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 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 [4] 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 [22] 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.

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

	CRS length	Argument length	Prover comp.	Verifier comp.
		Random-oracle based arguments		
[14]	$O(\|C\|^{\frac{1}{2}})G$	$O(\|C\|^{\frac{1}{2}})G$	$O(\|C\|)M$	$O(\|C\|)M$
		Knowledge-assumption based arguments from [15]		
$m = 1$	$\Theta(\|C\|^2)G$	$42G$	$\Theta(\|C\|^2)E$	$\Theta(\|C\|)M + \Theta(1)P$
$m = n^{\frac{1}{3}}$	$\Theta(\|C\|^{\frac{2}{3}})G$	$\Theta(\|C\|^{\frac{2}{3}})G$	$\Theta(\|C\|^{\frac{4}{3}})E$	$\Theta(\|C\|)M + \Theta(\|C\|^{\frac{2}{3}})P$
		Knowledge-assumption based arguments from the current paper		
$m = 1$	$\|C\|^{1+o(1)}G$	$39G$	$\Theta(\|C\|^2)A + \|C\|^{1+o(1)}E$	$(8\|C\| + 8)M + 62P$
$m = n^{\frac{1}{3}}$	$\|C\|^{\frac{1}{3}+o(1)}G$	$\Theta(\|C\|^{\frac{2}{3}})G$	$\Theta(\|C\|^{\frac{4}{3}})A + \|C\|^{1+o(1)}E$	$\Theta(\|C\|)M + \Theta(\|C\|^{\frac{2}{3}})P$
$m = n^{\frac{1}{2}}$	$\|C\|^{\frac{1}{2}+o(1)}G$	$\Theta(\|C\|^{\frac{1}{2}})G$	$\Theta(\|C\|^{\frac{3}{2}})A + \|C\|^{1+o(1)}E$	$\Theta(\|C\|)M + \Theta(\|C\|^{\frac{1}{2}})P$

Another NIZK argument for Circuit-SAT, proposed by Groth in 2009 [14], is also based on the random oracle model. It is well-known that some functionalities are secure in the random oracle model and insecure in the plain model. 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 [10], one can construct efficient NIZK *proofs* for all **NP**-languages in communication that is linear to the witness size [16]. 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 [15] 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 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 [15] 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 [15].

A balanced version of the new argument achieves the combined CRS and argument of $\Theta(|C|^{1/2+o(1)})$ group elements. In the full version, we describe a zap for Circuit-SAT that has communication complexity of $|C|^{1/2+o(1)}$ group elements, while Groth's zap from [15] 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 [22].

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 [15], 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 [9], $N = O(n \cdot 2^{2\sqrt{2(2+\log_2 n)}}) = n^{1+o(1)}$. See Sect. 3 for further elaboration on our techniques.

Thus, one can build an argument of $\Theta(1)$ group elements for every language in **NP**, by reducing the task at hand to a Circuit-SAT instance. Obviously, one can often design more efficient tailor-made protocols, see [21,7] for some follow-up work. In particular, [7] used our basic arguments to construct a non-interactive range proof with communication of $\Theta(1)$ group elements, while [21] used our techniques to design a new basic argument to construct a non-interactive shuffle. (See [6] for a previous use of additive combinatorics in the construction of zero-knowledge proofs.)

Due to the lack of space, several proofs have been deferred to the full version [20].

2 Preliminaries

Let $[n] = \{1, 2, \ldots, n\}$. Let S_n be the set of permutations from $[n]$ to $[n]$. Let $\boldsymbol{a} = (a_1, \ldots, a_n)$. 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 $\Lambda = (\lambda_1, \ldots, \lambda_n) \subset \mathbb{Z}$ 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* [25]. 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 $2\widehat{\ }\Lambda = \{\lambda_1 + \lambda_2 : \lambda_1 \in \lambda \wedge \lambda_2 \in \Lambda \wedge \lambda_1 \neq \lambda_2\} \subseteq \Lambda + \Lambda$ is a *restricted sumset*. (See [25].)

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 [18] over a subclass of Barreto-Naehrig curves [2]. 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 $n = \mathrm{poly}(\kappa)$, 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 $n = \mathrm{poly}(\kappa)$ 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 $\mathcal{C}\mathrm{om}^t(\mathsf{ck}^*; \mathbf{0}; r)$, and (c) a trapdoor opening algorithm that takes ck^*, td, $\boldsymbol{a}$ and r as an input and outputs an r' such that $\mathcal{C}\mathrm{om}^t(\mathsf{ck}^*; \mathbf{0}; r) = \mathcal{C}\mathrm{om}^t(\mathsf{ck}^*; \boldsymbol{a}; r')$.

Let $\mathcal{R} = \{(C, w)\}$ be an efficiently computable binary relation such that $|w| = \mathrm{poly}(|C|)$. Here, C is a statement, and w is a witness. Let $\mathcal{L} = \{C : \exists w, (C, w) \in \mathcal{R}\}$ be an **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 $\forall n = \mathrm{poly}(\kappa)$,

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

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 $n = \mathrm{poly}(\kappa)$, 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 $n = \mathrm{poly}(\kappa)$, 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 $n = \mathrm{poly}(\kappa)$ (with td being the *simulation trapdoor*),

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

3 Our Techniques

We will first give a more precise overview of Groth's Hadamard product and permutation arguments [15], 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 [12], Groth [15] 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 [17], 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 $F(x)=(r_a+\sum_{i=1}^n a_ix^i)\cdot(r_b+\sum_{i=1}^n b_ix^{i(n+1)})-(r_c+\sum_{i=1}^n c_ix^i)\cdot(\sum_{i=1}^n x^{i(n+1)})$ is the sum of two formal polynomials in x, $F(x)=F_{\mathsf{con}}(x)+F_\psi(x)$, where $F_{\mathsf{con}}(x)=\sum_{i=1}^n(a_ib_i-c_i)x^{i(n+2)}$ is a *constraint polynomial*, spanned by the powers of x from $\Lambda_{\mathsf{con}}=\{i(n+2):i\in[n]\}$, and

$$F_\psi(x)=r_ar_b+r_b\sum_{i=1}^n a_ix^i+\sum_{i=1}^n(r_ab_i-r_c)x^{i(n+1)}+\sum_{i=1}^n\sum_{\substack{j=1\\ j\neq i}}^n(a_ib_j-c_i)x^{i+j(n+1)}$$

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 $a_ib_i=c_i$ 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 $\psi=g^{r_ar_b}\cdot\cdots\cdot\prod_{i=1}^n\prod_{j=1:i\neq j}^n g_{i+j(n+1)}^{a_ib_j-c_i}$. 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_ar_b, …, $a_ib_j-c_i$. 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 $a_ib_i\neq c_i$ 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 [15], 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 $(g_1, \ldots, g_n)$ and $(g_{n+1}, \ldots, g_{n(n+1)})$ 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 $(g_{\lambda_1}, \ldots, g_{\lambda_n})$, 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 $\Lambda = (\lambda_1, \ldots, \lambda_n)$ 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 $(g_{\lambda_1}, \ldots, g_{\lambda_n})$. 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

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

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

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 $\Lambda_\psi = \hat{\Lambda} := \{0\} \cup \Lambda \cup 2 \widehat{\ } \Lambda$, where $2 \widehat{\ } \Lambda := \{\lambda_i + \lambda_j : i, j \in [n] \wedge i \neq j\}$.

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 [15], 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 $2 \cdot \Lambda \cap 2 \widehat{\ } \Lambda = \emptyset$ 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 [25]. 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 [9]) 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 the full version, we give another proof of this result that, while based on Green and Wolf's exposition [13] of [9], 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 [23]. 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 [15]. In both

[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.

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 [15], the prover's computation can be optimized even further by using efficient multi-exponentiation algorithms. 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 [15] used $[an^2]$-PKE (for a constant a) and $[an^2]$-CPDH (which is a presumably stronger assumption than PSDL). Since Λ, Λ_ψ are small subsets of $[an^2]$, then our assumptions can be expected to be somewhat weaker in general. Finally, the 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.

4 Progression-Free Sets

A set of positive integers $\Lambda = \{\lambda_1, \dots, \lambda_n\}$ is *progression-free* [25], 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, $2\widehat{\ }\Lambda \cap 2 \cdot \Lambda = \emptyset$.

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. 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 [3], 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 [9], 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 the full version. Our proof is closely based on [13] but it has a sharper constant inside Ω. Moreover, our proof is much more detailed than that given in [13]. 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 [9]. On the other hand, Bourgain [5] showed that $r_3(N) = O(N \cdot (\log N/\log\log N)^{1/2})$, and recently Sanders [24] 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 $2 \cdot \Lambda + 1 \in [n^{1+o(1)}]$.

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 [15] 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 [15] and prove that it is computationally binding under the PDL assumption. Groth proved in [15] 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 $n = \text{poly}(\kappa)$. 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 $\mathcal{A}$,

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

is negligible in κ. A version of P(S)DL assumption in a non pairing-based group was defined in [12]. Cheon showed in [8] 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 [15], P(S)DL holds in the generic group model.

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

Λ-Power Knowledge of Exponent Assumption (Λ-PKE). Abe and Fehr showed in [1] 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 $\mathbf{NP} \subseteq \mathbf{P}/\text{poly}$. (See also [11].) 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 [15], we will base our NIZK argument for circuit satisfiability on Λ-PKE, an explicit knowledge assumption. This assumption was proposed by Groth [15] (though only for $\Lambda = [n]$) .

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 Λ be an (n, κ)-nice tuple for some $n = \text{poly}(\kappa)$. The bilinear group generator $\mathcal{G}_{\mathsf{bp}}$ is Λ-*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

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

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 $\ell \in \{0\} \cup \Lambda$) 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 $(r, (a_\ell)_{\ell \in \Lambda})$ such that $c = g_t^r \cdot \prod_{\ell \in \Lambda} g_t^{a_\ell x^\ell}$. Groth [15] 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 [15] with a generalized choice of generators, defined as follows:

CRS generation: Let Λ be an (n, κ)-nice tuple with $n = \text{poly}(\kappa)$. 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 Λ 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 Λ-PDL assumption in $\mathbb{G}_t$. If the Λ-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.*

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 $A \leftarrow \mathcal{C}\mathrm{om}^t(\mathsf{ck}; \boldsymbol{0}; r) = g_t^r$ 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 [15] with communication of 5 group elements, 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

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

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

System parameters: Let $n = \operatorname{poly}(\kappa)$. 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 $\ell \in 2^\frown\Lambda$, the prover sets $\mu_\ell \leftarrow \sum_{(i,j) \in I_1(\ell)} (a_i b_j - c_i)$. He sets $\psi \leftarrow g_2^{r_a r_b} \cdot \prod_{i=1}^n g_{2,\lambda_i}^{r_a b_i + r_b a_i - r_c} \cdot \prod_{\ell \in 2^\frown\Lambda} g_{2\ell}^{\mu_\ell}$, and $\hat{\psi} \leftarrow \hat{g}_2^{r_a r_b} \cdot \prod_{i=1}^n \hat{g}_{2,\lambda_i}^{r_a b_i + r_b a_i - r_c} \cdot \prod_{\ell \in 2^\frown\Lambda} \hat{g}_{2\ell}^{\mu_\ell}$. 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 $[\![(A, \hat{A})]\!] \circ [\![(B, \hat{B}, B_2)]\!] = [\![(C, \hat{C})]\!]$

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 [15], 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., $\ell \in \{0\} \cup \Lambda$).

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 [19] 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 [15] are not zero-knowledge but the final argument (for circuit satisfiability) is.

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.

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 $\ell \in 2\widehat{\ }\Lambda$ 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 [15], 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 $\varrho(\llbracket(A, \tilde{A})\rrbracket) = \llbracket(B, \tilde{B})\rrbracket$, 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 [15]. 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 $\tilde{\Lambda} = \Lambda \cup \{2\lambda_k - \lambda_i\} \cup 2\widehat{\ }\Lambda \cup \{\lambda_i + 2\lambda_k - \lambda_j : i \neq j\}$ and $2 \cdot \Lambda$ 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 Λ is a progression-free set of odd integers. Since Λ consists of odd integers, $(\Lambda \cup \{2\lambda_k - \lambda_i\}) \cap 2 \cdot \Lambda = \emptyset$. Since Λ is a progression-free set, $2\widehat{\ }\Lambda \cap 2 \cdot \Lambda = \emptyset$. 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 $a = 2\lambda_{k^*} + \lambda_j$. See the full version.) The unique sequence $\Lambda = (\lambda_i)_{i \in \mathbb{Z}^+}$ (the *Moser-de Bruijn sequence* [23]) 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 $T_\Lambda(i, \varrho) := |\{j \in [n] : 2\lambda_{\varrho(i)} + \lambda_j = 2\lambda_{\varrho(j)} + \lambda_i\}|$. Note that $1 \le T_\Lambda(i, \varrho) \le n$, and that for fixed Λ and ϱ, the whole tuple $\boldsymbol{T_\Lambda(\varrho)} := (T_\Lambda(1, \varrho), \dots, T_\Lambda(n, \varrho))$ can be computed in $\Theta(n)$ simple arithmetic operations. We can then rewrite $F_\varrho(x)$ as

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

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

$$\tilde{\Lambda} = \Lambda \cup \{2\lambda_k - \lambda_i\} \cup 2\hat{}\Lambda \cup (\{\lambda_i + 2\lambda_k - \lambda_j : i \neq j\} \setminus 2 \cdot \Lambda) \ . \qquad (6)$$

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

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

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

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 5. *Prot. 2 is perfectly complete and perfectly witness-indistinguishable. If $\mathcal{G}_{\mathsf{bp}}$ is $\tilde{\Lambda}$-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 $w^{\mathsf{perm}} \leftarrow (\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, \boldsymbol{a^*}, r_{a^*}, (f'_{(\times,\ell)})_{\ell \in \hat{\Lambda}}, (f'_{(\varrho,\ell)})_{\ell \in \tilde{\Lambda}})$, s.t. $(A, \tilde{A}) = \mathcal{C}om^1(\widetilde{\mathsf{ck}}_1; \boldsymbol{a}; r_a)$, $(B, \hat{B}) = \mathcal{C}om^1(\widehat{\mathsf{ck}}_1; \boldsymbol{b}; r_b)$, $(B, \tilde{B}) = \mathcal{C}om^1(\widetilde{\mathsf{ck}}_1; \boldsymbol{b}; r_b)$, $(A^*, \hat{A}^*) = \mathcal{C}om^1(\widehat{\mathsf{ck}}_1; \boldsymbol{a^*}; r_{a^*})$, $(\psi^\times, \hat{\psi}^\times) = (g_2^{\sum_{\ell \in \hat{\Lambda}} f'_{(\times,\ell)}}, \hat{g}_2^{\sum_{\ell \in \hat{\Lambda}} f'_{(\times,\ell)}})$, $(\psi^\varrho, \hat{\psi}^\varrho) = (g_2^{\sum_{\ell \in \tilde{\Lambda}} f'_{(\varrho,\ell)}}, \tilde{g}_2^{\sum_{\ell \in \tilde{\Lambda}} f'_{(\varrho,\ell)}})$, $a_i^* = T_\Lambda(\varrho^{-1}(i), \varrho) \cdot a_i$ (for $i \in [n]$), and for some $i \in [n]$, $a_{\varrho(i)} \neq b_i$.*

System parameters: Same as in Prot. 1, but let $\tilde{\Lambda}$ 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 $\ell \in \{0\} \cup \tilde{\Lambda}$. 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

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

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

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

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 $\varrho([\![(A, \tilde{A})]\!]) = [\![(B, \tilde{B})]\!]$

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 $\{X^\ell\}_{\ell \in 2 \cdot \Lambda \cup \tilde{\Lambda}}$. If the prover is honest, then $b_i = a_{\varrho(i)}$ for $i \in [n]$, and thus $F_\varrho(X)$ is spanned by $\{X^\ell\}_{\ell \in \tilde{\Lambda}}$. Defining $\psi^\varrho \leftarrow (\prod_{i=1}^n g_{2,\lambda_i})^{r_{a^*}} \cdot (\prod_{i=1}^n g_{2,2\lambda_{\varrho(i)} - \lambda_i})^{-r_b} \cdot \prod_{i=1}^n (\prod_{j=1: j \neq i}^n g_{2,\lambda_i + \lambda_j})^{a_i^*} \cdot \prod_{i=1}^n (\prod_{j \in I_2^*(i,\ell)} g_{2,\lambda_i + 2\lambda_{\varrho(j)} - \lambda_j})^{-b_i} = D^{r_{a^*}} \cdot E_\varrho^{-r_b} \cdot \prod_{\ell \in \tilde{\Lambda}'_\varrho} g_{2\ell}^{\mu_{\varrho,\ell}}$, 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 $\tilde{\Lambda}$-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 $\mathsf{crs} \leftarrow (\mathsf{gk}; (g_1^{x^\ell}, g_2^{x^\ell})_{\ell \in \{0\} \cup \tilde{\Lambda}})$ 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 $\mathsf{crs}' \leftarrow (\mathsf{gk}; (g_1^{x^\ell}, g_1^{\hat{\alpha}x^\ell}, g_1^{\tilde{\alpha}x^\ell})_{\ell \in \{0\} \cup \Lambda}, (g_2^{x^\ell})_{\ell \in \{0\} \cup \tilde{\Lambda}}, (g_2^{\hat{\alpha}x^\ell})_{\ell \in \hat{\Lambda}}, (g_2^{\tilde{\alpha}x^\ell})_{\ell \in \tilde{\Lambda}}, \prod_{i=1}^n g_2^{x^{\lambda_i}}, \prod_{i=1}^n \tilde{g}_2^{x^{\lambda_i}})$, and forwards crs' to $\mathcal{A}$. Clearly, crs' has the same distribution as $\mathcal{G}_{\mathsf{crs}}(1^\kappa)$. Both parties also set $\hat{\mathsf{ck}}_1 \leftarrow (\mathsf{gk}; (g_1^{x^\ell}, g_1^{\hat{\alpha}x^\ell})_{\ell \in \{0\} \cup \Lambda})$ and $\tilde{\mathsf{ck}}_1 \leftarrow (\mathsf{gk}; (g_1^{x^\ell}, g_1^{\tilde{\alpha}x^\ell})_{\ell \in \{0\} \cup \Lambda})$.

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 $w^{\mathsf{perm}} = (\boldsymbol{a}, r_a, \boldsymbol{b}, r_b, \boldsymbol{a^*}, r_{a^*}, (f'_{(\times,\ell)})_{\ell \in \hat{\Lambda}}, (f'_{(\varrho,\ell)})_{\ell \in \tilde{\Lambda}})$.

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

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

Next, $\mathcal{A}'$ can use an efficient polynomial factorization algorithm [19] 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 $\tilde{\Lambda}$-PSDL assumption. □

Let Λ 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.

8 New NIZK Argument for Circuit Satisfiability

In a *NIZK argument for circuit satisfiability* (Circuit-SAT, well-known to be an **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 [15], 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

System parameters: Define Λ and $\hat{\Lambda}$ as in Prot. 1 and $\tilde{\Lambda}$ 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 $\mathsf{ck}_1 \leftarrow (\mathsf{gk}; (g_{1\ell}, \hat{g}_{1\ell}, \tilde{g}_{1\ell})_{\ell \in \{0\} \cup \Lambda})$.

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 $[\![(\mathsf{lr}, \widehat{\mathsf{lr}})]\!] \circ [\![(\mathsf{lr}, \widehat{\mathsf{lr}}, \mathsf{lr}_2)]\!] = [\![(\mathsf{lr}, \widehat{\mathsf{lr}})]\!]$, ψ_2 for $\mathsf{swap}([\![(\mathsf{rl}, \widetilde{\mathsf{rl}})]\!]) = [\![(\mathsf{lr}, \widehat{\mathsf{lr}}, \widetilde{\mathsf{lr}})]\!]$, ψ_3 for $[\![(\mathsf{rl}, \widetilde{\mathsf{rl}})]\!] \circ [\![(D, \hat{D}, D_2)]\!] = [\![(\mathsf{rz}, \widehat{\mathsf{rz}})]\!]$, ψ_4 for $[\![(\mathsf{ux}, \widehat{\mathsf{ux}})]\!] \circ [\![(D, \hat{D}, D_2)/(g_{1,\lambda_n}, \hat{g}_{1,\lambda_n}, g_{1,\lambda_n})]\!] = [\![(\mathsf{uz}, \widehat{\mathsf{uz}})/(g_{1,\lambda_n}, \hat{g}_{1,\lambda_n}, g_{1,\lambda_n})]\!]$, ψ_5 for $[\![(\mathsf{rz}, \widehat{\mathsf{rz}})]\!] \circ [\![(\mathsf{lr}, \widehat{\mathsf{lr}}, \mathsf{lr}_2)]\!] = [\![(D, \hat{D}) \cdot (\mathsf{uz}^{-1}, \widehat{\mathsf{uz}}^{-1})]\!]$, ψ_6 for $\tau([\![(\mathsf{lr}, \widetilde{\mathsf{lr}})]\!]) = [\![(\mathsf{lr}, \widehat{\mathsf{lr}}, \widetilde{\mathsf{lr}})]\!]$, and ψ_7 for $\zeta^{-1}([\![(\mathsf{ux}, \widetilde{\mathsf{ux}})]\!]) = [\![(\mathsf{lr}, \widehat{\mathsf{lr}}, \widetilde{\mathsf{lr}})]\!]$.
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

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.

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, $(U_1, \ldots, U_n, X_1, \ldots, X_{n+1})$ 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 $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ for $Y \in \{L, R, U, X\}$.

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 $L_{i_1}, \ldots, L_{i_s}, R_{j_1}, \ldots, R_{j_t}$ that correspond to the same wire, τ contains a cycle $i_1 \to i_2 \to \cdots \to i_s \to j_1 + n \to \cdots \to j_t + n \to i_1$. 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 6. *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 $\boldsymbol{L} = (L_1, \ldots, L_n)$ and $\boldsymbol{R} = (R_1, \ldots, R_n)$, and let $(U_1, \ldots, U_n, X_1, \ldots, X_n, X_{n+1})$ be the opening of $(\mathsf{ux}, \widehat{\mathsf{ux}})$. We now analyze the effect of every subargument in Prot. 3.

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 $(R, 0, \ldots, 0, 0)$ 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 $(U_1, \ldots, U_{n-1}, U_n', 0, \ldots, 0, 0)$ 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 $L_{i_1} = \dots = L_{i_s} = R_{j_1} = \dots = R_{j_t}$, that is, the values are internally consistent with the wires. The seventh argument ψ_7 shows that the "input wires" and "output" wires are consistent.

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''. □

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.

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

1. Abe, M., Fehr, S.: Perfect NIZK with Adaptive Soundness. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 118–136. Springer, Heidelberg (2007)
2. Barreto, P.S.L.M., Naehrig, M.: Pairing-Friendly Elliptic Curves of Prime Order. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 319–331. Springer, Heidelberg (2006)

3. Behrend, F.A.: On the Sets of Integers Which Contain No Three in Arithmetic Progression. Proceedings of the National Academy of Sciences 32(12), 331–332 (1946)
4. Blum, M., Feldman, P., Micali, S.: Non-Interactive Zero-Knowledge and Its Applications. In: STOC 1988, pp. 103–112. ACM Press (1988)
5. Bourgain, J.: On Triples in Arithmetic Progression. Geom. Funct. Anal. 9(5), 968–984 (1998)
6. Chaabouni, R., Lipmaa, H., Shelat, A.: Additive Combinatorics and Discrete Logarithm Based Range Protocols. In: Steinfeld, R., Hawkes, P. (eds.) ACISP 2010. LNCS, vol. 6168, pp. 336–351. Springer, Heidelberg (2010)
7. Chaabouni, R., Lipmaa, H., Zhang, B.: A Non-Interactive Range Proof with Constant Communication. In: Keromytis, A. (ed.) FC 2012. LNCS, Springer, Heidelberg (2012)
8. Cheon, J.H.: Security Analysis of the Strong Diffie-Hellman Problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 1–11. Springer, Heidelberg (2006)
9. Elkin, M.: An Improved Construction of Progression-Free Sets. Israeli J. Math. 184, 93–128 (2011)
10. Gentry, C.: Fully Homomorphic Encryption Using Ideal Lattices. In: Mitzenmacher, M. (ed.) STOC 2009, pp. 169–178. ACM Press (2009)
11. Gentry, C., Wichs, D.: Separating Succinct Non-Interactive Arguments from All Falsifiable Assumptions. In: Vadhan, S. (ed.) STOC 2011, pp. 99–108. ACM Press (2011)
12. Golle, P., Jarecki, S., Mironov, I.: Cryptographic Primitives Enforcing Communication and Storage Complexity. In: Blaze, M. (ed.) FC 2002. LNCS, vol. 2357, pp. 120–135. Springer, Heidelberg (2003)
13. Green, B., Wolf, J.: A Note on Elkin's Improvement of Behrend's Construction. In: Chudnovsky, D., Chudnovsky, G. (eds.) Additive Number Theory, pp. 141–144. Springer, New York (2010)
14. Groth, J.: Linear Algebra with Sub-linear Zero-Knowledge Arguments. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 192–208. Springer, Heidelberg (2009)
15. Groth, J.: Short Pairing-Based Non-interactive Zero-Knowledge Arguments. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 321–340. Springer, Heidelberg (2010)
16. Groth, J.: Minimizing Non-interactive Zero-Knowledge Proofs Using Fully Homomorphic Encryption. Tech. Rep. 2011/012, IACR (2011)
17. Groth, J., Sahai, A.: Efficient Non-interactive Proof Systems for Bilinear Groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
18. Hess, F., Smart, N.P., Vercauteren, F.: The Eta Pairing Revisited. IEEE Trans. Inf. Theory 52(10), 4595–4602 (2006)
19. van Hoeij, M., Novocin, A.: Gradual Sub-lattice Reduction and a New Complexity for Factoring Polynomials. In: López-Ortiz, A. (ed.) LATIN 2010. LNCS, vol. 6034, pp. 539–553. Springer, Heidelberg (2010)
20. Lipmaa, H.: Progression-Free Sets and Sublinear Pairing-Based Non-Interactive Zero-Knowledge Arguments. Tech. Rep. 2011/009, IACR (2011)
21. Lipmaa, H., Zhang, B.: A More Efficient Computationally Sound Non-Interactive Zero-Knowledge Shuffle Argument. Tech. Rep. 2011/394, IACR (2011)
22. Micali, S.: CS Proofs. In: Goldwasser, S. (ed.) FOCS 1994, pp. 436–453. IEEE (1994)
23. Moser, L.: An Application of Generating Series. Mathematics Magazine 35(1), 37–38 (1962)
24. Sanders, T.: On Roth's Theorem on Progressions. Ann. Math. 174(1), 619–636 (2011)
25. Tao, T., Vu, V.: Additive Combinatorics. Cambridge Studies in Advanced Mathematics. Cambridge University Press (2006)