{d}}]_2 = (s_2/s_1)[\mathbf{d}_1]_2 - [\mathbf{d}_2]_2$.
- Let $[out]_T$ be

$$[x]_1(\sum_j(m_{1ij}[\delta^{\mathbf{d}}_j]_2 + s_2em_{1ij}w_{0ij}[1]_2 + s_2em_{0ij}w_{1ij}[1]_2))$$
$$+[-1]_1(\sum_j s_2m_{0ij}w_{0j}[y]_2) + [1]_1(\sum_j(m_{0ij}[\delta^{\mathbf{d}}_j]_2 + s_2em_{0ij}w_{0ij}[1]_2)).$$

Output $(1/s_2)[out]_T$.

To see that $\mathcal{B}$'s output is equal to $g_{i,T}(x,y)$, we note that, if the verifier accepts, then, for $i \in \{1,2\}$,

$$\mathbf{M}([\lambda_x(X)]_1)[\mathbf{d}_i]_2 = \boldsymbol{\theta}([\lambda_x(X)]_1)[s_i e_i]_2 + [\mathbf{a}]_1[s_i]_2,$$

which implies

$$\mathbf{M}([\lambda_x(X)]_1)[\delta^{\mathbf{d}}]_2 = \boldsymbol{\theta}([\lambda_x(X)]_1)[s_2(Y-e)]_2,$$

where the last equality follows by observing that $e_1 = Y$, and $e_2 = e$, and then by multiplying the first equation with s_2/s_1 and subtracting the second. The i-th row of the previous equation defines the polynomial

$$\sum_j ((m_{1ij}X + m_{0ij})\delta_j^{\mathbf{d}}) = s_2(Y-e)\sum_j ((m_{1ij}X + m_{0ij})(w_{1j}X + w_{0j})).$$

Thus, we have

$$\begin{aligned} s_2 g_{i,T}(X,Y) = X &\left[\sum_j (m_{1ij}\delta_j^{\mathbf{d}} + s_2 e(m_{1ij}w_{0ij} + m_{0ij}w_{1ij}))\right] \\ &- Y(s_2\sum_j m_{0ij}w_{0j}) + \sum_j (m_{0ij}\delta_j^{\mathbf{d}} + s_2 e m_{0ij}w_{0ij}). \end{aligned}$$

This completes the proof. □

D Full Extractability for the CH Framework

D.1 Knowledge Soundness of CH Argument Systems in the AGM

We show knowledge soundness of the argument system in Fig. 2 in the AGM framework. We recall that AGM essentially states that for every efficient algorithm $\mathcal{A}$ that outputs the vector $[\mathbf{y}]_\iota$ of group elements in $\mathbb{G}_\iota$ when given inputs the vector $[\mathbf{x}]_\iota$ of group elements in $\mathbb{G}_\iota$, there exists an efficient extractor $\mathsf{Ext}_{\mathcal{A}}$ that returns a matrix $\mathbf{A}$ such that $\mathbf{y} = \mathbf{A}\mathbf{x}$. In particular, since we are working in the setting of asymmetric bilinear pairings, we require that any outputs in one group must depend only on the inputs it receives in that group.

Lemma 7. *The NIZK argument in Fig. 2 is knowledge sound in the algebraic group model for asymmetric pairings, under DL-assumption in $\mathbb{G}_2$.*

Proof. Let $\mathcal{A}$ be a knowledge soundness adversary that on input $[1]_1, [1,e]_2$ outputs $[\mathbf{x}, \mathbf{a}]_1, [\mathbf{d}]_2$. Since the verification equations hold, we have that,

$$\mathbf{M}(\mathbf{x}) \cdot \mathbf{d} = \boldsymbol{\theta}(\mathbf{x}) \cdot e + \mathbf{a} \cdot$$

Now, since $\mathcal{A}$ is an algebraic algorithm, there exists an extractor that outputs vectors $\mathbf{d}_0, \mathbf{d}_1, \mathbf{a}_0, \mathbf{a}_1$ such that $\mathbf{d} = \mathbf{d}_0 + \mathbf{d}_1 e$, $\mathbf{a} = \mathbf{a}_0$, $\mathbf{x} = \mathbf{a}_1$. The knowledge soundness extractor simply outputs $\mathbf{w} = \mathbf{d}_1$.

We show that this extractor outputs a witness whenever the verifier accepts, except with negligible probability. Each equation defined by the verifier's test can be written as a a univariate polynomial $Q_i(X) := d_{0i} + d_{1i}X = x_i X + a_i$ where $d_{0i} = (\mathbf{M}(\mathbf{a}_1) \cdot \mathbf{d}_0)_i$, $d_{1i} = (\mathbf{M}(\mathbf{a}_1) \cdot \mathbf{d}_1)_i$, $x_i = \boldsymbol{\theta}(\mathbf{a}_1)_i$ and $a_i = \mathbf{a}_{0i}$. Suppose for the sake of contradiction that $\mathcal{A}$ computed a valid proof $[\mathbf{a}]_1, [\mathbf{d}]_2$ for an adaptively chosen statement $[\mathbf{x}]_1$, but $\mathbf{w}$ output by the extractor as described above is not a valid witness, that is, $\mathbf{M}(\mathbf{x}) \cdot \mathbf{w} \neq \boldsymbol{\theta}(\mathbf{x})$. Then we can use $\mathcal{A}$ to break the DL assumption in $\mathbb{G}_2$.

The DL adversary receives a challenge $[e]_2$ and invokes $\mathcal{A}$ on input $\texttt{crs} = [e]_2$. Then, the DL adversary obtains $\mathbf{d}_0, \mathbf{d}_1, \mathbf{a}_0, \mathbf{a}_1$ as defined above by the extractor for the algebraic adversary $\mathcal{A}$. If each polynomial $Q_i(X)$ is identically 0, that is $Q_i(X) \equiv 0$ as a polynomial, then $\mathbf{M}(\mathbf{a}_1) \cdot \mathbf{d}_1 = \boldsymbol{\theta}(\mathbf{a}_1)$ which implies that $\mathbf{w} = \mathbf{d}_1$ and the extractor doesn't fail. Otherwise, there exists i such that $Q_i(e) = 0$, for a non-zero polynomial $Q_i(X)$. Then $e = (a_i - d_{0i})/(d_{1i} - x_i)$ is the only root of $Q_i(e)$. Note that $Q_i(X) \not\equiv 0$ implies that $d_{1i} \neq x_i$ and thus the DL adversary succeeds in breaking the DL-assumption in $\mathbb{G}_2$.

D.2 Semantic, BB and n-BB Extraction

Semantic extraction demands that for every adversary that implements a strategy (an efficiently computable function that outputs an accepting proof) there exists an extractor. Unlike n-BB extraction where there could be a different extractor for every machine, in semantic extraction, one extractor for a function is a good extractor for all machines that implement that function. Semantic extraction is non-blackbox only in the randomness of the adversary but treats the adversary's machine as a black-box; our formal definition allows the extractor access to a part of the adversary's randomness. By allowing the extractor to see all or none of the prover's randomness, the semantic definition recovers standard n-BB and BB extraction definitions. We show that a NIZK satisfies semantic extraction where the extractor is given all the randomness of the adversary (called semn-BB) if and only if it satisfies the standard n-BB extraction definition. While it seems intuitive that the extractor's (in)ability to see the adversary's random coins makes the semantic extractor (BB)n-BB, this is not straightforward, especially the equivalence with BB definition. A BB extractor is also a semantic extractor. For the other direction, consider the case when the semantic extractor is not allowed to see the adversary's randomness; here we would like to argue that such a semantic extractor (called semBB) is a BB extractor. However, semantic extraction only guarantees a (potentially different) extractor for every function implemented by a prover. We therefore have to switch the order of quantifiers in order to construct one *universal* extractor that works for all provers. For a relaxed concrete security notion of extraction, we can indeed show this concluding that a special case of semantic extraction semBB implies BB extraction.

At a high-level, we rely on the minimax theorem from game theory to construct a universal extractor from function-dependent extractors. We define a utility function to capture how well the extractor performs. The minimax theorem guarantees the existence of a distribution over extractors. Computing this distribution is not guaranteed to be efficient. This can be done efficiently by using a multiplicative weights algorithm [33] to implement an approximate minimax strategy by knowing the randomness used by the adversary. However, this use of the adversary's randomness makes the universal extractor non-blackbox. We then show how to make the universal extractor BB without the randomness of the adversary. Our use of minimax is reminiscent of its use in proving the equivalence of distinguisher-dependent and universal simulators in [24], and in switching the order of quantifiers in the proof of the leakage lemma in [37].

From Semantic to BB *and* $\mathsf{n\text{-}BB}$. In the definition of semantic extraction, the function implemented by the adversary uses randomness (s, t), and the extractor receives t, but not s; thus the extractor is allowed to see a part of the adversary's randomness. Let us consider the two extremes of the extractor's access: (i) the extractor is not given even r, that is, does not see the randomness of the adversary. (ii) the extractor is given both (s, t), that is, the extractor sees the entire randomness of the adversary. Intuitively, the former is black-box in the adversary, and the latter is white-box. However, in order to establish the equivalences, we also have to be careful with the order of quantifiers in the definition of extraction, which is different in the black-box and the semantic notion. In this section, we show that versions of semantic extraction where we control the randomness access of the extractor as in (i) and (ii) are equivalent to standard black-box (one side of the equivalence additionally needs a relaxed concrete (t, ϵ) variant of the definitions) and white-box definitions respectively.

We first give the concrete security definitions of black-box extraction, and *semantic black-box extraction* which is the semantic definition where the extractor is not given the randomness of the prover.

Definition 15. *A NIZK argument $\Pi = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$ is semantic black-box knowledge sound (semBB) if for each efficiently implementable knowledge soundness strategy f there exists a PPT extractor $\mathsf{Ext} = \mathsf{Ext}_f$, such that, for each (even unbounded) TM $\mathcal{A}^*$ that implements f*

$$\Pr\left[\begin{array}{c}\mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1\\ \wedge(\mathtt{x}, \mathtt{w}) \notin \mathcal{R}\end{array}\middle|\begin{array}{c}(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k); (s, t) \leftarrow D\\ (\mathtt{x}, \pi) \leftarrow \mathcal{A}^*(\mathtt{crs}; s, t); \mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi)\end{array}\right] \leq \mathsf{negl}(k).$$

Definition 16. *A NIZK argument $\Pi = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$ is (t, ϵ) black-box knowledge sound, if there exists an extractor $\mathsf{Ext}_{\mathsf{bb}}$ such that, for any t-time adversary $\mathcal{A}$:*

$$\Pr\left[\begin{array}{c}\mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1\\ \wedge(\mathtt{x}, \mathtt{w}) \notin \mathcal{R}\end{array}\middle|\begin{array}{c}(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k);\\ (\mathtt{x}, \pi) \leftarrow \mathcal{A}(\mathtt{crs}; r); \mathtt{w} \leftarrow \mathsf{Ext}_{\mathsf{bb}}(\mathtt{td}, \mathtt{x}, \pi)\end{array}\right] \leq \epsilon(k)$$

where r is the random coins of the adversary.

Definition 17. *A NIZK argument $\mathit{\Pi} = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$ is (t, ϵ) semBB (semantic black-box knowledge sound) if for each t-time implementable knowledge soundness strategy f, there exists a PPT extractor $\mathsf{Ext} = \mathsf{Ext}_f$, such that, for each (even unbounded) TM $\mathcal{A}^*$ that implements f*

$$\Pr\left[\begin{matrix}\mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 \\ \wedge(\mathtt{x}, \mathtt{w}) \notin \mathcal{R}\end{matrix}\middle| \begin{matrix}(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k); (s, t) \leftarrow D \\ (\mathtt{x}, \pi) \leftarrow \mathcal{A}^*(\mathtt{crs}; s, t); \mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi)\end{matrix}\right] \leq \epsilon(k).$$

Theorem 4. *Let $\mathit{\Pi}$ be a NIZK argument that is BB knowledge sound as in Definition 1. $\mathit{\Pi}$ is also semBB knowledge sound as in Definition 15. Conversely, if a NIZK argument $\mathit{\Pi}$ is (t, ϵ) semBB knowledge sound as in Definition 17, for each polynomial t and inverse polynomial ϵ, then $\mathit{\Pi}$ is (t', ϵ') BB knowledge sound for every polynomial t' and inverse polynomial ϵ' as in Definition 16.*

Proof. The first implication is straightforward. Let Ext be a BB extractor that satisfies Definition 1. Then this extractor is, by definition, a semantic black-box extractor for each efficiently implementable knowledge soundness strategy as in Definition 1.

We now prove the second implication. Suppose $\mathit{\Pi}$ is (t, ϵ) semBB as in Definition 17, for each polynomial t and inverse polynomial ϵ. Let t' be any polynomial, and ϵ' any inverse polynomial. We show that $\mathit{\Pi}$ is (t', ϵ') BB by constructing an extractor $\mathsf{Ext}_{\mathsf{BB}}$ and showing that it satisfies Definition 16.

High-level description of the extractor. The universal extractor $\mathsf{Ext}_{\mathsf{BB}}$ on input $\mathtt{td}, \mathtt{x}, \pi$ uses the multiplicative weights algorithm [33] to find a good set of extractors $(\mathsf{Ext}_1, \ldots, \mathsf{Ext}_L)$, then runs each of the extractors in the set and outputs a witness if at least one of the extractors succeeds.

We define the "advantage" of an extractor Ext with respect to a knowledge soundness strategy $\mathsf{kss} = f$ as follows. Since kss is efficiently implementable, we fix a PPT adversary $\mathcal{A}_{\mathsf{kss}}$ that implements kss.

$$\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}) := \Pr\left[\begin{matrix}\mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 \\ (\mathtt{x}, \mathtt{w}) \in \mathcal{R}\end{matrix}\middle| \begin{matrix}r \leftarrow D, \\ (\mathtt{x}, \pi) = f(\mathtt{crs}; r), \mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi)\end{matrix}\right].$$

Note that we define this advantage for a fixed pair of $(\mathtt{crs}, \mathtt{td})$. We would now like to define this advantage for a distribution over the set kss of knowledge soundness strategies $\mathsf{kss}_1, \ldots, \mathsf{kss}_k$; consider the set $\{\mathcal{A}_{\mathsf{kss}_1}, \mathcal{A}_{\mathsf{kss}_2}, \ldots, \mathcal{A}_{\mathsf{kss}_k}\}$ of efficient uniform machines with description of size $\leq \log k$ that are implementations of the set of kss. We also redefine each $\mathcal{A}_{\mathsf{kss}_j}$ such that it halts and outputs $\perp$ after t' steps. Each fixed t'-time machine $\mathcal{A}$ for $t' = \mathrm{poly}(k)$ will eventually appear in the set.

Given a distribution $\mathcal{D}$ over the set $\{\mathcal{A}_{\mathsf{kss}_1}, \mathcal{A}_{\mathsf{kss}_2}, \ldots, \mathcal{A}_{\mathsf{kss}_k}\}$, we define the advantage of the extractor Ext with respect to the distribution as

$$\mu(\mathsf{Ext}, \mathcal{D}) := \mathbb{E}_{\mathcal{A}_{\mathsf{kss}} \sim \mathcal{D}}\left[\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})\right] = \sum_{\mathcal{A}_{\mathsf{kss}} \in Supp(\mathcal{D})} \mathcal{D}(\mathcal{A}_{\mathsf{kss}}) \cdot \mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}).$$

where $(\mathcal{D}(\mathcal{A}_{\mathsf{kss}_1}), \ldots, \mathcal{D}(\mathcal{A}_{\mathsf{kss}_k}))$ is the vector of probability weights representing $\mathcal{D}$. Our goal is to construct an extractor Ext such that for every t' implementable strategy that is implemented by $\mathcal{A}_{\mathsf{kss}}$, we have that

$$\Pr\left[\begin{array}{c|c}\mathcal{V}(\mathtt{crs},\mathtt{x},\pi)=1 & (\mathtt{crs},\mathtt{td})\leftarrow\mathsf{CRSGen}(1^k); \\ \wedge(\mathtt{x},\mathtt{w})\notin\mathcal{R} & (\mathtt{x},\pi)\leftarrow\mathcal{A}_{\mathsf{kss}}(\mathtt{crs};r);\mathtt{w}\leftarrow\mathsf{Ext}(\mathtt{td},\mathtt{x},\pi)\end{array}\right]\leq\epsilon'(k).$$

We note that, this is equivalent to constructing Ext such that for every t' implementable strategy (implemented by $\mathcal{A}_{\mathsf{kss}}$),

$$\mathbb{E}_{(\mathtt{crs},\mathtt{td})\sim\mathsf{CRSGen}(1^k)}\left[\mu(\mathsf{Ext},\mathcal{A}_{\mathsf{kss}})\right]\geq 1-\mathcal{O}(\epsilon'(k)).$$

We now give an overview of the multiplicative weights algorithm. The extractor emulates a certain number of rounds of a zero-sum game between an extractor player and a knowledge soundness adversary. The payoff function for the extractor is the advantage $\mu(\cdot,\cdot)$. In each round, the knowledge soundness adversary chooses a distribution $\mathcal{D}$, and the extractor chooses Ext_i such that its expected payoff is high. We begin with the uniform distribution $\mathcal{D}^{(1)}$ over $\{\mathcal{A}_{\mathsf{kss}_1},\mathcal{A}_{\mathsf{kss}_2},\ldots,\mathcal{A}_{\mathsf{kss}_k}\}$. In each round, this is updated to $\mathcal{D}^{(i+1)}$ using the multiplicative weights algorithm using the advantage function $\mu(\cdot,\cdot)$. For this, the knowledge soundness adversary in the two player game needs to compute the payoff function $\mu(\mathsf{Ext},\mathcal{A}_{\mathsf{kss}})$ of an extractor that is good with respect to $\mathcal{A}_{\mathsf{kss}}$. We use a universal adversary that takes a description of a knowledge soundness adversary $\mathcal{A}_{\mathsf{kss}}$ as auxiliary input and runs $\mathcal{A}_{\mathsf{kss}}$ in order to compute the payoff function. Then, we choose an extractor that is good with respect to this universal adversary. The extractor, therefore needs to efficiently find the kss-dependent extractor for the mixed strategy $\mathcal{D}^{(i)}$ over kss implementations. This is done by using the universal adversary $\mathcal{A}_U$ that takes the vector of probability weights representing $\mathcal{D}$ as auxiliary input, samples a kss adversary from the distribution, and runs the sampled adversary. Let $\mathsf{Ext}_{\mathcal{A}_U}$ be the extractor for the kss implemented by $\mathcal{A}_U$ that is guaranteed to exist by semantic extraction. In the ith round, we choose Ext_i to be the machine that runs $\mathsf{Ext}_{\mathcal{A}_U}$ given the weights of $\mathcal{D}^{(i)}$ as auxiliary input. The description of this extractor is given in Fig. 9. Later, we show how to make $\mathsf{Ext}_{\mathsf{BB}}$ efficient when Ext_i is not given any auxiliary input.

It can be verified that $\mathsf{Ext}_{\mathsf{BB}}$ runs in time $\mathcal{O}(L[\gamma(t'+T_U)+T_U])=\mathcal{O}(\frac{\log k}{\epsilon'(k^2)}[\frac{\log(kL/\epsilon'(k))}{\epsilon'(k)^2}k(t'+T_U)+T_U])$, that is polynomial in t' and $1/\epsilon'$. To prove the theorem we must show that, for each t' implementable knowledge soundness strategy kss, $\mathbb{E}_{(\mathtt{crs},\mathtt{td})\sim\mathsf{CRSGen}(1^k)}\left[\mu(\mathsf{Ext}_{\mathsf{BB}},\mathcal{A}_{\mathsf{kss}})\right]\geq 1-\mathcal{O}(\epsilon'(k))$. In order to show this, we rely on two auxiliary lemmas: the first shows that if in each round Ext_i does well against $\mathcal{D}^{(i)}$ with respect to $\tilde{\mu}(\cdot,\cdot)$, then Ext does well against each $\mathcal{A}_{\mathsf{kss}}$. This follows from the analysis of the multiplicative weights algorithm. The second lemma shows that the above statement holds for $\mu(\cdot,\cdot)$.

Lemma 8. *For every knowledge soundness strategy implementation $\mathcal{A}_{\mathsf{kss}_j}\in\{\mathcal{A}_{\mathsf{kss}_1},\mathcal{A}_{\mathsf{kss}_2},\ldots,\mathcal{A}_{\mathsf{kss}_k}\}$, the extractor defined in Fig. 9 generates $\mathcal{D}^{(1)},\ldots,\mathcal{D}^{(L)}$ and $\mathsf{Ext}_1,\ldots,\mathsf{Ext}_L$ such that*

$$\frac{1}{L}\sum_{i=1}^{L}\tilde{\mu}(\mathsf{Ext}_i,\mathcal{A}_{\mathsf{kss}_j})\geq\frac{1}{L}\sum_{i=1}^{L}\tilde{\mu}(\mathsf{Ext}_i,\mathcal{D}^{(i)})-\mathcal{O}(\epsilon'(k)).$$

$\mathsf{Ext}_{\mathsf{BB}}(\mathtt{td}, \mathtt{x}, \pi)$.

- Let $\mathcal{D}$ be a distribution over a set of kss implementations $\{\mathcal{A}_{\mathsf{kss}_1}, \mathcal{A}_{\mathsf{kss}_2}, \ldots, \mathcal{A}_{\mathsf{kss}_k}\}$. Let $\mathfrak{w}_{\mathcal{D}}$ be the vector of weights representing $\mathcal{D}$. That is $(\mathfrak{w}_{\mathcal{D}})_j = \Pr\left[\mathcal{A}_{\mathsf{kss}_j} \xleftarrow{\$} \mathcal{D}\right]$. Let $\mathcal{A}_U$ be the PPT that on input $(\mathtt{crs}, r)$ interprets r as $\mathfrak{w}_{\mathcal{D}}||\chi||r'$, samples a knowledge soundness strategy adversary $\mathcal{A}_{\mathsf{kss}_j}$ from $\mathcal{D}$, using random coins χ, and runs $\mathcal{A}_{\mathsf{kss}_j}$ on $(\mathtt{crs}, r')$. Let f be the function implemented by $\mathcal{A}_U$, and T_U be a polynomial that bounds the running time of $\mathcal{A}_U$. Let $\mathsf{Ext}_{\mathcal{A}_U}$ be the (T_U, ϵ') black-box semantic extractor for $\mathcal{A}_U$ as in Definition 17.
- Let $L = \Theta(\frac{\log k}{\epsilon'(k^2)})$ and $\beta = \frac{1}{1+\sqrt{(2\log k)/L}}$.
- Let $\mathcal{D}^{(1)}$ be the uniform distribution over $\{\mathcal{A}_{\mathsf{kss}_1}, \mathcal{A}_{\mathsf{kss}_2}, \ldots, \mathcal{A}_{\mathsf{kss}_k}\}$. For $i = 1, \ldots, L$ do
  1. On input $(\mathtt{td}, \mathtt{x}, \pi)$, consider the adversary $\mathcal{A}_{U_i}(\mathtt{crs}, (\chi||r))$ which is defined as $\mathcal{A}_U(\mathtt{crs}, r')$, where $r' = \mathfrak{w}_{\mathcal{D}^{(i)}}||\chi||r$. Let f_i be the function implemented by $\mathcal{A}_{U_i}$. Note that $f_i(\mathtt{crs}, (\chi||r)) = f(\mathtt{crs}, (\mathfrak{w}_{\mathcal{D}^{(i)}}||\chi||r))$. Note also that f_i is an efficiently implementable knowledge soundness strategy, since the running time of $\mathcal{A}_{U_i}$ is bound by T_U. Let Ext_i be the (T_U, ϵ') black-box semantic extractor for $\mathcal{A}_{U_i}$.
  2. Let $\mathcal{D}^{(i+1)}$ be defined as $\beta^{\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j})} \cdot \mathcal{D}^{(i)}$ up to renormalizing. That is
  $$\mathcal{D}^{(i+1)}(\mathcal{A}_{\mathsf{kss}_j}) = \frac{\beta^{\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j})}\mathcal{D}^{(i)}(\mathcal{A}_{\mathsf{kss}_j})}{\sum_{l=1}^{k} \beta^{\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_l})}\mathcal{D}^{(i)}(\mathcal{A}_{\mathsf{kss}_l})}$$
  where $\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})$ is defined by the procedure in Fig. 10. $\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})$ can be thought of as an approximation of $\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})$.
- Run each extractor Ext_i in the set $\{\mathsf{Ext}_1, \ldots, \mathsf{Ext}_L\}$, and verify if one of them succeeded in computing a valid witness.
- Output a valid witness if available, else output $\perp$.

Fig. 9: The black-box (t', ϵ') extractor.

Let $\gamma = \Theta(\frac{\log(kL/\epsilon'(k))}{\epsilon'(k)^2})$. Let $\mathsf{freq} = 0$. For $i = 1, \ldots, \gamma$ do

1. Sample $r \leftarrow D$. Compute $(\mathtt{x}, \pi) = \mathcal{A}_{\mathsf{kss}}(\mathtt{crs}, r)$.
2. Compute $\mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi)$.
3. If $\mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1$ and $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}$, then $\mathsf{freq} = \mathsf{freq} + 1$.

$\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}) = \mathsf{freq}/\gamma$.

Fig. 10: $\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})$

Proof. Recall that the relative entropy of two random variables X and Y is defined as

$$\mathbf{KL}(X||Y) = \sum_{x \in supp(X)} \Pr[X = x] \ln \frac{\Pr[X = x]}{\Pr[Y = y]}.$$

Now, consider a strategy $\mathcal{A}_{\mathsf{kss}_j} \in \{\mathcal{A}_{\mathsf{kss}_1}, \dots, \mathcal{A}_{\mathsf{kss}_k}\}$. Fix a pair $(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}$. Lastly, fix the random tape of the extractor. in this way, all the random variables that appears in Fig. 9, became fixed.

We begin showing that for each $i \in \{1, \dots, L\}$ we have

$$\mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j}||\mathcal{D}^{(i+1)}) - \mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j}||\mathcal{D}^{(i)}) \leq \\ (\ln\frac{1}{\beta})\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) - (1-\beta)\sum_{b=1}^{k} \Pr\left[\mathcal{D}^{(i)} = \mathsf{kss}_b\right] \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}). \quad (5)$$

Upon fixed i, we have

$$\begin{aligned} &\mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j}||\mathcal{D}^{(i+1)}) - \mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j}||\mathcal{D}^{(i)}) \\ &= \ln\frac{1}{\Pr\left[\mathcal{D}^{(i+1)} = \mathsf{kss}_j\right]} - \ln\frac{1}{\Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_j}\right]} \\ &= \ln\frac{\Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_j}\right]}{\Pr\left[\mathcal{D}^{(i+1)} = \mathcal{A}_{\mathsf{kss}_j}\right]} \\ &= \ln\frac{\sum_{b=1}^{k} \beta^{\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b})} \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right]}{\beta^{\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j})}} \\ &= \ln(\frac{1}{\beta})\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \ln\sum_{b=1}^{k} \beta^{\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b})} \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right]. \end{aligned}$$

Since, $x \in [0,1]$ and $\beta > 0$ imply that $\beta^x \leq 1 - (1-\beta)x$, recalling that $\sum_{b=1}^{k} \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right] = 1$, we have

$$\ln(\frac{1}{\beta})\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \ln\sum_{b=1}^{k} \beta^{\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b})} \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right] \\ \leq \ln(\frac{1}{\beta})\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \ln\left(1 - (1-\beta)\sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right]\right)$$

Lastly, from $x < 1$ implies that $\ln(1-x) \leq -x$, we have

$$\ln(\frac{1}{\beta})\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \ln\left(1 - (1-\beta)\sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right]\right) \\ \leq \ln(\frac{1}{\beta})\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) - (1-\beta)\sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right],$$

which complete the proof of Eq. (5).

Now, summing Eq. (5) over $i \in \{1, ..., L\}$, we have

$$\begin{aligned} &\mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j} || \mathcal{D}^{(L+1)}) - \mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j} || \mathcal{D}^{(1)}) \\ &\leq \ln(\frac{1}{\beta}) \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) - (1-\beta) \sum_{i=1}^{L} \sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right]. \end{aligned}$$

From the last inequality and using $\mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j} || \mathcal{D}^{(L+1)}) \geq 0$, $\mathbf{KL}(\mathcal{A}_{\mathsf{kss}_j} || \mathcal{D}^{(1)}) \leq \ln k$ and $\ln(\frac{1}{\beta}) \leq \frac{1-\beta^2}{2\beta}$ (which holds because $\beta \in (0, 1]$), we have

$$-\ln k \leq \frac{1-\beta^2}{2\beta} \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) - (1-\beta) \sum_{i=1}^{L} \sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right].$$

Rearranging the last inequality we have

$$\begin{aligned} &\sum_{i=1}^{L} \sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right] \\ &\leq \frac{1-\beta^2}{2\beta(1-\beta)} \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}} j) + \frac{1}{1-\beta} \ln k \\ &= \frac{1+\beta}{2\beta} \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \frac{1}{1-\beta} \ln k \\ &= \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \left(\frac{1+\beta}{2\beta} - 1\right) \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \frac{1}{1-\beta} \ln k. \end{aligned}$$

We recall here that $\beta = \frac{1}{1-\sqrt{(2\log k)/L}}$. So $\frac{1}{1-\beta} \ln k = \frac{\sqrt{2L \ln k}}{2} + \ln k$ and $\frac{1-\beta}{2\beta} L + \frac{\sqrt{2L \ln k}}{2} = \sqrt{2L \ln k}$, which imply

$$\begin{aligned} &\sum_{i=1}^{L} \sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right] \\ &\leq \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \left(\frac{1+\beta}{2\beta} - 1\right) \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \frac{1}{1-\beta} \ln k \\ &= \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \left(\frac{1+\beta}{2\beta} - 1\right) \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \frac{\sqrt{2L \ln k}}{2} + \ln k \\ &\leq \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \frac{1-\beta}{2\beta} L + \frac{\sqrt{2L \ln k}}{2} + \ln k \\ &= \sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) + \sqrt{2L \ln k} + \ln k. \end{aligned}$$

Finally, rearranging the inequality and dividing by L we have the result. For the reader convenience, we also recall here that, by definition,

$$\tilde{\mu}(\mathsf{Ext}_i, \mathcal{D}^{(i)}) = \sum_{b=1}^{k} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right].$$

□

Lemma 9. *For each $\mathcal{A}_{\mathsf{kss}_j} \in \{\mathcal{A}_{\mathsf{kss}_1}, \mathcal{A}_{\mathsf{kss}_2}, \dots, \mathcal{A}_{\mathsf{kss}_k}\}$, with probability $1 - \mathcal{O}(\epsilon(k))$ over the random coins of the extractor, the extractor defined in Fig. 9 generates $\mathcal{D}^{(1)}, \dots, \mathcal{D}^{(L)}$ and $\mathsf{Ext}_1, \dots, \mathsf{Ext}_L$ such that*

$$\frac{1}{L}\sum_{i=1}^{L} \mu(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) \geq \frac{1}{L}\sum_{i=1}^{L} \mu(\mathsf{Ext}_i, \mathcal{D}^{(i)}) - \mathcal{O}(\epsilon'(k)).$$

Proof. As done in the previous lemma, fix a pair $(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}$, and fix the random tape of the extractor. in this way, all the random variables that appears in Fig. 9, become fixed. We begin to show that, $|\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}) - \mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})| \leq \mathcal{O}(\epsilon'(k))$ with probability $1 - \mathcal{O}(\frac{\epsilon'(k)}{kL})$, for each extractor Ext and each $\mathcal{A}_{\mathsf{kss}} \in \{\mathcal{A}_{\mathsf{kss}}1, \dots, \mathcal{A}_{\mathsf{kss}_k}\}$. Let X denotes the random variable that counts the number of success of the extractor Ext, when one compute $\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})$, as prescribed in Fig. 10. That is $X = freq$, where $freq$ is the variable defined in Fig. 10. Formally, $X = \gamma\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})$. Note that the expected value of X is $\mathbb{E}(X) = \gamma\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})$. Now,

$$\begin{aligned}
&\Pr\left[|\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}) - \mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})| \geq \epsilon'(k)\right] \\
&= \Pr\left[|X - \gamma\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})| \geq \epsilon'(k)\gamma\right] \\
&\leq \Pr\left[|X - \gamma\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})| \geq \epsilon'(k)\gamma\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})\right] \\
&= \Pr\left[|X - \mathbb{E}(X)| \geq \epsilon'(k)\mathbb{E}(X)\right].
\end{aligned}$$

We recall here the multiplicative form of Chernoff bound for a random variable X. For each $\delta > 0$, it holds that

$$\Pr\left[|X - \mathbb{E}(X)| \geq \delta\mathbb{E}(X)\right] \leq 2e^{-(\delta^2\mathbb{E}(X))/3}.$$

We also recall that $\gamma = \Theta(\frac{\log(kL/\epsilon'(k))}{\epsilon'(k)^2})$. Applying the Chernoff bound to the last term of the inequality above, we have

$$\begin{aligned}
\Pr\left[|\tilde{\mu}(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}) - \mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})| \geq \epsilon'(k)\right] &\leq 2e^{-(\epsilon(n)^2\mathbb{E}(X))/3} \\
&= 2e^{-(\epsilon(k)^2\gamma\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}))/3} \\
&= 2\left(\frac{kL}{\epsilon'(k)}\right)^{(-C\mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}}))} \\
&= \mathcal{O}(\frac{\epsilon'(k)}{kL}),
\end{aligned}$$

where C is a positive constant.

By the union bound, we have

$$|\tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}}) - \mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}})| \leq \mathcal{O}(\epsilon'(k)), \tag{6}$$

for each $i \in \{1, \dots, L\}$, with probability at least $1 - kL\mathcal{O}(\frac{\epsilon'(k)}{kL}) = 1 - \mathcal{O}(\epsilon'(k))$.

Finally, conditioned on the previous event, we have

$$\begin{aligned}
&\frac{1}{L}\sum_{i=1}^{L} \mu(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) \\
&\geq \frac{1}{L}\sum_{i=1}^{L} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) - \mathcal{O}(\epsilon'(k)) \\
&\geq \frac{1}{L}\sum_{i=1}^{L}\sum_{k=1}^{n} \tilde{\mu}(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_k}) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_k}\right] - \mathcal{O}(\epsilon'(k)) \\
&\geq \frac{1}{L}\sum_{i=1}^{L}\sum_{b=1}^{k} (\mu(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_b}) - \mathcal{O}(\epsilon'(k))) \Pr\left[\mathcal{D}^{(i)} = \mathcal{A}_{\mathsf{kss}_b}\right] - \mathcal{O}(\epsilon'(k)) \\
&= \frac{1}{L}\sum_{i=1}^{L} \mu(\mathsf{Ext}_i, \mathcal{D}^{(i)}) - \mathcal{O}(\epsilon'(k)).
\end{aligned}$$

Here the second inequality holds by Lemma 8 and the other inequalities follows by Eq. (6). □

Now, we show that, for each $i \in \{1, \dots, L\}$, Ext_i is a good extractor against $\mathcal{D}^{(i)}$, that is $\mathbb{E}_{(\mathtt{crs},\mathtt{td})\sim\mathsf{CRSGen}(1^k)}\left[\mu(\mathsf{Ext}_i, \mathcal{D}^{(i)})\right] \geq \mathcal{O}(\epsilon'(k))$ for each i. Consider,

$$\begin{aligned}
\mu(\mathsf{Ext}_i, \mathcal{D}^{(i)}) &= \sum_{j=1}^{k} \mathcal{D}^{(i)}(\mathcal{A}_{\mathsf{kss}_j}) \cdot \mu(\mathsf{Ext}, \mathcal{A}_{\mathsf{kss}_j}) \\
&= \sum_{j=1}^{k} \mathcal{D}^{(i)}(\mathcal{A}_{\mathsf{kss}_j}) \cdot \Pr\left[\begin{array}{c} \mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 \\ \wedge \\ (\mathtt{x}, \mathtt{w}) \in \mathcal{R} \end{array}\middle| \begin{array}{c} r \leftarrow D_j, \\ (\mathtt{x}, \pi) = f_j(\mathtt{crs}; r), \\ \mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi) \end{array}\right]. \\
&= \mu(\mathsf{Ext}_i, \mathcal{A}_{U_i}) = \mu(\mathsf{Ext}_{\mathcal{A}_{U_i}}, \mathcal{A}_{U_i})
\end{aligned}$$

The second equality is given by the definition of $\mathcal{A}_{U_i}$, and the third inequality follows from the definition of Ext_i. Let f be the kss implemented by $\mathcal{A}_{U_i}$. Now, since $\mathsf{Ext}_{\mathcal{A}_{U_i}}$ is a good extractor for f, by the (t', ϵ') semantic black-box extraction, we have that

$$\begin{aligned}
&\mathop{\mathbb{E}}_{(\mathtt{crs},\mathtt{td})\sim\mathsf{CRSGen}(1^k)}\left[\mu(\mathsf{Ext}_i, \mathcal{D}^{(i)})\right] = \mathop{\mathbb{E}}_{(\mathtt{crs},\mathtt{td})\sim\mathsf{CRSGen}(1^k)}\left[\mu(\mathsf{Ext}_{\mathcal{A}_{U_i}}, \mathcal{A}_{U_i})\right] \\
&= \Pr\left[\begin{array}{c} \mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 \\ \wedge(\mathtt{x}, \mathtt{w}) \in \mathcal{R} \end{array}\middle| \begin{array}{c} (\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k); s \leftarrow D \\ (\mathtt{x}, \pi) \leftarrow \mathcal{A}_{U_i}(\mathtt{crs}; s); \mathtt{w} \leftarrow \mathsf{Ext}_{\mathcal{A}_{U_i}}(\mathtt{td}, \mathtt{x}, \pi) \end{array}\right] \\
&\geq 1 - \epsilon'(k)
\end{aligned} \tag{7}$$

From Lemma 9, with probability $1-\mathcal{O}(\epsilon'(k))$, the generated $\{\mathsf{Ext}_1, \ldots, \mathsf{Ext}_L\}$ are such that

$$\frac{1}{L}\sum_{i=1}^{L}\mu(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j}) \geq \frac{1}{L}\sum_{i=1}^{L}\mu(\mathsf{Ext}_i, \mathcal{D}^{(i)}) - \mathcal{O}(\epsilon'(k))$$

$$\frac{1}{L}\mathop{\mathbb{E}}_{(\mathtt{crs},\mathtt{td})}\left[\sum_{i=1}^{L}\mu(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}_j})\right] \geq \frac{1}{L}\mathop{\mathbb{E}}_{(\mathtt{crs},\mathtt{td})}\left[\sum_{i=1}^{L}\mu(\mathsf{Ext}_i, \mathcal{D}^{(i)})\right] - \mathcal{O}(\epsilon'(k)) \qquad (8)$$

From Eq. (7) and Eq. (8),

$$\frac{1}{L}\sum_{i=1}^{L}\mathop{\mathbb{E}}_{(\mathtt{crs},\mathtt{td})}[\mu(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}})] \geq 1 - \mathcal{O}(\epsilon'(k)).$$

Finally, since $\mathsf{Ext}_{\mathsf{BB}}$ generates the set of $\{\mathsf{Ext}_i\}$, and fails only if all of them fail, we have

$$\mathop{\mathbb{E}}_{(\mathtt{crs},\mathtt{td})}[\mu(\mathsf{Ext}_{\mathsf{BB}}, \mathcal{A}_{\mathsf{kss}})] \geq \frac{1}{L}\sum_{i=1}^{L}\mathop{\mathbb{E}}_{(\mathtt{crs},\mathtt{td})}[\mu(\mathsf{Ext}_i, \mathcal{A}_{\mathsf{kss}})] \geq 1 - \mathcal{O}(\epsilon'(k)).$$

□

Unfortunately, the extractor depicted in Fig. 9 is not guaranteed to be polynomial time since it is not efficient to find the distribution-dependent extractor Ext_i. Using an auxiliary input to encode the distribution is an idea used in prior works like [24], however, since we are interested in a BB extractor, we cannot allow the extractor to read auxiliary inputs. Instead, we interpret the universal KSS – that takes an auxiliary input, a string encoding each distribution $\mathcal{D}^{(i)}$, samples a distribution, then samples a KSS as per that distribution – also as a knowledge soundness strategy. We then show that invoking the extractor corresponding to this universal KSS works well against distribution dependent $\mathcal{A}_{U_i}$. Note that such a $\mathcal{A}_U$ is indeed polynomial time: Each distribution $\mathcal{D}$ computed in the "for" loop of Fig. 9 can be represented as a weight vector $\mathfrak{w}_{\mathcal{D}}$ of polynomial length.

Let $\mathsf{Ext}_{\mathcal{A}_U}$ be a (t, ϵ^2) extractor against the knowledge soundness strategy $\mathcal{A}_U$, where $\mathcal{A}_U$ is as defined above. We show in the following lemma that $\mathsf{Ext}_{\mathcal{A}_U}$ is a good approximation of an extractor for any distribution dependent $\mathcal{A}_{U_i}$, with probability greater than $1-\mathcal{O}(\epsilon)$. Thus, we can define the efficient universal black-box extractor as follows: run L independent executions of $\mathsf{Ext}_{\mathcal{A}_U}$ and output a valid witness if at least one of the executions succeeds. The used $\mathsf{Ext}_{\mathcal{A}_U}$ has to be a (t, ϵ) with ϵ much better than ϵ'^2.

Lemma 10. *Let N be the number of times that any semantic extractor is called in the procedure defined in Fig. 9. Let L be defined as in Fig. 9. Let $\mathsf{Ext}_{\mathcal{A}_U}$ be the (t, ϵ) semantic black-box extractor against $\mathcal{A}_U$, where $\epsilon = \mathcal{O}(\epsilon'^2/N)$. Then the procedure defined by running L independent executions of $\mathsf{Ext}_{\mathcal{A}_U}$ is a black-box (Lt, ϵ') extractor, for every inverse polynomial ϵ, ϵ'.*

Proof. Let Y be the conditional expectation of the failure of $\mathsf{Ext}_{\mathcal{A}_U}$ against $\mathcal{A}_U$, given a fixed distribution $\mathfrak{w}_{\mathcal{D}}$ over $\mathcal{A}_{\mathsf{kss}}$-es. Formally we have

$$\begin{aligned} Y(\mathfrak{W}) &= \mathbb{E}\left[1-\mu(\mathsf{Ext}_{\mathcal{A}_U},\mathcal{A}_U)|\left[\mathfrak{W}=\mathfrak{w}\right]\right] \\ &= \Pr\left[\begin{array}{c|c} \mathcal{V}(\mathtt{crs},\mathtt{x},\pi)=1 & (\chi,r)\leftarrow D \\ \wedge(\mathtt{x},\mathtt{w})\notin\mathcal{R} & (\mathtt{x},\pi)\leftarrow\mathcal{A}_U(\mathtt{crs};(\mathfrak{w}||\chi||r));\mathtt{w}\leftarrow\mathsf{Ext}_{\mathcal{A}_U}(\mathtt{td},\mathtt{x},\pi) \end{array}\right]. \end{aligned}$$

Note that $\mathbb{E}\left[Y\right]=\epsilon(k)$, by definition. We now recall Markov inequality. For each non-negative random variable X that admits expected value $\mathbb{E}\left[X\right]$, for each value α it holds that

$$\Pr\left[X\geq\alpha\right]\leq\frac{\mathbb{E}\left[X\right]}{\alpha}.$$

Applying the inequality to Y we get

$$\Pr\left[Y\geq\epsilon'(k)\right]\leq\frac{\epsilon(k)}{\epsilon'(k)}=\mathcal{O}(\epsilon'(k)/N(k)).$$

Note that, for each of the k $\mathcal{A}_{\mathsf{kss}}$, there exists $\mathfrak{w}$ such that

$$\begin{aligned} &\Pr\left[\begin{array}{c|c} \mathcal{V}(\mathtt{crs},\mathtt{x},\pi)=1 & r\leftarrow D \\ \wedge(\mathtt{x},\mathtt{w})\notin\mathcal{R} & (\mathtt{x},\pi)\leftarrow\mathcal{A}_{\mathsf{kss}}(\mathtt{crs};r);\mathtt{w}\leftarrow\mathsf{Ext}_{\mathcal{A}_U}(\mathtt{td},\mathtt{x},\pi) \end{array}\right] \\ =&\Pr\left[\begin{array}{c|c} \mathcal{V}(\mathtt{crs},\mathtt{x},\pi)=1 & (\chi,r)\leftarrow D \\ \wedge(\mathtt{x},\mathtt{w})\notin\mathcal{R} & (\mathtt{x},\pi)\leftarrow\mathcal{A}_U(\mathtt{crs};(\mathfrak{w}||\chi||r));\mathtt{w}\leftarrow\mathsf{Ext}_{\mathcal{A}_U}(\mathtt{td},\mathtt{x},\pi) \end{array}\right] \\ =&\,Y(\mathfrak{w}). \end{aligned}$$

Here $\mathfrak{w}$ is the distribution that puts a weight of 1 on $\mathcal{A}_{\mathsf{kss}}$. Moreover, for each distributions $\mathcal{D}^{(i)}$, represented by vector of weights $\mathfrak{w}_i$ and the corresponding $\mathcal{A}_{U_i}$, we have

$$\begin{aligned} &\Pr\left[\begin{array}{c|c} \mathcal{V}(\mathtt{crs},\mathtt{x},\pi)=1 & (\chi||r)\leftarrow D \\ \wedge(\mathtt{x},\mathtt{w})\notin\mathcal{R} & (\mathtt{x},\pi)\leftarrow\mathcal{A}_{U_i}(\mathtt{crs};(\chi||r));\mathtt{w}\leftarrow\mathsf{Ext}_{\mathcal{A}_U}(\mathtt{td},\mathtt{x},\pi) \end{array}\right] \\ =&\Pr\left[\begin{array}{c|c} \mathcal{V}(\mathtt{crs},\mathtt{x},\pi)=1 & (\chi,r)\leftarrow D \\ \wedge(\mathtt{x},\mathtt{w})\notin\mathcal{R} & (\mathtt{x},\pi)\leftarrow\mathcal{A}_U(\mathtt{crs};(\mathfrak{w}_i||\chi||r));\mathtt{w}\leftarrow\mathsf{Ext}_{\mathcal{A}_U}(\mathtt{td},\mathtt{x},\pi) \end{array}\right] \\ =&\,Y(\mathfrak{w}_i). \end{aligned}$$

Suppose now, we define a universal extractor as the one defined in Fig. 9, except, we run $\mathcal{A}_U$ every time an extractor is called in the procedure. This new universal extractor is a good approximation of the one in Fig. 9, as long as the distribution of $Y(\mathfrak{W})$ is sufficiently dense around its average. Indeed, using Markov inequality, we show how to choose ϵ as a function of ϵ' so that each time we use $\mathsf{Ext}_{\mathcal{A}_U}$ instead of any other extractor in the proof of Theorem 4, with overwhelming probability, we have an average loss of $\mathcal{O}(\epsilon'(k)/N(k))$. Now applying the union bound we have the result. The resulting BB extractor runs in time Lt.

□

Semantic and white-box extraction. We now state the restricted semantic knowledge soundness definition for which the equivalence to white-box knowledge soundness holds. We consider knowledge soundness strategies f such that $f : \mathbf{CRS} \times \mathbf{\Gamma_t} \to \chi \times \Psi$ and $\mathbf{\Gamma_s}$ is the set that contains only the empty string.

Definition 18. *A NIZK argument $\Pi = (\mathsf{CRSGen}, \mathcal{P}, \mathcal{V}, \mathsf{Sim})$ is semantic white-box knowledge sound (semn-BB) if for each efficiently implementable knowledge soundness strategy f, there exists a PPT extractor $\mathsf{Ext} = \mathsf{Ext}_f$, such that, for each (even unbounded) TM $\mathcal{A}^*$ that implements f*

$$\Pr\left[\begin{array}{c}\mathcal{V}(\mathtt{crs}, \mathtt{x}, \pi) = 1 \\ \wedge(\mathtt{x}, \mathtt{w}) \notin \mathcal{R}\end{array}\middle|\begin{array}{c}(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{CRSGen}(1^k); r \leftarrow D \\ (\mathtt{x}, \pi) \leftarrow \mathcal{A}^*(\mathtt{crs}; r); \mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi, r)\end{array}\right] \leq \mathsf{negl}(k).$$

Theorem 5. *A NIZK argument is white-box knowledge sound (semn-BB) as in Definition 2 if and only if it is also semantic knowledge sound as in Definition 18.*

Proof. Any semantic extractor that satisfies Definition 18, is also, by definition, a white-box extractor for each PPT that implements a certain function. So "only if" side is trivial. To show the other direction, suppose that there exists two efficient PPT machines $\mathcal{A}, \mathcal{A}'$ that implement the same function f. Let $\mathsf{Ext}, \mathsf{Ext}'$ be the corresponding white-box extractors as in Definition 2. We show that Ext is a semantic extractor for $\mathcal{A}'$.

Consider the set of tuples $(\mathtt{crs}, \mathtt{x}, \pi)$ such that there exists r for which $(\mathtt{x}, \pi) = f(\mathtt{crs}, r)$. We can divide this set into two disjoint subset. The first subset is defined by the tuples such that $\mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi, r), \mathtt{w}' \leftarrow \mathsf{Ext}'(\mathtt{td}, \mathtt{x}, \pi, r)$ and $(\mathtt{x}, \mathtt{w}) \in \mathcal{R}, (\mathtt{x}, \mathtt{w}') \in \mathcal{R}$ with overwhelming probability. Given the tuple belongs to this set, then Ext will also be a good extractor for $\mathcal{A}'$, although in the general case it can return a different (but still valid) witness with respect to Ext'.

We now consider the set of tuples $(\mathtt{crs}, \mathtt{x}, \pi)$ such that at least one extractor fails with non-negligible probability. Consider the subset of tuples such that $\mathtt{w} \leftarrow \mathsf{Ext}(\mathtt{td}, \mathtt{x}, \pi, r), \mathtt{w}' \leftarrow \mathsf{Ext}'(\mathtt{td}, \mathtt{x}, \pi, r)$ and $(\mathtt{x}, \mathtt{w}) \notin \mathcal{R}, (\mathtt{x}, \mathtt{w}') \in \mathcal{R}$. This set is of negligible size, since it is a subset of the set of tuples for which Ext fails, which is negligible dy definition.

Thus, Ext is a good semantic extractor for each PPT that implements the function f. It only remains to show that Ext is a good extractor even against unbounded TMs that implement f. This is true since the set in which Ext fails is of negligible size. So, let $\mathcal{A}^*$ be an unbounded TM that implements f. Clearly, Ext is a good extractor for $\mathcal{A}^*$, since $\mathcal{A}(\mathtt{crs}, r) = \mathcal{A}^*(\mathtt{crs}, r)$ for each $(\mathtt{crs}, r)$. It is also a good semantic extractor for $\mathcal{A}^*$, since the set of $(\mathtt{crs}, \mathtt{x}, \pi)$ tuples such that Ext fails on $\mathcal{A}^*$ but not for $\mathcal{A}$ is the empty set.

D.3 Impossibility of Semantic Knowledge Soundness for CH-NIZK

Theorem 6 (Theorem 2 restated). *Let $\mathcal{L}_{\mathtt{lpar}}$ be any 1DL-friendly algebraic language with $\mathtt{lpar} = (\mathbf{M}, \boldsymbol{\theta})$. The NIZK argument in Fig. 2 cannot be semantic knowledge sound for $\mathcal{L}_{\mathtt{lpar}}$ under the SDL assumption.*

Proof. The proof is similar to the proof of Lemma 2. Fix a language with the properties mentioned in the statement; that is, fix suitable $\mathbf{M}, \boldsymbol{\theta}$. Suppose that the relative NIZK argument is semantic knowledge sound. Define the canonical prover adversary, on input $\mathtt{crs} = [e]_2$ and randomness $(s, \mathbf{r}, r')$, in the following way:

1. Sample uniformly random w_1 using seed s.
2. Using random coins r' sample all the other integer, $w_2, \ldots, w_d$ and define $\mathbf{w} = \lambda_w(w_1, \ldots, w_n)$.
3. Compute $\mathbf{x} = \lambda_x(w_1, \ldots, w_n)$.
4. Using random coins $\mathbf{r}$ compute $\mathbf{a}$, $[\mathbf{d}]_2$ as prescribed by the honest prover.
5. Output $([\mathbf{x}]_1, \pi = ([\mathbf{a}]_1, [\mathbf{d}]_2))$.

Let Ext_f be the semantic extractor defined for the canonical adversary. We can exploit it to define an adversary $\mathcal{A}$ for the SDL assumption. On input an SDL challenge $([w_1]_1, [w_1]_2)$, $\mathcal{A}$ do the following.

1. Sample $e, \mathbf{r}, r'$.
2. Using random coins r' sample all the other integer, $w_2, ...w_d$ and define $[\mathbf{w}]_2 = \lambda_w([w_1]_2, w_2 \ldots, w_n)$. Recall that this is efficiently computable since λ_w is linear in w_1.
3. Compute $[\mathbf{x}]_1 = \lambda_x([w_1]_1, w_2 \ldots, w_n)$. Recall that this is efficiently computable since λ_x is linear in w_1.
4. Compute $[\mathbf{a}]_1$ as $\mathbf{r}[\mathbf{M}(\mathbf{x})]_1$.
5. Compute $[\mathbf{d}]_2$ as $e[\lambda(\mathbf{w})]_2 + [\mathbf{r}]_2$.
6. Compute $\mathbf{w} \leftarrow \mathsf{Ext}_f(([\mathbf{a}]_1, [\mathbf{d}]_2), e, [\mathbf{x}]_1, (\mathbf{r}, r'))$.
7. Output w_1.

Since $\mathcal{A}$ computes the same function as an unbounded prover that is able to recover e from $[e]_2$, inputs provided to the extractor are correctly distributed. Thus, $\mathcal{A}$ computes discrete logarithm w_1 with the same probability that Ext_f is successful, breaking the SDL assumption.

D.4 Impossibility of Semantic Extractability for SPHF-based QA-NIZKs

Quasi-Adaptive NIZK (QA-NIZK) proofs are NIZK proofs where the CRS can depend on some parameters $\mathtt{lpar}$ of the language for which proofs have to be generated [45]. The language dependent preprocessing improves efficiency and leads to succinct proofs which have size as short as a single group element [45]. QA-NIZK arguments are not arguments of knowledge in general. While in [22] it has been shown that QA-NIZK arguments can satisfy this property in the generic/algebraic group model, showing this property in the case of black-box extraction where the extractor can extract a witness from the prover using only its input/output interface seems counterintuitive as the proof size is shorter than the witness. In this section, we prove this intuition to be correct by giving a stronger impossibility result which shows SPHF-based QA-NIZKs with semantic knowledge soundness cannot exist. More precisely, we consider the most efficient QA-NIZKs Π_{kw} by Kiltz and Wee [46] and show that it cannot be semantic knowledge sound.

```
$\mathsf{CRSGen}(\mathtt{lpar} = [\mathbf{M}]_1 \in \mathbb{G}_1^{\ell \times k})$
    $\mathbf{A} \leftarrow \mathcal{D}_t; \boldsymbol{\alpha} \leftarrow \mathbb{Z}_p^{\ell \times (t+1)}$
    $[\boldsymbol{\gamma}]_1 := [\mathbf{M}^\top \boldsymbol{\alpha}]_1; \mathbf{C} := \boldsymbol{\alpha}\mathbf{A}$
    $\mathtt{crs} := ([\boldsymbol{\gamma}]_1, [\mathbf{C}, \mathbf{A}]_2)$
    return $(\mathtt{crs}, \mathtt{td} = \boldsymbol{\alpha})$

$\mathsf{Sim}(\mathtt{crs}, \mathtt{td} = \boldsymbol{\alpha}, [\mathbf{x}]_1)$
    return $\boldsymbol{\pi} := [\mathbf{x}^\top \boldsymbol{\alpha}]_1$

$\mathcal{P}(\mathtt{crs}, [\mathbf{x}]_1 = [\mathbf{M}]_1 \mathbf{w}, \mathbf{w})$
    return $\boldsymbol{\pi} := ([\mathbf{w}^\top \boldsymbol{\gamma}]_1 \in \mathbb{G}_1^{t+1}$

$\mathcal{V}(\mathtt{crs}, [\mathbf{x}]_1, \boldsymbol{\pi})$
    $\hat{e}([\mathbf{x}^\top]_1, [\mathbf{C}]_2) \overset{?}{=} \hat{e}(\boldsymbol{\pi}, [\mathbf{A}]_2)$
```

Fig. 11: QA-NIZK proof system Π_{kw} under $\mathcal{D}_t$-KerMDH assumption

D.5 Overview of Kiltz-Wee QA-NIZK

The core idea of the NIZK proof system in [46] for linear space membership languages is as follows: starting from a DVS-based SPHF for the language (see Fig. 7), which can be seen as a symmetric-key analogue of NIZK with a designated verifier and then translating it to the bilinear group setting. To be more precise, let $\mathcal{L}_{\mathtt{lpar}}$ with $\mathtt{lpar} = [\mathbf{M}]_1 \in \mathbb{G}_1^{\ell \times k}$ be the linear language defined as

$$\mathcal{L}_{\mathtt{lpar}} = \left\{ [\mathbf{x}]_1 \in \mathbb{G}_1^{\ell} | \exists \mathbf{w} \in \mathbb{Z}_p^k : [\mathbf{x}]_1 = [\mathbf{M}]_1 \cdot \mathbf{w} \right\} \tag{9}$$

A designated-verifier ZK from a DVS-based SPHF (see A.4) for $\mathcal{L}_{\mathtt{lpar}}$ can be constructed as follows: the verifier first selects a key $\boldsymbol{\alpha} \in \mathbb{Z}_p^{\ell \times (t+1)}$, where t depends on the hardness assumption behind the soundness property. Next, the verifier sends $[\mathbf{M}^\top \boldsymbol{\alpha}]_1$ to the prover, who later computes and sends $\boldsymbol{\pi} = \mathbf{w}^\top [\mathbf{M}^\top \boldsymbol{\alpha}]_1$ to the verifier. Finally, the verifier checks if $[\mathbf{x}^\top]_1 \boldsymbol{\alpha} = \boldsymbol{\pi}$. Starting from this construction, Kiltz and Wee make it a publicly-verifiable QA-NIZK proof system in the CRS model by using pairing techniques as follows: the CRS includes $[\mathbf{M}^\top \boldsymbol{\alpha}]_1$ and $[\mathbf{A}, \boldsymbol{\alpha}\mathbf{A}]_2$ for a vector $\mathbf{A} \in \mathbb{Z}_p^{(t+1) \times t}$ chosen from a distribution $\mathcal{D}_t$. The proof remains the same as before, but the verification is the pairing check

$$\hat{e}([\mathbf{x}^\top]_1, [\boldsymbol{\alpha}\mathbf{A}]_2) \overset{?}{=} \hat{e}(\boldsymbol{\pi}, [\mathbf{A}]_2)$$

The soundness relies on the hardness of finding non-trivial cokernel elements of $\mathbf{A}$ and the smoothness of the underlying projective hash function (PHF). Also, for the right choice of the distribution for $\mathbf{A}$, the most efficient choice that the assumption is believed to hold is $t = 1$ which results in succinct proofs consisting of only two group elements. The protocol Π_{kw} is depicted in Fig. 11.

D.6 Impossibility of semantic extractor for Kiltz-Wee QA-NIZK.

We now prove that Π_{kw} cannot be semantic knowledge sound under the discrete logarithm assumption.

Theorem 7. *Let $\mathcal{L}_{\texttt{lpar}}$ be a linear language over some cyclic group $\mathbb{G}_1$ with $\texttt{lpar} = [\mathbf{M}]_1$. The QA-NIZK Π_{kw} depicted in Fig. 11 cannot be semantic knowledge sound under the DL assumption in $\mathbb{G}_1$.*

Proof. The proof is very similar to the proof of Lemma 2. Suppose Π_{kw} is semantic knowledge sound. Let $\mathcal{P}$ be the canonical prover adversary that on input a CRS $\texttt{crs} = ([\boldsymbol{\gamma}]_1, [\mathbf{C}, \mathbf{A}]_2)$ and random coins s, r' proceeds as follows:

1. Sample the first component of the witness w_1 using random coins s.
2. Sample other components of $\mathbf{w}$ using random coins r'.
3. Compute $\mathbf{x} = [\mathbf{M}]_1\mathbf{w}$.
4. Compute $\boldsymbol{\pi} := ([\mathbf{w}^\top\boldsymbol{\gamma}]_1 \in \mathbb{G}_1^{t+1}$.
5. Return $([\mathbf{x}]_1, \boldsymbol{\pi})$.

Let f be the function of honest prover strategy that $\mathcal{P}$ uses to compute a valid proof. Namely, $f(\texttt{crs}, (s, r')) = ([\mathbf{x}]_1, \boldsymbol{\pi})$. The semantic extractor $\mathsf{Ext}_{\mathcal{P}}$ given as input $([\mathbf{x}]_1, \boldsymbol{\pi}, \texttt{td} = \boldsymbol{\alpha}; r)$ can output $\mathbf{w}$ with overwhelming probability. Note that $r = \perp$ as the prover is deterministic. A DL adversary $\mathcal{A}$ can now use this extractor to break the DL assumption. Specifically, $\mathcal{A}$, on a DL challenge $[\varrho]_1$ proceeds as follows:

1. Samples a group element $[\mathbf{x}]_1$ using randomness r' such that the first element of $[\mathbf{x}]_1$ is defined as $[x_1]_1 = [\varrho]_1$.
2. Selects a trapdoor $\texttt{td} = \boldsymbol{\alpha}$ and computes an accepting proof $\boldsymbol{\pi} = [\mathbf{x}^\top]_1\boldsymbol{\alpha}$.
3. Invokes the extractor on $[\mathbf{x}]_1, \boldsymbol{\pi}$ who outputs $\mathbf{w}$. Return the first element w_1 of $\mathbf{w}$.

Now observe that $\mathcal{A}$ computes inputs of Ext_f exactly as an inefficient prover $\mathcal{P}^*$ for which the extraction is guaranteed. Hence, $\mathcal{A}$ computes the discrete logarithm $\varrho = w_1$ of $[\varrho]_1$ with the same probability that Ext_f succeeds.

□

Chapter 6

Set (Non-)Membership NIZKs from Determinantal Accumulators

Helger Lipmaa
Roberto Parisella

Cryptology ePrint Archive 2022, Paper Report 2022/1570, `https://eprint.iacr.org/2022/1570`

Currently on ePrint. Paper under submission.

Set (Non-)Membership NIZKs from Determinantal Accumulators

Helger Lipmaa and Roberto Parisella

Simula UiB, Bergen, Norway

Abstract. We construct the most efficient (in the argument size and the verifier's computation) known falsifiable set (non-)membership NIZK $\mathbf{\Pi}^*$, where the membership (resp., non-membership) argument consists of only 9 (resp., 15) group elements. It also has a universal CRS. $\mathbf{\Pi}^*$ is based on the novel concept of determinantal accumulators. Determinantal primitives have a similar relation to recent pairing-based (non-succinct) NIZKs of Couteau and Hartmann (Crypto 2020) and Couteau et al. (CLPØ, Asiacrypt 2021) that structure-preserving primitives have to the NIZKs of Groth and Sahai. $\mathbf{\Pi}^*$ is considerably more efficient than known falsifiable based set (non-)membership NIZKs. We also extend CLPØ by proposing efficient (non-succinct) set *non*-membership arguments for a large class of languages.

Keywords: Commit-and-prove · non-interactive zero-knowledge · set (non-)membership argument · universal accumulator

1 Introduction

In a set (non-)membership NIZK, the prover aims to convince the verifier that an encrypted element χ belongs (does not belong) to a public set $\mathcal{S}$. Fully succinct (constant size and constant-time verifiable) set (non-)membership NIZKs have many applications. Classical applications include anonymous credentials (one has to prove that one has a valid credit card), governmental whitelist (to prevent money laundering), and e-voting (one has to prove that one is an eligible voter). A non-membership NIZK can be used to prove that a key is *not* blacklisted. Set membership NIZKs are instrumental in ring signatures. Recently, set (non-)membership NIZKs have gained popularity in cryptocurrencies. For example, in Zcash, to validate a transaction that intends to spend a coin χ requires one to check that χ is in the set UTXO (unspent transaction outputs).

When χ is public, one can use an efficient (universal) accumulator [BdM93] for this task. A universal accumulator can be reframed as a set (non-)membership *non-zk* non-interactive argument system. Accumulator's completeness and collision-resistance (see Section 2) correspond directly to the completeness and soundness of the set (non-)membership argument system but with public input. To construct a set (non-)membership NIZK, one only needs to add a zero-knowledge (ZK) compiler to the accumulator. Unfortunately, the ZK compiler is quite complicated in existing constructions, resulting in set (non-)membership NIZKs that are either not falsifiable or not sufficiently efficient.

Related Work. Many set membership NIZKs use either signature schemes or accumulators. In a signature-based set membership NIZKs, the CRS includes signatures of all set elements. The prover proves it knows an (encrypted) signature on the (encrypted) χ. Such NIZKs have several undesirable properties. First, their CRS is non-universal[1] (i.e., it depends on the set). A universal CRS is important in practice since it allows one to rely on a single CRS to construct set (non-)membership NIZKs for different sets. Second, assuming that $|\mathcal{S}|$ is polynomial (and the complement of $\mathcal{S}$ has exponential size), it seems to disallow the construction of set non-membership arguments explicitly.

We will concentrate on accumulator-based constructions since they do not have these two problems. Recall briefly that a (CRS-model) universal accumulator enables one, given a CRS $\texttt{crs}$, to construct a succinct (non-hiding) commitment $\mathsf{C}_{\mathcal{S}}$ of the set $\mathcal{S}$, such that one can efficiently verify whether $\chi \in \mathcal{S}$, given $\texttt{crs}$, $\mathsf{C}_{\mathcal{S}}$, χ, and a succinct accumulator argument ψ of (non-)membership.

In a typical accumulator-based set membership NIZK, the CRS contains *set-independent* elements that are sufficient to compute the accumulator arguments of (non-)membership. (This depends on the underlying accumulator, but importantly, the efficient Nguyen accumulator [Ngu05] allows for that.) Hence, their CRS is universal. Moreover, since there is no need to add all accumulator arguments to the CRS, one can at least hope to construct efficient accumulator-based set *non-membership* NIZKs.

Next, we will summarize the published falsifiable set-membership NIZKs.[2] In all cases $\mathcal{S} \subset \mathbb{Z}_p$ and hence $\chi \in \mathbb{Z}_p$. Since the previous papers have not written down all efficiency numbers, our efficiency comparison (see Table 1) is not completely precise.

Belenkiy et al. (BCKL, [BCKL08]) construct a set-membership NIZK by first building a P-signature scheme [BCKL08]. They prove that a commitment opens to an element for which the prover knows a signature, using a Groth-Sahai NIZK [GS08]. Daza et al. (DGPRS-GS, [DGP$^+$19]) use the more efficient weak Boneh-Boyen (WBB) signature scheme instead of the P-signature scheme. Since the WBB signature scheme is not F-unforgeable [BCKL08], Daza et al. modify it slightly. However, using signature schemes means that the CRS of BCKL and DGPRS-GS is non-universal. In addition, Daza et al. [DGP$^+$19] propose a succinct set membership QA-NIZK. However, their verifier's computation is $O(|\mathcal{S}|)$; thus, it is not suitable in our applications.

Acar and Nguyen (AN, [AN11]) replace the signature scheme with the Nguyen accumulator [Ngu05] and then use Groth-Sahai to prove that the prover knows

[1] We follow the previous literature by using "universal " in the definition of universal accumulators (that have a non-membership argument) and universal CRS (that does not depend on the language).

[2] There are many non-falsifiable or random-oracle-based NIZKs (see, e.g., [CCs08,BCF$^+$21]); we do not compete with them, and thus we omit any discussion.

Table 1. Comparison of known fully succinct falsifiable set (non-)membership arguments for univariate sets of size $|\mathcal{S}| \leq N$. Here, $\mathfrak{g}_\iota$ denotes the bit-length of an element of $\mathbb{G}_\iota$, $\mathfrak{m}_\iota$ denotes the cost of a scalar multiplication in $\mathbb{G}_\iota$, $\mathfrak{m}$ denotes the cost of a scalar multiplication in either $\mathbb{G}_1$ or $\mathbb{G}_2$, and $\mathfrak{p}$ denotes the costs of a pairing. The numbers with $*$ are based on our estimation when the original paper did not give enough data. We give online prover's computation, i.e., assuming precomputation.

Paper	Belenkiy et al. [BCKL08]	Acar-Nguyen [AN11]	Daza et al. [DGP+19]	This work (Fig. 9)
		Building blocks		
Primitive	P-signature	Nguyen acc.	WBB signature	determinantal acc.
NIZK	Groth-Sahai	Groth-Sahai	Groth-Sahai	CLPØ
		Structural properties		
Universal CRS?	✗	✓	✗	✓
Updatable CRS?	✗	✗	✗	✓
Non-membership?	✗	✓	✗	✓
		Membership argument efficiency		
$\lvert\texttt{crs}\rvert$	$(2N+1)\mathfrak{g}_1 + (N+1)\mathfrak{g}_2$	$(N+5)\mathfrak{g}_1 + 4\mathfrak{g}_2{}^*$	$5\mathfrak{g}_1 + (N+5)\mathfrak{g}_2{}^*$	$(N+1)\mathfrak{g}_1 + 4\mathfrak{g}_2$
$\lvert\pi\rvert$	$18\mathfrak{g}_1 + 16\mathfrak{g}_2$	$8\mathfrak{g}_1 + 10\mathfrak{g}_2{}^*$	$10\mathfrak{g}_1 + 8\mathfrak{g}_2{}^*$	$6\mathfrak{g}_1 + 3\mathfrak{g}_2$
P computation	$34\mathfrak{m}$	$16\mathfrak{m}_1 + 16\mathfrak{m}_2{}^*$	$17\mathfrak{m}_1 + 18\mathfrak{m}_2{}^*$	$8\mathfrak{m}_1 + 6\mathfrak{m}_2$
V computation	$68\mathfrak{p}$	$30\mathfrak{p}^*$	$30\mathfrak{p}^*$	$13\mathfrak{p}$
		Non-membership argument efficiency		
$\lvert\texttt{crs}\rvert$	✗	$(N+5)\mathfrak{g}_1 + 4\mathfrak{g}_2{}^*$	✗	$(N+1)\mathfrak{g}_1 + 4\mathfrak{g}_2$
$\lvert\pi\rvert$	✗	$11\mathfrak{g}_1 + 16\mathfrak{g}_2{}^*$	✗	$10\mathfrak{g}_1 + 5\mathfrak{g}_2$
P computation	✗	$26\mathfrak{m}_1 + 28\mathfrak{m}_2{}^*$	✗	$14\mathfrak{m}_1 + 10\mathfrak{m}_2$
V computation	✗	$46\mathfrak{p}^*$	✗	$20\mathfrak{p}$

an accumulator argument. Due to the use of an accumulator, the AN NIZK has a universal CRS; they also propose a set non-membership argument.

BCKL, AN, and DGPRS-GS, and all rely on new (though falsifiable) security assumptions. The central intuition here is that the underlying signature schemes and accumulators are proven to be only secure when the adversary returns χ as an integer. In these NIZKs, χ is essentially encrypted, and the soundness reduction can only recover a group version (say[3], $[\chi]_1$) of χ. The new assumptions (that differ from work to work, see Table 1) guarantee that the adversary cannot break the underlying primitives even if it is allowed only to output $[\chi]_1$.

Structural properties. Another drawback of the signature-based solutions is that it is unclear how to define a universal argument that efficiently allows for non-membership proofs. From the above solutions, only [AN11] (that does not rely on signatures) proposes a set non-membership NIZK.

Efficiency. According to [BCKL08], BCKL's prover performs 34 multi-scalar-multiplications ([BCKL08] does not give separately the number of scalar-multiplications in $\mathbb{G}_1$ and $\mathbb{G}_2$) and the verifer 68 pairings. Neither AN [AN11] Daza et al. [DGP+19] give any efficiency numbers. Hence, the corresponding entries (marked with an asterisk) in Table 1 are based on our estimations.

[3] We use the standard additive bracket notation for pairing-based setting.

Recent NIZKs of Couteau et al. Most of the prior falsifiable set membership NIZKs are based on the Groth-Sahai NIZK [GS08]. Recently, Couteau and Hartmann (CH, [CH20]) proposed a methodology to transform a specific class of Σ-protocols to NIZKs. Intuitively, starting with a Σ-protocol with transcript (a, e, z), CH puts $[\mathsf{e}]_2$ to the CRS and then modifies the computation of z and the verifier's algorithm to work on $[\mathsf{e}]_2$ instead of e. The resulting NIZKs have a CRS consisting of a single group element.

Couteau et al. (CLPØ [CLPØ21]) significantly extended the CH methodology. They constructed efficient commit-and-prove NIZKs for many languages, including (Boolean and arithmetic) Circuit-SAT. Importantly, [CLPØ21] constructed efficient NIZKs for languages that can be described by small algebraic branching programs. The CLPØ NIZK is secure under a new assumption CED (*Computational Extended Determinant*). Depending on the parameters, CED can be either falsifiable or non-falsifiable. For many natural problems like Boolean Circuit-SAT and set membership for poly-sized sets, CED is falsifiable.

Both [CH20,CLPØ21] compare their work to the Groth-Sahai NIZK, showing that in several important use cases, their (falsifiable) NIZKs are more efficient than the Groth-Sahai NIZK. In particular, an important difference between Groth-Sahai and CH/CLPØ is that in the latter, all secret values are only encrypted in $\mathbb{G}_1$. Because of this, the encrypted witness is often three times shorter in CLPØ than in Groth-Sahai; see [CH20,CLPØ21] for examples.

Our first main question is whether one can construct CLPØ-based set (non-)membership NIZKs that are more efficient than the known falsifiable NIZKs [BCKL08,AN11,DGP$^+$19]. Moreover, Groth-Sahai-based NIZKs use specialized primitives (structure-preserving signatures [AFG$^+$16]) that are designed to allow for efficient Groth-Sahai NIZKs. Our second main question is whether one can define a similar class of primitives that allow for efficient CLPØ NIZKs.

1.1 Our Contributions

Summary. Recall that a universal accumulator is a non-zk (non-)membership non-interactive argument system. Thus, one can construct efficient set (non-)membership NIZKs by creating an efficient universal accumulator and then using an efficient ZK compiler to build a NIZK. Our approach is to make the latter part (ZK compiler) as efficient as possible without sacrificing the former part (accumulator) too much.

Differently from the previous work, we will ZK-compile the accumulator to a CLPØ NIZK. We define a *determinantal accumulator* as a universal accumulator with a structure that supports efficient ZK compilation to CLPØ. Determinantal accumulators are related to but different from structure-preserving signatures [AFG$^+$16] that support efficient Groth-Sahai NIZKs. After that, we construct AC^*, an updatable determinantal accumulator with efficient (non-)membership arguments. For this, we follow CLPØ's technique of using algebraic branching programs. Based on AC^*, we then construct $\boldsymbol{\Pi}^*$, a commit-and-prove, updatable set (non-)membership NIZK with a universal CRS.

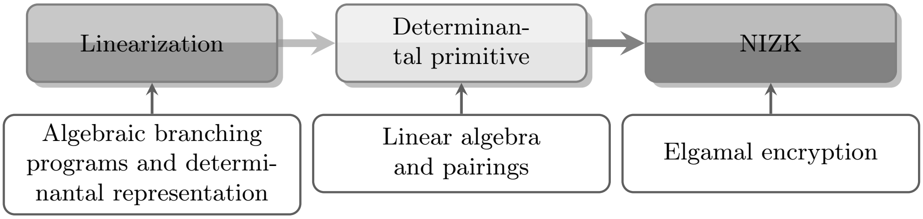

Linearization
Determinantal primitive
NIZK
Algebraic branching programs and determinantal representation
Linear algebra and pairings
Elgamal encryption

Fig. 1. Our general blueprint for constructing efficient falsifiable NIZKs.

We emphasize that this results in a clear, modular framework for constructing efficient falsifiable NIZKs: first, construct an efficient algebraic branching program for the task at hand. Second, construct a determinantal accumulator (or, in general, a non-zk non-interactive argument system). Third, use the efficient CLPØ-inspired ZK compiler to achieve zero knowledge. See Fig. 1 for a high-level diagram of the new approach.

Moreover, we develop a general efficient technique that allows one to construct non-membership NIZKs for a large class of languages where CLPØ only supported membership NIZKs. We use this technique in the case of AC^* and $\boldsymbol{\Pi}^*$, but it potentially has many more applications.

The pairing-based setting is ubiquitous in contemporary public-key cryptography. Any advancement in concrete efficiency in simple problems like set-membership proofs is challenging to come by. Our work demonstrates that in this case, the CH/CLPØ framework gives concretely better results than the seminal Groth-Sahai framework.

Determinantal Accumulators. We assume the standard pairing-based setting (see Section 2). We follow [CLPØ21], but we reinterpret their constructions. First, the verifier has access to input (namely, χ), auxiliary (for example, commitment to $\mathcal{S}$), and output (the accumulator's argument) only in $\mathbb{G}_1$, that is, not as integers. The availability of all private values in $\mathbb{G}_1$ enables us to use an efficient ZK compiler, where only elements of $\mathbb{G}_1$ will be encrypted. (In many pairing-based settings, elements of $\mathbb{G}_2$ are twice longer.) On the other hand, they are not available as integers since the ZK compiler encrypts these values by using Elgamal, and the decryption only returns group elements and not integers.

Second, a determinantal accumulator's verifier checks that the determinants (a potentially high-degree polynomial) of some fixed matrices, whose entries are affine maps, are zero. (On the other hand, in prior falsifiable pairing-based accumulators, the verification equations were pairing-product equations.) This can be seen as a linearization of a polynomial $F(\boldsymbol{X})$ by using affine maps. More precisely, the determinantal accumulator's verifier accepts iff $\det \boldsymbol{C}_i(\boldsymbol{\chi}) = 0$ for DRs $\boldsymbol{C}_i(\boldsymbol{X})$ of some well-chosen polynomials $F_i(\boldsymbol{X})$. Here, a DR (determinantal representation) $\boldsymbol{C}(\boldsymbol{X})$ of $F(\boldsymbol{X})$ is a matrix, where each entry of $\boldsymbol{C}(\boldsymbol{X})$ is an affine map of $\boldsymbol{X}$, and the determinant of $\boldsymbol{C}(\boldsymbol{X})$ is $F(\boldsymbol{X})$.

Since we only need to test that the determinant is zero, we follow the underlying ideas of [CH20,CLPØ21] to make the accumulator efficiently and publicly verifiable. Namely, we use the undergraduate linear algebraic fact that

$\det \boldsymbol{C}(\boldsymbol{X}) = 0$ iff there exists a non-zero vector $\boldsymbol{d}$, such that $\boldsymbol{C}(\boldsymbol{X}) \cdot \boldsymbol{d} = \boldsymbol{0}$. To simplify the construction of accumulators and NIZKs, we follow [CLPØ21] and require that the first coordinate of $\boldsymbol{d}$ is non-zero. Moreover, to achieve both soundness and zero-knowledge in the case of NIZKs, we define $\boldsymbol{d} = \left(\begin{smallmatrix}\mathsf{e}\\ \boldsymbol{\delta}\end{smallmatrix}\right)$ for a new trapdoor $\mathsf{e} \leftarrow_{\$} \mathbb{Z}_p$. (For such e to exist, the matrices $\boldsymbol{C}(\boldsymbol{X})$ need to satisfy an additional requirement, see [CLPØ21].) To achieve zero knowledge, we mask $\boldsymbol{\delta}$ additively with well-chosen randomness. To balance the randomness, we introduce an additional (e-independent) vector $\boldsymbol{\gamma}$ and prove that $\boldsymbol{C}(\boldsymbol{X}) \cdot \left(\begin{smallmatrix}\mathsf{e}\\ \boldsymbol{\delta}\end{smallmatrix}\right) = \boldsymbol{\gamma}$.

Hence, in the implementation of a determinantal accumulator, the prover outputs $[\boldsymbol{\chi}]_1$ (this includes $[\chi]_1$, the candidate element for $\chi \in \mathcal{S}$) and hints $[\boldsymbol{\delta}]_2$ and $[\boldsymbol{\gamma}]_1$. The verifier checks that $[\boldsymbol{C}(\boldsymbol{\chi})]_1 \bullet [\left(\begin{smallmatrix}\mathsf{e}\\ \boldsymbol{\delta}\end{smallmatrix}\right)]_2 = [\boldsymbol{\gamma}]_1 \bullet [1]_2$. (Here, $\boldsymbol{\chi}$ is the vector of concrete values of the indeterminates $\boldsymbol{X}$.) Assuming $\boldsymbol{C}(\boldsymbol{X})$ is small, the verification is constant time.

The definition of determinantal accumulators is an important independent contribution of the current paper. In particular, it is easy to take another primitive (for example, a signature scheme) and define its determinantal variant. This may result in other efficient CLPØ-style NIZKs, but we leave any such generalizations to future work.

New Determinantal Accumulator AC^*. AC^* uses a DR $\boldsymbol{C}(\boldsymbol{X})$ that is motivated by Ngyuen's accumulator [Ngu05]. Define

$$\boldsymbol{C}_\Sigma(\mathsf{X}, \mathsf{Q}) := \begin{bmatrix} \Sigma - \mathsf{X} & -1 \\ -\mathbf{Z}_{\mathcal{S}}(\Sigma) & \mathsf{Q} \end{bmatrix}_1 \quad \text{and} \quad \boldsymbol{C}_\sigma(\chi, \mathsf{q}) = \begin{bmatrix} \sigma - \chi & -1 \\ -\mathbf{Z}_{\mathcal{S}}(\sigma) & \mathsf{q} \end{bmatrix}_1 \enspace .$$

The AC^* verifier accepts a membership argument if $\det \boldsymbol{C}_\sigma(\chi, \mathsf{q}) = 0$ (that is, $(\sigma - \chi)\mathsf{q} = \mathbf{Z}_{\mathcal{S}}(\sigma)$). Here, χ is the statement (a candidate member of $\mathcal{S}$), $[\mathsf{q}]_1$ is given in the membership argument, σ is a CRS trapdoor, and $\mathbf{Z}_{\mathcal{S}}(\Sigma) := \prod_{s \in \mathcal{S}} (\Sigma - s)$ is the vanishing polynomial of $\mathcal{S}$.

In Nguyen's accumulator, given the membership argument $[\mathsf{q}]_2 \in \mathbb{G}_2$, the verifier checks that $[\sigma - \chi]_1 \bullet [\mathsf{q}]_2 \bullet = [\mathbf{Z}_{\mathcal{S}}(\sigma)]_1 \bullet [1]_2$. In all known Groth-Sahai based solutions, to verify that $\det \boldsymbol{C}_\sigma(\chi, \mathsf{q}) = 0$, either the encryption of χ or q has to be given in $\mathbb{G}_2$. In AC^*, however, all elements are given as members of $\mathbb{G}_1$. Using the approach from above, AC^*'s membership argument is equal to $([\mathsf{q}, \boldsymbol{\gamma}]_1, [\delta]_2)$, where $\boldsymbol{\gamma} \in \mathbb{Z}_p^2$ and $\delta \in \mathbb{Z}_p$. (We will define $\boldsymbol{\gamma}$ and δ in Section 5.)

Complications. Unfortunately, the described solution is not yet sufficient. The main reason why not is that the implication $(\Sigma - \chi) \mid (\mathbf{Z}_{\mathcal{S}}(\Sigma) - \mathsf{r}) \implies \mathbf{Z}_{\mathcal{S}}(\chi) = \mathsf{r}$ (where $\mathsf{r} = 0$ in the membership case and $\mathsf{r} = 1/\mathsf{s}$ in the non-membership case) holds only if χ and r are integers, that is, they do not depend on the trapdoor σ. Since the verifier only has access to $[\chi]_1$ (and $[\mathsf{s}]_1$ in the non-membership case) as group elements, there is no guarantee that χ (and s) does not depend on σ.

Previous works [BCKL08,AN11,DGP⁺19] solve this problem from scratch, each using a new assumption. We approach it systematically. We define a new security property, F-collision-resistancy. An accumulator is collision-resistant if it is hard for an efficient adversary to return a set $\mathcal{S}$, a candidate element χ, and an accumulator argument ψ, such that the verifier accepts χ as a member of $\mathcal{S}$ iff $\chi \notin \mathcal{S}$. An accumulator is F-collision-resistant if the same holds even if the adversary, instead of χ, outputs $F(\chi)$. (We always have $F(\chi) = [\chi]_1$.) This notion is related to that of F-unforgeable signatures [BCKL08].

We observed that Nguyen's accumulator (and thus the described version of AC^*) is not F-collision-resistant. We solve this issue by introducing another trapdoor τ. The goal of τ is to guarantee that if the verifier accepts, then χ and r do not depend on σ. We also carefully change AC^*'s verification equations. Crucially, we do it without increasing communication complexity. On the other hand, previous work [BCKL08,AN11,DGP$^+$19] introduced a new equation to prove the knowledge relation and thus added new group elements to the argument.

We prove the F-collision-resistance of AC^* under new, essentially tautological, security assumptions DETACM and $\mathsf{DETACNM}$ (determinantal accumulator membership/non-membership). We rely on DETACM (resp., $\mathsf{DETACNM}$) to prove that it is intractable to construct fake accepting membership (resp., non-membership) arguments. Crucially, DETACM and $\mathsf{DETACNM}$ are falsifiable. We prove the security of DETACM and $\mathsf{DETACNM}$ in the AGM. The AGM security proofs are far from trivial and profoundly rely on which elements of AC^*'s argument are or are not multiplied by τ. Note that also the most efficient structure-preserving signatures are proven secure in the generic group model or AGM, the main difference being that the collision-resistance of accumulators is a simpler assumption than the unforgeability of signature schemes.

General Non-Membership CLPØ NIZK. As a result of independent importance, in Section 3, we develop a generic technique for constructing efficient non-membership CLPØ NIZKs. This results, for example, in a very efficient falsifiable NIZKs that the Elgamal-encrypted value χ is non-zero or that two Elgamal-encrypted values are unequal, see Section 3. Both are more efficient than known alternatives [BCV15,BDSS16] based on Groth-Sahai. Such NIZKs have independent applications in, say, anonymous credential systems and privacy-preserving authenticated identification and key exchange protocols [BCV15,BDSS16] and controllable linkability of group signatures, [BDSS16].

New Succinct Set (Non-)Membership NIZK $\mathbf{\Pi}^*$. We are now ready to describe an efficient commit-and-prove NIZK $\mathbf{\Pi}^*$ for showing that an Elgamal-encrypted χ belongs (or does not belong) to the set $\mathcal{S}$. $\mathbf{\Pi}^*$ is just a simple ZK compilation of AC^*. On top of the work done in AC^*, the prover additionally (1) encrypts the data (including the accumulator input χ) one wants to hide, and (2) creates an additional randomizer $[\boldsymbol{z}]_2$ that balances off the randomizers used in such encryptions. The NIZK verifier performs the accumulator verification on the ciphertexts, taking $[\boldsymbol{z}]_2$ into account.

$\mathbf{\Pi}^*$ is computationally sound, assuming that AC^* is F-collision-resistant. Knowing the Elgamal secret key, the reduction decrypts the encrypted data and returns it together with the hint $[\boldsymbol{\delta}]_2$. We emphasize that $\mathbf{\Pi}^*$ is falsifiable. We prove that $\mathbf{\Pi}^*$ is computationally zero-knowledge, assuming that Elgamal is IND-CPA secure (that is, XDH holds).

Efficiency. In Table 1, we provide an efficiency comparison with some previously proposed set (non-)membership NIZKs. In the case of prover's computation, we have taken the standard approach and assumed that the accumulator argument ($[\mathsf{q}]_1$ in our case) is precomputed. This always makes sense if $\mathcal{S}$ is small (then all accumulator arguments can be precomputed), but it is also common in case $\mathcal{S}$ can be large. For example, in an anonymous credential system, one only needs to compute the accumulator argument for its own credential. Moreover, all signature-based solutions have precomputation built-in since the signatures are in the CRS. We hence assume precomputation in all cases.

Updatability. Notably, AC^* and $\mathbf{\Pi}^*$ have an updatable [GKM+18] CRS. That is, it is possible to update the CRS sequentially so that the soundness relies only on the honesty of at least one of the updaters (or the original CRS creator). This partially eliminates the undesirable need to trust the CRS creator. None of the previous falsifiable set membership NIZKs (see Table 1) is updatable: this is caused by the use of (non-updatable) signature schemes and Groth-Sahai NIZK. See [BLL00,Lip12] for work on "transparent" accumulators that do not need a trusted CRS at all. We leave it as another open problem to construct a transparent, efficient, falsifiable set (non-)membership NIZK.

Note that one can build set-membership arguments more efficiently by using (non-falsifiable) zk-SNARKs, but the most efficient zk-SNARKs are not updatable. On the other hand, $\mathbf{\Pi}^*$'s efficiency is comparable to that of most efficient updatable and universal zk-SNARKs like Vampire [LSZ22]. However, the latter are only known to be secure in the ROM.

We end the paper with some general discussion and generalization.

2 Preliminaries

Algebraic Branching Programs. An algebraic branching program (ABP) over a finite field $\mathbb{F}_p$ is defined by a directed acyclic graph (V, E), two special vertices $s, t \in V$, and a labeling function ϕ. It 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 (that is, paths from s to t) of the product of all the values along the path.

Ishai and Kushilevitz [IK00,IK02] related ABPs to matrix determinants. Given an ABP $\textsc{abp} = (V, E, s, t, \phi)$ computing $F : \mathbb{F}_p^\nu \to \mathbb{F}_p$, we can efficiently (and deterministically) compute a function $\mathsf{IK}_F(\boldsymbol{\chi})$ 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) $\det \mathsf{IK}_F(\boldsymbol{\chi}) = F(\boldsymbol{\chi})$, (2) each entry of $\mathsf{IK}_F(\boldsymbol{\chi})$ is an affine map in a single variable χ_i, (3) $\mathsf{IK}_F(\boldsymbol{\chi})$ contains only -1's in the upper 1-diagonal (the diagonal above the main diagonal) and 0's above the upper 1-diagonal.

IK_F is obtained by transposing the matrix you get by removing the column corresponding to s and the row corresponding to t in the matrix $\mathsf{adj}(\boldsymbol{X}) - \boldsymbol{I}$. Here, $\mathsf{adj}(\boldsymbol{X})$ is the adjacency matrix for $\textsc{abp}$ with $\mathsf{adj}(\boldsymbol{X})_{ij} = x$ iff $\phi(i \to j) = x$ and $\mathsf{adj}(\boldsymbol{X})_{ij} = 0$ if there is no edge $i \to j$.

For example, assuming $F(X) = X^2 - X$, one can define an ABP with

$$\mathsf{adj}(X) = \begin{pmatrix} 0 & X & 0 \\ 0 & 0 & X-1 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \mathsf{IK}_F(X) = \begin{pmatrix} X & -1 \\ 0 & X-1 \end{pmatrix} .$$

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, $\mathcal{P}_\iota = [1]_\iota$ is a generator of $\mathbb{G}_\iota$ for $\iota \in \{1, 2, T\}$ with $\mathcal{P}_T = [1]_T := \hat{e}([1]_1, [1]_2)$, 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; that is, we assume that there is no efficient isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$. We use the standard implicit additive "bracket" notation, writing $[a]_\iota$ to denote $a\mathcal{P}_\iota = a[1]_\iota$ 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 $\boldsymbol{A}\boldsymbol{B} = \boldsymbol{C}$ then $[\boldsymbol{A}]_1 \bullet [\boldsymbol{B}]_2 = [\boldsymbol{C}]_T$. We also define $[\boldsymbol{A}]_2 \bullet [\boldsymbol{B}]_1 := ([\boldsymbol{B}]_1^\top \bullet [\boldsymbol{A}]_2{}^\top)^\top = [\boldsymbol{A}\boldsymbol{B}]_T$.

We write $A \approx_c B$ if the distributions A and B are computationally indistinguishable. Let $\ell, \mathsf{k} \in \mathbb{N}$, with $\ell \geq \mathsf{k}$, be small constants. In the case of asymmetric pairings, usually $\mathsf{k} = 1$. Let p be a large prime. A PPT-sampleable distribution $\mathcal{D}_{\ell,\mathsf{k}}$ is a *matrix distribution* if it samples matrices $\boldsymbol{A} \in \mathbb{Z}_p^{\ell \times \mathsf{k}}$ of full rank k. $\mathcal{L}_1$ is the matrix distribution over matrices $\left(\begin{smallmatrix} 1 \\ a \end{smallmatrix}\right)$, where $a \leftarrow_\$ \mathbb{Z}_p$.

The XDH assumption in $\mathbb{G}_\iota$ holds relative to Pgen if for every PPT $\mathcal{A}$,

$$\Pr\left[b' = b \,\middle|\, \begin{array}{c} \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \sigma, \tau, \zeta \leftarrow_\$ \mathbb{Z}_p; b \leftarrow_\$ \{0, 1\}; \\ b' \leftarrow \mathcal{A}([1, \sigma, \tau, \sigma\tau + b\zeta]_\iota) \end{array} \right] \approx_c 0 \enspace .$$

Let $d_1, d_2 \in \mathsf{poly}(\lambda)$. The (d_1, d_2)-PDL *(Power Discrete Logarithm)* assumption holds relative to Pgen, if for any PPT $\mathcal{A}$,

$$\Pr\left[\sigma' = \sigma \,\middle|\, \begin{array}{c} \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \sigma \leftarrow_\$ \mathbb{Z}_p; \\ \sigma' \leftarrow \mathcal{A}(\mathsf{p}; [(\sigma^i)_{i=0}^{d_1}]_1, [(\sigma^i)_{i=0}^{d_2}]_2) \end{array} \right] \approx_c 0 \enspace .$$

Let $\ell, \mathsf{k} \in \mathbb{N}$, and $\mathcal{D}_\mathsf{k}$ be a matrix distribution. The $\mathcal{D}_\mathsf{k}$-$(\ell - 1)$-CED *assumption* [CLPØ21] holds in $\mathbb{G}_\iota$ relative to Pgen, if for all PPT $\mathcal{A}$,

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

CED may or may not be falsifiable, see [CLPØ21] for a discussion.

Following [CH20,CLPØ21], we will be only concerned with the case $\mathsf{k} = 1$ and $\mathcal{D}_\mathsf{k} = \mathcal{L}_1$. Then, $(\boldsymbol{\gamma} \| \boldsymbol{C}) \left(\begin{smallmatrix} \boldsymbol{D} \\ \boldsymbol{\delta} \end{smallmatrix} \right) = \boldsymbol{0}$ iff, after changing the sign of $\boldsymbol{\gamma}$, $\boldsymbol{C} \left(\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix} \right) = \boldsymbol{\gamma}$.

Elgamal encryption. In Elgamal, the public key is $\mathsf{pk} = [1 \| \mathsf{sk}]_1$, and $\mathsf{Enc}_\mathsf{pk}(\chi; \varrho) \leftarrow (\varrho[1]_1 \| \chi[1]_1 + \varrho[\mathsf{sk}]_1)$, where $\varrho \leftarrow_\$ \mathbb{Z}_p$. We also denote the encryption of $[\chi]_1$ by $\mathsf{Enc}_\mathsf{pk}([\chi]_1; \varrho) = (\varrho[1]_1 \| [\chi]_1 + \varrho[\mathsf{sk}]_1)$. To decrypt, one computes $[\chi]_1 = \mathsf{Dec}_\mathsf{sk}([\boldsymbol{c}]_1) \leftarrow -\mathsf{sk}[c_1]_1 + [c_2]_1$; clearly, the result $[\chi]_1$ of the decryption is a group element and not an integer. Note that $\mathsf{pk} = \mathsf{Enc}_\mathsf{pk}(0; 1)$ and $[0 \| \chi]_1 = \mathsf{Enc}_\mathsf{pk}(\chi; 0)$. As always, we denote $\mathsf{Enc}_\mathsf{pk}([\boldsymbol{a}]_1; \boldsymbol{\varrho}) := (\mathsf{Enc}_\mathsf{pk}([a_i]_1; \varrho_i))_i$. Elgamal is IND-CPA secure under the XDH assumption.

Algebraic Group Model. AGM [FKL18] is an idealized model for security proofs. In the AGM, adversaries are restricted to be *algebraic* in the following sense: if $\mathcal{A}$ inputs some group elements and outputs a group element, it can provide an algebraic representation of the latter in terms of the former.

More precisely, let $\mathbb{G}$ be a cyclic group of prime order p. Let $\mathcal{A}_{\text{alg}}$ be a PPT algorithm, run on initial inputs including description p with oracles or other parties and receive further inputs including obliviously sampled group elements (which it cannot sample directly). Let $\boldsymbol{L} \in \mathbb{G}^n$ be the list of all group elements $\mathcal{A}$ has been given so far such that all other inputs it has received do not depend in any way on group elements. $\mathcal{A}$ is *algebraic* if whenever it outputs a group element $G \in \mathbb{G}$ it also outputs a vector $\boldsymbol{a} = (a_i)_{i=1}^n \in \mathbb{Z}_p^n$, such that $G = \sum_{i=1}^n a_i L_i = \langle \boldsymbol{a}, \boldsymbol{L} \rangle$.

2.1 Universal NIZK Arguments

Let $\{\mathcal{D}_\mathsf{p}\}_\mathsf{p}$ be a family of distributions, s.t. each $\mathtt{lpar} \in \mathcal{D}_\mathsf{p}$ defines a language $\mathcal{L}_{\mathtt{lpar}}$. A universal NIZK Π for $\{\mathcal{D}_\mathsf{p}\}_\mathsf{p}$ consists of six probabilistic algorithms:

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

Key generation $\mathsf{Kgen}(\mathsf{p}, N)$: generates a CRS $\mathtt{crs}$ and a trapdoor $\mathtt{td}$. Here, N is a public size parameter (an upper bound of the size $\mathcal{S}$ in our case); we assume N is implicitly in the CRS. We omit N if the CRS does not depend on it. For simplicity of notation, we assume that any group parameters are implicitly included in the CRS. We often denote the sequence "$\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$; $(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{Kgen}(\mathsf{p}, N)$" by $(\mathsf{p}, \mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{Kgen}(1^\lambda, N)$.

Computation commitment $\mathsf{Com}(\mathtt{crs}, \mathtt{lpar})$: Given a CRS $\mathtt{crs}$ and a language description $\mathtt{lpar} \in \mathcal{D}_\mathsf{p}$, outputs a specialized CRS $\mathtt{crs}_{\mathtt{lpar}}$. We assume that $\mathtt{crs}_{\mathtt{lpar}}$ implicitly contains $\mathtt{lpar}$. Com is a deterministic algorithm that can hence be run by both the prover and the verifier. (This algorithm is also known as CRS specialization algorithm, indexer, or derive.)

Prover $\mathsf{P}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x}, \mathbb{w})$: Given a specialized CRS $\mathtt{crs}_{\mathtt{lpar}}$ and a statement $\mathbb{x}$ with witness $\mathbb{w}$, outputs an argument π for $\mathbb{x} \in \mathcal{L}_{\mathtt{lpar}}$.

Verifier $\mathsf{V}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x}, \pi)$: Given a specialized CRS $\mathtt{crs}_{\mathtt{lpar}}$, a statement, and an argument, either accepts or rejects the argument.

Simulator $\mathsf{Sim}(\mathtt{crs}_{\mathtt{lpar}}, \mathtt{td}, \mathbb{x})$: Given a specialized CRS $\mathtt{crs}_{\mathtt{lpar}}$, a trapdoor $\mathtt{td}$, and a statement $\mathbb{x}$, outputs a simulated argument for $\mathbb{x} \in \mathcal{L}_{\mathtt{lpar}}$.

The CRS does not depend on the language distribution or language parameters. However, Com (applied on public arguments) allows one to derive a specialized CRS such that the verifier's operation is efficient given $\mathtt{crs}_{\mathtt{lpar}}$.

The following properties need to hold for a NIZK argument.

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

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

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

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

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

Π is *commit-and-prove* if its input $\mathbb{x}$ is a ciphertext, such that the argument convinces the verifier some statement about $\mathsf{Dec}_{\mathsf{sk}}(\mathbb{x})$, i.e., that $\det_{\mathsf{sk}}(\mathbb{x}) \in \mathcal{L}$ for some language $\mathcal{L}$. Commit-and-prove argument systems are usually modular, i.e., one can share the encrypted inputs between several argument systems that prove different properties of the same input. The CLPØ argument system [CLPØ21] (see Section 2.3) is commit-and-prove.

A sound Π is *updatable* [GKM$^+$18] if one can sequentially update the CRS multiple times so that if at least one of the updaters (or the initial CRS creator) is honest, then Π remains sound. We will not give a formal definition. As shown by Groth et al. [GKM$^+$18], a (pairing-based) Π is updatable in the case its CRS is of shape $([f(\boldsymbol{x}) : f \in \mathcal{T}_1]_1, [f(\boldsymbol{x}) : f \in \mathcal{T}_2]_2)$, where $\boldsymbol{x}$ is a vector of trapdoors over $\mathbb{Z}_p$, and $\mathcal{T}_\iota$ are sets of monomials. For example, $\mathtt{crs} = ([1, \tau, \sigma\tau, \dots, \sigma^N\tau]_1, [1, \sigma, \tau, \sigma\tau]_2)$. On the other hand, Π is not updatable if either $\mathcal{T}_1$ or $\mathcal{T}_2$ contains a non-monomial.

Set (Non-)Membership NIZK. Let $\mathcal{D}$ be some finite domain; in the current paper, $\mathcal{D} = \mathbb{Z}_p$. Let pk be an Elgamal public key and $\mathcal{S}$ be a set of size $\mathcal{S} \in \mathcal{D}^{\leq N}$ for fixed $N = \mathsf{poly}(\lambda)$. Let $\mathtt{lpar} = (\mathsf{pk}, \mathcal{S})$. In the case of NIZKs for set membership and non-membership, we are interested in the following complementary (commit-and-prove) languages:

$$\mathcal{L}^{\mathsf{sm}}_{\mathtt{lpar}} = \left\{ [\mathbf{ct}_\chi]_1 \,\middle|\, \exists \chi, \varrho_\chi \text{ such that } \mathsf{Enc}_{\mathsf{pk}}([\boldsymbol{\chi}]_1; \varrho_\chi) = [\mathbf{ct}_\chi]_1 \wedge \chi \in \mathcal{S} \right\} ,$$
$$\bar{\mathcal{L}}^{\mathsf{sm}}_{\mathtt{lpar}} = \left\{ [\mathbf{ct}_\chi]_1 \,\middle|\, \exists \chi, \varrho_\chi \text{ such that } \mathsf{Enc}_{\mathsf{pk}}([\boldsymbol{\chi}]_1; \varrho_\chi) = [\mathbf{ct}_\chi]_1 \wedge \chi \notin \mathcal{S} \right\} .$$

Instead of defining two NIZKs (for $\mathcal{L}^{\mathsf{sm}}_{\mathtt{lpar}}$ and $\bar{\mathcal{L}}^{\mathsf{sm}}_{\mathtt{lpar}}$), we define a single NIZK where the two arguments share a common CRS. If $\chi \in \mathcal{S}$ (resp., $\chi \notin \mathcal{S}$), then the prover generates a membership (resp., non-membership) argument. The verifier/simulator take an additional argument $mem \in \{\mathsf{Member}, \mathsf{NotMember}\}$. The verifier assumes that its input is a membership argument if $mem = \mathsf{Member}$, and a non-membership argument otherwise. It outputs either Member, $\mathsf{NotMember}$, or Error. We generalize the simulator similarly.

2.2 Accumulators

Benaloh and de Mare defined accumulators in [BdM93]. Universal accumulators [BLL00,BLL02,LLX07] allow non-membership arguments.

We define accumulators in the CRS model only. Hence, within the context of the current paper, universal accumulators are set (non-)membership NIZKs in the case the input χ is public. That is, for $\mathtt{lpar} = \mathcal{S}$, a universal (CRS-model) accumulator is a (non-zk) set (non-)membership non-interactive argument system for the following complementary languages:

$$\mathcal{L}^{\mathsf{acc}}_{\mathtt{lpar}} = \mathcal{S} \ , \quad \bar{\mathcal{L}}^{\mathsf{acc}}_{\mathtt{lpar}} = \mathcal{D} \setminus \mathcal{S} \ .$$

The computation commitment algorithm Com corresponds to the accumulator's commitment algorithm that inputs a set $\mathcal{S}$ and outputs its short commitment. A CRS-model accumulator can have a trapdoor. However, since χ is public (and no zero-knowledge is required) then the trapdoor is not used.

As all argument systems, a universal accumulator must satisfy completeness and soundness properties. Because of the historical reasons, the latter is usually known as *collision-resistance*. Full definitions follow.

A universal accumulator ACC must be *perfectly complete*: for $(\mathtt{crs}, \mathtt{td}) \in \mathsf{Kgen}(1^\lambda)$, $\chi \in \mathcal{D}$, and $\mathcal{S} \in \mathcal{D}^{\leq N}$, $\mathsf{V}(\mathtt{crs}, \mathsf{Com}(\mathtt{crs}, \mathcal{S}), \chi, \mathsf{P}(\mathtt{crs}, \mathcal{S}, \chi))$ outputs Member if $\chi \in \mathcal{S}$ and $\mathsf{NotMember}$ if $\chi \notin \mathcal{S}$.

Definition 1. *Let* ACC *be a universal accumulator.* ACC *is* collision-resistant [BP97] *if for all* $N = \mathsf{poly}(\lambda)$ *and PPT adversaries* $\mathcal{A}$*,*

$$\Pr\left[\begin{array}{c|c} \mathcal{S} \in \mathcal{D}^{\leq N} \wedge & \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \\ \left(\begin{array}{c} (\chi \notin \mathcal{S} \,\wedge\, v = \mathsf{Member}) \vee \\ (\chi \in \mathcal{S} \,\wedge\, v = \mathsf{NotMember}) \end{array}\right) & \begin{array}{c} (\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{Kgen}(\mathsf{p}, N); \\ (\mathcal{S}, \chi, \psi) \leftarrow \mathcal{A}(\mathtt{crs}); \\ v \leftarrow \mathsf{V}(\mathtt{crs}, \mathsf{Com}(\mathtt{crs}, \mathcal{S}), \chi, \psi) \end{array} \end{array}\right] \approx_c 0 \enspace .$$

Nguyen [Ngu05] proposed a pairing-based CRS-model accumulator with $\mathcal{D} = \mathbb{Z}_p$. Damgård and Triandopoulos [DT08] and Au et al. [ATSM09] showed independently how to make it universal by adding a non-membership argument.

In Fig. 2, we depict the resulting CRS-model universal accumulator, assuming that $\mathcal{S} \in \mathcal{D}^{\leq N}$. Here, and in what follows, $\mathbf{Z}_\mathcal{S}(\Sigma) := \prod_{s \in \mathcal{S}} (\Sigma - s)$ is the vanishing polynomial of $\mathcal{S}$. We slightly simplified its description: Nguyen originally defined $\mathbf{Z}_\mathcal{S}(\Sigma) := \prod_{s \in \mathcal{S}} (\Sigma + s)$ (that is, $\mathbf{Z}_\mathcal{S}(\Sigma)$ was the vanishing polynomial of $-\mathcal{S} = \{-s : s \in \mathcal{S}\}$), while we define $\mathbf{Z}_\mathcal{S}(\Sigma) := \prod_{s \in \mathcal{S}} (\Sigma - s)$; we modified the rest of the formulas in a consistent manner to account for this change. Note that $\mathbf{Z}_\mathcal{S}(\chi) = 0$ iff $\chi \in \mathcal{S}$. Intuitively, the prover proves that $\chi \in \mathcal{S}$ by showing that $\mathbf{Z}_\mathcal{S}(\chi) = 0$ and $\chi \notin \mathcal{S}$ by showing that $\mathbf{Z}_\mathcal{S}(\chi) \neq 0$. A membership argument is shorter since in this case, $\mathbf{Z}_\mathcal{S}(\chi) = 0$ and thus the prover does not have to transfer $\mathbf{Z}_\mathcal{S}(\chi)$.

Note that $[\mathsf{q}]_1 \leftarrow [(\mathbf{Z}_\mathcal{S}(\sigma) - \mathsf{r})/(\sigma - \chi)]_1$ is well defined even if $\sigma = \chi$. In this case, $f(X) = (\mathbf{Z}_\mathcal{S}(X) - \mathbf{Z}_\mathcal{S}(\chi))/(X - \chi) = \prod_{s \in \mathcal{S}\{\chi\}} (X - s)$ is clearly a polynomial, and thus we can set $[\mathsf{q}]_1 \leftarrow [f(\chi)]_1$.

Com can be seen as a preprocessing algorithm. One can do even more preprocessing in typical accumulators (including Nguyen's). One can precompute accumulator arguments for all $\chi \in \mathcal{S}$ to speed up the online phase of a set membership (but not non-membership) argument. In some applications, one can precompute ψ corresponding to concrete χ. We will always assume this is the case, but, to avoid notational bloat, we will not study preprocessing formally.

2.3 CLPØ NIZK

Since we build on CLPØ [CLPØ21], we will give a lengthier description of their results. Fix $\mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda)$ and define $\mathcal{D}_\mathsf{p} := \{\mathtt{lpar} = (\mathsf{pk}, F)\}$, where (1) pk is a randomly chosen Elgamal public key for encrypting in $\mathbb{G}_1$, and (2) F is a

```
$\mathsf{Pgen}(1^\lambda)$: the same as the bilinear group generator; returns $\mathsf{p}$.

$\mathsf{Kgen}(\mathsf{p}, N)$: $\sigma \leftarrow_\$ \mathbb{Z}_p$; $\mathtt{crs} \leftarrow (\mathsf{p}, [(\sigma^i)_{i=0}^N]_1, [1, \sigma]_2)$;
    return $(\mathtt{crs}, \mathtt{td} = \sigma)$;

$\mathsf{Com}(\mathtt{crs}, \mathcal{S})$: given $|\mathcal{S}| = N$: output $[\mathsf{C}_\mathcal{S}]_1 \leftarrow [\mathbf{Z}_\mathcal{S}(\sigma)]_1$;

$\mathsf{P}(\mathtt{crs}, \mathcal{S}, \chi)$: $\mathsf{r} \leftarrow \mathbf{Z}_\mathcal{S}(\chi)$; $[\mathsf{q}]_1 \leftarrow [(\mathbf{Z}_\mathcal{S}(\sigma) - \mathsf{r})/(\sigma - \chi)]_1$;
    If $\chi \in \mathcal{S}$ then $\psi \leftarrow [\mathsf{q}]_1$ else $\psi \leftarrow ([\mathsf{q}]_1, \mathsf{r})$;
    return $\psi$;

$\mathsf{V}(\mathtt{crs}, \mathsf{C}_\mathcal{S}, \chi, \psi)$: If $\psi$ parses as $\psi = ([\mathsf{q}]_1, \mathsf{r})$ and $\mathsf{r} = 0$ then return $\mathsf{Error}$;
    If $\psi$ parses as $\psi = [\mathsf{q}]_1$ then $\mathsf{r} \leftarrow 0$;
    If $[\mathsf{q}]_1 \bullet ([\sigma]_2 - \chi[1]_2) + (\mathsf{r}[1]_1 - [\mathsf{C}_\mathcal{S}]_1) \bullet [1]_2 \neq [0]_T$ then return $\mathsf{Error}$;
    If $\mathsf{r} = 0$ then return $\mathsf{Member}$ else return $\mathsf{NotMember}$;
```

Fig. 2. Nguyen's universal accumulator $\mathsf{ACC}_{\mathsf{Nguyen}}$.

polynomial. The simplest version of CLPØ is a set membership NIZK for the set being defined as the set $\mathcal{Z}(F)$ of zeros of the fixed polynomial F.

More precisely, let $\mathcal{S} = \mathcal{Z}(F) := \{x : F(X) = 0\}$ for a 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 $[\mathbf{ct}_\chi]_1 := \mathsf{Enc}_{\mathsf{pk}}([\boldsymbol{\chi}]_1; \boldsymbol{\varrho}) = (\mathsf{Enc}_{\mathsf{pk}}([\chi_i]_1; \varrho_i))_i$. Define

$$\mathcal{L}_{\mathtt{lpar}} = \{[\mathbf{ct}_\chi]_1 : \exists \boldsymbol{\chi} \text{ such that } \mathsf{Dec}_{\mathsf{sk}}([\mathbf{ct}_\chi]_1) = [\boldsymbol{\chi}]_1 \wedge \boldsymbol{\chi} \in \mathcal{Z}(F)\} \ . \tag{1}$$

Hence, $\mathcal{L}_{\mathtt{lpar}}$ is a commit-and-prove language. 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.

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 quasideterminantal representation (QDR [CLPØ21]) of* F, if the following requirements hold. Here, $\boldsymbol{C}(\boldsymbol{X}) = (\boldsymbol{h}(\boldsymbol{X}) \| \boldsymbol{T}(\boldsymbol{X}))$, where $\boldsymbol{h}(\boldsymbol{X})$ is a column vector.

Affine map: $\boldsymbol{C}(\boldsymbol{X})$ is an affine map. That is, $\boldsymbol{C}(\boldsymbol{X}) = \sum_{k=1}^{\nu} \boldsymbol{P}_k X_k + \boldsymbol{Q}$, where $\boldsymbol{P}_k, \boldsymbol{Q} \in \mathbb{Z}_p^{\ell \times \ell}$ are public matrices.

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

First column dependence: For any $\boldsymbol{\chi} \in \mathcal{Z}(F)$, $\boldsymbol{h}(\boldsymbol{\chi}) \in \operatorname{colspace}(\boldsymbol{T}(\boldsymbol{\chi}))$. That is, $\boldsymbol{h}(\boldsymbol{\chi}) = \boldsymbol{T}(\boldsymbol{\chi})\mathbf{w}$ for some $\mathbf{w}$.

The quasideterminantal complexity $\mathsf{qdc}(F)$ of F is the smallest QDR size of F. (Clearly, $\mathsf{qdc}(F) \geq \deg(F)$.) We always assume that the polynomial F in $\mathtt{lpar}$ satisfies $\mathsf{qdc}(F) = \mathsf{poly}(\lambda)$, that is, there exists a $\mathsf{poly}(\lambda)$-size QDR $\boldsymbol{C}(\boldsymbol{X})$ of F. [CLPØ21] showed that such QDRs exist for many F-s.

CLPØ Argument. In Fig. 3, we depict the commit-and-prove updatable universal CLPØ NIZK $\boldsymbol{\Pi}_{\mathsf{clp}ø}$. Intuitively, the verifier checks that $[\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix}]_2 \bullet [\boldsymbol{C}(\mathbf{ct}_\chi)]_1 = [\boldsymbol{I}_\ell]_2 \bullet [\mathbf{ct}_\gamma]_1 + [\boldsymbol{z}]_2 \bullet \mathsf{pk}$, where $[\boldsymbol{C}(\mathbf{ct}_\chi)]_1 := \sum_{k=1}^{\nu} \boldsymbol{P}_k \cdot [\mathsf{ct}_{\chi k}]_1 + \boldsymbol{Q} \cdot \mathsf{Enc}_{\mathsf{pk}}(1; 0)$. Couteau et al. [CLPØ21] did not use the terminology of commit-and-prove, universal, and updatable NIZKs. Still, $\boldsymbol{\Pi}_{\mathsf{clp}ø}$ satisfies these properties.

```
$\mathsf{Pgen}(1^\lambda)$: returns the system parameters $\mathsf{p}$, as always.
$\mathsf{Kgen}(\mathsf{p})$: $\mathsf{e} \leftarrow_{\$} \mathbb{Z}_p$; return $(\mathtt{crs}, \mathtt{td}) \leftarrow ([\mathsf{e}]_2, \mathsf{e})$;
$\mathsf{Com}(\mathtt{crs}, \mathtt{lpar})$: return $\mathtt{crs}_{\mathtt{lpar}} \leftarrow (\mathtt{crs}, \mathtt{lpar})$;
$\mathsf{P}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x} = [\mathbf{ct}_\chi]_1, \mathbb{w} = (\boldsymbol{\chi}, \boldsymbol{\varrho}))$: Write $\boldsymbol{C}(\boldsymbol{\chi}) = (\boldsymbol{h} \| \boldsymbol{T})(\boldsymbol{\chi})$;
    $\boldsymbol{\varrho}_{\boldsymbol{\delta}} \leftarrow_{\$} \mathbb{Z}_p^{\ell-1}$; $\boldsymbol{\gamma} \leftarrow -\boldsymbol{T}(\boldsymbol{\chi}) \boldsymbol{\varrho}_{\boldsymbol{\delta}}$;
    Compute $\mathbf{w}$ such that $\boldsymbol{T}(\boldsymbol{\chi}) \mathbf{w} = \boldsymbol{h}(\boldsymbol{\chi})$;
    $[\boldsymbol{\delta}]_2 \leftarrow -(\mathbf{w}[\mathsf{e}]_2 + \boldsymbol{\varrho}_{\boldsymbol{\delta}}[1]_2)$;
    $\boldsymbol{\varrho}_\gamma \leftarrow_{\$} \mathbb{Z}_p^{\ell}$; $[\mathbf{ct}_\gamma]_1 \leftarrow \mathsf{Enc}_{\mathsf{pk}}([\boldsymbol{\gamma}]_1; \boldsymbol{\varrho}_\gamma) \in \mathbb{G}_1^{\ell \times 2}$;
    $[\boldsymbol{z}]_2 \leftarrow (\sum_{k=1}^{\nu} \varrho_k \boldsymbol{P}_k) \left[\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 - \boldsymbol{\varrho}_\gamma [1]_2 \in \mathbb{G}_2^{\ell}$;
    Return $\pi \leftarrow ([\mathbf{ct}_\gamma]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2) \in \mathbb{G}_1^{\ell \times 2} \times \mathbb{G}_2^{2\ell - 1}$;
$\mathsf{V}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x} = [\mathbf{ct}_\chi]_1, \pi)$: check $\sum_{k=1}^{\nu} (\boldsymbol{P}_k \left[\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 \bullet [\mathbf{ct}_k]_1) + \boldsymbol{Q} \left[\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix}\right]_2 \bullet [0 \| 1]_1 =$
    $[\boldsymbol{I}_\ell]_2 \bullet [\mathbf{ct}_\gamma]_1 + [\boldsymbol{z}]_2 \bullet \mathsf{pk}$;
$\mathsf{Sim}(\mathtt{crs}_{\mathtt{lpar}}, \mathtt{td}, \mathbb{x} = [\mathbf{ct}_\chi]_1)$: $\boldsymbol{\delta} \leftarrow_{\$} \mathbb{Z}_p^{\ell-1}$;
    $\boldsymbol{z} \leftarrow_{\$} \mathbb{Z}_p^{\ell}$; $[\mathbf{ct}_\gamma]_1 \leftarrow \sum_{k=1}^{\nu} \boldsymbol{P}_k (\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix}) [\mathbf{ct}_k]_1 + \mathsf{Enc}_{\mathsf{pk}}(\boldsymbol{Q}(\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix})[1]_1; -\boldsymbol{z})$;
    Return $\pi \leftarrow ([\mathbf{ct}_\gamma]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2) \in \mathbb{G}_1^{\ell \times 2} \times \mathbb{G}_2^{2\ell - 1}$;
```

Fig. 3. The commit-and-prove CLPØ NIZK $\mathbf{\Pi}_{\mathsf{clp}\emptyset}$ for $\mathcal{L}_{\mathsf{pk},F}$.

We will state Fact 1 for the sake of completeness.

Fact 1 ([CLPØ21]) *Let $\{\mathcal{D}_{\mathsf{p}}\}_{\mathsf{p}}$ be the family of language distributions, where $\mathcal{D}_{\mathsf{p}} = \{\mathtt{lpar} = (\mathsf{pk}, F)\}$. 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 $\mathbf{\Pi}_{\mathsf{clp}\emptyset}$ for $\{\mathcal{D}_{\mathsf{p}}\}_{\mathsf{p}}$ from Fig. 3 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.

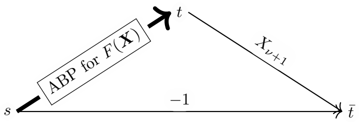
Efficient Instantiation Based on ABP. Couteau et al. [CLPØ21] constructed a QDR $\mathsf{IK}_F(\boldsymbol{X})$ for any polynomial F that can be efficiently computed by an algebraic branching program (ABP).

Fact 2 ([CLPØ21]) *Let $\text{ABP} = (V, E, s, t, \phi)$ be an ABP that computes a ν-variate polynomial $F(\boldsymbol{X})$. Then $\mathsf{IK}_F(\boldsymbol{X})$ is a QDR of F with $\ell = |V| - 1$.*

In particular, $\mathsf{qdc}(F) \leq |V| - 1$. This results in NIZKs for $\mathcal{L}_{\mathsf{pk},F}$ whenever F has a small ABP.

3 General Non-Membership NIZK Argument System

For a set $\mathcal{F}$ of polynomials, let $\mathcal{Z}(\mathcal{F})$ be the set of common zeros of all $F_i \in \mathcal{F}$. Next, we construct efficient (commit-and-prove, updatable, universal)



$$\left(\begin{array}{c|c} \mathsf{IK}_F(X_1,\dots,X_\nu) & \begin{matrix}0\\0\\ \vdots\\0\\-1\end{matrix} \\ \hline -1\ 0\ \dots\ 0 \qquad 0 & X_{\nu+1}\end{array}\right)$$

Fig. 4. ABP $\overline{\text{ABP}}$ for the $\bar{F}(\boldsymbol{X}, X_{\nu+1}) = F(\boldsymbol{X})X_{\nu+1} - 1$ and the matrix $\mathsf{IK}_{\bar{F}}(\boldsymbol{X}, X_{\nu+1})$.

non-membership NIZKs for $\mathcal{S} = \mathcal{Z}(\mathcal{F})$, given that for each $F_i \in \mathcal{F}$, there exists a small ABP that computes F_i. The modifications are at the level of ABP and thus do not depend on the inner workings of $\boldsymbol{\Pi}_{\mathsf{clp}\emptyset}$. The current section has independent importance since non-membership NIZKs have their own applications, [ATSM09,BCV15,BDSS16,BBLP21].

New Non-Membership NIZK. Assume $\mathcal{F} = \{F\}$, where $F(\boldsymbol{X}) : \mathbb{F}_p^\nu \mapsto \mathbb{F}_p$ is a polynomial that can be computed by a small ABP $\text{ABP} = (V, E, s, t, \phi)$. We construct a new ABP $\overline{\text{ABP}}$ as follows (see Fig. 4): we add to ABP a new target vertex $\bar{t}$ and two edges, $s \to \bar{t}$ and $t \to \bar{t}$. We naturally extend ϕ to a new labeling function $\bar{\phi}$, such that $\bar{\phi}(s \to \bar{t}) = -1$ and $\bar{\phi}(t \to \bar{t}) = X_{\nu+1}$, where $X_{\nu+1}$ is a new indeterminate. Let $\bar{F}(\boldsymbol{X}, X_{\nu+1}) : \mathbb{F}_p^{\nu+1} \mapsto \mathbb{F}_p$, $\bar{F}(\boldsymbol{X}, X_{\nu+1}) = F(\boldsymbol{X})X_{\nu+1} - 1$, be the polynomial computed by $\overline{\text{ABP}}$. Clearly, if $F(\boldsymbol{\chi}) = 0$ for a concrete input assignment $\boldsymbol{\chi}$, then $\bar{F}(\boldsymbol{\chi}, \chi_{\nu+1}) = -1 \neq 0$ for all values of $\chi_{\nu+1}$. On the other hand, if $F(\boldsymbol{\chi}) \neq 0$, then there exists $\chi_{\nu+1} = F(\boldsymbol{\chi})^{-1}$, such that $\bar{F}(\boldsymbol{\chi}, \chi_{\nu+1}) = 0$.

Thus, to obtain a non-membership NIZK for the algebraic set $\mathcal{S} = \mathcal{Z}(F)$, it suffices to construct a membership NIZK for the algebraic set $\bar{\mathcal{S}} = \mathcal{Z}(\bar{F})$. For this, one can use $\boldsymbol{\Pi}_{\mathsf{clp}\emptyset}$ from Fig. 4 for the QDR $\mathsf{IK}_{\bar{F}}$. The resulting NIZK is again secure under a CED assumption (see Fact 1). Moreover, if the NIZK for F relies on a falsifiable version of CED, then so does the NIZK for $\bar{F}$.

Examples. To show that $\chi \neq 0$, we can run $\boldsymbol{\Pi}_{\mathsf{clp}\emptyset}$ with the QDR

$$\bar{\boldsymbol{C}}(\mathsf{X}, \mathsf{S}) := \left(\begin{smallmatrix} \mathsf{X} & -1 \\ -1 & \mathsf{S} \end{smallmatrix}\right) ,$$

where in the honest case, $\mathsf{S} = 1/\mathsf{X}$. One can easily extend it to the proof that two plaintexts χ_1 and χ_2 are unequal, by using the QDR

$$\bar{\boldsymbol{C}}(\mathsf{X}_1, \mathsf{X}_2, \mathsf{S}) := \left(\begin{smallmatrix} \mathsf{X}_1 - \mathsf{X}_2 & -1 \\ -1 & \mathsf{S} \end{smallmatrix}\right) ,$$

where in the honest case, $\mathsf{S} = 1/(\mathsf{X}_1 - \mathsf{X}_2)$.

The argument length of the resulting NIZKs (including encryption of s but not of χ or χ_i) is $6\mathfrak{g}_1 + 3\mathfrak{g}_2$. They are based on a less standard and non-falsifiable assumption (CED instead of SXDH) but are significantly more efficient than Groth-Sahai-based constructions of [BCV15,BDSS16]. In particular, the communication of the NIZK of plaintext inequality of [BCV15] consists of 15 elements of $\mathbb{G}_1$, 4 elements of $\mathbb{G}_2$, and 2 elements of $\mathbb{Z}_p$. (The more efficient construction [BBLP21] works in the random oracle model.)

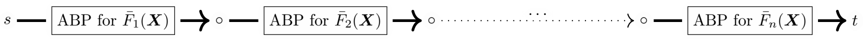

s
ABP for $\bar{F}_1(\boldsymbol{X})$
ABP for $\bar{F}_2(\boldsymbol{X})$
ABP for $\bar{F}_n(\boldsymbol{X})$
t

Fig. 5. ABP $\overline{\text{ABP}}$ for $\bar{F}(\boldsymbol{X}) = \prod \bar{F}_i(\boldsymbol{X})$.

Finally, consider the task of proving that an encrypted integer χ is non-Boolean. In this case, one can define the QDR

$$\boldsymbol{C}_{\{0,1\}}(X, \mathsf{S}) := \begin{pmatrix} X & -1 & 0 \\ 0 & X-1 & -1 \\ -1 & 0 & \mathsf{S} \end{pmatrix} .$$

Generalization. Let $\mathcal{F} = \{F_1, \ldots, F_\nu\}$ for $\nu > 1$. To obtain a set non-membership NIZK for $\mathcal{S} = \mathcal{Z}(\mathcal{F})$, we first construct an ABP that computes each $\bar{F}_i$ (see the previous subsubsection). After that, we construct an ABP that computes a polynomial $\bar{F}(\boldsymbol{X})$, such that $\bar{F}(\boldsymbol{\chi}) = 0$ iff $\bar{F}_i(\boldsymbol{\chi}) = 0$ for some i. Define $\bar{F}(\boldsymbol{X}) = \prod \bar{F}_i(\boldsymbol{X})$, and define its ABP as the concatenation of the ABPs for individual polynomials $\bar{F}_i$. See Fig. 5. We then use $\boldsymbol{\Pi}_{\mathsf{clp}\varnothing}$ for the QDR $\mathsf{IK}_{\bar{F}}$ from Fig. 4. The resulting NIZK is secure according to Fact 1.

4 Determinantal Accumulators

It is easy to see that universal accumulator is a *non-zk* set (non-)membership non-interactive argument system (i.e., one that possesses both membership and non-membership arguments). Hence, it is logical to try to construct a set (non-)membership NIZK by first constructing an accumulator and then adding a zero-knowledge layer to obtain privacy.

While the end goal is to define efficient NIZKs, both steps of the descrived blueprint can be expensive per se. In the current paper, we are interested in constructing a CLPØ-style set (non-)membership NIZK where the second step is as simple as possible. To achieve this, we first reinterpret $\boldsymbol{\Pi}_{\mathsf{clp}\varnothing}$. We then use the obtained understanding to define and construct *determinantal accumulators* that allow for a simple zero-knowledge layer. For latter, a determinantal accumulator must have a specific structure consistent with $\boldsymbol{\Pi}_{\mathsf{clp}\varnothing}$'s design.

The relation between determinantal accumulators and CLPØ is similar to the relation between structure-preserving signatures and Groth-Sahai. Hence, we also compare both primitives.

Intuition. Recall that in $\boldsymbol{\Pi}_{\mathsf{clp}\varnothing}$ [CLPØ21], one rewrites the condition $\boldsymbol{\chi} \in \mathcal{S}$ as the condition $F_i(\boldsymbol{\chi}) = 0$ for a set of polynomials $\{F_i\}$.[4] After that, one constructs QDRs $\boldsymbol{C}_i(\boldsymbol{X})$ for each F_i, such that $\det \boldsymbol{C}_i(\boldsymbol{X}) = F_i(\boldsymbol{X})$. This step can be seen as linearization: while F_i can be a high-degree polynomial, each entry of $\boldsymbol{C}_i$ is an affine map. As typical in group-based cryptography, it is easier to solve linearized tasks. After that, [CLPØ21] proposes a technique of constructing QDRs (i.e., linearization algorithm) by using algebraic branching programs.

[4] In our new primitives, the set consists of only one polyomial. However, the framework is valid in the more general case.

Given the QDRs, $\mathbf{\Pi}_{\mathsf{clp\emptyset}}$'s prover P aims to convince the verifier that each $\det \boldsymbol{C}_i(\boldsymbol{\chi})$ is zero. Crucially, the verifier has access only to encrypted $[\boldsymbol{\chi}]_1$ but not to $\boldsymbol{\chi}$ or even $[\boldsymbol{\chi}]_1$. Since each entry of $\boldsymbol{C}_i$ is affine and the cryptosystem is additively homomorphic, the verifier can compute an encryption of $[\boldsymbol{C}_i(\boldsymbol{\chi})]_1$ given an encryption of $[\boldsymbol{\chi}]_1$. Knowing sk, the soundness reduction decrypts ciphertexts, obtains $[\boldsymbol{C}_i(\boldsymbol{\chi})]_1$, and uses it to break CED. To preserve privacy, the verifier cannot $[\boldsymbol{C}_i(\boldsymbol{\chi})]_1$ and thus also not $\det \boldsymbol{C}_i(\boldsymbol{\chi})$.

In a *non-zk* CLPØ-style non-interactive argument system, we proceed as in CLPØ, except that we do not encrypt any of the values. In particular, similarly to the soundness reduction in $\mathbf{\Pi}_{\mathsf{clp\emptyset}}$, the verifier has access to $[\chi]_1$ and thus also to $[\boldsymbol{C}_i(\boldsymbol{\chi})]_1$. To be compatible with CLPØ, the verifier is not however given access to $\det \boldsymbol{C}_i(\boldsymbol{\chi})$ or even $\boldsymbol{\chi}$ as integers. Given this, we must take additional care to ensure that the accumulator will be secure.

4.1 Determinant Verification

The verifier needs to check efficiently that the determinant of a given matrix $\boldsymbol{C}_i(\boldsymbol{\chi})$ is zero. The main problem is that since the verifier sees $[\boldsymbol{C}_i(\boldsymbol{\chi})]_1$ but not $\boldsymbol{C}_i(\boldsymbol{\chi})$, the verifier's task is intractable. Next, we outline a straightforward but non-satisfactory solution to this problem together with three modifications.

First, without any additional hints given to the verifier, we have an accumulator with inefficient verification, where the verifier computes the discrete logarithm of $[\boldsymbol{C}_i(\boldsymbol{\chi})]_1$ to obtain $\boldsymbol{C}_i(\boldsymbol{\chi})$. This might be fine in the NIZK since the NIZK verifier does not have to perform the accumulator verification; instead, the NIZK verifier checks (efficiently) the NIZK argument showing that the accumulator verifier accepts. However, since also the soundness reduction does not get any hints about $\boldsymbol{C}_i(\boldsymbol{\chi})$, it will not be able to verify whether this results in a non-falsifiable NIZK, as explained in [CH20,CLPØ21].

Second, following [ALSZ20], we can allow the prover to output as hints all partial multiplications needed in the Leibniz formula for the determinant. In that case, one can obtain a PPT verifiable accumulator and thus a NIZK based on falsifiable assumptions. However, while PPT, it is concretely very expensive: if the dimension of the matrix is large, the hint is potentially huge [ALSZ20]. [5] Moreover, since in the NIZK, one has to encrypt the matrix elements in both groups, one has to use the less efficient DLIN encryption, see [CLPØ21].

Third, we can use the undergraduate linear-algebraic fact that $\det \boldsymbol{C} = 0$ iff there exists a non-zero vector $\boldsymbol{x}$ such that $\boldsymbol{C}\boldsymbol{x} = \boldsymbol{0}$. We can utilize this fact by outputting $[\boldsymbol{x}]_2$ as a hint to the verifier/soundness reduction. However, $[\boldsymbol{x}]_2$ can reveal secret information and thus must be hidden. We do not want to encrypt $[\boldsymbol{x}]_2$: since $[\boldsymbol{x}]_2$ is given in $\mathbb{G}_2$, this means that one again needs to use DLIN.

Fourth, we rely on CED as follows: recall that CED states that $\det \boldsymbol{C} = 0$ iff one can compute vectors $\boldsymbol{\gamma}$ and $\boldsymbol{\delta}$ such that $\boldsymbol{C}\left(\begin{smallmatrix}\mathsf{e}\\ \boldsymbol{\delta}\end{smallmatrix}\right) = \boldsymbol{\gamma}$, where $\mathsf{e} \leftarrow_{\$} \mathbb{Z}_p$. (The

[5] In the case of 2×2 matrices, the hint can be $[\boldsymbol{C}_1]_2$, where $\boldsymbol{C}_1$ is the first row of $\boldsymbol{C} \in \mathbb{Z}_p^{2\times 2}$ [ALSZ20]. In the case of a 3×3 matrix $\boldsymbol{C}$, the prover already needs to output six values $[C_{1i}C_{2j}]_2$ for $i \neq j$.

first coordinate of $\boldsymbol{x} = \binom{\mathsf{e}}{\boldsymbol{\delta}}$ is non-zero w.p. $1 - 1/p$ since $\boldsymbol{C}$ is a QDR.) For the security of CED, $\boldsymbol{\gamma}$ must not depend on e. Here, as in [CH20,CLPØ21], $\boldsymbol{\delta}$ is masked by uniformly random addend $\boldsymbol{\varrho_\delta}$ and $\boldsymbol{\gamma}$ is needed to balance $\boldsymbol{\varrho_\delta}$. Thus, the prover gives $([\boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$ as a hint to the verifier/soundness reduction. In the NIZK, $[\boldsymbol{\gamma}]_1$ is encrypted but $[\boldsymbol{\delta}]_2$ (that looks uniformly random after adding $\boldsymbol{\varrho_\delta}$) is not. While the resulting accumulator is less efficient than Nguyen's, the new NIZK (see Section 7) is very efficient since it reuses the hints $([\boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$.

4.2 Definition

The reasoning from Section 4.1 shows that one can construct an efficient accumulator (and NIZK) even if $\boldsymbol{\chi}$ is only given to the verifier in one source group. This motivates the new definition of determinantal accumulators. For comparison purposes only, we will first define structure-preserving signature schemes [AFG$^+$16].

Definition 2 (Structure-preserving signature scheme [AFG$^+$16]). *A digital signature scheme is* structure preserving *relative to bilinear group generator* Pgen *if (1) the common parameters* p *consist of group description generated by* Pgen*, some constants, and some source group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, (2) the verification algorithm* V *consists only of evaluating membership in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *and relations described by paring product equations, (3) verification keys* vk*, messages* χ *and signatures* σ *solely consist of group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*.*

Our definition of determinantal accumulators is very close in spirit. For clarity, we highlight the differences between "structure preserving" and "determinantal" primitives. Other differences are caused by having an accumulator instead of a signature scheme.

Definition 3 (Determinantal accumulator). *An accumulator is* determinantal *relative to bilinear group generator* Pgen *if*

(a) the common parameters p *consist of group description generated by* Pgen*, some constants, and some source group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*,*

(b) the verification algorithm V *consists only of evaluating membership in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$ *and relations described by checking that* $\boldsymbol{C}_i(\chi) = 0$*, where each* $\boldsymbol{C}_i(X)$ *is a QDR,*

(c) the CRS $\mathtt{crs}$*, messages* χ*, commitments* $\mathsf{C}_\mathcal{S}$*, and membership arguments* ψ *solely consist of group elements in* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*,*

(d) messages χ *are given to the verifier as elements of* $\mathbb{G}_1$*,*

(e) the set of $\mathbb{G}_2$ *elements in* ψ *is independent of* χ*.*

Items d and e help creating efficient NIZKs, where one only has to encrypt elements of $\mathbb{G}_1$. We assume that all determinantal accumulators use the fourth method from Section 4.1. Since in that case, the only $\mathbb{G}_2$ element in ψ is $\boldsymbol{\delta}$ and the latter is chosen uniformly from $\mathbb{G}_2$ in [CLPØ21], Item e follows automatically.

Clearly, this approach is not restricted to accumulators.

Comparison to Structure-Preserving Primitives (SPPs). Determinantal primitives are quite different from SPPs. First, compared to SPPs, we restrict the inputs to be from a single source group. While this is a restriction, it potentially boosts efficiency: since all inputs have to be encrypted in one source group, one can use Elgamal instead of less efficient DLIN or Groth-Sahai commitments. Because $\mathbb{G}_2$ elements are often twice longer than $\mathbb{G}_1$ elements, this can make the statement of the NIZK (commitment to χ) three times shorter.

Second, the verifier is not restricted to quadratic equations: the QDRs $\boldsymbol{C}_i$ can be polynomially large. In the new non-membership accumulator, the determinant of the used $\boldsymbol{C}_i$ is a cubic polynomial. This means that some of the known lower-bounds for SPPs (e.g., [AFG$^+$16]) *might* not apply.

Third, and crucially, determinantal accumulators are (efficient) CLPØ-style non-zk non-interactive argument systems. On the other hand, structure-preserving signatures are independent primitives with the property that one can construct (efficient) Groth-Sahai NIZKs for tasks like signature possession. It is not known how to construct structure-preserving accumulators.

5 The New Determinantal Accumulator AC^*

5.1 F-Collision-Resistance

In the new set (non-)membership NIZK, χ is Elgamal-encrypted. In the soundness reduction, the reduction decrypts it to obtain $[\chi]_1$ but does not obtain χ. Because of that, the collision-resistance property must hold against adversaries who return $[\chi]_1$ but not χ. Definition 4 is inspired by the definition of F-unforgeable signature schemes, [BCKL08], where F is an efficiently computable one-way bijection. Since F is a bijection, $\chi \in \mathcal{S}$ iff $F(\chi) \in F(\mathcal{S})$ iff $\exists s \in \mathcal{S}.F(\chi) = F(s)$.

Definition 4. *Let $\mathcal{D}$ be a domain and F be an efficiently computable (one-way) bijection. A universal accumulator* ACC *is F-*collision resistant *if for any $N = \mathsf{poly}(\lambda)$ and PPT adversaries $\mathcal{A}$,* $\mathsf{Adv}^{\mathsf{f-cr}}_{\mathsf{Pgen},F,\mathsf{ACC},\mathcal{A}}(\lambda) :=$

$$\Pr\left[\begin{array}{c|c} \mathcal{S} \in \mathcal{D}^{\leq N} \wedge & \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); (\mathtt{crs}, \sigma) \leftarrow \mathsf{Kgen}(\mathsf{p}, N); \\ \left(\begin{array}{c} (\chi \notin \mathcal{S} \wedge v = \mathsf{Member}) \vee \\ (\chi \in \mathcal{S} \wedge v = \mathsf{NotMember}) \end{array}\right) & (\mathcal{S}, F(\chi), \psi) \leftarrow \mathcal{A}(\mathtt{crs}); \\ & v \leftarrow \mathsf{V}(\mathtt{crs}, \mathsf{Com}(\mathtt{crs}, \mathcal{S}), F(\chi), \psi) \end{array}\right] \approx_c 0 \enspace .$$

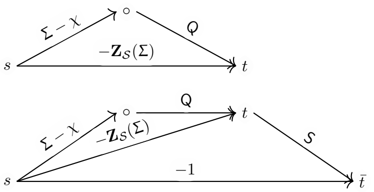
Here, we highlighted the differences with Definition 1.

In what follows, $F = [\cdot]_1$.

5.2 Construction

In Fig. 7, we propose a new F-collision-resistant determinantal (CRS-model, universal) accumulator AC^*. Next, we give the intuition behind its construction.

The first task constructing AC^* is to fix suitable verification equation that defines a polynomial $F(\boldsymbol{X})$, such that the verifier accepts iff $F(\boldsymbol{\chi}) = 0$. Given F, we use an ABP to define a QDR $\boldsymbol{C}(\boldsymbol{X})$ for F.



$$\boldsymbol{C}_\Sigma(\mathsf{X},\mathsf{Q}) = \begin{pmatrix} \Sigma - \mathsf{X} & -1 \\ -\mathbf{Z}_{\mathcal{S}}(\Sigma) & \mathsf{Q} \end{pmatrix}$$

$$\bar{\boldsymbol{C}}_\Sigma(\mathsf{X},\mathsf{Q},\mathsf{S}) = \begin{pmatrix} \Sigma - \mathsf{X} & -1 & 0 \\ -\mathbf{Z}_{\mathcal{S}}(\Sigma) & \mathsf{Q} & -1 \\ -1 & 0 & \mathsf{S} \end{pmatrix}$$

Fig. 6. Above: ABP for $F_\Sigma(\mathsf{X},\mathsf{Q})$ and the corresponding QDR $\boldsymbol{C}_\Sigma(\mathsf{X},\mathsf{Q})$. Below: ABP for $\bar{F}_\Sigma(\mathsf{X},\mathsf{Q},\mathsf{S})$ and the corresponding QDR $\bar{\boldsymbol{C}}_\Sigma(\mathsf{X},\mathsf{Q},\mathsf{S})$.

In the case of the membership argument, we start with the verification equation of $\mathsf{ACC}_{\mathsf{Nguyen}}$ from Fig. 2, which defines the bivariate polynomial

$$F_\Sigma(\mathsf{X},\mathsf{Q}) := (\Sigma - \mathsf{X})\mathsf{Q} - \mathbf{Z}_{\mathcal{S}}(\Sigma) \enspace .$$

Here, say, Q is the indeterminate corresponding to $\mathsf{q} \in \psi$ (see Fig. 2). Clearly, the membership argument verifier of $\mathsf{ACC}_{\mathsf{Nguyen}}$ accepts iff $[F_\sigma(\chi,\mathsf{q})]_1 = [0]_1$.

In the non-membership argument, we need to prove that $F_\Sigma(\mathsf{X},\mathsf{Q}) \neq 0$. We use the method of Section 3 by defining the polynomial

$$\tilde{F}_\Sigma(\mathsf{X},\mathsf{Q},\mathsf{S}) := ((\Sigma - \mathsf{X})\mathsf{Q} - \mathbf{Z}_{\mathcal{S}}(\Sigma))\,\mathsf{S} - 1 \enspace .$$

We index F and $\tilde{F}$ with Σ instead of giving Σ as a formal argument. We do it because Σ (a trapdoor indeterminate, with various powers like $[\sigma^i]_1$ being present in the CRS) has a different semantics compared to indeterminates X, Q, and S that correspond to the argument elements. In particular, $[\sigma^i]_1$ do not have to stay hidden in the set (non-)membership NIZK. Crucially, this allows to think of F_Σ and $\tilde{F}_\Sigma$ as low-degree polynomials with coefficients from $\mathcal{R} = \mathbb{Z}_p[\Sigma]$.

Since F_Σ and $\tilde{F}_\Sigma$ have degrees ≤ 2 and ≤ 3, they have respectively 2×2 and 3×3 QDRs $\boldsymbol{C}_\Sigma(\mathsf{X},\mathsf{Q})$ and $\bar{\boldsymbol{C}}_\Sigma(\mathsf{X},\mathsf{Q},\mathsf{S})$. We construct these QDRs from algebraic branching programs for F_Σ and $\bar{F}_\Sigma$. See Fig. 6 for the description of the resulting ABP and QDR for F_Σ and $\bar{F}_\Sigma$. The membership (resp., non-membership) argument verifier needs to check that $\det \boldsymbol{C}(\chi,\mathsf{q}) = 0$ (resp., $\det \bar{\boldsymbol{C}}(\chi,\mathsf{q},\mathsf{s}) = 0$).

Membership Argument. Since we construct a determinantal accumulator, we check $\det \boldsymbol{C}(\chi,\mathsf{q}) = 0$ by using the hints $[\boldsymbol{\gamma}]_1$ and $[\delta]_2$. The membership argument verifier checks that $[\boldsymbol{C}(\boldsymbol{\chi})]_1 \bullet [\begin{smallmatrix} \mathsf{e} \\ \delta \end{smallmatrix}]_2 = [\boldsymbol{\gamma}]_1 \bullet [1]_2$, which can be rewritten as checking

$$\begin{aligned} ([\sigma]_1 - [\chi]_1) \bullet [\mathsf{e}]_2 - [1]_1 \bullet [\delta]_2 &= \ [\gamma_1]_1 \bullet [1]_2 \enspace , \\ -[\mathbf{Z}_{\mathcal{S}}(\sigma)]_1 \bullet [\mathsf{e}]_2 + [\mathsf{q}]_1 \bullet [\delta]_2 &= [\gamma_2]_1 \bullet [1]_2 \enspace . \end{aligned} \tag{2}$$

Here, $[\chi]_1$ is the input, $([\mathsf{q},\gamma]_1, [\delta]_2)$ are parts of the (non-)membership argument, and $[\sigma, \mathbf{Z}_{\mathcal{S}}(\sigma)]_1$ can be computed from $\mathtt{crs}$.

Unfortunately, this is not sufficient. A maliciously chosen $\chi = \chi(\Sigma)$, $\mathsf{q} = \mathsf{q}(\Sigma)$, and $\delta = \delta(\Sigma)$ can depend non-trivially on σ. In a AGM security proof, Eq. (2) guarantees that $\mathbf{Z}_{\mathcal{S}}(\Sigma) = (\Sigma - \chi(\Sigma))\mathsf{q}(\Sigma)$ and thus $(\Sigma - \chi(\Sigma)) \mid \mathbf{Z}_{\mathcal{S}}(\Sigma)$. If χ is an integer, then from this we will get that $\mathbf{Z}_{\mathcal{S}}(\chi) = 0$. However, if χ is not an integer (it depends on σ), then $\mathbf{Z}_{\mathcal{S}}(\chi) = 0$ does not follow. For example, to break the membership argument, the adversary can fix any $\delta_1, \delta_2 \in \mathbb{Z}_p$ and set $[\chi]_1 \leftarrow [\sigma]_1 - \delta_2[1]_1$, $[\delta]_2 \leftarrow \delta_1[1]_2 + \delta_2[\mathsf{e}]_2$, $[\mathsf{q}]_1 \leftarrow [\mathbf{Z}_{\mathcal{S}}(\sigma)]_1/\delta_2$, $[\gamma_1]_1 \leftarrow -[\delta_1]_1$, $[\gamma_2]_1 \leftarrow \delta_1/\delta_2 \cdot [\mathbf{Z}_{\mathcal{S}}(\sigma)]_1$. This results in Eq. (2) holding and thus breaks the F-collision-resistance of the version of AC^* that only uses Eq. (2) as verification equations. Breaking F-collision-resistance of $\mathsf{ACC}_{\mathsf{Nguyen}}$ is even more trivial.[6]

To counteract this problem, we must guarantee that χ does not depend on σ. We do this by introducing an additional trapdoor τ. We then slightly modify Eq. (2), making the checks explicitly dependent on τ. The resulting modified checks result in b_1 and b_2 in the final construction of AC^* in Fig. 7.

Since now $\mathtt{crs}$ depends on τ, the adversary can make its outputs depend on τ; this opens a new cheating avenue. Hence, our use of τ is non-trivial, especially since we achieve F-collision-resistance without hampering the efficiency of AC^*. We explicitly multiply each term of type $[\alpha]_1 \bullet [\beta]_2$ in b_1 and b_2 by τ, except the terms $[\mathsf{q}]_1 \bullet [\delta]_2$ and $[\boldsymbol{\gamma}]_1 \bullet [1]_2$. In the AGM security proof, we get that values like χ, which are multiplied by τ, are in the span of 1 (that is, integers). However, q must be a polynomial (it depends on σ), that is, in the span of $\{\sigma^i \tau\}$; thus we do not multiply $[\mathsf{q}]_1 \bullet [\delta]_2$ by τ. The same holds for γ_2. Finally, it is not essential whether γ_1 depends on σ or not; not multiplying it by τ simplifies the AGM proof slightly since then we do not need to add $[\tau]_2$ to the CRS. Nevertheless, the AGM proof is very delicate.

Note that the verification equations ($b_1 = b_2 = \mathsf{true}$) are mathematically (but not computationally) equivalent to checking that $\boldsymbol{C}'(\chi, \mathsf{q})\left(\begin{smallmatrix} \mathsf{e} \\ \delta \end{smallmatrix}\right) = \boldsymbol{\gamma}$, where

$$\boldsymbol{C}'(\mathsf{X}, \mathsf{Q}) := \left(\begin{smallmatrix} (\Sigma - \mathsf{X})\mathsf{T} & -\mathsf{T} \\ -\mathbf{Z}_{\mathcal{S}}(\Sigma)\mathsf{T} & \mathsf{Q} \end{smallmatrix}\right) .$$

Here, $\det \boldsymbol{C}'(\mathsf{X}, \mathsf{Q}) = ((\Sigma - \mathsf{X})\mathsf{Q} - \mathbf{Z}_{\mathcal{S}}(\Sigma)\mathsf{T})\,\mathsf{T}$. That is, we really use the QDR framework of [CLPØ21]. The description of V in Fig. 7 just spells out how to do this verification in PPT.

Non-Membership Argument. The non-membership argument verifier must check that $[\bar{\boldsymbol{C}}(\boldsymbol{\chi})]_1 \bullet [\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix}]_2 = [\boldsymbol{\gamma}]_1 \bullet [1]_2$ (where now $\boldsymbol{\delta} \in \mathbb{Z}_p^2$ and $\boldsymbol{\gamma} \in \mathbb{Z}_p^3$; see Fig. 6), which can be rewritten as three checks

$$\begin{aligned} ([\sigma]_1 - [\chi]_1) \bullet [\mathsf{e}]_2 - [1]_1 \bullet [\delta]_2 &= [\gamma_1]_1 \bullet [1]_2 \ , \\ -[\mathbf{Z}_{\mathcal{S}}(\sigma)]_1 \bullet [\mathsf{e}]_2 + [\mathsf{q}]_1 \bullet [\delta_1]_2 - [1]_1 \bullet [\delta_2]_2 &= [\gamma_2]_1 \bullet [1]_2 \ , \\ -[1]_1 \bullet [\mathsf{e}]_2 + [\mathsf{s}]_1 \bullet [\delta_2]_2 &= [\gamma_3]_1 \bullet [1]_2 \ . \end{aligned} \tag{3}$$

As in the case of the membership argument, we need to modify the first two equations by using τ. However, since we require s to be an integer, we do not have to modify the third verification equation.

[6] In the collision-resistance proof of $\mathsf{ACC}_{\mathsf{Nguyen}}$, χ and r are given as integers and thus do not depend on σ. Such a problem did also not exist in [CH20,CLPØ21] since there the CRS only contained a single element $[\mathsf{e}]_2$ and thus did not depend on σ.

```
$\mathsf{Kgen}(\mathsf{p}, N)$: $\sigma, \tau, \mathsf{e} \leftarrow_{\$} \mathbb{Z}_p$; $\mathtt{crs} \leftarrow ([1, (\sigma^i \tau)_{i=0}^{N}]_1, [1, \mathsf{e}, \sigma\mathsf{e}, \tau\mathsf{e}]_2)$; $\mathtt{td} \leftarrow (\mathsf{e}, \tau)$;
    return $(\mathtt{crs}, \mathtt{td})$.

$\mathsf{Com}(\mathtt{crs}, \mathcal{S})$: $[\mathsf{C}_{\mathcal{S}}]_1 \leftarrow [\mathbf{Z}_{\mathcal{S}}(\sigma)\tau]_1$; return $\mathtt{crs}_{\mathtt{lpar}} \leftarrow (\mathtt{crs}, [\mathsf{C}_{\mathcal{S}}]_1, \mathcal{S})$;

$\mathsf{P}(\mathtt{crs}_{\mathtt{lpar}}, \chi)$: $\mathsf{r} \leftarrow \mathbf{Z}_{\mathcal{S}}(\chi)$; $f(X) \leftarrow (\mathbf{Z}_{\mathcal{S}}(X) - \mathsf{r})/(X - \chi)$; $[\mathsf{q}]_1 \leftarrow [f(\sigma)\tau]_1$;
    if $\chi \in \mathcal{S}$ then
     1. $\varrho_\delta \leftarrow_{\$} \mathbb{Z}_p$; $[\boldsymbol{\gamma}]_1 \leftarrow -\left[\begin{smallmatrix}-\tau \\ \mathsf{q}\end{smallmatrix}\right]_1 \varrho_\delta$; $[\delta]_2 \leftarrow [\sigma\mathsf{e}]_2 - \chi[\mathsf{e}]_2 - \varrho_\delta[1]_2$;
     2. $\psi \leftarrow ([\mathsf{q}, \boldsymbol{\gamma}]_1, [\delta]_2)$;    // $3\mathfrak{g}_1 + \mathfrak{g}_2$
    else
     1. $\mathsf{s} \leftarrow \frac{1}{\mathsf{r}}$; $\boldsymbol{\varrho_\delta} \leftarrow_{\$} \mathbb{Z}_p^2$; $[\boldsymbol{\gamma}]_1 \leftarrow -\left[\begin{smallmatrix}-\tau & 0 \\ \mathsf{q} & -\tau \\ 0 & \mathsf{s}\end{smallmatrix}\right]_1 \boldsymbol{\varrho_\delta}$; $[\boldsymbol{\delta}]_2 \leftarrow \left(\begin{smallmatrix}[\sigma\mathsf{e}]_2 - \chi[\mathsf{e}]_2 \\ \mathsf{r}\cdot[\mathsf{e}]_2\end{smallmatrix}\right) - \boldsymbol{\varrho_\delta}[1]_2$;
     2. $\psi \leftarrow ([\mathsf{q}, \mathsf{s}, \boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$;    // $5\mathfrak{g}_1 + 2\mathfrak{g}_2$
    return $\psi$;

$\mathsf{V}(\mathtt{crs}_{\mathtt{lpar}}, [\chi]_1, \psi)$ : $mem \leftarrow \mathsf{NotMember}$;
    If $\psi$ parses as $\psi = ([\mathsf{q}, \boldsymbol{\gamma}]_1, [\delta]_2)$ then $mem \leftarrow \mathsf{Member}$;
    If $mem = \mathsf{Member}$ then
     1. $b_1 \leftarrow [\sigma\tau]_1 \bullet [\mathsf{e}]_2 - [\chi]_1 \bullet [\tau\mathsf{e}]_2 - [\tau]_1 \bullet [\delta]_2 =^? [\gamma_1]_1 \bullet [1]_2$;
     2. $b_2 \leftarrow -[\mathsf{C}_{\mathcal{S}}]_1 \bullet [\mathsf{e}]_2 + [\mathsf{q}]_1 \bullet [\delta]_2 =^? [\gamma_2]_1 \bullet [1]_2$;
     3. if $b_1$ and $b_2$ then return $\mathsf{Member}$ else return $\mathsf{Error}$;
    else
     1. $\bar{b}_1 \leftarrow ([\sigma]_1 - [\chi]_1) \bullet [\tau\mathsf{e}]_2 - [\tau]_1 \bullet [\delta_1]_2 =^? [\gamma_1]_1 \bullet [1]_2$;
     2. $\bar{b}_2 \leftarrow -[\mathsf{C}_{\mathcal{S}}]_1 \bullet [\mathsf{e}]_2 + [\mathsf{q}]_1 \bullet [\delta_1]_2 - [\tau]_1 \bullet [\delta_2]_2 =^? [\gamma_2]_1 \bullet [1]_2$;
     3. $\bar{b}_3 \leftarrow -[1]_1 \bullet [\mathsf{e}]_2 + [\mathsf{s}]_1 \bullet [\delta_2]_2 =^? [\gamma_3]_1 \bullet [1]_2$;
     4. if $\bar{b}_1$ and $\bar{b}_2$ and $\bar{b}_3$ then return $\mathsf{NotMember}$ else return $\mathsf{Error}$;
```

Fig. 7. The new $[\cdot]_1$-collision-resistant determinantal universal accumulator AC^*.

The verification equations (that is, $\bar{b}_1 = \bar{b}_2 = \bar{b}_3 = \mathsf{true}$, see Fig. 7) are equivalent to checking that $\bar{\boldsymbol{C}}'(\chi, \mathsf{q}, \mathsf{s})\left(\begin{smallmatrix}\mathsf{e} \\ \boldsymbol{\delta}\end{smallmatrix}\right) = \boldsymbol{\gamma}$, where

$$\bar{\boldsymbol{C}}'(\mathsf{X}, \mathsf{Q}, \mathsf{S}) := \begin{pmatrix} (\Sigma - \mathsf{X})\mathsf{T} & -\mathsf{T} & 0 \\ -\mathbf{Z}_{\mathcal{S}}(\Sigma)\mathsf{T} & \mathsf{Q} & -\mathsf{T} \\ -1 & 0 & \mathsf{S} \end{pmatrix} ,$$

with $\det \bar{\boldsymbol{C}}'(\mathsf{X}, \mathsf{Q}) = ((\Sigma - \chi)\mathsf{Q} - \mathbf{Z}_{\mathcal{S}}(\Sigma)\mathsf{T})\,\mathsf{s}\mathsf{T} - \mathsf{T}^2$.

Description. We depict AC^* in Fig. 7. As explained before, the membership verifier checks (on pairings) that $\boldsymbol{C}'(\chi, \mathsf{q}) \cdot \left(\begin{smallmatrix}\mathsf{e} \\ \delta\end{smallmatrix}\right) = \boldsymbol{\gamma}$, and the non-membership verifier checks that $\bar{\boldsymbol{C}}'(\chi, \mathsf{q}, \mathsf{s}) \cdot \left(\begin{smallmatrix}\mathsf{e} \\ \boldsymbol{\delta}\end{smallmatrix}\right) = \boldsymbol{\gamma}$. Fig. 7 does it in PPT.

Lemma 1. AC^* *is perfectly complete.*

Proof. One can straightforwardly check that the choice of $\boldsymbol{\varrho_\delta}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\delta}$ is consistent with Fig. 3 when one uses the correct matrices $\boldsymbol{C}'$ and $\bar{\boldsymbol{C}}'$. Completeness follows straightforwardly. In particular, writing $\boldsymbol{C}' = (\boldsymbol{h}' \| \boldsymbol{T}')$, we get that $\boldsymbol{h}' = \boldsymbol{T}'\mathsf{w}'$, where $\mathsf{w}' = -(\sigma - \chi)$. This explains why say $[\delta]_2 = -\mathsf{w}'[\mathsf{e}]_2 - \varrho_\delta[1]_2 = [(\sigma - \chi)\mathsf{e}]_2 - \varrho_\delta[1]_2 = [\sigma\mathsf{e}]_2 - \chi[\mathsf{e}]_2 - \varrho_\delta[1]_2$. Then, say $b_1 = \mathsf{true}$

since $(\sigma - \chi)\tau\mathsf{e} - \tau\delta = \gamma_1 \iff (\sigma - \chi)\tau\mathsf{e} - \tau((\sigma - \chi)\mathsf{e} - \varrho_\delta) = \tau\varrho_\delta$, which is trivially true. In the case of non-membership proof, writing $\bar{\boldsymbol{C}}' = (\bar{\boldsymbol{h}}' \| \bar{\boldsymbol{T}}')$, we get similarly that $\bar{\boldsymbol{h}}' = \bar{\boldsymbol{T}}'\bar{\mathbf{w}}'$, where $\bar{\mathbf{w}}' = \left(\begin{smallmatrix} -(\sigma-\chi) \\ -\mathsf{r} \end{smallmatrix}\right)$. □

On Semantics of Non-Membership. Recall that AC^* must be F-collision-resistant. Since the CRS contains trapdoor-dependent elements, one must make it precise how to define non-membership. As a motivating example, if $\mathcal{S} = \{0, 1\}$, then $[\chi]_1 \leftarrow [\sigma]_1$ satisfies $\chi \in \mathcal{S}$ iff $\sigma \in \{0, 1\}$. The AGM security proof handles σ as an indeterminate, and thus it cannot decide whether σ (or, more generally, some known affine map of σ) belongs to $\mathcal{S}$. To avoid such artefacts, we constructed AC^* so that the verifier returns Error when the prover makes $[\chi]_1$ to depend on $[\sigma]_1$ (see the proof of Theorems 1 and 2). While we do not do it here, it allows one to define the extractability of the accumulator naturally; from the proof of Theorems 1 and 2, it is easy to see that AC^* is extractable.

6 AC^*'s F-Collision-Resistance

The actual F-collision-resistance proof is complicated. We first define two tautological assumptions N-DETACM and N-$\mathsf{DETACNM}$ that essentially state that AC^* is F-collision-resistant against adversaries that try to create fake membership (resp., non-membership) arguments. After that, we prove in AGM that DETACM and $\mathsf{DETACNM}$ reduce to PDL.

The most efficient structure-preserving signatures are proven to be secure in the AGM (or in the generic group model), though the assumption of their security by itself is a falsifiable assumption. We can similarly prove the security of AC^* in AGM. However, the collision-resistance of an accumulator is a much simpler (in particular, it is non-interactive) assumption than the unforgeability of a signature scheme and thus the tautological assumption looks less intimidating.

6.1 DETACM And DETACNM

Next, we define assumptions N-DETACM and N-$\mathsf{DETACNM}$.

Definition 5. *Let $\mathcal{A}$ be a PPT adversary. Let $N = \mathsf{poly}(\lambda)$. N-DETACM holds relative to Pgen, if for every PPT $\mathcal{A}$,*

$$\Pr\left[\begin{array}{c|c} & \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \sigma, \tau, \mathsf{e} \leftarrow_\$ \mathbb{Z}_p; \\ \mathcal{S} \in \mathcal{D}^{\leq N} \wedge & \mathtt{crs} \leftarrow (\mathsf{p}, [1, (\sigma^i\tau)_{i=0}^N]_1, [1, \mathsf{e}, \sigma\mathsf{e}, \tau\mathsf{e}]_2); \\ \chi \notin \mathcal{S} \wedge & (\mathcal{S}, [\chi, \mathsf{q}, \boldsymbol{\gamma}]_1, [\delta]_2) \leftarrow \mathcal{A}(\mathtt{crs}); \\ \boldsymbol{C}'(\chi, \mathsf{q})\left(\begin{smallmatrix} \mathsf{e} \\ \delta \end{smallmatrix}\right) = \boldsymbol{\gamma} & \boldsymbol{C}'(\chi, \mathsf{q}) \leftarrow \left(\begin{smallmatrix} (\sigma-\chi)\tau & -\tau \\ -\mathbf{Z}_\mathcal{S}(\sigma)\tau & \mathsf{q} \end{smallmatrix}\right) \end{array}\right] \approx_c 0 \enspace .$$

N-$\mathsf{DETACNM}$ holds relative to Pgen, if for every PPT $\mathcal{A}$,

$$\Pr\left[\begin{array}{c|c} & \mathsf{p} \leftarrow \mathsf{Pgen}(1^\lambda); \sigma, \tau, \mathsf{e} \leftarrow_\$ \mathbb{Z}_p; \\ \mathcal{S} \in \mathcal{D}^{\leq N} \wedge & \mathtt{crs} \leftarrow (\mathsf{p}, [1, (\sigma^i\tau)_{i=0}^N]_1, [1, \mathsf{e}, \sigma\mathsf{e}, \tau\mathsf{e}]_2); \\ \chi \in \mathcal{S} \wedge & (\mathcal{S}, [\chi, \mathsf{q}, \mathsf{s}, \boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2) \leftarrow \mathcal{A}(\mathtt{crs}); \\ \bar{\boldsymbol{C}}'(\chi, \mathsf{q}, \mathsf{s})\left(\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix}\right) = \boldsymbol{\gamma} & \bar{\boldsymbol{C}}'(\chi, \mathsf{q}, \mathsf{s}) \leftarrow \left(\begin{smallmatrix} (\sigma-\chi)\tau & -\tau & 0 \\ -\mathbf{Z}_\mathcal{S}(\sigma)\tau & \mathsf{q} & -\tau \\ -1 & 0 & \mathsf{s} \end{smallmatrix}\right) \end{array}\right] \approx_c 0 \enspace .$$

```
$\mathcal{B}(\mathtt{crs} = (\mathsf{p}, [1, (\sigma^i \tau)_{i=0}^N]_1, [1, \mathsf{e}, \sigma\mathsf{e}, \tau\mathsf{e}]_2))$
$(\mathcal{S}, [\chi]_1, \psi) \leftarrow \mathcal{A}(\mathtt{crs})$;
return $(\mathcal{S}, [\chi]_1, \psi)$; endif
```

Fig. 8. The adversary $\mathcal{B}$ in the proof of Lemma 2

Compared to CED, DETACM and DETACNM do not rely on the (possibly, inefficiently verifiable) condition that $\boldsymbol{C}(\boldsymbol{\chi})$ has a full rank. Thus, importantly, DETACM and DETACNM are efficiently verifiable and thus falsifiable. For example, as explained above, the verification $\bar{\boldsymbol{C}}'(\chi, \mathsf{q}, \mathsf{s})\left(\begin{smallmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{smallmatrix}\right) = \boldsymbol{\gamma}$ is equivalent to checking that $\bar{b}_1$, $\bar{b}_2$, and $\bar{b}_3$ hold. Thus, it can be checked efficiently.

6.2 $\boldsymbol{F}$-Collision-Resistance of AC*

Lemma 2 is trivial since DETACM and DETACNM are tautological assumptions for the F-collision-resistance of AC^*. The complicated step (see Section 6.3) is establishing that DETACM and DETACNM are secure in the AGM.

Lemma 2. *Let $F = [\cdot]_1$ and $N = \mathsf{poly}(\lambda)$. AC^* is F-collision-resistant under N-DETACM and N-DETACNM.*

Proof. Let $\mathcal{A}$ be an F-collision-resistance (see Definition 4) adversary for AC^*, such that $\mathsf{Adv}^{\mathrm{f-cr}}_{\mathsf{Pgen}, F, \mathsf{AC}^*, \mathcal{A}}(\lambda) = \varepsilon_{\mathcal{A}}$ for non-negligible $\varepsilon_{\mathcal{A}}$. In Fig. 8, we depict a trivial DETACM/DETACNM adversary $\mathcal{B}$. Clearly, with probability at least $\varepsilon_{\mathcal{A}}$, $\mathcal{B}$ succeeds in breaking N-DETACM (resp., N-DETACNM), given $\mathcal{A}$ outputs an accepting fake membership (resp., non-membership) argument. □

6.3 AGM Security of DETACM And DETACNM

Theorem 1. *If $(N+1, 2)$-PDL holds, then N-DETACM is secure in the AGM.*

Proof. Let $\mathcal{A}_{\mathrm{alg}}$ be an algebraic DETACM adversary. Assume that $\mathcal{A}_{\mathrm{alg}}(\mathtt{crs})$ outputs $\psi = (\mathcal{S}, [\chi, \mathsf{q}, \boldsymbol{\gamma}]_1, [\delta]_2)$, such that V accepts with a non-negligible probability. Since $\mathcal{A}_{\mathrm{alg}}$ is algebraic, with every group element $G \in \mathbb{G}_\iota$, it also outputs a vector $\boldsymbol{a}$ explaining how G is constructed from the elements of $\mathtt{crs}$ that belong to $\mathbb{G}_i$. Next, we will make this more precise.

Let $\boldsymbol{X} = (\Sigma, \mathsf{T}, \mathsf{E})$ and $\boldsymbol{x} = (\sigma, \tau, \mathsf{e})$. Here, say T is the indeterminate corresponding to the trapdoor τ. We express each output of $\mathcal{A}_{\mathrm{alg}}$ as a polynomial evaluation, with say $[\chi]_1 = [\chi(\boldsymbol{x})]_1$. The involved polynomials are

$$\begin{aligned} \chi(\boldsymbol{X}) &= \chi_1(\Sigma)\mathsf{T} + \chi_2 \ , & \mathsf{q}(\boldsymbol{X}) &= \mathsf{q}_1(\Sigma)\mathsf{T} + \mathsf{q}_2 \ , \\ \gamma_1(\boldsymbol{X}) &= \gamma_{11}(\Sigma)\mathsf{T} + \gamma_{12} \ , & \gamma_2(\boldsymbol{X}) &= \gamma_{21}(\Sigma)\mathsf{T} + \gamma_{22} \ , \\ \delta(\boldsymbol{X}) &= \delta_1 + \delta_2\mathsf{E} + \delta_3\Sigma\mathsf{E} + \delta_4\mathsf{T}\mathsf{E} \ , \end{aligned}$$

where each polynomial (like q_1) on the RHS is of degree $\leq N$. That is, the algebraic adversary $\mathcal{A}_{\mathrm{alg}}$ also outputs coefficients of all above polynomials. The DETACM verifier's checks guarantee that $V_1(\sigma, \tau, \mathsf{e}) = V_2(\sigma, \tau, \mathsf{e}) = 0$, where

$$\begin{aligned}
V_1(\boldsymbol{X}) &= ((\Sigma - \chi(\boldsymbol{X}))\,\mathsf{E} - \delta(\boldsymbol{X})) \cdot \mathsf{T} - \gamma_1(\boldsymbol{X}) \ , \\
V_2(\boldsymbol{X}) &= (\mathsf{r}(\boldsymbol{X}) - \mathbf{Z}_{\mathcal{S}}(\Sigma))\,\mathsf{TE} + \mathsf{q}(\boldsymbol{X})\delta(\boldsymbol{X}) - \gamma_2(\boldsymbol{X}) \ .
\end{aligned}$$

Consider separately the cases (1) $V_1 = V_2 = 0$ as polynomials, and (2) either $V_1 \neq 0$ or $V_2 \neq 0$.

Case 1. First, assume $V_1 = V_2 = 0$ as a polynomial. Think of the polynomials as members of $\mathcal{R}[\mathsf{T}, \mathsf{E}]$, where $\mathcal{R} = \mathbb{Z}_p[\Sigma]$. We now enlist the coefficients of $\mathsf{T}^i\mathsf{E}^j$ in both V_1 and V_2, highlighting the coefficients that are actually needed in this proof (we give other coefficients only for the sake of completeness):

(i,j)	V_1
$(2,1)$	$-\delta_4 - \chi_1(\Sigma)$
$(1,1)$	$-\delta_2 + (1 - \delta_3)\,\Sigma - \chi_2$
$(1,0)$	$-\gamma_{11}(\Sigma) - \delta_1$
$(0,0)$	$-\gamma_{12}$

(i,j)	V_2
$(2,1)$	$\delta_4\mathsf{q}_1(\Sigma)$
$(1,1)$	$\delta_4\mathsf{q}_2 + (\delta_2 + \delta_3\Sigma)\,\mathsf{q}_1(\Sigma) - \mathbf{Z}_{\mathcal{S}}(\Sigma)$
$(1,0)$	$\delta_1\mathsf{q}_1(\Sigma) - \gamma_{21}(\Sigma)$
$(0,1)$	$(\delta_2 + \delta_3\Sigma)\mathsf{q}_2$
$(0,0)$	$-\gamma_{22} + \delta_1\mathsf{q}_2$

For example, the coefficient of $\mathsf{T}^2\mathsf{E}^1 = \mathsf{T}^2\mathsf{E}$ in V_1 is $-\delta_4 - \chi_1(\Sigma)$. Since $V_i = 0$ as a polynomial, the coefficient of any monomial $\mathsf{T}^j\mathsf{E}^k$ in any V_i is also 0.

From the coefficient of $\mathsf{T}^2\mathsf{E}$ of V_1, we get $\chi_1(\Sigma) = -\delta_4$. From the coefficient of TE of V_1, after separating the coefficients of different Σ^i, we get $\delta_3 = 1$ and $\delta_2 = -\chi_2$. From the coefficient of $\mathsf{T}^2\mathsf{E}$ of V_2, we get $\delta_4\mathsf{q}_1(\Sigma) = 0$. Thus, either $\mathsf{q}_1(\Sigma) = 0$ or $\delta_4 = 0$. Taking into account what we already know, from the coefficient of TE of V_2, we get $\mathbf{Z}_{\mathcal{S}}(\Sigma) = \delta_4\mathsf{q}_2 + (\Sigma - \chi_2)\,\mathsf{q}_1(\Sigma)$. Recall that we have either $\mathsf{q}_1(\Sigma) = 0$ or $\delta_4 = 0$. If $\mathsf{q}_1(\Sigma) = 0$, then $\mathbf{Z}_{\mathcal{S}}(\Sigma) = \delta_4\mathsf{q}_2 \in \mathbb{Z}_p$, a contradiction. Hence, $\delta_4 = 0$. Thus, $\mathbf{Z}_{\mathcal{S}}(\Sigma) = (\Sigma - \chi_2)\,\mathsf{q}_1(\Sigma)$ and $(\Sigma - \chi_2) \mid \mathbf{Z}_{\mathcal{S}}(\Sigma)$, which gives us $\mathbf{Z}_{\mathcal{S}}(\chi_2) = 0$. Moreover, $\chi(\boldsymbol{X}) = \chi_1(\Sigma)\mathsf{T} + \chi_2 = \chi_2$, and thus we have proven AGM security in Case 1.

Case 2. The case $V_i \neq 0$ for some i can be handled in a standard way. Assume for example that $V_2 \neq 0$. We construct a PDL reduction $\mathcal{B}(\{[\sigma^i]_1\}_{i=0}^{N+1}, \{[\sigma^i]_1\}_{i=0}^{2})$. $\mathcal{B}$ samples $\alpha_1, \alpha_2, \beta_1, \beta_2 \leftarrow_{\$} \mathbb{Z}_p$ and sets implicitly $\tau \leftarrow \alpha_1\sigma + \beta_1$ and $\mathsf{e} \leftarrow \alpha_2\sigma + \beta_2$. Then, $\mathcal{B}$ creates $\mathtt{crs}$ for the DETACM adversary $\mathcal{A}_{\mathrm{alg}}$ and calls $\mathcal{A}_{\mathrm{alg}}$ with $\mathtt{crs}$. After obtaining π, together with the coefficients of the polynomials like $\chi(\Sigma)$, from $\mathcal{A}_{\mathrm{alg}}$, $\mathcal{B}$ reconstructs the coefficients of the degree-$\leq (N+2)$ polynomial V_2 (which is now univariate since τ and e are affine maps of σ). We know $V_2 \neq 0$ but $V_2(\sigma) = 0$. $\mathcal{B}$ factorizes V_2 and finds up to $N + 2$ roots x_i of V_2. $\mathcal{B}$ tests which one of them is equal to σ, and returns σ. □

Theorem 2. *If* $(N+1, 2)$-PDL *holds, then* N-$\mathsf{DETACNM}$ *is secure in the AGM.*

We postpone the proof of this theorem to Appendix A.1.

7 New Set (Non-)Membership NIZK

Next, we use AC^* to construct a succinct set (non-)membership NIZK $\mathbf{\Pi}^*$. First, $\mathbf{\Pi}^*$'s CRS is equal to AC^*'s CRS. Second, the NIZK prover proves that AC^*'s honest verifier accepts the encrypted χ and the encrypted accumulator argument $\psi = \mathsf{AC}^*.\mathsf{P}(\mathtt{crs}, \mathcal{S}, \chi)$. That is, the prover encrypts χ and ψ, and then proves that the verification equation is satisfied.

Description. Following the described blueprint, we construct the new set (non-)membership NIZK $\mathbf{\Pi}^*$ (see Fig. 9). $\mathbf{\Pi}^*$ handles both $\mathcal{L}^{\mathsf{sm}}_{\mathtt{lpar}}$ (set membership arguments, $mem = \mathsf{Member}$) and $\overline{\mathcal{L}}_{\mathtt{lpar}}$ (set non-membership arguments, $mem = \mathsf{NotMember}$). The prover of $\mathbf{\Pi}^*$ implements the prover of AC^* but it also additionally encrypts all $\mathbb{G}_1$. To make the verification on ciphertexts possible, the prover outputs addtional randomizer hints $[\boldsymbol{z}]_2$. The verifier performs AC^* verification on ciphertexts (this relies on the homomorphic properties of Elgamal), taking $[\boldsymbol{z}]_2$ into account. $\mathbf{\Pi}^*$ also defines the simulator algorithm.

Alternatively, $\mathbf{\Pi}^*$ is a version of $\mathbf{\Pi}_{\mathsf{clp}\varnothing}$ for the concrete choice of the QDRs (and different CRS). To see the connection between Fig. 9 and Fig. 3, note that

$$\boldsymbol{C}'(\mathsf{X}, \mathsf{Q}) = \underbrace{\begin{pmatrix} \Sigma\mathsf{T} & -1 \\ -\mathbf{Z}_{\mathcal{S}}(\Sigma)\mathsf{T} & 0 \end{pmatrix}}_{\boldsymbol{Q}} + \underbrace{\begin{pmatrix} -\mathsf{T} & 0 \\ 0 & 0 \end{pmatrix}}_{\boldsymbol{P}_1} \mathsf{X} + \underbrace{\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}}_{\boldsymbol{P}_2} \mathsf{Q} \ .$$

For example, starting with Fig. 3, $[\boldsymbol{z}]_2 = (\sum_{k=1}^{\nu} \varrho_k \boldsymbol{P}_k \left[\begin{smallmatrix} \mathsf{e} \\ \delta \end{smallmatrix}\right]_2) - \boldsymbol{\varrho}_\gamma [1]_2 = \varrho_\chi \left(\begin{smallmatrix} -\tau & 0 \\ 0 & 0 \end{smallmatrix}\right) \left[\begin{smallmatrix} \mathsf{e} \\ \delta \end{smallmatrix}\right]_2 + \varrho_{\mathsf{q}} \left(\begin{smallmatrix} 0 & 0 \\ 0 & 1 \end{smallmatrix}\right) \left[\begin{smallmatrix} \mathsf{e} \\ \delta \end{smallmatrix}\right]_2 - \boldsymbol{\varrho}_\gamma [1]_2 = \varrho_\chi \left[\begin{smallmatrix} -\tau\mathsf{e} \\ 0 \end{smallmatrix}\right]_2 + \varrho_{\mathsf{q}} \left[\begin{smallmatrix} 0 \\ \delta \end{smallmatrix}\right]_2 - \boldsymbol{\varrho}_\gamma [1]_2 = \left(\begin{smallmatrix} -\varrho_\chi [\tau\mathsf{e}]_2 \\ \varrho_{\mathsf{q}} [\delta]_2 \end{smallmatrix}\right) - \boldsymbol{\varrho}_\gamma [1]_2$. One can represent $\bar{\boldsymbol{C}}'(\mathsf{X}, \mathsf{Q}, \mathsf{R})$ similarly.

Clearly, $\mathbf{\Pi}^*$ is commit-and-prove, updatable, and universal.

7.1 Security

Theorem 3. *The set membership argument $\mathbf{\Pi}^*$ in Fig. 9 is perfectly complete. Assuming Elgamal is IND-CPA secure, it is computationally zero-knowledge.*

We postpone the proof of this theorem to Appendix A.2. The following straightforward soundness reduction relies on the security of AC^*.

Theorem 4. *Let $\ell = 2$ and $\mathsf{k} = 1$. Let $\mathcal{D}_{\mathsf{k}}$ be the distribution of $\left[\begin{smallmatrix} 1 \\ \mathsf{e} \end{smallmatrix}\right]_2$ for $\mathsf{e} \leftarrow_{\$} \mathbb{Z}_p$. Let $N = \mathsf{poly}(\lambda)$ be an upper bound on $|\mathcal{S}|$. The set membership NIZK $\mathbf{\Pi}^*$ in Fig. 9 is sound, assuming AC^* is $[\cdot]_1$-collision-resistant.*

Proof. Let $\mathcal{A}_{\mathbf{\Pi}^*}$ be a successful soundness adversary (as defined in Section 2.1) for $\mathbf{\Pi}^*$. That is, with a non-negligible probability $\varepsilon_{\mathcal{A}_{\mathbf{\Pi}^*}}$, for $(\mathsf{p}, \mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{K}_{\mathtt{crs}}(1^\lambda)$ and for any valid $\mathtt{lpar}$, $\mathcal{A}_{\mathbf{\Pi}^*}(\mathtt{crs}, \mathtt{lpar})$ outputs $(\mathbb{x}, \pi)$, such that $\mathsf{V}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x}, \pi) = 1$ but either (1) π is a membership argument but $\mathbb{x} \notin \mathcal{L}^{\mathsf{sm}}_{\mathtt{lpar}}$ or (1) π is a non-membership argument but $\mathbb{x} \in \mathcal{L}^{\mathsf{sm}}_{\mathtt{lpar}}$.

Decrypting all verification equations, the verifier checks guarantee that the AC^* verifier accepts $[\chi]_1 \leftarrow \mathsf{Dec}_{\mathsf{sk}}(\mathbf{ct}_\chi)$. Essentially, the constructed adversary

```
$\mathsf{Pgen}(1^\lambda)$: $\mathsf{p} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, [1]_1, [1]_2) \leftarrow \mathsf{Pgen}(1^\lambda)$.
--------------------------------------------------------------------------------
$\mathsf{Kgen}(\mathsf{p})$: $(\mathtt{crs}, \mathtt{td}) \leftarrow \mathsf{AC}^*.\mathsf{Kgen}(\mathsf{p})$;
--------------------------------------------------------------------------------
$\mathsf{Com}(\mathtt{crs}, \mathtt{lpar} = (\mathsf{pk}, \mathcal{S}))$: $\mathsf{AC}^*.\mathtt{lpar} \leftarrow \mathcal{S}$; $\mathsf{AC}^*.\mathtt{crs}_{\mathtt{lpar}} \leftarrow$
    $\mathsf{AC}^*.\mathsf{Com}(\mathtt{crs}, \mathsf{AC}^*.\mathtt{lpar})$; return $\mathtt{crs}_{\mathtt{lpar}} \leftarrow (\mathsf{AC}^*.\mathtt{crs}_{\mathtt{lpar}}, \mathsf{pk})$;
--------------------------------------------------------------------------------
$\mathsf{P}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x} = [\mathbf{ct}_\chi]_1, \mathbb{w} = (\chi, \varrho_\chi))$:
    $\mathsf{AC}^*.\psi \leftarrow \mathsf{AC}^*.\mathsf{P}(\mathsf{AC}^*.\mathtt{crs}_{\mathtt{lpar}}, \chi)$;   // $\psi = ([\mathsf{q}, \boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$ or $\psi = ([\mathsf{q}, \mathsf{s}, \boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$
    $\varrho_\mathsf{q} \leftarrow_{\$} \mathbb{Z}_p$; $[\mathsf{ct}_\mathsf{q}]_1 \leftarrow \mathsf{Enc}_\mathsf{pk}([\mathsf{q}]_1; \varrho_\mathsf{q})$;
    If $\chi \in \mathcal{S}$ then
      1. $\boldsymbol{\varrho}_\gamma \leftarrow_{\$} \mathbb{Z}_p^2$; $[\mathbf{ct}_\gamma]_1 \leftarrow \mathsf{Enc}_\mathsf{pk}([\boldsymbol{\gamma}]_1; \boldsymbol{\varrho}_\gamma) \in \mathbb{G}_1^{2\times 2}$; $[\boldsymbol{z}]_2 \leftarrow \begin{pmatrix} -\varrho_\chi[\tau\mathsf{e}]_2 \\ \varrho_\mathsf{q}[\delta]_2 \end{pmatrix} - \boldsymbol{\varrho}_\gamma[1]_2 \in \mathbb{G}_2^2$;
      2. $\pi \leftarrow ([\mathsf{ct}_\mathsf{q}, \mathbf{ct}_\gamma]_1, [\delta, \boldsymbol{z}]_2)$
    else
      1. $\varrho_\mathsf{s} \leftarrow_{\$} \mathbb{Z}_p$; $[\mathsf{ct}_\mathsf{s}]_1 \leftarrow \mathsf{Enc}_\mathsf{pk}([\mathsf{s}]_1; \varrho_\mathsf{s}) \in \mathbb{G}_1^{1\times 2}$;
      2. $\boldsymbol{\varrho}_\gamma \leftarrow_{\$} \mathbb{Z}_p^3$; $[\mathbf{ct}_\gamma]_1 \leftarrow \mathsf{Enc}_\mathsf{pk}([\boldsymbol{\gamma}]_1; \boldsymbol{\varrho}_\gamma) \in \mathbb{G}_1^{3\times 2}$; $[\boldsymbol{z}]_2 \leftarrow \begin{pmatrix} -\varrho_\chi[\tau\mathsf{e}]_2 \\ \varrho_\mathsf{q}[\delta_1]_2 \\ \varrho_\mathsf{s}[\delta_2]_2 \end{pmatrix} - \boldsymbol{\varrho}_\gamma[1]_2 \in \mathbb{G}_2^3$;
      3. $\pi \leftarrow ([\mathsf{ct}_\mathsf{q}, \mathsf{ct}_\mathsf{s}, \mathbf{ct}_\gamma]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2)$;
    return $\pi$;   // membership: $6\mathfrak{g}_1 + 3\mathfrak{g}_2$; non-membership: $10\mathfrak{g}_1 + 5\mathfrak{g}_2$
--------------------------------------------------------------------------------
$\mathsf{Sim}(\mathtt{crs}_{\mathtt{lpar}}, \mathtt{td} = (\mathsf{e}, \tau), \mathbb{x} = [\mathbf{ct}_\chi]_1, mem \in \{\mathsf{Member}, \mathsf{NotMember}\})$:
    If $mem = \mathsf{Member}$ then
      1. $\delta \leftarrow_{\$} \mathbb{Z}_p$; $\boldsymbol{z} \leftarrow_{\$} \mathbb{Z}_p^2$; $\varrho_\mathsf{q} \leftarrow_{\$} \mathbb{Z}_p$; $[\mathsf{ct}_\mathsf{q}]_1 \leftarrow \mathsf{Enc}_\mathsf{pk}(0; \varrho_\mathsf{q})$;
      2. $[\mathbf{ct}_\gamma]_1 \leftarrow \begin{pmatrix} \mathsf{Enc}_\mathsf{pk}([\sigma\tau]_1;0) - [\mathbf{ct}_\chi]_1 \cdot \tau & -\mathsf{Enc}_\mathsf{pk}([\tau]_1;0) \\ -\mathsf{Enc}_\mathsf{pk}([\mathsf{C}_\mathcal{S}]_1;0) & [\mathsf{ct}_\mathsf{q}]_1 \end{pmatrix} \begin{pmatrix} \mathsf{e} \\ \delta \end{pmatrix} - \mathsf{Enc}_\mathsf{pk}(\mathbf{0}; \boldsymbol{z})$;
      3. $\pi \leftarrow ([\mathsf{ct}_\mathsf{q}, \mathbf{ct}_\gamma]_1, [\delta, \boldsymbol{z}]_2)$
    else
      1. $\boldsymbol{\delta} \leftarrow_{\$} \mathbb{Z}_p^2$; $\boldsymbol{z} \leftarrow_{\$} \mathbb{Z}_p^3$;
      2. $\varrho_\mathsf{q}, \varrho_\mathsf{s} \leftarrow_{\$} \mathbb{Z}_p$; $[\mathsf{ct}_\mathsf{q}]_1 \leftarrow \mathsf{Enc}_\mathsf{pk}(0; \varrho_\mathsf{q})$; $[\mathsf{ct}_\mathsf{s}]_1 \leftarrow \mathsf{Enc}_\mathsf{pk}(0; \varrho_\mathsf{s})$;
      3. $[\mathsf{ct}_\gamma]_1 \leftarrow -\begin{pmatrix} \mathsf{Enc}_\mathsf{pk}([\sigma\tau]_1;0) - [\mathbf{ct}_\chi]_1 \cdot \tau & -\mathsf{Enc}_\mathsf{pk}([\tau]_1;0) & \mathsf{Enc}_\mathsf{pk}(0;0) \\ -\mathsf{Enc}_\mathsf{pk}([\mathsf{C}_\mathcal{S}]_1;0) & [\mathsf{ct}_\mathsf{q}]_1 & -\mathsf{Enc}_\mathsf{pk}([\tau]_1;0) \\ -\mathsf{Enc}_\mathsf{pk}(1;0) & \mathsf{Enc}_\mathsf{pk}(0;0) & [\mathsf{ct}_\mathsf{s}]_1 \end{pmatrix} \begin{pmatrix} \mathsf{e} \\ \boldsymbol{\delta} \end{pmatrix} - \mathsf{Enc}_\mathsf{pk}(\mathbf{0}; \boldsymbol{z})$;
      4. $\pi \leftarrow ([\mathsf{ct}_\mathsf{q}, \mathsf{ct}_\mathsf{s}, \mathbf{ct}_\gamma]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2)$;
    return $\pi$;
--------------------------------------------------------------------------------
$\mathsf{V}(\mathtt{crs}_{\mathtt{lpar}}, \mathbb{x} = [\mathbf{ct}_\chi]_1, \pi)$ : $mem \leftarrow \mathsf{NotMember}$;
    if $\pi$ parses as $\pi = ([\mathsf{ct}_\mathsf{q}, \mathbf{ct}_\gamma]_1, [\delta, \boldsymbol{z}]_2)$ then $mem \leftarrow \mathsf{Member}$;
    If $mem = \mathsf{Member}$ then check
      1. $b_1 \leftarrow \mathsf{Enc}_\mathsf{pk}([\sigma\tau]_1;0) \bullet [\mathsf{e}]_2 - [\mathbf{ct}_\chi]_1 \bullet [\tau\mathsf{e}]_2 - \mathsf{Enc}_\mathsf{pk}([\tau]_1;0) \bullet [\delta]_2 \stackrel{?}{=} [\mathsf{ct}_{\gamma 1}]_1 \bullet$
         $[1]_2 + [z_1]_2 \bullet \mathsf{pk}$;
      2. $b_2 \leftarrow -\mathsf{Enc}_\mathsf{pk}([\mathsf{C}_\mathcal{S}]_1;0) \bullet [\mathsf{e}]_2 + [\mathsf{ct}_\mathsf{q}]_1 \bullet [\delta]_2 \stackrel{?}{=} [\mathsf{ct}_{\gamma 2}]_1 \bullet [1]_2 + [z_2]_2 \bullet \mathsf{pk}$;
      3. if $b_1$ and $b_2$ then return $\mathsf{Member}$ else return $\mathsf{Error}$;
    else check
      1. $\bar{b}_1 \leftarrow \mathsf{Enc}_\mathsf{pk}([\sigma\tau]_1;0) \bullet [\mathsf{e}]_2 - [\mathbf{ct}_\chi]_1 \bullet [\tau\mathsf{e}]_2 - \mathsf{Enc}_\mathsf{pk}([\tau]_1;0) \bullet [\delta_1]_2 \stackrel{?}{=} [\mathsf{ct}_{\gamma 1}]_1 \bullet$
         $[1]_2 + [z_1]_2 \bullet \mathsf{pk}$;
      2. $\bar{b}_2 \leftarrow -\mathsf{Enc}_\mathsf{pk}([\mathsf{C}_\mathcal{S}]_1;0) \bullet [\mathsf{e}]_2 + [\mathsf{ct}_\mathsf{q}]_1 \bullet [\delta_1]_2 - \mathsf{Enc}_\mathsf{pk}([\tau]_1;0) \bullet [\delta_2]_2 \stackrel{?}{=}$
         $[\mathsf{ct}_{\gamma 2}]_1 \bullet [1]_2 + [z_2]_2 \bullet \mathsf{pk}$;
      3. $\bar{b}_3 \leftarrow -\mathsf{Enc}(1;0) \bullet [\mathsf{e}]_2 + [\mathsf{ct}_\mathsf{s}]_1 \bullet [\delta_2]_2 \stackrel{?}{=} [\mathsf{ct}_{\gamma 3}]_1 \bullet [1]_2 + [z_3]_2 \bullet \mathsf{pk}$;
      4. if $\bar{b}_1$ and $\bar{b}_2$ and $\bar{b}_3$ then return $\mathsf{NotMember}$ else return $\mathsf{Error}$;
```

Fig. 9. The new set (non-)membership NIZK $\mathbf{\Pi}^*$.

```
$\mathcal{B}_{cr}(\mathtt{crs} = (\mathsf{p}, [1, (\sigma^i\tau)_{i=0}^N]_1, [1, \mathsf{e}, \sigma\mathsf{e}, \tau\mathsf{e}]_2))$   // $[\cdot]_1$-CR adversary, see Definition 4
Choose any set $\mathcal{S}$ of size $\leq N$;
$\mathsf{sk} \leftarrow_{\$} \mathbb{Z}_p; \mathsf{pk} \leftarrow [1\|\mathsf{sk}]_1; \mathtt{lpar} \leftarrow (\mathsf{pk}, \mathcal{S})$;
$\mathtt{crs}_{\mathtt{lpar}} \leftarrow \mathsf{Com}(\mathtt{crs}, \mathtt{lpar})$;
$(\mathbb{x}, \pi) \leftarrow \mathcal{A}_{\mathbf{\Pi}^*}(\mathtt{crs}_{\mathtt{lpar}})$;
$[\chi]_1 \leftarrow \mathsf{Dec}_{\mathsf{sk}}([\mathbf{ct}_\chi]_1); [\mathsf{q}]_1 \leftarrow \mathsf{Dec}_{\mathsf{sk}}([\mathsf{ct}_\mathsf{q}]_1); [\boldsymbol{\gamma}]_1 \leftarrow \mathsf{Dec}_{\mathsf{sk}}([\mathbf{ct}_\gamma]_1)$;
if $\pi$ parses as $([\mathsf{ct}_\mathsf{q}, \mathbf{ct}_\gamma]_1, [\delta, \boldsymbol{z}]_2)$ then $\psi \leftarrow ([\mathsf{q}, \boldsymbol{\gamma}]_1, [\delta]_2)$; return $(\mathcal{S}, [\chi]_1, \psi)$;
else $[\mathsf{s}]_1 \leftarrow \mathsf{Dec}_{\mathsf{sk}}([\mathsf{ct}_\mathsf{s}]_1); \psi \leftarrow ([\mathsf{q}, \mathsf{s}, \boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$; return $(\mathcal{S}, [\chi]_1, \psi)$; fi
```

Fig. 10. Reduction $\mathcal{B}_{cr}$ in the proof of $\mathbf{\Pi}^*$

$\mathcal{B}_{cr}$ (see Fig. 10), on its input, creates a new Elgamal key-pair. Based on that, $\mathcal{B}_{cr}$ then creates a correct $\mathtt{crs}_{\mathtt{lpar}}$ for $\mathcal{A}_{\mathbf{\Pi}^*}$. After obtaining $(\mathbb{x}, \pi)$ from $\mathcal{A}_{\mathbf{\Pi}^*}$, $\mathcal{B}_{cr}$ decrypts $\mathcal{A}_{\mathbf{\Pi}^*}$'s answer, obtaining and returning the input and the argument as expected from a $[\cdot]_1$-collision-resistance adversary.

Clearly, $\mathcal{B}_{cr}$ succeeds iff $\mathcal{A}_{\mathbf{\Pi}^*}$ succeeds. □

7.2 Efficiency

$\mathbf{\Pi}^*$'s CRS length is $N + 1$ elements of $\mathbb{G}_1$ and 4 elements of $\mathbb{G}_2$. The set membership argument length is $6\mathfrak{g}_1 + 3\mathfrak{g}_2$, which comes close to the $\mathbf{\Pi}_{\mathsf{clp}\emptyset}$ argument length $4\mathfrak{g}_1 + 3\mathfrak{g}_2$ for the simple OR language (this corresponds to $\ell = 2$). The difference comes from the fact that here we also need to encrypt AC^*'s argument ψ. On the other hand, the set non-membership argument length is ten elements of $\mathbb{G}_1$ and five elements of $\mathbb{G}_2$.

The prover's computation can be divided into precomputation and online computation. In precomputation, P computes $f(X)$ ($\Theta(|\mathcal{S}|)$ field operations) and $[\mathsf{q}]_1$ ($|\mathcal{S}|$ scalar multiplications in $\mathbb{G}_1$). In online precomputation, (1) the membership prover computes 8 scalar multiplications in $\mathbb{G}_1$ and 6 in $\mathbb{G}_2$ ($2\mathfrak{m}_1 + 2\mathfrak{m}_2$ to compute $\mathsf{AC}^*.\psi$ and $6\mathfrak{m}_1 + 4\mathfrak{m}_2$ in the rest of $\mathbf{\Pi}^*$), and (2) the non-membership prover computes fourteen scalar multiplications in $\mathbb{G}_1$ and ten in $\mathbb{G}_2$ ($4\mathfrak{m}_1 + 4\mathfrak{m}_2$ to compute $\mathsf{AC}^*.\psi$ and $10\mathfrak{m}_1 + 6\mathfrak{m}_2$ in the rest of $\mathbf{\Pi}^*$). (The online computation includes the computation of $[\mathsf{ct}_\mathsf{q}]_1$ and other ciphertexts.)

The set membership verifier's computation is dominated by fifteen pairings (eight to check b_1, seven to check b_2). However, two pairings (the pairings involved in $\mathsf{Enc}_{\mathsf{pk}}([\sigma\tau]_1; 0) \bullet [\mathsf{e}]_2$ and $\mathsf{Enc}_{\mathsf{pk}}([\mathsf{C}_\mathcal{S}]_1; 0) \bullet [\mathsf{e}]_2$) can be precomputed. Thus, online the verifier has to only compute thirteen pairings. Similarly, the set non-membership verifier's computation is dominated by 23 pairings (eight to check $\bar{b}_1$, eight to check $\bar{b}_2$, and seven to check $\bar{b}_3$), but three can be precomputed so the online computation is 20 pairings.

We refer to Table 1 for an extensive efficiency comparison.

8 On Handling Group Elements with CLPØ

The CLPØ NIZK [CLPØ21] works in the case where the prover knows all the elements of all DRs as integers. This seems to exclude applications where one needs to prove statements about group elements. In $\mathbf{\Pi}^*$, we overcome this issue by making the following observation. Consider the case of a single DR $\boldsymbol{C}(\boldsymbol{X}) = (h(\boldsymbol{X}) \| T(\boldsymbol{X}))$, where $h(\boldsymbol{X})$ is a column vector. Then, for CLPØ to work, it suffices that the prover (1) knows $[\boldsymbol{C}(\boldsymbol{\chi})]_1$, and (2) is able to compute $[\boldsymbol{\delta}]_2$; for this, it suffices to compute $[\mathsf{we}]_2$, where w is such that $h(\boldsymbol{X}) = T(\boldsymbol{X})\mathsf{w}$ (this follows from CLPØ's construction).

In the case of $\mathbf{\Pi}^*$, (1) means that the prover must be able to compute $[\mathsf{q}, \mathbf{Z}_{\mathcal{S}}(\sigma), \mathsf{s}]_1$ (and thus χ, but not σ, must be available as an integer, and one must include to the CRS information needed to recompute $[\mathbf{Z}_{\mathcal{S}}(\sigma)]_1$), and (2) means that $[\sigma\mathsf{e}, \mathsf{e}]_2$ must be given as part of the CRS.

We leave the grand generalization of this observation for future work.

References

AFG$^+$16. Masayuki Abe, Georg Fuchsbauer, Jens Groth, Kristiyan Haralambiev, and Miyako Ohkubo. Structure-preserving signatures and commitments to group elements. *Journal of Cryptology*, 29(2):363–421, April 2016.

ALSZ20. Behzad Abdolmaleki, Helger Lipmaa, Janno Siim, and Michal Zajac. On QA-NIZK in the BPK model. In Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas, editors, *PKC 2020, Part I*, volume 12110 of *LNCS*, pages 590–620. Springer, Heidelberg, May 2020.

AN11. Tolga Acar and Lan Nguyen. Revocation for delegatable anonymous credentials. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 423–440. Springer, Heidelberg, March 2011.

ATSM09. Man Ho Au, Patrick P. Tsang, Willy Susilo, and Yi Mu. Dynamic universal accumulators for DDH groups and their application to attribute-based anonymous credential systems. In Marc Fischlin, editor, *CT-RSA 2009*, volume 5473 of *LNCS*, pages 295–308. Springer, Heidelberg, April 2009.

BBLP21. Olivier Blazy, Xavier Bultel, Pascal Lafourcade, and Octavio Perez-Kempner. Generic Plaintext Equality and Inequality Proofs. In Borisov and Diaz [BD21], pages 415–435.

BCF$^+$21. Daniel Benarroch, Matteo Campanelli, Dario Fiore, Kobi Gurkan, and Dimitris Kolonelos. Zero-Knowledge Proofs for Set Membership: Efficient, Succinct, Modular. In Borisov and Diaz [BD21], pages 393–414.

BCKL08. Mira Belenkiy, Melissa Chase, Markulf Kohlweiss, and Anna Lysyanskaya. P-signatures and noninteractive anonymous credentials. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 356–374. Springer, Heidelberg, March 2008.

BCV15. Olivier Blazy, Céline Chevalier, and Damien Vergnaud. Non-interactive zero-knowledge proofs of non-membership. In Kaisa Nyberg, editor, *CT-RSA 2015*, volume 9048 of *LNCS*, pages 145–164. Springer, Heidelberg, April 2015.

BD21. Nikita Borisov and Claudia Diaz, editors. *FC 2021 (1)*, volume 12674 of *LNCS*, Virtual, March 1–15, 2021. Springer, Cham.

BdM93. Josh Benaloh and Michael de Mare. One-Way Accumulators: A Decentralized Alternative to Digital Signatures. In Tor Helleseth, editor, *EUROCRYPT 1993*, volume 765 of *LNCS*, pages 274–285, Lofthus, Norway, May 23–27, 1993. Springer, Heidelberg, 1994.

BDSS16. Olivier Blazy, David Derler, Daniel Slamanig, and Raphael Spreitzer. Non-interactive plaintext (in-)equality proofs and group signatures with verifiable controllable linkability. In Kazue Sako, editor, *CT-RSA 2016*, volume 9610 of *LNCS*, pages 127–143. Springer, Heidelberg, February / March 2016.

BLL00. Ahto Buldas, Peeter Laud, and Helger Lipmaa. Accountable certificate management using undeniable attestations. In Dimitris Gritzalis, Sushil Jajodia, and Pierangela Samarati, editors, *ACM CCS 2000*, pages 9–17. ACM Press, November 2000.

BLL02. Ahto Buldas, Peeter Laud, and Helger Lipmaa. Eliminating Counterevidence with Applications to Accountable Certificate Management. *Journal of Computer Security*, 10(3):273–296, 2002.

BP97. Niko Barić and Birgit Pfitzmann. Collision-Free Accumulators and Fail-Stop Signature Schemes without Trees. In Walter Fumy, editor, *EUROCRYPT 1997*, volume 1233 of *LNCS*, pages 480–494, Konstanz, Germany, 11–15 May 1997. Springer, Heidelberg.

CCs08. Jan Camenisch, Rafik Chaabouni, and abhi shelat. Efficient protocols for set membership and range proofs. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 234–252. Springer, Heidelberg, December 2008.

CH20. Geoffroy Couteau and Dominik Hartmann. Shorter non-interactive zero-knowledge arguments and ZAPs for algebraic languages. In Daniele Micciancio and Thomas Ristenpart, editors, *CRYPTO 2020, Part III*, volume 12172 of *LNCS*, pages 768–798. Springer, Heidelberg, August 2020.

CLPØ21. Geoffroy Couteau, Helger Lipmaa, Roberto Parisella, and Arne Tobias Ødegaard. Efficient NIZKs for algebraic sets. In Mehdi Tibouchi and Huaxiong Wang, editors, *ASIACRYPT 2021, Part III*, volume 13092 of *LNCS*, pages 128–158. Springer, Heidelberg, December 2021.

DGP$^+$19. Vanesa Daza, Alonso González, Zaira Pindado, Carla Ràfols, and Javier Silva. Shorter quadratic QA-NIZK proofs. In Dongdai Lin and Kazue Sako, editors, *PKC 2019, Part I*, volume 11442 of *LNCS*, pages 314–343. Springer, Heidelberg, April 2019.

DT08. Ivan Damgård and Nikos Triandopoulos. Supporting non-membership proofs with bilinear-map accumulators. Cryptology ePrint Archive, Report 2008/538, 2008. `https://eprint.iacr.org/2008/538`.

FKL18. Georg Fuchsbauer, Eike Kiltz, and Julian Loss. The algebraic group model and its applications. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part II*, volume 10992 of *LNCS*, pages 33–62. Springer, Heidelberg, August 2018.

GKM$^+$18. Jens Groth, Markulf Kohlweiss, Mary Maller, Sarah Meiklejohn, and Ian Miers. Updatable and universal common reference strings with applications to zk-SNARKs. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 698–728. Springer, Heidelberg, August 2018.

GS08. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, Heidelberg, April 2008.

IK00. Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *41st FOCS*, pages 294–304. IEEE Computer Society Press, November 2000.

IK02. Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In Peter Widmayer, Francisco Triguero Ruiz, Rafael Morales Bueno, Matthew Hennessy, Stephan Eidenbenz, and Ricardo Conejo, editors, *ICALP 2002*, volume 2380 of *LNCS*, pages 244–256. Springer, Heidelberg, July 2002.

Lip12. Helger Lipmaa. Secure accumulators from euclidean rings without trusted setup. In Feng Bao, Pierangela Samarati, and Jianying Zhou, editors, *ACNS 12*, volume 7341 of *LNCS*, pages 224–240. Springer, Heidelberg, June 2012.

LLX07. Jiangtao Li, Ninghui Li, and Rui Xue. Universal accumulators with efficient nonmembership proofs. In Jonathan Katz and Moti Yung, editors, *ACNS 07*, volume 4521 of *LNCS*, pages 253–269. Springer, Heidelberg, June 2007.

LSZ22. Helger Lipmaa, Janno Siim, and Michał Zając. Counting Vampires: From Univariate Sumcheck to Updatable ZK-SNARK. In Shweta Agrawal and Dongdai Lin, editors, *ASIACRYPT 2022*, volume ? of *LNCS*, pages ?–?, Taipei, Taiwan, December 5–9, 2022. Springer, Cham. Accepted.

Ngu05. Lan Nguyen. Accumulators from bilinear pairings and applications. In Alfred Menezes, editor, *CT-RSA 2005*, volume 3376 of *LNCS*, pages 275–292. Springer, Heidelberg, February 2005.

A Some Missing Proofs

A.1 Proof of Theorem 2

Proof. Let $\mathcal{A}_{\mathrm{alg}}$ be an algebraic $\mathsf{DETACNM}$ adversary. Assume that $\mathcal{A}_{\mathrm{alg}}(\mathtt{crs})$ outputs $\psi = (\mathcal{S}, [\chi, \mathsf{q}, \mathsf{s}, \boldsymbol{\gamma}]_1, [\boldsymbol{\delta}]_2)$, such that V accepts with a non-negligible probability. Since $\mathcal{A}_{\mathrm{alg}}$ is algebraic, with every group element $G \in \mathbb{G}_\iota$, $\mathcal{A}_{\mathrm{alg}}$ also outputs a vector $\boldsymbol{a}$ explaining how G is constructed from the elements of $\mathtt{crs}$ that belong to $\mathbb{G}_i$. Next, we will make this more precise.

Let $\boldsymbol{X} = (\Sigma, \mathsf{T}, \mathsf{E})$ and $\boldsymbol{x} = (\sigma, \tau, \mathsf{e})$. E.g., T is the indeterminate corresponding to the trapdoor τ. We express each output of the $\mathsf{DETACNM}$ adversary $\mathcal{A}_{\mathrm{alg}}$ as a polynomial evaluation, with say $[\chi]_1 = [\chi(\boldsymbol{x})]_1$. The relevant polynomials are

$$\begin{aligned}
\chi(\boldsymbol{X}) =& \chi_1(\Sigma)\mathsf{T} + \chi_2 \ , & \mathsf{q}(\boldsymbol{X}) =& \mathsf{q}_1(\Sigma)\mathsf{T} + \mathsf{q}_2 \ , \\
\mathsf{s}(\boldsymbol{X}) =& \mathsf{s}_1(\Sigma)\mathsf{T} + \mathsf{s}_2 \ , & \gamma_1(\boldsymbol{X}) =& \gamma_{11}(\Sigma)\mathsf{T} + \gamma_{12} \ , \\
\gamma_2(\boldsymbol{X}) =& \gamma_{21}(\Sigma)\mathsf{T} + \gamma_{22} \ , & \gamma_3(\boldsymbol{X}) =& \gamma_{31}(\Sigma)\mathsf{T} + \gamma_{32} \ , \\
\delta_1(\boldsymbol{X}) =& \delta_{11} + \delta_{12}\mathsf{E} + \delta_{13}\Sigma\mathsf{E} + \delta_{14}\mathsf{T}\mathsf{E} \ , & \delta_2(\boldsymbol{X}) =& \delta_{21} + \delta_{22}\mathsf{E} + \delta_{23}\Sigma\mathsf{E} + \delta_{24}\mathsf{T}\mathsf{E} \ ,
\end{aligned}$$

where each polynomial (like q_1) on the RHS is of degree $\leq N$. That is, the algebraic adversary $\mathcal{A}_{\mathrm{alg}}$ also outputs coefficients of all above polynomials.

The $\mathsf{DETACNM}$ verifier's checks guarantee that $V_1(\sigma, \tau, \mathsf{e}) = V_2(\sigma, \tau, \mathsf{e}) = 0$. Moreover, if $R(\boldsymbol{X}) \neq 0$ then $V_3(\sigma, \tau, \mathsf{e}) = V_4(\sigma, \tau, \mathsf{e}) = 0$. Here,

$$\begin{aligned} V_1(\boldsymbol{X}) &= ((\mathsf{\Sigma} - \chi(\boldsymbol{X}))\,\mathsf{E} - \delta_1(\boldsymbol{X})) \cdot \mathsf{T} - \gamma_1(\boldsymbol{X}) \ , \\ V_2(\boldsymbol{X}) &= -\,\mathbf{Z}_{\mathcal{S}}(\mathsf{\Sigma})\mathsf{TE} + \mathsf{q}(\boldsymbol{X})\delta_1(\boldsymbol{X}) - \delta_2(\boldsymbol{X})\mathsf{T} - \gamma_2(\boldsymbol{X}) \ , \\ V_3(\boldsymbol{X}) &= -\,\mathsf{E} + \mathsf{s}(\boldsymbol{X})\delta_2(\boldsymbol{X}) - \gamma_3(\boldsymbol{X}) \ . \end{aligned}$$

Consider separately the cases (1) $V_1 = V_2 = V_3 = 0$ as polynomials, and (2) either $V_1 \neq 0$ or $V_2 \neq 0$ or $V_3 \neq 0$.

Case 1. Assume $V_1 = V_2 = V_3 = 0$ as a polynomial. Think of the polynomials as members of $\mathcal{R}[\mathsf{T}, \mathsf{E}]$, where $\mathcal{R} = \mathbb{Z}_p[\mathsf{\Sigma}]$. We now enlist the non-zero coefficients of all monomials $\mathsf{T}^i\mathsf{E}^j$ of all polynomials, highlighting the coefficients that are actually needed in this proof (we give other coefficients for completeness' sake):

(i,j)	V_1
$(2,1)$	$-\delta_{14} - \chi_1(\mathsf{\Sigma})$
$(1,1)$	$-\delta_{12} + (1 - \delta_{13})\mathsf{\Sigma} - \chi_2$
$(1,0)$	$-\delta_{11} - \gamma_{11}(\mathsf{\Sigma})$
$(0,0)$	$-\gamma_{12}$

(i,j)	V_3
$(2,1)$	$\delta_{24}\mathsf{s}_1(\mathsf{\Sigma})$
$(1,1)$	$\delta_{22}\mathsf{s}_1(\mathsf{\Sigma}) + \delta_{23}\mathsf{s}_1(\mathsf{\Sigma})\mathsf{\Sigma} + \mathsf{s}_2\delta_{24}$
$(1,0)$	$\delta_{21}\mathsf{s}_1(\mathsf{\Sigma}) - \gamma_{31}(\mathsf{\Sigma})$
$(0,1)$	$\mathsf{s}_2\delta_{23}\mathsf{\Sigma} + \mathsf{s}_2\delta_{22} - 1$
$(0,0)$	$\mathsf{s}_2\delta_{21} - \gamma_{32}$

(i,j)	V_2
$(2,1)$	$\delta_{14}\mathsf{q}_1(\mathsf{\Sigma}) - \delta_{24}$
$(1,1)$	$\delta_{14}\mathsf{q}_2 + \delta_{12}\mathsf{q}_1(\mathsf{\Sigma}) + (\delta_{13}\mathsf{q}_1(\mathsf{\Sigma}) - \delta_{23})\mathsf{\Sigma} - \mathbf{Z}_{\mathcal{S}}(\mathsf{\Sigma}) - \delta_{22}$
$(1,0)$	$\delta_{11}\mathsf{q}_1(\mathsf{\Sigma}) - \gamma_{21}(\mathsf{\Sigma}) - \delta_{21},$
$(0,1)$	$\delta_{12}\mathsf{q}_2 + \delta_{13}\mathsf{q}_2\mathsf{\Sigma}$
$(0,0)$	$\delta_{11}\mathsf{q}_2 - \gamma_{22}$

For example, the coefficient of $\mathsf{T}^2\mathsf{E}$ in V_1 is $-\delta_{14} - \chi_1(\mathsf{\Sigma})$. Since $V_i = 0$ as a polynomial, the coefficient of any monomial $\mathsf{T}^j\mathsf{E}^k$ in any V_i is also 0.

From the coefficient of $\mathsf{T}^2\mathsf{E}$ of V_1, we get $\chi_1(\mathsf{\Sigma}) = -\delta_{14}$. From the coefficient of TE of V_1, after separating the coefficients of $\mathsf{\Sigma}^i$, we get $\delta_{13} = 1$ and $\delta_{12} = -\chi_2$. Consider the coefficients of V_3:

- E: separating the coefficients of $\mathsf{\Sigma}$, $\mathsf{s}_2\delta_{23} = 0$ and $\mathsf{s}_2\delta_{22} = 1$. Hence $\mathsf{s}_2 \neq 0$, and thus $\delta_{23} = 0$. Moreover, $\delta_{22} = 1/\mathsf{s}_2$.
- TE: $\mathsf{s}_1(\mathsf{\Sigma})/\mathsf{s}_2 + \mathsf{s}_2\delta_{24} = 0$ and thus $\mathsf{s}_1(\mathsf{\Sigma}) = \mathsf{s}_2^2\delta_{24}$.
- $\mathsf{T}^2\mathsf{E}$: $\mathsf{s}_2^2\delta_{24}^2 = 0$. Since $\mathsf{s}_2 \neq 0$, $\delta_{24} = 0$.

  Going back to the coefficient of TE, we get $\mathsf{s}_1(\mathsf{\Sigma}) = 0$.

Consider the coefficients of V_2:

- TE: $\mathbf{Z}_{\mathcal{S}}(\mathsf{\Sigma}) - \delta_{14}\mathsf{q}_2 + 1/\mathsf{s}_2 = (\mathsf{\Sigma} - \chi_2)\mathsf{q}_1(\mathsf{\Sigma})$.

  Since $\mathbf{Z}_{\mathcal{S}}(\mathsf{\Sigma})$ is non-constant, $\mathsf{q}_1(\mathsf{\Sigma}) \neq 0$.
- $\mathsf{T}^2\mathsf{E}$: $\delta_{14}\mathsf{q}_1(\mathsf{\Sigma}) = 0$. Since $\mathsf{q}_1(\mathsf{\Sigma}) \neq 0$, we get $\delta_{14} = 0$.

Hence, the coefficient of TE of V_2 gives $\mathbf{Z}_{\mathcal{S}}(\mathsf{\Sigma}) + 1/\mathsf{s}_2 = (\mathsf{\Sigma} - \chi_2)\mathsf{q}_1(\mathsf{\Sigma})$. Thus, $(\mathsf{\Sigma} - \chi_2) \mid \mathbf{Z}_{\mathcal{S}}(\mathsf{\Sigma}) + 1/\mathsf{s}_2$. Since $\chi(\boldsymbol{X}) = \chi_2$, $\mathbf{Z}_{\mathcal{S}}(\chi_2) = -1/\mathsf{s}_2 \neq 0$.

Case 2. The case $V_i \neq 0$ for some i can be handled in a standard way. Assume for example that $V_2 \neq 0$. We construct a PDL reduction

$\mathcal{B}(\{[\sigma^i]_1\}_{i=0}^{N+1}, \{[\sigma^i]_1\}_{i=0}^{2})$. $\mathcal{B}$ samples $\alpha_1, \alpha_2, \beta_1, \beta_2 \leftarrow_{\$} \mathbb{Z}_p$ and sets implicitly $\tau \leftarrow \alpha_1\sigma + \beta_1$ and $\mathsf{e} \leftarrow \alpha_2\sigma + \beta_2$. Then, $\mathcal{B}$ creates $\mathtt{crs}$ for an $\mathsf{DETACNM}$ adversary $\mathcal{A}_{\mathrm{alg}}$, and calls $\mathcal{A}_{\mathrm{alg}}$ with $\mathtt{crs}$. After obtaining π, together with the coefficients of the polynomials like $\chi(\Sigma)$, from $\mathsf{Ext}_{\mathcal{A}_{\mathrm{alg}}}$, $\mathcal{B}$ reconstructs the coefficients of the degree-$\leq (N+2)$ polynomial V_2 (which is now univariate since τ and e are affine maps of σ). We know $V_2 \neq 0$ but $V_2(\sigma) = 0$. $\mathcal{B}$ factorizes V_2 and finds up to $N+2$ roots x_i of V_2. $\mathcal{B}$ tests which one of them is equal to σ, and returns σ. □

A.2 Proof of Theorem 3

Proof. **Perfect completeness.** We consider separately membership and non-membership arguments.

Membership Argument. Clearly, $b_1 = \mathsf{true}$ iff $\mathsf{Enc}_{\mathsf{pk}}([\sigma]_1; 0)\tau\mathsf{e} - [\mathbf{ct}_\chi]_1\tau\mathsf{e} - \mathsf{Enc}_{\mathsf{pk}}([\tau]_1; 0)\delta \overset{?}{=} [\mathsf{ct}_{\gamma 1}]_1 + z_1 \cdot \mathsf{pk} \iff \mathsf{Enc}\left((\sigma - \chi)\tau\mathsf{e} - \tau\delta; -\varrho_\chi\tau\mathsf{e}\right) \overset{?}{=} \mathsf{Enc}\left(\gamma_1; \varrho_{\gamma 1}\right) + \mathsf{Enc}_{\mathsf{pk}}(0; z_1)$. Clearly, $(\sigma-\chi)\tau\mathsf{e} - \tau\delta = (\sigma-\chi)\tau\mathsf{e} - \tau((\sigma-\chi)\mathsf{e} - \varrho_\delta) = \varrho_\delta\tau = \gamma_1\tau$, and thus the ciphertext part is correct. On the other hand, randomizers are correct by definition.

Similarly, $b_2 = \mathsf{true}$ iff $-\mathsf{Enc}_{\mathsf{pk}}(\mathbf{Z}_{\mathcal{S}}(\sigma); 0)\tau\mathsf{e} + [\mathsf{ct}_{\mathsf{q}}]_1\delta \overset{?}{=} [\mathsf{ct}_{\gamma 2}]_1 + z_2 \cdot \mathsf{pk} \iff \mathsf{Enc}\left(-\mathbf{Z}_{\mathcal{S}}(\sigma)\tau\mathsf{e} + \mathsf{q}\delta; \varrho_{\mathsf{q}}\delta\right) \overset{?}{=} \mathsf{Enc}\left(\gamma_2; \varrho_{\gamma 2}\right) + \mathsf{Enc}_{\mathsf{pk}}(0; z_2)$. Consider first the ciphertexts. Clearly, $-\mathbf{Z}_{\mathcal{S}}(\sigma)\tau\mathsf{e} + \mathsf{q}\delta = -\mathbf{Z}_{\mathcal{S}}(\sigma)\tau\mathsf{e} + \mathsf{q} \cdot ((\sigma-\chi)\mathsf{e} - \varrho_\delta) = -\varrho_\delta\mathsf{q} = \gamma_2$. On the other hand, randomizers are correct by definition.

Non-Membership Argument. Clearly, $\bar{b}_1 = \mathsf{true}$ iff $\mathsf{Enc}_{\mathsf{pk}}([\sigma]_1; 0)\tau\mathsf{e} - [\mathbf{ct}_\chi]_1\tau\mathsf{e} - \mathsf{Enc}_{\mathsf{pk}}([\tau]_1; 0)\delta_1 \overset{?}{=} [\mathsf{ct}_{\gamma 1}]_1 + z_1 \cdot \mathsf{pk} \iff \mathsf{Enc}_{\mathsf{pk}}\left((\sigma - \chi)\tau\mathsf{e} - \tau\delta_1; \varrho_\chi\tau\mathsf{e}\right) \overset{?}{=} \mathsf{Enc}_{\mathsf{pk}}(\gamma_1; \varrho_{\gamma 1}) + \mathsf{Enc}_{\mathsf{pk}}(0; z_1)$. Consider first the ciphertexts. Clearly, $(\sigma - \chi)\tau\mathsf{e} - \tau\delta_1 = y_1\tau = \gamma_1\tau$. On the other hand, randomizers are correct by definition.

Similarly, $\bar{b}_2 = \mathsf{true}$ iff $-\mathsf{Enc}_{\mathsf{pk}}(\mathbf{Z}_{\mathcal{S}}(\sigma); 0)\tau\mathsf{e} + [\mathsf{ct}_{\mathsf{q}}]_1\delta_1 - \mathsf{Enc}_{\mathsf{pk}}(1; 0)\tau\delta_2 \overset{?}{=} [\mathsf{ct}_{\gamma 2}]_1 + z_2 \cdot \mathsf{pk} \iff \mathsf{Enc}_{\mathsf{pk}}\left(-\mathbf{Z}_{\mathcal{S}}(\sigma)\tau\mathsf{e} + \mathsf{q}\delta_1 - \tau\delta_2; \varrho_{\mathsf{q}}\delta_1\right) \overset{?}{=} \mathsf{Enc}_{\mathsf{pk}}(\gamma_2; \varrho_{\gamma 2}) + \mathsf{Enc}_{\mathsf{pk}}(0; z_2)$. Consider first the ciphertexts. Clearly, $-\mathbf{Z}_{\mathcal{S}}(\sigma)\tau\mathsf{e} + \mathsf{q}\delta_1 - \tau\delta_2 = -\mathbf{Z}_{\mathcal{S}}(\sigma)\tau\mathsf{e} + \mathsf{q} \cdot ((\sigma - \chi)\mathsf{e} - y_1) - \tau(1/\mathsf{s} \cdot \mathsf{e} - y_2) = -y_1\mathsf{q} + y_2\tau = \gamma_2$. On the other hand, randomizers are correct by definition.

Finally, $\bar{b}_3 = \mathsf{true}$ iff $-\mathsf{Enc}(1; 0)\mathsf{e} + \mathsf{ct}_{\mathsf{s}}\delta_2 \overset{?}{=} [\mathsf{ct}_{\gamma 3}]_1 + z_3 \cdot \mathsf{pk} \iff \mathsf{Enc}_{\mathsf{pk}}\left(-\mathsf{e} + \mathsf{s}\delta_2; \varrho_{\mathsf{s}}\delta_2\right) \overset{?}{=} \mathsf{Enc}_{\mathsf{pk}}(\gamma_3; \varrho_{\gamma 3}) + \mathsf{Enc}_{\mathsf{pk}}(0; z_3)$. Consider first the ciphertexts. Clearly, $-\mathsf{e} + \mathsf{s}\delta_2 = -\mathsf{e} + \mathsf{s}(1/\mathsf{s} \cdot \mathsf{e} - y_2) = -y_2\mathsf{s} = \gamma_3$. On the other hand, randomizers are correct by definition.

Computational zero-knowledge: First, consider the membership argument. Fix any λ, and let $(\mathtt{crs}, \mathtt{td}) \in \mathrm{Supp}(\mathsf{K}_{\mathsf{crs}}(1^\lambda))$. Let $\mathtt{lpar} = (\mathsf{pk}, \mathcal{S})$ and $(\mathbb{x}, \mathbb{w}) \in \mathcal{R}_{\mathtt{lpar}}$. To show zero-knowledge we first define an hybrid simulator Sim_H. The hybrid Sim_H receives as additional input an Elgamal ciphertext $[\mathsf{ct}_{\mathsf{q}}]_1$, that is an encryption of $[\mathsf{q}]_1$ such that $\mathsf{q}(\sigma - \chi) = \mathbf{Z}_{\mathcal{S}}(\sigma)$, where $[\chi]_1 = \mathsf{Dec}_{\mathsf{sk}}([\mathsf{ct}_\chi]_1)$. Then Sim_H computes its output as the simulator in Fig. 9, except that it computes $[\mathsf{ct}_{\mathsf{q}}]_1$ as an encryption of $[\mathsf{q}]_1$ and not of 0. The output of Sim_H is perfectly close to the output of the honest prover. The proof of the last statement is the same as the perfect zero-knowledge proof in [CLPØ21]. For completeness, we state a proof for this concrete case. In the honest prover's algorithm, since $\boldsymbol{\varrho}_\gamma$ is uniformly random, then also $\boldsymbol{z}$ is uniformly random. As in

Fact 1, $\boldsymbol{\delta}$ output by an honest prover is uniformly random. On the other hand, Sim_H also samples uniformly random $\boldsymbol{\delta}$ and $\boldsymbol{z}$. Finally, in both the prover's and simulator's case, one can verify manually that $[\mathbf{ct}_\gamma]_1$ is the unique value that makes the verifier accept the argument π. Then we show that the output of the real simulator (see Fig. 9) is computationally close to the output of Sim_H. This follows directly from Elgamal IND-CPA security (which holds under the XDH assumption).

In the non-membership argument, zero-knowledge holds analogously. □

NORDIC ECOLABEL 241 731 PRINTED MATTER

Graphic design: Communication Division, UiB / Print: Skipnes Kommunikasjon AS

UNIVERSITAS BERGENSIS

uib.no

ISBN: 9788230854853 (print)
9788230847855 (PDF)