Non-interactive Zaps and New Techniques for NIZK

Jens Groth[1⋆], Rafail Ostrovsky[1⋆⋆], and Amit Sahai[1⋆⋆⋆]

UCLA, Computer Science Department
4732 Boelter Hall
Los Angeles, CA 90095, USA
{jg,rafail,sahai}@cs.ucla.edu

Abstract. In 2000, Dwork and Naor proved a very surprising result: that there exist "Zaps", two-round witness-indistinguishable proofs in the plain model without a common reference string, where the Verifier asks a single question and the Prover sends back a single answer. This left open the following tantalizing question: does there exist a *non-interactive* witness indistinguishable proof, where the Prover sends a single message to the Verifier for some non-trivial NP-language? In 2003, Barak, Ong and Vadhan answered this question affirmatively by derandomizing Dwork and Naor's construction under a complexity theoretic assumption, namely that Hitting Set Generators against co-nondeterministic circuits exist.

In this paper, we construct non-interactive Zaps for all NP-languages. We accomplish this by introducing new techniques for building Non-Interactive Zero Knowledge (NIZK) Proof and Argument systems, which we believe to be of independent interest, and then modifying these to yield our main result. Our construction is based on the Decisional Linear Assumption, which can be seen as a bilinear group variant of the Decisional Diffie-Hellman Assumption.

Furthermore, our single message witness-indistinguishable proof for Circuit Satisfiability is of size $O(k|C|)$ bits, where k is a security parameter, and $|C|$ is the size of the circuit. This is much more efficient than previous constructions of 1- or 2-move Zaps.

Keywords: Non-interactive zero-knowledge, witness indistinguishability, bilinear groups, Decisional Linear Assumption.

1 Introduction

In 2000, Dwork and Naor [DN00] proved a very surprising result: that there exist "Zaps", two-round Witness-Indistinguishable (WI) proofs in the plain model

⋆ Supported by NSF grant No. 0456717, and NSF Cybertrust grant.

⋆⋆ Supported in part by a gift from Teradata, Intel equipment grant, NSF Cybertrust grant No. 0430254, OKAWA research award, B. John Garrick Foundation and Xerox Innovation group Award.

⋆⋆⋆ Supported by grant No. 0456717 from the NSF ITR and Cybertrust programs, an equipment grant from Intel, and an Alfred P. Sloan Foundation Research Fellowship.

without a common reference string, where the Verifier asks a single question and the Prover sends back a single answer. This left open the following tantalizing question: does there exist a *non-interactive* witness indistinguishable proof, where the Prover sends a single message to the Verifier for some non-trivial NP-language? Such zaps were shown to have a number of fascinating and important applications, beyond the numerous applications of WI proofs already present in the literature.

In this paper, we introduce new techniques for constructing Non-Interactive Zero Knowledge (NIZK) Proofs and Arguments, based on the hardness of computational problems that arise in bilinear groups. Based on these new techniques, we are able to construct *non-interactive* Witness-Indistinguishable proofs for any NP relation, without any setup assumptions, based on a number-theoretic computational assumption. Furthermore, our construction is significantly more efficient than previous constructions of zaps, as we discuss below. We believe our new techniques for NIZK will have a number of other applications, as well. In the remainder of this introduction, we describe our setting and our results, and present our results in the context of previous work.

OUR SETTING. Throughout the paper we will make use of groups of prime order equipped with non-trivial bilinear maps. In other words, we let $\mathbb{G}, \mathbb{G}_T$ be abelian groups of order p, and let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a non-degenerate bilinear map such that $e(u^a, v^b) = e(u, v)^{ab}$. Such groups have been widely used in cryptography in recent years.

Our underlying security assumption is the Decisional Linear Assumption: Given groups elements $(g, f = g^x, h = g^y, f^r, h^s, g^d)$ for $x, y \leftarrow \mathbb{Z}_p^*$ and $r, s \leftarrow \mathbb{Z}_p$, it is hard to distinguish between the case where $d = r + s$ or d is random. The assumption was introduced by Boneh, Boyen and Shacham in [BBS04]. The assumption gives rise to an ElGamal-like cryptosystem with public key $pk = (p, \mathbb{G}, \mathbb{G}_T, e, g, f, h)$, where $f = g^x, h = g^y$ and the secret key is $sk = (x, y)$. Encryption of $m \in \mathbb{G}$ is done by picking $r, s \leftarrow \mathbb{Z}_p$ at random and letting the ciphertext be $(f^r, h^s, g^{r+s}m)$. An encryption of 1 is called a *linear tuple* (with respect to f, h, g).

OUR TECHNIQUES AND RESULTS. The conceptual starting point for our work is our recent work [GOS06], which constructed NIZK proofs and arguments for any NP relation, based on a different computational assumption for bilinear groups of composite order, called the Subgroup Decision Assumption. In that paper, we gave a construction for NIZK proof systems, such that if the Common Reference String (CRS) was of one form, it would be perfectly sound and computational ZK; whereas if the CRS was of a different form, then the *same* construction would yield a system that is computationally sound but perfectly ZK.

Our key idea for achieving non-interactive WI proofs *without* a CRS is as follows: If we could somehow force the prover to produce a perfect soundness CRS on its own, we would be done – but this is not possible. Instead, can we somehow force a prover to produce *two* CRS's, such that at least *one* is of the perfect soundness type?

Unfortunately, in the original GOS proof system, the CRS's that force perfectly sound proofs are negligibly rare, and are computationally indistinguishable from CRS's that give only computational soundness (and indeed these CRS's have trapdoors allowing proofs of false theorems).

The main technical contribution of our paper is to construct a new NIZK system based on the Decisional Linear Assumption where perfect soundness CRS's are common, whereas computational soundness CRS's are negligibly rare. Furthermore, in our new system, we show that a simple operation – multiplication of one element in the CRS by a generator – can always transform a computational soundness CRS into a perfect soundness CRS. (Roughly speaking, we accomplish the following: if the CRS is a linear tuple, then we obtain a computationally sound proof system; whereas if the CRS is *any* non-linear tuple, then we obtain a perfectly sound proof system.) This allows us to achieve non-interactive WI proofs as follows: The prover can generate a CRS on its own, but it must provide proofs under *both* the chosen CRS *and* the transformation of that CRS. This forces perfect soundness. We show that the WI property still holds because of a hybrid argument.

We note that our constructions yield NIZK proofs and non-interactive WI proofs for Circuit Satisfiability where the proof size is $O(k|C|)$ bits, where k is the security parameter, and C is the size of the circuit. For NIZK proofs this matches the previous best bound by [GOS06], which relies on the Subgroup Decision assumption. Our NIZK proofs[1] have the advantage of being realizable in the Common *Random* String model, whereas the constructions of [GOS06] required the Common Reference String Model. For WI proofs, as far as we know, our proof size is a significant improvement over all previous constructions of zaps for NP relations.

We believe our techniques and ideas for constructing NIZK proofs using the Decisional Linear Assumption will have other applications, as well. In a companion paper, Groth [Gro06] constructs a wide variety of novel and efficient NIZK proofs under the Decisional Linear Assumption, and uses these to obtain group signatures and other important applications.

Previous Work and Context for our Work. NIZK proofs were introduced by Blum, Feldman, and Micali [BFM88], following the introduction of interactive Zero-Knowledge proofs by Goldwasser, Micali, and Rackoff [GMR89]. Witness-Indistinguishable protocols were introduced by Feige and Shamir [FS90].

Dwork and Naor [DN00] constructed 2-round WI proofs, called zaps[2], for any NP relation (assuming trapdoor permutations exist), and showed a wide variety of applications for zaps. Furthermore, [DN00] showed that their constructions allowed for the first message (from Verifier to Prover) to be reused – so that between a particular pair of prover and verifier, only one message from verifier to prover is required even if many statements are to be proven. Barak, Ong, and Vadhan [BOV03] constructed the first non-interactive zaps for any NP relation

[1] These are computational zero knowledge, perfectly sound proofs.

[2] In the spirit of the name, we interpret zaps to mean WI proofs that require 2 rounds *or less*.

by applying derandomization techniques to the construction of Dwork and Naor, based on trapdoor permutations and the assumption that (very good) Hitting Set Generators (HSG) against co-nondeterministic circuits exist. It is known that such HSG's can be built if there is a function in E that requires exponential-size *nondeterministic* circuits – *i.e.* the assumption states that some uniform exponential deterministic computations can (only) be sped up by at most a constant power (Time 2^{cn} becomes $2^{\varepsilon n}$), when given the added power of nondeterminism and advice specific to the length of the input.

We mainly wish to emphasize that our construction is completely different and uses completely different, number-theoretic computational assumptions. Furthermore, our construction is much more efficient than both the constructions of Dwork-Naor and Barak-Ong-Vadhan (even when these constructions are instantiated with very efficient NIZK proofs such as [GOS06]).

A further point of comparison would be to look more closely at the assumptions used, for instance in the context of Naor's classification of assumptions based on falsifiability [Nao03]. While our assumption, the Decisional Linear Assumption, is an "efficiently falsifiable" assumption according to Naor's classification, it appears that the assumption about the existence of HSG's against co-nondeterministic circuits, or the assumption about functions in E with large nondeterministic circuits, are "none of the above" assumptions according to Naor's classification, since we wouldn't have time to actually "run" a suggested nondeterministic (or co-nondeterministic) circuit that claims to break the assumption.[3]

2 Definitions: Non-interactive Proofs

Let R be an efficiently computable binary relation. For pairs $(x, w) \in R$ we call x the statement and w the witness. Let L be the language consisting of statements in R.

A non-interactive proof system for a relation R consists of a CRS generation algorithm K, a prover P and a verifier V. The CRS generation algorithm produces a common reference string σ. The prover takes as input (σ, x, w) and produces a proof π. The verifier takes as input (σ, x, π) and outputs 1 if the

[3] We note that there is some uncertainty as to how to interpret Naor's classification with respect to these derandomization-style assumptions. We take a view that we think is consistent with the spirit of Naor's classification by asking the question – if the assumption is false, then is there necessarily a reasonably efficient (PPT) algorithmic demonstration of the falsehood of this assumption? To us, it appears that the answer is "Yes" for our assumption, but appears to be "No" for the [BOV03] assumptions; this is simply because for the latter assumptions, it is important that the breaking algorithm could be non-deterministic – and if it is, then how can we efficiently verify that it indeed does break the assumption? It would be very interesting if in fact there were a positive answer to this. Of course the question of falsifiability is less important than the question of whether an assumption is actually true; alas, we find ourselves unequipped to address this issue.

proof is acceptable and 0 if rejecting the proof. We call (K, P, V) a proof system for R if it has the completeness and soundness properties described below.

PERFECT COMPLETENESS. For all adversaries $\mathcal{A}$ we have

$$\Pr\left[\sigma \leftarrow K(1^k); (x, w) \leftarrow \mathcal{A}(\sigma); \pi \leftarrow P(\sigma, x, w) : V(\sigma, x, \pi) = 1 \text{ if } (x, w) \in R\right] = 1.$$

PERFECT SOUNDNESS. For all adversaries $\mathcal{A}$ we have

$$\Pr\left[\sigma \leftarrow K(1^k); (x, \pi) \leftarrow \mathcal{A}(\sigma) : V(\sigma, x, \pi) = 0 \text{ if } x \notin L\right] = 1.$$

COMPUTATIONAL ZERO-KNOWLEDGE [FLS99]. We call (K, P, V) an NIZK proof for R if there exists a simulator $S = (S_1, S_2)$ with the following zero-knowledge property. For all non-uniform polynomial time adversaries $\mathcal{A}$ we have

$$\Pr\left[\sigma \leftarrow K(1^k) : \mathcal{A}^{P(\sigma,\cdot,\cdot)}(\sigma) = 1\right] \approx \Pr\left[(\sigma, \tau) \leftarrow S_1(1^k) : \mathcal{A}^{S(\sigma,\tau,\cdot,\cdot)}(\sigma) = 1\right],$$

where $S(\sigma, \tau, x, w) = S_2(\sigma, \tau, x)$ for $(x, w) \in R$ and both oracles output `failure` if $(x, w) \notin R$.

2.1 Witness Indistinguishablity

A prerequisite for NIZK proofs is the common reference string. However, many times a witness indistinguishable proof is sufficient. Witness indistinguishability means that an adversary cannot tell which of two possible witnesses w_1, w_2 that has been used in constructing the proof. We will show how to construct a WI proof system without any setup assumptions.

COMPUTATIONAL WITNESS INDISTINGUISHABILITY. We call (K, P, V) a non-interactive zap for R or a non-interactive WI proof for R in the plain model if for all non-uniform polynomial time interactive adversaries $\mathcal{A}$ we have

$$\begin{aligned} &\Pr\left[(x, w_1, w_2) \leftarrow \mathcal{A}(1^k); \pi \leftarrow P(1^k, x, w_1) : \mathcal{A}(\pi) = 1 \text{ and } (x, w_1), (x, w_2) \in R\right] \\ \approx &\Pr\left[(x, w_1, w_2) \leftarrow \mathcal{A}(1^k); \pi \leftarrow P(1^k, x, w_2) : \mathcal{A}(\pi) = 1 \text{ and } (x, w_1), (x, w_2) \in R\right]. \end{aligned}$$

A hybrid argument shows that this definition is equivalent to one where we give the adversary access to multiple proofs using either witness w_1 or w_2. The definition of perfect WI is similar, except there we require equality of the above probabilities for all adversaries.

3 Bilinear groups

BILINEAR GROUPS. We use two cyclic groups $\mathbb{G}, \mathbb{G}_T$ of order p, where p is a prime. We make use of a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. I.e., for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}$ we have $e(u^a, v^b) = e(u, v)^{ab}$. We require that $e(g, g)$ is a generator of $\mathbb{G}_T$ if g is a generator of $\mathbb{G}$. We also require that group operations, group membership and the bilinear map be efficiently computable.

Throughout the paper we let $\mathcal{G}$ be a randomized algorithm that takes a security parameter as input and outputs $(p, \mathbb{G}, \mathbb{G}_T, e, g)$ such that p is prime, $\mathbb{G}, \mathbb{G}_T$ are descriptions of groups of order p, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map as described above and g is a random generator of $\mathbb{G}$.

Boneh and Franklin [BF03] give an example of a bilinear group. Let $p = 2 \bmod 3$ be a prime, and choose a small ℓ so $q = \ell p - 1$ is prime and $p^2 \not| q + 1$. Then the elliptic curve $y^2 = x^3 + 1$ over $\mathbb{Z}_q$ has ℓp points. We can let $\mathbb{G}$ be the order p subgroup of this curve and $\mathbb{G}_T = \mathbb{F}^*_{q^2}$. The bilinear map is the modified Weil-pairing. To get a random generator g for this group, pick x at random such that $x^3 + 1$ is a square and let y be a randomly chosen squareroot. Then $g = (x, y)^\ell$ is a random generator for $\mathbb{G}$ provided $g \neq 1$.

We say the bilinear group is verifiable, if there is a verification algorithm that outputs 1 if and only if $(p, \mathbb{G}, \mathbb{G}_T, e, g)$ is a bilinear group. The bilinear group from [BF03] described above is verifiable, we just need to check that $p = 2 \bmod 3$ is a prime and g is a generator for $\mathbb{G}$.

Definition 1 (Decisional Linear Assumption). *We say the Decisional Linear Assumption holds for the bilinear group generator $\mathcal{G}$ if for all non-uniform polynomial time adversaries $\mathcal{A}$ we have*

$$\begin{aligned}\Pr\Big[&(p, \mathbb{G}, \mathbb{G}_1, e, g) \leftarrow \mathcal{G}(1^k); x, y \leftarrow \mathbb{Z}_p^*; r, s \leftarrow \mathbb{Z}_p : \\ &\qquad \mathcal{A}(p, \mathbb{G}, \mathbb{G}_T, e, g, g^x, g^y, g^{xr}, g^{ys}, g^{r+s}) = 1\Big] \\ \approx \Pr\Big[&(p, \mathbb{G}, \mathbb{G}_1, e, g) \leftarrow \mathcal{G}(1^k); x, y \leftarrow \mathbb{Z}_p^*; r, s, d \leftarrow \mathbb{Z}_p : \\ &\qquad \mathcal{A}(p, \mathbb{G}, \mathbb{G}_T, e, g, g^x, g^y, g^{xr}, g^{ys}, g^d) = 1\Big].\end{aligned}$$

The Decisional Linear Assumption was first introduced by Boneh, Boyen and Shacham [BBS04] and has since been used in several cryptographic constructions. We call a tuple of the form (f^r, h^s, g^{r+s}) a *linear tuple* with respect to (f, h, g). When the basis (f, h, g) is obvious from context, we omit mention of it.

4 Homomorphic Encryption and Commitment from Bilinear Maps

4.1 A Homomorphic Cryptosystem

We recall the homomorphic cryptosystem given by [BBS04]. It uses ideas similar to ElGamal encryption, but since the Decisional Diffie-Hellman (DDH) problem is easy in bilinear groups, we have to insert an extra element in the ciphertext.

Key generation:

1. $(p, \mathbb{G}, \mathbb{G}_1, e, g) \leftarrow \mathcal{G}(1^k)$

2. Let $x, y \leftarrow \mathbb{Z}_p^*$; let $f = g^x, h = g^y$
3. Let $pk = (p, \mathbb{G}, \mathbb{G}_T, e, g, f, h)$
4. Let $sk = (pk, x, y)$
5. Return (pk, sk)

Encryption: To encrypt $m \in \mathbb{G}$, let $r, s \leftarrow \mathbb{Z}_p$, and return $(u, v, w) = E(m; r, s) = (f^r, h^s, g^{r+s}m)$.

Decryption: To decrypt ciphertext $(u, v, w) \in \mathbb{G}^3$, return $m = D_{sk}(u, v, w) = u^{-1/x}v^{-1/y}w$.

The cryptosystem (K_{cpa}, E, D) has several nice properties. The Decisional Linear Assumption for $\mathcal{G}$ implies semantic security under chosen plaintext attack. All triples $(u, v, w) \in \mathbb{G}^3$ are valid ciphertexts. Also, the cryptosystem is homomorphic in the sense that

$$E(m_1; r_1, s_1)E(m_2, r_2, s_2) = E(m_1m_2; r_1 + r_2, s_1 + s_2).$$

4.2 A Homomorphic Commitment Scheme

We will use the cryptosystem to create a homomorphic commitment scheme with the property that depending on how we generate the public key we get either a perfectly hiding trapdoor commitment scheme or a perfectly binding commitment scheme.

Perfectly hiding key generation:
1. $(pk, sk) \leftarrow K_{\text{cpa}}(1^k)$
2. $r_u, s_v \leftarrow \mathbb{Z}_p$
3. $(u, v, w) = E(1; r_u, s_v) = (f^{r_u}, h^{s_v}, g^{r_u+s_v})$
4. Return $ck = (pk, u, v, w)$

Perfectly binding key generation:
1. $(pk, sk) \leftarrow K_{\text{cpa}}(1^k)$
2. $r_u, s_v \leftarrow \mathbb{Z}_p$
3. $(u, v, w) = E(m; r_u, s_v) = (f^{r_u}, h^{s_v}, g^{r_u+s_v}m)$, where $m = g^{\pm 1}$ can be arbitrarily chosen
4. Return $ck = (pk, u, v, w)$

Commitment: To commit to message $m \in \mathbb{Z}_p$ do
1. $r, s \leftarrow \mathbb{Z}_p$
2. Return $c = (c_1, c_2, c_3) = \text{com}(m; r, s) = (u^m f^r, v^m h^s, w^m g^{r+s})$

Trapdoor opening: Given a commitment $c = \text{com}(m; r, s)$ under a perfectly hiding commitment key we have $c = \text{com}(m'; r - (m' - m)r_u, s - (m' - m)s_v)$. So we can create a perfectly hiding commitment and open it to any value we wish if we have the trapdoor key (r_u, s_v).

The semantic security of the cryptosystem implies that no polynomial time adversary can distinguish between perfectly hiding keys and perfectly binding keys. This implies that the perfectly binding commitment scheme is computationally hiding, and the perfectly hiding commitment scheme is computationally binding.

5 NIZK proofs for Circuit Satisfiability

In this section, we show how to construct NIZK proofs for Circuit Satisfiability based on the Decisional Linear Assumption. To do this, we follow but somewhat change the general outline of the [GOS06] construction. We review this now:

In the GOS construction, and in ours, the overall approach is to commit[4] to the value of all the wires in the circuit (including the input wires) using an *additively homomorphic* commitment scheme, and then prove that for every gate in the circuit (W.L.O.G. a NAND gate), the 3 wires incident to the gate obey its rule. In [GOS06], we then showed how to reduce this task to just proving that a committed value is either 0 or 1. This is done by way of the homomorphic properties of the commitment scheme, together with the following simple observation: for three values $b_0, b_1, b_2 \in \{0, 1\}$, we have that $b_0 + b_1 + 2b_2 - 2 \in \{0, 1\}$ iff $b_2 = \neg(b_0 \wedge b_1)$.

In [GOS06], then all that was needed was a NIZK proof that a committed value is either 0 or 1. Here, we look a little closer at the GOS methodology, and take a slightly different route. This consists of two main observations:

1. First, we take a look at our homomorphic commitment scheme (given in the last section), and observe the following: Given a commitment $c = (c_1, c_2, c_3)$, the committed value being either 0 or 1 is equivalent to the following statement: that either c is a commitment to 0, *or* that $c' = (c_1/u, c_2/v, c_3/w)$ is a commitment to 0. Further, we note that a commitment (c_1, c_2, c_3) is a commitment to 0 iff it forms a linear tuple. Thus, we can equivalently prove that given two tuples, that either one or the other is a linear tuple, i.e., of the form (f^r, h^s, g^{r+s}).
2. Second, we take a closer look at the simulation strategy. The overall strategy is as follows: The CRS consists of the parameters for the homomorphic commitment scheme. As we have already observed, however, the Decisional Linear Assumption implies that a CRS that leads to perfectly binding commitments is indistinguishable from one that leads to perfectly hiding commitments. If we want perfect soundness for our NIZK proof system, then the "real-life" CRS should lead to perfectly binding commitments. The simulation can use a CRS of the perfectly hiding type, and the simulator can remember the trapdoor information that allows it to produce equivocal commitments that it can later open to any value.

   A key observation we make here is that the homomorphic properties of the commitment *preserves* equivocality: if one applies the homomorphic operations to multiple equivocal commitments, then the resulting commitment is still equivocal. So, we observe that the simulation can simply produce such equivocal commitments for each wire value, and then when it comes to

[4] In [GOS06] we called this an encryption. The fact that it was an encryption and not just a commitment is not important for the ZK property, and was used there to achieve proofs of knowledge. We can also obtain NIZK proofs of knowledge, but that is not our focus here.

proving that one of two commitments (that were generated via homomorphic operations) is a commitment to zero, the simulation will actually have the necessary information to prove this for *both* commitments. What this means is that we need the proof that one of two commitments is a commitment to zero (*i.e.* that one out of two tuples is a linear tuple) to merely be *witness-indistinguishable* rather than fully NIZK.

Before giving the NIZK proof for Circuit Satisfiability more formally, we first construct a (perfect) WI proof for one out of two tuples being a linear tuple.

5.1 Perfect WI proof

Consider the following situation. We have two tuples (A_1, A_2, A_3) and (B_1, B_2, B_3) with discrete logarithms (a_1, a_2, a_3) and (b_1, b_2, b_3) with respect to (f, h, g), where $f = g^x$ and $h = g^y$. We want to prove that $a_1 + a_2 + a_3 = 0$ or $b_1 + b_2 + b_3 = 0$. Note, this corresponds to $(A_1^{-1}, A_2^{-1}, A_3)$ or $(B_1^{-1}, B_2^{-1}, B_3)$ being a linear tuple. We will do this by showing that

$$0 = \sum_{i=1}^{3} \sum_{j=1}^{3} a_i b_j = (a_1 + a_2 + a_3)(b_1 + b_2 + b_3).$$

We first give the intuition behind our scheme, and then the formal description and proof of correctness. Using the bilinear map, we can compute

$$\begin{array}{rl} e(A_1, B_1) = e(f, f)^{a_1 b_1} & e(A_1, B_2)e(A_2, B_1) = e(f, h)^{a_1 b_2 + a_2 b_1} \\ e(A_2, B_2) = e(h, h)^{a_2 b_2} & e(A_1, B_3)e(A_3, B_1) = e(f, g)^{a_1 b_3 + a_3 b_1} \\ e(A_3, B_3) = e(g, g)^{a_3 b_3} & e(A_3, B_2)e(A_2, B_3) = e(h, g)^{a_2 b_3 + a_3 b_2} \end{array}$$

The goal is to show that these six exponents sum to 0.

Consider the following matrix

$$M = \begin{pmatrix} e(A_1, B_1) & e(f, h)^t e(A_1, B_2) & e(f, g)^{-t} e(A_1, B_3) \\ e(h, f)^{-t} e(A_2, B_1) & e(A_2, B_2) & e(h, g)^t e(A_2, B_3) \\ e(g, f)^t e(A_3, B_1) & e(g, h)^{-t} e(A_3, B_2) & e(A_3, B_3) \end{pmatrix},$$

with $t \leftarrow \mathbb{Z}_p$ chosen at random.

If both $a_1 + a_2 + a_3 = 0$ and $b_1 + b_2 + b_3 = 0$, then this matrix is distributed identically to its transpose. To see this, we observe that since $a_1(b_1 + b_2 + b_3) = b_1(a_1 + a_2 + a_3) = 0$, we have that $a_1 b_2 - a_2 b_1 = a_3 b_1 - a_1 b_3$. Similarly, we have that $a_1 b_2 - a_2 b_1 = a_2 b_3 - a_3 b_2$. Therefore if we set $t' = t + (a_1 b_2 - a_2 b_1) = t + (a_3 b_1 - a_1 b_3) = t + (a_2 b_3 - a_3 b_2)$, but interchange the roles of a_1, a_2, a_3 and b_1, b_2, b_3, we have the same matrix. This is what will give us witness indistinguishability. If $a_1 + 2_3 + a_3 \neq 0$ or $b_1 + b_2 + b_3 \neq 0$ we only have one witness and therefore we automatically have witness indistinguishability.

So, W.L.O.G., assume that we know a_1, a_2, a_3. We can rearrange the matrix as

$$\begin{pmatrix} e(f, B_1^{a_1}) & e(f, h^t B_2^{a_1}) & e(f, g^{-t} B_3^{a_1}) \\ e(h, f^{-t} B_1^{a_2}) & e(h, B_2^{a_2}) & e(h, g^t B_3^{a_2}) \\ e(g, f^t B_1^{a_3}) & e(g, h^{-t} B_2^{a_3}) & e(g, B_3^{a_3}) \end{pmatrix}.$$

In our proof system, we will reveal the 9 right-hand-side inputs to the bilinear maps for each entry of the matrix.

Observe, that we have $e(A_i, B_i) = M_{ii}$, so M_{ii} has exponent $a_i b_i$. We also have $e(A_i, B_j)e(A_j, B_i) = M_{ij}M_{ji}$, which has exponent $a_i b_j + a_j b_i$ for $i \neq j$. The verifier can check these equations, meaning he knows the sum of all the 9 exponents of M is $\sum_{i=1}^{3}\sum_{j=1}^{3} a_i b_j$. We therefore just need to show that the exponents of each of the 3 column vectors of M is 0.

Observe in the matrix above that for $j = 1, 2, 3$ we have $M_{1j}M_{2j}M_{3j} = 1$. This means we do not need to reveal M_{3j}, the verifier can compute it as $M_{3j} = M_{1j}^{-1}M_{2j}^{-1}$ himself. This also corresponds to asserting the fact that the column logarithms sum to 0: Taking discrete logarithms of these elements we have $m_{1j} + m_{2j} + m_{3j} = 0$.

These are the ideas in the WI proof, let us now write down the protocol.

Statement: A bilinear group $(p, \mathbb{G}, \mathbb{G}_T, e, g)$ and generators (f, h). The claim is that at least one of two given tuples (c_1, c_2, c_3) and (d_1, d_2, d_3) is a linear tuple with respect to f, h, g.

Witness: The witness is of the form (r, s) so $c = (f^r, h^s, g^{r+s})$ or $d = (f^r, h^s, g^{r+s})$.

Proof: Define $a_1 = -r, a_2 = -s, a_3 = r + s$. This means $a_1 + a_2 + a_3 = 0$.

If the prover has a witness for c then let $B_1 = d_1^{-1}, B_2 = d_2^{-1}, B_3 = d_3$, else let $B_1 = c_1^{-1}, B_2 = c_2^{-1}, B_3 = c_3$.

Choose $t \leftarrow \mathbb{Z}_p$ and let

$$\pi_{11} = B_1^{a_1} \qquad \pi_{12} = h^t B_2^{a_1} \qquad \pi_{13} = g^{-t} B_3^{a_1}$$

$$\pi_{21} = f^{-t} B_1^{a_2} \qquad \pi_{22} = B_2^{a_2} \qquad \pi_{23} = g^t B_3^{a_3}$$

Return the proof $\pi = (\pi_{11}, \pi_{12}, \pi_{13}, \pi_{21}, \pi_{22}, \pi_{23})$.

Verification: Compute $\pi_{3j} = (\pi_{1j}\pi_{2j})^{-1}$ for $j = 1, 2, 3$. For sake of notation consistent with the intuition above, let $\tilde{c_1} = c_1^{-1}$, $\tilde{c_2} = c_2^{-1}$, $\tilde{c_3} = c_3$, $\tilde{d_1} = d_1^{-1}$, $\tilde{d_1} = d_2^{-1}$, and $\tilde{d_1} = d_3$. Accept if and only if the bilinear group is correctly formed, and

$$\begin{aligned} e(f, \pi_{11}) &= e(\tilde{c}_1, \tilde{d}_1) & e(f, \pi_{12})e(h, \pi_{21}) &= e(\tilde{c}_1, \tilde{d}_2)e(\tilde{c}_2, \tilde{d}_1) \\ e(h, \pi_{22}) &= e(\tilde{c}_2, \tilde{d}_2) & e(f, \pi_{13})e(g, \pi_{31}) &= e(\tilde{c}_1, \tilde{d}_3)e(\tilde{c}_3, \tilde{d}_1) \\ e(g, \pi_{33}) &= e(\tilde{c}_3, \tilde{d}_3). & e(h, \pi_{23})e(g, \pi_{32}) &= e(\tilde{c}_2, \tilde{d}_3)e(\tilde{c}_3, \tilde{d}_2) \end{aligned}$$

Theorem 1. *The protocol described above is a non-interactive proof system for one of (c_1, c_2, c_3) or (d_1, d_2, d_3) being a linear tuple with respect to f, h, g. It has perfect completeness, perfect soundness and perfect witness-indistinguishability. The proof consists of 6 elements from $\mathbb{G}$.*

Proof.

Perfect completeness: This follows by straightforward computation.

Perfect soundness: Define r_c, s_c, t_c and r_d, s_d, t_d so $c = (f^{r_d}, h^{s_d}, g^{t_c})$ and $d = (f^{r_d}, h^{s_d}, g^{t_d})$.

For $i = 1, 2$ let

$$m_{i1} = \log_f(\pi_{i1}) \qquad m_{i2} = \log_h(\pi_{i1}) \qquad m_{i3} = \log_g(\pi_{i3}).$$

Let

$$m_{31} = -m_{11} - m_{21} \qquad m_{32} = -m_{12} - m_{22} \qquad m_{33} = -m_{13} - m_{23}.$$

From the equalities we get

$$\begin{array}{ll} m_{11} = r_c r_d & m_{12} + m_{21} = r_c s_d + s_c r_d \\ m_{22} = s_c s_d & m_{13} + m_{31} = -r_c t_d - t_c r_d \\ m_{33} = t_c t_d. & m_{23} + m_{32} = -s_c t_d - t_c s_d \end{array}$$

This means

$$\begin{aligned} & (r_c + s_c - t_c)(r_d + s_d - t_d) \\ = & \, r_c r_d + r_c s_d + s_c r_d + s_c s_d + t_c t_d - (r_c t_d + t_c r_d + s_c t_d + t_c s_d) \\ = & \sum_{i=1}^{3} \sum_{j=1}^{3} m_{ij} = 0. \end{aligned}$$

We conclude

$$t_c = r_c + s_c \quad \text{ or } \quad t_d = r_d + s_d.$$

Perfect witness indistinguishability: For WI, we may assume that both tuples are linear tuples. Define $a_1 = -r, a_2 = -s, a_3 = r + s$ and B_1, B_2, B_3 as in the proof. We define b_1, b_2, b_3 so $B_1 = f^{b_1}, B_2 = h^{b_2}, B_3 = g^{b_3}$, observe that $b_1 + b_2 + b_3 = 0$. In the proof we pick $t \leftarrow \mathbb{Z}_p$. Interchanging the roles of a_1, a_2, a_3 and b_1, b_2, b_3 and using $t' = t + (a_1 b_2 - a_2 b_1) = t + (a_3 b_1 - a_1 b_3) = t + (a_2 b_3 - a_3 b_2)$ leads to exactly the same proof. □

5.2 Circuit Satisfiability NIZK construction

Based on the intuition given earlier, we now give an NIZK proof for Circuit Satisfiability, based on the (perfect) WI proof that one out of two tuples is a linear tuple, given in the last section.

Common reference string:

1. $(p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathcal{G}(1^k)$
2. f, h random generators of $\mathbb{G}$
3. $u = f^{r_0}$, $v = h^{s_0}$, and $w = g^{r_0 + s_0} m$, for random r_0, s_0 in $\mathbb{Z}_p$ and $m = g$ or $m = g^{-1}$. Note that the choice of $m = g$ or $m = g^{-1}$ is arbitrary.
4. Return $\sigma = (p, \mathbb{G}, \mathbb{G}_T, e, g, f, h, u, v, w)$.

Statement: The statement is a circuit C built from NAND-gates. The claim is that there exist input bits w so $C(w) = 1$.

Proof: The prover has a witness w consisting of input bits so $C(w) = 1$.

1. Extend w to contain the bits of all wires in the circuit.
2. Commit to each bit w_i as a tuple $(c_1 = u^{w_i} f^r, c_2 = v^{w_i} h^s, c_3 = w^{w_i} g^{r+s})$ with $r, s \leftarrow \mathbb{Z}_p$ chosen independently for each wire.
3. For the output wire, create a special commitment $c^* = (u, v, w)$ that can easily be checked to be a commitment to 1, as required.
4. For each commitment $c = (c_1, c_2, c_3)$ to each wire value w_i, generate a commitment $c' = (c_1/u, c_2/v, c_3/w)$, and give a WI proof that either c or c' is a linear tuple with respect to f, h, g. Note that if $w_i = 0$, then c is a linear tuple, and if $w_i = 1$, then c' is a linear tuple.
5. For all NAND-gates, do the following. We write the input commitments tuples as $a = (a_1, a_2, a_3), b = (b_1, b_2, b_3)$, and the output commitment tuple as $c = (c_1, c_2, c_3)$. From these commitments, create two new tuples: $C = (C_1 = a_1 b_1 c_1^2 u^{-2}, C_2 = a_2 b_2 c_2^2 v^{-2}, C_3 = a_3 b_3 c_3^2 w^{-2})$ and $C' = (C_1/u, C_2/v, C_3/w)$. Note that either C or C' is a linear tuple iff the values underlying the commitments a, b, c respect the NAND gate. Then give a WI proof that either C or C' is a linear tuple, noting that the witness for this can be derived from the wire values and randomness used to prepare the commitments a, b, and c.
6. Return π consisting of all the commitments and WI proofs.

Verification: The verifier is given a circuit C and a proof π.

1. Check that all wires have a corresponding commitment tuple and that the output wire's commitment tuple is (u, v, w).
2. Check that all WI proofs showing that each wire has a committed value in $\{0, 1\}$ are valid.
3. Check that all WI proofs corresponding to NAND-gates are valid.
4. Return 1 if all checks pass, else return 0.

Remark. We note that in the common reference string, if p is a prime number, then if we let g, f, h, u, v, w be randomly chosen elements of $\mathbb{G}$, with overwhelming probability they will form a viable CRS such that (u, v, w) are a non-linear tuple with respect to (f, h, g), and therefore the resulting commitment scheme is perfectly binding. If, for instance, the group is the one suggested by Boneh and Franklin [BF03], then all that is needed to define $\mathbb{G}$ is the prime p. Thus, we can implement our NIZK Proofs in the Common *Random* String model, where the random string is first used to obtain a k-bit prime p using standard methods (just dividing up the CRS into k-bit chunks and checking one-by-one if they are prime will do), and then the remaining randomness is used to randomly determine g, f, h, u, v, w (by picking random order p points on the curve). Such an NIZK Proof will not have perfect soundness, but statistical soundness, since the probability of (u, v, w) being a linear tuple is exponentially small in k. In the common random string model this is optimal, since for any NIZK proof system with a common random string there is a risk of accidentally selecting a simulation string.

Theorem 2. *The protocol above is an NIZK proof system for Circuit Satisfiability with perfect completeness, perfect soundness and computational zero-knowledge if the Decisional Linear Assumption holds for the bilinear group generator $\mathcal{G}$.*

Proof sketch.

Perfect completeness and soundness are clear. We now argue that our NIZK proof system is computational zero knowledge. We present this in two stages.

We first examine a hybrid in which the prover uses the witness to generate a proof, but where the CRS is simulated so that (u, v, w) form a linear tuple, instead of a non-linear tuple. We note that (by means of intermediate hybrid in which (u, v, w) are random, and a reduction to the Decisional Linear Assumption) this hybrid produces computationally indistinguishable proofs.

The simulator will now produce proofs that are *distributed identically* to the hybrid above (assuming that the underlying WI proofs are perfectly WI). It starts by choosing $u = f^{r_0}, v = h^{s_0}, w = g^{r_0+s_0}$, and remembering these values $r_0, s_0 \leftarrow \mathbb{Z}_p$.

Now, for each wire w, the simulator picks a commitment $c = (c_1 = f^r, c_2 = h^s, c_3 = g^{r+s})$ as a random linear tuple. Because (u, v, w) is a linear tuple, all commitment strings are distributed identically as random linear tuples.

For generating the WI-proofs corresponding to the commitments for wires, since the simulator directly has a witness for showing that the commitment is a linear tuple, it uses this to complete the proof.

For generating the WI-proofs corresponding to NAND gates, we note that for each NAND gate, if the 3 wire commitments are $a = (f^{r_1}, h^{s_1}, g^{r_1+s_1})$, $b = (f^{r_2}, h^{s_2}, g^{r_2+s_2})$, and $c = (f^{r_3}, h^{s_3}, g^{r_3+s_3})$, then the commitment $C = (f^{r_1+r_2+2r_3-2r_0}, h^{s_1+s_2+2s_3-2s_0}, g^{(r_1+r_2+2r_3-2r_0)+(s_1+s_2+2s_3-2s_0)})$, and therefore we have a witness to C being a linear tuple, and we can use this to complete the WI proof.

The only difference between how the simulator proceeds and how the honest prover algorithm proceeds is the choice of which witnesses to use in the WI proof. Therefore, the (perfect) indistinguishability of the simulation from the hybrid follows from the (perfect) witness-indistinguishability of the WI proofs. $\square$

6 Non-Interactive Zaps for Circuit Satisfiability

We now give our construction of non-interactive zaps for Circuit Satisfiability, following the intuition presented in the Introduction.

Statement: A circuit C.

Proof: The prover given $1^k, C$ and input values w such that $C(w) = 1$ proceeds as follows:

1. Generate a *verifiable* bilinear group $(p, \mathbb{G}, \mathbb{G}_T, e, g) \leftarrow \mathcal{G}(1^k)$.
2. Choose a *perfectly hiding* CRS, namely generators f, h, and a linear tuple (u, v, w).

3. Use the NIZK prover to obtain a proof π_1 of the statement with respect to the CRS $(p, \mathbb{G}, \mathbb{G}_1, e, g, f, h, u, v, w)$.
4. Use the NIZK prover to obtain a proof π_2 of the statement with respect to the CRS $(p, \mathbb{G}, \mathbb{G}_1, e, g, f, h, u, v, wg)$. Observe, we are using $w' = wg$.
5. The resulting proof is $\pi = (p, \mathbb{G}, \mathbb{G}_T, e, g, f, h, u, v, w, \pi_1, \pi_2)$.

Verification: On input C and a proof π as described above, accept iff the following procedure succeeds:

1. Use the verification algorithm to check that $(p, \mathbb{G}, \mathbb{G}_T, e, g)$ is a bilinear group.
2. Verify that $f \neq 1, h \neq 1$, i.e., that f and h are generators of $\mathbb{G}$.
3. Verify π_1 with respect to the CRS $(p, \mathbb{G}, \mathbb{G}_1, g, f, h, u, v, w)$.
4. Verify π_2 with respect to the CRS $(p, \mathbb{G}, \mathbb{G}_T, g, f, h, u, v, wg)$.

Theorem 3. *The protocol described above is a non-interactive proof for Circuit Satisfiability with perfect completeness, perfect soundness and computational witness indistinguishability if the Decisional Linear Assumption holds for the verifiable bilinear group generator $\mathcal{G}$.*

Proof.

Perfect completeness: The protocol is perfectly complete because the NIZK proofs for Circuit Satisfiability are perfectly complete.

Perfect soundness: Perfect soundness follows from the fact that at least one of the two CRS's – $(p, \mathbb{G}, \mathbb{G}_T, e, g, f, h, u, v, w)$ and $(p, \mathbb{G}, \mathbb{G}_T, g, f, h, u, v, wg)$ 1– must have perfectly binding parameters for the commitment scheme. Perfect soundness of the corresponding NIZK proof implies that C must be satisfiable.

Computational witness indistinguishability: We now argue (computational) witness indistinguishability assuming the Decisional Linear Assumption, by means of a hybrid argument:

1. The first hybrid is simply the prover algorithm above using witness w_1. That is, it chooses a group $(p, \mathbb{G}, \mathbb{G}_T, e, g)$ and a public key (f, h) and a random linear tuple (u, v, w), then uses the NIZK prover with witness w_1 to obtain π_1, and uses the NIZK prover with witness w_1 to obtain π_2.
2. The second hybrid proceeds as in the first, except that for π_1, it uses the NIZK prover with witness w_2 to obtain π_1 instead of using witness w_1. Hybrid 1 and Hybrid 2 are identically distributed, by means of an intermediate hybrid using the NIZK simulator for π_1, and the fact that the NIZK simulator is a perfect simulator in the case where the CRS is based on a linear tuple.
3. The third hybrid proceeds as the second, except that it chooses random generators (f, h, g), and a linear tuple (u, v, w'), and sets $w = w'/g$. Note that now, (u, v, w) is a perfectly binding CRS, while (u, v, w') is a perfectly hiding CRS.
   Hybrid 2 and Hybrid 3 are computationally indistinguishable by a reduction to the Decisional Linear Assumption. This is seen by means of an intermediate hybrid in which (u, v, w) are set to a random tuple.

4. The fourth hybrid proceeds as the third, except that for π_2, it uses the NIZK prover with witness w_2 to obtain π_2 instead of using witness w_1. Hybrid 3 and Hybrid 4 are identically distributed for the same reasons Hybrids 1 and 2 were identically distributed.
5. Finally, the fifth hybrid proceeds as the fourth, except that it chooses random generators (f, h, g), and a linear tuple (u, v, w), and sets $w' = wg$. This is precisely the WI prover algorithm using witness w_2.
   Hybrid 4 and Hybrid 5 are computationally indistinguishable by a reduction to the Decisional Linear Assumption, by the same argument showing that Hybrids 2 and 3 were computationally indistinguishable.

□

7 Acknowledgment

We wish to thank Brent Waters for important collaboration at an early stage of this research, in particular for his collaboration on ideas for building NIZK proof systems based on the Decisional Linear Assumption.

References

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *proceedings of CRYPTO '04, LNCS series, volume 3152*, pages 41–55, 2004.

[BF03] Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *proceedings of STOC '88*, pages 103–112, 1988.

[BOV03] Boaz Barak, Shien Jin Ong, and Salil P. Vadhan. Derandomization in cryptography. In *proceedings of CRYPTO '03, LNCS series, volume 2729*, pages 299–315, 2003.

[DN00] Cynthia Dwork and Moni Naor. Zaps and their applications. In *proceedings of FOCS '00*, pages 283–293, 2000.

[FLS99] 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. Earlier version entitled Multiple Non-Interactive Zero Knowledge Proofs Based on a Single Random String appeared at FOCS '90.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *proceedings of STOC '90*, pages 416–426, 1990.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proofs. *SIAM Journal of Computing*, 18(1):186–208, 1989.

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero-knowledge for np. In *proceedings of EUROCRYPT '06, LNCS series, volume 4004*, pages 339–358, 2006.

[Gro06] Jens Groth. Simulation-sound non-interactive zero-knowledge proofs for a practical language and constant size group signatures. Manuscript, 2006.

[Nao03] Moni Naor. On cryptographic assumptions and challenges. In *proceedings of CRYPTO '03, LNCS series, volume 2729*, pages 96–109, 2003.