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 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) 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, which provides simple and efficient NIZKs but only offers heuristic security guarantees, and pairing-based NIZKs such as the Groth-Sahai proof system [21] (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. 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 offers simultaneously simple, efficient, and provably secure proof systems, but these are restricted to a small class of languages – namely, linear languages. Some recent SNARK proof systems also offer generic and efficient methods to handle a large class of languages given by their high-level description; however, they all rely on very strong knowledge-of-exponent style assumptions.

The Couteau-Hartmann argument system. Very recently, Couteau and Hartmann put forth a new framework for constructing pairing based NIZKs [9]. 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* [5,8], 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[3], 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

[3] 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.

simplicity. As many previous works pointed out (see e.g. [25]), 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[4].

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 [9]. We focus on the important setting of commit-and-prove NIZK argument systems, i.e. languages of the form $\{\mathsf{Com}(x_1), \ldots, \mathsf{Com}(x_n) \mid R(x_1, \ldots, x_n)\}$, where R is some efficiently computable relation. Our method allows us to automatically obtain a compact matrix description for many types of high-level relations.

New NIZKs: improved efficiency or generality. As a first byproduct, we obtain improved NIZKs for some important statements, such as set membership (see

[4] 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.

Table 1) or the language of commitments to points on an elliptic curve[5], 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) [17] for new languages (SPHFs were the original motivation for introducing the notion of algebraic language, and [5] 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 [9] 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

[5] NIZKs for this type of languages have recently found important applications in blockchain applications, such as the zcash cryptocurrency, see [25] 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 7 and 8 and an additional lemma in the full version [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 [33]	$d\lvert\mathbb{G}_1\rvert + (3d+2)\lvert\mathbb{G}_2\rvert$	-	-
CHM NIZK + [9] $(\boldsymbol{\Gamma}, \boldsymbol{\theta})$, full version [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 8	$2d\lvert\mathbb{G}_1\rvert + (2d-1)\lvert\mathbb{G}_2\rvert$	$(5d-3)\mathfrak{e}_1 + 4d\mathfrak{e}_2$	$7d-1$
New NIZK, Lemma 7	$2d\lvert\mathbb{G}_1\rvert + (2d-1)\lvert\mathbb{G}_2\rvert$	$\leq 3d\mathfrak{e}_1 + (4d-2)\mathfrak{e}_2$	$7d-1$

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 [9], 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 function F_i depends on at most q of its input locations.[6] In particular, this efficiently captures arithmetic circuits, hence all NP languages.[7]

Rank-1 constraints systems (R1CS) are well-known to be powerful, since they capture *compactly* many languages of interest [16]. 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 find-

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

[7] 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.

ing 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 [22] 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 [20] 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 [18]), 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 double-encrypting certain values. This comes at the cost of increasing the commitment and argument size. Third, we argue that the gap assumption [30] 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 *quasideterminental (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 [29]) 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 [11] 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 $\mathcal{A}$, $\mathcal{I}(\mathcal{A}) := \{f \in \mathbb{F}[\boldsymbol{X}] : f(\boldsymbol{\chi}) \text{ for all } \boldsymbol{\chi} \in \mathcal{A}\}$. Since each ideal of $\mathcal{F}[\boldsymbol{X}]$ is finitely generated [11], 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 [27] 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. 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 $\mathsf{ExtKerMDH}$ assumption from [9], 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 $\mathsf{ExtKerMDH}$ 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 [31]). 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 [23,24], 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 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 [9, 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, [4]. 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 [14].

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.

Full Version. Due to the lack of space, a significant amount of additional material (including all proofs) can be found in the full version of this paper, [10].

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

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

Algebraic languages [8,9] 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 [17,5,3]) 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, 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 $\mathsf{pk} = \left[\begin{smallmatrix}1\\\mathsf{sk}\end{smallmatrix}\right]_1$. Elgamal's IND-CPA security is based on $\mathcal{L}_1$-KerMDH, that is, DDH.

The following Extended Kernel Diffie-Hellman assumption ExtKerMDH [9] generalizes the well-known KerMDH assumption [28]. We also define in parallel a new, slightly weaker version of this assumption, CED (*computational extended determinant*).

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

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

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 $\mathrm{rk}(\boldsymbol{C}) \geq \ell$.

CED is *weaker* than ExtKerMDH since a successful adversary has to satisfy a stronger condition ($\mathrm{rk}(\boldsymbol{C}) \geq \ell$ instead of $\mathrm{rk}(\boldsymbol{\gamma}\|\boldsymbol{C}) \geq \ell$). (See the full version [10]

for a reduction.) 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 [9]), 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

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

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

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

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 [12] 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 [9] 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 [26], 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 $[\boldsymbol{D}]_2 = \left[\begin{smallmatrix} 1 \\ e \end{smallmatrix}\right]_2 \leftarrow_\$ \mathcal{L}_1$ (i.e., $e \leftarrow_\$ \mathbb{Z}_p$), he returns $([\boldsymbol{\gamma} \| \boldsymbol{C}]_1 \in \mathbb{G}_1^{\ell \times (\ell+1)}, [\boldsymbol{\delta}]_2 \in \mathbb{G}_2^{(\ell-1)\times 1})$, such that $\mathrm{rk}(\boldsymbol{C}) \geq \ell$ and

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

Following [9], 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, $\mathrm{rk}(\boldsymbol{C}) = d$.

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 the full version [10] 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 $\boldsymbol{\chi} \in \mathcal{Z}(F)$, $\boldsymbol{h}(\boldsymbol{\chi}) \in \mathrm{colspace}(\boldsymbol{T}(\boldsymbol{\chi}))$.*

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

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

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

Lemma 1. *Suppose a matrix $\boldsymbol{C}$ satisfies the affine map and F-rank properties. If it in addition satisfies one of the following properties, it also satisfies the first column dependence property.*

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

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

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 $\mathcal{L}_{\mathsf{pk},F}$. 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.

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

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

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 2. *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.*

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 3. *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.*

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 $\boldsymbol{C}(\boldsymbol{X}) = \sum_{k=1}^{\nu} \boldsymbol{P}_k X_k + \boldsymbol{Q}$. 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 $[\boldsymbol{C}]_1$.

$\mathsf{kgen}(\mathsf{p}, \mathtt{lpar})$: $e \leftarrow_\$ \mathbb{Z}_p$; 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 $\pi \leftarrow ([\mathbf{ct}^\gamma]_1, [\boldsymbol{\delta}, \boldsymbol{z}]_2) \in \mathbb{G}_1^{2\ell} \times \mathbb{G}_2^{2\ell-1}$.

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

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

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

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., $\boldsymbol{\gamma} + \boldsymbol{C}(\boldsymbol{\chi})(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}) = \mathbf{0}$. The latter is equivalent to $\boldsymbol{\gamma} + (\sum_k \boldsymbol{P}_k \chi_k)(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix}) = -\boldsymbol{Q}(\begin{smallmatrix} e \\ \boldsymbol{\delta} \end{smallmatrix})$. 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 of 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 $\left[\begin{smallmatrix} e \\ \delta \end{smallmatrix}\right]_2 = \left(\begin{smallmatrix} 1 \\ -\boldsymbol{w} \end{smallmatrix}\right)[e]_2 + \left(\begin{smallmatrix} 0 \\ -\boldsymbol{y} \end{smallmatrix}\right)[1]_2$. 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 4. *Consider $\boldsymbol{\Pi}_{\mathsf{nizk}}$ with QDR $\boldsymbol{C}$. Define $T_P(\boldsymbol{C}) := |\{(i,j) : \exists k, P_{kij} \neq 0\}|$, and $T_Q(\boldsymbol{C}) := |\{(i,j) : Q_{ij} \neq 0\}|$. Let $\mathfrak{c}$ be the time needed to run comp, $\mathfrak{e}_\iota$ is the time of an exponentiation in $\mathbb{G}_\iota$, and $\mathfrak{p}$ is the time of a pairing. Then (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 as 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 $\boldsymbol{Q}$ 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 4 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.*

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). 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, [29]) over a finite field $\mathbb{F}_p$ computes a function $F : \mathbb{F}_p^n \to \mathbb{F}_p$. 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 [23,24] and the references therein.

Ishai and Kushilevitz [23,24] related ABPs to matrix determinants as follows.

Proposition 1. *[24, 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 [24, Lemma 1], to ensure consistency with the notation from the CED assumption.

5.2 NIZK for Algebraic Branching Programs

Lemma 5. *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* $\ell = |V| - 1$*.*

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 $i \in [1, \ell]$. (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 6. *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* $\mathbb{G}_2$*.*

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 $F_1, \ldots, F_\tau \in \mathbb{Z}_p[X]$. As before, we aim to prove that an Elgamal-encrypted χ satisfies $\chi \in \mathcal{A}$, i.e., $F_i(\chi) = 0$ for all i. In the univariate case, all ideals are principal [11, 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 $F \leftarrow \gcd(F_1, \ldots, F_\tau)$ [11, Section 1.5].

Moreover, $\mathcal{I}(\mathcal{A}(F)) = \mathcal{I}(F_{red})$ [11, 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}

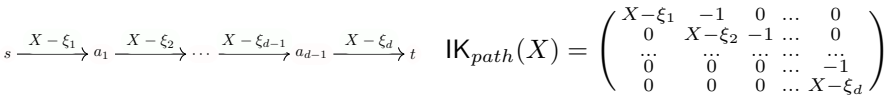


Fig. 3. The ABP $\mathsf{abp}^d_{\mathsf{path}}(X, \boldsymbol{\xi})$ for $F(X) = \prod_{i=1}^{d}(X - \xi_i)$ and $\mathsf{IK}_{path}(X)$

can be efficiently computed as $F_{red} = F/\gcd(F, F')$, [11, 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.

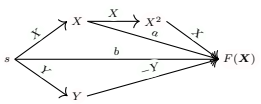
Lemma 7. *Let $F(X)$ be a univariate reduced polynomial. The ABP-based NIZK argument for $\mathcal{L}_{\mathsf{pk},F}$ has prover's computation of at most $3d$ exponentiations in $\mathbb{G}_1$ and $4d - 2$ exponentiations in $\mathbb{G}_2$, verifier's computation of $7d - 1$ pairings and at most d exponentiations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

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 [9] is a new OR proof with the argument consisting of 7 group elements. Due to the importance of $\mathcal{L}_{\{0,1\}}$, in the full version [10], 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. 4, 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.



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

Fig. 4. ABP example for $F(X,Y) = X^3 + aX + b - Y^2$.

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 [4], 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 [32], 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 [14] 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.* [32] 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 [14,32] 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.* [32] 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 the full version [10], we will provide an example for $d=5$. We refer to [32] 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, [32]. In both cases, we chose the matrix by inspection.

Case $F(X,Y) = X^3 + aX + b - Y^2$ for $a \neq 0$. In the full version [10], 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 the full version [10].

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 [16]) 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 [20] 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.

Arithmetic circuits. Let $\mathfrak{C}$ be an arithmetic circuit over $\mathbb{Z}_p$, with n gates (including input gates) and m wires. We construct an algebraic set $\mathcal{A}_{\mathfrak{C}} = (\chi_1, \ldots, \chi_n) \in \mathbb{Z}_p^n$, such that $\boldsymbol{\chi} \in \mathcal{A}_{\mathfrak{C}}$ iff $\mathfrak{C}(\boldsymbol{\chi}) = 0$, as follows. First, $\boldsymbol{\chi}$ corresponds to the vector of wire values. As in the case of QAP [16], we assume that each gate is a weighted multiplication gate that computes $F_i : (\sum_j u_{ij}\chi_{i_j})(\sum_j v_{ij}\chi_{i_j}) \mapsto \chi_i$ for public u_{ij}, v_{ij}, and i_j, where for the sake of efficiency, the sum is taken over a constant number of values.

1. First, each χ_i corresponds to the value of the output wire of ith gate, with χ_j, $j \leq \mathsf{m}_0$ corresponding to the inputs of the circuit. We also assume that the last few wire values correspond to the output values of the circuit.
2. Second, for each gate $i > \mathsf{m}_0$, we introduce the polynomial $F_i(\boldsymbol{\chi}) = \chi_i - (\sum u_{ij}\chi_{i_j})(\sum v_{ij}\chi_{i_j})$.

Then $\mathcal{A}_{\mathfrak{C}} = \{(\chi_1, \ldots, \chi_m) : F_i(\boldsymbol{\chi}) = 0 \text{ for all } i > \mathsf{m}_0\}$. To construct a NIZK for showing $\boldsymbol{\chi} \in \mathcal{A}_{\mathfrak{C}}$, we do as before: (1) We let the prover Elgamal-encrypt $\boldsymbol{\chi}$. (2) We show that $F_i(\boldsymbol{\chi}) = 0$ for all i by using the NIZK argument from Section 4. Note that each polynomial in this case is quadratic, and thus one can construct a 2×2 QDR $\boldsymbol{C}(\boldsymbol{\chi}) = \left(\begin{smallmatrix} \sum u_{ij}\chi_{i_j} & -1 \\ -\chi_i & \sum v_{ij}\chi_{i_j} \end{smallmatrix}\right)$.

According to [21], 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 [21] can probably be optimized, we have not included complete comparison.

Extension: R1CS. In R1CS (*rank-1 constraint system* [16]), 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 $\boldsymbol{C}(\boldsymbol{\chi}) = \left(\begin{smallmatrix} \sum u_{ij}\chi_{i_j} & -1 \\ -\sum w_{ij}\chi_{i_j} & \sum v_{ij}\chi_{i_j} \end{smallmatrix}\right)$.

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

One can extend R1CS to q-aCSP for small constant q, assuming that F_j are (small-degree) polynomials for which one can construct poly-size QDRs. Intuitively, F is the generating set for some polynomial ideal $\mathcal{I} = \mathcal{I}(\mathcal{A})$, and thus the examples of this subsection fall under our general methodology. One can possibly use some general techniques (see Section 8.2 for some examples) to minimize the generating sets so as to obtain more efficient NIZKs.

Specialization: Boolean Circuits. By using techniques from [20], one can construct a NIZK for any Boolean circuit that, w.l.o.g., consists of only NAND gates. Intuitively, one does this by showing that each wire value is Boolean, and then showing that each NAND gate is followed correctly. The latter can be shown by showing that a certain linear combination of the input and output wires of the NAND gate is Boolean. Thus, here one only uses polynomials of type $f_i(\boldsymbol{\chi}) = A(\boldsymbol{\chi})^2 - A(\boldsymbol{\chi})$, where $A(\boldsymbol{\chi}) = \sum a_{ij}\chi_j$ for some coefficients a_{ij}.

In Table 2, we compare the resulting NIZK with the optimized Groth-Sahai proof for Boolean circuits by Ghadafi *et al.* [18]. Here, m is the number of wires and n is the number of gates. In the case of the AES circuit described in [18], $m = 33880$ 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 [18] 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

Table 2. Comparison of falsifiable NIZKs for Boolean circuit satisfiability: the Groth-Sahai proof, as optimized by Ghadafi *et al.* [18], and the new NIZK from Section 8.1. Here, $|\mathbb{G}_\iota|$ is the length of one element from $\mathbb{G}_\iota$

Protocol	$\|\mathsf{crs}\|$	$\|\mathsf{com}\|$	$\|\pi\|$	P comp.	V comp.
Groth-Sahai [18]	$4(\|\mathbb{G}_1\|+\|\mathbb{G}_2\|)$	$2(m+1)(\|\mathbb{G}_1\|+\|\mathbb{G}_2\|)$	$(6m+2n+2)(\|\mathbb{G}_1\|+\|\mathbb{G}_2\|)$	$(12m+4n+4)(\mathfrak{e}_1+\mathfrak{e}_2)$	$16(2m+n)\mathfrak{p}$
New, Section 8.1	$\|\mathbb{G}_2\|$	$2m\cdot\|\mathbb{G}_1\|$	$(m+n)(4\|\mathbb{G}_1\|+3\|\mathbb{G}_2\|)$	$(m+n)(5\mathfrak{e}_1+4\mathfrak{e}_2)$	$13(m+n)\mathfrak{p}$

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 4 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 [7] 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 [27] (as say implemented by `CoCoA`[8]) can compute a Gröbner basis for $\mathcal{I}(\mathcal{A})$, given any finite set $\mathcal{A}$.

Worst-Case Multi-Dimensional Set-Membership Proof. We performed an exhaustive computer search to come up with an example of a 3-dimensional set of five points that has the least efficient NIZK argument in our framework. One of the examples we found[9] is $\mathcal{A}=\{(2,5,1),(2,4,2),(2,5,3),(1,2,4),(3,1,5)\}$. In this case, we found a reduced degree-lexicographic Gröbner basis

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

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

[9] In the case of many other sets, the NIZK will be much more efficient. We will provide one concrete example in the full version [10].

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. 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 the full version [10], 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 the full version [10], 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, [5,3]) 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, [1,9]. 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 $\langle F_1, \ldots, F_\tau \rangle$ is some generating set of $\mathcal{I}(\mathcal{A})$, and output concrete parameters $(\boldsymbol{\Gamma}, \boldsymbol{\theta})$. The problem of constructing such $\boldsymbol{C}_i$ was already tackled in the current paper, with many examples (including the case when $\boldsymbol{C}_i$ is based on an ABP). As the end result, we construct explicit language parameters $(\boldsymbol{\Gamma}, \boldsymbol{\theta})$ for a variety of languages where no such small parameters were known before. Moreover, even in the simple case of univariate polynomials, where previous solutions were known [5,9], 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 [1]

and usually more efficient than the CHM NIZK of [9]. Finally, we briefly discuss resulting GL-SPHFs [17] based on the new language parameters.

Preliminaries. We describe the CHM (Couteau-Hartmann-Maurer) Σ-protocol and the resulting NIZK in the full version [10]. There, we will also state the efficiency of their construction as a function of $(\boldsymbol{\Gamma}, \boldsymbol{\theta})$. We also restate Theorem 18 from [9] 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 $\mathcal{I}(\mathcal{A}) = \langle F_1, \ldots, F_\tau \rangle$ is not a principal ideal, one can then "concatenate" all τ parameters $\boldsymbol{\Gamma}(\mathtt{x})$ and $\boldsymbol{\theta}(\mathtt{x})$.

We start the derivation from the equation $\boldsymbol{T}(\boldsymbol{\chi})\boldsymbol{w} = \boldsymbol{h}(\boldsymbol{\chi})$ in 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 $[\mathsf{ct}_i]_1^\top$) and constants are encrypted by using the randomizer 0. We define $[\boldsymbol{\Gamma}(\mathtt{x})]_1$ and $[\boldsymbol{\theta}(\mathtt{x})]_1$ as follows:

$$[\boldsymbol{\Gamma}(\mathtt{x})]_1 = (\mathcal{E}(\boldsymbol{T}(\boldsymbol{\chi})) \| \mathcal{E}(\mathbf{0}_{d\times d}; \boldsymbol{I}_d)) \in \mathbb{G}_2^{2d\times(2d-1)} \ , \ [\boldsymbol{\theta}(\mathtt{x})]_1 = \mathcal{E}(\boldsymbol{h}(\boldsymbol{\chi})) \in \mathbb{G}_2^{2d} \ . \quad (3)$$

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. (3) results in

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

A variation of this $[\boldsymbol{\Gamma}, \boldsymbol{\theta}]_1$ was given in [5,9]. 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 $\mathbf{0}_4$ iff $\chi \in \{0, 1\}$.

Theorem 2. $\mathcal{L}_{\mathsf{pk},F} = \mathcal{L}_{\boldsymbol{\Gamma},\boldsymbol{\theta}}$.

In the full version [10], 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 $\boldsymbol{\Gamma} = \left(\begin{smallmatrix} \boldsymbol{\Gamma}_1 & \ldots & 0 \\ \ldots & \ldots & \ldots \\ 0 & \ldots & \boldsymbol{\Gamma}_\tau \end{smallmatrix}\right)$ and $\boldsymbol{\theta} = \left(\begin{smallmatrix} \boldsymbol{\theta}_1 \\ \ldots \\ \boldsymbol{\theta}_\tau \end{smallmatrix}\right)$.

On the Couteau-Hartmann Disjunction. In the full version [10], we describe the Couteau-Hartmann disjunction that results in $\boldsymbol{\Gamma}$ of size $(3d-1)\times(3d-2)$ and compare it to Eq. (3). For the sake of completeness, we also reprove the efficiency of the CHM NIZK from [9].

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 [9] 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 [9] for $\mathcal{L}_{\boldsymbol{\Gamma},\boldsymbol{\theta}} = \mathcal{L}_{\mathsf{pk},F}$ has quite a big overhead. Thus, the NIZK in Lemma 7 is quite a bit more efficient. However, it compares favorably to [9]. In the following lemma, we state its efficiency.

Lemma 8. *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. (3). 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.*

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 [9] 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, [17]) based on the new $\mathtt{lpar} = (\boldsymbol{\Gamma},\boldsymbol{\theta})$. We refer the reader to [13,5,3] 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 [17], $\mathtt{lpar}$ 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 [5], 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. (3). In the full version [10], we will describe a GL-SPHF for the language of elliptic curve points.

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. 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 (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). We will prove that CED assumptions are implied by a new but natural *gap assumption* [30] 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 [6].

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 V_f, such that given an arbitrary ciphertext tuple $\mathtt{x} = [\mathbf{ct}_1, \ldots, \mathbf{ct}_\nu]_1$ and an sk-dependent trapdoor $\mathbf{T}$, $\mathsf{V}_\mathsf{f}(\mathsf{p}, \mathsf{pk}, \mathtt{x}, \mathbf{T})$ can efficiently check whether $\mathsf{Dec}_\mathsf{sk}([\mathbf{ct}_1, \ldots, \mathbf{ct}_\nu]_1) \in \mathcal{L}_{\mathsf{pk},F}$. As in the rest of the paper, we take $\mathbf{T} = \mathsf{sk}$. Thus, given a ciphertext tuple $[\mathbf{ct}]_1$, V_f can use sk to decrypt it and obtain the plaintext $[\boldsymbol{\chi}]_1$. 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 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 $\mathcal{A}(\mathcal{I})$ has polynomial size.

The outliers are the cases of principal ideals of multivariate polynomials (since then $|\mathcal{A}(\mathcal{I})|$ can be exponential as in the set of points (X, Y) on an elliptic curve) and some instances of non-principal ideals where $|\mathcal{A}(\mathcal{I})|$ is super-polynomial. In the latter case, we can clarify the situation further. Namely, given a generating set $\langle F_1, \ldots, F_\tau \rangle$, by Bézout's theorem, $\mathcal{A}(\mathcal{I})$ has at most size $\prod \deg F_i$. Assuming each $\deg F_i$ is $\mathsf{poly}(\lambda)$, $\prod \deg F_i$ is super-polynomial if $\tau = \omega(1)$. Thus, constant-size set-membership arguments in Section 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 $\mathcal{A}(\mathcal{I})$ does not mean that efficient $\mathsf{V_f}$ does not exist. E.g., assume $F_j(\boldsymbol{X}) = \prod_i (X_i - s_j)$ for each j. The ideal $\langle F_j \rangle$, for a single j, has exponential size. However, given $[\boldsymbol{\chi}]_1$, one can check if $F_j(\boldsymbol{\chi}) = 0$ by checking if $\chi_i = s_j$ for some j. This can be generalized to the case F_j is a product of affine multivariate polynomials $\sum a_{ik} X_k + b_{ik}$. Clearly, $F(\boldsymbol{\chi}) = 0$ iff one of its affine factors is equal to 0. So, $\mathsf{V_f}$ can check if there exists an i such that $\sum a_{ik}[\chi_k]_1 + b_{ik}[1]_1 = [0]_1$. Generalizing this, one can efficiently establish whether $[\boldsymbol{C}]_1$ is full-rank if the Leibniz formula for the determinant, $\det(\boldsymbol{C}) = \sum_{\sigma \in S_n} (\mathrm{sgn}(\sigma) \prod_{i=1}^n C_{i,\sigma_i})$, contains only one non-zero addend.

On the other hand, since $\mathsf{V_f}$ has only access to $[\boldsymbol{\chi}]_1$, there is not much hope that the CED assumption is falsifiable if F is a product of irreducible polynomials, such that at least one of them has a total degree greater than one, unless we add some additional, carefully chosen, elements to the proof for this purpose. In the general case, this is not efficient, but the number of additional needed elements might not be prohibitive for some applications.

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

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 [28] 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 $\mathcal{A}$,*

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

Theorem 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.*

Note that in particular, this re-proves the result of [9] 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, [2] 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 [2], consider the case $[\boldsymbol{C}]_1 \in \mathbb{G}_1^{2\times 2}$. [2] 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 [2] was that it changed the assumption they used from KerMDH to the less standard SKerMDH assumption [19]. As we see next, we have to use the DLIN cryptosystem [6] 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 (see the full version [10] for its definition). Given $\mathsf{sk} = (\mathsf{sk}_1, \mathsf{sk}_2)$ and $\mathsf{pk}_\iota = [1, \mathsf{sk}_1, \mathsf{sk}_2]_\iota$, we define $\mathtt{lpar} := (\mathsf{pk}_1, \mathsf{pk}_2, F)$. Then, $\mathcal{L}_{\mathtt{lpar}} := \{([\mathbf{ct}_1, \mathbf{ct}_2]_1, [\mathbf{ct}_1]_2)\}$, where $[\mathbf{ct}_1]_\iota = \mathsf{Enc}_\iota(X; r_1, r_2) = [r_1\mathsf{sk}_1, r_2\mathsf{sk}_2, X + r_1 + r_2]_\iota$ and $[\mathbf{ct}_2]_1 = \mathsf{Enc}_1(Y; r_3, r_4) = [r_3\mathsf{sk}_1, r_4\mathsf{sk}_2, Y + r_3 + r_4]_1$. We prove that $[\mathbf{ct}_1, \mathbf{ct}_2]_1$ are encryptions of X and Y such that $X^2 = Y$, by using the QDR $\boldsymbol{C}(X, Y) = \left(\begin{smallmatrix} X & -1 \\ -Y & X \end{smallmatrix}\right)$. The use of the DLIN encryption scheme just affects the efficiency and the communication size of the protocol. In addition, one can check that $[\mathbf{ct}_1]_1$ and $[\mathbf{ct}_1]_2$ encrypt the same X in two different groups by checking that $[\mathbf{ct}_1]_1 \bullet [1]_2 = [1]_1 \bullet [\mathbf{ct}_1]_2$.

Since the DLIN encryption is doubly-homomorphic like Elgamal, then the argument of Section 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

1. Abdalla, M., Benhamouda, F., Pointcheval, D.: Disjunctions for hash proof systems: New constructions and applications. In: EUROCRYPT 2015, Part II. LNCS, vol. 9057, pp. 69–100
2. Abdolmaleki, B., Lipmaa, H., Siim, J., Zajac, M.: On QA-NIZK in the BPK model. In: PKC 2020, Part I. LNCS, vol. 12110, pp. 590–620
3. Ben Hamouda-Guichoux, F.: Diverse Modules and Zero-Knowledge. PhD thesis, PSL Research University (2016)
4. Ben-Sasson, E., Chiesa, A., Tromer, E., Virza, M.: Scalable zero knowledge via cycles of elliptic curves. In: CRYPTO 2014, Part II. LNCS, vol. 8617, pp. 276–294
5. Benhamouda, F., Blazy, O., Chevalier, C., Pointcheval, D., Vergnaud, D.: New techniques for SPHFs and efficient one-round PAKE protocols. In: CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 449–475
6. Boneh, D., Boyen, X., Shacham, H.: Short group signatures. In: CRYPTO 2004. LNCS, vol. 3152, pp. 41–55
7. Buchberger, B.: An Algorithm for Finding the Basis Elements of the Residue Class Ring of a Zero Dimensional Polynomial Ideal. PhD thesis, University of Innsbruck (1965)
8. Chaidos, P., Couteau, G.: Efficient designated-verifier non-interactive zero-knowledge proofs of knowledge. In: EUROCRYPT 2018, Part III. LNCS, vol. 10822, pp. 193–221
9. Couteau, G., Hartmann, D.: Shorter non-interactive zero-knowledge arguments and ZAPs for algebraic languages. In: CRYPTO 2020, Part III. LNCS, vol. 12172, pp. 768–798
10. Couteau, G., Lipmaa, H., Parisella, R., Ødegaard, A.T.: Efficient NIZKs for Algebraic Sets. Technical report, IACR (2021)
11. Cox, D.A., Little, J., O'Shea, D.: Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra. 4 edn. Undergraduate Texts in Mathematics. Springer (2015)
12. Cramer, R., Damgård, I., Schoenmakers, B.: Proofs of partial knowledge and simplified design of witness hiding protocols. In: CRYPTO'94. LNCS, vol. 839, pp. 174–187
13. Cramer, R., Shoup, V.: Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In: EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64

14. Dickson, L.E.: Determination of All General Homogeneous Polynomials Expressible as Determinants with Linear Elements. Trans. of the American Mathematical Society **22**(2) (1921) pp. 167–179
15. Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for Diffie-Hellman assumptions. In: CRYPTO 2013, Part II. LNCS, vol. 8043, pp. 129–147
16. Gennaro, R., Gentry, C., Parno, B., Raykova, M.: Quadratic span programs and succinct NIZKs without PCPs. In: EUROCRYPT 2013. LNCS, vol. 7881, pp. 626–645
17. Gennaro, R., Lindell, Y.: A framework for password-based authenticated key exchange. In: EUROCRYPT 2003. LNCS, vol. 2656, pp. 524–543. `https://eprint.iacr.org/2003/032.ps.gz`.
18. Ghadafi, E., Smart, N.P., Warinschi, B.: Practical zero-knowledge proofs for circuit evaluation. In: 12th IMA International Conference on Cryptography and Coding. LNCS, vol. 5921, pp. 469–494
19. González, A., Hevia, A., Ràfols, C.: QA-NIZK arguments in asymmetric groups: New tools and new constructions. In: ASIACRYPT 2015, Part I. LNCS, vol. 9452, pp. 605–629
20. Groth, J., Ostrovsky, R., Sahai, A.: Non-interactive zaps and new techniques for NIZK. In: CRYPTO 2006. LNCS, vol. 4117, pp. 97–111
21. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432
22. Hoffmann, M., Klooß, M., Rupp, A.: Efficient zero-knowledge arguments in the discrete log setting, revisited. In: ACM CCS 2019, pp. 2093–2110
23. Ishai, Y., Kushilevitz, E.: Randomizing polynomials: A new representation with applications to round-efficient secure computation. In: 41st FOCS, pp. 294–304
24. Ishai, Y., Kushilevitz, E.: Perfect constant-round secure computation via perfect randomizing polynomials. In: ICALP 2002. LNCS, vol. 2380, pp. 244–256
25. Kosba, A.E., Zhao, Z., Miller, A., Qian, Y., Chan, T.H., Papamanthou, C., Pass, R., Shelat, A., Shi, E.: CØCØ: A Framework for Building Composable Zero-Knowledge Proofs. Technical Report 2015/1093, International Association for Cryptologic Research (2015) `https://ia.cr/2015/1093`, last accessed version 9 Apr 2017.
26. Maurer, U.M.: Unifying zero-knowledge proofs of knowledge. In: AFRICACRYPT 09. LNCS, vol. 5580, pp. 272–286
27. Möller, H.M., Buchberger, B.: The Construction of Multivariate Polynomials with Preassigned Zeros. In: EUROCAM 1982. LNCS, vol. 144, pp. 24–31
28. Morillo, P., Ràfols, C., Villar, J.L.: The kernel matrix Diffie-Hellman assumption. In: ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 729–758
29. Nisan, N.: Lower bounds for non-commutative computation (extended abstract). In: 23rd ACM STOC, pp. 410–418
30. Okamoto, T., Pointcheval, D.: The gap-problems: A new class of problems for the security of cryptographic schemes. In: PKC 2001. LNCS, vol. 1992, pp. 104–118
31. Pass, R.: Unprovable security of perfect NIZK and non-interactive non-malleable commitments. In: TCC 2013. LNCS, vol. 7785, pp. 334–354
32. Plaumann, D., Sturmfels, B., Vinzant, C.: Computing Linear Matrix Representations of Helton-Vinnikov Curves. Mathematical Methods in Systems, Optimization, and Control Operator Theory **222** (2012) pp. 259–277
33. Ràfols, C.: Stretching groth-sahai: NIZK proofs of partial satisfiability. In: TCC 2015, Part II. LNCS, vol. 9015, pp. 247–276