Succinct Diophantine-Satisfiability Arguments

Patrick Towa[1,2], Damien Vergnaud[3]

[1] IBM Research – Zurich
[2] DIENS, École Normale Supérieure, CNRS, PSL University, Paris, France
[3] Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
[4] Institut Universitaire de France

Abstract. A *Diophantine equation* is a multi-variate polynomial equation with integer coefficients, and it is satisfiable if it has a solution with all unknowns taking integer values. Davis, Putnam, Robinson and Matiyasevich showed that the general Diophantine satisfiability problem is undecidable (giving a negative answer to Hilbert's tenth problem) but it is nevertheless possible to argue in zero-knowledge the knowledge of a solution, if a solution is known to a prover.

We provide the first succinct honest-verifier zero-knowledge argument for the satisfiability of Diophantine equations with a communication complexity and a round complexity that grows logarithmically in the size of the polynomial equation. The security of our argument relies on standard assumptions on hidden-order groups. As the argument requires to commit to integers, we introduce a new integer-commitment scheme that has much smaller parameters than Damgård and Fujisaki's scheme. We finally show how to succinctly argue knowledge of solutions to several NP-complete problems and cryptographic problems by encoding them as Diophantine equations.

1 Introduction

A *Diophantine equation* is a multi-variate polynomial equation with integer coefficients, and it is satisfiable if it has a solution with all unknowns taking integer values. Davis, Putnam, Robinson and Matiyasevich [?] showed that any computational problem can be modeled as finding a solution to such equations, thereby proving that the general Diophantine-satisfiability problem is undecidable and giving a negative answer to Hilbert's tenth problem. For instance, several classical NP-problems such as 3-SAT, Graph 3-colorability or Integer Linear Programming can be readily encoded as Diophantine equations. Several cryptographic problems such as proving knowledge of an RSA signature, that a committed value is non-negative or that encrypted votes are honestly shuffled by a mix-net, can also be encoded as Diophantine equations.

Efficient zero-knowledge arguments of knowledge of solutions to Diophantine equations, if a solution is known to a party, can thus be useful for many practical cryptographic tasks; and doing so requires to do zero-knowledge proofs on committed integers.

1.1 Prior Work

Integer Commitments. Fujisaki and Okamoto [?] presented the first efficient integer commitment scheme and also suggested a zero-knowledge protocol for verifying multiplicative relations over committed values. Such a commitment scheme allows to commit to any $x \in \mathbb{Z}$ in a group of unknown order, with a Pedersen-like commitment scheme. This makes the security analysis more intricate since division modulo the unknown group order cannot be performed in general. As an evidence that this setting is error-prone, Michels showed that the Fujisaki–Okamoto proof system was flawed. Damgård and Fujisaki [?] later proposed a statistically hiding and computationally binding integer commitment scheme under standard assumptions in a hidden-order group $\mathbb{G}$ with an efficient argument of knowledge of openings to commitments, and arguments of multiplicative relations over committed values. This primitive gives rise to a (honest-verifier) zero-knowledge proof of satisfiability of a Diophantine equation with M multiplications over $\mathbb{Z}$ that requires $\Omega(M)$ integer commitments and $\Omega(M)$ proofs of multiplicative relations [?, ?]. These complexities have not been improved since then.

Circuit Satisfiability over $\mathbb{Z}_p$. Similarly, it is possible to design a zero-knowledge proof of satisfiability of an arithmetic circuit over $\mathbb{Z}_p$ using Pedersen's commitment scheme [?] in a group $\mathbb{G}$ of public prime order p. An immediate solution is to use the additive homomorphic properties of Pedersen's commitment and zero-knowledge protocols for proving knowledge of the contents of commitments and for verifying multiplicative relations over committed values [?, ?].

For an arithmetic circuit with M multiplication gates, this protocol requires $\Omega(M)$ commitments and $\Omega(M)$ arguments of multiplication consistency and has a communication complexity of $\Omega(M)$ group elements. In 2009, Groth [?] proposed a sub-linear size zero-knowledge arguments for statements involving linear algebra and used it to reduce this communication complexity to $O\left(\sqrt{M}\right)$ group elements. This breakthrough initiated a decade of progress for zero-knowledge proofs for various statements (see e.g., [?, ?, ?, ?] and references therein). It culminated with the argument system "*Bulletproofs*" proposed by Bünz, Bootle, Boneh, Poelstra, Wuille and Maxwell [?] which permits to prove the satisfiability of such an arithmetic circuit with communication complexity $O(\log(M))$ and round complexity $O(\log(M))$. The corner stone of their protocol is an argument that two committed vectors satisfy an inner-product relation. It has logarithmic communication and round complexity in the vector length, and its security only relies on the discrete-logarithm assumption and does not require a trusted setup.

Circuit satisfiability over any finite field is an NP-complete problem so the "*Bulletproofs*" argument system has widespread applications. However, as mentioned above, in many cryptographic settings, it is desirable to prove statements such as "the committed value x is a valid RSA signature on a message m for an RSA public key (N, e)". In this case, the prover has to convince the verifier that $x^e = H(m) \bmod N$, or in other words that there exists an integer k such that $x^e + kN = H(m)$ where this equality holds over the integers for $|k| \leq N^{e-1}$

and H some cryptographic hash function. In order to use directly an argument of satisfiability of an arithmetic circuit to prove the knowledge of a pair (x, k) which satisfies this equation, one needs to use a group $\mathbb{G}$ a prime order p with $p > N^e$ (and to additionally prove that $x < N$ and $k < N^e$). For a large e, this approach results in a proof with prohibitive communication complexity.

Moreover, in various settings, such as the Integer-Linear-Programming problem, there is no *a priori* upper-bound on the sizes of the integer solutions during setup when p is defined. Being able to argue on integers instead of residue classes modulo a fixed prime integer then becomes necessary. Besides, generic reductions to circuit satisfiability over prime-order fields for some simple problems naturally defined over the integers may return circuits with a very large number of multiplication gates and even the "*Bulletproofs*" argument system could produce large proofs. Modeling computational problems using Diophantine equations is more versatile, and a succinct argument system for Diophantine satisfiability thus has many potential applications.

1.2 Contributions

We provide the first succinct argument for the satisfiability of Diophantine equations with a communication complexity and a round complexity that grows logarithmically in the size of the polynomial equation[5]. It is statistical honest-verifier zero-knowledge and is extractable under standard computational assumptions over hidden-order groups such as RSA groups or ideal-class groups.

Integer Commitments. Section **??** introduces a new computationally hiding and binding commitment scheme that allows to commit to vectors of integers. It is close to Damgård and Fujisaki's seminal proposal, but has much smaller parameters. Denoting by λ the security parameter and letting $2^{b_{\mathbb{G}}}$ be an upper bound on the group order, the version of our scheme which allows to commit to n integers at once has parameters consisting of $O(b_{\mathbb{G}} + \log n)$ bits instead of $\Omega\left(nb_{\mathbb{G}} \cdot \operatorname{polylog}(\lambda)\right)$ as with the generalized version of Damgård and Fujisaki's scheme.

Damgård and Fujisaki's commitment scheme, for $n = 1$, is a variant of Pedersen's commitment in a hidden-order group $\mathbb{G}$: given two group elements $g, h \in \mathbb{G}$, the commitment to an integer value $x \in \mathbb{Z}$ is $C = g^x h^r$, where r is an integer of appropriate size. The hiding property of their scheme crucially relies on the fact that $g \in \langle h \rangle$, which is not always guaranteed as the group may not be cyclic. Damgård and Fujisaki's proposed a Schnorr-type [**?**] protocol to prove such statements, but their challenge set is restricted to $\{0, 1\}$ to guarantee soundness under the assumptions on the group. Their protocol must then be repeated logarithmically many times to achieve negligible soundness, and the resulting parameters

[5] Our goals and techniques differ completely from those proposed by Bünz, Fisch and Szepieniec [**?**] where they used what they called *Diophantine Arguments of Knowledge (DARK)* to construct a commitment scheme for polynomials over prime finite fields (using the so-called *Kronecker substitution* for determining the coefficients of a polynomial by evaluating it at a single value, see e.g., [**?**, p. 245]).

are large. The situation is worse when n is large as commitments are computed as $g_1^{x_1} \cdots g_n^{x_n} h^r$ and a proof for each g_i must be computed.

Our scheme is based on the observation that proving that $g^2 \in \left\langle h^2 \right\rangle$ can be done more efficiently in a single protocol run under the assumptions on the group. Our commitments are thus computed as $(g^x h^r)^2 \in \mathbb{G}$. We further such how to aggregate the proofs of several such statements to reduce the size of our parameters when n is large.

Succinct Inner-Product Arguments on Integers. Section **??** presents a succinct argument that two integer vectors committed with our scheme satisfy an inner-product relation. That is, an argument of knowledge of vectors $\mathbf{a}$ and $\mathbf{b} \in \mathbb{Z}^n$ (and of a randomness $r \in \mathbb{Z}$) that open a commitment C and such that $\langle \mathbf{a}, \mathbf{b} \rangle = z$ given a public integer z. Succinct here means that the communication complexity of the prover is of order $O(\ell + \log(n) b_{\mathbb{G}})$, where ℓ is the bit length of the largest witness. The complexity is measured in bits as during the protocol, the prover sends logarithmically many group elements and three integers, but these latter could be arbitrarily large.

The argument of Bünz et al. [**?**] for inner-product relations over $\mathbb{Z}_p$ is not applicable to integers as their proof of extractability relies on the generalized discrete-logarithm assumption for which there is no equivalent in hidden-order groups that may not even be cyclic, and on the invertibility of elements in $\mathbb{Z}_p^*$ since it requires to solve linear systems over $\mathbb{Z}_p$. Besides, their argument is not zero-knowledge and is on vectors committed with the non-hiding version of Pedersen's scheme (i.e., with nil randomness). Therefore, whenever it is used as a sub-protocol of another one, techniques specific to the larger protocol must always be used to guarantee that it is zero-knowledge. del Pino, Seiler and Lyubashevsky [**?**] later solved this issue by adapting the argument of Bünz et al. in prime-order groups to make it perfectly honest-verifier zero-knowledge with the full-fledged Pedersen's scheme.

Our protocol uses halve-then-recurse techniques similar to those of Bünz et al. for the Section-**??** commitment scheme in hidden-order groups and thus allows to succinctly argue on integers, but only uses the integrality of $\mathbb{Z}$ as a ring since one cannot invert modulo the unknown order. (Note that these techniques are themselves inspired by the recursive inner-product argument of Bootle et al. [**?**].) In particular, we prove that even though one cannot a priori solve in $\mathbb{Z}$ the linear system of Bünz et al. required to prove the extractability of their protocol, one can instead solve a "relaxed" system in $\mathbb{Z}$. Then, under the assumptions on the hidden-order group, we show that the solution to the relaxed system is enough to extract a representation of the commitment in the public bases. In groups with public prime orders, the assumption that discrete-logarithm relations are hard to compute allows to conclude that this representation of the commitment actually leads to a valid witness, but this assumption is not a priori translatable to hidden-order groups. Instead, we prove that a similar assumption in the subgroup generated by a randomly sampled element is weaker than the assumptions on the group, and that suffices to prove the extractability of the protocol. The details of these technical challenges are outlined in Section **??**.

Furthermore, as the group order is unknown to all parties, the argument is only statistically honest-verifier zero-knowledge. To ensure this property, the randomness range of the prover is carefully[6] adapted to allow for simulatability without knowledge of a witness.

Succinct Arguments for Diophantine Equations. Section **??** presents our succinct protocol to argue satisfiability of Diophantine equations. Our approach is inspired by Skolem's method [**?**] which consists in reducing the degree of the polynomial by introducing new variables to obtain a new polynomial of degree at most 4, in such a way that the satisfiability of one polynomial implies that of the other. Tailoring Skolem's method to the problem of arguing satisfiability, we show how to reduce the satisfiability of any polynomial in $\mathbb{Z}[x_1, \ldots, x_\nu]$ of total degree δ with μ monomials to the existence of vectors $\mathbf{a}_L = \left[a_{L,1} \cdots a_{L,n}\right]$, $\mathbf{a}_R = \left[a_{R,1} \cdots a_{R,n}\right]$ and $\mathbf{a}_O = \left[a_{O,1} \cdots a_{O,n}\right]$ in $\mathbb{Z}^n$, for $n \leq \nu\lfloor\log\delta\rfloor + (\delta - 1)\mu$, such that $a_{O,i} = a_{L,i}a_{R,i}$ for all $i \in \{1, \ldots, n\}$, and that satisfy $1 \leq Q \leq 1 + 2\nu(\lfloor\log\delta\rfloor - 1) + (\delta - 2)\mu$ linear constraints of the form

$$\langle \mathbf{w}_{L,q}, \mathbf{a}_L \rangle + \langle \mathbf{w}_{R,q}, \mathbf{a}_R \rangle + \langle \mathbf{w}_{O,q}, \mathbf{a}_O \rangle = c_q,$$

where $\mathbf{w}_{L,q}, \mathbf{w}_{R,q}, \mathbf{w}_{O,q} \in \mathbb{Z}^n$ and $c_q \in \mathbb{Z}$ for all $q \in \{1, \ldots, Q\}$. Our reduction is constructive as it allows to infer the vectors and the constraints directly from the original polynomial.

Bootle et al. [**?**] then Bünz et al. [**?**] gave an argument system for proving knowledge of vectors in $\mathbb{Z}_p$ (instead of $\mathbb{Z}$) that satisfy such constraints. They use this protocol to argue for the satisfiability of arithmetic circuits over $\mathbb{Z}_p$. Our argument shares similarities with theirs, but again there are key technical differences that arise from the fact that $\mathbb{Z}$ is not a field. Indeed, as one cannot invert nor reduce integers modulo the unknown orders of the bases, we use different techniques notably to prevent the integers involved in the argument from increasing too much, and to ensure consistency between the variables in the entry-wise product and those in the linear constraints. Guaranteeing this latter consistency requires to construct new polynomials for the argument that do not involve inverting integers. Besides, one cannot use their commitment-key switching technique which consists in interpreting g^a as a commitment to xa to the base $g^{x^{-1}}$ in groups of public prime order. Finally, extra precaution must be taken to guarantee the zero-knowledge property as integers are not reduced modulo p and may carry information about the witness. These challenges and the ways we overcome them are described in details in the full version [**?**, Section 6.2].

As a result, the communication complexity of our Diophantine-satisfiability argument has a communication complexity of $O\left(\delta\ell + \min(\nu, \delta)\log(\nu + \delta)\, b_{\mathbb{G}} + H\right)$ bits, if the absolute value of all the polynomial coefficients is upper-bounded

[6] As another evidence that cryptography in hidden-order groups is error prone, Fouque and Poupard [**?**] broke the RDSA signature from [**?**] for which this randomness range was not wisely selected.

by 2^H for some integer H. In contrast, the overall communication complexity using Damgård and Fujisaki's multiplication argument is upper-bounded by $O\left(\binom{\nu+\delta}{\delta}\left(\delta\ell+\log\left(\binom{\nu+\delta}{\delta}\right)H+b_{\mathbb{G}}\right)\right)$ and lower-bounded by $\Omega\left(\binom{\nu+\delta}{\delta}(\ell+b_{\mathbb{G}})\right)$.

Applications. The full version [?] presents several applications of our Diophantine-satisfiability argument. We provide explicit reductions to Diophantine satisfiability for the following problems:

- argument of knowledge of a (possibly committed) RSA e-th root in $\mathbb{Z}_N$ of some public value with $O(\log(\log(e))b_{\mathbb{G}})$ bits. This has application to credential systems when combined with proofs of non-algebraic statements [?];
- argument of knowledge of $O(\log(\log p)b_{\mathbb{G}})$ bits for ECDSA signatures with a prime p, and of $O(\log(\log q)b_{\mathbb{G}}+\log(\log p))$ bits for DSA signatures with primes p and q. The signed message is public, but can be committed if the argument is combined with proofs of non-algebraic statements [?];
- argument that two committed lists of integers of length n are permutations of each other with $O(\ell+\log(n)b_{\mathbb{G}})$ bits
- argument of satisfiability of a 3-SAT Boolean formula with m clauses and n variables with $O(\log(n+m)b_{\mathbb{G}})$ bits;
- argument of satisfiability of an Integer-Linear-Programming problem of the form $\mathbf{x}\in\mathbb{N}^n$ and $\mathbf{A}\mathbf{x}^{\mathrm{T}}\geq\mathbf{b}^{\mathrm{T}}$, for $\mathbf{A}\in\mathbb{Z}^{m\times n}$ and $\mathbf{b}\in\mathbb{Z}^m$, with $O(\ell+\log(4n+3m)$ $b_{\mathbb{G}}+\log\|\mathbf{A}\|_\infty+\log\|\mathbf{b}\|_\infty)$ bits.

2 Preliminaries

This section introduces the notation used throughout the paper, recalls standard assumptions on generators of hidden-order groups, and defines commitment schemes and argument systems.

2.1 Notation

For $x\in\mathbb{Z}$, $|x|$ denotes its absolute value. All logarithms are in base 2. For any two integers $a\leq b\in\mathbb{Z}$, $[\![a;b]\!]$ denotes the set $\{a\}$ if $a=b$ and $\{a,a+1,\ldots,b\}$ if $a<b$. For an integer $n\geq 1$, $[\![n]\!]$ stands for the set $[\![1;n]\!]$. Given a vector $\mathbf{a}\in\mathbb{Z}^n$, $\mathbf{a}X$ denotes the vector $\begin{bmatrix}a_1X & a_2X & \cdots & a_nX\end{bmatrix}\in\mathbb{Z}^n[X]$.

For a given group $(\mathbb{G},\cdot)$, $T_{\mathbb{G}}$ denotes the binary complexity of computing group operations. For $h\in\mathbb{G}$, $\sqrt{\langle h^2\rangle}$ denotes the subgroup $\left\{g\in\mathbb{G}\colon \exists\alpha\in\mathbb{Z}, g^2=h^{2\alpha}\right\}$.

For $\mathbf{g}\in\mathbb{G}^n$, if n is even, set $\mathbf{g}_1:=\begin{bmatrix}g_1 & \cdots & g_{n/2}\end{bmatrix}$ and $\mathbf{g}_2:=\begin{bmatrix}g_{n/2+1} & \cdots & g_n\end{bmatrix}$, and if n is odd, set $\mathbf{g}_1:=\begin{bmatrix}g_1 & \cdots & g_{\lfloor n/2\rfloor} & 1_{\mathbb{G}}\end{bmatrix}$ and $\mathbf{g}_2:=\begin{bmatrix}g_{\lceil n/2\rceil} & \cdots & g_n\end{bmatrix}$. For $\mathbf{a}\in\mathbb{Z}^n$, if n is even, set $\mathbf{a}_1:=\begin{bmatrix}a_1 & \cdots & a_{n/2}\end{bmatrix}$ and $\mathbf{a}_2:=\begin{bmatrix}a_{n/2+1} & \cdots & a_n\end{bmatrix}$, and if n is odd, set $\mathbf{a}_1:=\begin{bmatrix}a_1 & \cdots & a_{\lfloor n/2\rfloor} & 0\end{bmatrix}$ and $\mathbf{a}_2:=\begin{bmatrix}a_{\lceil n/2\rceil} & \cdots & a_n\end{bmatrix}$.

For $n\in\mathbb{N}^*$, $z\in\mathbb{Z}$ and $\mathbf{g}=\begin{bmatrix}g_1 & \ldots & g_n\end{bmatrix}\in\mathbb{G}^n$, let $\mathbf{g}^z:=\begin{bmatrix}g_1^z & \cdots & g_n^z\end{bmatrix}\in\mathbb{G}^n$. For $\mathbf{a}=\begin{bmatrix}a_1 & \ldots & a_n\end{bmatrix}\in\mathbb{Z}^n$, define $\mathbf{g}^{\mathbf{a}}:=\prod_{i=1}^n g_i^{a_i}$. For $\mathbf{g}$ and $\mathbf{h}$ in $\mathbb{G}^n$, $\mathbf{g}\circ\mathbf{h}\in\mathbb{G}^n$ denotes their Hadamard product, i.e., their component-wise product.

2.2 Hidden-Order-Group Generators and Hardness Assumptions

A hidden-order-group generator G is an algorithm which takes as input a security parameter 1^λ and returns the description of a finite Abelian group $(\mathbb{G}, \cdot)$ and an integer $P \geq 2$. Integer P is assumed to be smaller than the order of $\mathbb{G}$, but to still be a super-polynomial function of the security parameter. The role of P is mainly to adjust the soundness of the protocols herein, as their challenge spaces will typically be $\llbracket 0; P^{\Omega(1)} - 1 \rrbracket$.

It is also assumed that given the description of $\mathbb{G}$, the group law and the inversion of group elements can be efficiently computed, that group elements can be sampled uniformly at random and that an upper bound $2^{b_\mathbb{G}}$ on $\operatorname{ord}(\mathbb{G})$ can be efficiently computed, with $b_\mathbb{G} \coloneqq b_\mathbb{G}(\lambda)$ polynomial in λ (it is further assumed that $b_\mathbb{G} = \Omega(\lambda)$). Recall that the bit complexity of an elementary operation in a group $\mathbb{G}$ is denoted $T_\mathbb{G}$.

The following assumptions are classical for hidden-order-group generators and were introduced by Damgård and Fujisaki [**?**]. They are best illustrated for P such that natural integers less than P are factorizable in polynomial time in λ (e.g., $\lambda^{\log^{\Omega(1)}(\lambda)}$ given current knowledge in computational number theory), and for $\mathbb{G}$ as the group $\mathbb{Z}_N^*$ for an RSA modulus N with prime factors p and q such that $p = q = 3 \mod 4$, $\gcd(p-1, q-1) = 2$ and the number of divisors of $p-1$ and $q-1$ with prime factors less than P is of magnitude $O(\lambda)$. However, these assumptions are believed to also hold over generators of ideal-class groups.

Definition 2.1 (Strong-Root Assumption). *A group generator* G *satisfies the* (T, ε)*-strong-root assumption if for all* $\lambda \in \mathbb{N}$*, for every adversary* $\mathcal{A}$ *that runs in time at most* $T(\lambda)$*,*

$$\Pr\left[g^n = h \wedge n > 1 \colon \begin{array}{c} (\mathbb{G}, P) \leftarrow \mathsf{G}\left(1^\lambda\right) \\ h \leftarrow_\$ \mathbb{G} \\ (g, n) \leftarrow \mathcal{A}(\mathbb{G}, P, h) \end{array} \right] \leq \varepsilon(\lambda).$$

This assumption is simply a generalization of the strong RSA assumption [**?**, **?**] to hidden-order groups.

Definition 2.2 (Small-Order Assumption). *A group generator* G *satisfies the* (T, ε)*-small-order assumption if for all* $\lambda \in \mathbb{N}$*, for every adversary* $\mathcal{A}$ *that runs in time at most* $T(\lambda)$*,*

$$\Pr\left[\begin{array}{c} g^n = 1_\mathbb{G} \wedge g^2 \neq 1_\mathbb{G} \\ 0 < n < P \end{array} : \begin{array}{c} (\mathbb{G}, P) \leftarrow \mathsf{G}\left(1^\lambda\right) \\ (g, n) \leftarrow \mathcal{A}(\mathbb{G}, P) \end{array} \right] \leq \varepsilon(\lambda).$$

The small-order assumption simply states that it should be hard to find low-order elements in the group (different from $1_\mathbb{G}$), except for square roots of unity which may be easy to compute (e.g., -1 in RSA groups). In the group $\mathbb{Z}_N^*$ for $N = pq$ with p and q prime such that $\gcd(p-1, q-1) = 2$, Damgård and Fujisaki [**?**] showed that factoring N can be reduced to this problem in polynomial time if integers less than P are factorizable in polynomial time in λ.

Definition 2.3 (Orders with Low Dyadic Valuation). *A group generator* G *satisfies the low-dyadic-valuation assumption on orders if for all* $\lambda \in \mathbb{N}$*, for every* $(\mathbb{G}, P) \leftarrow \mathsf{G}\left(1^{\lambda}\right)$*, for every* $g \in \mathbb{G}$*,* $\operatorname{ord}(g)$ *is divisible by* 2 *at most once.*

Notice that in the group $\mathbb{Z}_N^*$ for $N = pq$ with p and q prime such that $p = q = 3 \mod 4$, the order of any element is divisible by 2 at most once since 2 divides $p - 1$ and $q - 1$ exactly once.

Definition 2.4 (Many Rough-Order Elements or μ-Assumption). *An integer is said to be* P*-rough if all its prime factors are greater than or equal to* P*. A group generator* G *satisfies the* μ*-assumption that there are many rough-order elements in the groups generated by* G *(or simply the* μ*-assumption) if for all every parameter* $\lambda \in \mathbb{N}$*,*

$$\Pr\left[\operatorname{ord}(h) \textit{ is } P\textit{-rough} : \begin{array}{c}(\mathbb{G}, P) \leftarrow \mathsf{G}\left(1^{\lambda}\right) \\ h \leftarrow_{\$} \mathbb{G}\end{array}\right] \geq \mu(\lambda).$$

2.3 Non-interactive Commitments

A (non-interactive) commitment scheme consists of an algorithm $\mathsf{Setup}\left(1^{\lambda}\right) \rightarrow pp$ which generates public parameters (implicit inputs to the other algorithms); a key-generation algorithm $\mathsf{KG}\,(pp) \rightarrow ck$; a probabilistic algorithm $\mathsf{Com}\,(ck, x) \rightarrow (C, d)$ that computes a commitment C to a value x and an opening or decommitment information d on the input of ck; and a deterministic algorithm $\mathsf{ComVf}(ck, C, x, d) \rightarrow b \in \{0, 1\}$ which returns a bit indicating whether the decommitment d is valid (bit 1) for C and x w.r.t. key ck, or not (bit 0). Formal definitions of the correctness, hiding and binding properties of commitment schemes are given in the full version [**?**].

Discussion. The syntax above separates the commitment-key generation algorithm from the setup algorithm although these are often tacitly combined, especially for commitments in public-order groups. The main reason is that doing so allows to define the hiding property for schemes even when the keys are possibly *invalid.* This question does not arise for schemes with keys that are elements of a prime-order group $\mathbb{G} = \langle g \rangle$ (e.g., Pedersen's scheme [**?**]) since any element $h \in \mathbb{G}^*$ is a valid commitment key. However, when the scheme is defined over an unknown-order group $\mathbb{G}$ which may not be cyclic, and that keys are elements of the *subgroup* generated by an element (as it is the case for Damgård–Fujisaki commitments recalled in Section **??**), say h, there may not be an efficient way to test whether another element $g \in \mathbb{G}$ is in $\langle h \rangle$. Computing a commitment with an invalid key may then not guarantee that the commitment is hiding. That is why the scheme will be required to be hiding even if commitments are computed with a potentially invalid key.

2.4 Argument Systems

This section defines argument systems for families of languages. The languages are parametrized by public parameters and Common-Reference Strings (CRSs). As a simple example, given an Abelian group $\mathbb{G}$ (which could be non-cyclic) and an element $h \in \mathbb{G}$ (the parameters) and another element $g \in \langle h \rangle$ (the CRS), consider the language of group elements $C \in \mathbb{G}$ such that there exists $x, y \in \mathbb{Z}$ for which $C = g^x h^y$. This language is clearly parametrized by the parameters and the CRS, and one can give an argument system for this parametrized language in the same vein as what is subsequently done in the paper. However, to lighten the notation, arguments will be (abusively) referred to as arguments for languages rather than arguments for families of languages.

Formally, an argument system (or protocol) for a language $\mathcal{L} = \mathcal{L}_{pp,crs}$ (or equivalently, for the corresponding relation $\mathcal{R} = \mathcal{R}_{pp,crs}$) consists of a quadruple $\Pi = (\mathsf{Setup}, \mathsf{CRSGen},\ \mathsf{Prove}, \mathsf{Vf})$ such that $\mathsf{Setup}\left(1^\lambda\right) \rightarrow pp$ returns public parameters on the input of a security parameter, $\mathsf{CRSGen}(pp) \rightarrow crs$ returns a CRS, and $\langle \mathsf{Prove}(crs, x, w) \rightleftharpoons \mathsf{Vf}(crs, x) \rangle \rightarrow (\tau, b) \in \{0,1\}^* \times \{0,1\}$ are interactive algorithms (τ denotes the transcript of the interaction and b the decision bit of Vf). The public parameters are assumed to be tacit inputs to algorithms Prove and Vf, even though they may at times be made explicit for instantiated protocols, especially when the CRS is the empty string (in which case the CRS is omitted from the syntax). The definitions of the (culpable) soundness, extractability and honest-verifier zero-knowledge properties of argument systems are given in the full version [?].

The non-interactive argument system derived from an interactive one Π via the Fiat–Shamir heuristic [?] with a random oracle $\mathcal{H}$ is denoted $FS.\Pi^{\mathcal{H}}$.

3 Integer Commitments

This section recalls a scheme due to Damgård and Fujisaki which allows to commit to integers[7]. Then comes a new integer-commitment scheme with parameters smaller than those of Damgård and Fujisaki's scheme, and which are also more efficient to compute. For the version of our scheme which allows to commit to n integers, the parameters are of $O(b_\mathbb{G} + \log n)$ bits instead of $\Omega(n b_\mathbb{G} \log P)$ as with the generalized version of Damgård and Fujisaki's scheme, where $2^{b_\mathbb{G}}$ is an upper bound on the group order.

3.1 Damgård–Fujisaki Commitments

The Damgård–Fujisaki commitment scheme [?, ?], parameterized by a group generator G, consists of the following algorithms.

[7] Couteau, Peters and Pointcheval [?] proved that in the case of RSA groups (with Blum integers), the security of Damgård and Fujisaki's scheme is provable under (a variant of) the RSA assumption instead of the strong RSA assumption. This also holds for our scheme. However, this result does not concern generic hidden-order groups.

$\mathsf{Setup}\left(1^\lambda\right) \to pp$: run $(\mathbb{G}, P) \leftarrow \mathsf{G}\left(1^\lambda\right)$, generate $h \leftarrow_\$ \mathbb{G}$ and return $(\mathbb{G}, P, h)$. Recall that these parameters are implicit inputs to all the other algorithms.

$\mathsf{KG}(pp) \to ck$: generate $\alpha \leftarrow_\$ \left[\!\left[0; 2^{b_\mathbb{G}+\lambda}\right]\!\right]$ ($2^{b_\mathbb{G}}$ is an upper bound on $\mathrm{ord}(\mathbb{G})$), compute and return $g \leftarrow h^\alpha$.

$\mathsf{Com}(g, x \in \mathbb{Z}) \to (C, d)$: generate $r \leftarrow_\$ \left[\!\left[0; 2^{b_\mathbb{G}+\lambda}\right]\!\right]$, compute $C \leftarrow g^x h^r$, set $d \leftarrow (r, 1_\mathbb{G})$ and return (C, d).

$\mathsf{ComVf}\,(g, C, x, d) \to b \in \{0,1\}$: parse d as $(r, \tilde{g})$. If $C = g^x h^r \tilde{g}$ and $\tilde{g}^2 = 1_\mathbb{G}$, return 1, else return 0.

Equivalently, the commitment-algorithm could simply set the decommitment information d to r, and the commitment-verification would return 1 if the equality $C^2 = \left(g^x h^d\right)^2$ holds and 0 otherwise. The squaring in the verification is due to the fact that the small-order assumption does not exclude the possibility to efficiently compute square roots of unity, and they thus relaxed the verification equation to allow for sound argument of knowledge of openings to commitments. In other words, the scheme would still be binding without the squaring in the verification equation, and the relaxation is simply an artifact to allow for sound arguments.

More precisely, suppose that the verification were not relaxed, i.e., that it would only check that $C = g^x h^d$. Two accepting transcripts (D, e_1, z_1, t_1) and (D, e_2, z_2, t_2) of a standard Schnorr-type argument of knowledge of an opening would imply that $C^{e_1 - e_2} = g^{z_2 - z_1} h^{t_2 - t_1}$. Assuming $e_1, e_2 \in [\![0; P-1]\!]$, $e_1 \neq e_2$, and that $e_1 - e_2$ divides $z_2 - z_1$ and $t_2 - t_1$ (Damgård and Fujisaki showed that this latter event occurs with probability negligibly close to 1/2 under the assumptions on the group generator), the previous equality would imply that $\left(g^{(z_2-z_1)/(e_1-e_2)} h^{(t_2-t_1)/(e_1-e_2)} C^{-1}\right)^{e_1-e_2} = 1_\mathbb{G}$, and the small-order assumption would only allow to conclude that $C^2 = \left(g^{(z_2-z_1)/(e_1-e_2)} h^{(t_2-t_1)/(e_1-e_2)}\right)^2$. The trivial attack in which an adversary computes C as $g^x h^d \tilde{g}$ with $\tilde{g} \in \mathbb{G}$ such that $\tilde{g}^2 = 1_\mathbb{G}$ would then not be excluded by the protocol.

Properties. Damgård and Fujisaki's scheme is correct, is computationally binding under the strong-root assumption and the μ-assumption, and is statistically hiding. Its hiding property crucially relies on the fact that $g \in \langle h \rangle$. To guarantee the statistical hiding property of the scheme *without trusted key generation*, the party which computes g is then also required to compute a non-interactive proof that $g \in \langle h \rangle$. The commitment algorithm would then verify the proof and proceed as above if it is valid, and otherwise return $\perp$. Damgård and Fujisaki proposed to compute such a proof with a Schnorr-type protocol with $\{0,1\}$ as challenge set. To attain a soundness error of at most $1/P$, the proof must then be repeated at least $\lceil \log P \rceil$ times. With the Fiat–Shamir heuristic, each proof consists of (c, z), and the total proof in the public parameters then consists of $\lceil \log P \rceil \left(b_\mathbb{G} + 2\lambda + 2\right) = \Omega\left(b_\mathbb{G} \log P\right)$ bits (recall that P is super-polynomial in λ, e.g., $\lambda^{\log \lambda}$).

3.2 A new Integer-Commitment Scheme

This section introduces a novel integer-commitment scheme that is close to Damgård and Fujisaki's scheme, but with an argument (rather than a proof) of only $O(b_{\mathbb{G}})$ (with b such that $\operatorname{ord}(\mathbb{G}) \leq 2^{b_{\mathbb{G}}}$) bits in non-trusted keys, and the argument only requires a single protocol run to reach the same soundness error. As the soundness of the protocol relies on computational assumptions on the group generator, the scheme is only computationally hiding, whereas Damgård and Fujisaki's cut-and-choose protocol is perfectly sound (the prover is not assumed to be computationally bounded) but inefficient.

Formally, let G be a group generator and let $FS.\Pi^{\mathcal{H}}$ be a Fiat–Shamir non-interactive argument system with random oracle $\mathcal{H}$ for the language $\{g \in \mathbb{G}, \ell \in \mathbb{N}^* : \exists \alpha \in \left[\!\left[0; 2^{\ell}\right]\!\right], g = h^{\alpha}\right\}$, given parameters $(\mathbb{G}, P, h, 1)$ (integer 1 is just to indicate that there is only one group element g in the word for which the proof is computed) and the empty string as CRS. The proof of the hiding property will require the protocol to satisfy culpable soundness w.r.t. the language $\sqrt{\langle h^2 \rangle}$. The scheme, parameterized by G and further denoted $\mathscr{C}$, consists of the following algorithms:

$\mathsf{Setup}\left(1^{\lambda}\right) \to pp$: run $(\mathbb{G}, P) \leftarrow \mathsf{G}\left(1^{\lambda}\right)$, generate $h \leftarrow_{\$} \mathbb{G}$ and return $(\mathbb{G}, P, h)$.
Recall that these parameters are implicit inputs to all the other algorithms.

$\mathsf{KG}(pp) \to ck$: generate $\alpha \leftarrow_{\$} \left[\!\left[0; 2^{b_{\mathbb{G}}+\lambda}\right]\!\right]$, compute $g \leftarrow h^{\alpha}$ and a proof $\pi \leftarrow FS.\Pi^{\mathcal{H}}.\mathsf{Prove}((\mathbb{G}, P, h, 1), (g, b_{\mathbb{G}} + \lambda), \alpha)$, and return (g, π).

$\mathsf{Com}\left((g, \pi), x \in \mathbb{Z}\right) \to (C, d)$: if $FS.\Pi^{\mathcal{H}}.\mathsf{Vf}((\mathbb{G}, P, h, 1), (g, b_{\mathbb{G}} + \lambda), \pi) = 0$, then return $\perp$; else generate $r \leftarrow_{\$} \left[\!\left[0; 2^{b_{\mathbb{G}}+\lambda}\right]\!\right]$, compute $C \leftarrow (g^x h^r)^2$, set $d \leftarrow r$ and return (C, d).

$\mathsf{ComVf}\left((g, \pi), C, x, d\right) \to b \in \{0, 1\}$: if $C^2 = \left(g^x h^d\right)^4$ return 1, else return 0.

See the full version [?] for the proofs of correctness and security of the scheme.

Comparison with Damgård–Fujisaki Commitments. As for Damgård and Fujisaki's commitments, the squaring in the verification equation (compared to the computation of commitments) is again to later allow for sound arguments of knowledge of openings. The main difference compared to Damgård and Fujisaki's commitments is that commitments are computed as $(g^x h^r)^2$ instead of $g^x h^r$. It is simply due to the fact that π only guarantees that $g^2 \in \left\langle h^2 \right\rangle$, not that $g \in \langle h \rangle$, hence the power 2 in the computation of commitments to ascertain that they are hiding. However, only requiring that $g^2 \in \left\langle h^2 \right\rangle$ instead of $g \in \langle h \rangle$ is precisely what allows to have much smaller arguments that can be computed in a single protocol run.

Argument System FS.$\Pi^{\mathcal{H}}$. It only remains to provide a protocol $FS.\Pi^{\mathcal{H}}$ to argue knowledge of an integer $\alpha \in \mathbb{Z}$ such that $g^2 = h^{2\alpha}$, which is sufficient for the commitment scheme to be computationally hiding. We first give an interactive protocol Π for the language $\left\{g \in \mathbb{G}, \ell \in \mathbb{N}^* : \exists \alpha \in \left[\!\left[0; 2^{\ell}\right]\!\right], g = h^{\alpha}\right\}$ given parameters $(\mathbb{G}, P) \leftarrow \mathsf{G}\left(1^\lambda\right)$ and that satisfies culpable soundness w.r.t. $\sqrt{\langle h^2 \rangle}$, and then apply the Fiat–Shamir heuristic to obtain $FS.\Pi^{\mathcal{H}}$.

In more detail, the (interactive) protocol Π is as follows: the prover generates $k \leftarrow_\$ \left[\!\left[0; 2^{\ell+\lambda} P\right]\!\right]$, computes $t \leftarrow h^k$ and sends t to the verifier; the verifier chooses $c \leftarrow_\$ [\![0; P-1]\!]$ and sends it to the prover; the prover then replies with $r \leftarrow k - c\alpha$, and the verifier accepts if and only if $h^r g^c = t$. With the Fiat–Shamir heuristic, the proof consists of (c, r), i.e., $2\lfloor \log P \rfloor + \ell + \lambda + 3$ bits. For $\ell = b_\mathbb{G} + \lambda$, that is $2\lfloor \log P \rfloor + b_\mathbb{G} + 2\lambda + 3 = O(b_\mathbb{G})$ bits (recall that $P \leq 2^{b_\mathbb{G}}$ and $b_\mathbb{G} = \Omega(\lambda)$).

The completeness, statistical honest-verifier zero-knowledge and extractability properties of this protocol are proved in the full version [?].

Arguing Knowledge of Openings. As for Damgård and Fujisaki's commitments, one can efficiently argue knowledge of openings, i.e., of integers x and r such that a given commitment C satisfies $C^2 = (g^x h^r)^4$.

The protocol imposes an upper bound of ℓ on the bit length of the witness, with ℓ being part of the (public) word. It is simply to adapt the randomness range of the prover (and of the honest-verifier zero-knowledge simulator) to ensure that the protocol remains statistically honest-verifier zero-knowledge; and ℓ can be arbitrarily large. The protocol does *not* guarantee that the largest absolute value in the extracted witness is at most ℓ bits long [8]. In technical terms, the protocol is for the relation $\left\{\left(C \in \mathbb{G}, \ell \in \mathbb{N}^*; x, r \in \left[\!\left[0; 2^\ell\right]\!\right]\right) : C^2 = (g^x h^r)^4\right\}$ that satisfies culpable extractability for the relation $\Sigma := \left\{(C \in \mathbb{G}, \ell \in \mathbb{N}^*; x, r \in \mathbb{Z}) : C^2 = (g^x h^r)^4\right\}$.

More precisely, consider the problem of arguing in zero-knowledge knowledge of integers x and r such that $C^2 = (g^x h^r)^4$ and $|x|, |r| \leq 2^\ell$, for a group element C chosen by the prover and public bases h and g, and a public proof π that $g \in \sqrt{\langle h^2 \rangle}$. The prover first verifies π and aborts if it is invalid. The prover generates $y, s \leftarrow_\$ \left[\!\left[0; P 2^{\ell+\lambda}\right]\!\right]$, computes and sends $D \leftarrow (g^y h^s)^2$ to the verifier. The verifier then chooses $e \leftarrow_\$ [\![0; P-1]\!]$, sends it to the prover, and this latter replies with $z \leftarrow y - ex$ and $t \leftarrow s - er$ (computed in $\mathbb{Z}$). The verifier then accepts if and only if $(g^z h^t)^2 C^e = D$.

The properties of this protocol are proved in the full version [?].

Multi-Integer Commitments. The above commitments can be generalized to vectors of integers just like Damgård–Fujisaki commitments (as Couteau, Peters and Pointcheval did [?]). That is to say, the scheme can be extended to commit to several integers at once.

Formally, let G be a group generator and suppose that there exists a non-interactive argument system $FS.\Pi^{\mathcal{H}}$ with random oracle $\mathcal{H}$ for the language $\left\{g_1, \ldots, g_n \in \mathbb{G}, \ell \in \mathbb{N}^* : \exists \alpha_1, \ldots, \alpha_n \in \left[\!\left[0; 2^\ell\right]\!\right], \forall i \in [\![n]\!]\ g_i = h^{\alpha_i}\right\}$ given parameters $(\mathbb{G}, P, h, n)$ and the empty string as CRS.

$\mathsf{Setup}\left(1^\lambda, n\right) \rightarrow pp :$ run $(\mathbb{G}, P) \leftarrow \mathsf{G}\left(1^\lambda\right)$, generate $h \leftarrow_\$ \mathbb{G}$ and return $(\mathbb{G}, P, h, n)$.

[8] To prove such statements using hidden-order groups, Lipmaa's range argument [?], corrected by Couteau, Peters and Pointcheval [?], is suitable.

$\mathsf{KG}(pp) \to ck$: generate $\alpha_i \leftarrow_\$ \left[\!\left[0; 2^{b_\mathbb{G}+\lambda}\right]\!\right]$ for $i \in [\![n]\!]$, compute $g_i \leftarrow h^{\alpha_i}$ and $\pi \leftarrow FS.\Pi^{\mathcal{H}}.\mathsf{Prove}\left((\mathbb{G}, P, h, n), (\mathbf{g}, b_\mathbb{G} + \lambda), (\alpha_i)_{i=1}^n\right)$, and return $(\mathbf{g}, \pi)$.

$\mathsf{Com}\left((\mathbf{g}, \pi), x_1, \ldots, x_n \in \mathbb{Z}\right) \to (C, d)$: if $FS.\Pi^{\mathcal{H}}.\mathsf{Vf}((\mathbb{G}, P, h, n), (\mathbf{g}, b_\mathbb{G} + \lambda), \pi) = 0$ return $\perp$; generate $r \leftarrow_\$ \left[\!\left[0; 2^{b_\mathbb{G}+\lambda}\right]\!\right]$, compute $C \leftarrow \left(\prod_{i=1}^n g_i^{x_i} h^r\right)^2$, set $d \leftarrow r$ and return (C, d).

$\mathsf{ComVf}\left((\mathbf{g}, \pi), C, x_1, \ldots, x_n, d\right) \to b \in \{0, 1\}$: if $C^2 = \left(\prod_i g_i^{x_i} h^d\right)^4$ return 1, else return 0.

The only missing component is an interactive protocol Π that satisfies culpable soundness w.r.t. $\left\{g_1, \ldots, g_n \in \mathbb{G} \colon \exists \alpha_1, \ldots, \alpha_n \in \mathbb{Z}, \forall i \in [\![n]\!]\ g_i^2 = h^{2\alpha_i}\right\}$. A possible solution is to run n times in parallel the protocol from the case $n = 1$ for each of the α_i values. However, they achieve an overall $2^{-\lambda}$ statistical distance from n simulated arguments, the range of the prover's randomness in the protocol must be multiplied by n so that each argument is $2^{-\lambda}n^{-1}$-zero-knowledge. A better solution is to use the protocol presented in the full version [**?**, Section 5.3], which results in arguments of $O(b_\mathbb{G} + \log n)$ bits. This should be compared to the $\Omega(nb_\mathbb{G} \log P)$-bit parameters of the generalized Damgård–Fujisaki commitments.

4 Succinct Inner-Product Arguments on Integers

This section gives a statistically honest-verifier zero-knowledge, logarithmic-size inner-product argument on integers committed in hidden-order groups with the scheme from Section **??**. That is, an argument of knowledge of vectors $\mathbf{a}$ and $\mathbf{b} \in \mathbb{Z}^n$, and of a randomness $r \in \mathbb{Z}$ such that $C^2 = \left(\mathbf{g}^\mathbf{a}\mathbf{h}^\mathbf{b} f^r\right)^4$ and $\langle \mathbf{a}, \mathbf{b} \rangle = z$ given public bases $\mathbf{g}$ and $\mathbf{h}$, a public commitment C and a public integer z; and the bit-communication complexity of the protocol is logarithmic in of order $O(\ell + \log nb_\mathbb{G})$ where ℓ is an upper-bound on the bit length of the largest integer witness and $2^{b_\mathbb{G}}$ an upper-bound on the order of the group.

4.1 Formal Description

This section formalizes the protocol and states the properties it satisfies.

Relations. The protocol is an honest-verifier zero-knowledge argument for

$$\mathcal{R} := \Big\{(C \in \mathbb{G}, z \in \mathbb{Z}, \ell \in \mathbb{N}^*; \mathbf{a}, \mathbf{b} \in \mathbb{Z}^n, r \in \mathbb{Z}) : C^2 = \left(\mathbf{g}^\mathbf{a}\mathbf{h}^\mathbf{b} f^r\right)^4 \wedge \langle \mathbf{a}, \mathbf{b} \rangle = z$$
$$\wedge \left\| \left[\mathbf{a}\ \mathbf{b}\ r\right] \right\|_\infty < 2^\ell \Big\}$$

given parameters $(\mathbb{G}, P, f, n)$ with $f \in \mathbb{G}$ and $n \in \mathbb{N}^*$, and $(\mathbf{g}, \mathbf{h}, \pi_{crs}) \in \mathbb{G}^{2n} \times \{0, 1\}^*$ as CRS.

The relation imposes the largest value (in absolute value) in the witness $\left[\mathbf{a}\ \mathbf{b}\ r\right]$ to be at most ℓ bits long, with ℓ being part of the (public) word. As for the argument of knowledge of openings in Section **??**, it is again to adapt the randomness range of the prover and of the honest-verifier zero-knowledge simulator to make sure that the protocol remains statistically honest-verifier zero-knowledge; and ℓ can be arbitrarily large. However, the protocol does not necessarily return a witness with integers of at most ℓ bits in absolute value. In other words, the protocol satisfies culpable extractability w.r.t. the relation

$$\Sigma := \left\{(C \in \mathbb{G}, z \in \mathbb{Z}, \ell \in \mathbb{N}^*; \mathbf{a}, \mathbf{b} \in \mathbb{Z}^n, r \in \mathbb{Z}) : C^2 = \left(\mathbf{g}^{\mathbf{a}}\mathbf{h}^{\mathbf{b}} f^r\right)^4 \wedge \langle \mathbf{a}, \mathbf{b} \rangle = z\right\}.$$

The argument for $\mathcal{R}$ is actually reduced to a logarithm-size argument (given on Figure **??**) for the following relation in which the inner product is also committed:

$$\mathcal{R}' := \left\{(C \in \mathbb{G}, \ell \in \mathbb{N}^*; \mathbf{a}, \mathbf{b} \in \mathbb{Z}^n, r \in \mathbb{Z}) : C^2 = \left(\mathbf{g}^{\mathbf{a}}\mathbf{h}^{\mathbf{b}} e^{\langle \mathbf{a}, \mathbf{b} \rangle} f^r\right)^4 \wedge \left\| \left[\mathbf{a}\ \mathbf{b}\ r\right] \right\|_\infty < 2^\ell\right\}$$

given parameters $(\mathbb{G}, P, f, n)$ with $f \in \mathbb{G}$ and $n \in \mathbb{N}^*$, and $(\mathbf{g}, \mathbf{h}, e, \pi_{crs}) \in \mathbb{G}^{2n+1} \times \{0,1\}^*$ as CRS. Again, the protocol does not guarantee that the extracted witness satisfies the bounds on its bit length – denote by Σ' the relation defined as $\mathcal{R}'$ without the restriction on the size of the witness.

During the reduction, the verifier chooses a base $e \in \langle f \rangle$ and proves to the prover that e is in $\sqrt{\langle f^2 \rangle}$, which guarantees to the prover that the commitment Ce^{2z} remains hiding. (As explained in Section **??**, this precaution is not needed in groups of public prime orders.) However, since the protocol in Section **??** is only honest-verifier, and the extractability of the argument system partly relies on the fact that the prover does not know a discrete-logarithm relation between e and f, the verifier must compute a non-interactive argument with a random oracle. In other words, the extractability of the argument relies on the zero-knowledge property of the protocol in Section **??**. Moreover, the CRS of the protocol includes a proof that $\mathbf{g}$ and $\mathbf{h}$ are in $\sqrt{\langle f^2 \rangle}^n$, and the argument is only guaranteed to be honest-verifier zero-knowledge if it is indeed the case; that is, the zero-knowledge property of the argument relies on the soundness of the protocol. This mirroring in the properties of two protocols is simply due to the fact that at the beginning of the inner-product argument, the prover becomes the verifier of the protocol for $\mathbf{g}, \mathbf{h} \in \sqrt{\langle f^2 \rangle}^n$.

Main Insights. The goal is to have a protocol for $\mathcal{R}'$ in which the prover sends only $2\lceil \log n \rceil + 2$ group elements and three integers of at most $O(\ell + b_{\mathbb{G}} + \log(n)\log(P))$ bits. The main idea is to have the prover first send a constant number of commitments that depend on the witness vectors (which are in $\mathbb{Z}^n$), so that the verifier can thereafter choose *integer* linear combinations (defined by an integer x) of the witness vectors that are of length $n/2$ (to ease the explanation, further assume n to be a power of 2 in this section). These new vectors then serve as witness for a new commitment derived from the original commitment on which the proof is computed, the commitments sent by the prover and x; in bases of length $n/2$ and determined by the original bases and x. The prover and the verifier can thus recursively run the protocol with vectors of length $n/2$. After $\log n$ recursive calls, the vectors are of length 1, and the parties run a protocol that two committed integers a and b satisfy $ab = z$ for a public z.

In more detail, given $\mathbf{a}, \mathbf{b} \in \mathbb{Z}^n$ and $r \in \mathbb{Z}$ such that $C^2 = \left(\mathbf{g}^{\mathbf{a}}\mathbf{h}^{\mathbf{b}}e^{\langle \mathbf{a},\mathbf{b}\rangle}f^r\right)^4$, the prover first sends commitments $U \leftarrow \left({\mathbf{g}_1}^{\mathbf{a}_2}{\mathbf{h}_2}^{\mathbf{b}_1}e^{\langle \mathbf{a}_2,\mathbf{b}_1\rangle}f^{s_u}\right)^2$ and $V \leftarrow \left({\mathbf{g}_2}^{\mathbf{a}_1}{\mathbf{h}_1}^{\mathbf{b}_2} e^{\langle \mathbf{a}_1,\mathbf{b}_2\rangle}f^{s_v}\right)^2$, for s_u and s_v with uniform distribution over an integer set large enough for the commitments to be hiding. The verifier chooses $x \leftarrow_{\$} [\![0; P-1]\!]$, sends it to the prover, and this latter computes $\mathbf{a}' \leftarrow \mathbf{a}_1 + x\mathbf{a}_2$, $\mathbf{b}' \leftarrow x\mathbf{b}_1 + \mathbf{b}_2$ and $t \leftarrow s_v + rx + s_u x^2$. Note that all these operations are performed in $\mathbb{Z}$ and do not require to invert any integer. Now note that

$$\left((\mathbf{g}_1^x \circ \mathbf{g}_2)^{\mathbf{a}'}\,(\mathbf{h}_1 \circ \mathbf{h}_2^x)^{\mathbf{b}'}\,e^{\langle \mathbf{a}',\mathbf{b}'\rangle}f^t\right)^4 = \left(U^{x^2}C^xV\right)^2,$$

which means that the prover and verifier can run the protocol again with $\mathbf{g}_1^x \circ \mathbf{g}_2$ and $\mathbf{h}_1 \circ \mathbf{h}_2^x$ as bases and $\mathbf{a}'$ and $\mathbf{b}'$ (all of size $n/2$ instead of n) as witness for $U^{x^2}C^xV$.

To understand how a witness consisting of integer vectors can be extracted, suppose that one can obtain three transcripts $\left(U, V, x_j, \mathbf{a}'_j, \mathbf{b}'_j, t'_j\right)_{i=1}^3$ such that

$$\left(({\mathbf{g}_1}^{x_j} \circ \mathbf{g}_2)^{\mathbf{a}'_j}\,(\mathbf{h}_1 \circ {\mathbf{h}_2}^{x_j})^{\mathbf{b}'_j}\,e^{\left\langle \mathbf{a}'_j,\mathbf{b}'_j\right\rangle}f^{t_j}\right)^4 = \left(U^{x_j^2}C^{x_j}V\right)^2$$

for all $j \in [\![3]\!]$. The goal is to find a representation of C in the bases $\mathbf{g}$, $\mathbf{h}$, e and f. To do so, consider the linear system:

$$\mathbf{X}\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \text{ for } \mathbf{X} \coloneqq \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ x_1^2 & x_2^2 & x_3^2 \end{bmatrix} \text{ and indeterminate } \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}.$$

It does not necessarily have a solution in $\mathbb{Z}^3$ (and this is the first major difference with Bulletproofs in groups with public prime orders). However, denoting by $\operatorname{adj}(\mathbf{X})$ the adjugate matrix of $\mathbf{X}$ (which is in $\mathbb{Z}^{3\times 3}$), the column vector

$$v_C \coloneqq \operatorname{adj}(\mathbf{X})\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \text{ satisfies } \mathbf{X}v_C = \mathbf{X}\operatorname{adj}(\mathbf{X})\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \det(\mathbf{X}) \\ 0 \end{bmatrix}$$

since $\mathbf{X}\operatorname{adj}(\mathbf{X}) = \det(\mathbf{X})\mathbf{I}_3$. Therefore, via linear combinations with coefficient determined by v_C, one can obtain $\mathbf{a}_C, \mathbf{b}_C \in \mathbb{Z}^n$ and $z_C, r_C \in \mathbb{Z}$ such that $U^{2\det \mathbf{X}} = \left(\mathbf{g}^{\mathbf{a}_C}\mathbf{h}^{\mathbf{b}_C}e^{z_C}f^{r_C}\right)^4$. If the challenges x_1, x_2, x_3 are pairwise distinct, then $\det \mathbf{X} \neq 0$, and Lemma **??** shows that under the assumptions on the group generator, $2\det \mathbf{X}$ must divide (with overwhelming probability) $4z_C$, $4r_C$ and each of the components of $4\mathbf{a}_C$, $4\mathbf{b}_C$. Therefore, up to a relabeling of $2\mathbf{a}_C/\det \mathbf{X}$ and so on, one can extract $\mathbf{a}_C, \mathbf{b}_C \in \mathbb{Z}^n$ and $z_C, r_C \in \mathbb{Z}$ such that $U = \left(\mathbf{g}^{\mathbf{a}_C}\mathbf{h}^{\mathbf{b}_C}e^{z_C}f^{r_C}\right)^2\tilde{g}_C$ for $\tilde{g}_C \in \mathbb{G}$ that satisfies $\tilde{g}_C^2 = 1_{\mathbb{G}}$.

Nonetheless, it is not yet certain that $z_C = \langle \mathbf{a}_C, \mathbf{b}_C\rangle$. To guarantee it, it suffices to extract similar representations for U and V, and replacing U, C and V by those representations in the equality $\left(\left(\mathbf{g}_1^x \circ \mathbf{g}_2\right)^{\mathbf{a}'}\left(\mathbf{h}_1 \circ \mathbf{h}_2^x\right)^{\mathbf{b}'}e^{\langle \mathbf{a}',\mathbf{b}'\rangle}f^t\right)^4 =$

$\left(U^{x^2}C^xV\right)^2$ for any $x \in \{x_1, x_2, x_3\}$. This leads to a discrete-logarithm relation $1_{\mathbb{G}} = \mathbf{g}_1{}^{p_{\mathbf{g}_1}(x)}\mathbf{g}_2{}^{p_{\mathbf{g}_2}(x)}\mathbf{h}_1{}^{p_{\mathbf{h}_1}(x)}\mathbf{h}_2{}^{p_{\mathbf{h}_2}(x)}e^{p_e(x)}f^{p_f(x)}$ with $p_{\mathbf{g}_1}, p_{\mathbf{g}_2}, p_{\mathbf{h}_1}, p_{\mathbf{h}_2}, p_e, p_f$ polynomials in $\mathbb{Z}[x]$ of degree at most 2. Lemma **??** essentially states that it is hard to find discrete-logarithm relations in the subgroup generated by a group element $f \leftarrow_\$ \mathbb{G}$ (this is the second main difference with Bulletproofs in groups with public prime orders). It thus implies that if the bases are all in $\langle f \rangle$ with exponents chosen uniformly at random over a large integer set, these polynomials must all be zero (with overwhelming probability) when evaluated at x; and $p_{\mathbf{g}_1}, p_{\mathbf{h}_2}$ and p_e together lead to an integer polynomial of degree 4, with leading coefficient $z_C - \langle \mathbf{a}_c, \mathbf{b}_C \rangle$, which must then be nil when evaluated at x. Therefore, starting with five accepting transcripts instead of three entails that this polynomial of degree 4 must be nil and thus $z_C = \langle \mathbf{a}_c, \mathbf{b}_C \rangle$, i.e., $\mathbf{a}_C, \mathbf{b}_C \in \mathbb{Z}^n, r_C \in \mathbb{Z}$ is a valid witness for C.

As for the zero-knowledge property of the scheme, the ranges of s_u and s_v at each of the $\log n$ recursion step are chosen so that the statistical distance of (U, V) to a pair of uniform values in $\left\langle f^2 \right\rangle$ is at most $\left(\log(n)2^\lambda\right)^{-1}$. It then remains to compute an upper-bound on the bit length of the witness at the last step of the protocol so that the randomness of the prover can be chosen from a set of which the bit length is λ times larger. The calculation is detailed in the proof of the zero-knowledge property presented in the full version [**?**].

Protocol Algorithms. The argument system for relation $\mathcal{R}$ is further denoted Π. It uses as building blocks a group generator G and the Fiat–Shamir non-interactive variant $FS.\tilde{\Pi}^{\mathcal{H}}$ with a random oracle $\mathcal{H}$ of a protocol $\tilde{\Pi}$ for the language $\left\{(\mathbf{g}, \mathbf{h}) \in \mathbb{G}^{2n} \colon \exists \alpha, \beta \in \mathbb{Z}^{2n}, \forall i \in [\![n]\!]\; g_i = f^{\alpha_i} \wedge h_i = f^{\beta_i}\right\}$ given parameters $(\mathbb{G}, P, f, 2n)$ and the empty string as CRS. Protocol $\tilde{\Pi}^{\mathcal{H}}$ is later assumed to satisfy culpable soundness w.r.t. the language $\left\{(\mathbf{g}, \mathbf{h}) \in \mathbb{G}^{2n} \colon \exists \alpha, \beta \in \mathbb{Z}^{2n}, \forall i \in [\![n]\!]\; g_i^2 = f^{2\alpha_i} \wedge h_i^2 = f^{2\beta_i}\right\}$. The protocol algorithms are then as follows:

- $\Pi.\mathsf{Setup}\left(1^\lambda, n \in \mathbb{N}^*\right)$ runs $(\mathbb{G}, P') \leftarrow \mathsf{G}\left(1^\lambda\right)$, computes $P := \left\lfloor P'^{1/3} \right\rfloor$ (the power 1/3 is to ensure extractability under the assumptions on the group generator), generates $f \leftarrow_\$ \mathbb{G}$ and returns $pp \leftarrow (\mathbb{G}, P, n, f)$ as public parameters.
- $\Pi.\mathsf{CRSGen}(pp)$ generates $\alpha_i, \beta_i \leftarrow_\$ \left[\!\left[0; 2^{b_{\mathbb{G}}+2\lambda}\right]\!\right]$ for $i \in [\![n]\!]$, computes $g_i \leftarrow f^{\alpha_i}$, $h_i \leftarrow f^{\beta_i}$ and $\pi_{crs} \leftarrow FS.\tilde{\Pi}^{\mathcal{H}}.\mathsf{Prove}\left((\mathbb{G}, P, f, 2n), (\mathbf{g}, \mathbf{h}), \alpha, \beta\right)$, and returns $(\mathbf{g}, \mathbf{h}, \pi_{crs})$.
- $\Pi.\mathsf{Prove}$ and $\Pi.\mathsf{Vf}$ are as on Figure **??**. They run as sub-routines the proving and verification algorithms of a protocol Π' for relation $\mathcal{R}'$. Algorithms $\Pi'.\mathsf{Prove}$ and $\Pi'.\mathsf{Vf}$ additionally take as input a variable i which keeps track of the recursion depth during the protocol execution to adjust the randomness of the prover.

Prover-Communication Complexity. Throughout the protocol, the prover sends $2n'+2$ group elements (with $n' = \lceil \log n \rceil$), two integers ($a'$ and b') less than

$\mathcal{P}\ (n, f, \mathbf{g}, \mathbf{h}, \pi_{crs}, C, z, \ell; \mathbf{a}, \mathbf{b}, r)$		$\mathcal{V}\ (n, f, \mathbf{g}, \mathbf{h}, \pi_{crs}, C, z, \ell)$

$$C^2 = \left(\mathbf{g}^{\mathbf{a}}\mathbf{h}^{\mathbf{b}} f^r\right)^4 \wedge \langle \mathbf{a}, \mathbf{b} \rangle = z \wedge \left\| \left[\mathbf{a}\ \mathbf{b}\ r\right] \right\|_{\infty} < 2^{\ell}$$

```
if $FS.\tilde{\Pi}^{\mathcal{H}}.\mathsf{Vf}\left((\mathbb{G}, P, f, 2n), (\mathbf{g}, \mathbf{h}), \pi_{crs}\right) = 0$        $\alpha \leftarrow_{\$} [\![0; 2^{b+2\lambda}]\!]$; $e \leftarrow f^{\alpha}$
  then return $\perp$                    $\pi \leftarrow FS.\tilde{\Pi}^{\mathcal{H}}.\mathsf{Prove}((\mathbb{G}, P, f, 1), e, \alpha)$
                              $\xleftarrow{e,\pi}$
if $FS.\tilde{\Pi}^{\mathcal{H}}.\mathsf{Vf}((\mathbb{G}, P, f, 1), e, \pi) = 0$
  then return $\perp$
    run the protocol on Figure ?? on input $\left(1, n, f, \mathbf{g}, \mathbf{h}, e, C_1 := Ce^{2z}, \ell; \mathbf{a}, \mathbf{b}, r\right)$
```

Fig. 1. Inner-Product Argument on Integers.

$2^{\ell} P^{n'}$ in absolute value and an integer (u) less than $\left(2n' 2^{b_{\mathbb{G}}+\lambda} P^{n'+3} + 2^{\ell}(P-1)^{n'+2}\right)$ $\left(1 + 2^{\lambda}\right)$ in absolute value. The bit communication complexity of the prover is then of order $O\left(\ell + \log(n)(b_{\mathbb{G}} + \log P) + \lambda + \max\left(\log\log n + b_{\mathbb{G}} + \lambda, \ell\right)\right)$. Since $\log P \leq b_{\mathbb{G}} = \Omega(\lambda)$, that is $O\left(\ell + \log(n) b_{\mathbb{G}} + \max\left(\log\log n + b_{\mathbb{G}}, \ell\right)\right)$, or even $O\left(\ell + \log(n) b_{\mathbb{G}}\right)$ bits (n is here assumed to be greater than 1).

Verification via a Single Multi-Exponentiation. As described on Figure **??**, the verifier computes a new commitment $U_i^{x_i^2} C_i^{x_i} V_i$, and new vectors $\mathbf{g}_1^{x_i} \circ \mathbf{g}_2$ and $\mathbf{h}_1 \circ \mathbf{h}_2^{x_i}$ at each recursion step i. In total, the verifier then has to compute $n' := \lceil \log n \rceil$ 3-exponentiations with exponents less than P^2 and two $\left\lceil n 2^{-i} \right\rceil$-exponentiations with exponents less than P for $i = 0, \ldots, n'-1$. At the last stage of the protocol, the verifier also has to check that $\left(g^{x_{n'+1} a'} h^{x_{n'+1} b'} e^{a'b'} f^u\right)^4 = \left(C_{n'+1}^{x_{n'+1}^2} \Gamma^{x_{n'+1}} \Delta\right)^2$, i.e., a 7-exponentiation with exponents (in absolute value) less than the bit length of the largest exponent.

Alternatively, the verifier could simply generate the challenges after receiving the U_i and V_i values, delay its verification to the last stage of the protocol and then do a single multi-exponentiation. As shown below, this multi-exponentiation is a $(2n + 2n' + 5)$-exponentiation, which results in computational savings in practice since computing a k-exponentiation with ℓ-bit exponents requires ℓ group operations with a pre-computed table of 2^k group elements following classical sliding-window methods [?], which is much faster than computing k separate single exponentiations with ℓ-bit exponents (which requires $k\ell$ group operations with a single group element in memory) and multiplying the result[9].

[9] If n is large, then the pre-computation might be prohibitively long with the standard multi-exponentiation method, in which case one would rather split the multi-exponentiation in small batches. In any case, delaying the verification until the last step already has the benefit of eliminating latency in the verification.

```
$\mathcal{P}\ (i, n, f, \mathbf{g}, \mathbf{h}, e, C_i, \ell; \mathbf{a}, \mathbf{b}, r)$                                   $\mathcal{V}\ (i, n, f, \mathbf{g}, \mathbf{h}, e, C_i, \ell)$

$C_i^2 = \left(\mathbf{g}^{\mathbf{a}}\mathbf{h}^{\mathbf{b}}e^{\langle\mathbf{a},\mathbf{b}\rangle}f^r\right)^4 \wedge \left\|\left[\mathbf{a}\ \mathbf{b}\ r\right]\right\|_\infty < 2^\ell$
----------------------------------------------------------------------------------------------
if $n = 1$
$\alpha, \beta \leftarrow_\$ \left[\!\left[0; 2^{\ell+\lambda}P^i\right]\!\right]$
$s \leftarrow_\$ \left[\!\left[0; 2(i-1)2^{b_{\mathbb{G}}+\lambda}\right]\!\right]$
$t \leftarrow_\$ \left[\!\left[0; 2(i-1)2^{b_{\mathbb{G}}+2\lambda}P^{i+2} + 2^{\ell+\lambda}(P-1)^{i+1}\right]\!\right]$
Replace $i-1$ by 1 if $i = 1$
$\Gamma \leftarrow \left(g^\alpha h^\beta e^{\alpha b + a\beta} f^s\right)^2$
$\Delta \leftarrow \left(e^{\alpha\beta} f^t\right)^2$
                              $\xrightarrow{\Gamma,\Delta}$
                                                                    $x_i \leftarrow_\$ [\![0; P-1]\!]$
                              $\xleftarrow{x_i}$
$a' \leftarrow \alpha + a x_i$
$b' \leftarrow \beta + b x_i$
$u \leftarrow t + s x_i + r x_i^2$
                              $\xrightarrow{a',b',u}$
                                      $\left(g^{x_i a'} h^{x_i b'} e^{a'b'} f^u\right)^4 \overset{?}{=} \left(C_i^{x_i}\Gamma^{x_i}\Delta\right)^2$

else
$s_u, s_v \leftarrow_\$ \left[\!\left[0; 2\left(\lceil\log n\rceil + i - 1\right)2^{b_{\mathbb{G}}+\lambda}\right]\!\right]$
$U_i \leftarrow \left(\mathbf{g}_1{}^{\mathbf{a}_2}\mathbf{h}_2{}^{\mathbf{b}_1}e^{\langle\mathbf{a}_2,\mathbf{b}_1\rangle}f^{s_u}\right)^2$
$V_i \leftarrow \left(\mathbf{g}_2{}^{\mathbf{a}_1}\mathbf{h}_1{}^{\mathbf{b}_2}e^{\langle\mathbf{a}_1,\mathbf{b}_2\rangle}f^{s_v}\right)^2$
                              $\xrightarrow{U_i,V_i}$
                                                                    $x_i \leftarrow_\$ [\![0; P-1]\!]$
                              $\xleftarrow{x_i}$
$\mathbf{a}' \leftarrow \mathbf{a}_1 + x_i\mathbf{a}_2$
$\mathbf{b}' \leftarrow x_i\mathbf{b}_1 + \mathbf{b}_2$
$t \leftarrow s_v + r x_i + s_u x_i^2$

recurse on $\left(i+1, \lceil n/2\rceil, f, \mathbf{g}_1^{x_i} \circ \mathbf{g}_2, \mathbf{h}_1 \circ \mathbf{h}_2^{x_i}, e, C_{i+1} \coloneqq U_i^{x_i^2} C_i^{x_i} V_i, \ell; \mathbf{a}', \mathbf{b}', t\right)$
```

Fig. 2. Argument for Relation $\mathcal{R}'$.

In the full version [**?**], we show that in case n is a power of 2, the verifier then only has to check that

$$\left(\prod_{i=1}^{n} g_i^{\prod_{j\in S_i} x_j}\right)^{4x_{n'+1}a'} \left(\prod_{i=1}^{n} h_i^{\prod_{j\in[\![n]\!]\setminus S_i} x_j}\right)^{4x_{n'+1}b'} e^{4a'b'} f^{4u}$$

$$= \left(U_{n'}^{x_{n'}} \prod_{i=1}^{n'-1} U_i^{x_i x_{i+1}\cdots x_{n'}} C^{x_1\cdots x_{n'}} \prod_{i=1}^{n'-1} V_i^{x_{i+1}\cdots x_{n'}} V_{n'}\right)^{2x_{n'+1}^2} \Gamma^{2x_{n'+1}}\Delta^2,$$

with $S_i \coloneqq \{j \in [\![n']\!] : n' + 1 - j\text{th bit of } i - 1 \text{ is } 0\}$, i.e., do a $(2n + 2n' + 5)$-exponentiation with exponents (in absolute value) less than

$$4 \max\left(2^{\ell} P^{2n'+1}, \overbrace{2^{2\ell} P^{2n'}}^{|a'b'|<}, \overbrace{\left(2n' 2^{b_{\mathbb{G}}+\lambda} P^{n'+1} + 2^{\ell} (P-1)^{n'+2}\right)\left(1 + 2^{\lambda}\right)}^{|u|<}\right).$$

Verification thus requires $O(\ell + b_{\mathbb{G}} + \log(n)\log(P))$ group operations ($n \geq 2$). We also show how to handle verification with a single multi-exponentiation in case n is not a power of 2 unlike previous work.

4.2 Completeness and Security

In the full version [?], we prove that the protocol is complete, honest-verifier zero-knowledge if $\tilde{\Pi}$ is sound, and that it is extractable under the assumptions on the group generator presented in Section **??**. The proof of extractability is based on Lemma **??** and Lemma **??**, and Lemma **??** relies on Lemma **??**. The proof of Lemmas **??** and **??** are given in this section as they are the main ingredients of the proof of extractability which differ from those in the case of groups with public prime orders. The proof of Lemma **??** relies on elementary arithmetic and is given in full version.

Lemma 4.1. *Let n be a natural integer and let $a_0, \ldots, a_n, b$ and N be integers, with $N \geq 1$. Assuming that the a_i integers are not all nil modulo N, the number of tuples $(x_0, \ldots, x_n) \in \mathbb{Z}_N^{n+1}$ such that $a_0 x_0 + \cdots + a_n x_n + b = 0 \bmod N$ is either 0 or $N^n \gcd(a_0, \ldots, a_n, N)$.*

Lemma 4.2. *Consider the problem (depending on λ) of computing, on input $(\mathbb{G}, P) \leftarrow \mathsf{G}\left(1^{\lambda}\right)$ and $f \leftarrow_{\$} \mathbb{G}$ and $(f^{x_i})_{i=0}^{n}$ (for integers $x_i \leftarrow_{\$} \left[\!\left[0; 2^{2b_{\mathbb{G}}+\lambda}(n+1)\right]\!\right]$) an element $C \in \mathbb{G}$ and integers $a_0, \ldots, a_n, b, \delta$ such that $1 < |\delta| < P$, δ does not divide b or at least one of the a_i integers, and $C^{\delta} = f_0^{a_0} \cdots f_n^{a_n} f^{b}$.*

Under the $(T^{\mathrm{strg}}, \varepsilon^{\mathrm{strg}})$-strong-root assumption, the $\left(T^{\mathrm{ord}}, \varepsilon^{\mathrm{ord}}\right)$-small-order assumption, the low-dyadic-valuation assumption and the μ-assumption over G, the probability that any probabilistic algorithm running in time T solves this problem is at most $\left(1/2 - 2^{-\lambda} - (1-\mu)\right)^{-1}\left(\varepsilon^{\mathrm{ord}} + \varepsilon^{\mathrm{strg}} + 1 - \mu\right)$, if T is such that $(n+1)\max(\log(n+1), 1)\log(P) b_{\mathbb{G}} T T_{\mathbb{G}} \leq \Omega\left(\min\left(T^{\mathrm{strg}}, T^{\mathrm{ord}}\right)\right)$.

Proof. Let $\mathcal{A}$ be an algorithm as in the statement of the lemma and assume without loss of generality that $\delta > 0$ (if $\delta < 0$, raise the equality to the power -1). The equality $C^{\delta} = f_0^{a_0} \cdots f_n^{a_n} f^{b}$ implies that $C^{\delta} = f^{\sum_i a_i x_i + b}$. The goal is to show that in case δ does not divide $\sum_i a_i x_i + b$, algorithm $\mathcal{A}$ can be used to violate the assumptions on generator G; and to show that conditioned on the event in which $\mathcal{A}$ solves the problem, the probability that δ divides $\sum_i a_i x_i + b$ is at most $1/2 + 2^{-\lambda} + (1-\mu)$.

More precisely, if δ does not divide $\sum_i a_i x_i + b$, let $d \coloneqq \gcd\left(\delta, \sum_i a_i x_i + b\right)$ and $u, v \in \mathbb{Z}$ such that $d = u\delta + v\left(\sum_i a_i x_i + b\right)$. Then, $f^{d} = \left(f^{u} C^{v}\right)^{\delta}$, i.e.,

$\left((f^u C^v)^{\delta/d} f^{-1}\right)^d = 1_{\mathbb{G}}$. Since $1 \leq d < \delta < P$ by assumption, the small-order assumption over G implies that the element $\tilde{g} := (f^u C^v)^{\delta/d} f^{-1}$ is such that $\tilde{g}^2 = 1_{\mathbb{G}}$ with probability at least $\varepsilon^{\mathrm{ord}}$. If $\tilde{g} = 1_{\mathbb{G}}$ and $d > 1$, then $\left((f^u C^v)^{\delta/d}, d\right)$ is a solution to the strong-root problem. Otherwise,

* if δ/d is odd, then $\tilde{g}^{\delta/d} = \tilde{g}$ and therefore, $(f^u C^v \tilde{g}, \delta/d)$ is a solution to the strong-root problem
* if δ/d is even, then the low-dyadic-valuation assumption on orders implies that $\operatorname{ord}\left((f^u C^v)^{\delta/d}\right)$ is odd, which is impossible if $\operatorname{ord}(f)$ is P-rough (and thus odd) since $\operatorname{ord}(f\tilde{g}) = 2\operatorname{ord}(f)$ in this case.

Consequently, δ does not divide $\sum_i a_i x_i + b$ with probability at most $\varepsilon^{\mathrm{ord}} + \varepsilon^{\mathrm{strg}} + 1 - \mu$.

Since $|a_i|, |b| \leq 2^{O(T)}$, $\sum_i a_i x_i + b$ can be computed in time $O\left((n+1)T(b_{\mathbb{G}} + \log(n+1))\right)$. Then, u and v can be computed in time $O((T + b_{\mathbb{G}} + \log(n+1)) \log P)$ with the extended Euclidean algorithm as $|\sum_i a_i x_i + b| \leq n(n+1)2^{O(T)} 2^{2b_{\mathbb{G}}+\lambda} + 2^{O(T)}$ and $|\delta| \leq P$; and u and v are such that $|u|, |v| \leq \max\left(|\delta|, |\sum_i a_i x_i + b|\right)/d$. Besides, computing δ/d can be done in time $O\left(\log^2 P\right)$ and then $f^u C^v \tilde{g}$ in $O\left(\max\left(T + b_{\mathbb{G}} + \log(n+1), \log P\right)\right) = O\left(T + b_{\mathbb{G}} + \log(n+1)\right)$ group operations since $P \leq 2^{b_{\mathbb{G}}}$. The solution to the strong-root problem can thus be computed in time $O\left((n+1)(b_{\mathbb{G}} + \log(n+1))T + (T + b_{\mathbb{G}} + \log(n+1)) \log(P) T_{\mathbb{G}}\right)$, after the bases $f_0, \ldots, f_n$ have been computed in $O((n+1)\max(\log(n+1), 1) b_{\mathbb{G}})$ group operations.

It remains to show that δ divides $\sum_i a_i x_i + b$ with probability at most $1/2 + 2^{-\lambda} + 1 - \mu$ conditioned on the event in which $\mathcal{A}$ solves the problem. To do so, consider the event in which it occurs. Let p and j respectively be a prime and a positive integer such that p^j divides δ and p^j does not divide b or at least one of the a_i integers. Such p and j necessarily exist for an assumption of the lemma is that δ does not divide b or at least one of the a_i integers. Note that p^j cannot divide all the a_i integers as it would otherwise divide b as well, since it divides $\sum_i a_i x_i + b$. Moreover, if μ-assumption that there are many rough-order elements in the groups generated by G holds, p does not divide $\operatorname{ord}(f)$. Therefore, if the μ-assumption holds, p^j does not divide $a_i \operatorname{ord}(f)$ for some $i \in \llbracket 0; n \rrbracket$.

For $i \in \llbracket 0; n \rrbracket$, let $0 \leq \rho_i < \operatorname{ord}(f)$ be the unique integer such that $x_i = \operatorname{ord}(f) \lfloor x_i / \operatorname{ord}(f) \rfloor + \rho_i$, and note that $f^{x_i} = f^{\rho_i}$. Then, $\sum_i a_i x_i + b = \sum_i a_i \operatorname{ord}(f) \lfloor x_i / \operatorname{ord}(f) \rfloor + \sum_i a_i \rho_i + b = 0 \bmod p^j$ and $a_i \operatorname{ord}(f) \neq 0 \bmod p^j$ for some $i \in \llbracket 0; n \rrbracket$. Lemma **??** shows that the equation $\sum_i A_i X_i + B = 0 \bmod p^j$ with $A_i := a_i \operatorname{ord}(f)$ and $B := \sum_i a_i \rho_i + b$ has at most $p^{jn} \gcd\left(a_0 \operatorname{ord}(f), \ldots, a_n \operatorname{ord}(f), p^j\right)$ solutions, and $\gcd\left(a_0 \operatorname{ord}(f), \ldots, a_n \operatorname{ord}(f), p^j\right)$ is at most p^{j-1} since $a_i \operatorname{ord}(f) \neq 0 \bmod p^j$ for some $i \in \llbracket 0; n \rrbracket$. However, the variables $X_i := \lfloor x_i / \operatorname{ord}(f) \rfloor$ are identically distributed and independent of the values returned by $\mathcal{A}\left(\mathbb{G}, P, f, f^{\rho_0}, \ldots, f^{\rho_n}\right)$; and their distribution is at a statistical distance of at most $\operatorname{ord}(f) 2^{-2b_{\mathbb{G}}-\lambda}(n+1)^{-1} \leq 2^{-b_{\mathbb{G}}-\lambda}(n+1)^{-1}$ from the uniform distribution over $\left\llbracket 0; \left\lfloor (n+1)2^{2b_{\mathbb{G}}+\lambda} / \operatorname{ord}(f) \right\rfloor \right\rrbracket \supseteq \left\llbracket 0; (n+1)2^{b_{\mathbb{G}}+\lambda} \right\rrbracket$. Besides, if a variable X is uniformly distributed over the set $\left\llbracket 0; (n+1)2^{b_{\mathbb{G}}+\lambda} \right\rrbracket$, then the distribution of $X \bmod p^j$ is at a statistical distance of at most $p^j 2^{-b_{\mathbb{G}}-\lambda}(n+1)^{-1} \leq (P-1)2^{-b_{\mathbb{G}}-\lambda}(n+1)^{-1}$ from the uniform distribution over $\mathbb{Z}_{p^j}$. The distribution of the random vector $\left[X_0 \bmod p^j \cdots X_n \bmod p^j\right]$ is then at a statistical distance of at most $P2^{-b_{\mathbb{G}}-\lambda} \leq 2^{-\lambda}$ from the uniform distribution over $\mathbb{Z}_{p^j}^{n+1}$. Consequently, the equation $\sum_i a_i x_i + b = 0 \bmod p^j$ can then be satisfied with probability at most $2^{-\lambda} + p^{j(n+1)-1} / \left(p^j\right)^{n+1} \leq 1/2 + 2^{-\lambda}$ and thus, δ divides $\sum_i a_i x_i + b$ with probability at most $1/2 + 2^{-\lambda} + 1 - \mu$.

In summary, denoting by ε the probability that $\mathcal{A}$ solves the problem of the statement of the lemma, $\varepsilon \leq \varepsilon^{\text{ord}} + \varepsilon^{\text{strg}} + 1 - \mu + \left(1/2 + 2^{-\lambda} + 1 - \mu\right)\varepsilon$, which is equivalent to $\varepsilon \leq \left(1/2 - 2^{-\lambda} - (1-\mu)\right)^{-1} \left(\varepsilon^{\text{ord}} + \varepsilon^{\text{strg}} + 1 - \mu\right)$. □

Lemma 4.3 (Discrete-Logarithm Relations). *Let n be a non-negative integer. Consider the problem (depending on λ) of computing, on the input of $(\mathbb{G}, P) \leftarrow \mathsf{G}\left(1^\lambda\right)$ and of group elements $f \leftarrow_{\$} \mathbb{G}$ and $(f^{x_i})_{i=0}^n$ (for $x_i \leftarrow_{\$} [\![0; 2^{2b_{\mathbb{G}}+\lambda} (n+1)]\!]$), integers $a_0, \ldots, a_n$, b such that $f_0^{a_0} \cdots f_n^{a_n} f^b = 1_{\mathbb{G}}$ although at least one of $a_0, \ldots, a_n, b$ is non-zero. Under the $(T^{\text{strg}}, \varepsilon^{\text{strg}})$-strong-root assumption, the $\left(T^{\text{ord}}, \varepsilon^{\text{ord}}\right)$-small-order assumption, the low-dyadic-valuation assumption and the μ-assumption over G, the probability that any probabilistic algorithm running in time at most T solves this problem is at most*

$$\varepsilon^{\text{strg}} + \max\left(2^{-b_{\mathbb{G}}-\lambda+1}, \left(1/2 - 2^{-\lambda} - (1-\mu)\right)^{-1} \left(\varepsilon^{\text{ord}} + \varepsilon^{\text{strg}} \ +1 - \mu\right)\right)$$

if T is such that $(n+1)\max(\log(n+1), 1)\log(P) b_{\mathbb{G}} T T_{\mathbb{G}} \leq \Omega\left(\min\left(T^{\text{strg}}, T^{\text{ord}}\right)\right)$.

Proof. Let $\mathcal{A}$ be an algorithm as in the statement of the lemma and denote the probability that it solves the problem by ε. If $a_0 = \cdots = a_n = 0$, then $b \neq 0$ by assumption and a lemma in the full version [?, Lemma 3.4] shows that since $f^b = 1_{\mathbb{G}}$, there exists an algorithm that solves the strong-root problem in time at most $T + O(\log b)$ with probability at least ε, and since $b = 2^{O(T)}$, $\varepsilon \leq \varepsilon^{\text{strg}}$. Now turn to the case in which $a_i \neq 0$ for some $i \in [\![0; n]\!]$. If $n = 0$, then $f^{a_0 x_0 + b} = 1_{\mathbb{G}}$ by assumption. Writing x_0 as $x_0 = \operatorname{ord}(f) \lfloor x_0 / \operatorname{ord}(f) \rfloor + \rho_0$ for $0 \leq \rho_0 < \operatorname{ord}(f)$, the random variable $X_0 := \lfloor x_0 / \operatorname{ord}(f) \rfloor$ is independent of the values returned by $\mathcal{A}(\mathbb{G}, P, f, f^{\rho_0})$, and is at a statistical distance of at most $\operatorname{ord}(f) 2^{-2b_{\mathbb{G}}-\lambda} \leq 2^{-b_{\mathbb{G}}-\lambda}$ from the uniform distribution over $\left[\!\left[0; \left\lfloor 2^{2b_{\mathbb{G}}+\lambda} / \operatorname{ord}(f)\right\rfloor\right]\!\right] \supseteq \left[\!\left[0; 2^{b_{\mathbb{G}}+\lambda}\right]\!\right]$. However, for $A_0 := a_0 \operatorname{ord}(f)$ and $B := a_0 \rho_0 + b$, the equation $A_0 X_0 + B = 0$ in $\mathbb{Z}$ has no solution if $A_0 \nmid B$ and exactly one otherwise. Therefore, the probability that $a_0 x_0 + b = 0$ in $\mathbb{Z}$ is at most $2^{-b_{\mathbb{G}}-\lambda+1}$, and there exists an algorithm that solves the strong-root problem in time at most $O(T)$ with probability at least $\varepsilon - 2^{-b-\mathbb{G}-\lambda+1}$, so $\varepsilon \leq \varepsilon^{\text{strg}} + 2^{-b_{\mathbb{G}}-\lambda+1}$.

If $n > 0$, it suffices to prove that the probability that $f_0^{a_0} \cdots f_n^{a_n} f^b = 1_{\mathbb{G}}$ and $\sum_i a_i x_i + b = 0$ is at most $\left(1/2 - 2^{-\lambda} - (1-\mu)\right)^{-1} \left(\varepsilon^{\text{ord}} + \varepsilon^{\text{strg}} + 1 - \mu\right)$. Then, in case $f^{\sum_i a_i x_i + b} = 1_{\mathbb{G}}$ and $\sum_i a_i x_i + b \neq 0$, a lemma in the full version [?, Lemma 3.4] shows that this probability is at most $\varepsilon^{\text{strg}}$. This then would imply that

$$\varepsilon \leq \varepsilon^{\text{strg}} + \left(1/2 - 2^{-\lambda} - (1-\mu)\right)^{-1} \left(\varepsilon^{\text{ord}} + \varepsilon^{\text{strg}} + 1 - \mu\right).$$

Suppose that $\sum_i a_i x_i + b = 0$ (and $f_0^{a_0} \cdots f_n^{a_n} f^b = 1_{\mathbb{G}}$). Let $d \coloneqq \gcd(a_0, \ldots, a_n)$ and note that d necessarily divides b. Besides, $\sum_i a_i x_i + b = 0$ if and only if $\sum_i (a_i/d) x_i + (b/d) = 0$ and therefore, we have $f_0^{a_0/d} \cdots f_n^{a_n/d} f^{b/d} = 1_{\mathbb{G}}$ with $\gcd(a_0/d, \ldots, a_n/d) = 1$. However, $1_{\mathbb{G}}^2 = 1_{\mathbb{G}} = f_0^{a_0/d} \cdots f_n^{a_n/d} f^{b/d}$ although the integers a_i/d cannot all be even as they are coprime. Lemma **??** then implies that $\sum_i a_i x_i + b = 0$ with probability at most $\left(1/2 - 2^{-\lambda} - (1-\mu)\right)^{-1} \left(\varepsilon^{\mathrm{ord}} + \varepsilon^{\mathrm{strg}} \right.$ $\left. +1 - \mu\right)$. □

5 Succinct Argument for Diophantine Equations

This section gives a succinct argument to argue satisfiability of Diophantine equations. Although Davis, Putnam, Robinson and Matiyasevich [?] showed that there does not exist an algorithm that can decide whether any Diophantine equation has a solution (thereby giving a negative answer to Hilbert's tenth problem), one can argue in zero-knowledge knowledge of a solution, if a solution is known to the prover, which convinces the verifier that the equation is satisfiable.

Damgård and Fujisaki gave [?, Section 4.2] a protocol to argue, given three commitments C_1, C_2, C_3 computed with their scheme, knowledge of openings x_1, x_2, x_3 such that $x_3 = x_1 x_2$. Therefore, to show the satisfiability of an ν-variate polynomial $\sum_{\mathbf{i} \in \mathbb{N}^\nu} a_{\mathbf{i}} x_1^{i_1} \cdots x_\nu^{i_\nu}$ of total degree δ using their scheme, if the polynomial can be computed in $M(\nu, \delta)$ multiplications, then one would have to compute $2M(\nu, \delta) + 1$ integer commitments and compute $M(\nu, \delta)$ multiplication-consistency arguments. As Damgård and Fujisaki's scheme is additively homomorphic, the verifier can verify addition itself.

Computing a monomial $x_1^{i_1} \cdots x_\nu^{i_\nu}$ can be done in at most $\delta - 1$ multiplications since the polynomial is of total degree δ. Without any further restriction on the polynomial than its number of variables ν and its total degree δ, the best bound on the number of multiplications (between variables) one can give is $\delta - 1$ as δ could be less than ν, and all i_k at most 1. Evaluating an ν-variate polynomial of total degree δ thus *a priori* requires $(\delta - 1)\binom{\nu+\delta}{\delta}$ multiplications as such a polynomial has at most $\binom{\nu+\delta}{\delta}$ monomials. This can be improved to $\binom{\nu+\delta}{\delta} - \nu - 1 \leq \binom{\nu+\delta}{\delta}$ multiplications by evaluating all possible monomials (even those which may have coefficient 0) recursively by increasing degree and storing the previous evaluations. There exist more efficient methods for specific polynomials (e.g., recursive Horner's method for polynomials with a small numbers of monomials of large degree) but no better upper-bound on the number of multiplications is known for generic polynomials.

Consider a prover that wants to argue the satisfiability of a (generic) ν-variate polynomial of total degree δ with integer coefficients of absolute value upper-bounded by 2^H for some integer H. The communication complexity of the arguments of the first multiplication gates are of order $\Omega(\log P + \ell + b_{\mathbb{G}})$ if ℓ denotes the maximum bit length of any coordinate in the solution. Since the total degree of the polynomial is δ, the bit length of the witness at the maximum-depth multiplication gates can be as large as $\delta\ell + \log\left(\binom{\nu+\delta}{\delta}\right) H$ and the communication complexity of the argument of the satisfiability of the Diophantine equation (i.e., the proof that the polynomial actually evaluates to 0) is $\Omega\left(\delta\ell + \log\left(\binom{\nu+\delta}{\delta}\right) H + b_{\mathbb{G}}\right)$. The overall communication complexity with Damgård and Fujisaki's scheme is therefore upper-bounded by $O\left(\binom{\nu+\delta}{\delta}\left(\delta\ell + \log\left(\binom{\nu+\delta}{\delta}\right) H + b_{\mathbb{G}}\right)\right)$ and lower-bounded by $\Omega\left(\binom{\nu+\delta}{\delta}(\ell + b_{\mathbb{G}})\right)$ for generic polynomials.

This section shows how to argue the satisfiability of Diophantine equations with a communication complexity of order $O\left(\delta\ell + \min(\nu,\delta)\log\left(\nu+\delta\right) b_{\mathbb{G}} + H\right)$.

5.1 Arguments via Polynomial-Degree Reductions

Our approach to argue for Diophantine satisfiability is different and is inspired by Skolem's method [?]. The idea is to give a systematic method to turn any polynomial equation to another of degree at most 4 by increasing the number of variables so that the satisfiability of one polynomial implies that of the other. The resulting polynomial is such that its satisfiability is equivalent to the satisfiability (over the integers) of a Hadamard product of the form $\mathbf{a}_L \circ \mathbf{a}_R = \mathbf{a}_O$ and of linear equations with the entries of $\mathbf{a}_L$, $\mathbf{a}_R$ and $\mathbf{a}_O$ as indeterminate. The length of these latter vectors is the number of variables in the resulting polynomial, and if the original polynomial is ν-variate and of total degree at most δ, then the new polynomial has at most $\nu\lfloor\log\delta\rfloor + (\delta-1)\mu$ variables, where $\mu \leq \binom{\nu+\delta}{\delta}$ is the number of monomials in the original polynomial.

On this account, if one can argue for the satisfiability of such Hadamard products and linear constraints, then one can argue for the satisfiability of the original polynomial. In the protocol given in Section **??**, the prover only sends logarithmically many group elements in the length of the vectors in the Hadamard product, and a constant number of integers. The bit length of those integers is upper-bounded by $O\left(\delta\ell + b_{\mathbb{G}} + \min(\nu,\delta)\log\left(\nu+\delta\right)\log P + H\right)$ if the bit length of the witness is upper-bounded by ℓ and the bit length of each coefficient of the polynomial is at most H.

Reducing Arbitrary Polynomials to Polynomials of Degree at most 4. We now give a systematic procedure to reduce any Diophantine equation into an equation of degree at most 4 of which the satisfiability can be reduced to the satisfiability of a Hadamard product and linear constraints; and the Hadamard product and the constraints can be read immediately from the resulting polynomial. The presentation is gradual as it starts with ν-variate affine equations, proceeds with ν-variate Diophantine equations in which the degree in each variable is at most 1, further tackles univariate polynomials of arbitrary degree and then considers arbitrary Diophantine equations. The method applies to every multivariate integer polynomial, but for specific polynomials, more astute techniques could lead to a smaller number of new variables and/or constraints.

Step 1–Affine Equations. Given an integer polynomial $a_1x_1 + \cdots + a_\nu x_\nu + b \in \mathbb{Z}[x_1, \ldots, x_\nu]$, set $\mathbf{a}_O \leftarrow \begin{bmatrix} x_1 \cdots x_\nu \end{bmatrix}$ and for all $i \in [\![\nu]\!]$, set $\mathbf{a}_{L,i} = 1$ and $\mathbf{a}_{R,i} = x_i$. The equation $a_1x_1 + \cdots + a_\nu x_\nu + b = 0$ is satisfied if and only if

$\left\langle \begin{bmatrix} a_1 \cdots a_\nu \end{bmatrix}, \mathbf{a}_O \right\rangle = -b$ and $\mathbf{a}_L \circ \mathbf{a}_R = \mathbf{a}_O$. Note that no variable or linear constraint was added to the system of equations.

Step 2–Restricted Diophantine Equations. Consider an integer polynomial $\sum_{\mathbf{i}\in\mathbb{N}^\nu} a_\mathbf{i} x_1^{i_1} \cdots x_\nu^{i_\nu} \in \mathbb{Z}[x_1, \ldots, x_\nu]$ of total degree δ s.t. $a_\mathbf{i} \neq 0_\mathbb{Z} \implies \mathbf{i} \in \{0,1\}^\nu$, i.e., the polynomial is of degree at most 1 in each variable. For all $\mathbf{i} \in \mathbb{N}^\nu \backslash \{0_{\mathbb{N}^\nu}\}$ such that $a_\mathbf{i} \neq 0_\mathbb{Z}$, let $\{j_1, \ldots, j_{w(\mathbf{i})}\}$ be the subset of $[\![\nu]\!]$ such that $j_1 < \cdots < j_{w(\mathbf{i})}$ and $i_{j_1} = \cdots = i_{w(\mathbf{i})} = 1$, with $w(\mathbf{i})$ denoting the Hamming weight of $\mathbf{i}$ (which is necessarily less than δ). If $w(\mathbf{i}) > 1$, introduce new variables

$$u_{\mathbf{i},1} \leftarrow x_{j_1} x_{j_2}, \quad u_{\mathbf{i},2} \leftarrow u_{\mathbf{i},1} x_{j_3}, \quad \ldots, \quad u_{\mathbf{i},w(\mathbf{i})-1} \leftarrow u_{\mathbf{i},w(\mathbf{i})-2} x_{j_{w(\mathbf{i})}},$$

with the convention that $u_{\mathbf{i},0} \coloneqq x_{j_1}$. Note that $\sum_{\mathbf{i}\in\mathbb{N}^\nu} a_\mathbf{i} x_1^{i_1} \cdots x_\nu^{i_\nu} = 0$ if and only if

$$\sum_{\substack{\mathbf{i}\in\mathbb{N}^\nu :\, a_\mathbf{i} \neq 0_\mathbb{Z} \\ w(\mathbf{i})>1}} \sum_{k=1}^{w(\mathbf{i})-1} \left(u_{\mathbf{i},k} - u_{\mathbf{i},k-1} x_{j_{k+1}} \right)^2 + \left(\sum_{\mathbf{i}\in\mathbb{N}^\nu} \mathbf{a}_\mathbf{i} u_{\mathbf{i},w(\mathbf{i})-1} \right)^2 = 0,$$

with the convention that $u_{\mathbf{0}_{\mathbb{N}^\nu},-1} = 1$. This latter polynomial is of degree 4, and the equation is satisfied if and only if the linear equation $\sum_{\mathbf{i}\in\mathbb{N}^\nu} \mathbf{a}_\mathbf{i} u_{\mathbf{i},w(\mathbf{i})-1} = 0$ is as well as the constraints $u_{\mathbf{i},k} - u_{\mathbf{i},k-1} x_{j_{k+1}} = 0$. Set then

$$\mathbf{a}_L \leftarrow \begin{bmatrix} x_{j_1} \ u_{\mathbf{i},1} \cdots u_{\mathbf{i},w(\mathbf{i})-2} \end{bmatrix} \ \mathbf{a}_R \leftarrow \begin{bmatrix} x_{j_2} \ x_{j_3} \cdots x_{j_{w(\mathbf{i})}} \end{bmatrix} \ \mathbf{a}_O \leftarrow \begin{bmatrix} u_{\mathbf{i},1} \ u_{\mathbf{i},2} \cdots u_{\mathbf{i},w(\mathbf{i})-1} \end{bmatrix},$$

and introduce the linear constraints $\mathbf{a}_{L,i+1} - \mathbf{a}_{O,i} = 0$ for $i \in \{1, \ldots, w(\mathbf{i}) - 2\}$. The procedure introduces at most $\delta - 1$ new variables and $\delta - 2$ new linear constraints per monomial, and since there are at most $\binom{\nu+\delta}{\delta}$ monomials in an ν-variate polynomial of total degree δ, that is at most $(\delta - 1)\binom{\nu+\delta}{\delta}$ variables and $(\delta - 2)\binom{\nu+\delta}{\delta}$ constraints.

Step 3–Univariate Polynomials. Given a polynomial $Z = a_0 + a_1 x + \cdots + a_\delta x^\delta \in \mathbb{Z}[x]$ of degree $\delta \geq 2$, introduce variables

$$u_1 \leftarrow x^2, \quad u_2 \leftarrow u_1^2, \quad \ldots, \quad u_{\lfloor \log \delta \rfloor} \leftarrow u_{\lfloor \log \delta \rfloor - 1}^2.$$

Now notice that $a_0 + a_1 x + \cdots + a_\delta x^\delta = 0$ if and only if

$$\left(u_1 - x^2 \right)^2 + \sum_{i=2}^{\lfloor \log \delta \rfloor} \left(u_i - u_{i-1}^2 \right)^2 + \left(Z'(x, u_1, \ldots, u_{\lfloor \log \delta \rfloor}) \right)^2 = 0,$$

where $Z'(x, u_1, \ldots, u_{\lfloor \log \delta \rfloor})$ is $\lfloor \log \delta \rfloor + 1$-variate integer polynomial in which the degree of each variable is at most 1, i.e., if and only if $Z'(x, u_1, \ldots, u_{\lfloor \log \delta \rfloor}) = 0$ and the constraints $u_1 - x^2 = 0$ and $u_{i+1} - u_i^2 = 0$ are satisfied.

Since

$$\sum_{i=0}^{\delta} a_i x^i = a_0 + \sum_{k=0}^{\lfloor \log \delta \rfloor} \sum_{i=2^k}^{2^{k+1}-1} a_i x^i = a_0 + \sum_{k=0}^{\lfloor \log \delta \rfloor} \sum_{i=2^k}^{2^{k+1}-1} a_i x^{i_0} u_1^{i_1} \cdots u_{k-1}^{i_{k-1}} u_k,$$

where $i_0, \ldots, i_{k-1}$ is the binary decomposition of i and $a_i \coloneqq 0$ for $i > \delta$, this give an explicit expression for Z'.

Set then $\mathbf{a}_L \leftarrow \mathbf{a}_R \leftarrow \left[x\ u_1 \cdots u_{\lfloor \log \delta \rfloor - 1}\right]$ and $\mathbf{a}_O \leftarrow \left[u_1\ u_2 \cdots u_{\lfloor \log \delta \rfloor}\right]$, and introduce constraints

$$\mathbf{a}_{L,i+1} - \mathbf{a}_{O,i} = \mathbf{a}_{R,i+1} - \mathbf{a}_{O,i} = 0$$

for all $i \in [\![\lfloor \log \delta \rfloor - 1]\!]$.

As the second step shows that the satisfiability of Z' can be reduced to a Hadamard product and linear constraints, the satisfiability of Z can be reduced to a Hadamard product and linear constraints. This procedure introduces $\lfloor \log \delta \rfloor$ new variables and $2\left(\lfloor \log \delta \rfloor - 1\right)$ new linear constraints. It is important for Step 4 to remark that the number of monomial of Z' is at most the same as the number of monomials in Z.

Step 4–Arbitrary Diophantine Equations. For any integer polynomial $Z = \sum_{\mathbf{i} \in \mathbb{N}^\nu} a_{\mathbf{i}} x_1^{i_1} \cdots x_\nu^{i_\nu} \in \mathbb{Z}[x_1, \ldots, x_\nu]$ (for $\nu \geq 2$) of total degree δ, apply Step 3 to Z considering it as a polynomial in $\mathbb{Z}[x_2, \ldots, x_\nu][x_1]$, i.e., a polynomial in x_1 with coefficients in $\mathbb{Z}[x_2, \ldots, x_\nu]$. Let Z' be the resulting polynomial with coefficients in $\mathbb{Z}[x_2, \ldots, x_\nu]$ and of degree at most 1 in each variable as in Step 3. Repeat Step 3 with Z' and variable x_2. After Step 3 has been repeated for each $x_1, \ldots, x_\nu$, at most $\nu \lfloor \log \delta \rfloor$ new variables and $2\nu(\lfloor \log \delta \rfloor - 1)$ new linear constraints have been introduced, the resulting polynomial is of degree at most 1 in all variables and has coefficients in $\mathbb{Z}$. Concerning its total degree, note that during the process, for each monomial $x_1^{i_1} \cdots x_\nu^{i_\nu}$, the term $x_k^{i_k}$ is replaced by at most one variable if $i_k \leq 2$ and by the product of $\log i_k + 1 \leq i_k$ variables if ß$_k > 2$ for all $k \in [\![\nu]\!]$, so the total degree remains at most δ. Now apply then Step 2 to the resulting polynomial.

In summary, the procedure reduces the satisfiability of any polynomial in $\mathbb{Z}[x_1, \ldots, x_\nu]$ of total degree δ with μ monomials ($\mu \leq \binom{\nu+\delta}{\delta}$ necessarily) to the satisfiability of a Hadamard product $\mathbf{a}_L \circ \mathbf{a}_R = \mathbf{a}_O$, with $\mathbf{a}_L$, $\mathbf{a}_R$ and $\mathbf{a}_O$ integer vectors of length at most $\nu \lfloor \log \delta \rfloor + (\delta - 1)\mu$, and Q linear constraints of the form

$$\langle \mathbf{w}_{L,q}, \mathbf{a}_L \rangle + \langle \mathbf{w}_{R,q}, \mathbf{a}_R \rangle + \langle \mathbf{w}_{O,q}, \mathbf{a}_O \rangle = c_q$$

for all $q \in [\![Q]\!]$ with $Q \leq 1 + 2\nu(\lfloor \log \delta \rfloor - 1) + (\delta - 2)\mu$ and with $\mathbf{w}_{L,q}, \mathbf{w}_{R,q}, \mathbf{w}_{O,q}$ integer vectors and $c_q \in \mathbb{Z}$. The coefficients of the linear constraints introduced by the procedure are in $\{-1, 0, 1\}$, except for one of which the coefficients are the coefficients of the original polynomial.

Example. As a simple illustration of the procedure, consider the polynomial $2x^3 + xy - 1$. The procedure introduces new variables $u \leftarrow x^2$, $v \leftarrow xy$ and $w \leftarrow ux$, and the equation $2x^3 + xy - 1 = 0$ is satisfiable if and only if $\left(u - x^2\right)^2 + (v - xy)^2 + (w - ux)^2 + (2w + v - 1)^2 = 0$ also is, which allows to write a Hadamard product and linear constraints which are satisfiable if and only if this latter equation is.

Diophantine Equations as Circuits. It is worth noting that any polynomial in $\mathbb{Z}[x_1, \ldots, x_\nu]$ can naturally be viewed as an arithmetic circuit with integer inputs, and addition gates correspond to addition between two integers and similarly for multiplication gates. One could then think of applying the procedure of Bootle et al. [**?**, Appendix A] to turn the polynomial in a system of linear constraints and a Hadamard product. However, their procedure a priori requires to put matrices in reduced Row-Echelon form, which is not always possible with integer matrices as one cannot divide in $\mathbb{Z}$. We explain how to overcome this obstacle in the full version [**?**].

In any case, the issue with using this procedure to argue for Diophantine satisfiability is that one cannot readily infer the constraints from the initial polynomial and one must always determine them on a case-by-case basis. Besides, if one uses the circuit directly inferred by the monomials of the polynomial without introducing new variables to decrease its degree (which would amount to modifying the circuit), computing x_1^δ for instance requires $\delta - 1$ multiplications instead of $\lfloor \log \delta \rfloor$ as with our method.

5.2 Protocol

Section **??** shows how to reduce the satisfiability of any polynomial in $\mathbb{Z}[x_1, \ldots, x_\nu]$ of total degree δ with μ monomials ($\mu \leq \binom{\nu+\delta}{\delta}$ necessarily) to the satisfiability of a Hadamard product $\mathbf{a}_L \circ \mathbf{a}_R = \mathbf{a}_O$, with $\mathbf{a}_L$, $\mathbf{a}_R$ and $\mathbf{a}_O$ integer vectors of length at most $\nu \lfloor \log \delta \rfloor + (\delta - 1)\mu$, and $1 + 2\nu(\lfloor \log \delta \rfloor - 1) + (\delta - 2)\mu$ linear constraints of the form

$$\langle \mathbf{w}_{L,q}, \mathbf{a}_L \rangle + \langle \mathbf{w}_{R,q}, \mathbf{a}_R \rangle + \langle \mathbf{w}_{O,q}, \mathbf{a}_O \rangle = c_q$$

for all $q \in [\![Q]\!]$, with $\mathbf{w}_{L,q}, \mathbf{w}_{R,q}, \mathbf{w}_{O,q}$ integer vectors and $c_q \in \mathbb{Z}$.

To argue for Diophantine satisfiability, it thus suffices to give a protocol protocol such relations. The following protocol is actually for more general relations in which variables of the polynomial can be committed (with the scheme in Section **??**), which allows to argue on committed values while saving the cost of encoding the commitment scheme as an integer polynomial. More precisely, the protocol is for the relation

$$\Big\{\Big(\mathbf{W}_L, \mathbf{W}_R, \mathbf{W}_O \in \mathbb{Z}^{Q \times n}, \mathbf{W}_V \in \mathbb{Z}^{Q \times m}, \mathbf{V} \in \mathbb{G}^m, \mathbf{c} \in \mathbb{Z}^Q, \ell \in \mathbb{N}^*; \mathbf{a}_L, \mathbf{a}_R, \mathbf{a}_O \in \mathbb{Z}^n, \mathbf{v}, \rho \in \mathbb{Z}^m\Big) :$$
$$\mathbf{a}_L \circ \mathbf{a}_R = \mathbf{a}_O \wedge \mathbf{W}_L \mathbf{a}_L^{\mathrm{T}} + \mathbf{W}_R \mathbf{a}_R^{\mathrm{T}} + \mathbf{W}_O \mathbf{a}_O^{\mathrm{T}} = \mathbf{W}_V \mathbf{v}^{\mathrm{T}} + \mathbf{c}^{\mathrm{T}} \wedge \forall i \in [\![m]\!]\ V_i^2 = (e^{v_i} f^{\rho_i})^4 \Big\}$$

given parameters $(\mathbb{G}, P, n, Q, m, f)$ such that $f \in \mathbb{G}$ and $n, Q, m \in \mathbb{N}^*$, and $(\mathbf{g}, \mathbf{h}, \pi_{crs}) \in \mathbb{G}^{2n} \times \{0,1\}^*$. For fixed parameters n, Q and m, Section **??** shows that the protocol allows to prove the satisfiability of any polynomial in $\mathbb{Z}[X_1, \ldots, X_\nu]$ of total degree δ and with μ monomials if $\nu \lfloor \log \delta \rfloor + (\delta - 1)\mu \leq n$ and $1 + 2\nu\,(\lfloor \log \delta \rfloor - 1) + (\delta - 2)\mu + m \leq Q$. The additional term m in the number of constraints compared to the previous section is to ensure the consistency between the committed variables $\mathbf{v}$ and the ones in the inner product.

Bünz et al. [**?**] gave a protocol for a similar relation in $\mathbb{Z}_p$ instead of $\mathbb{Z}$ to argue for the satisfiability of arithmetic circuits over $\mathbb{Z}_p$ (without the bounds related to integer polynomials as it was not their target) that is inspired by the one of Bootle et al. [?]. The general idea of our protocol for this relation is similar to the two previous ones, but there are key differences that arise from the fact that $\mathbb{Z}$ is not a field. The full version [?] gives details about the construction of the protocol. The main differences with that of Bünz et al. is that (1) one cannot send integers in the protocol as they may contain information about the witness (2) the polynomials $l(X)$ and $r(X)$ are different and of higher degree again because $\mathbb{Z}$ is not a field and (3) the commitment-key switching techniques used in their protocol is not applicable because the group order is unknown.

Building Blocks. The protocol builds mainly on the protocol on Figure **??**, and on three auxiliary protocols: a protocol Π_{crs} to prove that the CRS is well-formed [?, Section 5.3], a protocol Π' to aggregate arguments of opening to integer commitments [?, Section 5.2] and a protocol $\tilde{\Pi}$ to argue knowledge of an integer vector that opens to commitments in different bases [?, Section 5.4], i.e., a base-switching argument. These arguments may be in the random-oracle model with an oracle $\mathcal{H}$.

Protocol Algorithms. The protocol is denoted Π. The parameter-generation algorithm and the CRS generator are as in Section **??**. The algorithms of the prover and the verifier are given on Figure **??**. On that figure, $\mathbf{W}$ denotes the matrix $\left[\mathbf{W}_L\ \mathbf{W}_R\ \mathbf{W}_O\ \mathbf{W}_V\right]$. The values ℓ', $\tilde{\ell}$ and $\ell_{??}$ are given in Section **??**.

Prover-Communication Complexity. In the full version [?], we show that the prover sends $O\left(\ell + \log(n) b_{\mathbb{G}} + \log Q + \log m + \log \|W\|_\infty\right)$ bits during the protocol (the term $\log m$ disappears in case $m = 0$). Therefore, for a polynomial in $\mathbb{Z}[X_1, \ldots, X_\nu]$ of total degree δ, with μ monomials and with coefficients less than 2^H in absolute value, assuming that $\nu\lfloor\log\delta\rfloor + (\delta-1)\mu \leq n$ and that $1 + 2\nu\left(\lfloor\log\delta\rfloor - 1\right) + (\delta-2)\mu + m \leq Q$, the communication complexity of the protocol is of order $O\left(\delta\ell'' + \log\left(\delta\binom{\nu+\delta}{\delta}\right) b_{\mathbb{G}} + H\right) = O\left(\delta\ell'' + \min(\nu,\delta)\log\left(\nu+\delta\right) b_{\mathbb{G}} + H\right)$, where ℓ'' is the maximum bit length of the integers in the solution. Here $H = \lfloor\log\|W\|_\infty\rfloor + 1$ as the procedure gives linear constraints determined by the coefficients of the polynomial.

Verification Efficiency. Similarly to Section **??**, the verifications of Π', $\tilde{\Pi}$ and the protocol on Figure **??** can each be done via single multi-exponentiations, with exponents of at most $O\left(\ell + b_{\mathbb{G}} + \log(n)\log(P) + \log Q + \log m + \log\|W\|_\infty\right)$ bits. For a polynomial in $\mathbb{Z}[X_1, \ldots, X_\nu]$ of total degree δ, with μ monomials and with coefficients less than 2^H in absolute value, that is $O\left(\delta\ell'' + b_{\mathbb{G}} + \min(\nu,\delta)\log\left(\nu+\delta\right)\right.$ $\left.\log P + H\right)$ bits, where ℓ'' is the maximum bit length of the integers in the solution.

Completeness and Security. In the full version [?], we show that the protocol Π is complete, honest-verifier zero-knowledge, and extractable under the assumptions on the group generator.

$\mathcal{P}\ (f, \mathbf{g}, \mathbf{h}, e, \pi_{crs}, \mathbf{W}, \mathbf{V}, \mathbf{c}, \ell; \mathbf{a}_L, \mathbf{a}_R, \mathbf{a}_O, \mathbf{v}, \rho)$ $\mathcal{V}$

$$\mathbf{a}_L \circ \mathbf{a}_R = \mathbf{a}_O \wedge \mathbf{W}_L \mathbf{a}_L^{\mathrm{T}} + \mathbf{W}_R \mathbf{a}_R^{\mathrm{T}} + \mathbf{W}_O \mathbf{a}_O^{\mathrm{T}} = \mathbf{W}_V \mathbf{v}^{\mathrm{T}} + \mathbf{c}^{\mathrm{T}} \wedge \forall i \in [\![m]\!]\ V_i^2 = (e^{v_i} f^{\rho_i})^4$$
$$\left\| \left[\mathbf{a}_L\ \mathbf{a}_R\ \mathbf{a}_O\ \mathbf{v}\ \rho \right] \right\|_\infty < 2^\ell$$

---

return $\bot$ if $FS.\Pi_{crs}^{\mathcal{H}}.\mathsf{Vf}\left((\mathbb{G}, P, f, 2n+1), (\mathbf{g}, \mathbf{h}, e), \pi_{crs}\right) = 0$
$\rho_I, \rho_O \leftarrow_\$ \left[\!\!\left[0; 2^{b_{\mathbb{G}}+\lambda+3}\right]\!\!\right]$
$C_I \leftarrow (\mathbf{g}^{\mathbf{a}_L} \mathbf{h}^{\mathbf{a}_R} f^{\rho_I})^2$
$C_O \leftarrow (\mathbf{g}^{\mathbf{a}_O} f^{\rho_O})^2$

$$\xrightarrow{C_I, C_O}$$

$\mathbf{y} \leftarrow_\$ [\![0; P-1]\!]^n$
$\mathbf{z} \leftarrow_\$ [\![0; P-1]\!]^Q$

$$\xleftarrow{\mathbf{y}, \mathbf{z}}$$

$l(X) \leftarrow (\mathbf{a}_L + \mathbf{z}\mathbf{W}_R)\, X + \mathbf{a}_O X^2 + \mathbf{a}_L X^3 - \mathbf{a}_O X^4$
$r(X) \leftarrow -\mathbf{1}^n + \mathbf{z}\mathbf{W}_O + (\mathbf{a}_R + \mathbf{z}\mathbf{W}_L)\, X + \mathbf{y} X^2 + \mathbf{y} \circ \mathbf{a}_R X^3$
$l(X), r(X) \in \mathbb{Z}^n[X]$ and $\mathbf{1}^n := \left[1\ 1\ \cdots\ 1\right] \in \mathbb{Z}^n$
$t(X) \leftarrow \langle l(X), r(X) \rangle = \sum_{i=1}^{7} t_i X^i$
$t(X) \in \mathbb{Z}[X]$ and $t_6 = 0$
$\forall i \in [\![7]\!] \setminus \{2, 6\}$, $s_i \leftarrow_\$ \left[\!\!\left[0; 2^{b_{\mathbb{G}}+\lambda+3}\right]\!\!\right], T_i \leftarrow \left(e^{t_i} f^{s_i}\right)^2$
Commit to the non-zero coefficients of $t(X)$ except for t_2
$\rho_I' \leftarrow_\$ \left[\!\!\left[0; 2^{b_{\mathbb{G}}+\lambda+3}\right]\!\!\right]$, $C_I' \leftarrow \left(\mathbf{g}^{\mathbf{a}_L} \mathbf{h}^{\mathbf{y} \circ \mathbf{a}_R} f^{\rho_I'}\right)^2$
Commit to the inputs in $(\mathbf{g}, \mathbf{h}')$ with $\mathbf{h}' := \left[h_1^{y_1}\ h_2^{y_2}\ \cdots\ h_n^{y_n}\right]$

$$\xrightarrow{\mathbf{T} := \left[T_1\ T_3\ T_4\ T_5\ T_7\right], C_I'}$$

$$\xleftrightarrow{(\tau', b') \leftarrow \langle \Pi'.\mathsf{Prove}(m+5, f, e, \mathbf{T}, \mathbf{V}, \ell'; \mathbf{t}, \mathbf{s}) \rightleftharpoons \Pi'.\mathsf{Vf}(m+5, f, e, \mathbf{T}, \mathbf{V}, \ell') \rangle}$$

Argue knowledge of representations of $(T_i)_{i \neq 2,6}$ and $(V_j)_{j=1}^m$ in (e, f)

if $b' = 0$ return $\bot$

$$\xleftrightarrow{\left(\tilde{\tau}, \tilde{b}\right) \leftarrow \left\langle \tilde{\Pi}.\mathsf{Prove}\left(2, f, (\mathbf{g}, \mathbf{h}), (\mathbf{g}, \mathbf{h}'), C_I, C_I', \tilde{\ell}; \mathbf{a}_L, \mathbf{a}_R, \rho_I, \rho_I'\right) \rightleftharpoons \tilde{\Pi}.\mathsf{Vf}\left(2, f, (\mathbf{g}, \mathbf{h}), (\mathbf{g}, \mathbf{h}'), C_I, C_I', \tilde{\ell}\right) \right\rangle}$$

Argue that C_I and C_I' are commitments to the same inputs in $(\mathbf{g}, \mathbf{h})$ and $(\mathbf{g}, \mathbf{h}')$

if $\tilde{b} = 0$ return $\bot$

$x \leftarrow_\$ [\![0; P-1]\!]$

$$\xleftarrow{x}$$

$\mathbf{l}, \mathbf{r} \leftarrow l(x), r(x) \in \mathbb{Z}^n$
$\sigma \leftarrow \rho_I x + \rho_I' x^3 + \rho_O \left(x^2 - x^4\right) + s_1 x + \langle \mathbf{z}\mathbf{W}_V, \rho \rangle\, x^2$
$+ \sum_{3 \leq i \neq 6 \leq 7} s_i x^i$ // $\rho = \left[\rho_1\ \cdots\ \rho_m\right]$

$$W_L, W_R, W_O \leftarrow \mathbf{h}^{\mathbf{z}\mathbf{W}_L}, \mathbf{g}^{\mathbf{z}\mathbf{W}_R}, \mathbf{h}^{\mathbf{z}\mathbf{W}_O}$$
$$C_{\mathbf{l},\mathbf{r}} \leftarrow C_I^x {C_I'}^{x^3} C_O^{(x^2 - x^4)} \left(\mathbf{h}^{-\mathbf{1}^n} {\mathbf{h}'}^{x^2} W_L^x W_R^x W_O\right)^2$$

$C_{\mathbf{l},\mathbf{r}}$ is a commitment to $\mathbf{l}$ and $\mathbf{r}$ in $(\mathbf{g}, \mathbf{h})$

$$C = C(x) \leftarrow C_{\mathbf{l},\mathbf{r}} T_1^x \left(e^{2(\langle \mathbf{z}, \mathbf{c} \rangle + \delta(\mathbf{z}))} \mathbf{V}^{\mathbf{z}\mathbf{W}_V}\right)^{x^2} \prod_{3 \leq i \neq 6 \leq 7} T_i^{x^i}$$
$$C = \left(\mathbf{g}^{\mathbf{l}} \mathbf{h}^{\mathbf{r}} e^{\langle \mathbf{l}, \mathbf{r} \rangle} f^\sigma\right)^2$$

run the protocol on Figure **??** on input $(1, n, f, \mathbf{g}, \mathbf{h}, e, C, \ell_{??}; \mathbf{l}, \mathbf{r}, \sigma)$

Fig. 3. Succinct Argument of Diophantine-Equation Satisfiability.

Acknowledgements

This work was supported by the French ANR ALAMBIC Project (ANR-16-CE39-0006) and the EU H2020 Research and Innovation Program under Grant Agreement No. 786725 (OLYMPUS).

References

1. M. Artin. *Algebra.* Pearson, 2010.
2. R. M. Avanzi. The complexity of certain multi-exponentiation techniques in cryptography. *Journal of Cryptology*, 18(4):357–373, 2005.
3. E. H. Bareiss. Sylvester's identity and multistep integer-preserving Gaussian elimination. *Math. Comput.*, 22(103):565–578, 1968.
4. N. Bari and B. Pfitzmann. Collision-free accumulators and fail-stop signature schemes without trees. In W. Fumy, editor, *EUROCRYPT'97*, vol. 1233 of *LNCS*, pages 480–494. 1997.
5. S. Bayer and J. Groth. Zero-knowledge argument for polynomial evaluation with application to blacklists. In T. Johansson and P. Q. Nguyen, editors, *EUROCRYPT 2013*, vol. 7881 of *LNCS*, pages 646–663. 2013.
6. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In D. E. Denning, R. Pyle, R. Ganesan, R. S. Sandhu, and V. Ashby, editors, *ACM CCS 93*, pages 62–73. ACM Press, 1993.
7. I. Biehl, J. A. Buchmann, S. Hamdy, and A. Meyer. A signature scheme based on the intractability of computing roots. *Des. Codes Cryptogr.*, 25(3):223–236, 2002.
8. J. Bootle, A. Cerulli, P. Chaidos, J. Groth, and C. Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In M. Fischlin and J.-S. Coron, editors, *EUROCRYPT 2016, Part II*, vol. 9666 of *LNCS*, pages 327–357. 2016.
9. F. Boudot. Efficient proofs that a committed number lies in an interval. In B. Preneel, editor, *EUROCRYPT 2000*, vol. 1807 of *LNCS*, pages 431–444. 2000.
10. B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy*, pages 315–334. IEEE Computer Society Press, 2018.
11. B. Bünz, B. Fisch, and A. Szepieniec. Transparent SNARKs from DARK compilers. In A. Canteaut and Y. Ishai, editors, *EUROCRYPT 2020, Part I*, vol. 12105 of *LNCS*, pages 677–706. 2020.
12. J. Camenisch and M. Stadler. Efficient group signature schemes for large groups (extended abstract). In B. S. Kaliski Jr., editor, *CRYPTO'97*, vol. 1294 of *LNCS*, pages 410–424. 1997.
13. M. Chase, C. Ganesh, and P. Mohassel. Efficient zero-knowledge proof of algebraic and non-algebraic statements with applications to privacy preserving credentials. In M. Robshaw and J. Katz, editors, *CRYPTO 2016, Part III*, vol. 9816 of *LNCS*, pages 499–530. 2016.
14. G. Couteau, T. Peters, and D. Pointcheval. Removing the strong RSA assumption from arguments over the integers. In J. Coron and J. B. Nielsen, editors, *EUROCRYPT 2017, Part II*, vol. 10211 of *LNCS*, pages 321–350. 2017.
15. I. Damgård and E. Fujisaki. A statistically-hiding integer commitment scheme based on groups with hidden order. In Y. Zheng, editor, *ASIACRYPT 2002*, vol. 2501 of *LNCS*, pages 125–142. 2002.

16. R. del Pino, V. Lyubashevsky, and G. Seiler. Short discrete log proofs for FHE and ring-LWE ciphertexts. In D. Lin and K. Sako, editors, *PKC 2019, Part I*, vol. 11442 of *LNCS*, pages 344–373. 2019.
17. A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *CRYPTO'86*, vol. 263 of *LNCS*, pages 186–194. 1987.
18. P.-A. Fouque and G. Poupard. On the security of RDSA. In E. Biham, editor, *EUROCRYPT 2003*, vol. 2656 of *LNCS*, pages 462–476. 2003.
19. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In B. S. Kaliski Jr., editor, *CRYPTO'97*, vol. 1294 of *LNCS*, pages 16–30. 1997.
20. J. v. z. Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, USA, 3rd edition, 2013.
21. J. Groth. Non-interactive zero-knowledge arguments for voting. In J. Ioannidis, A. Keromytis, and M. Yung, editors, *ACNS 05*, vol. 3531 of *LNCS*, pages 467–482. 2005.
22. J. Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In X. Lai and K. Chen, editors, *ASIACRYPT 2006*, vol. 4284 of *LNCS*, pages 444–459. 2006.
23. J. Groth. Linear algebra with sub-linear zero-knowledge arguments. In S. Halevi, editor, *CRYPTO 2009*, vol. 5677 of *LNCS*, pages 192–208. 2009.
24. J. Groth and M. Kohlweiss. One-out-of-many proofs: Or how to leak a secret and spend a coin. In E. Oswald and M. Fischlin, editors, *EUROCRYPT 2015, Part II*, vol. 9057 of *LNCS*, pages 253–280. 2015.
25. J. Groth, R. Ostrovsky, and A. Sahai. Perfect non-interactive zero knowledge for NP. In S. Vaudenay, editor, *EUROCRYPT 2006*, vol. 4004 of *LNCS*, pages 339–358. 2006.
26. L. C. Guillou and J.-J. Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both trasmission and memory. In C. G. Günther, editor, *EUROCRYPT'88*, vol. 330 of *LNCS*, pages 123–128. 1988.
27. H. Lipmaa. On diophantine complexity and statistical zero-knowledge arguments. In C.-S. Laih, editor, *ASIACRYPT 2003*, vol. 2894 of *LNCS*, pages 398–415. 2003.
28. Y. V. Matiyasevich. Enumerable sets are diophantine. *Sov. Math., Dokl.*, 11:354–358, 1970.
29. T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In J. Feigenbaum, editor, *CRYPTO'91*, vol. 576 of *LNCS*, pages 129–140. 1992.
30. C.-P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.
31. T. Skolem. *Diophantische Gleichungen*. Ergebnisse der Mathematik und ihrer Grenzgebiete. New York, Chelsea Pub. Co., 1950.
32. A. Slinko. A generalization of Komlos theorem on random matrices. *New Zealand J. Math.*, 30(1):81–86, 2001.
33. M. Stadler. Publicly verifiable secret sharing. In U. M. Maurer, editor, *EUROCRYPT'96*, vol. 1070 of *LNCS*, pages 190–199. 1996.
34. P. Towa and D. Vergnaud. Succinct diophantine-satisfiability arguments. Cryptology ePrint Archive, Report 2020/682, 2020. `https://eprint.iacr.org/2020/682`.