Short Pairing-based Non-interactive Zero-Knowledge Arguments

Jens Groth[*]

University College London
j.groth@ucl.ac.uk

Abstract. We construct non-interactive zero-knowledge arguments for circuit satisfiability with perfect completeness, perfect zero-knowledge and computational soundness. The non-interactive zero-knowledge arguments have sub-linear size and very efficient public verification. The size of the non-interactive zero-knowledge arguments can even be reduced to a constant number of group elements if we allow the common reference string to be large. Our constructions rely on groups with pairings and security is based on two new cryptographic assumptions; we do not use the Fiat-Shamir heuristic or random oracles.

Keywords: Sub-linear size non-interactive zero-knowledge arguments, pairing-based cryptography, power knowledge of exponent assumption, computational power Diffie-Hellman assumption.

1 Introduction

Zero-knowledge proofs introduced by Goldwasser, Micali and Rackoff [24] are fundamental building blocks in cryptography that are used in numerous protocols. Zero-knowledge proofs enable a prover to convince a verifier of the truth of a statement without leaking any other information. The central properties are captured in the notions of completeness, soundness and zero-knowledge.

Completeness: The prover can convince the verifier if the prover knows a witness testifying to the truth of the statement.

Soundness: A malicious prover cannot convince the verifier if the statement is false. We distinguish between computational soundness that protects against polynomial time cheating provers and statistical or perfect soundness where even an unbounded prover cannot convince the verifier of a false statement. We will call computationally sound proofs for *arguments*.

Zero-knowledge: A malicious verifier learns nothing except that the statement is true. We distinguish between computational zero-knowledge, where a polynomial time verifier learns nothing from the proof and statistical or perfect zero-knowledge, where even a verifier with unlimited resources learns nothing from the proof.

[*] Supported by EPSRC grant EP/G013829/1.

The first zero-knowledge proofs relied on interaction between the prover and the verifier. Many cryptographic tasks are carried out off-line though; for instance signing or encrypting messages. For these tasks it is desirable to have *non-interactive* zero-knowledge (NIZK) proofs, where there is no interaction and a proof just consists of a single message from the prover to the verifier. Unfortunately, only languages in BPP have NIZK proofs in the plain model without any setup [22, 21]. However, Blum, Feldman and Micali [6] introduced NIZK proofs in the *common reference string model*, where both the prover and verifier have access to a common reference string generated in a trusted way. Such NIZK proofs have many applications, ranging from early chosen ciphertext attack secure public-key cryptosystems [17, 38] to recent advanced signature schemes [11, 7]. For this reason there has been a lot of research into the underlying assumptions [19, 2, 28], the efficiency [13, 15, 33, 27], and the security guarantees offered by NIZK proofs [16, 38, 14].

NIZK proofs based on standard cryptographic assumptions used to be inefficient and not useful in practice. To get around this inefficiency, applied cryptographers have relied on the so-called Fiat-Shamir heuristic for transforming public-coin interactive zero-knowledge proofs into NIZK arguments by using a cryptographic hash-function to compute the verifier's challenges. The Fiat-Shamir heuristic can give very efficient NIZK arguments that are secure in the random oracle model [5], where the cryptographic hash-function is modeled as a random function. It is for instance possible to use the Fiat-Shamir heuristic to transform sub-linear size interactive public-coin zero-knowledge arguments [32] into sub-linear size non-interactive zero-knowledge arguments [35]. Unfortunately, there are several examples of protocols that are secure in the random oracle model, but do not have any secure standard model instantiation no matter which hash-function is used [9, 10, 34, 3, 37]. Particularly relevant here is Goldwasser and Kalai's [23] demonstration of a signature scheme built from a public-coin identification scheme that is secure in the random oracle model but insecure in real life.

Recent works on NIZK proofs has used bilinear groups to improve efficiency. Groth, Ostrovsky and Sahai [30, 29] gave NIZK proofs for circuit satisfiability where the proof consists of $O(|C|)$ group elements, with $|C|$ being the number of gates in the circuit. Their NIZK proofs have the property that they can be set up to give either perfect soundness and computational zero-knowledge, or alternatively computational soundness and perfect zero-knowledge. Works by Boyen, Waters, Groth and Sahai [7, 8, 25, 31] have explored how to build efficient NIZK proofs that are directly applicable in bilinear groups instead of going through circuit satisfiability. In some special cases, for instance in the ring signature of Chandran, Groth and Sahai [11], these techniques lead to sub-linear size NIZK proofs but in general the number of group elements in an NIZK proof grows linearly in the size of the statement. Abe and Fehr [1] gave a construction based on commitments instead of encryptions, but since there is a commitment for each wire they also get a linear growth in the size of the circuit.

Looking at the NP-complete problem of circuit satisfiability, the reason the NIZK proofs grow linearly in the circuit size is that they encrypt the value of each wire in the circuit. Gentry's new fully homomorphic cryptosystem [20] can reduce the NIZK proof to being linear in the size of the witness: The prover encrypts the inputs to the circuit and uses the homomorphic properties of the cryptosystem to compute the output of the circuit. The prover then gives NIZK proofs for the input ciphertexts being valid and the output ciphertext containing 1. Fully homomorphic encryption only helps when the circuit has a small witness though; if the circuit has a linear number of input wires the resulting NIZK proof will also be linear in the circuit size.

1.1 Our Contribution

Micali's CS proofs [35] indicated the possibility of sub-linear size NIZK arguments, but despite more than a decade of research the Fiat-Shamir heuristic is the only known strategy for constructing sub-linear size NIZK arguments. Our goal is to introduce a new type of sub-linear size NIZK arguments where security does not rely on the random oracle model.

We construct NIZK arguments for circuit satisfiability with perfect completeness, computational soundness and perfect zero-knowledge (see Section 2 for definitions). The NIZK arguments are short and very efficient to verify, but the prover uses a super-linear number of group operations. We first give an NIZK argument consisting of a constant number of group elements but having a long common reference string. We then show that it is possible to reduce the size of the common reference string at the cost of increasing the size of the NIZK argument making them simultaneously sub-linear in the circuit size.

The soundness of our NIZK argument relies on the q-computational power Diffie-Hellman and the q-power knowledge of exponent assumptions (see Section 3). The q-CPDH assumption is a normal computational intractability assumption but the q-PKE is a so-called knowledge of exponent assumption. Knowledge of exponent assumptions have been criticized for being unfalsifiable [36] but the use of a non-standard assumption may be unavoidable since Abe and Fehr [1] have demonstrated that no statistical zero-knowledge NIZK argument for an NP-complete language has a "direct black-box" reduction to a standard cryptographic assumption unless NP $\subseteq$ P/poly.[1][2]

[1] Abe and Fehr do not rule out the existence of statistical NIZK arguments with non-adaptive soundness, where the adversary chooses the statement oblivously of the common reference string. Since the common reference string is public it is more natural to define soundness adaptively though; indeed we do not know of any practical applications of NIZK arguments with non-adaptive soundness.

[2] The very assumption that an NIZK argument is sound seems to be unfalsifiable as well since even if an adversary outputs a false statement and a convincing NIZK argument it may be hard to verify that the statement is false. Groth, Ostrovsky and Sahai [30] circumvented this problem by defining co-soundness for languages in NP $\cap$ coNP, which is falsifiable since the adversary can produce a coNP-witness certifying that the statement is false.

	CRS size	Proof size	Prov. comp.	Ver. comp.	Assumption
Groth [27]	$\tilde{O}(\lvert C\rvert)$ G	$\tilde{O}(\lvert C\rvert)$ G	$\tilde{O}(\lvert C\rvert)$ E	$\tilde{O}(\lvert C\rvert)$ M	trapdoor perm.
Groth [27]	$\tilde{O}(\lvert C\rvert)$ bits	$\tilde{O}(\lvert C\rvert)$ bits	$\tilde{O}(\lvert C\rvert)$ M	$\tilde{O}(\lvert C\rvert)$ M	Naccache-Stern
Gentry [20]	$O(1)$ G	$\lvert w\rvert k^{O(1)}$ G	$\lvert C\rvert k^{O(1)}$ M	$\lvert C\rvert k^{O(1)}$ M	lattice-based
G-Ostrovsky-Sahai	$O(1)$ G	$O(\lvert C\rvert)$ G	$O(\lvert C\rvert)$ E	$O(\lvert C\rvert)$ P	pairing-
[30, 29]	$O(1)$ G	$O(\lvert C\rvert)$ G	$O(\lvert C\rvert)$ E	$O(\lvert C\rvert)$ E	based
Abe-Fehr [1]	$O(1)$ G	$O(\lvert C\rvert)$ G	$O(\lvert C\rvert)$ E	$O(\lvert C\rvert)$ E	knowledge of expo.
Groth [26]	$O(\lvert C\rvert^{\frac{1}{2}})$ G	$O(\lvert C\rvert^{\frac{1}{2}})$ G	$O(\lvert C\rvert)$ M	$O(\lvert C\rvert)$ M	random oracle
This paper	$O(\lvert C\rvert^2)$ G	$O(1)$ G	$O(\lvert C\rvert^2)$ M	$O(\lvert C\rvert)$ M	PKE and CDHP
This paper	$O(\lvert C\rvert^{\frac{2}{3}})$ G	$O(\lvert C\rvert^{\frac{2}{3}})$ G	$O(\lvert C\rvert^{\frac{4}{3}})$ M	$O(\lvert C\rvert)$ M	PKE and CDHP

Table 1. Comparison of NIZK proofs and arguments.

Table 1 gives a comparison to other NIZK proofs and arguments for circuit satisfiability, where k is a security parameter, G stands for the size of a group element, M and E are the costs of respectively multiplications and exponentiations, and P is the cost of a pairing in a bilinear group (see Section 3).

Compared to other pairing-based NIZK arguments, our arguments are smaller and faster to verify. The prover uses a super-linear number of multiplications and the computational cost may grow beyond a linear number of exponentiations. The public verifiability means that the NIZK arguments are transferable though; they can be copied and distributed to many different entities that can do their own independent verification. The prover only pays a one-time cost for computing the NIZK argument, while all verifiers enjoy the benefits of low transmission bandwidth and efficient verification.

Perfect Zaps. The common reference string model assumes a trusted setup for generating common reference strings and making them available to the prover and verifier. In case no such setup is available[3] we can still get a sub-linear size 2-move publicly verifiable witness-indistinguishable argument where the verifiers first message can be reused many times, a so-called Zap [18], as follows: The verifier generates a common reference string. The prover verifies that the common reference string is well-formed (our common reference string is not a random bit-string, but it does have a certain structure that makes it possible to verify that it is well-formed) and can now make arbitrarily many Zaps using the verifier initial message as the common reference string. Since our NIZK argument is perfectly zero-knowledge, the Zaps will be perfectly witness-indistinguishable.

1.2 Outline of Our NIZK Argument

We will construct NIZK arguments for the existence of an input to a binary circuit C making it output 1. At a loss of a constant factor, we may assume C

[3] We remark that even if the common reference string is adversarially chosen the sub-linearity of our NIZK arguments impose an information theoretic upper bound on how much information can be leaked.

consists of NAND-gates. Furthermore, if we label the output wire a we may add a self-loop to the circuit consisting of a NAND-gate $a = \neg(a \wedge b)$ forcing a to be 1. This reduces the challenge to prove that there is an assignment of truth-values to the wires that respect all the $N = |C|$ NAND-gates in the circuit.

The NIZK argument relies on length-reducing commitments where we commit to n values in a finite field $\mathbb{Z}_p$ using only a constant number of group elements. We will also use non-interactive arguments consisting of a constant number of group elements for proving the following properties about committed values:

Entry-wise product: Commitments c, d, v contain values $a_1, \ldots, a_n, b_1, \ldots, b_n$ and $u_1, \ldots, u_n$ that satisfy $u_i = a_i b_i$ for all i.

Permutation: Commitments c, d contain values $a_1, \ldots, a_n$ and $b_1, \ldots, b_n$ that satisfy $b_i = a_{\rho(i)}$ for all i, where ρ is a publicly known permutation of n elements.

Let us sketch how commitments combined with these two types of non-interactive arguments give us a constant size NIZK argument for circuit satisfiability when $n = 2N$. The prover gets as a witness for the satisfiability of the circuit $a_1, \ldots, a_N$ and $b_1, \ldots, b_N$ such that a_i, b_i are the inputs to gate i and all the values are consistent with the wires and respect the NAND-gates. We use the convention that -1 corresponds to false and $+1$ corresponds to true, so if u_i is the output of gate i we have $u_i = -a_i b_i$.

The prover makes commitments to the $2N$-tuples

$$(a_1, \ldots, a_N, b_1, \ldots, b_N) \qquad (b_1, \ldots, b_N, 0, \ldots, 0) \qquad (-u_1, \ldots, -u_N, 0, \ldots, 0).$$

The prover gives an entry-wise product argument on the commitment to $(a_1, \ldots, a_N, b_1, \ldots, b_N)$ with itself to show $a_i^2 = 1$ and $b_i^2 = 1$ for all i. This shows that $a_1, \ldots, a_N, b_1, \ldots, b_N \in \{-1, 1\}$ are appropriate truth values.

An output of one NAND-gate may be the input of other NAND-gates, which means the corresponding values $a_{i_1}, \ldots, a_{i_\ell}, b_{j_1}, \ldots, b_{j_m}$ have to have the same assignment. The prover picks a permutation ρ that contains cycles $i_1 \to i_2 \to \ldots \to i_\ell \to j_1 + N \to j_2 + N \to \ldots \to j_m + N \to i_1$ for all such sets of values that have to be consistent and gives a permutation argument on the commitment to $(a_1, \ldots, a_N, b_1, \ldots, b_N)$. This shows for each set of values corresponding to the same output wire that $a_{i_2} = a_{i_1}, \ldots, b_{j_1} = a_{i_\ell}, \ldots, b_{j_m} = b_{j_{m-1}}$ so the values $(a_1, \ldots, a_N, b_1, \ldots, b_N)$ are consistent with the wiring of the circuit.

The prover uses additional commitments, entry-wise product and permutation arguments to show that the other committed values $(b_1, \ldots, b_N, 0, \ldots, 0)$ and $(-u_1, \ldots, -u_N, 0, \ldots, 0)$ are consistent with the wiring of the circuit and the values $(a_1, \ldots, a_N, b_1, \ldots, b_N)$, we refer to Section 8 for the details.

Finally, the prover uses the entry-wise product argument to show that the entry-wise product of $(a_1, \ldots, a_N, b_1, \ldots, b_N)$ and $(b_1, \ldots, b_N, 0, \ldots, 0)$ is $(-u_1, \ldots, -u_N, 0, \ldots, 0)$ so all the values respect the NAND gates.

This outline shows how to get a constant size NIZK argument for circuit satisfiability, but to enable the entry-wise product arguments and the permutation arguments the common reference string has size $O(N^2)$ group elements. In

Section 9 we reduce the common reference string size by using commitments to n elements where $n < N$. With n smaller than $2N$ we need to give permutation arguments that span accross multiple commitments though. Using permutation network techniques [12] we manage to build such large permutations from many smaller permutations.

The technical contribution of this paper is the construction of an appropriate commitment scheme with corresponding non-interactive entry-wise product and permutation arguments. The commitment scheme is a variant of the Pedersen commitment scheme, where the commitment key is of the form $(g, g^x, \ldots, g^{x^q})$. A commitment to $a_1, \ldots, a_q$ is a single group element computed as $g^r \prod_{i=1}^q (g^{x^i})^{a_i}$.

The nice thing about such a commitment is that the discrete logarithm is a polynomial $r + \sum_{i=1}^q a_i x^i$. When we pair two commitments with each other we get a product of two polynomials in the exponent. By taking appropriate linear combinations over products of polynomials, we can express entry-wise products and permutations as equations over the coefficients of these polynomials. The q-CPDH assumption then allows us to conclude that these coefficients are identical and therefore the committed values satisfy an entry-wise multiplication relationship or a permutation relationship to each other.

When pairing commitments (equivalent to multiplying polynomials in the exponent) there will be various cross-terms. The role of the non-interactive arguments will be to cancel out these terms. Usually, a single group element paired with g suffices to cancel out all the cross-terms, so the non-interactive arguments for entry-wise products and permutations are highly efficient themselves.

To prove that our NIZK argument is sound, we need to reason about the coefficient of these polynomials. However, a cheating prover might create a commitment without knowing an opening of it. This is where the q-PKE assumption comes in handy: the prover gives non-interactive arguments demonstrating that it "knows" the openings of the commitments. By this we mean that there is an extractor that given the same input as the prover can reconstruct the commitments together with the openings of the commitments.

2 Definitions

Let R be an efficiently computable binary relation. For pairs $(C, w) \in R$ we call C the statement and w the witness. Let L be the NP-language consisting of statements with witnesses in R. When we restrict ourselves to statements of size N, we write respectively L_N and R_N.

A non-interactive argument for a relation R consists of a common reference string generator algorithm K, a prover algorithm P and a verifier algorithm V that run in probabilistic polynomial time. The common reference string generator takes as input a security parameter k and the statement size N and produces a common reference string σ. The prover on input (σ, C, w) produces an argument π. The verifier on input (σ, C, π) outputs 1 if the argument is acceptable and 0 if rejecting the argument. We call (K, P, V) an argument for R if it has the completeness and soundness property described below.

Perfect completeness. Completeness captures the notion that an honest prover should be able to convince an honest verifier if the statement is true. For $N = k^{O(1)}$ and all adversaries $\mathcal{A}$ outputting $(C, w) \in R_N$:

$$\Pr\Big[\sigma \leftarrow K(1^k, N); (C, w) \leftarrow \mathcal{A}(\sigma); \pi \leftarrow P(\sigma, C, w) : V(\sigma, C, \pi) = 1\Big] = 1.$$

Computational soundness. Soundness captures the notion that it should be infeasible for an adversary to come up with an accepting argument for a false statement. For $N = k^{O(1)}$ and all non-uniform polynomial time adversaries $\mathcal{A}$:

$$\Pr\Big[\sigma \leftarrow K(1^k, N); (C, \pi) \leftarrow \mathcal{A}(\sigma) : C \notin L \text{ and } V(\sigma, C, \pi) = 1\Big] \approx 0.$$

Perfect witness-indistinguishability. We say a non-interactive argument (K, P, V) is perfectly witness-indistinguishable if it is impossible to tell which witness the prover when there are many possible witnesses. For $N = k^{O(1)}$ and all stateful interactive adversaries $\mathcal{A}$ outputting $(C, w_0), (C, w_1) \in R_N$:

$$\begin{aligned} &\Pr\Big[\sigma \leftarrow K(1^k, N); (C, w_0, w_1) \leftarrow \mathcal{A}(\sigma); \pi \leftarrow P(\sigma, C, w_0) : \mathcal{A}(\pi) = 1\Big] \\ = &\Pr\Big[\sigma \leftarrow K(1^k, N); (C, w_0, w_1) \leftarrow \mathcal{A}(\sigma); \pi \leftarrow P(\sigma, C, w_1) : \mathcal{A}(\pi) = 1\Big]. \end{aligned}$$

Perfect zero-knowledge. An argument is zero-knowledge if it does not leak any information besides the truth of the statement. We say a non-interactive argument (K, P, V) is perfect zero-knowledge if there exists a polynomial time simulator $S = (S_1, S_2)$ with the following zero-knowledge property. S_1 outputs a simulated common reference string and a simulation trapdoor. S_2 takes the common reference string, the simulation trapdoor and a statement as input and produces a simulated argument. For $N = k^{O(1)}$ and all stateful interactive adversaries $\mathcal{A}$ outputting $(C, w) \in R_N$:

$$\begin{aligned} &\Pr\Big[\sigma \leftarrow K(1^k, N); (C, w) \leftarrow \mathcal{A}(\sigma); \pi \leftarrow P(\sigma, C, w) : \mathcal{A}(\pi) = 1\Big] \\ = &\Pr\Big[(\sigma, \tau) \leftarrow S_1(1^k, N); (C, w) \leftarrow \mathcal{A}(\sigma); \pi \leftarrow S_2(\sigma, \tau, C) : \mathcal{A}(\pi) = 1\Big]. \end{aligned}$$

3 Bilinear Groups

Notation. Given two functions $f, g : \mathbb{N} \to [0, 1]$ we write $f(k) \approx g(k)$ when $|f(k) - g(k)| = O(k^{-c})$ for every constant $c > 0$. We say that f is *negligible* when $f(k) \approx 0$ and that it is *overwhelming* when $f(k) \approx 1$.

We write $y = A(x; r)$ when the algorithm A on input x and randomness r, outputs y. We write $y \leftarrow A(x)$ for the process of picking randomness r at random and setting $y = A(x; r)$. We also write $y \leftarrow S$ for sampling y uniformly at random from the set S. We will assume it is possible to sample uniformly at random from sets such as $\mathbb{Z}_p$. We define $[n]$ to be the set $\{1, 2, \ldots, n\}$.

Bilinear groups. Let $\mathcal{G}$ take a security parameter k written in unary as input and output a description of a bilinear group $(p, G, G_T, e) \leftarrow \mathcal{G}(1^k)$ such that

1. p is a k-bit prime.
2. G, G_T are cyclic groups of order p.
3. $e : G \times G$ is a bilinear map (pairing) such that $\forall a, b : e(g^a, g^b) = e(g, g)^{ab}$.
4. If g generates G then $e(g, g)$ generates G_T.
5. Membership in G, G_T can be efficiently decided, group operations and the pairing e are efficiently computable, generators are efficiently sampleable, and the descriptions of the groups and group elements each have size $O(k)$ bits.

The security of our NIZK arguments will be based on two new assumptions, which we call respectively the q-power knowledge of exponent assumption and the q-computational power Diffie-Hellman assumption.

THE q-POWER KNOWLEDGE OF EXPONENT ASSUMPTION. The knowledge of exponent (KEA) assumption says that given g, g^α it is infeasible to create $c, \hat{c}$ so $\hat{c} = c^\alpha$ without knowing a so $c = g^a$ and $\hat{c} = (g^\alpha)^a$. Bellare and Palacio [4] extended this to the KEA3 assumption, which says that given $g, g^x, g^\alpha, g^{\alpha x}$ it is infeasible to create $c, \hat{c}$ so $\hat{c} = c^\alpha$ without knowing a_0, a_1 so $c = g^{a_0}(g^x)^{a_1}$ and $\hat{c} = (g^\alpha)^{a_0}(g^{\alpha x})^{a_1}$.

The q-power knowledge of exponent assumption is a generalization of KEA and KEA3. It says that given $(g, g^x, \ldots, g^{x^q}, g^\alpha, g^{\alpha x}, \ldots, g^{\alpha x^q})$ it is infeasible to create $c, \hat{c}$ so $\hat{c} = c^\alpha$ without knowing $a_0, \ldots, a_q$ so $c = \prod_{i=0}^q (g^{x^i})^{a_i}$ and $\hat{c} = \prod_{i=0}^q (g^{\alpha x^i})^{a_i}$.

We will now give the formal definition of the q-power knowledge of exponent assumption. Following Abe and Fehr [1] we write $(y; z) \leftarrow (\mathcal{A} \parallel \mathcal{X}_\mathcal{A})(x)$ when $\mathcal{A}$ on input x outputs y and $\mathcal{X}_\mathcal{A}$ on the same input (including the random tape of $\mathcal{A}$) outputs z.

Definition 1 (q-PKE). *The $q(k)$-power knowledge of exponent assumption holds for $\mathcal{G}$ if for every non-uniform probabilistic polynomial time adversary $\mathcal{A}$ there exists a non-uniform probabilistic polynomial time extractor $\mathcal{X}_\mathcal{A}$ so*

$$\Pr\Big[(p, G, G_T, e) \leftarrow \mathcal{G}(1^k) \ ; \ g \leftarrow G \setminus \{1\} \ ; \ \alpha, x \leftarrow \mathbb{Z}_p^* \ ; \\ \sigma = (p, G, G_T, e, g, g^x, \ldots, g^{x^q}, g^\alpha, g^{\alpha x}, \ldots, g^{\alpha x^q}) \ ; \\ (c, \hat{c} \ ; \ a_0, \ldots, a_q) \leftarrow (\mathcal{A} \parallel \mathcal{X}_\mathcal{A})(\sigma) \ : \ \hat{c} = c^\alpha \wedge c \neq \prod_{i=0}^n g^{a_i x^i}\Big] \ \approx \ 0.$$

THE q-COMPUTATIONAL POWER DIFFIE-HELLMAN ASSUMPTION. The computational Diffie-Hellman (CDH) assumption says that given g, g^β, g^x it is infeasible to compute $g^{\beta x}$. The q-computational power Diffie-Hellman assumption is a generalization of the CDH assumption that says given $(g, g^x, \ldots, g^{x^q}, g^\beta, g^{\beta x}, \ldots, g^{\beta x^q})$ except for one missing elements $g^{\beta x^j}$, it is hard to compute the missing element.

Definition 2 (q-CPDH). *The $q(k)$-computational power Diffie-Hellman assumption holds for $\mathcal{G}$ if for all $j \in \{0, \ldots, q\}$ and all non-uniform probabilistic* *polynomial time adversaries $\mathcal{A}$ we have*

$$\Pr\Big[(p, G, G_T, e) \leftarrow \mathcal{G}(1^k) \ ; \ g \leftarrow G \setminus \{1\} \ ; \ \beta, x \leftarrow \mathbb{Z}_p^* \ ; \\ y \leftarrow (\mathcal{A}, \mathcal{X}_{\mathcal{A}})(p, G, G_T, e, g, g^x, \ldots, g^{x^q}, g^\beta, g^{\beta x}, \ldots, \\ g^{\beta x^{j-1}}, g^{\beta x^{j+1}}, \ldots, g^{\beta x^q}) \ : \quad y = g^{\beta x^j}\Big] \ \approx \ 0.$$

In the full paper we give heuristic arguments for believing in the q-PKE and q-CPDH assumptions by proving that they hold in the generic group model.

4 Knowledge Commitment

We will use a variant of the Pedersen commitment scheme in our NIZK proof where we commit to $a_1, \ldots, a_q$ as $c = g^r \prod_{i\in[q]} g_i^{a_i}$. In the security proof of our NIZK argument for 3SAT we will need to extract the committed values $a_1, \ldots, a_q$; but the commitment scheme itself is perfectly hiding and does not reveal the committed values. For this reason, we will require the prover to create a related commitment $\hat{c} = \hat{g} \prod_{i\in[q]} \hat{g}_i^{a_i}$ and will rely on the q-PKE assumption for extracting the committed values. We call $(c, \hat{c})$ a knowledge commitment, since the prover cannot make a valid commitment without "knowing" the committed values.

Key generation: Pick $gk = (p, G, G_T, e) \leftarrow \mathcal{G}(1^k) \quad g \leftarrow G \setminus \{1\} \ ; \ x, \alpha \leftarrow \mathbb{Z}_p^*$. The commitment key is $ck = (gk, g, g_1, \ldots, g_q, \hat{g}, \hat{g}_1 \ldots, \hat{g}_q) = (gk, g, g^x, \ldots, g^{x^q}, g^\alpha, g^{\alpha x}, \ldots, g^{\alpha x^q})$ and the trapdoor key is $tk = x$.

Commitment: To commit to $a_1, \ldots, a_q$ pick $r \leftarrow \mathbb{Z}_p$ and compute the knowledge commitment $(c, \hat{c})$ as

$$c = g^r \prod_{i\in[q]} g_i^{a_i} \qquad\qquad \hat{c} = \hat{g}^r \prod_{i\in[q]} \hat{g}_i^{a_i}.$$

Given $(c, \hat{c}) \in G^2$ we can verify that it is well-formed by checking $e(g, \hat{c}) = e(c, \hat{g})$.

Trapdoor commitment: To make a trapdoor commitment sample trapdoor randomness $t \leftarrow \mathbb{Z}_p$ and compute the knowledge commitment $(c, \hat{c})$ as $c = g^t \ ; \ \hat{c} = \hat{g}^t$.

Trapdoor opening: The trapdoor opening algorithm on messages $a_1, \ldots, a_q \in \mathbb{Z}_p$ returns the randomizer $r = t - \sum_{i\in[q]} a_i x^i$. The trapdoor opening satisfies $c = g^r \prod_{i\in[q]} g_i^{a_i}$ and $\hat{c} = \hat{g}^r \prod_{i\in[q]} \hat{g}_i^{a_i}$.

The commitment scheme has properties similar to those of standard Pedersen commitments as the following theorem shows. We refer to the full paper for the proof of the following theorem.

Theorem 1. *The commitment scheme is perfectly trapdoor and computationally binding. Assuming the q-PKE assumption holds, there exists for any non-uniform probabilistic polynomial time committer $\mathcal{A}$ a non-uniform probabilistic polynomial time extractor $\mathcal{X}_{\mathcal{A}}$ that computes the contents of the commitment when given the input of $\mathcal{A}$ (including any random coins).*

4.1 Restriction Argument

Consider a subset $S \subset [q]$ and a commitment c. We will need an argument for the opening $r, a_1, \ldots, a_q$ being such that the indices of non-zero values are restricted to S. In other words, we need an argument for the commitment being of the form $c = g^r \prod_{i \in S} g_i^{a_i}$. The argument will take the form $\pi = h^r \prod_{i \in S} h_i^{a_i}$, which intuitively corresponds to an additional argument of knowledge with respect to a small base $(h, \{h_i\}_{i \in S})$.

Setup: $gk \leftarrow \mathcal{G}(1^k)$; $ck \leftarrow K_{\text{commit}}(gk)$.
Common reference string: Given (ck, S) as input pick at random $\beta \leftarrow \mathbb{Z}_p^*$ and compute the common reference string as $\sigma = (h, \{h_i\}_{i \in S}) = (g^\beta, \{g_i^\beta\}_{i \in S})$.
Statement: A valid knowledge commitment $(c, \hat{c})$.
Prover's witness: Opening $r, \{a_i\}_{i \in S}$ so $c = g^r \prod_{i \in S} g_i^{a_i}$ and $\hat{c} = \hat{g}^r \prod_{i \in S} \hat{g}_i^{a_i}$.
Argument: Compute the argument as $\pi = h^r \prod_{i \in S} h_i^{a_i}$.
Verification: Output 1 if and only if $e(c, h) = e(g, \pi)$.

Theorem 2. *The restriction argument is perfectly complete and perfectly witness-indistinguishable. If the q-CPDH assumption holds, all non-uniform probabilistic polynomial time adversaries have negligible probability of outputting* $(r, a_1, \ldots, a_q, \pi)$ *so* $a_i \neq 0$ *for some* $i \notin S$ *and* π *is an acceptable restriction argument for the commitment corresponding to the opening.*

We refer to the full paper for the proof. Observe that we phrase the soundness of the restriction argument as the inability to find an opening of the commitment that violates the restriction. Since the commitment scheme is perfectly hiding we cannot exclude the existence of openings that violate the restriction. However, if it holds that it is a knowledge commitment (Theorem 1) we see that the opening we extract from the committer must respect the restriction.

5 Common Reference String

We will now describe how to generate the common reference string for our NIZK argument. The common reference string will consist of a knowledge commitment key ck for $q = n^2 + 3n - 2$ values together with three common reference strings for restriction to the sets

$$\tilde{S} = \{1, \ldots, n\}\ ,\ \bar{S} = \{(n+1), \ldots, n(n+1)\}\ ,\ \dot{S} = \{\ell \in [q] \mid \ell \neq 0 \bmod n+2\}.$$

The zero-knowledge simulation of the argument will use the same type of common reference string, and the simulation trapdoor for our NIZK argument will be the trapdoor for the knowledge commitment.

Common Reference String Generation:

On input 1^k and n do

1. Generate $(p, G, G_T, e) \leftarrow \mathcal{G}(1^k)$ and set $gk = (p, G, G_T, e)$.

2. Pick $g \leftarrow G \setminus \{1\}$; $x, \alpha \leftarrow \mathbb{Z}_p^*$ and compute

$$ck = (gk, g, \ldots, g_q, \hat{g}, \ldots, \hat{g}_q) = (gk, g, \ldots, g^{x^{n^2+3n-2}}, g^{\alpha}, \ldots, g^{\alpha x^{n^2+3n-2}}).$$

3. Generate $\tilde{\sigma} \leftarrow K_{\text{restrict}}(ck, \tilde{S})$ where $\tilde{S} = \{1, 2, \ldots, n\}$.
4. Generate $\bar{\sigma} \leftarrow K_{\text{restrict}}(ck, \bar{S})$ where $\bar{S} = \{(n+1), 2(n+1), \ldots, n(n+1)\}$.
5. Generate $\dot{\sigma} \leftarrow K_{\text{restrict}}(ck, \dot{S})$ where $\dot{S} = \{\ell \in [q] \mid \ell \neq 0 \bmod n+2\}$.

The common reference string is $\sigma = (ck, \tilde{\sigma}, \bar{\sigma}, \dot{\sigma})$ and the simulation trapdoor is $tk = x$.

Given a common reference string, it is hard to find a non-trivial linear combination of $1, x, \ldots, x^q$ because we could run a polynomial factorization algorithm in $\mathbb{Z}_p[X]$ to compute the root x. We will repeatedly use this fact, so we prove the following Lemma in the full paper.

Lemma 1. *If the q-CPDH assumption holds for $\mathcal{G}$ with $q = n^2 + 3n - 2$, a non-uniform probabilistic polynomial time adversary has negligible chance of finding a non-trivial linear combination $(a_0, \ldots, a_q)$ such that $\sum_{i=0}^{q} a_i x^i = 0$ given a random common reference string σ.*

6 Product Argument

Consider three commitments

$$c = g^r \prod_{i \in [n]} g_i^{a_i} \qquad d = g^s \prod_{j \in [n]} g_{j(n+1)}^{b_j} \qquad v = g^t \prod_{i \in [n]} g_i^{u_i} \qquad \forall i \in [n] : u_i = a_i b_i.$$

With the corresponding restriction arguments, $\hat{c}, \tilde{c}, \hat{d}, \bar{d}, \hat{v}, \tilde{v}$ we can assume the committer knows openings to values $a_1, \ldots, a_n, b_1, \ldots, b_n$ and $u_1, \ldots, u_n$. We will give an argument $(\pi, \hat{\pi}, \dot{\pi})$ consisting of three group elements for the committed values satisfying $u_1 = a_1 b_1, \ldots, u_n = a_n b_n$.

In order to explain the intuition in the argument, let us consider the following toy example $c = \prod_{i \in [n]} g_i^{a_i}$ and $d = \prod_{j \in [n]} g_{j(n+1)}^{b_j}$, where we want to show $a_1 b_1 = 0, \ldots, a_n b_n = 0$. The discrete logarithms of the two commitments are $\sum_{i \in [n]} a_i x^i$ and $\sum_{j \in [n]} b_j x^{j(n+1)}$ and the discrete logarithm of $e(c, d)$ is

$$\left(\sum_{i \in [n]} a_i x^i \right) \cdot \left(\sum_{j \in [n]} b_j x^{j(n+1)} \right) = \sum_{i \in [n]} a_i b_i x^{i(n+2)} + \sum_{i \in [n]} \sum_{j \in [n] \setminus \{i\}} a_i b_j x^{j(n+1)+i}.$$

In the final sum, the left term contains the coefficients $a_1 b_1, \ldots, a_n b_n$ that are supposed to be 0, however, the right term complicates matters. The argument π will be constructed such that it can be used to cancel out the latter term.

Notice that the left term isolates the coefficients of $x^{n+2}, \ldots, x^{n(n+2)}$, while the right term does not contain any such coefficients. By giving an appropriate restriction argument, the prover can guarantee that she only cancels out the right term without interfering with the left term containing $x^{n+2},\dots,x^{n(n+2)}$. The prover computes $\pi = \prod_{i\in[n]}\prod_{j\in[n]\setminus\{i\}} g_{j(n+1)+i}^{a_ib_j}$ and gives corresponding $\hat{\pi},\dot{\pi}$ values demonstrating that it knows an opening $(z,\{z_\ell\}_{\ell\in\dot{S}})$ of π restricted to $\dot{S}$. The verifier will check

$$e(c,d)=e(g,\pi).$$

Let us now argue that we have soundness: given $\pi,\hat{\pi},\dot{\pi}$ such that $e(c,d)=e(g,\pi)$ the verifier can be assured that $a_1b_1=0,\dots,a_nb_n=0$. Taking discrete logarithms, the verification equation tells us that

$$\sum_{i\in[n]} a_ib_ix^{i(n+2)} + \sum_{i\in[n]}\sum_{j\in[n]\setminus\{i\}} a_ib_jx^{j(n+1)+i} = z + \sum_{\ell\in\dot{S}} z_\ell x^\ell.$$

Recall, $\dot{S}=\{\ell\in[n^2+3n-2] \mid \ell\neq 0 \bmod n+2\}$ so the argument π will not contain any coefficients of the form $x^{n+2},\dots,x^{n(n+2)}$. This means the coefficients of $x^{n+2},\dots,x^{n(n+2)}$ are $a_1b_1,\dots,a_nb_n$. If there is an i such that $a_ib_i\neq 0$, then we have a non-trivial polynomial equation in x. By Lemma 1 this would allow us to recover x and breaking the q-PKE assumption.

In the general case we want to give an argument for $a_ib_i=u_i$ instead of just $a_ib_i=0$. However, if we evaluate $e(v,\prod_{j\in[n]} g_{j(n+1)})$ we can view the latter as a commitment to $(1,1,\dots,1)$ and we will get their products $u_1\cdot 1,\dots,u_n\cdot 1$ as coefficients of $x^{n+2},\dots,x^{n(n+2)}$. If $u_1=a_1b_1,\dots,u_n=a_nb_n$ the two pairings $e(c,d)$ and $e(v,\prod_{j\in[n]} g_{j(n+1)})$ therefore have the same coefficients of $x^{n+2},\dots,x^{n(n+2)}$ and otherwise the coefficients are different. As in the toy example above, we may choose π such that it cancels out all the other terms. Due to the restriction to $\dot{S}$ the argument will not have any $x^{n+2},\dots,x^{n(n+2)}$ terms and we therefore get soundness. In the general case, the commitments also have randomizers and we will choose π such that it also cancels out these terms.

Statement: Commitments $c,d,v\in G$.
Prover's witness: Openings $r,a_1,\dots,a_n$ and $s,b_1,\dots,b_n$ and $t,u_1,\dots,u_n$ so

$$c=g^r\prod_{i\in[n]} g_i^{a_i}\ ,\ d=g^s\prod_{j\in[n]} g_{j(n+1)}^{b_j}\ ,\ v=g^t\prod_{i\in[n]} g_i^{u_i}\ ,\ \forall i\in[n]: u_i=a_ib_i.$$

Argument: Compute the argument $(\pi,\hat{\pi},\dot{\pi})$ as

$$\pi = g^{rs}\prod_{i\in[n]} g_i^{a_is}\prod_{j\in[n]} g_{j(n+1)}^{b_jr-t}\prod_{i\in[n]}\prod_{j\in[n]\setminus\{i\}} g_{j(n+1)+i}^{a_ib_j-u_i}$$
$$\hat{\pi} = \hat{g}^{rs}\prod_{i\in[n]} \hat{g}_i^{a_is}\prod_{j\in[n]} \hat{g}_{j(n+1)}^{b_jr-t}\prod_{i\in[n]}\prod_{j\in[n]\setminus\{i\}} \hat{g}_{j(n+1)+i}^{a_ib_j-u_i}$$
$$\dot{\pi} = \dot{h}^{rs}\prod_{i\in[n]} \dot{h}_i^{a_is}\prod_{j\in[n]} \dot{h}_{j(n+1)}^{b_jr-t}\prod_{i\in[n]}\prod_{j\in[n]\setminus\{i\}} \dot{h}_{j(n+1)+i}^{a_ib_j-u_i}$$

Verification: Output 1 if and only if

$$e(g, \hat{\pi}) = e(\pi, \hat{g}) \ \wedge \ e(g, \dot{\pi}) = e(\pi, \dot{h}) \ \wedge \ e(c, d) = e(v, \prod_{j \in [n]} g_{j(n+1)}) e(g, \pi).$$

Theorem 3. *The product argument has perfect completeness and perfect witness-indistinguishability. If the q-CPDH assumption holds, then a non-uniform probabilistic polynomial time adversary has negligible chance of outputting commitments (c, d, v) and an accepting argument π with corresponding openings of the commitments and the argument such that for some $i \in [n]$ we have $a_i b_i \neq u_i$.*

The proof can be found in the full paper.

The product argument has two commitments with restriction to $\tilde{S}$ and one commitment restricted to $\bar{S}$. It is quite easy to translate commitments back and forth between $\tilde{S}$ and $\bar{S}$ though. If we have two commitments v and d restricted to respectively $\tilde{S}$ and $\bar{S}$, we can give a product argument for the values in v being the product of the values in $c = \prod_{i \in [n]} g_i$ and d. Since c is a commitment to $(1, \ldots, 1)$ this proves that v and d contain the same values.

The product argument makes it possible to prove that the committed values in a commitment c are bits encoded as ±1. If we give a product argument for $\prod_{i \in [n]} g_i$ (a commitment to $(1, \ldots, 1)$) being the product of the values in c and in d, where d contains the same values as c, then we have that the values satisfy $a_i^2 = 1$, which implies $a_i = \pm 1$.

7 Permutation Argument

Consider two commitments and a permutation

$$c = g^r \prod_{i \in [n]} g_i^{a_i} \qquad d = g^s \prod_{i \in [n]} g_i^{b_i} \qquad \rho \in S_n \qquad \forall i \in [n] : \ b_i = a_{\rho(i)}.$$

We will now give an argument for the committed values satisfying $b_i = a_{\rho(i)}$, where $\rho \in S_n$ is a publicly known permutation.

The idea behind the permutation argument is to show

$$\sum_{i \in [n]} a_i x^{i(n+2)} = \sum_{i \in [n]} b_i x^{\rho(i)(n+2)}.$$

By Lemma 1 this implies $b_i = a_{\rho(i)}$ for all $i \in [n]$.

To get the desired linear combination we compute $e(c, \prod_{j \in [n]} g_{j(n+1)})$ and $e(d, \prod_{j \in [n]} g_{\rho(j)(n+2)-j})$. They have discrete logarithms

$$r \sum_{j \in [n]} x^{j(n+1)} + \sum_{i \in [n]} a_i x^{i(n+2)} + \sum_{i \in [n]} \sum_{j \in [n] \setminus \{i\}} a_i x^{j(n+1)+i}$$
$$s \sum_{j \in [n]} x^{\rho(j)(n+2)-j} + \sum_{i \in [n]} b_i x^{\rho(i)(n+2)} + \sum_{i \in [n]} \sum_{j \in [n] \setminus \{i\}} b_i x^{\rho(j)(n+2)+i-j}$$

We have the desired sums $\sum_{i\in[n]} a_i x^{i(n+2)}$ and $\sum_{i\in[n]} b_i x^{\rho(i)(n+2)}$ but due to the extra terms it is not the case that $e(c,\prod_{j\in[n]} g_{j(n+1)}) = e(d,\prod_{j\in[n]} g_{\rho(j)(n+2)-j})$.

The prover will construct an argument π that cancels out the extra terms and the verifier will check that

$$e(c,\prod_{j\in[n]} g_{j(n+1)}) = e(d,\prod_{j\in[n]} g_{\rho(j)(n+2)-j})e(g,\pi).$$

The prover also gives a restriction argument $\hat{\pi},\dot{\pi}$ such that the verifier is guaranteed that π does not contain any $x^{n+2},\ldots,x^{n(n+2)}$ terms. Soundness now follows from the verification equation giving us $\sum_{i\in[n]} a_i x^{i(n+2)} = \sum_{i\in[n]} b_i x^{\rho(i)(n+2)}$ when π is free of $x^{n+2},\ldots,x^{n(n+2)}$ terms.

Statement: Commitments $c,d\in G$ and permutation $\rho\in S_n$.
Prover's witness: Openings $r,a_1,\ldots,a_n\in\mathbb{Z}_p$ and $s,b_1,\ldots,b_n\in\mathbb{Z}_p$ so

$$c = g^r\prod_{i\in[n]} g_i^{a_i} \quad\text{and}\quad d = g^s\prod_{i\in[n]} g_i^{b_i} \quad\text{and}\quad \forall i\in[n]: b_i = a_{\rho(i)}.$$

Argument: Compute the argument as

$$\begin{aligned}
\pi &= \prod_{j\in[n]} g_{j(n+1)}^{r} g_{\rho(j)(n+2)-j}^{-s} \prod_{i\in[n]}\prod_{j\in[n]\setminus\{i\}} g_{j(n+1)+i}^{a_i} g_{\rho(j)(n+2)+i-j}^{-b_i}\\
\hat{\pi} &= \prod_{j\in[n]} \hat{g}_{j(n+1)}^{r} \hat{g}_{\rho(j)(n+2)-j}^{-s} \prod_{i\in[n]}\prod_{j\in[n]\setminus\{i\}} \hat{g}_{j(n+1)+i}^{a_i} \hat{g}_{\rho(j)(n+2)+i-j}^{-b_i}\\
\dot{\pi} &= \prod_{j\in[n]} \dot{h}_{j(n+1)}^{r} \dot{h}_{\rho(j)(n+2)-j}^{-s} \prod_{i\in[n]}\prod_{j\in[n]\setminus\{i\}} \dot{h}_{j(n+1)+i}^{a_i} \dot{h}_{\rho(j)(n+2)+i-j}^{-b_i}
\end{aligned}$$

Verification: Output 1 if and only if $e(g,\hat{\pi}) = e(\pi,\hat{g})$, $e(g,\dot{\pi}) = e(\pi,\dot{h})$ and $e(c,\prod_{j\in[n]} g_{j(n+1)}) = e(d,\prod_{j\in[n]} g_{\rho(j)(n+2)-j})e(g,\pi)$.

Theorem 4. *The permutation argument has perfect completeness and perfect witness-indistinguishability. If the q-CPDH assumption holds, a non-uniform probabilistic polynomial time adversary has negligible chance of outputting a permutation ρ, commitments (c,d) and an acceptable argument $(\pi,\hat{\pi},\dot{\pi})$ with corresponding openings of the commitments and the argument such that for some $i\in[n]$ we have $b_i\neq a_{\rho(i)}$.*

The proof can be found in the full paper.

8 Constant Size NIZK Argument for Circuit Satisfiability

We will now give an NIZK argument for the satisfiability of a NAND-gate circuit C, which consists of a constant number of group elements but has a large common reference string. Let a be the output wire of the circuit and add an extra self-looping NAND gate $a = \neg(a\wedge b)$ to force a to be true. This reduces the satisfiability problem to demonstrating that there is a truth-value assignment to the wires such that C is internally consistent with all the NAND-gates. In the following let the value -1 correspond to false and $+1$ correspond to true. We now give the full NIZK argument outlined in the introduction.

CRS: Generate common reference string $\sigma = (ck, \tilde{\sigma}, \bar{\sigma}, \dot{\sigma})$ with $n = 2N$.

Statement: A circuit C with N NAND-gates, where we want to prove the wires can be assigned values such that the circuit is internally consistent.

Witness: $2N$ input values $a_1, \ldots, a_N, b_1, \ldots, b_N \in \{-1, 1\}$ for the N gates that are consistent with the wires in the circuit and respect the NAND-gates. Define $u_1, \ldots, u_N$ to be values of the output wires and let $r_1, \ldots, r_N$ be the remaining values in $(a_1, \ldots, a_N, b_1, \ldots, b_N)$ (either inputs to the circuit or duplicates of NAND-gate output wires appearing multiple times as inputs to other NAND-gates).

Argument:

1. Make restricted commitment $(c_{a\|b}, \hat{c}_{a\|b}, \tilde{c}_{a\|b})$ to $(a_1, \ldots, a_N, b_1, \ldots, b_N)$.
2. Make restricted commitment $(d_{a\|b}, \hat{d}_{a\|b}, \bar{d}_{a\|b})$ to $(a_1, \ldots, a_N, b_1, \ldots, b_N)$.
3. Make restricted commitment $(c_{b\|a}, \hat{c}_{b\|a}, \tilde{c}_{b\|a})$ to $(b_1, \ldots, b_N, a_1, \ldots, a_N)$.
4. Make restricted commitment $(c_{b\|0}, \hat{c}_{b\|0}, \tilde{c}_{b\|0})$ to $(b_1, \ldots, b_N, 0, \ldots, 0)$.
5. Make restricted commitment $(c_{u\|r}, \hat{c}_{u\|r}, \tilde{c}_{u\|r})$ to $(u_1, \ldots, u_N, r_1, \ldots, r_N)$.
6. Make restricted comm. $(c_{-u\|0}, \hat{c}_{-u\|0}, \tilde{c}_{-u\|0})$ to $(-u_1, \ldots, -u_N, 0, \ldots, 0)$.
7. Show that $c_{a\|b}$ and $d_{a\|b}$ contain the same values by giving a product argument for $c_{a\|b}$ containing the entry-wise product of the values in $\prod_{i=1}^{2N} g_i$ (a commitment to $(1, \ldots, 1, 1, \ldots, 1)$) and $d_{a\|b}$.
8. Show that $a_1, \ldots, a_N, b_1, \ldots, b_N \in \{-1, 1\}$ by giving a product argument for $\prod_{i=1}^{2N} g_i$ (a commitment to $(1, \ldots, 1, 1, \ldots, 1)$) containing the entry-wise product of the values in $c_{a\|b}$ and $d_{a\|b}$.
9. Show that the values are internally consistent with the wires. The values $a_{i_1}, \ldots, a_{i_\ell}, b_{j_1}, \ldots, b_{j_m}$ may for instance all correspond to the same wire. Pick a permutation $\rho \in S_{2N}$ such that it contains cycles of the form $i_1 \to i_2 \to \ldots \to i_\ell \to j_1 + N \to j_2 + N \to \ldots \to j_m + N \to i_1$ for all sets of values corresponding to the same wire. Give a permutation argument for $c_{a\|b}$ containing the ρ-permutation of the values in $c_{a\|b}$. For each set corresponding to the same wire, this shows $a_{i_2} = a_{i_1}, \ldots, b_{j_1} = a_{i_\ell}, \ldots, b_{j_m} = b_{j_{m-1}}$ so the values must be the same.
10. Give a permutation argument for $c_{u\|r}$ and $c_{a\|b}$ showing that the outputs values $(u_1, \ldots, u_n)$ are consistent with the input values $(a_1, \ldots, a_N, b_1, \ldots, b_N)$. (The $(r_1, \ldots, r_N)$ values are the remaining N values in $(a_1, \ldots, a_N, b_1, \ldots, b_N)$ that correspond to duplicates of an output wire or input wires to the circuit.
11. Give a permutation argument for $c_{b\|a}$ containing the swap of the values in $c_{a\|b}$.
12. Give a product argument for $c_{b\|0}$ containing the entry-wise product of the values in $c_{b\|a}$ and $\prod_{j=1}^{N} g_{j(n+1)}$ (contains $(1, \ldots, 1, 0, \ldots, 0)$).
13. Give a product argument for $c_{-u\|0}$ containing the entry-wise product of the values in $c_{u\|r}$ and $\prod_{j=1}^{N} g_{j(n+1)}^{-1}$ (contains $(-1, \ldots, -1, 0, \ldots, 0)$).

14. Show the NAND-gates are respected by giving a product argument for $c_{-u\|0}$ containing the entry-wise product of the values in $c_{b\|0}$ and $d_{a\|b}$.

The argument consists of the 6 knowledge commitments with corresponding restriction arguments, the 5 product arguments and the 3 permutation arguments given above. The total size is 42 group elements.

Verification: Accept the argument if and only if the 6 knowledge commitments are well-formed, their corresponding restriction arguments are acceptable, the 5 product arguments are acceptable and the 3 permutation arguments are acceptable.

Theorem 5. *The NIZK argument for circuit satisfiability is perfectly complete and perfectly zero-knowledge. If the q-PKE and q-CPDH assumptions hold with* $q = (4N^2 + 6N - 2)$*, then the NIZK argument is computationally sound.*

The proof can be found in the full paper.

Arithmetic circuits. It is possible to adjust our NIZK argument to handle arithmetic circuits consisting of addition and multiplications gates over $\mathbb{Z}_p$. The commitment scheme is homomorphic so if we multiply two commitments we get the sum of their values, which can be used to handle the addition gates. The multiplication gates can be handled with our product arguments.

9 Reducing the Common Reference String

In the last section, we constructed constant size NIZK arguments. The common reference string, however, grows quadratically in the size of the circuit. If the NIZK argument is only used a few times the cost of setting up the common reference string may be prohibitive. In this section, we will outline how to reduce the size of the common reference string in return for increasing the size of the argument. If the circuit has $2N = n^d$ wires for some constant $d \geq 1$ we get a common reference string with $O(n^2)$ group elements and an NIZK argument with $O(n^{d-1})$ group elements. If we choose $d = 3$, the combined size of the CRS and the NIZK argument is $O(N^{2/3})$ group elements making both components sub-linear in the circuit size.

The idea is to reduce the common reference string and let each commitment hold fewer values. If we have a circuit with n^d wires and a common reference string of size $q = n^2 + 3n - 2 = O(n^2)$, the set $\tilde{S}$ will permit the commitment of n elements at a time. Each commitment is a constant number of group elements, but now we use n^{d-1} commitments to commit to all the $2N = n^d$ input values to the gates. The product and permutation arguments are also of constant size, but they only work on commitments to n values. If we look at our NIZK argument, the product argument can be used on each of the n^{d-1} triples of commitments containing n values each so there is no problem here. The permutation argument is not useful though, because we need to permute $2N = n^d$ committed values spread across n^{d-1} commitments. The goal in this section is to build a permutation argument for two n^{d-1}-tuples of commitments to a total of $2N = n^d$ values each. The permutation argument consists of $O(n^{d-1})$ group elements and uses the existing CRS consisting of $O(n^2)$ group elements.

9.1 Permutation Argument Spanning Multiple Commitments

Consider two sets of n commitments $c_1, \ldots, c_n, d_1, \ldots, d_n$ to values $a_{11}, \ldots, a_{nn}$ and $b_{11}, \ldots, b_{nn}$. We will use a Clos-network [12] to give an argument for the two sets of committed values being permutations of each other for a publicly known permutation $\rho \in S_{n^2}$. The idea in a Clos network is to build large permutations from smaller permutations. Consider a permutation $\rho \in S_{n^2}$. First we divide the elements into n blocks of n elements and permute the elements within each block. Next, we distribute the elements in each block evenly on n other blocks giving us a new set of n blocks each containing one element from each of the previous blocks. We permute the elements in each block again. Once again, we distribute the elements in each block evenly on n new blocks. Finally, we permute the elements within the last blocks to get the elements permuted in the desired order. The permutations in the Clos network vary depending on ρ, whereas the distributions between blocks are fixed and independent of ρ.

To give a permutation argument for $\{c_i\}_{i\in[n]}, \{d\}_{i\in[n]}$ containing the same values permuted according to $\rho \in S_{n^2}$ the prover builds a Clos-network for the permutation ρ. She constructs 4 sets of n intermediate commitments $\{c'_i\}_{i\in[n]}, \{v_i\}_{i\in[n]}, \{v'_i\}_{i\in[n]}, \{d'_i\}_{i\in[n]}$ together with arguments of knowledge and restriction arguments. Each commitment contains a block of n values in the middle stages of the Clos network. She uses the permutation argument from Section 7 to show that for all $i \in [n]$ the pairs of commitments $(c_i, c'_i), (d_i, d'_i)$ and (v_i, v'_i) contain the same elements in permuted order as dictated by $\rho \in S_{n^2}$. The remaining problem is to give an argument for having dispersed the values between $\{c'_i\}_{i\in[n]}$ and $\{v_j\}_{j\in[n]}$ such that for each c'_i the values have been dispersed to n different v_j's and to give a dispersion argument for having spread the values in $\{v'_i\}_{i\in[n]}$ to $\{d'_j\}_{j\in[n]}$ such that for each v'_i the n committed values have been dispersed to n different d'_js. We present a dispersion argument in Section 9.2, which uses the existing CRS consisting of $O(n^2)$ group elements and has an argument size of $O(n)$ group elements. Counting the cost of commitments, within-block permutation arguments and the dispersion arguments, we get a total size of $O(n)$ group elements for proving that two sets of n commitments to n values each are related by a publicly known permutation $\rho \in S_{n^2}$.

Once we have a permutation argument for n^2 values spread over n commitments, we can recursively get permutation arguments for larger permutations. The cost for a permutation of n^d elements spread over two sets of n^{d-1} commitments is $O(n^{d-1})$ group elements for any constant d.

9.2 Dispersion Argument

Consider a matrix of n^2 values $a_{11}, \ldots, a_{nn}$. We can view commitments $c_1, \ldots, c_n$ given by $c_j = g^{r_j} \prod_{i\in[n]} g_i^{a_{ij}}$ as commitments to the columns of the matrix. Similarly, we can view $d_1, \ldots, d_n$ given by $d_i = g^{s_i} \prod_{i\in[n]} g_{j(n+1)}^{a_{ij}}$ as commitments to the rows of the matrix. We give an argument for demonstrating that $c_1, \ldots, c_n$ and $d_1, \ldots, d_n$ contain respectively the columns and the rows of the same $n \times n$

matrix. This means that for each c_j the n committed values have been distributed to n different commitments $d_1, \dots, d_n$.

To get some intuition for the construction consider first the simple case where all the randomizers are 0. We then have

$$\prod_{j\in[n]} e(c_j, g_{j(n+1)}) = \prod_{i\in[n]} (g_i, d_i).$$

Taking discrete logarithms on both sides of the equation we get

$$\sum_{j\in[n]}\sum_{i\in[n]} a_{ij}x^{j(n+1)+i} = \sum_{i\in[n]}\sum_{j\in[n]} b_{ij}x^{j(n+1)+i},$$

which by Lemma 1 implies $a_{ij} = b_{ij}$ for all $i, j \in [n]$. Due to the randomizers this verification equation will not hold in general though. The prover therefore constructs an argument $(\pi_L, \pi_R, \hat{\pi}_L, \hat{\pi}_R, \bar{\pi}_L, \tilde{\pi}_R)$ consisting of six group elements such that the cross-terms arising from the randomizers cancel out.

Statement: Commitments $c_1, \dots, c_n, d_1, \dots, d_n \in G$.
Prover's witness: Openings $r_1, \dots, r_n, a_{11}, \dots, a_{nn}, s_1, \dots, s_n, b_{11}, \dots, b_{nn}$

$$\forall i, j \in [n]: \quad c_j = g^{r_j}\prod_{i\in[n]} g_i^{a_{ij}} \qquad d_i = g^{s_i}\prod_{j\in[n]} g_{j(n+1)}^{b_{ij}} \qquad a_{ij} = b_{ij}.$$

Argument: Pick $t \leftarrow \mathbb{Z}_p$ at random and compute the argument $(\pi_L, \pi_R, \hat{\pi}_L, \hat{\pi}_R, \bar{\pi}_L, \tilde{\pi}_R)$ as

$$\begin{aligned}
\pi_L &= g^t \prod_{j\in[n]} g_{j(n+1)}^{-r_j} & \pi_R &= g^t \prod_{i\in[n]} g_i^{-s_i}\\
\hat{\pi}_L &= \hat{g}^t \prod_{j\in[n]} \hat{g}_{j(n+1)}^{-r_j} & \hat{\pi}_R &= \hat{g}^t \prod_{i\in[n]} \hat{g}_i^{-s_i}\\
\bar{\pi}_L &= \bar{h}^t \prod_{j\in[n]} \bar{h}_{j(n+1)}^{-r_j} & \tilde{\pi}_R &= \tilde{h}^t \prod_{i\in[n]} \tilde{h}_i^{-s_i}
\end{aligned}$$

Verification: Output 1 if and only if

$$e(g, \hat{\pi}_R) = e(\pi_R, \hat{g}) \qquad e(g, \tilde{\pi}_R) = e(\pi_R, \tilde{h}) \qquad e(g, \hat{\pi}_L) = e(\pi_L, \hat{g})$$

$$e(g, \bar{\pi}_L) = e(\pi_L, \bar{h}) \qquad e(g, \pi_L)\prod_{j\in[n]} e(c_j, g_{j(n+1)}) = e(g, \pi_R)\prod_{i\in[n]} e(g_i, d_i).$$

Theorem 6. *The dispersion argument is perfectly complete and perfectly witness-indistinguishable. If the q-CPDH assumption holds, a non-uniform probabilistic polynomial time adversary has negligible chance of producing commitments $c_1, \dots, c_n, d_1, \dots, d_n$ and an accepting argument $(\pi_L, \pi_R, \hat{\pi}_L, \hat{\pi}_R, \bar{\pi}_L, \tilde{\pi}_R)$ with corresponding openings of the commitments and the argument such that $c_1, \dots, c_n$ and $d_1, \dots, d_n$ are commitments to two different matrices.*

We refer to the full paper for the proof.

References

1. Masayuki Abe and Serge Fehr. Perfect NIZK with adaptive soundness. In *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 118–136, 2007.
2. Boaz Barak, Ran Canetti, Jesper Buus Nielsen, and Rafael Pass. Universally composable protocols with relaxed set-up assumptions. In *FOCS*, pages 186–195, 2004.
3. Mihir Bellare, Alexandra Boldyreva, and Adriana Palacio. An uninstantiable random-oracle-model scheme for a hybrid encryption problem. In *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 171–188, 2004.
4. Mihir Bellare and Adriana Palacio. Towards plaintext-aware public-key encryption without random oracles. In *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 48–62, 2004.
5. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM CCS*, pages 62–73, 1993.
6. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *STOC*, pages 103–112, 1988.
7. Xavier Boyen and Brent Waters. Compact group signatures without random oracles. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 427–444, 2006.
8. Xavier Boyen and Brent Waters. Full-domain subgroup hiding and constant-size group signatures. In *PKC*, volume 4450 of *Lecture Notes in Computer Science*, pages 1–15, 2007.
9. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. In *STOC*, pages 209–218, 1998.
10. Ran Canetti, Oded Goldreich, and Shai Halevi. On the random-oracle methodology as applied to length-restricted signature schemes. In *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 40–57, 2004.
11. Nishanth Chandran, Jens Groth, and Amit Sahai. Ring signatures of sub-linear size without random oracles. In *ICALP*, volume 4596 of *Lecture Notes in Computer Science*, pages 423–434, 2007.
12. Charles Clos. A study of non-blocking switching networks. *Bell System Technical Journal*, 32(2):406–424, 1953.
13. Ivan Damgård. Non-interactive circuit based proofs and non-interactive perfect zero-knowledge with preprocessing. In *EUROCRYPT*, volume 658 of *Lecture Notes in Computer Science*, pages 341–355, 1992.
14. Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 566–598, 2002.
15. Alfredo De Santis, Giovanni Di Crescenzo, and Giuseppe Persiano. Randomness-optimal characterization of two NP proof systems. In *RANDOM*, volume 2483 of *Lecture Notes in Computer Science*, pages 179–193, 2002.
16. Alfredo De Santis and Giuseppe Persiano. Zero-knowledge proofs of knowledge without interaction. In *FOCS*, pages 427–436, 1992.
17. Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. *SIAM Journal of Computing*, 30(2):391–437, 2000.
18. Cynthia Dwork and Moni Naor. Zaps and their applications. In *FOCS*, pages 283–293, 2000.
19. Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs under general assumptions. *SIAM Journal of Computing*, 29(1):1–28, 1999.

20. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178, 2009.
21. Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal of Computing*, 25(1):169–192, 1996.
22. Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, 1994.
23. Shafi Goldwasser and Yael Tauman Kalai. On the (in)security of the Fiat-Shamir paradigm. In *FOCS*, pages 102–113, 2003.
24. Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proofs. *SIAM Journal of Computing*, 18(1):186–208, 1989.
25. Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In *ASIACRYPT*, volume 4248 of *Lecture Notes in Computer Science*, pages 444–459, 2006.
26. Jens Groth. Linear algebra with sub-linear zero-knowledge arguments. In *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 192–208, 2009.
27. Jens Groth. Short non-interactive zero-knowledge proofs. In *ASIACRYPT*, volume ???? of *Lecture Notes in Computer Science*, pages ???–???, 2010.
28. Jens Groth and Rafail Ostrovsky. Cryptography in the multi-string model. In *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 323–341, 2007.
29. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In *CRYPTO*, volume 4117 of *Lecture Notes in Computer Science*, pages 97–111, 2006.
30. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero-knowledge for NP. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 339–358, 2006.
31. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432, 2008.
32. Joe Kilian. A note on efficient zero-knowledge proofs and arguments. In *STOC*, pages 723–732, 1992.
33. Joe Kilian and Erez Petrank. An efficient noninteractive zero-knowledge proof system for NP with general assumptions. *Journal of Cryptology*, 11(1):1–27, 1998.
34. Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39, 2004.
35. Silvio Micali. Computationally sound proofs. *SIAM Journal of Computing*, 30(4):1253–1298, 2000.
36. Moni Naor. On cryptographic assumptions and challenges. In *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 96–109, 2003.
37. Jesper Buus Nielsen. Separating random oracle proofs from complexity theoretic proofs: The non-committing encryption case. In *CRYPTO*, volume 2442 of *Lecture Notes in Computer Science*, pages 111–126, 2002.
38. Amit Sahai. Non-malleable non-interactive zero-knowledge and adaptive chosen-ciphertext security. In *FOCS*, pages 543–553, 2001.