Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments

Helger Lipmaa

Institute of Computer Science, University of Tartu, Estonia

Abstract. In 2010, Groth constructed the only previously known sublinear-communication NIZK circuit satisfiability argument in the common reference string model. We optimize Groth's argument by, in particular, reducing both the CRS length and the prover's computational complexity from quadratic to quasilinear in the circuit size. We also use a (presumably) weaker security assumption, and have tighter security reductions. Our main contribution is to show that the complexity of Groth's basic arguments is dominated by the quadratic number of monomials in certain polynomials. We collapse the number of monomials to quasilinear by using a recent construction of progression-free sets.

Keywords. Additive combinatorics, bilinear pairings, circuit satisfiability, non-interactive zero-knowledge, progression-free sets.

1 Introduction

By using a zero-knowledge [GMR85] proof, a prover can convince a verifier that some statement is true without leaking any side information. Due to the wide applications of zero-knowledge, it is of utmost importance to construct efficient zero-knowledge proofs. *Non-interactive zero-knowledge* (NIZK) proofs can be generated once can be verified many times by different verifiers and are thus useful in applications like e-voting.

NIZK proofs (or arguments, that is, computationally sound proofs) cannot be constructed in the plain model (that is, without random oracles or any trusted setup assumptions). Blum, Feldman and Micali showed in [BFM88] how to construct NIZK proofs in the common reference string (CRS) model. During the last years, a substantial amount of research has been done towards constructing efficient NIZK proofs (and arguments). Since the communication complexity and the verifier's computational complexity are arguably more important than the prover's computational complexity (again, an NIZK proof/argument is generated once but can be verified many times), a special effort has been made to minimize these two parameters.

One related research direction is to construct efficient NIZK proofs for $\mathbf{NP}$-complete languages. Given an efficient NIZK proof for a $\mathbf{NP}$-complete language, one can hope to construct NIZK proofs of similar complexity for the whole $\mathbf{NP}$ either by reduction or implicitly or explicitly using the developed techniques. In some NIZK proofs for the $\mathbf{NP}$-complete problem circuit satisfiability (Circuit-SAT), see Tbl. 1, the communication complexity is sublinear in the circuit size. Micali [Mic94] proposed polylogarithmic-communication NIZK *arguments* for all $\mathbf{NP}$-languages, but they are based on the PCP theorem (making them computationally unattractive) and on the random oracle model. Another NIZK argument for Circuit-SAT, proposed by Groth in 2009 [Gro09], is also based on the random oracle model. (Without random oracles, this argument takes $\log_2 |C| + 5$ rounds, where $|C|$ is the circuit size. Seo [Seo11] somewhat reduced the number of rounds in this argument.) It is well-known [CGH98] that some functionalities are secure in the random oracle model and insecure in the plain model. In particular, Goldwasser and Kalai [GK03] designed a signature scheme, built by using the Fiat-Shamir heuristics (which is predominant in the construction of NIZK arguments in the random oracle model), that is secure in the random oracle model but insecure when instantiated with any "real" hash function. As a safeguard, it is important to design efficient NIZK proofs and arguments that do not rely on the random oracles. Given a fully-homomorphic cryptosystem [Gen09], one can construct efficient NIZK *proofs* for all $\mathbf{NP}$-languages in communication that is linear to the witness size [Gro11]. However, since the witness size can be linear in the circuit size, in the worst case the corresponding NIZK proofs are not sublinear.

In 2010, Groth [Gro10] proposed the first (worst-case) sublinear-communication NIZK Circuit-SAT argument in the CRS model. First, he constructed two basic arguments for Hadamard product (the prover knows how to open commitments A, B and C to three tuples $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ of dimension n, such that $a_i b_i = c_i$ for $i \in [n]$) and permutation (the prover knows how to open commitments A and B to two tuples $\boldsymbol{a}$ and $\boldsymbol{b}$ of dimension n, such that $a_{\varrho(i)} = b_i$ for $i \in [n]$). Groth's Circuit-SAT argument can then be seen as a program in a program language that has two primitive instructions, for Hadamard product and permutation. Some of the public permutations depend

	CRS length	Argument length	Prover comp.	Verifier comp.
Random-oracle based arguments				
[Gro09]	$O(\|C\|^{\frac{1}{2}})G$	$O(\|C\|^{\frac{1}{2}})G$	$O(\|C\|)M$	$O(\|C\|)M$
Knowledge-assumption based arguments from [Gro10]				
$m = 1$	$\Theta(\|C\|^2)G$	$42G$	$\Theta(\|C\|^2)E$	$\Theta(\|C\|)M + \Theta(1)P$
$m = n^{\frac{1}{3}}$	$\Theta(\|C\|^{\frac{2}{3}})G$	$\Theta(\|C\|^{\frac{2}{3}})G$	$\Theta(\|C\|^{\frac{4}{3}})E$	$\Theta(\|C\|)M + \Theta(\|C\|^{\frac{2}{3}})P$
Knowledge-assumption based arguments from the current paper				
$m = 1$	$\|C\|^{1+o(1)}G$	$39G$	$\Theta(\|C\|^2)A + \|C\|^{1+o(1)}E$	$(8\|C\| + 8)M + 62P$
$m = n^{\frac{1}{3}}$	$\|C\|^{\frac{1}{3}+o(1)}G$	$\Theta(\|C\|^{\frac{2}{3}})G$	$\Theta(\|C\|^{\frac{4}{3}})A + \|C\|^{1+o(1)}E$	$\Theta(\|C\|)M + \Theta(\|C\|^{\frac{2}{3}})P$
$m = n^{\frac{1}{2}}$	$\|C\|^{\frac{1}{2}+o(1)}G$	$\Theta(\|C\|^{\frac{1}{2}})G$	$\Theta(\|C\|^{\frac{3}{2}})A + \|C\|^{1+o(1)}E$	$\Theta(\|C\|)M + \Theta(\|C\|^{\frac{1}{2}})P$

Table 1. Comparison of NIZK Circuit-SAT arguments with (worst-case) sublinear argument size. Note that the summary length of the CRS and the argument corresponds to the zap length. $|C|$ is the size of circuit, G corresponds to 1 group element and $A/M/E/P$ corresponds to 1 addition/multiplication/exponentiation/pairing

on the circuit, while the secret input tuples of the basic arguments depend on the values, assigned to the input and output wires of all gates according to a satisfying assignment. The basic arguments then show that this wire assignment is internally consistent and corresponds indeed to an satisfying input assignment. For example, Groth used one permutation argument to verify that all input wires of all gates have been assigned the same values as the corresponding output values of their predecessor gates.

In the basic variant of Groth's pairing-based Circuit-SAT argument, see Tbl. 1, the argument has $\Theta(1)$ group elements, but on the other hand the CRS has $\Theta(|C|)^2$ group elements, and the prover's computational complexity is dominated by $\Theta(|C|^2)$ bilinear-group exponentiations. A balanced version of Groth's argument has the CRS and argument of $\Theta(|C|^{2/3})$ group elements and prover's computational complexity dominated by $\Theta(|C|^{4/3})$ exponentiations. (See [Gro10] for more details on balancing. Basically, one applies basic arguments on length-m inputs, $m < n$, n/m times in parallel.)

We propose a new Circuit-SAT argument (see Sect. 3 for a description of the new techniques, and subsequent sections for the actual argument) that is strongly related to Groth's argument, but improves upon every step. We first propose more efficient basic arguments. We then use them to construct a (slightly shorter) new Circuit-SAT argument. In the basic variant, while the argument is again $\Theta(1)$ group elements, it is one commitment and one Hadamard product argument shorter. Moreover, in Groth's argument, every commitment consisted of 3 group elements while every basic argument consisted of 2 group elements. In the new argument, most of the commitments consist of 2 group elements. Thus, we saved 3 group elements, reducing the argument size from 42 to 39 group elements, even taking into account that the new permutation argument has higher communication complexity (12 instead of 5 group elements) than that of [Gro10].

A balanced version of the new argument achieves the combined CRS and argument of $\Theta(|C|^{1/2+o(1)})$ group elements. In App. M, we describe a zap [DN00] for Circuit-SAT that has communication complexity of $|C|^{1/2+o(1)}$ group elements, while Groth's zap from [Gro10] has the communication complexity of $\Theta(|C|^{2/3})$ group elements. We also use much more efficient asymmetric pairings instead of symmetric ones, a (presumably) weaker security assumption (Power Symmetric Discrete Logarithm instead of Power Computational Diffie-Hellman), and have more precise security reductions. The basic version of the new Circuit-SAT argument is more communication-efficient than any prior-art random-oracle based NIZK argument, and it also has a smaller prover's computational complexity than [Mic94].

Our main contribution is to note that the complexity of Groth's basic arguments is correlated to the number of monomials of a certain polynomial. In [Gro10], this polynomial has $\Theta(n^2)$ monomials, where $n = 2|C| + 1$. We show that one can "collapse" the $\Theta(n^2)$ monomials to $\Theta(N)$ monomials, where N is such that $[N]$ has a progression-free subset (that is, a subset that does not contain arithmetic progressions of length 3) of odd integers of cardinality n. By a recent breakthrough of Elkin [Elk11], $N = O(n \cdot 2^{2\sqrt{2(2+\log_2 n)}}) = n^{1+o(1)}$. See Sect. 3 for further elaboration on our techniques.

Thus, one can build an argument of $\Theta(1)$ group elements for every language in $\mathbf{NP}$, by reducing the task at hand to a Circuit-SAT instance. Obviously, one can often design more efficient tailor-made protocols, see [LZ11,CLZ12] for some follow-up work. In particular, [CLZ12] used our basic arguments to construct a non-interactive range proof with communication of $\Theta(1)$ group elements, while [LZ11] used our techniques to design a new basic argument to construct a non-interactive shuffle. (See [CLs10] for a previous use of additive combinatorics in the construction of zero-knowledge proofs.)

2 Preliminaries

Let $[n] = \{1, 2, \dots, n\}$. Let S_n be the set of permutations from $[n]$ to $[n]$. Let $\boldsymbol{a} = (a_1, \dots, a_n)$. Let $\boldsymbol{a} \circ \boldsymbol{b}$ denote the Hadamard (entry-wise) product of $\boldsymbol{a}$ and $\boldsymbol{b}$, that is, if $\boldsymbol{c} = \boldsymbol{a} \circ \boldsymbol{b}$, then $c_i = a_i b_i$ for $i \in [n]$. If $y = h^x$, then $\log_h y := x$. Let κ be the security parameter. If $0 < \lambda_1 < \cdots < \lambda_i < \cdots < \lambda_n = \text{poly}(\kappa)$. then $\Lambda = (\lambda_1, \dots, \lambda_n) \subset \mathbb{Z}$ is an *(n, κ)-nice tuple*. We abbreviate probabilistic polynomial-time as PPT. If Λ_1 and Λ_2 are subsets of some additive group ($\mathbb{Z}$ or $\mathbb{Z}_p$ in this paper), then $\Lambda_1 + \Lambda_2 = \{\lambda_1 + \lambda_2 : \lambda_1 \in \Lambda_1 \wedge \lambda_2 \in \Lambda_2\}$ is their *sum set* and $\Lambda_1 - \Lambda_2 = \{\lambda_1 - \lambda_2 : \lambda_1 \in \Lambda_1 \wedge \lambda_2 \in \Lambda_2\}$ is their *difference set* [TV06]. If Λ is a set, then $k\Lambda = \{\lambda_1 + \cdots + \lambda_k : \lambda_i \in \Lambda\}$ is an *iterated sumset*, $k \cdot \Lambda = \{k\lambda : \lambda \in \Lambda\}$ is a *dilation* of Λ, and $2\hat{}\Lambda = \{\lambda_1 + \lambda_2 : \lambda_1 \in \lambda \wedge \lambda_2 \in \Lambda \wedge \lambda_1 \neq \lambda_2\} \subseteq \Lambda + \Lambda$ is a *restricted sumset*. (See [TV06].)

Let $\mathcal{G}_{\mathsf{bp}}(1^\kappa)$ be a bilinear group generator that outputs a description of a bilinear group $\mathsf{gk} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}_{\mathsf{bp}}(1^\kappa)$, such that p is a κ-bit prime, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are multiplicative cyclic groups of order p, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a bilinear map (pairing) such that $\forall a, b \in \mathbb{Z}$ and $g_t \in \mathbb{G}_t$, $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$. If g_t generates $\mathbb{G}_t$ for $t \in \{1, 2\}$, then $\hat{e}(g_1, g_2)$ generates $\mathbb{G}_T$. Deciding the membership in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, group operations, the pairing $\hat{e}$, and sampling the generators are efficient, and the descriptions of the groups and group elements are $O(\kappa)$ bit long each. Well-chosen asymmetric pairings (with no efficient isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$) are much more efficient than symmetric pairings (where $\mathbb{G}_1 = \mathbb{G}_2$). For $\kappa = 128$, the current recommendation is to use an optimal (asymmetric) Ate pairing [HSV06] over a subclass of Barreto-Naehrig curves [BN05,PSNB11]. In that case, at security level of $\kappa = 128$, an element of $\mathbb{G}_1$/$\mathbb{G}_2$/$\mathbb{G}_T$ can be represented in respectively 512/256/3072 bits.

A (tuple) commitment scheme $(\mathcal{G}_{\mathsf{com}}, \mathcal{C}\mathrm{om})$ in a bilinear group consists of two PPT algorithms: a randomized CRS generation algorithm $\mathcal{G}_{\mathsf{com}}$, and a randomized commitment algorithm $\mathcal{C}\mathrm{om}$. Here, $\mathcal{G}^t_{\mathsf{com}}(1^\kappa, n)$, $t \in \{1, 2\}$, produces a CRS ck_t, and $\mathcal{C}\mathrm{om}^t(\mathsf{ck}_t; \boldsymbol{a}; r)$, with $\boldsymbol{a} = (a_1, \dots, a_n)$, outputs a commitment value A in $\mathbb{G}_t$ (or in $\mathbb{G}_t^b$ for some $b > 1$). We open $\mathcal{C}\mathrm{om}^t(\mathsf{ck}_t; \boldsymbol{a}; r)$ by outputting $\boldsymbol{a}$ and r.

A commitment scheme $(\mathcal{G}_{\mathsf{com}}, \mathcal{C}\mathrm{om})$ is *computationally binding in group* $\mathbb{G}_t$, if for every non-uniform PPT adversary $\mathcal{A}$ and positive integer $n = \text{poly}(\kappa)$, the probability

$$\Pr\left[\begin{array}{l}\mathsf{ck}_t \leftarrow \mathcal{G}^t_{\mathsf{com}}(1^\kappa, n), (\boldsymbol{a_1}, r_1, \boldsymbol{a_2}, r_2) \leftarrow \mathcal{A}(\mathsf{ck}_t) : \\ (\boldsymbol{a_1}, r_1) \neq (\boldsymbol{a_2}, r_2) \wedge \mathcal{C}\mathrm{om}^t(\mathsf{ck}_t; \boldsymbol{a_1}; r_1) = \mathcal{C}\mathrm{om}^t(\mathsf{ck}_t; \boldsymbol{a_2}; r_2)\end{array}\right]$$

is negligible in κ. A commitment scheme $(\mathcal{G}_{\mathsf{com}}, \mathcal{C}\mathrm{om})$ is *perfectly hiding in group* $\mathbb{G}_t$, if for any positive integer $n = \text{poly}(\kappa)$ and $\mathsf{ck}_t \in \mathcal{G}^t_{\mathsf{com}}(1^\kappa, n)$ and any two messages $\boldsymbol{a_1}, \boldsymbol{a_2}$, the distributions $\mathcal{C}\mathrm{om}^t(\mathsf{ck}_t; \boldsymbol{a_1}; \cdot)$ and $\mathcal{C}\mathrm{om}^t(\mathsf{ck}_t; \boldsymbol{a_2}; \cdot)$ are equal.

A trapdoor commitment scheme has three additional efficient algorithms: (a) A trapdoor CRS generation algorithm inputs t, n and 1^κ, and outputs a CRS ck^* (that has the same distribution as $\mathcal{G}^t_{\mathsf{com}}(1^\kappa, n)$) and a trapdoor td, (b) a randomized trapdoor commitment that takes ck^* and a randomizer r as inputs and outputs the value $\mathcal{C}\mathrm{om}^t(\mathsf{ck}^*; \boldsymbol{0}; r)$, and (c) a trapdoor opening algorithm that takes ck^*, td, $\boldsymbol{a}$ and r as an input and outputs an r' such that $\mathcal{C}\mathrm{om}^t(\mathsf{ck}^*; \boldsymbol{0}; r) = \mathcal{C}\mathrm{om}^t(\mathsf{ck}^*; \boldsymbol{a}; r')$.

Let $\mathcal{R} = \{(C, w)\}$ be an efficiently computable binary relation such that $|w| = \text{poly}(|C|)$. Here, C is a statement, and w is a witness. Let $\mathcal{L} = \{C : \exists w, (C, w) \in \mathcal{R}\}$ be an $\mathbf{NP}$-language. Let n be some fixed input length $n = |C|$. For fixed n, we have a relation $\mathcal{R}_n$ and a language $\mathcal{L}_n$. A *non-interactive argument* for $\mathcal{R}$ consists of the following PPT algorithms: a common reference string (CRS) generator $\mathcal{G}_{\mathsf{crs}}$, a prover $\mathcal{P}$, and a verifier $\mathcal{V}$. For $\mathsf{crs} \leftarrow \mathcal{G}_{\mathsf{crs}}(1^\kappa, n)$, $\mathcal{P}(\mathsf{crs}; C, w)$ produces an argument ψ. The verifier $\mathcal{V}(\mathsf{crs}; C, \psi)$ outputs either 1 (accept) or 0 (reject).

A non-interactive argument $(\mathcal{G}_{\mathsf{crs}}, \mathcal{P}, \mathcal{V})$ is *perfectly complete*, if $\forall n = \text{poly}(\kappa)$,

$$\Pr[\mathsf{crs} \leftarrow \mathcal{G}_{\mathsf{crs}}(1^\kappa, n), (C, w) \leftarrow \mathcal{R}_n : \mathcal{V}(\mathsf{crs}; C, \mathcal{P}(\mathsf{crs}; C, w)) = 1] = 1 \enspace .$$

A non-interactive argument $(\mathcal{G}_{\mathsf{crs}}, \mathcal{P}, \mathcal{V})$ is *(adaptively) computationally sound*, if for all non-uniform PPT adversaries $\mathcal{A}$ and all $n = \text{poly}(\kappa)$, the probability

$$\Pr[\mathsf{crs} \leftarrow \mathcal{G}_{\mathsf{crs}}(1^\kappa, n), (C, \psi) \leftarrow \mathcal{A}(\mathsf{crs}) : C \notin \mathcal{L} \wedge \mathcal{V}(\mathsf{crs}; C, \psi) = 1]$$

is negligible in κ. The soundness is adaptive, that is, the adversary sees the CRS before producing the statement C. A non-interactive argument $(\mathcal{G}_{\mathsf{crs}}, \mathcal{P}, \mathcal{V})$ is *perfectly witness-indistinguishable*, if for all $n = \operatorname{poly}(\kappa)$, if $\mathsf{crs} \in \mathcal{G}_{\mathsf{crs}}(1^\kappa, n)$ and $((C, w_0), (C, w_1)) \in \mathcal{R}_n^2$, then the distributions $\mathcal{P}(\mathsf{crs}; C, w_0)$ and $\mathcal{P}(\mathsf{crs}; C, w_1)$ are equal.

A non-interactive argument $(\mathcal{G}_{\mathsf{crs}}, \mathcal{P}, \mathcal{V})$ is *perfectly zero-knowledge*, if there exists a PPT simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$, such that for all stateful non-uniform PPT adversaries $\mathcal{A}$ and $n = \operatorname{poly}(\kappa)$ (with td being the *simulation trapdoor*),

$$\Pr\left[\begin{array}{l}\mathsf{crs} \leftarrow \mathcal{G}_{\mathsf{crs}}(1^\kappa, n),\\ (C, w) \leftarrow \mathcal{A}(\mathsf{crs}),\\ \psi \leftarrow \mathcal{P}(\mathsf{crs}; C, w) :\\ (C, w) \in \mathcal{R}_n \wedge \mathcal{A}(\psi) = 1\end{array}\right] = \Pr\left[\begin{array}{l}(\mathsf{crs}; \mathsf{td}) \leftarrow \mathcal{S}_1(1^\kappa, n),\\ (C, w) \leftarrow \mathcal{A}(\mathsf{crs}),\\ \psi \leftarrow \mathcal{S}_2(\mathsf{crs}; C, \mathsf{td}) :\\ (C, w) \in \mathcal{R}_n \wedge \mathcal{A}(\psi) = 1\end{array}\right] .$$

3 Our Techniques

We will first give a more precise overview of Groth's Hadamard product and permutation arguments [Gro10], followed by a short description of our own main contribution. For the sake of simplicity, we will make several simplifications (like the use of symmetric pairings) during this discussion.

Groth uses an additively homomorphic tuple commitment scheme that allows one to commit to a long tuple, while the commitment itself is short. The best known such commitment scheme is the extended Pedersen commitment scheme in a multiplicative cyclic group of order p and a generator g, where the commitment of a tuple $\boldsymbol{a} = (a_1, \dots, a_n)$ with randomness r_a is equal to $\mathcal{C}\mathrm{om}(\boldsymbol{a}; r_a) := g^{r_a} \cdot \prod g_i^{a_i}$. Here, one usually chooses n random secrets $x_i \leftarrow \mathbb{Z}_p$, and then sets $g_i \leftarrow g^{x_i}$. Following [GJM02], Groth [Gro10] chooses a single random secret $x \leftarrow \mathbb{Z}_p$ and then sets $g_i \leftarrow g^{x^i}$. In this case, the commitment

$$\mathcal{C}\mathrm{om}(\boldsymbol{a}; r_a) := g^{r_a} \cdot \prod_{i=1}^{n} g_i^{a_i} = g^{r_a + \sum_{i=1}^{n} a_i x^i}$$

can be seen as a lifted polynomial $r_a + \sum_{i=1}^n a_i x^i$ in x, that the committer (who does not know x) computes from n given values $g_i = g^{x^i}$. The first obvious benefit of this commitment scheme is that it has a shorter secret (1 element instead of n elements).

Groth's Hadamard product argument, where the prover aims to convince the verifier that the opening of $C = \mathcal{C}\mathrm{om}(\boldsymbol{c}; r_c)$ is equal to the Hadamard product of the openings of $A = \mathcal{C}\mathrm{om}(\boldsymbol{a}; r_a)$ and $B = \mathcal{C}\mathrm{om}(\boldsymbol{b}; r_b)$ (that is, $a_i b_i \equiv c_i \pmod{p}$ for $i \in [n]$), is constructed as follows. Let $A = g^{r_a} \cdot \prod_{i=1}^n g_i^{a_i}$ be a commitment of $\boldsymbol{a}$ and $B = g^{r_b} \cdot \prod_{i=1}^n g_i^{b_i}$ be a commitment of $\boldsymbol{b}$ by using the generator tuple $(g_1, \dots, g_n)$. Let $C = g^{r_c} \cdot \prod_{i=1}^n g_{i(n+1)}^{c_i}$ be a commitment of $\boldsymbol{b}$ and $D = \prod_{i=1}^n g_{i(n+1)}$ be a commitment of $\mathbf{1} = (1, \dots, 1)$ by using a different generator tuple $(g_{n+1}, \dots, g_{n(n+1)})$.

Groth's Hadamard product argument is based around the verification equation

$$\hat{e}(A, B) = \hat{e}(C, D) \cdot \hat{e}(\psi, g) \tag{1}$$

that (analogously to the Groth-Sahai proofs [GS08], though the latter only considers the much simpler case $n = 1$) can be seen as a mapping of the required equality $\boldsymbol{a} \circ \boldsymbol{b} = \boldsymbol{c} \circ \mathbf{1}$ to another algebraic domain, with ψ compensating for the use of a randomized commitment scheme. One gets that $\hat{e}(A, B)/\hat{e}(C, D)$ is equal to $\hat{e}(g, g)^{F(x)}$, where $F(x) = (r_a + \sum_{i=1}^n a_i x^i) \cdot (r_b + \sum_{i=1}^n b_i x^{i(n+1)}) - (r_c + \sum_{i=1}^n c_i x^i) \cdot (\sum_{i=1}^n x^{i(n+1)})$ is the sum of two formal polynomials in x, $F(x) = F_{\mathsf{con}}(x) + F_\psi(x)$, where $F_{\mathsf{con}}(x) = \sum_{i=1}^n (a_i b_i - c_i) x^{i(n+2)}$ is a *constraint polynomial*, spanned by the powers of x from $\Lambda_{\mathsf{con}} = \{i(n+2) : i \in [n]\}$, and

$$F_\psi(x) = r_a r_b + r_b \sum_{i=1}^{n} a_i x^i + \sum_{i=1}^{n} (r_a b_i - r_c) x^{i(n+1)} + \sum_{i=1}^{n} \sum_{\substack{j=1\\ j \neq i}}^{n} (a_i b_j - c_i) x^{i + j(n+1)}$$

is an *argument polynomial*, spanned by the powers of x from $\Lambda_\psi = \{0\} \cup [n] \cup \{i(n+1) : i \in [n]\} \cup \{i + j(n+1) : i, j \in [n] \wedge i \neq j\}$. One coefficient of $F_{\mathsf{con}}(x)$ corresponds to one constraint $a_i b_i = c_i$ that the honest prover has to satisfy, and is 0 if this constraint is true. Thus, all coefficients of F_{con} are equal to 0 iff the prover is honest.

By using homomorphic properties of the commitment scheme, the prover constructs the argument $\psi = g^{F_\psi(x)}$ as $\psi = g^{r_a r_b} \cdot \dots \cdot \prod_{i=1}^n \prod_{j=1: i \neq j}^n g_{i+j(n+1)}^{a_i b_j - c_i}$. This can be done, since the prover — who knows how to open the commitments but does not know the secret x — knows all coefficients $r_a r_b$, …, $a_i b_j - c_i$. He also knows the generators g, …, $g_{i+j(n+1)}$ if the $\Theta(n^2)$ generators g_ℓ, for $\ell \in \Lambda_\psi$, are included to the CRS. Thus, the CRS has $\Theta(n^2)$ group elements and the computational complexity of the prover is $\Theta(n^2)$ bilinear-group exponentiations. On the other hand, the verifier's computational complexity is $\Theta(1)$ pairings, since she only has to check Eq. (1).

For the soundness, one needs that when $a_i b_i \neq c_i$ for some $i \in [n]$, then a satisfying ψ cannot be computed from the elements g^{x^ℓ} that are in the CRS; otherwise, a dishonest prover would be able to compute a satisfying argument. This means that for $i \in [n]$, $g^{x^{i(n+2)}}$ should not belong to the CRS. To be certain that this is true, one needs

(a) that g^{x^ℓ} is in the CRS for values $\ell \in \Lambda_\psi$ but if $\ell \in \Lambda_{\mathsf{con}}$, then g^{x^ℓ} does not belong to the CRS (elements from $2 \cdot \Lambda \setminus \hat{\Lambda}$ are allowed),

(b) an appropriate security assumption that states that computing g^{F_ψ} for $F_\psi = \sum_{\ell \in \Lambda_\psi} \mu_\ell x^\ell$ is only possible if one knows all values g^{x^ℓ} for $\ell \in \Lambda_\psi$, and

(c) that $\Lambda_{\mathsf{con}} \cap \Lambda_\psi = \emptyset$. (This is also a prerequisite for (a).)

One can guarantee (a) by the choice of the CRS. But also (c) is clearly true, since Λ_{con} and Λ_ψ do not intersect.

To finish off the whole argument, one has to define an appropriate security assumption for (b). Since constructing sublinear NIZK arguments is known to be impossible under standard assumptions (see Sect. 2), one of the underlying assumptions is a knowledge assumption (PKE assumption, as in [Gro10], see Sect. 5). The whole argument will become (slightly!) more complex since all commitments and arguments also have to include a knowledge component.

Groth's permutation argument is based on a very similar idea and has basically the same complexities. The only major difference is that if the permutation is a part of the prover's statement, then the verifier also has to perform $\Theta(n)$ bilinear-group multiplications. Since Groth's Circuit-SAT argument consists of a very small (< 10) number of Hadamard product and permutation arguments, then it just inherits the complexities of the basic arguments, as also seen from Tbl. 1, where, in the basic variation, $|C| = n$ and thus the CRS has $\Theta(|C|^2)$ group elements, the argument length is 42 group elements, the prover's computational complexity is $\Theta(|C|^2)$ exponentiations, and the prover's computational complexity is dominated by $\Theta(|C|)$ bilinear-group multiplications.

Groth's Circuit-SAT argument has several sub-optimal properties that are all inherited from the basic arguments. While it has succinct communication and efficient verification, its CRS of $\Theta(|C|^2)$ group elements and prover's computation of $\Theta(|C|^2)$ exponentiations (in the basic variant) seriously limit applicability. Recall that here $n = 2|C| + 1$. A smaller problem is the use of different generators $(g_1, \dots, g_n)$ and $(g_{n+1}, \dots, g_{n(n+1)})$ while committing to different elements.

We note that F_{con} has n monomials (1 per every constraint $a_i b_i = c_i$ that a honest prover must satisfy). On the other hand, F_ψ has $\Theta(n^2)$ distinct — since $i_1 + j_1(n+1) \neq i_2 + j_2(n+1)$ if $i_1, j_1, i_2, j_2 \in [n]$ and $(i_1, j_1) \neq (i_2, j_2)$ — monomials. The number of those monomials is the only reason why the CRS has $\Theta(n^2)$ group elements and the prover has to perform $\Theta(n^2)$ bilinear-group exponentiations.

We now show how to collapse many of the unnecessary monomials into one, so that the full argument still remains secure, obtaining a polynomial $F_\psi(x)$ that has only $n^{1+o(1)}$ monomials. First, we generalize the underlying commitment scheme. We still choose a single $x \leftarrow \mathbb{Z}_p$ and set $g_i \leftarrow g^{x^i}$, but we allow the indexes of n generators $(g_{\lambda_1}, \dots, g_{\lambda_n})$, that are used to commit, to actually depend on the concrete argument — with the main purpose to be able to obtain as small Λ_ψ as possible, while still guaranteeing that $F_{\mathsf{con}} = 0$ iff the prover is honest, and that $\Lambda_{\mathsf{con}} \cap \Lambda_\psi = \emptyset$. Assume that $\Lambda = (\lambda_1, \dots, \lambda_n)$ is an (n, κ)-nice tuple of integers, so $\lambda_n = \max_i \lambda_i$. Thus,

$$\mathcal{C}\mathrm{om}(\boldsymbol{a}; r_a) := g^{r_a} \prod_{i=1}^{n} g_{\lambda_i}^{a_i} = g^{r_a + \sum_{i=1}^{n} a_i x^{\lambda_i}} \quad .$$

The polynomial $r_a + \sum_{i=1}^{n} a_i x^{\lambda_i}$ has degree (up to) λ_n, but it only has (up to) $n + 1$ non-zero monomials. We now start again with the verification equation Eq. (1), but this time we assume that all A, B, C and D have been committed by using the same set of generators $(g_{\lambda_1}, \dots, g_{\lambda_n})$. Since $F(x) = (r_a + \sum_{i=1}^{n} a_i x^{\lambda_i})(r_b +$

$\sum_{i=1}^n b_i x^{\lambda_i}) - (r_c + \sum_{i=1}^n c_i x^{\lambda_i})(\sum_{i=1}^n x^{\lambda_i})$, we get that $F(x) = F_{\mathsf{con}}(x) + F_\psi(x)$, where

$$F_{\mathsf{con}}(x) = \sum_{i=1}^{n} (a_i b_i - c_i) x^{2\lambda_i} \ , \tag{2}$$

$$F_\psi(x) = r_a r_b + \sum_{i=1}^{n} (r_a b_i + r_b a_i - r_c) x^{\lambda_i} + \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} (a_i b_j - c_i) x^{\lambda_i + \lambda_j} \ . \tag{3}$$

Here, the powers corresponding to nonzero coefficients belong either to the set $\Lambda_{\mathsf{con}} = 2 \cdot \Lambda := \{2\lambda_i : i \in [n]\}$ or to the set $\Lambda_\psi = \hat{\Lambda} := \{0\} \cup \Lambda \cup 2\hat{\ }\Lambda$, where $2\hat{\ }\Lambda := \{\lambda_i + \lambda_j : i, j \in [n] \wedge i \neq j\}$.

If the prover is honest (that is, $a_i b_i - c_i = 0$ for all i), then the coefficients $a_i b_i - c_i$ corresponding to the powers in the set $2 \cdot \Lambda$ are equal to 0. Therefore, an honest prover can compute the argument $\psi = g^{F_\psi(x)}$ as $g^{\sum_{\ell \in \hat{\Lambda}} \mu_\ell x^\ell} = \prod_{\ell \in \hat{\Lambda}} (g^{x^\ell})^{\mu_\ell}$, where the coefficients μ_ℓ are known to the prover. This means that all elements g^{x^ℓ}, $\ell \in \hat{\Lambda}$, have to belong to the CRS, and thus the CRS contains at least $|\hat{\Lambda}| < 2\lambda_n$ group elements. Recall that in [Gro10], one had to specify $\Theta(n^2)$ elements in the CRS.

For the soundness, we again need (a–c), as in the case of Groth's argument, to be true. One can again guarantee (a) by the choice of the CRS, and one has to define a reasonable security assumption (PKE assumption) for (b). Finally, achieving (c) is also relatively easy. Namely, one can guarantee that $0 \notin 2 \cdot \Lambda$ and $\Lambda \cap 2 \cdot \Lambda = \emptyset$ by choosing Λ to be a set of odd[1] integers. It is almost as easy to guarantee that $2 \cdot \Lambda \cap 2\hat{\ }\Lambda = \emptyset$ as soon as one rewrites this condition as $2\lambda_k \neq \lambda_i + \lambda_j$ for $i \neq j$, and notices that this is equivalent to requiring that no 3 elements of Λ are in an arithmetic progression. That is, Λ is a progression-free set [TV06]. Thus, it is sufficient to assume that Λ is a progression-free set of odd integers.

Recall that the CRS length (and the prover's computational complexity) depend on $|\hat{\Lambda}|$ and thus it is beneficial to have as small $|\hat{\Lambda}| < 2\lambda_n$ possible. This can be guaranteed by upper bounding λ_n, that is, by finding as small λ_n as possible such that $[\lambda_n]$ contains a progression-free subset of odd integers of cardinality n. To bound λ_n, we show in Sect. 4 (following a recent breakthrough of Elkin [Elk11]) that any range $[N] = \{1, \dots, N\}$ contains a progression-free set of odd integers of size $n = \Theta(N (\log_2 N)^{1/4} / 2^{2\sqrt{2 \log_2 N}}) = N^{1-o(1)}$, and thus one can assume that $\lambda_n = n^{1+o(1)}$. (One can obtain $\lambda_n = O(n \cdot 2^{2\sqrt{2(2+\log_2 n)}})$ by inverting a weaker version of Elkin's result.) In App. B, we give another proof of this result that, while based on Green and Wolf's exposition [GW10] of [Elk11], provides more details and is slightly sharper. In particular, Elkin's progression-free set is efficiently constructible.

Groth's permutation argument uses similar ideas for a different choice of A, B, C, and D, and thus also for a different set Λ_ψ. Unfortunately, if we use it with the new generalized commitment scheme (that is, with general Λ), we obtain the guarantee $a_{\varrho(i)} = b_i$ only if Λ is a part of the Moser-de Bruijn sequence [Mos62,dB64]. But then $\lambda_n = \Theta(n^2)$ and one ends up with a CRS of $\Theta(n^2)$ group elements. We use the following idea to get the same guarantees when Λ is an arbitrary progression-free set of odd integers. We show that if Λ is a progression-free set of odd integers, then Groth's permutation argument guarantees that $a_{\varrho(i)} = T_\Lambda(i, \varrho) \cdot b_i$, where $T_\Lambda(i, \varrho) \geq 1$ is an easily computable and public integer. We use this result to show that for some separately committed tuple $\boldsymbol{a}^*$, $a^*_{\varrho(i)} = T_\Lambda(i, \varrho) \cdot b_i$ for $i \in [n]$. We then employ an additional product argument to show that $a^*_i = T_\Lambda(\varrho^{-1}(i), \varrho) \cdot a_i$ for $i \in [n]$. Thus, $a_{\varrho(i)} = b_i$ for $i \in [n]$.

We obtain basic arguments that only use $\Theta(\lambda_n) = n^{1+o(1)}$ generators $\{g^{x^\ell} : \ell \in \hat{\Lambda}\}$. This means that the CRS has $n^{1+o(1)}$ group elements and not $\Theta(n^2)$ as in [Gro10]. In both basic arguments, the prover has to compute ψ (which takes $\Theta(n^2)$ scalar multiplications or additions in $\mathbb{Z}_p$ and $n^{1+o(1)}$ bilinear-group exponentiations). As in [Gro10], the prover's computation can be optimized even further by using efficient multi-exponentiation algorithms [Str64,Pip80]. The verifier has to only perform $\Theta(1)$ bilinear pairings. In the case of the permutation argument, she also has to compute $\Theta(n)$ bilinear-group multiplications, though the multiplications can be done offline if the permutation is fixed. Thus, the new basic arguments are considerably more efficient than Groth's.

The soundness of the new product argument is based on two assumptions, a computational assumption ($\hat{\Lambda}$-PSDL, see Sect. 5) and a knowledge assumption (Λ-PKE, see Sect. 5). Groth [Gro10] used $[an^2]$-PKE (for a constant a) and $[an^2]$-CPDH (which is a presumably stronger assumption than PSDL, see App. A). Since Λ, Λ_ψ are small subsets of $[an^2]$, then our assumptions can be expected to be somewhat weaker in general. Finally, the

[1] Oddity is not strictly required. For $\Lambda \cap 2 \cdot \Lambda = \emptyset$ to hold, one can take $\Lambda := \{(2i+1)2^{2j} : i, j \geq 0\}$, see OEIS sequence A003159. Dealing with odd integers is however almost as good.

security reduction in the proof of the product argument takes time $\Theta(t(\lambda_n))$ in our case and $\Theta(t(an^2))$ in Groth's case, where $t(m)$ is the time to factor a degree-m polynomial. See Sect. K for a detailed comparison.

4 Progression-Free Sets

A set of positive integers $\Lambda = \{\lambda_1, \dots, \lambda_n\}$ is *progression-free* [TV06], if no three elements of Λ are in an arithmetic progression, that is, $\lambda_i + \lambda_j = 2\lambda_k$ only if $i = j = k$, or equivalently, $2^\frown\Lambda \cap 2 \cdot \Lambda = \emptyset$.

Let $r_3(N)$ denote the cardinality of the largest progression-free set that belongs to $[N]$. For any $N > 1$, the set of integers in $[N]$ that have no ternary digit equal to 2 is progression-free [ET36]. If $N = 3^k$, then there are $2^N - 1$ such integers, and thus $r_3(N) = \Omega(N^{\log_3 2}) = \Omega(N^{0.63})$. Clearly, this set can be efficiently constructed. As shown by Behrend in 1946 [Beh46], this idea can be generalized to non-ternary bases, with $r_3(N) = \Omega(N/(2^{2\sqrt{2\log_2 N}} \cdot \log_2^{1/4} N))$. Behrend's result was improved in a recent breakthrough by Elkin [Elk11], who showed that $r_3(N) = \Omega(N \cdot \log_2^{1/4} N / 2^{2\sqrt{2\log_2 N}})$. We have included a proof of Elkin's result in App. B. Our proof is closely based on [GW10] but it has a sharper constant inside Ω. Moreover, our proof is much more detailed than that given in [GW10]. While both constructions employ the pigeonhole principle, Elkin's methodology can be used to compute his progression-free set in quasi-linear time $N \cdot 2^{O(\sqrt{\log N})}$, see [Elk11]. On the other hand, Bourgain [Bou98] showed that $r_3(N) = O(N \cdot (\log N / \log\log N)^{1/2})$, and recently Sanders [San11] showed that $r_3(N) = O(N \cdot (\log\log N)^5 / \log N)$. Thus, according to Behrend and Elkin, the minimal N such that $r_3(N) = n$ is $N = n^{1+o(1)}$, while according to Sanders, $N = \omega(n)$.

We need the progression-free subset to also consist of odd integers. For this, one can take Elkin's set $\Lambda = \{\lambda_1, \dots, \lambda_n\} \subset [N]$, and then use the set $2 \cdot \Lambda + 1 = \{2\lambda_1 + 1, \dots, 2\lambda_n + 1\}$. Clearly, if $\Lambda \in [n^{1+o(1)}]$ then also $2 \cdot \Lambda + 1 \in [n^{1+o(1)}]$. We provide an expository proof of the following result in App. B.

Theorem 1. *Let $r_3^{odd}(N)$ be the size of the largest progression-free set in $[N]$ that only consists of odd integers. For any n, there exists $N = n^{1+o(1)}$, such that $r_3^{odd}(N) = n$.*

5 Cryptographic Tools

In this section, we generalize the PKE assumption from [Gro10] and then define two new cryptographic assumptions, PDL and PSDL, and prove that PSDL is secure in the generic group model. After that, we proceed to describe a generalization of Groth's knowledge commitment scheme from [Gro10] and prove that it is computationally binding under the PDL assumption. Groth proved in [Gro10] that his commitment scheme is computationally binding under the (potentially stronger) CPDH assumption.

***Λ*-Power (Symmetric) Discrete Logarithm Assumption.** Let Λ be an (n, κ)-nice tuple for some $n = \mathrm{poly}(\kappa)$. We say that a bilinear group generator $\mathcal{G}_{\mathsf{bp}}$ is (n, κ)*-PDL secure in group* $\mathbb{G}_t$ for $t \in \{1, 2\}$, if for any non-uniform PPT adversary $\mathcal{A}$, $\Pr[\mathsf{gk} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}_{\mathsf{bp}}(1^\kappa), g_t \leftarrow \mathbb{G}_t \setminus \{1\}, x \leftarrow \mathbb{Z}_p : \mathcal{A}(\mathsf{gk}; (g_t^{x^\ell})_{\ell \in \{0\} \cup \Lambda}) = x]$ is negligible in κ. Similarly, we say that a bilinear group generator $\mathcal{G}_{\mathsf{bp}}$ is Λ*-PSDL secure*, if for any non-uniform PPT adversary $\mathcal{A}$,

$$\Pr\left[\begin{array}{l}\mathsf{gk} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}_{\mathsf{bp}}(1^\kappa), g_1 \leftarrow \mathbb{G}_1 \setminus \{1\}, \\ g_2 \leftarrow \mathbb{G}_2 \setminus \{1\}, x \leftarrow \mathbb{Z}_p : \mathcal{A}(\mathsf{gk}; (g_1^{x^\ell}, g_2^{x^\ell})_{\ell \in \{0\} \cup \Lambda}) = x\end{array}\right]$$

is negligible in κ. A version of P(S)DL assumption in a non pairing-based group was defined in [GJM02]. Cheon showed in [Che06] that if n is a prime divisor of $p - 1$ or $p + 1$, then the $[n]$-PDL assumption can be broken by a generic adversary in $O((\sqrt{p/n} + \sqrt{n}) \log p)$ group operations. Clearly, if the Λ-PSDL assumption is hard, then the Λ-PDL assumption is hard in both $\mathbb{G}_1$ and $\mathbb{G}_2$. Moreover, if the bilinear group generator is CPDH secure, then it is also P(S)DL secure. Therefore, by the results of [Gro10], P(S)DL holds in the generic group model. We give a direct proof of the next result in App. D.

Theorem 2. *The Λ-PSDL assumption holds in the generic group model for any (n, κ)-nice tuple Λ given that $n = \mathrm{poly}(\kappa)$. Any successful generic adversary for Λ-PSDL requires time $\Omega(\sqrt{p/\lambda_n})$ where λ_n is the largest element of Λ.*

$\varLambda$-Power Knowledge of Exponent Assumption ($\varLambda$-PKE). Abe and Fehr showed in [AF07] that no statistically zero-knowledge non-interactive argument for an $\mathbf{NP}$-complete language can have a "direct black-box" security reduction to a standard cryptographic assumption unless $\mathbf{NP} \subseteq \mathbf{P}/\mathrm{poly}$. (See also [GW11].) In fact, the soundness of NIZK arguments (for example, of an argument that a perfectly hiding commitment scheme commits to 0) is often unfalsifiable by itself. Similarly to Groth [Gro10], we will base our NIZK argument for circuit satisfiability on $\varLambda$-PKE, an explicit knowledge assumption. This assumption was proposed by Groth [Gro10] (though only for $\varLambda = [n]$) as a generalization of the KEA assumption of Damgård [Dam91] and of the KEA3 assumption of Bellare and Palacio [BP04].

Let $t \in \{1, 2\}$. For two algorithms $\mathcal{A}$ and $X_{\mathcal{A}}$, we write $(y; z) \leftarrow (\mathcal{A}||X_{\mathcal{A}})(x)$ if $\mathcal{A}$ on input x outputs y, and $X_{\mathcal{A}}$ on the same input (including the random tape of $\mathcal{A}$) outputs z. Let $\varLambda$ be an (n, κ)-nice tuple for some $n = \mathrm{poly}(\kappa)$. The bilinear group generator $\mathcal{G}_{\mathsf{bp}}$ is *$\varLambda$-PKE secure in group* $\mathbb{G}_t$ if for any non-uniform PPT adversary $\mathcal{A}$ there exists a non-uniform PPT extractor $X_{\mathcal{A}}$, such that

$$\Pr\left[\begin{array}{l}\mathsf{gk} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}_{\mathsf{bp}}(1^\kappa), g_t \leftarrow \mathbb{G}_t \setminus \{1\}, (\hat{\alpha}, x) \leftarrow \mathbb{Z}_p^2, \\ \mathsf{crs} \leftarrow (\mathsf{gk}; (g_t^{x^\ell}, g_t^{\hat{\alpha}x^\ell})_{\ell \in \{0\} \cup \varLambda}), (c, \hat{c}; r, (a_\ell)_{\ell \in \varLambda}) \leftarrow (\mathcal{A}||X_{\mathcal{A}})(\mathsf{crs}) : \\ \hat{c} = c^{\hat{\alpha}} \wedge c \neq g_t^r \cdot \prod_{\ell \in \varLambda} g_t^{a_\ell x^\ell}\end{array}\right]$$

is negligible in κ. That is, if $\mathcal{A}$ (given access to crs that for a random $\hat{\alpha}$ contains both $g_t^{x^\ell}$ and $g_t^{\hat{\alpha}x^\ell}$ iff $\ell \in \{0\} \cup \varLambda$) can produce c and $\hat{c}$ such that $\hat{c} = c^{\hat{\alpha}}$, then $X_{\mathcal{A}}$ (given access to crs and to the random coins of $\mathcal{A}$) can produce a tuple $(r, (a_\ell)_{\ell \in \varLambda})$ such that $c = g_t^r \cdot \prod_{\ell \in \varLambda} g_t^{a_\ell x^\ell}$. Groth [Gro10] proved that the $[n]$-PKE assumption holds in the generic group model; his proof can be straightforwardly modified to the general case.

New Commitment Scheme. We use the following variant of the *knowledge commitment scheme* from [Gro10] with a generalized choice of generators, defined as follows:

CRS generation: Let $\varLambda$ be an (n, κ)-nice tuple with $n = \mathrm{poly}(\kappa)$. Define $\lambda_0 = 0$. Given a bilinear group generator $\mathcal{G}_{\mathsf{bp}}$, set $\mathsf{gk} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}_{\mathsf{bp}}(1^\kappa)$. Let $g_1 \leftarrow \mathbb{G}_1 \setminus \{1\}$, $g_2 \leftarrow \mathbb{G}_2 \setminus \{1\}$, and $\hat{\alpha}, x \leftarrow \mathbb{Z}_p$. Let $t \in \{1, 2\}$. The CRS is $\mathsf{ck}_t \leftarrow (\mathsf{gk}; (g_{t,\lambda_i}, \hat{g}_{t,\lambda_i})_{i \in \{0,\dots,n\}})$, where $g_{t\ell} = g_t^{x^\ell}$ and $\hat{g}_{t\ell} = g_t^{\hat{\alpha}x^\ell}$.

Commitment: To commit to $\boldsymbol{a} = (a_1, \dots, a_n) \in \mathbb{Z}_p^n$, the committing party chooses a random $r \leftarrow \mathbb{Z}_p$, and defines

$$\mathcal{C}\mathrm{om}^t(\mathsf{ck}_t; \boldsymbol{a}; r) := (g_t^r \cdot \prod_{i=1}^n g_{t,\lambda_i}^{a_i}, \hat{g}_t^r \cdot \prod_{i=1}^n \hat{g}_{t,\lambda_i}^{a_i}) \enspace .$$

Importantly, we allow $\varLambda$ to depend on the concrete application. Let $t = 1$. Fix a commitment key ck_1 that in particular specifies $g_2, \hat{g}_2 \in \mathbb{G}_2$. A commitment $(A, \hat{A}) \in \mathbb{G}_1^2$ is *valid* if $\hat{e}(A, \hat{g}_2) = \hat{e}(\hat{A}, g_2)$. The case $t = 2$ is dual.

Theorem 3. *Let $t \in \{1, 2\}$. The knowledge commitment scheme is perfectly hiding in $\mathbb{G}_t$, and computationally binding in $\mathbb{G}_t$ under the $\varLambda$-PDL assumption in $\mathbb{G}_t$. If the $\varLambda$-PKE assumption holds in $\mathbb{G}_t$, then for any non-uniform PPT $\mathcal{A}$ that outputs some valid knowledge commitments, there exists a non-uniform PPT extractor $X_{\mathcal{A}}$ that, given the input of $\mathcal{A}$ together with $\mathcal{A}$'s random coins, extracts the contents of these commitments.*

The proof of this theorem is given in App. E. In the case of all security reductions in this paper, the tightness of the security reduction depends on the value λ_n. Clearly, the knowledge commitment scheme is also trapdoor, with the trapdoor being $\mathsf{td} = x$: after trapdoor-committing $A \leftarrow \mathcal{C}\mathrm{om}^t(\mathsf{ck}; \mathbf{0}; r) = g_t^r$ for $r \leftarrow \mathbb{Z}_p$, the committer can open it to $(\boldsymbol{a}; r - \sum_{i=1}^n a_i x^{\lambda_i})$ for any $\boldsymbol{a}$.

6 New Hadamard Product Argument

Assume that $(\mathcal{G}_{\mathsf{com}}, \mathcal{C}\mathrm{om})$ is the knowledge commitment scheme. In an *Hadamard product argument* (in group $\mathbb{G}_1$, the case of $\mathbb{G}_2$ is dual), the prover aims to convince the verifier that given commitments A, B and C, he can open them as $A = \mathcal{C}\mathrm{om}^1(\mathsf{ck}; \boldsymbol{a}; r_a)$, $B = \mathcal{C}\mathrm{om}^1(\mathsf{ck}; \boldsymbol{b}; r_b)$, and $C = \mathcal{C}\mathrm{om}^1(\mathsf{ck}; \boldsymbol{c}; r_c)$, s.t. $c_j = a_j b_j$ for $j \in [n]$. Groth constructed an Hadamard product argument [Gro10] with communication of 5 group elements,

System parameters: Let $n = \mathrm{poly}(\kappa)$. Let $\Lambda = \{\lambda_i : i \in [n]\}$ be a progression-free set of odd integers, such that $\lambda_{i+1} > \lambda_i > 0$. Denote $\lambda_0 := 0$. Let $\hat{\Lambda}$ be as in Eq. (4).

CRS generation $\mathcal{G}_{\mathsf{crs}}(1^\kappa)$**:** Let $\mathsf{gk} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}_{\mathsf{bp}}(1^\kappa)$. Let $\hat{\alpha}, x \leftarrow \mathbb{Z}_p$. Let $g_1 \leftarrow \mathbb{G}_1 \setminus \{1\}$ and $g_2 \leftarrow \mathbb{G}_2 \setminus \{1\}$. Denote $g_{t\ell} \leftarrow g_t^{x^\ell}$ and $\hat{g}_{t\ell} \leftarrow g_t^{\hat{\alpha}x^\ell}$ for $t \in \{1, 2\}$ and $\ell \in \{0\} \cup \hat{\Lambda}$. Let $D \leftarrow \prod_{i=1}^n g_{2,\lambda_i}$. The CRS is $\mathsf{crs} \leftarrow (\mathsf{gk}; (g_{1\ell}, \hat{g}_{1\ell})_{\ell \in \{0\} \cup \Lambda}, (g_{2\ell}, \hat{g}_{2\ell})_{\ell \in \hat{\Lambda}}, D)$. Let $\widehat{\mathsf{ck}}_1 \leftarrow (\mathsf{gk}; (g_{1\ell}, \hat{g}_{1\ell})_{\ell \in \{0\} \cup \Lambda})$.

Common inputs: $(A, \hat{A}, B, \hat{B}, B_2, C, \hat{C})$, where $(A, \hat{A}) \leftarrow \mathcal{C}\mathrm{om}^1(\widehat{\mathsf{ck}}_1; \boldsymbol{a}; r_a)$, $(B, \hat{B}) \leftarrow \mathcal{C}\mathrm{om}^1(\widehat{\mathsf{ck}}_1; \boldsymbol{b}; r_b)$, $B_2 \leftarrow g_2^{r_b} \cdot \prod_{i=1}^n g_{2,\lambda_i}^{b_i}$, $(C, \hat{C}) \leftarrow \mathcal{C}\mathrm{om}^1(\widehat{\mathsf{ck}}_1; \boldsymbol{c}; r_c)$, s.t. $a_i b_i = c_i$ for $i \in [n]$.

Argument generation $\mathcal{P}_\times(\mathsf{crs}; (A, \hat{A}, B, \hat{B}, B_2, C, \hat{C}), (\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, \boldsymbol{c}, r_c))$**:** Let $I_1(\ell) := \{(i, j) : i, j \in [n] \wedge j \neq i \wedge \lambda_i + \lambda_j = \ell\}$. For $\ell \in 2 \widehat{\ } \Lambda$, the prover sets $\mu_\ell \leftarrow \sum_{(i,j) \in I_1(\ell)} (a_i b_j - c_i)$. He sets $\psi \leftarrow g_2^{r_a r_b} \cdot \prod_{i=1}^n g_{2,\lambda_i}^{r_a b_i + r_b a_i - r_c} \cdot \prod_{\ell \in 2\widehat{\ }\Lambda} g_{2\ell}^{\mu_\ell}$, and $\hat{\psi} \leftarrow \hat{g}_2^{r_a r_b} \cdot \prod_{i=1}^n \hat{g}_{2,\lambda_i}^{r_a b_i + r_b a_i - r_c} \cdot \prod_{\ell \in 2\widehat{\ }\Lambda} \hat{g}_{2\ell}^{\mu_\ell}$. He sends $\psi^\times \leftarrow (\psi, \hat{\psi}) \in \mathbb{G}_2^2$ to the verifier as the argument.

Verification $\mathcal{V}_\times(\mathsf{crs}; (A, \hat{A}, B, \hat{B}, B_2, C, \hat{C}), \psi^\times)$**:** accept iff $\hat{e}(A, B_2)/\hat{e}(C, D) = \hat{e}(g_1, \psi)$ and $\hat{e}(g_1, \hat{\psi}) = \hat{e}(\hat{g}_1, \psi)$.

Protocol 1: New Hadamard product argument $[\![(A, \hat{A})]\!] \circ [\![(B, \hat{B}, B_2)]\!] = [\![(C, \hat{C})]\!]$

verifier's computation $\Theta(n)$, prover's computation of $\Theta(n^2)$ exponentiations and the CRS of $\Theta(n^2)$ group elements. We present a more efficient argument in Prot. 1. Intuitively, the discrete logarithm on basis $h = \hat{e}(g_1, g_2)$ of $\hat{e}(A, B_2)/\hat{e}(C, D) = \hat{e}(g_1, \psi)$ is a degree-n formal polynomial in X, which is spanned by $\{X^\ell\}_{\ell \in 2 \cdot \Lambda \cup \hat{\Lambda}}$, where

$$\hat{\Lambda} := \{0\} \cup \Lambda \cup 2\widehat{\ }\Lambda \ . \tag{4}$$

We need that $2 \cdot \Lambda$ and $\hat{\Lambda}$ do not intersect. The next lemma is straightforward to prove.

Lemma 1. *1) If Λ is a progression-free set of odd integers, then $2 \cdot \Lambda \cap \hat{\Lambda} = \emptyset$. 2) If $2 \cdot \Lambda \cap \hat{\Lambda} = \emptyset$, then Λ is a progression-free set.*

Moreover, since $\hat{\Lambda} \in \{0, \dots, 2\lambda_n\}$, then by Thm. 1,

Lemma 2. *For any value n there exists a choice of Λ such that $|\hat{\Lambda}| = n^{1+o(1)}$.*

We are now ready to state the security of the new Hadamard product argument for the knowledge commitment scheme. The (knowledge) commitments are $(A, \hat{A})$, $(B, \hat{B})$ and $(C, \hat{C})$. For efficiency reasons, we include another element B_2 to the Hadamard product language. We denote the argument in Prot. 1 by $[\![(A, \hat{A})]\!] \circ [\![(B, \hat{B}, B_2)]\!] = [\![(C, \hat{C})]\!]$. Since $(C, \hat{C})$ is always a commitment of $(a_1 b_1, \dots, a_n b_n)$ for *some* value of r_c, we cannot claim that Prot. 1 is computationally sound (even under a knowledge assumption). Instead, analogously to [Gro10], we prove a somewhat weaker version of soundness that is however sufficient to achieve soundness of the Circuit-SAT argument. Note that the last statement of the theorem basically says that no efficient adversary can output an input to the Hadamard product argument together with an accepting argument and openings to all commitments and all other pairs of type $(y, \hat{y})$ that are present in the argument, such that $a_i b_i \neq c_i$ for some $i \in [n]$. Intuitively, the theorem statement includes f'_ℓ only for $\ell \in \hat{\Lambda}$ (resp., a_ℓ for $\ell \in \Lambda$ together with r) since $\hat{g}_{2\ell}$ (resp., $\hat{g}_{1\ell}$) belongs to the CRS only for $\ell \in \hat{\Lambda}$ (resp., $\ell \in \{0\} \cup \Lambda$). This "weak" soundness is similar to the co-soundness as defined in [GL07]. However, in the case of co-soundness, the adversary would not be required to open the argument (by presenting values f'_ℓ as in the theorem statement). One could define the corresponding formal security notion, but in our opinion, it would not increase readability.

Theorem 4. *Prot. 1 is perfectly complete and perfectly witness-indistinguishable. If $\mathcal{G}_{\mathsf{bp}}$ is $\hat{\Lambda}$-PSDL secure, then a non-uniform PPT adversary has negligible chance of outputting $inp^\times \leftarrow (A, \hat{A}, B, \hat{B}, B_2, C, \hat{C})$ and an accepting argument $\psi^\times \leftarrow (\psi, \hat{\psi})$ together with a witness $w^\times \leftarrow (\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, \boldsymbol{c}, r_c, (f'_\ell)_{\ell \in \hat{\Lambda}})$, s.t. $(A, \hat{A}) = \mathcal{C}om^1(\widehat{\mathsf{ck}}_1; \boldsymbol{a}; r_a)$, $(B, \hat{B}) = \mathcal{C}om^1(\widehat{\mathsf{ck}}_1; \boldsymbol{b}; r_b)$, $B_2 = g_2^{r_b} \cdot \prod_{i=1}^n g_{2,\lambda_i}^{b_i}$, $(C, \hat{C}) = \mathcal{C}om^1(\widehat{\mathsf{ck}}_1; \boldsymbol{c}; r_c)$, $(\psi, \hat{\psi}) = (g_2^{\sum_{\ell \in \hat{\Lambda}} f'_\ell x^\ell}, \hat{g}_2^{\sum_{\ell \in \hat{\Lambda}} f'_\ell x^\ell})$, and for some $i \in [n]$, $a_i b_i \neq c_i$.*

The commitment scheme is defined as in Sect. 5 with respect to the set Λ. The following proof will make the intuition of Sect. 3 more formal. Note that the tightness of the reduction depends on the time it takes to factor a degree $(2\lambda_n + 1)$-polynomial.

Proof. Let $h \leftarrow \hat{e}(g_1, g_2)$ and $F(x) \leftarrow \log_h(\hat{e}(A, B_2)/\hat{e}(C, D))$ like in Sect. 3. WITNESS-INDISTINGUISHABILITY: since the argument $\psi^\times = (\psi, \hat{\psi})$ that satisfies the verification equations is unique, all witnesses result in the same argument, and therefore the Hadamard product argument is witness-indistinguishable.

PERFECT COMPLETENESS. Assume that the prover is honest. The second verification is straightforward. For the first one, due to discussion in Sect. 3, $F(x) = F_{\mathsf{con}}(x) + F_{\psi}(x)$, where $F_{\mathsf{con}}(x)$ and $F_{\psi}(x)$ are as defined by Eq. (2) and Eq. (3). Consider x to be a formal variable, then $F(X)$ is a formal polynomial of X. This formal polynomial is spanned by $\{X^{\ell}\}_{\ell \in 2 \cdot \Lambda \cup \hat{\Lambda}}$. If the prover is honest, then $c_i = a_i \cdot b_i$ for $i \in [n]$, and thus $F(X) = F_{\psi}(X)$ is spanned by $\{X^{\ell}\}_{\ell \in \hat{\Lambda}}$. Denoting $\psi \leftarrow g_2^{r_a r_b} \cdot \prod_{i=1}^{n} g_{2,\lambda_i}^{r_a b_i + r_b a_i - r_c} \cdot \prod_{i=1}^{n} \prod_{j=1:j \neq i}^{n} g_{2,\lambda_i+\lambda_j}^{a_i b_j - c_i} = g_2^{r_a r_b} \cdot \prod_{i=1}^{n} g_{2,\lambda_i}^{r_a b_i + r_b a_i - r_c} \cdot \prod_{\ell \in 2 \frown \Lambda} g_{2\ell}^{\mu_\ell}$, we see that clearly $e(g_1, \psi) = h$. Thus, the first verification succeeds.

WEAKER VERSION OF SOUNDNESS. Assume that $\mathcal{A}$ is an adversary that can break the last statement of the theorem. We construct an adversary $\mathcal{A}'$ against the $\hat{\Lambda}$-PSDL assumption. Let $\mathsf{gk} \leftarrow \mathcal{G}_{\mathsf{bp}}(1^{\kappa})$, $x \leftarrow \mathbb{Z}_p$, $g_1 \leftarrow \mathbb{G}_1 \setminus \{1\}$, and $g_2 \leftarrow \mathbb{G}_2 \setminus \{1\}$. The adversary $\mathcal{A}'$ receives $\mathsf{crs} \leftarrow (\mathsf{gk}; (g_1^{x^\ell}, g_2^{x^\ell})_{\ell \in \hat{\Lambda}})$ as her input, and her task is to output x. She sets $\hat{\alpha} \leftarrow \mathbb{Z}_p$, $\mathsf{crs}' \leftarrow (\mathsf{gk}; (g_1^{x^\ell}, g_1^{\hat{\alpha} x^\ell})_{\ell \in \{0\} \cup \Lambda}, (g_2^{x^\ell}, g_2^{\hat{\alpha} x^\ell})_{\ell \in \hat{\Lambda}}, \prod_{i=1}^{n} g_2^{x^{\lambda_i}})$, and then sends crs' to $\mathcal{A}$. Clearly, crs' has the same distribution as $\mathcal{G}_{\mathsf{crs}}(1^{\kappa})$. Both $\mathcal{A}$ and $\mathcal{A}'$ set $\mathsf{ck}_t \leftarrow (\mathsf{gk}; (g_t^{x^\ell}, g_t^{\hat{\alpha} x^\ell})_{\ell \in \{0\} \cup \Lambda})$ for $t \in \{1, 2\}$. Assume that $\mathcal{A}$ returns $(inp^{\times}, w^{\times}, \psi^{\times})$ such that the conditions in the theorem statement hold, and $\mathcal{V}(\mathsf{crs}'; inp^{\times}, \psi^{\times})$ accepts. Here, $inp^{\times} = (A, \hat{A}, B, \hat{B}, B_2, C, \hat{C})$ and $w^{\times} = (\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, \boldsymbol{c}, r_c, (f'_\ell)_{\ell \in \hat{\Lambda}})$.

If $\mathcal{A}$ is successful, $(A, \hat{A}) = \mathcal{C}\mathrm{om}^1(\widehat{\mathsf{ck}}_1; \boldsymbol{a}; r_a)$, $(B, \hat{B}) = \mathcal{C}\mathrm{om}^1(\widehat{\mathsf{ck}}_1; \boldsymbol{b}; r_b)$, $B_2 = g_2^{r_b} \cdot \prod_{i=1}^{n} g_{2,\lambda_i}^{b_i}$, $(C, \hat{C}) = \mathcal{C}\mathrm{om}^1(\widehat{\mathsf{ck}}_1; \boldsymbol{c}; r_c)$, and for some $i \in [n]$, $c_i \neq a_i b_i$. Since $2 \cdot \Lambda \cap \hat{\Lambda} = \emptyset$, $\mathcal{A}'$ has thus expressed $F(X)$ as a polynomial $f(X)$ where at least for some $\ell \in 2 \cdot \Lambda$, X^ℓ has a non-zero coefficient $a_i b_i - c_i$.

On the other hand, $\mathcal{A}$ also outputs $(f'_\ell)_{\ell \in \hat{\Lambda}}$, s.t. $F(x) = \log_{g_2} \psi = f'(x)$, where all non-zero coefficients of $f'(X) := \sum_{\ell \in \hat{\Lambda}} f'_\ell X^\ell$ correspond to X^ℓ for some $\ell \in \hat{\Lambda}$. Since Λ is a progression-free set of odd integers and all elements of $2 \cdot \Lambda$ are distinct, then by Lem. 1, $\ell \notin 2 \cdot \Lambda$. Thus, all coefficients of $f'(X)$ corresponding to any X^ℓ, $\ell \in 2 \cdot \Lambda$, are equal to 0. Thus $f(X) = \sum_{\ell \in \hat{\Lambda} \cup (2 \cdot \Lambda)} f_\ell X^\ell$ and $f'(X) = \sum_{\ell \in \hat{\Lambda}} f'_\ell X^\ell$ are different polynomials with $f(x) = f'(x) = F(x)$. Thus, $\mathcal{A}'$ has succeeded in creating a non-zero polynomial $d(X) = f(X) - f'(X)$, such that $d(x) = \sum_{\ell \in \hat{\Lambda} \cup (2 \cdot \Lambda)} d_\ell x^\ell = 0$.

Next, $\mathcal{A}'$ uses an efficient polynomial factorization algorithm [vHN10] in $\mathbb{Z}_p[X]$ to efficiently compute all $< 2\lambda_n + 1$ roots of $d(X)$. For some root y, $g_1^{x^\ell} = g_1^{y^\ell}$. The adversary $\mathcal{A}'$ sets $x \leftarrow y$, thus violating the $\hat{\Lambda}$-PSDL assumption. □

The Hadamard product argument is not perfectly zero-knowledge. The problem is that the simulator knows $\mathsf{td} = (\hat{\alpha}, x)$, but given td and the common input she will not be able to generate $\psi^{\times}$. E.g., she has to compute $\psi = g_2^{r_a r_b} \cdot \prod_{i=1}^{n} g_{2,\lambda_i}^{r_a b_i + r_b a_i - r_c} \cdot \prod_{i=1}^{n} \prod_{j=1}^{n} g_{2,\lambda_i+\lambda_j}^{a_i b_j - c_i}$ based on the input, $\hat{\alpha}$ and x, but without knowing the witness. This seems to be impossible. Technically, the problem is that due to the knowledge of the trapdoor, the simulator can, knowing one opening $(\boldsymbol{a}, r)$, produce an opening $(\boldsymbol{a}', r')$ to any other $\boldsymbol{a}'$. However, here she does not know any openings. Similarly, the permutation argument of Sect. 7 is not zero-knowledge. On the other hand, in the final circuit satisfiability argument of Sect. 8, the simulator creates all commitments by herself and can thus properly simulate the argument. By the same reason, the subarguments of [Gro10] are not zero-knowledge but the final argument (for circuit satisfiability) is.

Theorem 5. *Let Λ be as described in Thm. 1. The communication (argument size) of Prot. 1 is 2 elements from $\mathbb{G}_2$. The prover's computational complexity is $\Theta(n^2)$ scalar multiplications in $\mathbb{Z}_p$ and $n^{1+o(1)}$ exponentiations in $\mathbb{G}_2$. The verifier's computational complexity is dominated by 5 bilinear pairings and 1 bilinear-group multiplication. The CRS consists of $n^{1+o(1)}$ group elements, with the verifier's part of the CRS consisting of only the bilinear group description plus 5 group elements.*

The proof of this theorem is given in App. F.

In the Circuit-SAT argument, all a_i, b_i and c_i are Boolean, and thus all $n^{1+o(1)}$ values μ_ℓ can be computed in $n(n-1) = \Theta(n^2)$ scalar additions (the server also needs to use other operations like comparisons $j \neq i$, but they can be eliminated by using loop unrolling, and λ_i and λ_j can be computed by using table lookups), as follows:

```
1. For $\ell \in 2 \frown \Lambda$ do: $\mu_\ell \leftarrow 0$
2. For $i = 1$ to $n$ do:
   – If $a_i = 0$ then for $j = 1$ to $n$ do: if $j \neq i$ then $\mu_{\lambda_i+\lambda_j} \leftarrow \mu_{\lambda_i+\lambda_j} - c_i$
   – Else for $j = 1$ to $n$ do: if $j \neq i$ then $\mu_{\lambda_i+\lambda_j} \leftarrow \mu_{\lambda_i+\lambda_j} + b_j - c_i$
```

7 New Permutation Argument

In a *permutation argument*, the prover aims to convince the verifier that for given permutation $\varrho \in S_n$ and two commitments A and B, he knows how to open them as $A = \mathcal{C}\mathrm{om}^1(\mathsf{ck}; \boldsymbol{a}; r_a)$ and $B = \mathcal{C}\mathrm{om}^1(\mathsf{ck}; \boldsymbol{b}; r_b)$, such that $b_j = a_{\varrho(j)}$ for $j \in [n]$. We assume that ϱ is a part of the statement. In [Gro10], Groth constructed a permutation argument, where the prover's computation is $\Theta(n^2)$ exponentiations and the CRS has $\Theta(n^2)$ group elements. We now propose a new argument with the CRS of $n^{1+o(1)}$ group elements. We also improve the prover's concrete computation, and the argument is based on a (probably) weaker assumption.

The new permutation argument $\varrho([\![(A, \tilde{A})]\!]) = [\![(B, \tilde{B})]\!]$, see Prot. 2, uses (almost) the same high-level ideas as the Hadamard product argument from Sect. 6. However, the situation is more complicated. Consider the verification equation $\hat{e}(g_1, \psi^\varrho) = \hat{e}(A, g_2^{\sum_{i=1}^n x^{\lambda_i}}) / \hat{e}(B, g_2^{\sum_{i=1}^n x^{2\lambda_{\varrho_i} - \lambda_i}})$ from [Gro10]. (See App. H for why this definition of $D = g_2^{\sum_{i=1}^n x^{\lambda_i}}$ and $E_\varrho = g_2^{\sum_{i=1}^n x^{2\lambda_{\varrho_i} - \lambda_i}}$ may make sense.) Letting $h = \hat{e}(g_1, g_2)$, $F_\varrho(x) := \log_{g_2} \psi^\varrho = \sum_i (a_{\varrho(i)} - b_i) x^{2\lambda_{\varrho(i)}} + r_a \sum_i x^{\lambda_i} - r_b \sum_i x^{2\lambda_{\varrho(i)} - \lambda_i} + \sum_i a_{\varrho(i)} \cdot \sum_{j \neq i} x^{\lambda_{\varrho(i)} + \lambda_{\varrho(j)}} - \sum_i b_i \cdot \sum_{j \neq i}^n x^{\lambda_i + 2\lambda_{\varrho(j)} - \lambda_j}$. Following Sect. 6, we require that $\tilde{\varLambda} = \varLambda \cup \{2\lambda_k - \lambda_i\} \cup 2^\frown\varLambda \cup \{\lambda_i + 2\lambda_k - \lambda_j : i \neq j\}$ and $2 \cdot \varLambda$ do not intersect. Since ϱ is a part of the statement, we replaced $\varrho(i)$ and $\varrho(j)$ with a new element k.

Assume that $\varLambda$ is a progression-free set of odd integers. Since $\varLambda$ consists of odd integers, $(\varLambda \cup \{2\lambda_k - \lambda_i\}) \cap 2 \cdot \varLambda = \emptyset$. Since $\varLambda$ is a progression-free set, $2^\frown\varLambda \cap 2 \cdot \varLambda = \emptyset$. However, we also need that $2\lambda_{k^*} \neq 2\lambda_k + \lambda_i - \lambda_j$ for $i \neq j$. That is, one can uniquely represent any non-negative integer a as $a = 2\lambda_{k^*} + \lambda_j$. (It is only required that any non-negative integer a has at most one representation as $a = 2\lambda_{k^*} + \lambda_j$. As we show in App. J, this does not help our case.) The unique sequence $\varLambda = (\lambda_i)_{i \in \mathbb{Z}^+}$ (the *Moser-de Bruijn sequence* [Mos62,dB64]) that satisfies this property is the sequence of all non-negative integers that have only 0 or 1 as their radix-4 digits. Since $\lambda_n = \Theta(n^2)$, this sequence is not good enough.

Fortunately, we can overcome this problem as follows. For $i \in [n]$ and a permutation ϱ, let $T_\varLambda(i, \varrho) := |\{j \in [n] : 2\lambda_{\varrho(i)} + \lambda_j = 2\lambda_{\varrho(j)} + \lambda_i\}|$. Note that $1 \leq T_\varLambda(i, \varrho) \leq n$, and that for fixed $\varLambda$ and ϱ, the whole tuple $\boldsymbol{T_\varLambda(\varrho)} := (T_\varLambda(1, \varrho), \dots, T_\varLambda(n, \varrho))$ can be computed in $\Theta(n)$ simple arithmetic operations. We can then rewrite $F_\varrho(x)$ as

$$\begin{aligned} F_\varrho(x) = &\sum_{i=1}^n (a_{\varrho(i)} - T_\varLambda(i, \varrho) \cdot b_i) x^{2\lambda_{\varrho(i)}} + r_a \sum_{i=1}^n x^{\lambda_i} - r_b \sum_{i=1}^n x^{2\lambda_{\varrho(i)} - \lambda_i} + \\ &\sum_{i=1}^n a_{\varrho(i)} \sum_{\substack{j=1 \\ j \neq i}}^n x^{\lambda_{\varrho(i)} + \lambda_{\varrho(j)}} - \sum_{i=1}^n b_i \sum_{\substack{j=1 \\ j \neq i \\ 2\lambda_{\varrho(i)} + \lambda_j \neq \lambda_i + 2\lambda_{\varrho(j)}}}^n x^{\lambda_i + 2\lambda_{\varrho(j)} - \lambda_j} \quad , \end{aligned} \tag{5}$$

with $\tilde{\varLambda}$ being redefined as

$$\tilde{\varLambda} = \varLambda \cup \{2\lambda_k - \lambda_i\} \cup 2^\frown\varLambda \cup (\{\lambda_i + 2\lambda_k - \lambda_j : i \neq j\} \setminus 2 \cdot \varLambda) \quad . \tag{6}$$

Since $\tilde{\varLambda} \cap 2 \cdot \varLambda = \emptyset$, $\hat{e}(A, D) / \hat{e}(B, E_\varrho) = \hat{e}(g_1, \psi^\varrho)$ (with $D = g_2^{\sum_{i=1}^n x^{\lambda_i}}$ and $E = E_\varrho = g_2^{\sum_{i=1}^n x^{2\lambda_{\varrho(i)} - \lambda_i}}$, as it follows from Eq. (8) and (9)) convinces the verifier that $a_{\varrho(i)} = T_\varLambda(i, \varrho) \cdot b_i$ for $i \in [n]$. To finish the permutation argument, we let $(A^*, \hat{A}^*)$ to be a commitment to $(a_1^*, \dots, a_n^*) := (T_\varLambda(\varrho^{-1}(1), \varrho) \cdot a_1, \dots, T_\varLambda(\varrho^{-1}(n), \varrho) \cdot a_n)$, use an Hadamard product argument to show that $a_i^* = T_\varLambda(\varrho^{-1}(i), \varrho) \cdot a_i$ (and thus $a_{\varrho(i)}^* = T_\varLambda(i, \varrho) \cdot a_{\varrho(i)}$) for $i \in [n]$, and an argument as described above in this section to show that $a_{\varrho(i)}^* = T_\varLambda(i, \varrho) \cdot b_i$ for $i \in [n]$. Therefore, $a_{\varrho(i)} = b_i$ for $i \in [n]$.

Clearly $\hat{\varLambda} \cup \tilde{\varLambda} = \{0\} \cup \tilde{\varLambda}$. Since $\tilde{\varLambda} \subset \{-\lambda_n + 1, \dots, 3\lambda_n\}$, then by Thm. 1

Lemma 3. *For any n there exists a choice of $\varLambda$ such that $|\tilde{\varLambda}| = n^{1+o(1)}$.*

We are now ready to state the security of the new permutation argument. The (weaker version of) soundness of this argument is based on exactly the same ideas as that of the Hadamard product argument.

Theorem 6. *Prot. 2 is perfectly complete and perfectly witness-indistinguishable. If $\mathcal{G}_{\mathsf{bp}}$ is $\tilde{\varLambda}$-PSDL secure, then a non-uniform PPT adversary has negligible chance of outputting $inp^{\mathsf{perm}} \leftarrow (A, \tilde{A}, B, \hat{B}, \tilde{B}, \varrho)$ and an accepting $\psi^{\mathsf{perm}} \leftarrow (A^*, \hat{A}^*, \psi^\times, \hat{\psi}^\times, \psi^\varrho, \tilde{\psi}^\varrho)$ together with a witness $w^{\mathsf{perm}} \leftarrow (\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, \boldsymbol{a^*}, r_{a^*}, (f'_{(\times,\ell)})_{\ell \in \hat{\varLambda}}, (f'_{(\varrho,\ell)})_{\ell \in \tilde{\varLambda}})$, s.t. $(A, \tilde{A}) = \mathcal{C}om^1(\widetilde{\mathsf{ck}}_1; \boldsymbol{a}; r_a)$, $(B, \hat{B}) = \mathcal{C}om^1(\widehat{\mathsf{ck}}_1; \boldsymbol{b}; r_b)$, $(B, \tilde{B}) = \mathcal{C}om^1(\widetilde{\mathsf{ck}}_1; \boldsymbol{b}; r_b)$, $(A^*, \hat{A}^*) =$*

System parameters: Same as in Prot. 1, but let $\tilde{\varLambda}$ be as in Eq. (6).

CRS generation $\mathcal{G}_{\mathsf{crs}}(1^\kappa)$**:** Let $\mathsf{gk} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}_{\mathsf{bp}}(1^\kappa)$. Let $\hat{\alpha}, \tilde{\alpha}, x \leftarrow \mathbb{Z}_p$. Let $g_1 \leftarrow \mathbb{G}_1 \setminus \{1\}$ and $g_2 \leftarrow \mathbb{G}_2 \setminus \{1\}$. Let $\hat{g}_t \leftarrow \hat{g}_t^{\hat{\alpha}}$ and $\tilde{g}_t \leftarrow \tilde{g}_t^{\tilde{\alpha}}$ for $t \in \{1,2\}$. Denote $g_{t\ell} \leftarrow g_t^{x^\ell}$, $\hat{g}_{t\ell} \leftarrow \hat{g}_t^{x^\ell}$, and $\tilde{g}_{t\ell} \leftarrow \tilde{g}_t^{x^\ell}$ for $t \in \{1,2\}$ and $\ell \in \{0\} \cup \tilde{\varLambda}$. Let $(D, \tilde{D}) \leftarrow (\prod_{i=1}^n g_{2,\lambda_i}, \prod_{i=1}^n \tilde{g}_{2,\lambda_i})$. The CRS is

$$\mathsf{crs} \leftarrow (\mathsf{gk}; (g_{1\ell}, \hat{g}_{1\ell}, \tilde{g}_{1\ell})_{\ell \in \{0\} \cup \varLambda}, (g_{2\ell})_{\ell \in \{0\} \cup \tilde{\varLambda}}, (\hat{g}_{2\ell})_{\ell \in \hat{\varLambda}}, (\tilde{g}_{2\ell})_{\ell \in \tilde{\varLambda}}, D, \tilde{D}) \enspace .$$

Let $\widehat{\mathsf{ck}}_1 \leftarrow (\mathsf{gk}; (g_{1\ell}, \hat{g}_{1\ell})_{\ell \in \{0\} \cup \varLambda})$, $\widetilde{\mathsf{ck}}_1 \leftarrow (\mathsf{gk}; (g_{1\ell}, \tilde{g}_{1\ell})_{\ell \in \{0\} \cup \varLambda})$.

Common inputs: $(A, \tilde{A}, B, \hat{B}, \tilde{B}, \varrho)$, where $\varrho \in S_n$, $(A, \tilde{A}) \leftarrow \mathcal{C}\mathrm{om}^1(\widetilde{\mathsf{ck}}_1; \boldsymbol{a}; r_a)$, $(B, \hat{B}) \leftarrow \mathcal{C}\mathrm{om}^1(\widehat{\mathsf{ck}}_1; \boldsymbol{b}; r_b)$, and $(B, \tilde{B}) \leftarrow \mathcal{C}\mathrm{om}^1(\widetilde{\mathsf{ck}}_1; \boldsymbol{b}; r_b)$, s.t. $b_j = a_{\varrho(j)}$ for $j \in [n]$.

Argument generation $\mathcal{P}_{\mathsf{perm}}(\mathsf{crs}; (A, \tilde{A}, B, \hat{B}, \tilde{B}, \varrho), (\boldsymbol{a}, r_a, \boldsymbol{b}, r_b))$**:**

1. Let $(T^*, \hat{T}^*, T_2^*) \leftarrow (\prod_{i=1}^n g_{1,\lambda_i}^{T_\varLambda(\varrho^{-1}(i),\varrho)}, \prod_{i=1}^n \hat{g}_{1,\lambda_i}^{T_\varLambda(\varrho^{-1}(i),\varrho)}, \prod_{i=1}^n g_{2,\lambda_i}^{T_\varLambda(\varrho^{-1}(i),\varrho)})$.
2. Let $r_{a^*} \leftarrow \mathbb{Z}_p$, $(A^*, \hat{A}^*) \leftarrow \mathcal{C}\mathrm{om}_1(\widehat{\mathsf{ck}}_1; T_\varLambda(\varrho^{-1}(1), \varrho) \cdot a_1, \dots, T_\varLambda(\varrho^{-1}(n), \varrho) \cdot a_n; r_{a^*})$. Create an argument $\psi^\times$ for $[\![(A, \hat{A})]\!] \circ [\![(T^*, \hat{T}^*, T_2^*)]\!] = [\![(A^*, \hat{A}^*)]\!]$.
3. Let $\tilde{\varLambda}'_\varrho := 2\widehat{\ }\varLambda \cup (\{2\lambda_{\varrho(j)} + \lambda_i - \lambda_j : i, j \in [n] \wedge i \neq j\} \setminus 2 \cdot \varLambda) \subset \{-\lambda_n + 1, \dots, 3\lambda_n\}$.
4. For $\ell \in \tilde{\varLambda}'_\varrho$, $I_1(\ell)$ as in Prot. 1, and $I_2(\ell) := \{(i,j) : i, j \in [n] \wedge j \neq i \wedge 2\lambda_{\varrho(i)} + \lambda_j \neq \lambda_i + 2\lambda_{\varrho(j)} \wedge 2\lambda_{\varrho(j)} + \lambda_i - \lambda_j = \ell\}$, set
$$\mu_{\varrho,\ell} \leftarrow \sum_{(i,j) \in I_1(\ell)} a_i^* - \sum_{(i,j) \in I_2(\ell)} b_i \enspace .$$
5. Let $(E_\varrho, \tilde{E}_\varrho) \leftarrow (\prod_{i=1}^n g_{2,2\lambda_{\varrho(i)} - \lambda_i}, \prod_{i=1}^n \tilde{g}_{2,2\lambda_{\varrho(i)} - \lambda_i})$.
6. Let $\psi^\varrho \leftarrow D^{r_a^*} \cdot E_\varrho^{-r_b} \cdot \prod_{\ell \in \tilde{\varLambda}'_\varrho} g_{2\ell}^{\mu_{\varrho,\ell}}$, $\tilde{\psi}^\varrho \leftarrow \tilde{D}^{r_a^*} \cdot \tilde{E}_\varrho^{-r_b} \cdot \prod_{\ell \in \tilde{\varLambda}'_\varrho} \tilde{g}_{2\ell}^{\mu_{\varrho,\ell}}$,

Send $\psi^{\mathsf{perm}} \leftarrow (A^*, \hat{A}^*, \psi^\times, \psi^\varrho, \tilde{\psi}^\varrho) \in \mathbb{G}_1^2 \times \mathbb{G}_2^4$ to the verifier as the argument.

Verification $\mathcal{V}_{\mathsf{perm}}(\mathsf{crs}; (A, \tilde{A}, B, \hat{B}, \tilde{B}, \varrho), \psi^{\mathsf{perm}})$**:** Let E_ϱ and $(T^*, \hat{T}^*, T_2^*)$ be computed as in $\mathcal{P}_{\mathsf{perm}}$. If $\psi^\times$ verifies, $\hat{e}(A^*, D)/\hat{e}(B, E_\varrho) = \hat{e}(g_1, \psi^\varrho)$, $\hat{e}(A^*, \hat{g}_2) = \hat{e}(\hat{A}^*, g_2)$, and $\hat{e}(g_1, \tilde{\psi}^\varrho) = \hat{e}(\tilde{g}_1, \psi^\varrho)$, then $\mathcal{V}_{\mathsf{perm}}$ accepts. Otherwise, $\mathcal{V}_{\mathsf{perm}}$ rejects.

Protocol 2: New permutation argument $\varrho([\![(A, \tilde{A})]\!]) = [\![(B, \tilde{B})]\!]$

$\mathcal{C}om^1(\widehat{\mathsf{ck}}_1; \boldsymbol{a^*}; r_{a^*})$, $(\psi^\times, \hat{\psi}^\times) = (g_2^{\sum_{\ell \in \hat{\varLambda}} f'_{(\times,\ell)}}, \hat{g}_2^{\sum_{\ell \in \hat{\varLambda}} f'_{(\times,\ell)}})$, $(\psi^\varrho, \hat{\psi}^\varrho) = (g_2^{\sum_{\ell \in \tilde{\varLambda}} f'_{(\varrho,\ell)}}, \tilde{g}_2^{\sum_{\ell \in \tilde{\varLambda}} f'_{(\varrho,\ell)}})$, $a_i^* = T_\varLambda(\varrho^{-1}(i), \varrho) \cdot a_i$ (for $i \in [n]$), and for some $i \in [n]$, $a_{\varrho(i)} \neq b_i$.

Proof. Denote $h \leftarrow \hat{e}(g_1, g_2)$ and $F_\varrho(x) := \log_h(\hat{e}(A^*, D)/\hat{e}(B, E_\varrho))$. WITNESS-INDISTINGUISHABILITY: since argument ψ^{perm} that satisfies the verification equations is unique, all witnesses result in the same argument, and therefore the permutation argument is witness-indistinguishable.

PERFECT COMPLETENESS. Completeness of $\psi^\times$ follows from the completeness of the Hadamard product argument. The third and the fourth verifications are straightforward. For the verification $\hat{e}(A^*, D)/\hat{e}(B, E_\varrho) = \hat{e}(g_1, \psi^\varrho)$, consider $F_\varrho(x)$ in Eq. (5). Consider X as a formal variable, then the right-hand side (and thus also $F_\varrho(X)$) is a formal polynomial of X, spanned by $\{X^\ell\}_{\ell \in 2 \cdot \varLambda \cup \tilde{\varLambda}}$. If the prover is honest, then $b_i = a_{\varrho(i)}$ for $i \in [n]$, and thus $F_\varrho(X)$ is spanned by $\{X^\ell\}_{\ell \in \tilde{\varLambda}}$. Defining $\psi^\varrho \leftarrow (\prod_{i=1}^n g_{2,\lambda_i})^{r_{a^*}} \cdot (\prod_{i=1}^n g_{2,2\lambda_{\varrho(i)} - \lambda_i})^{-r_b} \cdot \prod_{i=1}^n (\prod_{j=1:j \neq i}^n g_{2,\lambda_i + \lambda_j})^{a_i^*} \cdot \prod_{i=1}^n (\prod_{j \in I_2^*(i,\ell)} g_{2,\lambda_i + 2\lambda_{\varrho(j)} - \lambda_j})^{-b_i} = D^{r_{a^*}} \cdot E_\varrho^{-r_b} \cdot \prod_{\ell \in \tilde{\varLambda}'_\varrho} g_{2\ell}^{\mu_{\varrho,\ell}}$, where $I_2^*(i, \ell) := \{j \in [n] : j \neq i \wedge 2\lambda_{\varrho(i)} + \lambda_j \neq \lambda_i + 2\lambda_{\varrho(j)}\}$, we see that the second verification holds.

WEAKER VERSION OF SOUNDNESS. Assume that $\mathcal{A}$ is an adversary that can break the last statement of the theorem. We construct an adversary $\mathcal{A}'$ against the $\tilde{\varLambda}$-PSDL assumption. Let $\mathsf{gk} \leftarrow \mathcal{G}_{\mathsf{bp}}(1^\kappa)$, $x \leftarrow \mathbb{Z}_p$, $g_1 \leftarrow \mathbb{G}_1 \setminus \{1\}$, and $g_2 \leftarrow \mathbb{G}_2 \setminus \{1\}$. The adversary $\mathcal{A}'$ receives $\mathsf{crs} \leftarrow (\mathsf{gk}; (g_1^{x^\ell}, g_2^{x^\ell})_{\ell \in \{0\} \cup \tilde{\varLambda}})$ as her input, and her task is to output x. She sets $\hat{\alpha} \leftarrow \mathbb{Z}_p$, $\tilde{\alpha} \leftarrow \mathbb{Z}_p$, and $\mathsf{crs}' \leftarrow (\mathsf{gk}; (g_1^{x^\ell}, g_1^{\hat{\alpha} x^\ell}, g_1^{\tilde{\alpha} x^\ell})_{\ell \in \{0\} \cup \varLambda}, (g_2^{x^\ell})_{\ell \in \{0\} \cup \tilde{\varLambda}}, (g_2^{\hat{\alpha} x^\ell})_{\ell \in \hat{\varLambda}}, (g_2^{\tilde{\alpha} x^\ell})_{\ell \in \tilde{\varLambda}}, \prod_{i=1}^n g_2^{x^{\lambda_i}}, \prod_{i=1}^n \tilde{g}_2^{x^{\lambda_i}})$, and forwards crs' to $\mathcal{A}$. Clearly, crs' has the same distribution as $\mathcal{G}_{\mathsf{crs}}(1^\kappa)$. Both parties also set $\hat{\mathsf{ck}}_1 \leftarrow (\mathsf{gk}; (g_1^{x^\ell}, g_1^{\hat{\alpha} x^\ell})_{\ell \in \{0\} \cup \varLambda})$ and $\widetilde{\mathsf{ck}}_1 \leftarrow (\mathsf{gk}; (g_1^{x^\ell}, g_1^{\tilde{\alpha} x^\ell})_{\ell \in \{0\} \cup \varLambda})$.

Assume that $\mathcal{A}$ returns $(inp^{\mathsf{perm}}, w^{\mathsf{perm}}, \psi^{\mathsf{perm}})$ such that the conditions in the theorem statement hold, and $\mathcal{V}(\mathsf{crs}'; inp^{\mathsf{perm}}, \psi^{\mathsf{perm}})$ accepts. Here, $inp^{\mathsf{perm}} = (A, \tilde{A}, B, \hat{B}, \tilde{B}, \varrho)$ and $w^{\mathsf{perm}} = (\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, \boldsymbol{a^*}, r_{a^*}, (f'_{(\times,\ell)})_{\ell \in \hat{\varLambda}}, (f'_{(\varrho,\ell)})_{\ell \in \tilde{\varLambda}})$.

If $\mathcal{A}$ is successful, $(A, \tilde{A}) = \mathcal{C}\mathrm{om}^1(\widetilde{\mathsf{ck}}_1; \boldsymbol{a}; r_a)$, $(B, \hat{B}) = \mathcal{C}\mathrm{om}^1(\widehat{\mathsf{ck}}_1; \boldsymbol{b}; r_b)$, $(B, \tilde{B}) = \mathcal{C}\mathrm{om}^1(\widetilde{\mathsf{ck}}_1; \boldsymbol{b}; r_b)$, $\psi^\times$ verifies, and for some $i \in [n]$, $a_{\varrho(i)} \neq T_\varLambda(i, \varrho) \cdot b_i$. Since $\psi^\times$ verifies and the Hadamard product argument is (weakly) sound, we have that $(A^*, \hat{A}^*)$ commits to $(T_\mathit{\Lambda}(\varrho^{-1}(1), \varrho) \cdot a_1, \dots, T_\mathit{\Lambda}(\varrho^{-1}(n), \varrho) \cdot a_n)$. (Otherwise, we have broken the PSDL assumption.) Since $2 \cdot \mathit{\Lambda} \cap \tilde{\mathit{\Lambda}} = \emptyset$, $\mathcal{A}'$ has expressed $F_\varrho(X)$ as a polynomial $f(X)$ where at least for some $\ell \in 2 \cdot \mathit{\Lambda}$, X^ℓ has a non-zero coefficient.

On the other hand, $\mathcal{A}$ also outputs $(f'_{(\varrho,\ell)})_{\ell \in \tilde{\mathit{\Lambda}}}$, s.t. $F_\varrho(x) = \log_{g_2} \psi = f'_\varrho(x)$, where all non-zero coefficients of $f'_\varrho(X) := \sum_{\ell \in \tilde{\mathit{\Lambda}}} f'_{(\varrho,\ell)} X^\ell$ correspond to X^ℓ for some $\ell \in \tilde{\mathit{\Lambda}}$. Since $\mathit{\Lambda}$ is a progression-free set of odd integers and all elements of $2 \cdot \tilde{\mathit{\Lambda}}$ are distinct, then by the discussion in the beginning of Sect. 7, $\ell \notin 2 \cdot \mathit{\Lambda}$. Thus, all coefficients of $f'_\varrho(X)$ corresponding to any X^ℓ, $\ell \in 2 \cdot \mathit{\Lambda}$, are equal to 0. Thus, $f(X) \cdot X^{\lambda_n} = \sum_{\ell \in \tilde{\mathit{\Lambda}} \cup (2 \cdot \mathit{\Lambda})} f_\ell X^{\ell + \lambda_n}$ and $f'_\varrho(X) = \sum_{\ell \in \tilde{\mathit{\Lambda}}} f'_{(\varrho,\ell)} X^{\ell + \lambda_n}$ are different polynomials with $f(x) = f'_\varrho(x) = F_\varrho(x)$. Thus, $\mathcal{A}'$ has succeeded in creating a nonzero polynomial $d_\varrho(X) = f(X) \cdot X^{\lambda_n} - f'_\varrho(X)$, such that $d_\varrho(x) = \sum_{\ell \in \tilde{\mathit{\Lambda}}} d_\ell x^\ell = 0$.

Next, $\mathcal{A}'$ can use an efficient polynomial factorization algorithm [vHN10] in $\mathbb{Z}_p[X]$ to efficiently compute all $\leq 4\lambda_n + 1$ roots of $d_\varrho(X)$. For some root y, $g_1^{x^\ell} = g_1^{y^\ell}$. The adversary $\mathcal{A}'$ sets $x \leftarrow y$, thus violating the $\tilde{\mathit{\Lambda}}$-PSDL assumption. □

Note that in an upper level argument, the verifier must check that $\hat{e}(A, \tilde{g}_2) = \hat{e}(\tilde{A}, g_2)$, $\hat{e}(B, \hat{g}_2) = \hat{e}(\hat{B}, g_2)$, and $\hat{e}(B, \tilde{g}_2) = \hat{e}(\tilde{B}, g_2)$.

Theorem 7. *Let $\mathit{\Lambda}$ be as described in Thm. 1. The CRS consists of $n^{1+o(1)}$ group elements. The argument size of Prot. 2 is 2 elements from $\mathbb{G}_1$ and 4 elements from $\mathbb{G}_2$. The prover's computational complexity is dominated by $\Theta(n^2)$ scalar additions in $\mathbb{Z}_p$ and by $n^{1+o(1)}$ exponentiations in $\mathbb{G}_2$. The verifier's computational complexity is dominated by 12 bilinear pairings and $4n - 2$ bilinear-group multiplications.*

The proof of this theorem can be found in App. I.

8 New NIZK Argument for Circuit Satisfiability

In a *NIZK argument for circuit satisfiability* (Circuit-SAT, well-known to be an $\mathbf{NP}$-complete language), the prover and the verifier share a circuit C. The prover aims to prove in non-interactive zero-knowledge that she knows an assignment of input values that makes the circuit output 1. As in [Gro10], the Circuit-SAT argument will use the Hadamard product argument, the permutation argument and a trivial argument for element-wise sum of two tuples — in our case, all operating in parallel on $(2|C| + 1)$-dimensional tuples, where $|C|$ is the circuit size. Those three arguments can be seen as basic operations in an NIZK "programming language" for all languages in $\mathbf{NP}$. We show that a small constant number of such basic operations is sufficient for Circuit-SAT. The full argument then contains additional cryptographic sugar: a precise definition of the used CRS, computational/communication optimizations, etc.

To make it easier to read the following (and to construct future protocols based on this "programming language"), we give first a high-level description of the resulting argument. The first task is to express the underlying argument as a parallel composition of some addition, permutation and Hadamard product arguments. These arguments may include intermediate variables (that will be committed to by the prover) and constants (that can be online committed to by both of the parties separately). When choosing the arguments, one has to keep in mind that we work in an asymmetric setting. This may mean that for some of the inputs to the circuit satisfiability argument, one has to commit to them both in $\mathbb{G}_1$ and $\mathbb{G}_2$ (and the verifier has to check that this is done correctly).

The CRS is basically the CRS of the permutation argument. The total argument consists of commitments to intermediate variables and of all arguments in the program of this "programming language". Finally, the verifier has to check that all commitments are internally consistent, and then verify all used arguments.

Let us now turn to the concrete case of circuit satisfiability. For the sake of simplicity, assume that the circuit C is only composed of NAND gates. Let C have n gates. Assume that the output gate of the circuit is n, and U_n is the output of the circuit. For every gate $j \in [n]$ of C, let the input wires of its jth gate be L_j and R_j, and let U_j be one of its output wires. We also define an extra value $R_{n+1} = 1$. We let X_j be other "output" wires that correspond to some L_k or R_k that were not already covered by U_k (that is, inputs to the circuit, or duplicates of output wires). That is, $(U_1, \dots, U_n, X_1, \dots, X_{n+1})$ is chosen so that for some permutation ζ, $(\boldsymbol{U}, \boldsymbol{X}, X_{n+1})$ is a ζ-permutation of $(\boldsymbol{L}, \boldsymbol{R}, R_{n+1})$, where $\boldsymbol{Y} = (Y_1, \dots, Y_n)$ for $Y \in \{L, R, U, X\}$. (See Fig. 1 in appendix for an example circuit with values R_i, L_i, U_i and X_i.)

More precisely, the prover and the verifier share the following three permutations, the first two of which completely describe the circuit C. First, $\tau \in S_{2n+1}$ is a permutation, such that for any values $L_{i_1}, \dots, L_{i_s}, R_{j_1}, \dots, R_{j_t}$ that correspond to the same wire, τ contains a cycle $i_1 \to i_2 \to \cdots \to i_s \to j_1 + n \to$

System parameters: Define $\varLambda$ and $\hat{\varLambda}$ as in Prot. 1 and $\tilde{\varLambda}$ as in Prot. 2, but in all cases with n replaced by $2|C|+1$. Permutation swap.

CRS generation $\mathcal{G}_{\mathsf{crs}}(1^\kappa)$: Let all other variables (including the secret ones) be defined as in the CRS generation of Prot. 2, but let $\mathsf{crs}^{\mathsf{perm}}$ be the CRS of Prot. 2. In addition, let $(\hat{D}, D_2) \leftarrow (\prod_{i=1}^n \hat{g}_{1,\lambda_i}, \prod_{i=1}^n g_{2,\lambda_i})$. The CRS is $\mathsf{crs} \leftarrow (\mathsf{crs}^{\mathsf{perm}}, \hat{D}, D_2)$. Let $\mathsf{ck}_1 \leftarrow (\mathsf{gk}; (g_{1\ell}, \hat{g}_{1\ell}, \tilde{g}_{1\ell})_{\ell\in\{0\}\cup\varLambda})$.

Common inputs: A satisfiable circuit C, and permutations τ and ζ generated based on C, such that $(\boldsymbol{L}, \boldsymbol{R}, R_{n+1}, \boldsymbol{U}, \boldsymbol{X}, X_{n+1})$ is a "satisfying assignment".

Argument generation $\mathcal{P}(\mathsf{crs}; C, (\boldsymbol{L}, \boldsymbol{R}, R_{n+1}, \boldsymbol{U}, \boldsymbol{X}))$: Denote $\boldsymbol{Y} := (Y_1, \dots, Y_n)$ for $Y \in \{L, R, U, X\}$. The prover does the following.

1. Set $r_1, \dots, r_4 \leftarrow \mathbb{Z}_p$, and then compute $(\mathsf{lr}, \widehat{\mathsf{lr}}, \widetilde{\mathsf{lr}}) \leftarrow \mathcal{C}\mathrm{om}^1(\mathsf{ck}_1; \boldsymbol{L}, \boldsymbol{R}, R_{n+1}; r_1)$, $\mathsf{lr}_2 \leftarrow g_2^{r_1} \cdot \prod_{i=1}^n g_{2,\lambda_i}^{L_i} \cdot \prod_{i=1}^{n+1} g_{2,\lambda_{i+n}}^{R_i}$, $(\mathsf{rl}, \widetilde{\mathsf{rl}}) \leftarrow \mathcal{C}\mathrm{om}^1(\widetilde{\mathsf{ck}}_1; \boldsymbol{R}, \boldsymbol{L}, R_{n+1}; r_1)$, $(\mathsf{rz}, \widehat{\mathsf{rz}}) \leftarrow \mathcal{C}\mathrm{om}^1(\widehat{\mathsf{ck}}_1; \boldsymbol{R}, 0, \dots, 0, 0; r_2)$, $(\mathsf{uz}, \widehat{\mathsf{uz}}) \leftarrow \mathcal{C}\mathrm{om}^1(\widehat{\mathsf{ck}}_1; \boldsymbol{U}, 0, \dots, 0, 0; r_3)$, $(\mathsf{ux}, \widehat{\mathsf{ux}}, \widetilde{\mathsf{ux}}) \leftarrow \mathcal{C}\mathrm{om}^1(\mathsf{ck}_1; \boldsymbol{U}, \boldsymbol{X}, X_{n+1}; r_4)$.
2. Create an argument ψ_1 for $\llbracket(\mathsf{lr}, \widehat{\mathsf{lr}})\rrbracket \circ \llbracket(\mathsf{lr}, \widehat{\mathsf{lr}}, \mathsf{lr}_2)\rrbracket = \llbracket(\mathsf{lr}, \widehat{\mathsf{lr}})\rrbracket$, ψ_2 for $\mathsf{swap}(\llbracket(\mathsf{rl}, \widetilde{\mathsf{rl}})\rrbracket) = \llbracket(\mathsf{lr}, \widehat{\mathsf{lr}}, \widetilde{\mathsf{lr}})\rrbracket$, ψ_3 for $\llbracket(\mathsf{rl}, \widetilde{\mathsf{rl}})\rrbracket \circ \llbracket(D, \hat{D}, D_2)\rrbracket = \llbracket(\mathsf{rz}, \widehat{\mathsf{rz}})\rrbracket$, ψ_4 for $\llbracket(\mathsf{ux}, \widehat{\mathsf{ux}})\rrbracket \circ \llbracket(D, \hat{D}, D_2)/(g_{1,\lambda_n}, \hat{g}_{1,\lambda_n}, g_{1,\lambda_n})\rrbracket = \llbracket(\mathsf{uz}, \widehat{\mathsf{uz}})/(g_{1,\lambda_n}, \hat{g}_{1,\lambda_n}, g_{1,\lambda_n})\rrbracket$, ψ_5 for $\llbracket(\mathsf{rz}, \widehat{\mathsf{rz}})\rrbracket \circ \llbracket(\mathsf{lr}, \widehat{\mathsf{lr}}, \mathsf{lr}_2)\rrbracket = \llbracket(D, \hat{D}) \cdot (\mathsf{uz}^{-1}, \widehat{\mathsf{uz}}^{-1})\rrbracket$, ψ_6 for $\tau(\llbracket(\mathsf{lr}, \widetilde{\mathsf{lr}})\rrbracket) = \llbracket(\mathsf{lr}, \widehat{\mathsf{lr}}, \widetilde{\mathsf{lr}})\rrbracket$, and ψ_7 for $\zeta^{-1}(\llbracket(\mathsf{ux}, \widetilde{\mathsf{ux}})\rrbracket) = \llbracket(\mathsf{lr}, \widehat{\mathsf{lr}}, \widetilde{\mathsf{lr}})\rrbracket$.
3. Send $\psi \leftarrow (\mathsf{lr}, \widehat{\mathsf{lr}}, \widetilde{\mathsf{lr}}, \mathsf{lr}_2, \mathsf{rl}, \widetilde{\mathsf{rl}}, \mathsf{rz}, \widehat{\mathsf{rz}}, \mathsf{uz}, \widehat{\mathsf{uz}}, \mathsf{ux}, \widehat{\mathsf{ux}}, \widetilde{\mathsf{ux}}, \psi_1, \dots, \psi_7)$ to the verifier.

Verification $\mathcal{V}(\mathsf{crs}; C, \psi)$: The verifier does the following:

- For $A \in \{\mathsf{lr}, \mathsf{rz}, \mathsf{uz}, \mathsf{ux}\}$ check that $\hat{e}(\hat{A}, g_2) = \hat{e}(A, \hat{g}_2)$.
- Check that $\hat{e}(g_1, \mathsf{lr}_2) = \hat{e}(\mathsf{lr}, g_2)$.
- For $A \in \{\mathsf{lr}, \mathsf{rl}, \mathsf{ux}\}$ check that $\hat{e}(\tilde{A}, g_2) = \hat{e}(A, \tilde{g}_2)$.
- Verify all 7 arguments $\psi_1, \dots, \psi_7$ with corresponding inputs.

Protocol 3: New NIZK argument for Circuit-SAT

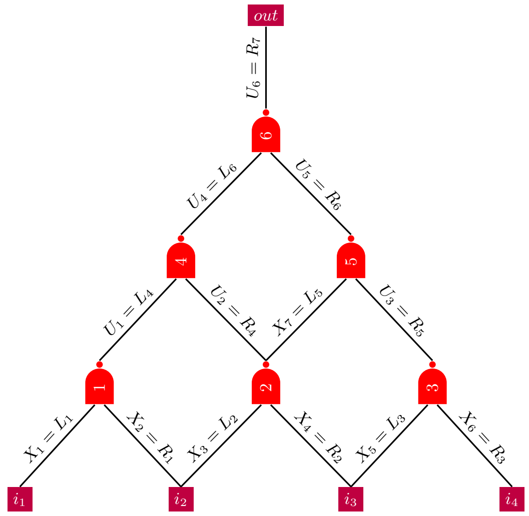

Fig. 1. Example circuit

$\cdots \to j_t + n \to i_1$. For unique wires i, $\tau(i) = i$. Second, $\zeta \in S_{2n+1}$ is a permutation that for every input wire (either L_i or R_{i-n}), outputs an index $j \leftarrow \zeta(i)$, such that the output wire U_j or X_{j-n} is equal to that input wire. Third, $\mathsf{swap} \in S_{2n+1}$ is a permutation, with $\mathsf{swap}(i) = i + n$ and $\mathsf{swap}(i + n) = i$ for $i \in [n]$, and $\mathsf{swap}(2n + 1) = 2n + 1$. Note that $\mathsf{swap} = \mathsf{swap}^{-1}$.

The argument is given by Prot. 3. In every subargument used in Prot. 3, the prover and the verifier use a substring of crs as the CRS. The corresponding substrings are easy to compute, and in what follows, we do not mention this issue. Instead of computing two different commitments $\mathcal{C}\mathrm{om}^t(\widehat{\mathsf{ck}}_t; \boldsymbol{a}; r) = (g_t^r \cdot \prod g_{t,\lambda_i}^{a_i}, \hat{g}_t^r \cdot \prod \hat{g}_{t,\lambda_i}^{a_i})$ and $\mathcal{C}\mathrm{om}^t(\widetilde{\mathsf{ck}}_t; \boldsymbol{a}; r) = (g_t^r \cdot \prod g_{t,\lambda_i}^{a_i}, \tilde{g}_t^r \cdot \prod \tilde{g}_{t,\lambda_i}^{a_i})$, we sometimes compute a composed commitment $\mathcal{C}\mathrm{om}^t(\mathsf{ck}_t; \boldsymbol{a}; r) = (g_t^r \cdot \prod g_{t,\lambda_i}^{a_i}, \hat{g}_t^r \prod \hat{g}_{t,\lambda_i}^{a_i}, \tilde{g}_t^r \cdot \prod \tilde{g}_{t,\lambda_i}^{a_i})$. We assume that the same value $\hat{\alpha}$ is used when creating product arguments and permutation arguments.

Theorem 8. *Let $\mathcal{G}_{\mathsf{bp}}$ be $\tilde{\Lambda}$-PSDL secure, and Λ-PKE secure in both $\mathbb{G}_1$ and $\mathbb{G}_2$. Then Prot. 3 is a perfectly complete, computationally adaptively sound and perfectly zero-knowledge non-interactive Circuit-SAT argument.*

Proof. PERFECT COMPLETENESS: follows from the perfect completeness of the Hadamard product and permutation arguments.

ADAPTIVE COMPUTATIONAL SOUNDNESS: Let $\mathcal{A}$ be a non-uniform PPT adversary that creates a circuit C and an accepting NIZK argument ψ. By the Λ-PKE assumption, there exists a non-uniform PPT extractor $X_{\mathcal{A}}$ that, running on the same input and seeing $\mathcal{A}$'s random tape, extracts all openings. From the (weaker version of) soundness of the product and permutation arguments and by the $\tilde{\Lambda}$-PSDL assumption, it follows that the corresponding relations are satisfied between the opened values. Moreover, by the $\tilde{\Lambda}$-PSDL assumption, the opened values belong to corresponding sets $\hat{\Lambda}$ and $\tilde{\Lambda}$. Let $(\boldsymbol{L}, \boldsymbol{R}, R_{n+1})$ be the opening of $(\mathsf{lr}, \widehat{\mathsf{lr}})$, where $\boldsymbol{L} = (L_1, \dots, L_n)$ and $\boldsymbol{R} = (R_1, \dots, R_n)$, and let $(U_1, \dots, U_n, X_1, \dots, X_n, X_{n+1})$ be the opening of $(\mathsf{ux}, \widehat{\mathsf{ux}})$. We now analyze the effect of every subargument in Prot. 3. See Fig. 1 for a visual reference.

The successful verification of $\hat{e}(g_1, \mathsf{lr}_2) = \hat{e}(\mathsf{lr}, g_2)$ shows that lr_2 is correctly formed. The first argument ψ_1 shows that $L_i, R_i \in \{0, 1\}$. The second argument ψ_2 shows that $(\mathsf{rl}, \widetilde{\mathsf{rl}})$ commits to $(\boldsymbol{R}, \boldsymbol{L}, R_{n+1})$. The third argument ψ_3 shows that $(\mathsf{rz}, \widehat{\mathsf{rz}})$ commits to $(R, 0, \dots, 0, 0)$ and is thus consistent with the opening of $(\mathsf{lr}, \widehat{\mathsf{lr}})$. The fourth argument ψ_4 shows that $(\mathsf{uz}, \widehat{\mathsf{uz}})$ commits to $(U_1, \dots, U_{n-1}, U'_n, 0, \dots, 0, 0)$ for some U'_n. It also shows that $U_n \cdot 0 = U'_n - 1$, and thus $U'_n = 1$. (The value of U_n is not important to get soundness, since it is not used in any other argument.)

The fifth argument shows ψ_5 that the NAND gates are followed. That is, $\neg(L_i \wedge R_i) = U_i$ for $i \in [n - 1]$. It also shows that the circuit outputs 1. Really, since $(\mathsf{uz}, \widehat{\mathsf{uz}})$ commits to $(U_1, \dots, U_{n-1}, U'_n = 1, 0, \dots, 0, 0)$, then $(D, \hat{D}) \cdot (\mathsf{uz}^{-1}, \widehat{\mathsf{uz}}^{-1})$ commits to $(1 - U_1, \dots, 1 - U_{n-1}, 1 - 1 = 0, 0, \dots, 0, 0)$. Thus, the Hadamard product argument verifies only if $L_i \cdot R_i = 1 - U_i$ for $i \in [n - 1]$, and $L_n \cdot R_n = 0$, that is, $\neg(L_n \wedge R_n) = 1$.

The sixth argument ψ_6 shows that if $i_1, \dots, i_s, j_1 + n, \dots, j_t + n$ correspond to the same wire, then $L_{i_1} = \cdots = L_{i_s} = R_{j_1} = \cdots = R_{j_t}$, that is, the values are internally consistent with the wires. As an example, in Fig. 1, the permutation τ contains cycles $R_1 \to L_2 \to R_1$, $R_2 \to L_3 \to R_2$, and $R_4 \to L_5 \to R_4$. The seventh argument ψ_7 shows that the "input wires" and "output" wires are consistent. In Fig. 1, $\zeta(L_1) = X_1$, $\zeta(R_1) = X_2$, etc.

PERFECT ZERO-KNOWLEDGE: we construct the next simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$. The simulator $\mathcal{S}_1(1^\kappa, n)$ creates a correctly formed CRS together with a simulation trapdoor $\mathsf{td} = (\hat{\alpha}, \tilde{\alpha}, x) \in \mathbb{Z}_p^3$. The adversary then outputs a statement C (a circuit) together with a witness (a satisfying assignment) w. The simulator $\mathcal{S}_2(\mathsf{crs}; C, \mathsf{td})$ creates $(\mathsf{lr}, \widehat{\mathsf{lr}}, \widetilde{\mathsf{lr}}, \mathsf{lr}_2)$, $(\mathsf{rl}, \widetilde{\mathsf{rl}})$, $(\mathsf{rz}, \widehat{\mathsf{rz}})$, $(\mathsf{uz}, \widehat{\mathsf{uz}})$ and $(\mathsf{ux}, \widehat{\mathsf{ux}})$ as commitments to $(0, \dots, 0)$. Due to the knowledge of trapdoor td, the simulator can simulate all product and permutation arguments. More precisely, he uses $L_i = R_i = U_i = U'_n = 1$ to simulate all product and permutation arguments, except in the case of ψ_5 where he uses $U_i = U'_n = 0$ instead. (Obviously, $(\mathsf{rz}, \widehat{\mathsf{rz}})$ and $(\mathsf{uz}, \widehat{\mathsf{uz}})$ commit to consistent tuples.)

To show that this argument ψ'' simulates the real argument ψ, note that ψ is perfectly indistinguishable from the simulated NIZK argument ψ' where one makes trapdoor commitments but opens them to *real* witnesses L_i, R_i when making product and permutation arguments. On the other hand, also ψ' and ψ'' are perfectly indistinguishable, and thus so are ψ and ψ''. □

Theorem 9. *Let Λ be chosen as in Thm. 1. The CRS consists of $|C|^{1+o(1)}$ group elements. The communication (argument length) of the argument in Prot. 3 is 18 elements from $\mathbb{G}_1$ and 21 elements from $\mathbb{G}_2$. The prover's computational complexity is dominated by $\Theta(|C|^2)$ simple arithmetical operations in $\mathbb{Z}_p$ and $|C|^{1+o(1)}$ exponentiations in $\mathbb{G}$. The verifier's computational complexity is dominated by 72 bilinear pairings and $8|C| + 8$ bilinear-group multiplications.*

The proof of this theorem can be found in App. L.

Moreover, the CRS depends on $\hat{\Lambda} \cup \tilde{\Lambda}$. Since 0 may or may not belong to $\tilde{\Lambda}$ (this depends on the choice of Λ) and $\Lambda \cup 2^\frown\Lambda \subseteq \tilde{\Lambda}$, $\hat{\Lambda} \cup \tilde{\Lambda} = \{0\} \cup \tilde{\Lambda}$. Recalling that elements of $\mathbb{G}_1$ can be represented by 512 bits and elements of $\mathbb{G}_2$ can be represented by 256 bits, the communication (argument length) is $18 \cdot 512 + 21 \cdot 256 = 14\,592$ bits.

Acknowledgments. We would like to thank Jens Groth, Amit Sahai, Naomi Benger, Bingsheng Zhang, Rafik Chaabouni, and anonymous reviewers for comments. The author was supported by Estonian Science Foundation, grant #9303, and European Union through the European Regional Development Fund.

References

[AF07] Masayuki Abe and Serge Fehr. Perfect NIZK with Adaptive Soundness. In Salil Vadhan, editor, *TCC 2007*, volume 4392 of *LNCS*, pages 118–136, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Heidelberg.

[Beh46] Felix A. Behrend. On the Sets of Integers Which Contain No Three in Arithmetic Progression. *Proceedings of the National Academy of Sciences*, 32(12):331–332, December 1946.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-Interactive Zero-Knowledge and Its Applications. In *STOC 1988*, pages 103–112, Chicago, Illinois, USA, May 2–4, 1988. ACM Press.

[BN05] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-Friendly Elliptic Curves of Prime Order. In Bart Preneel and Stafford E. Tavares, editors, *SAC 2005*, volume 3897 of *LNCS*, pages 319–331, Kingston, ON, Canada, August 11–12, 2005. Springer, Heidelberg.

[Bou98] Jean Bourgain. On Triples in Arithmetic Progression. *Geometric and Functional Analysis*, 9(5):968–984, 1998.

[BP04] Mihir Bellare and Adriana Palacio. The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols. In Matthew K. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 273–289, Santa Barbara, USA, August 15–19, 2004. Springer, Heidelberg.

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The Random Oracle Methodology, Revisited. In *STOC 1998*, pages 209–218, New York, May 23–26, 1998.

[Che06] Jung Hee Cheon. Security Analysis of the Strong Diffie-Hellman Problem. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 1–11, St. Petersburg, Russia, May 28–June 1, 2006. Springer, Heidelberg.

[CLs10] Rafik Chaabouni, Helger Lipmaa, and abhi shelat. Additive Combinatorics and Discrete Logarithm Based Range Protocols. In Ron Steinfeld and Philip Hawkes, editors, *ACISP 2010*, volume 6168 of *LNCS*, pages 336–351, Sydney, Australia, July 5–7, 2010. Springer, Heidelberg.

[CLZ12] Rafik Chaabouni, Helger Lipmaa, and Bingsheng Zhang. A Non-Interactive Range Proof with Constant Communication. In Angelos Keromytis, editor, *FC 2012*, volume ? of *LNCS*, pages ?–?, Bonaire, The Netherlands, February 27–March 2, 2012. Springer-Verlag.

[Dam91] Ivan Damgård. Towards Practical Public Key Systems Secure against Chosen Ciphertext Attacks. In Joan Feigenbaum, editor, *CRYPTO 1991*, volume 576 of *LNCS*, pages 445–456, Santa Barbara, California, USA, August 11–15, 1991. Springer, Heidelberg, 1992.

[dB64] Nicolaas Govert de Bruijn. Some Direct Decompositions of the Set of Integers. *Mathematics of Computation*, 18:537–546, 1964.

[DL08] Giovanni Di Crescenzo and Helger Lipmaa. Succinct NP Proofs from an Extractability Assumption. In Arnold Beckmann, Costas Dimitracopoulos, and Benedikt Löwe, editors, *Computability in Europe, CIE 2008*, volume 5028 of *LNCS*, pages 175–185, Athens, Greece, June 15–20, 2008. Springer, Heidelberg.

[DN00] Cynthia Dwork and Moni Naor. Zaps and Their Applications. In *FOCS 2000*, pages 283–293, Redondo Beach, California, USA, November 12–14, 2000. IEEE Computer Society Press.

[Elk11] Michael Elkin. An Improved Construction of Progression-Free Sets. *Israeli Journal of Mathematics*, 184:93–128, 2011.

[ET36] Paul Erdős and Paul Turán. On Some Sequences of Integers. *Journal of the London Mathematical Society*, 11(4):261–263, 1936.

[Gen09] Craig Gentry. Fully Homomorphic Encryption Using Ideal Lattices. In Michael Mitzenmacher, editor, *STOC 2009*, pages 169–178, Bethesda, Maryland, USA, May 31 — June 2, 2009. ACM Press.

[GJM02] Philippe Golle, Stanislaw Jarecki, and Ilya Mironov. Cryptographic Primitives Enforcing Communication and Storage Complexity. In Matt Blaze, editor, *FC 2002*, volume 2357 of *LNCS*, pages 120–135, Southhampton Beach, Bermuda, March 11–14, 2002. Springer-Verlag.

[GK03] Shafi Goldwasser and Yael Tauman Kalai. On the (In)security of the Fiat-Shamir Paradigm. In *FOCS 2003*, pages 102–113, Cambridge, MA, USA, October, 11–14 2003. IEEE, IEEE Computer Society Press.

[GL07] Jens Groth and Steve Lu. A Non-interactive Shuffle with Pairing Based Verifiability. In Kaoru Kurosawa, editor, *ASIACRYPT 2007*, volume 4833 of *LNCS*, pages 51–67, Kuching, Malaysia, December 2–6, 2007. Springer, Heidelberg.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The Knowledge Complexity of Interactive Proof-Systems. In *STOC 1985*, pages 291–304, Providence, Rhode Island, USA, May 6–8, 1985. ACM Press.

[Gro09] Jens Groth. Linear Algebra with Sub-linear Zero-Knowledge Arguments. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 192–208, Santa Barbara, California, USA, August 16–20, 2009. Springer, Heidelberg.

[Gro10] Jens Groth. Short Pairing-Based Non-interactive Zero-Knowledge Arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340, Singapore, December 5–9 2010. Springer, Heidelberg.

[Gro11] Jens Groth. Minimizing Non-interactive Zero-Knowledge Proofs Using Fully Homomorphic Encryption. Technical Report 2011/012, International Association for Cryptologic Research, January 6, 2011. Available at http://eprint.iacr.org/2011/012.

[GS08] Jens Groth and Amit Sahai. Efficient Non-interactive Proof Systems for Bilinear Groups. In Nigel Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432, Istanbul, Turkey, April 13–17, 2008. Springer, Heidelberg.

[GW10] Ben Green and Julia Wolf. A Note on Elkin's Improvement of Behrend's Construction. In David Chudnovsky and Gregory Chudnovsky, editors, *Additive Number Theory*, pages 141–144. Springer New York, 2010.

[GW11] Craig Gentry and Daniel Wichs. Separating Succinct Non-Interactive Arguments from All Falsifiable Assumptions. In Salil Vadhan, editor, *STOC 2011*, pages 99–108, San Jose, California, USA, June 6–8, 2011. ACM Press.

[HSV06] Florian Hess, Nigel P. Smart, and Frederik Vercauteren. The Eta Pairing Revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, 2006.

[LZ11] Helger Lipmaa and Bingsheng Zhang. A More Efficient Computationally Sound Non-Interactive Zero-Knowledge Shuffle Argument. Technical Report 2011/394, International Association for Cryptologic Research, July 21, 2011. Available at http://eprint.iacr.org/2011/394.

[Mic94] Silvio Micali. CS Proofs. In Shafi Goldwasser, editor, *FOCS 1994*, pages 436–453, Los Alamitos, California, USA, November 1994. IEEE, IEEE Computer Society Press.

[Mos62] Leo Moser. An Application of Generating Series. *Mathematics Magazine*, 35(1):37–38, January 1962.

[Pip80] Nicholas Pippenger. On the Evaluation of Powers and Monomials. *SIAM Journal of Computing*, 9(2):230–250, 1980.

[PSNB11] C. C. F. Pereira Geovandro, Marcos A. Simplício Jr., Michael Naehrig, and Paulo S. L. M. Barreto. A Family of Implementation-Friendly BN Elliptic Curves. *Journal of Systems and Software*, 84(8):1319–1326, 2011.

[San11] Tom Sanders. On Roth's Theorem on Progressions. *Annals of Mathematics*, 174(1):619–636, July 2011.

[Sch80] Jacob T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *Journal of the ACM*, 27(4):701–717, 1980.

[Seo11] Jae Hong Seo. Round-Efficient Sub-linear Zero-Knowledge Arguments for Linear Algebra. In Dario Catalano, Nelly Fazio, Rosario Gennaro, and Antonio Nicolosi, editors, *PKC 2011*, volume 6571 of *LNCS*, pages 387–402, Taormina, Italy, March 6–9, 2011. Springer, Heidelberg.

[Str64] Ernst G. Straus. Addition Chains of Vectors. *American Mathematical Monthly*, 70:806–808, 1964.

[TV06] Terrence Tao and Van Vu. *Additive Combinatorics*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2006.

[vHN10] Mark van Hoeij and Andrew Novocin. Gradual Sub-lattice Reduction and a New Complexity for Factoring Polynomials. In Alejandro López-Ortiz, editor, *LATIN 2010*, volume 6034 of *LNCS*, pages 539–553, Oaxaca, Mexico, April 19–23, 2010. Springer, Heidelberg.

A CPDH Assumption

Computational Power Diffie-Hellman Assumption. Following [Gro10], we say that a bilinear group generator $\mathcal{G}_{\mathsf{bp}}$ is *$q(\kappa)$-CPDH secure in group $\mathbb{G}_t$* for $t \in \{1, 2\}$, if for any non-uniform PPT adversary $\mathcal{A}$ and for any $j \in [q]$,

$$\Pr\left[\begin{array}{l}\mathsf{gk} := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}) \leftarrow \mathcal{G}_{\mathsf{bp}}(1^\kappa), g_t \leftarrow \mathbb{G}_t \setminus \{1\}, (x, \alpha) \leftarrow \mathbb{Z}_p^2 : \\ \mathcal{A}(\mathsf{gk}; (g_t^{x^\ell})_{\ell \in \{0,\ldots,q\}}, (g_t^{\alpha x^\ell})_{\ell \in \{0,\ldots,q\} \setminus \{j\}}) = g_t^{\alpha x^j}\end{array}\right]$$

is negligible in κ. As shown in [Gro10], q-CPDH holds in generic group model for any polynomial q. Note that in [Gro10], $\mathbb{G}_1 = \mathbb{G}_2$, while here, the two groups are different, and the q-CPDH assumption is only required to be true in one of them. This does not change the security proof in the generic group model. A related assumption in a non pairing-based group (called power computationally Diffie-Hellman assumption, and only for $i = q$) was defined in [GJM02], followed by many related assumptions (like Strong Diffie Hellman, Bilinear Diffie-Hellman Inversion) in subsequent years in pairing-based groups.

B Proof of Elkin's Result

Theorem 10. *For any fixed positive integer n, there exists $N = n^{1+o(1)}$, such that $r_3^{odd}(N) = n$.*

Proof. For any $S \subset \mathbb{R}^d$, let $\mathrm{vol}(S)$ be the volume of S. Let $\mathbb{T}^d = \mathbb{R}^d/\mathbb{Z}^d$ be the d-dimensional torus, we identify $\mathbb{T}^d$ with $[0,1)^d$ in an obvious way. For each $r \leq \frac{1}{2} \cdot \sqrt{d}$, let

$$S_d(r,\delta) := \{X \in [0,1/2]^d : r - \delta \leq ||X||_2 \leq r\} \ .$$

We need the next technical lemma, proof of which is given in App. C.

Lemma 4. *For some choice of $r \sim \sqrt{\frac{d}{12}}$, $\mathrm{vol}(S_d(r,\delta)) \geq \frac{4\sqrt{5}}{3}\delta \cdot 2^{-d} \cdot (1 - O(d^{-1/2}))$.*

Denote $S := S_d(r,\delta)$ for this choice of r and δ. Since there is no overlap, we identify $[0,1)^d$ with $\mathbb{T}^d$, and S with the corresponding subset of $\mathbb{R}^d$.

Now, let x and y be such that $x-y, x, x+y$ lie in S. But then $2\,||x||_2^2 + 2\,||y||_2^2 = ||x+y||_2^2 + ||x-y||_2^2$, and thus

$$\begin{aligned} ||y||_2 =& \sqrt{(||x+y||_2^2 + ||x-y||_2^2 - 2\,||x||_2^2)/2} \leq \sqrt{r^2 - (r-\delta)^2} = \sqrt{2\delta r - \delta^2} \\ \leq& \sqrt{2\delta r} \ . \end{aligned}$$

The volume of d-dimensional ball $B_d(R) := \{x \in \mathbb{R}^d : ||x||_2 \leq R\}$ is

$$\mathrm{vol}(B_d(R)) = \frac{\pi^{d/2}}{\Gamma(d/2+1)} \cdot R^d \approx \frac{(2e\pi)^{d/2}}{\sqrt{\pi d} \cdot d^{d/2}} \cdot R^d \ ,$$

where $\Gamma(x)$ is the standard Γ function, and we have used Stirling's approximation. Thus,

$$\mathrm{vol}(B_d(\sqrt{2\delta r})) \approx \frac{(2e\pi)^{d/2}}{\sqrt{\pi d} \cdot d^{d/2}} \cdot (2\delta r)^{d/2} = \frac{1}{\sqrt{\pi d}} \cdot (4e\pi\delta r/d)^{d/2} \ .$$

The volume of the region $B \subset \mathbb{T}^d \times \mathbb{T}^d$ where all such values (x,y) belong cannot be larger than $\mathrm{vol}(S) \cdot \mathrm{vol}(B_d(\sqrt{2\delta r}))$. Since

$$r \leq \sqrt{\frac{d}{12}} + \frac{1}{2\sqrt{5}} + O(d^{-1/2}) = \sqrt{\frac{d}{12}} + O(1) \ ,$$

then

$$\begin{aligned} \mathrm{vol}(B_d(\sqrt{2\delta r})) \approx& \frac{1}{\sqrt{\pi d}} \cdot \left(\frac{4e\pi\delta r}{d}\right)^{d/2} \leq \frac{1}{\sqrt{\pi d}} \cdot \left(\frac{4e\pi\delta(\sqrt{\frac{d}{12}} + O(1))}{d}\right)^{d/2} \\ =& \frac{1}{\sqrt{\pi d}} \cdot \left(\frac{2e\pi\delta(1 + O(d^{-1/2}))}{\sqrt{3d}}\right)^{d/2} \\ =& \frac{1}{\sqrt{\pi d}} \cdot \left(\frac{2e\pi\delta}{\sqrt{3d}}\right)^{d/2} (1 + O(d^{-1/2})) \ . \end{aligned}$$

We have established that $\mathbb{T}^d$ has a relatively large progression-free area. We now show how to use this result to establish a similar result for $[N]$. Let $\Psi_{\theta,\alpha} : [N] \to \mathbb{T}^d$ be defined as $\Psi_{\theta,\alpha}(s) = \theta s + \alpha \mod 1$. If s is an integer, then $\Psi_{\theta,\alpha}(s)$ clearly is uniformly distributed on $\mathbb{T}^d$ as θ, α vary uniformly and independently on $\mathbb{T}^d$. Moreover, one can show [GW10] that when s and s' are distinct positive integers, then $(\Psi_{\theta,\alpha}(s), \Psi_{\theta,\alpha}(s'))$ is uniformly distributed on $\mathbb{T}^d \times \mathbb{T}^d$ as θ and α vary uniformly and independently over $\mathbb{T}^d$. Define $A_{\theta,\alpha} := \{s \in [N] : \Psi_{\theta,\alpha}(s) \in S\}$. Then due to the uniform distribution of $\Psi_{\theta,\alpha}(s)$, $\mathbb{E}_{\theta,\alpha}|A_{\theta,\alpha}| = N \cdot \mathrm{vol}(S)$.

Next, let $T(A_{\theta,\alpha})$ be the number of nontrivial 3-term arithmetic progressions in $A_{\theta,\alpha}$ and assume that N is even. Each nontrivial 3-term progression is of the form $(x-y, x, x+y)$ with $y \neq 0$. Since N is even, the possible values of y range from 1 to $N/2 - 1$. For every y, the possible values of x range from $y+1$ to $N-y$. Thus, there are $\sum_{y=1}^{N/2-1}(N-2y) = N(N-2)/4$ possible such 3-term progressions. The probability that an arbitrary such progression (x, y) belongs to $A_{\theta,\alpha}$ is $\mathrm{vol}(B)$, and thus $\mathbb{E}_{\theta,\alpha}[|T(A_{\theta,\alpha})|] = \frac{1}{4} \cdot N(N-2)\,\mathrm{vol}(B)$.

Now, for a moment, assume that

$$\frac{1}{4}(N-2)\,\mathrm{vol}(B) \leq \frac{1}{3}\,\mathrm{vol}(S) \ . \tag{7}$$

Then

$$\begin{aligned}\mathbb{E}[\frac{2}{3}|A_{\theta,\alpha}| - T(A_{\theta,\alpha})] =& \mathbb{E}[\frac{2}{3}|A_{\theta,\alpha}|] - \mathbb{E}[T(A_{\theta,\alpha})] \\ =& \frac{2}{3} \cdot N\,\mathrm{vol}(S) - \frac{1}{4} \cdot N(N-2)\,\mathrm{vol}(B) \geq \frac{1}{3} \cdot N\,\mathrm{vol}(S) \ .\end{aligned}$$

Clearly, thus there is a choice $A = A_{\theta,\alpha}$ for which both $T(A) \leq 2|A|/3$ and $|A| \geq \frac{1}{2} \cdot N\,\mathrm{vol}(S)$. Deleting up to $2/3$ of the elements of A, we get a set of size at least $\frac{1}{6} \cdot N\,\mathrm{vol}(S)$ that is progression-free.

For Eq. (7) to hold, we must have

$$\frac{1}{3} \geq \frac{1}{4}(N-2)\,\mathrm{vol}(B_d(\sqrt{2\delta r})) \ ,$$

or

$$\frac{1}{\sqrt{\pi d}} \cdot \left(\frac{2e\pi\delta}{\sqrt{3d}}\right)^{d/2} (1 + O(d^{-1/2})) \leq \frac{4}{3(N-2)} = \frac{4}{3N} \cdot (1 - O(N^{-1})) \ ,$$

or

$$\begin{aligned}\frac{1}{\sqrt{\pi d}} \cdot \left(\frac{2e\pi\delta}{\sqrt{3d}}\right)^{d/2} \leq& \frac{4}{3N} \cdot (1 - O(N^{-1}))(1 - O(d^{-1/2})) \\ =& \frac{4}{3N} \cdot (1 - O(d^{-1/2})) \ .\end{aligned}$$

(Here, we assumed $d \ll \sqrt{N}$.) The latter is true whenever

$$\delta \leq \frac{\sqrt{3d}}{2e\pi} \cdot \left(\frac{16\pi d}{9(1 + O(d^{-1/2}))^2 N^2}\right)^{1/d} = \frac{\sqrt{3d}}{2e\pi} \cdot \left(\frac{16\pi d}{9N^2}\right)^{1/d} \cdot (1 - O(d^{-1/2})) \ ,$$

and we can take δ to be equal to the right-hand side of this inequality. But then

$$\begin{aligned}\frac{1}{6}N\,\mathrm{vol}(S) =& \frac{N}{6} \cdot \frac{4\sqrt{5}\delta}{3}(1 - O(d^{-1/2})) \cdot \frac{1}{2^d} \\ =& \frac{N}{6} \cdot \frac{4\sqrt{5}(1 - O(d^{-1/2}))}{3 \cdot 2^d} \cdot \frac{\sqrt{3d}(1 - O(d^{-1/2}))}{2e\pi} \cdot \left(\frac{16\pi d}{9N^2}\right)^{1/d} \\ =& \frac{\sqrt{5}}{3\sqrt{3} \cdot e\pi} \cdot \frac{N^{1-2/d}\sqrt{d}}{2^d} \cdot \left(\frac{16\pi d}{9}\right)^{1/d} \cdot (1 - O(d^{-1/2})) \\ \geq& \frac{\sqrt{5}}{3\sqrt{3} \cdot e\pi} \cdot \frac{N^{1-2/d}\sqrt{d}}{2^d} \cdot (1 - O(d^{-1/2})) \ .\end{aligned}$$

This is approximately maximized if $d = \sqrt{2\log_2 N}$, then

$$\frac{1}{6}N\,\mathrm{vol}(S) \geq \frac{\sqrt[4]{2}\sqrt{5}}{3\sqrt{3}e\pi} \cdot \frac{N}{2^{2\sqrt{2}\sqrt{\log_2 N}}} \cdot (\log_2 N)^{1/4} \cdot (1 - O(\log^{-1/4} N)) \ .$$

Thus, $r_3(N) = \Theta(N \log_2^{1/4} N / 2^{2\sqrt{2\log_2 N}})$, or $r_3(N) = N^{1-O(1)}$. This proves the theorem. □

One can obtain a better expression of N from $r_3(N)$, though it will not be precise. Namely, assume $n = r_3(N)$. Since $n = \Omega(N/2^{2\log_2 N})$, we have $N = O(n \cdot 2^{4+2\sqrt{2+2\log_2 n}}) = O(n \cdot 2^{2\sqrt{2(2+\log_2 n)}})$. We will leave it as an open question to derive a tighter lowerbound on N.

C Proof of Lem. 4

Proof. For $i \in [d]$, let $X_i \in \mathbb{R}$ be a uniformly random variable in $[0, 1/2]$. Let $||X||_2 = \sqrt{\sum_{i=1}^{d} X_i}$ be the Euclidean norm of $X = (X_1, \ldots, X_d)$. Then

$$\begin{aligned}\mathbb{E}[X_i^2] =& 2 \cdot \int_0^{1/2} x^2 dx = \frac{1}{12} \ , \\ \mathbb{E}[X_i^4] =& 2 \cdot \int_0^{1/2} x^4 dx = \frac{1}{80} \ , \\ \mathrm{var}[X_i^2] =& \mathbb{E}[X_i^4] - \mathbb{E}[X_i^2]^2 = \frac{1}{180} \ , \\ \mathbb{E}[||X||_2^2] =& \sum_{i=1}^{d} \mathbb{E}[X_i^2] = \frac{d}{12} \ , \\ \mathrm{var}[||X||_2^2] =& \sum_{i=1}^{d} \mathrm{var}[X_i^2] = \frac{d}{180} \ .\end{aligned}$$

By Chebyshev's inequality,

$$\Pr[|\ ||X||_2^2 - \mathbb{E}[||X||_2^2]| \geq t \cdot \sqrt{\mathrm{var}[||X||_2^2]}] \leq 1/t^2$$

for any $t > 0$. That is, $\Pr[|\ ||X||_2^2 - \frac{d}{12}| \geq t\sqrt{\frac{d}{180}}] \leq 1/t^2$. Thus, with probability at least $1 - 1/t^2$,

$$\frac{d}{12} + t\sqrt{\frac{d}{180}} \geq ||X||_2^2 \geq \frac{d}{12} - t\sqrt{\frac{d}{180}} \ ,$$

and thus also

$$\sqrt{\frac{d}{12} + t\sqrt{\frac{d}{180}}} \geq ||X||_2 \geq \sqrt{\frac{d}{12} - t\sqrt{\frac{d}{180}}} \ .$$

If $d \to \infty$, then

$$\sqrt{\frac{d}{12} \pm t\sqrt{\frac{d}{180}}} \to \sqrt{\frac{d}{12}} \pm \frac{t}{2\sqrt{15}} + O(d^{-1/2}) \ .$$

Thus, for large d, with probability $1 - 1/t^2$,

$$\sqrt{\frac{d}{12}} + \frac{t}{2\sqrt{15}} + O(d^{-1/2}) \geq ||X||_2 \geq \sqrt{\frac{d}{12}} - \frac{t}{2\sqrt{15}} + O(d^{-1/2}) \ ,$$

or equivalently,

$$\left| ||X||_2 - \sqrt{\frac{d}{12}} \right| \leq \frac{t}{2\sqrt{15}} + O(d^{-1/2}) \ .$$

Fix a small δ, $t/\sqrt{15} \geq \delta > 0$. Then, by the pigeonhole principle, there exists an r, such that with probability at least

$$\left(1 - \frac{1}{t^2}\right) \cdot \frac{\delta}{\frac{t}{2\sqrt{15}}(1 + O(d^{-1/2}))} = \left(1 - \frac{1}{t^2}\right) \cdot \frac{2\sqrt{15}\delta}{t} \cdot (1 - O(d^{-1/2})) \ ,$$

$||X||_2 \in [r - \delta, r]$. This is maximized when $t = \sqrt{3}$, then the corresponding probability is $4\sqrt{5}\delta/3 \cdot (1 - O(d^{-1/2})$. But then

$$\mathrm{vol}(S_d(r, \delta)) \geq \frac{4\sqrt{5}}{3} \cdot \delta \cdot 2^{-d} \cdot (1 - O(d^{-1/2}))$$

as required. □

D Proof of Thm. 2

Proof. In the generic group model, an adversary only performs generic group operations (multiplications in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, bilinear pairings, and equality tests). A generic adversary produces an element of $\mathbb{Z}_p$, which depends only on gk and $(g_1^{x^\ell}, g_2^{x^\ell})_{\ell \in \{0\} \cup \Lambda}$. The only information that the adversary gets is when an equality (collision) between two previously computed elements of either $\mathbb{G}_1$, $\mathbb{G}_2$ or $\mathbb{G}_T$ occurs. We prove that finding even a single collision is difficult even if the adversary can compute an arbitrary group element in unit time.

Assume that the adversary can find a collision $y = y'$ in group $\mathbb{G}_1$. Then it must be the case that $y = \prod_{\ell \in \{0\} \cup \Lambda} g_1^{a_\ell x^\ell}$ and $y' = \prod_{\ell \in \{0\} \cup \Lambda} g_1^{a'_\ell x^\ell}$ for some known values of a_ℓ and a'_ℓ. But then also

$$\sum_{\ell \in \{0\} \cup \Lambda} (a_\ell - a'_\ell) x^\ell \equiv 0 \pmod{p} \enspace .$$

Since the adversary does not know the actual representations of the group elements, it will perform the same group operations independently of x. Thus a_ℓ and a'_ℓ are independent of x. By the Schwartz-Zippel lemma [Sch80] modulo p, the probability that $\sum_{\ell \in \{0\} \cup \Lambda} (a_\ell - a'_\ell) x^\ell \equiv 0 \pmod{p}$ is equal to λ_n / p for randomly chosen a_ℓ and a'_ℓ. If the adversary works in polynomial time $\tau = \mathrm{poly}(\kappa)$, it can generate at most τ such group elements. The total probability that there exists a collision between any two generated group elements is thus upper bounded by $\binom{\tau}{2} \cdot \lambda_n / p$, and thus a successful adversary requires time $\Omega(\sqrt{p/\lambda_n})$ to produce one collision.

A similar bound holds for collisions in $\mathbb{G}_2$. In the case of $\mathbb{G}_T$, the pairing enables the adversary to compute up to τ different values

$$y = \hat{e}(g_1, g_2)^{\sum_{\ell \in \{0\} \cup \Lambda} \sum_{j \in \{0\} \cup \Lambda} a_{\ell j} x^{\ell + j}} \enspace ,$$

and thus we get an upper bound $\binom{\tau}{2} \cdot 2\lambda_n / p$, and thus a successful adversary requires time $\Omega(\sqrt{p/\lambda_n})$ to produce one collision. □

E Proof of Thm. 3

Proof. PERFECT HIDING: follows from the fact that the output of $\mathcal{C}\mathsf{om}^t$ is a random element of $\mathbb{G}_t$. COMPUTATIONAL BINDING: Assume that $\mathcal{A}$ is an adversary that can break the binding property with some non-negligible probability. We construct the next adversary $\mathcal{A}'$ against the Λ-PDL assumption in group $\mathbb{G}_t$ that works with the same probability. Let $\mathsf{gk} \leftarrow \mathcal{G}_{\mathsf{bp}}(1^\kappa)$, $g_t \leftarrow \mathbb{G}_t \setminus \{1\}$, and $x \leftarrow \mathbb{Z}_p$. The adversary $\mathcal{A}'$ gets $(\mathsf{gk}; (g_t^{x^\ell})_{\ell \in \{0\} \cup \Lambda})$ as her input. She creates a random $\hat{\alpha}' \leftarrow \mathbb{Z}_p$, and sets $\mathsf{ck}_t \leftarrow (\mathsf{gk}; (g_t^{x^\ell}, g_t^{\hat{\alpha}' x^\ell})_{\ell \in \{0\} \cup \Lambda})$. She forwards ck_t to $\mathcal{A}$. By definition, $\mathcal{A}(\mathsf{ck}_t)$ produces a tuple $(\boldsymbol{a}, r_a, \boldsymbol{b}, r_b)$ with $(\boldsymbol{a}, r_a) \neq (\boldsymbol{b}, r_b)$, such that

$$g_t^{r_a} \cdot \prod_{i \in [n]} g_t^{a_i x^{\lambda_i}} = g_t^{r_b} \cdot \prod_{i \in [n]} g_t^{b_i x^{\lambda_i}} \enspace .$$

But then

$$g_t^{r_a - r_b + \sum_{i=1}^n (a_i - b_i) x^{\lambda_i}} = 1 \enspace ,$$

and thus

$$r_a - r_b + \sum_{i=1}^n (a_i - b_i) x^{\lambda_i} \equiv 0 \pmod{p} \enspace .$$

The adversary has generated a non-trivial polynomial $f(X)$ such that $f(x) = 0$. One can now use an efficient polynomial factorization algorithm [vHN10] in $\mathbb{Z}_p[X]$ to find all $\lambda_n + 1$ possible roots of f. For one of those roots, say y, it must be the case that $g_t^{x^{\lambda_1}} = g_t^{y^{\lambda_1}}$. Set $x \leftarrow y$. Thus, $\mathcal{A}'$ has broken the Λ-PDL assumption in group $\mathbb{G}_t$.

EXTRACTABILITY: By the Λ-PKE assumption in group $\mathbb{G}_t$, for every committer $\mathcal{A}$ there exists an extractor $X_{\mathcal{A}}$ that can open the commitment in group $\mathbb{G}_t$, given access to $\mathcal{A}$'s inputs and random tape. Since the commitment scheme is computationally binding, then the extracted opening has to be the same that $\mathcal{A}$ used. □

F Proof of Thm. 5

Proof. By Lem. 2, the size of the CRS is $\Theta(|\hat{\Lambda}|) = n^{1+o(1)}$. From the CRS, the verifier clearly only needs to access g_1, $\hat{g}_1$, g_2, $\hat{g}_2$ and D. Since $2^\frown\Lambda \subseteq \hat{\Lambda}$, the statement about the prover's computational complexity follows. The prover's computational complexity comes from the fact that one can compute all values μ_ℓ simultaneously by looping over n^2 possible values of i and j. The verifier's computational complexity follows from the description of the argument. □

G Fast Boolean Product Argument

```
1. For $\ell \in 2^\frown\Lambda$ do: $\mu_\ell \leftarrow 0$
2. For $i = 1$ to $n$ do:
     – If $a_i = 0$ then for $j = 1$ to $n$ do: if $j \neq i$ then $\mu_{\lambda_i+\lambda_j} \leftarrow \mu_{\lambda_i+\lambda_j} - c_i$
     – Else for $j = 1$ to $n$ do: if $j \neq i$ then $\mu_{\lambda_i+\lambda_j} \leftarrow \mu_{\lambda_i+\lambda_j} + b_j - c_i$
```

H Derivation of Permutation Argument

Consider a verification equation of type $\hat{e}(A, D)/\hat{e}(B, E_\varrho) = \hat{e}(g_1, \psi^\varrho)$ where $D = g_2^{\sum_{i=1}^n x^{\varphi_i}}$ and $E_\varrho = g_2^{\sum_{i=1}^n x^{\xi_i}}$ for some φ_i and ξ_i to be specified later. Letting $h = \hat{e}(g_1, g_2)$ and $\eta_\varrho = \hat{e}(A, D)/\hat{e}(B, E_\varrho)$, we get

$$\begin{aligned}
\log_h \eta_\varrho =& (r_a + \sum_{i=1}^n a_i x^{\lambda_i})(\sum_{i=1}^n x^{\varphi_i}) - (r_b + \sum_{i=1}^n b_i x^{\lambda_i})(\sum_{i=1}^n x^{\xi_i}) \\
=& r_a \sum_{i=1}^n x^{\varphi_i} - r_b \sum_{i=1}^n x^{\xi_i} + \sum_{i=1}^n \sum_{j=1}^n a_{\varrho(i)} x^{\lambda_{\varrho(i)}+\varphi_{\varrho(j)}} - \sum_{i=1}^n \sum_{j=1}^n b_i x^{\lambda_i+\xi_j} \\
=& r_a \sum_{i=1}^n x^{\varphi_i} - r_b \sum_{i=1}^n x^{\xi_i} + \sum_{i=1}^n a_{\varrho(i)} x^{\lambda_{\varrho(i)}+\varphi_{\varrho(i)}} - \sum_{i=1}^n b_i x^{\lambda_i+\xi_i} + \\
& \sum_{i=1}^n \sum_{j=1:j\neq i}^n a_{\varrho(i)} x^{\lambda_{\varrho(i)}+\varphi_{\varrho(j)}} - \sum_{i=1}^n \sum_{j=1:j\neq i}^n b_i x^{\lambda_i+\xi_j} \enspace .
\end{aligned}$$

Since we are interested in the case $a_{\varrho(i)} = b_i$, it is natural to define

$$\xi_i := \lambda_{\varrho(i)} + \varphi_{\varrho(i)} - \lambda_i \enspace . \tag{8}$$

Then,

$$\begin{aligned}
\log_h \eta_\varrho =& \sum_{i=1}^n (a_{\varrho(i)} - b_i) x^{\lambda_{\varrho(i)}+\varphi_{\varrho(i)}} + r_a \sum_{i=1}^n x^{\varphi_i} - r_b \sum_{i=1}^n x^{\lambda_{\varrho(i)}+\varphi_{\varrho(i)}-\lambda_i} + \\
& \sum_{i=1}^n \sum_{j=1:j\neq i}^n a_{\varrho(i)} x^{\lambda_{\varrho(i)}+\varphi_{\varrho(j)}} - \sum_{i=1}^n \sum_{j=1:j\neq i}^n b_i x^{\lambda_i+\lambda_{\varrho(j)}+\varphi_{\varrho(j)}-\lambda_j} \enspace .
\end{aligned}$$

If one wants to use the same methodology as in Sect. 6, one has to define φ_i so that $\tilde{\Lambda} := \{\varphi_i\} \cup \{\lambda_{\varrho(i)} + \varphi_{\varrho(i)} - \lambda_i\} \cup \{\lambda_{\varrho(i)} + \varphi_{\varrho(j)} : i \neq j\} \cup \{\lambda_i + \lambda_{\varrho(j)} + \varphi_{\varrho(j)} - \lambda_j : i \neq j\}$ and $\{\lambda_{\varrho(i)} + \varphi_{\varrho(i)}\}$ will not intersect. To this end, it is natural to define $\varphi_i = k\lambda_i + k^*\lambda_{\varrho^{-1}(i)}$ for some k and k^*. Most of those linear combinations (not even talking about nonlinear combinations!) do not make the task simpler, and thus k and k^* should be chosen so that at least some of the terms in $\tilde{\Lambda}$ would simplify. For example[2], one can choose

$$\varphi_i = \lambda_i \enspace , \tag{9}$$

[2] We tried many other choices of φ_i, but none of them gave a simpler solution. Details are omitted due to the lack of space.

and we get

$$\begin{aligned}\log_h \eta_\varrho = &\sum_{i=1}^{n}(a_{\varrho(i)} - b_i)x^{2\lambda_{\varrho(i)}} + r_a\sum_{i=1}^{n} x^{\lambda_i} - r_b\sum_{i=1}^{n} x^{2\lambda_{\varrho(i)}-\lambda_i} + \\ &\sum_{i=1}^{n} a_{\varrho(i)} \cdot \sum_{\substack{j=1\\ j\neq i}}^{n} x^{\lambda_{\varrho(i)}+\lambda_{\varrho(j)}} - \sum_{i=1}^{n} b_i \cdot \sum_{\substack{j=1\\ j\neq i}}^{n} x^{\lambda_i+2\lambda_{\varrho(j)}-\lambda_j} \enspace .\end{aligned}$$

In this case, $\tilde{\Lambda} = \Lambda \cup \{2\lambda_{\varrho(i)} - \lambda_i\} \cup 2\hat{}\Lambda \cup \{\lambda_i + 2\lambda_{\varrho(j)} - \lambda_j : i \neq j\}$ and the second set is $2 \cdot \Lambda = \{2\lambda_{\varrho(i)}\}$. Since ϱ is not known when creating the CRS, we have to replace $\varrho(i)$ and $\varrho(j)$ with a new element k, so

$$\tilde{\Lambda} = \Lambda \cup \{2\lambda_k - \lambda_i\} \cup 2\hat{}\Lambda \cup \{\lambda_i + 2\lambda_k - \lambda_j : i \neq j\} \enspace .$$

(Note that choosing an arbitrary k may be an overkill, depending on the permutation ϱ.)

I Proof of Thm. 7

Proof. It is clear that with this choice of Λ, by Lem. 3, the size of the CRS is $\Theta(|\tilde{\Lambda}|) = n^{1+o(1)}$. Since $\tilde{\Lambda}'_\varrho \subseteq \tilde{\Lambda}$, the statement about the prover's computational complexity follows from the fact that one can compute all values $\mu_{\varrho,\ell}$ simultaneously by looping over n^2 possible values of i and j, as follows:

```
1. For $\ell \in 2\hat{}\Lambda$ do: $\mu_{\varrho,\ell} \leftarrow 0$
2. For $i = 1$ to $n$ do:
     – For $j = 1$ to $n$ do: if $j \neq i$ then $\mu_{\varrho,\lambda_i+\lambda_j} \leftarrow \mu_{\varrho,\lambda_i+\lambda_j} + a_i^*$
3. For $j = 1$ to $n$ do:
     – Set $t \leftarrow 2\lambda_{\varrho(j)} - \lambda_j$
     – For $i = 1$ to $n$ do: if $j \neq i$ and $2\lambda_{\varrho(i)} + \lambda_j \neq \lambda_i + 2\lambda_{\varrho(j)}$ then $\mu_{\varrho,t+\lambda_i} \leftarrow \mu_{\varrho,t+\lambda_i} - b_i$
```

Here, one has to do $\Theta(n^2)$ scalar additions and $\Theta(n)$ evaluations of ϱ (say, by using table-lookup). The verifier's computational complexity follows from the description of the argument. □

J On Impossibility of Improving over the Moser-de Bruijn Sequence

As we mentioned in Sect. 7, it is required that any non-negative integer a has at most one representation of type $a = 2\lambda_i + \lambda_j$. By using the techniques of [Mos62], it is easy to see that the Moser-de Bruijn sequence is the densest sequence also in this case.

As in [Mos62], define $f(x) = \sum_{i=0}^{\infty} x^{\lambda_i}$, where $(\lambda_i)_{i=0}^{\infty}$ is some (strictly increasing) sequence and $|x| < 1$. Then any non-negative integer has at most one unique representation $2\lambda_i + \lambda_j$ if

$$f(x)f(x^2) = \sum_{i=0}^{\infty} b_i x^i$$

for some $b_i \in \{0, 1\}$. For example, if it is required that there is precisely one representation, then $b_i = 1$ and $f(x)f(x^2) = 1/(1 - x)$. In this case,

$$\frac{f(x)f(x^2)}{f(x^2)f(x^4)} = \frac{1 - x^2}{1 - x} = 1 + x$$

and thus $f(x) = (1 + x)f(x^4)$. Continuing recursively, we get

$$f(x) = (1 + x)(1 + x^4)(1 + x^{16})(1 + x^{64}) \dots \enspace .$$

In the right-hand side, the coefficient of x^n is equal to the number of representations of n as the sum of distinct powers of 4. A number has at most one representation as a sum of distinct powers of 4 and thus $\{\lambda_i\}$ must consist precisely of those numbers that have such a representation. In this case, clearly $\lambda_n = \Theta(n^2)$.

Now, to construct a set $\{\lambda_i\}$ such that λ_n is smaller than $\Theta(n^2)$, we need that $f(x)/f(x^4) > 1 + x$ whenever $|x| < 1$. This is impossible under the assumption that $b_i \in \{0, 1\}$, since $\sum b_i x^i$ is maximized if $b_i = 1$.

K Comparison: Basic Arguments

Efficiency. In [Gro10], Groth used basically the same commitment scheme but with $\Lambda = \{i : i \in [n]\}$. In fact, he committed to $\boldsymbol{b}$ by using a different set $\Lambda' = \{i(n+1) : i \in [n]\}$. According to our terminology, in both of Groth's basic arguments $|\hat{\Lambda}| = \Theta(n^2)$ (this is since $|\{g^{i+j(n+1)} : i \in [n] \wedge j \in [n]\}| = \Theta(n^2)$), and thus the CRS length has $\Theta(n^2)$ group elements (since it contains $\{g^{x^\ell} : \ell \in \hat{\Lambda}\}$) but also in addition, it takes $\Theta(n^2)$ bilinear-group exponentiations to compute ψ. We emphasize that the difference between $\Theta(n^2)$ scalar additions (as in the new arguments) and $\Theta(n^2)$ bilinear-group exponentiations (as in Groth's argument) is very important in practice; and in both cases, Θ hides approximately the same constant. Moreover, the choice of Λ and Λ' in [Gro10] is somewhat ad hoc. One of the contributions of the current paper is the intuitive reasoning for the choice of Λ, and a near-optimal (up to a factor of 2!) construction of Λ. On the other hand, [Gro10] did not consider any other possible sets Λ at all.

The fact that we use the same n generators to commit to all A, B, and C means that we can more readily reuse these commitments in different arguments, even if B is a commitment in a different group. For example, to prove that $A \in \mathbb{G}_1$ and $B \in \mathbb{G}_2$ commit to the same tuple (by using the same randomizer), Groth [Gro10] had to use a separate Hadamard product argument. In our case, it is just sufficient to verify that $\hat{e}(A, g_2) = \hat{e}(g_1, B)$. Finally, in the case of the Hadamard product argument, we made the verifier's computational complexity to be $\Theta(1)$ pairings by adding 1 group element to the CRS. (In the permutation argument, one can also make the verifier's computational complexity to be $\Theta(1)$ pairings by adding a few elements to the CRS. However, this optimization is valid only when the permutation is previously known like the permutation swap in the case of the Circuit-SAT argument in Sect. 8.)

Differently from [Gro10], we use asymmetric pairings. While at least currently, asymmetric pairings are much more efficient than symmetric pairings, it also means that we have to take additional care by modifying several aspects of the security assumptions, of the arguments and of the proofs.

By giving a somewhat clearer proof, we were able to avoid Groth's CPDH assumption, and rely on the (potentially weaker) PSDL assumption instead. Moreover, since the adversary in both security proofs has to factorize a degree λ_n polynomial, our security proofs have a tighter reduction than those in [Gro10].

L Proof of Thm. 9

Proof. This argument contains 6 commitments, 4 product arguments and 3 permutation arguments. Thus, the Circuit-SAT argument consists of a number of group elements, $12 + 3 \cdot 2 = 18$ belong to $\mathbb{G}_1$ and $1 + 4 \cdot 2 + 3 \cdot 4 = 21$ belong to $\mathbb{G}_2$.

The verifier has to perform $2 \cdot (4 + 1 + 3) = 16$ bilinear pairings to check the correctness of all commitments, and $4 \cdot 5 + 3 \cdot 12 = 56$ bilinear pairings to check all 7 arguments. It takes $3 \cdot 12 = 36$ bilinear pairings to compute all permutation arguments, and thus $20 + 36 = 56$ bilinear pairings to check all 7 arguments. She also has to perform 1 bilinear-group multiplication to verify each of the 4 product arguments, and $4 \cdot (|C| + 1) - 2 = 4|C| + 2$ (online) bilinear-group multiplications to verify each of the two permutation arguments ψ_6 and ψ_7. (Since swap does not depend on the circuit, one can optimize the bilinear-group multiplications out in the case of ψ_2.)

Statements about the prover's computational complexity and the CRS size follow directly from the complexities of the basic arguments. For the prover's computational complexity note that all committed elements in product arguments are Boolean, and that we can assume that all three used permutations are simple to compute (e.g., ζ and τ can be computed by a table-lookup). Moreover, since all three permutation arguments share their second argument, one can optimize the permutation arguments by only performing certain computations once. □

The verifier also has to compute the values T_Λ, but since it does not require any bilinear-group multiplications, it will not change her computational complexity much.

M Perfect Zaps

The presented NIZK argument is in the CRS model, and thus requires some trusted third party to construct the CRS. If this is not possible, then one can instead construct a sublinear-size zap [DN00] (a 2-move publicly-verifiable witness-indistinguishable argument, where the verifier's first message can be reused polynomially many times), just by letting the verifier's first message to be equal to the CRS, and then letting the prover to verify the well-formedness of the CRS and then send his argument. It is straightforward to see that all CRS-s constructed in the current paper are publicly verifiable. This zap will be perfectly witness-indistinguishable. Due to Groth's balancing technique [Gro10], the new zap for circuit satisfiability has the communication complexity of $|C|^{1/2+o(1)}$ group elements, while Groth's zap from [Gro10] has the communication complexity of $\Theta(|C|^{2/3})$ group elements. Note that Di Crescenzo and Lipmaa [DL08] constructed a 2-message perfectly complete and computationally sound polylogarithmic argument for $\mathbf{NP}$ under a knowledge assumption (and without random oracles), but their argument is not witness-indistinguishable.