Non-interactive zaps of knowledge

Georg Fuchsbauer[1,2] and Michele Orrù[2,1]

[1] Inria
[2] École normale supérieure, CNRS, PSL Research University, Paris, France

first.last@ens.fr

Abstract. While non-interactive zero-knowledge (NIZK) proofs require trusted parameters, Groth, Ostrovsky and Sahai constructed non-interactive witness-indistinguishable (NIWI) proofs without any setup; they called their scheme a non-interactive zap. More recently, Bellare, Fuchsbauer and Scafuro investigated the security of NIZK in the face of parameter subversion and observe that NI zaps provide subversion-resistant soundness and WI.
Arguments of knowledge prove that not only the statement is true, but also that the prover knows a witness for it, which is essential for anonymous identification. We present the first NIWI argument of knowledge *without* parameters, i.e., a NI zap of knowledge. Consequently, our scheme is also the first subversion-resistant knowledge-sound proof system, a notion recently proposed by Fuchsbauer.

Keywords: Non-interactive proofs, argument of knowledge, subversion resistance.

1 Introduction

The concept of zero-knowledge proof systems, first proposed by Goldwasser, Micali and Rackoff [GMR89], is a central tool in modern cryptography. Consider an NP relation R which defines the language of all statements x for which there exists a witness w so that $R(x, w) = \mathbf{true}$. In a zero-knowledge proof for R a prover, knowing a witness, wants to convince a verifier that x is in the language. The protocol must be *complete*, that is, if the prover knows a witness for x then it can convince the verifier; it should be *sound*, in that no malicious prover can convince the verifier of a false statement, and *zero-knowledge*: the execution of the protocol reveals no information to the verifier (beyond the fact that x is in the language).

Feige and Shamir [FS90] proposed a relaxation of zero-knowledge called *witness indistinguishability*, which only requires that it is indistinguishable which witness was used to compute a proof. This notion turns to be sufficient in many contexts. *Non-interactive* zero-knowledge proofs (NIZK) [BFM88] allow the prover to convince the verifier by only sending a single message. However, they rely on the existence of a common-reference string (CRS) to which prover and verifier have access. The CRS is assumed to have been set up by some trusted party, which represents a serious limitation for all applications of NIZK in scenarios where parties mutually distrust each other.

Dwork and Naor [DN00] constructed a two-round witness-indistinguishable proof system for NP in the plain model, that is, where no trusted CRS is assumed. In their protocol the first message (sent from the verifier to the prover) can be fixed once and for all, and the second one provides the actual proof. They called such protocols *zaps*. Groth, Ostrovsky and Sahai [GOS06a] showed the existence of *non-interactive* zaps, where the prover sends a single message to deliver the proof. Non-interactive zaps are thus non-interactive proof systems *without* a CRS. Since in this scenario it is impossible to achieve zero-knowledge [GO94], witness indistinguishability (WI) is the best one can hope for. Following [GOS06a], there have been many works extending this line of research [BW06, BW07, Gro06].

All aforementioned schemes guarantee that proofs can only be computed for valid statements. Arguments of knowledge are proof systems that satisfy a stronger notion of soundness. They require the prover to *know* a witness for the proved statement. This is formalized via the notion of knowledge soundness that demands that for each prover there exists an efficient extractor which can extract a witness from the prover whenever it makes a valid proof. (When this holds for computationally bounded provers, we speak of *arguments* rather than proofs.) Since, by definition, false statements have no witnesses, knowledge soundness implies the standard notion of (computational) soundness.

Succinct non-interactive arguments of knowledge (SNARKs) are non-interactive proof systems with short (that is, independent of the size of the statement or the witness) efficiently verifiable proofs that satisfy knowledge soundness. SNARKs were initially introduced for verifiable computation and are now the most widely deployed proof systems in the real world. They are used in cryptocurrencies such as Zcash [BCG+14], which guarantees anonymity via zero-knowledge SNARKs. As for all NIZK systems, a drawback of SNARKs is that they require a CRS, that is, they require a one-time trusted setup of public parameters. Since for SNARKs every CRS has a simulation trapdoor, subversion of these parameters leads to full compromise of soundness.

Subversion resistance. Motivated by the subversion of trusted public parameters in standardized cryptographic protocols led by mass-surveillance activities, Bellare, Fuchsbauer and Scafuro [BFS16] investigate what security properties can be maintained for NIZK when its trusted parameters are subverted. CRS's for NIZK are especially easy to subvert, since they must be subvertible by design: zero knowledge requires that an honest CRS must be indistinguishable from a backdoored one, where the backdoor is the trapdoor used to simulate proofs.

Bellare et al. defined multiple security properties that protect against parameter subversion: subversion soundness (S-SND) means that no adversary can generate a malicious CRS together with a valid proof for a false statement; subversion zero knowledge (S-ZK) requires that even if the adversary sets up the CRS, there exists a simulator able to produce its full view; and subversion witness indistinguishability (S-WI) formalizes that even for proofs that were made under a subverted CRS, it is still infeasible to tell which of two witnesses was used.

Following Goldreich and Oren [GO94], Bellare et al. [BFS16] also showed that it is impossible to achieve subversion soundness and (standard) zero-knowledge simultaneously. For subversion-sound proof systems, subversion witness indistinguishability is thus the best one can hope for. The authors [BFS16] observe that since proof systems that do not rely on a CRS cannot succumb to CRS-subversion attacks, non-interactive zaps [GOS06a] achieve both S-SND and S-WI.

Bellare et al. did not consider the stronger notion of knowledge soundness, which is the notion achieved by SNARKs, and which in many applications is the required notion for the used proof systems. For example, for all kinds of anonymous authentication, users prove knowledge of signatures (often called *certificates* or *credentials*, depending on the context); in this case soundness is not sufficient, as signatures always exist, but in the security proof they must actually be extracted in order to rely on their unforgeability. Fuchsbauer [Fuc18] has recently defined a subversion-resistant notion of knowledge soundness but left it as an open problem to give a scheme that achieves it. Such a scheme would protect against possible parameter subversion in any context where proving knowledge of a witness is required.

Our contribution. Our result can be summarized as follows:

(i) We provide the first non-interactive zap with knowledge soundness; that is, a witness-indistinguishable proof system without parameters for which there exists an extractor that recovers a witness from every valid proof.

(ii) Our construction is also the first fully subversion-resistant WI argument-of-knowledge system. In particular, it satisfies the recently defined notion of subversion knowledge soundness [Fuc18], as well as subversion witness indistinguishability [BFS16] (the strongest notion compatible with S-SND).

Bellare et al. [BFS16] introduce a new type of knowledge-of-exponent assumption, which they call DH-KE. They prove (standard) soundness and subversion zero knowledge of their main construction under DH-KE and the decision linear assumption (DLin) [BBS04].

Our construction builds on the original non-interactive zap [GOS06a], whose soundness we upgrade to knowledge soundness. As for zaps, the language of our proof system is circuit satisfiability and thus universal. Groth, Ostrovsky and Sahai's [GOS06a] starting point is a "dual-mode" [GOS06b, PVW08] non-interactive proof system, for which there are two indistinguishable types of CRS: one leading to proofs that are perfectly sound and the other leading to proofs that are perfectly WI. To construct a non-interactive zap, they let the prover choose the CRS. As the prover could choose a CRS that leads to "unsound" proofs, the prover must actually choose two CRS's that are related in a way that guarantees that at least one of them is of the "sound" type. It must then provide a proof of the statement under both of them. The authors [GOS06a] then show that this protocol still achieves computational WI.

We turn their construction into a proof of knowledge by again doubling the proof, thereby forcing the prover to prove knowledge of a trapdoor which allows to extract the witness from one of the sound proofs. We prove our non-interactive zap of knowledge secure under the same assumptions as Bellare et al.'s S-ZK+SND scheme. Our result is summarized in the following theorem.

Theorem 1. *Assuming DLin and DH-KE, there exists a non-interactive zap for circuit satisfiability that satisfies knowledge soundness. The proof size is $O(\lambda k)$, where λ is the security parameter and k is the size of the circuit.*

Let us finally note that our system also implies a proof system which achieves (standard) knowledge soundness, (standard) zero knowledge and *subversion* witness indistinguishability. This is obtained by plugging our zap of knowledge into the construction by Bellare et al. [BFS16] that achieves SND, ZK and S-WI.

Their scheme uses a length-doubling pseudorandom generator (PRG) and a CRS contains a random bit string σ of length 2λ (where λ is the security parameter). A proof for statement x is a zap for the following statement: either x is a valid statement or σ is in the range of the PRG. Using a zap of knowledge (ZaK), knowledge soundness follows from knowledge soundness of the ZaK since with overwhelming probability σ is *not* in the range of the PRG. (The extractor must thus extract a witness for x.) Zero knowledge follows from WI of the zap, as after replacing σ with an image under the PRG, proofs can be simulated using the preimage. Finally, S-WI follows from S-WI of the zap.

Related work. Since the introduction of non-interactive zaps in [GOS06a], a number of papers have studied and provided different (and more efficient) implementations of zaps. Groth and Sahai [GS08] provided a more general framework for NIWI and NIZK proofs, which leads to more efficient proofs for concrete languages (instead of circuit satisfiability). Furthermore, their proof system can also be based on other assumptions apart from DLin, such as SXDH, allowing for shorter proofs.

Bitanski and Paneth [BP15] presented a different approach to constructing zaps and WI proofs based on indistinguishability obfuscation (iO), but constructions using iO are only of theoretical interest. Ràfols [Ràf15] showed how to base non-interactive zaps on Groth-Sahai

Table 1. Efficiency and security of the original zaps and our constructions of zaps of knowledge, where w is the number of wires, g the number of gates and $|\mathbb{G}|$ is the size of an element of a group $\mathbb{G}$.

Protocol	Efficiency	Assumptions
Zap [GOS06a]	$(18w + 12g + 5)\,|\mathbb{G}|$	DLin
Zap of knowledge, Section 5	$(36w + 24g + 14)\,|\mathbb{G}|$	DLin, DH-KE
Zap (of knowledge; Appendix B)	$(12w + 8g + 3)\,(|\mathbb{G}_1|+|\mathbb{G}_2|)$	SXDH (ADH-KE)

proofs, which achieves an improvement in efficiency (by a constant factor) over the original construction [GOS06a]. Her construction can be implemented in asymmetric ("Type-1") pairing groups.

Her scheme can also serve as the starting point for a scheme achieving knowledge soundness and we explore this in Appendix B. (See Table 1 for an overview of efficiency.) Although this scheme is more efficient, we decided to concentrate on building a scheme from [GOS06a], as we can prove it secure under the same assumptions that underly Bellare et al.'s [BFS16] SND+S-ZK scheme; in contrast, a scheme based on an asymmetric bilinear group would require to make the DH-KE assumption in such groups (we refer to it as ADH-KE in Appendix B). This is a qualitatively different assumption, as without a symmetric pairing it cannot be checked whether the triple returned by the adversary is of the right form (see Fig. 3); it would thus not be a falsifiable assumption, as it is not efficiently decidable whether an adversary has broken the assumption. Finally, our main scheme achieves *tight* security, whereas our proof of knowledge soundness in Appendix B has a security loss that is linear in the circuit size.

2 Preliminaries

Notation. Let λ be the security parameter. We let $\mathsf{M.rl}(\lambda)$ be a *length function* (i.e. a function $\mathbb{N} \to \mathbb{N}$ polynomially bounded) in λ defining the length of the randomness for a probabilistic machine M. When sampling the value a uniformly at random from the set S, we write $a \leftarrow_{\$} S$. When sampling the value a from the probabilistic algorithm M, we write $a \leftarrow \mathsf{M}$. We use $:=$ to denote assignment. Elements of $\mathbb{Z}_p$ are denoted in lower case, group elements are denoted with capital letters. Vectors are denoted in bold. We employ additive notation for groups. For a vector $\vec{a}$ we use the subscript notation a_i to indicate the i-th component of a. Let R be a relation between statements denoted by ϕ and witnesses denoted by w. By $\mathsf{R}(\phi)$ we denote the set of possible witnesses for the statement ϕ in R. We let $\mathcal{L}(\mathsf{R}) := \{\phi : \mathsf{R}(\phi) \neq \emptyset\}$ be the *language* associated to R. A proof that $w \in \mathsf{R}(\phi)$ is denoted with π.

In this work, we consider the language of circuit satisfiability, which is NP-complete. For a binary circuit $\mathtt{C}$, which we always assume of size $O(\lambda)$, the set $\mathsf{R}(\mathtt{C})$ is the set of inputs w that satisfy $\mathtt{C}(w) = 1$. Without loss of generality, we assume that circuits consist solely of NAND gates. Unless otherwise specified, all following algorithms are assumed to be randomized and to run in time $\mathsf{poly}(\lambda)$. As Bellare et al. [BFS16], who follow [Gol93], we only consider uniform machines to model the adversary A and the extractor Ext. (See [BFS16, Fuc18] for discussions on how this choice affects the hardness assumptions and security guarantees.)

Bilinear groups. Throughout this work, we make use of prime-order abelian groups equipped with a (symmetric) bilinear map. Concretely, we assume the existence of groups $\mathbb{G}, \mathbb{G}_T$ of odd prime order p of length λ and an efficiently computable non-degenerate bilinear map $e \colon \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. That is, the map e is such that $\forall\, U, V \in \mathbb{G}$ and $a, b \in \mathbb{Z}_p : e(aU, bV) = ab \cdot e(U, V)$, and if U is a generator of $\mathbb{G}$, then $e(U, U)$ is a generator of $\mathbb{G}_T$. We say that a bilinear group is *verifiable* if there exists an efficient verification algorithm that outputs **true** if and only if $\Gamma = (p, \mathbb{G}, \mathbb{G}_T, e)$

is the description of a bilinear group. For instance, the elliptic-curve group of [BBS04] equipped with the Weil pairing is publicly verifiable. In most practical scenarios, the group description is embedded as a part of the protocol specification and agreed upon in advance; in these cases there is no need for verification.

Throughout this paper, we assume the existence of a deterministic algorithm G that, given as input the security parameter in unary 1^λ, outputs a bilinear group description Γ. The same assumption was already employed by Bellare et al. [BFS16]. The main advantage in choosing G to be deterministic is that every entity in the scheme can (re)compute the group from the security parameter, and no party must be trusted with generating the group. Moreover, real-world pairing schemes are defined for groups that are fixed for some λ. For the sake of simplicity, we define all our schemes w.r.t. a group description Γ and assume that the security parameter ($\lambda \in \mathbb{N}$ such that $\Gamma := \mathsf{G}(1^\lambda)$) can be derived from Γ.

Extractable commitment schemes. A commitment scheme Com consists of the following three algorithms:

- $(\sigma, \tau) \leftarrow \mathsf{Com.K}(\Gamma)$, the key generation algorithm, outputs a CRS σ together with the trapdoor information τ.
- $(C, r) \leftarrow \mathsf{Com.C}(\sigma, v)$, the commitment algorithm, outputs a commitment C to the given value v together with the *opening information* r.
- $\mathit{bool} \leftarrow \mathsf{Com.O}(\sigma, C, v, r)$, the opening algorithm, outputs **true** if C is a commitment to v witnessed by r, and **false** otherwise.

In our case, $\mathsf{Com.C}$ returns the used randomness and $\mathsf{Com.O}$ simply recomputes the commitment and checks that $C = \mathsf{Com.C}(V; r)$. Consequently, *correctness* of the scheme is trivial. To ease notation for commitments and openings, we will always assume that the group description Γ can be deduced from σ, and omit the opening information from the returned value.

Generally, we require commitment schemes to be *hiding* and *binding*. Loosely speaking, a scheme is *hiding* if the commitment C reveals no information about v. A scheme is *binding* if a cheating committer cannot change its mind about the value it committed to. Formally, it is hard to find C, v, r, v' and r' such that $\mathsf{Com.O}(\sigma, C, v, r) = \textbf{true} = \mathsf{Com.O}(\sigma, C, v', r')$.

Throughout this work, we also require a perfectly binding commitment scheme to be *extractable*, that is, Com is equipped with an efficient extraction algorithm $\mathsf{Com.E}$ that, given as input the trapdoor information τ, recovers the value v to which C is bound.

Proof systems. A non-interactive proof system Π for a relation R consists of the following three algorithms:

- $(\sigma, \tau) \leftarrow \Pi.\mathsf{K}(\Gamma)$, the CRS generation algorithm that outputs a CRS σ (and possibly some trapdoor information τ). Since we are dealing with *publicly verifiable protocols*, the trapdoor information τ will be omitted in most cases and used solely for clarity in the proofs or when combining protocols.
- $\pi \leftarrow \Pi.\mathsf{P}(\sigma, \phi, w)$, a prover which takes as input some $(\phi, w) \in \mathsf{R}$ and a CRS σ, and outputs a proof π.
- $\mathit{bool} \leftarrow \Pi.\mathsf{V}(\sigma, \phi, \pi)$ a verifier that, given as input a statement ϕ together with a proof π outputs **true** or **false**, indicating acceptance of the proof.

To ease notation for prover and verifier, we will assume that the group description Γ can be inferred from the CRS σ. A proof is complete if every correctly generated proof verifies. If the CRS is clear from the context, we omit σ from the arguments of $\Pi.\mathsf{P}$ or $\Pi.\mathsf{V}$.

```
Game $\mathrm{WI}_{\Pi,\mathsf{R},\mathsf{A}}(\lambda)$
$b \leftarrow_{\$} \{0,1\}\,;\ \Gamma \coloneqq \mathsf{G}(1^\lambda)$
$(\sigma,\tau) \leftarrow \Pi.\mathsf{K}(\Gamma)$
$b' \leftarrow \mathsf{A}^{\textsc{Prove}}(\sigma)$
return $(b = b')$
```

```
Oracle $\textsc{Prove}(\phi, w_0, w_1)$
if $\mathsf{R}(\phi,w_0) = \mathbf{false}\ \vee\ \mathsf{R}(\phi,w_1) = \mathbf{false}$
    return $\perp$
$\pi \leftarrow \Pi.\mathsf{P}(\sigma,\phi,w_b)$
return $\pi$
```

Fig. 1. Witness indistinguishability (WI) game.

Zaps. A zap is a two-round, *witness-indistinguishable* proof system where the first-round message is fixed “once and for all” [DN00] for all future instances of the protocol. The notion of *witness-indistinguishability* [FLS90] informally states that no PPT adversary can tell which of any two possible witnesses has been used to construct a proof.

Definition 2. *A proof system* Π *is witness-indistinguishable (WI) for the relation* R *if, for any PPT adversary* A, $\mathsf{Adv}^{\mathrm{wi}}_{\Pi,\mathsf{R},\mathsf{A}}(\lambda)$ *is negligible in* λ, *where*

$$\mathsf{Adv}^{\mathrm{wi}}_{\Pi,\mathsf{R},\mathsf{A}}(\lambda) \coloneqq \Pr\left[\mathrm{WI}_{\Pi,\mathsf{R},\mathsf{A}}(\lambda)\right] - 1/2 = \mathsf{negl}(\lambda)$$

and $\mathrm{WI}_{\Pi,\mathsf{R},\mathsf{A}}(\lambda)$ *is the game depicted in Fig. 1.*

A zap is *non-interactive* if there is no first-round message from the verifier to the prover: the prover simply sends a single message. The proof system thus reduces to a pair (P,V) or can be considered as defined above, but with a CRS generation algorithm that always outputs $\perp$. We next define the soundness notion for *non-interactive arguments of knowledge.*

Knowledge soundness [BG93] means that for any prover able to produce a valid proof, there exists an efficient algorithm which has access to the prover’s random coins capable of extracting a witness for the given statement.

Definition 3. *A proof system* Π *is* knowledge-sound *for* R *if for any PPT adversary* A *there exists an extractor* Ext *such that* $\mathsf{Adv}^{\mathrm{ksnd}}_{\mathsf{A},\mathsf{Ext},\mathsf{R},\Pi}(\lambda)$ *is negligible in* λ, *where:*

$$\mathsf{Adv}^{\mathrm{ksnd}}_{\Pi,\mathsf{R},\mathsf{A},\mathsf{Ext}}(\lambda) \coloneqq \Pr\left[\mathrm{KSND}_{\Pi,\mathsf{R},\mathsf{A},\mathsf{Ext}}(\lambda)\right] = \mathsf{negl}(\lambda)\,.$$

and $\mathrm{KSND}_{\mathsf{A},\mathsf{Ext},\mathsf{R},\Pi}(\lambda)$ *is defined in Fig. 2. An* argument of knowledge *is a* knowledge-sound *proof.*

Variations of this argument are often found in the literature. Most of them allow the extractor to rewind the adversary for interactive proof systems in addition to black-box access, most notably for Σ-protocols. In case of non-interactive provers the extractor is provided with the adversary’s random coins.

```
Game $\mathrm{KSND}_{\Pi,\mathsf{R},\mathsf{A},\mathsf{Ext}}(\lambda)$
$\Gamma \coloneqq \mathsf{G}(1^\lambda); (\sigma,\tau) \leftarrow \Pi.\mathsf{K}(\Gamma)$
$r \leftarrow_{\$} \{0,1\}^{\mathsf{A}.\mathsf{rl}(\lambda)}; (\phi,\pi) \coloneqq \mathsf{A}(\sigma; r)$
$w \leftarrow \mathsf{Ext}(\sigma, r)$
return $(\Pi.\mathsf{V}(\sigma,\phi,\pi)$ and $\mathsf{R}(\phi,w) = \mathbf{false})$
```

Fig. 2. Game for knowledge soundness.

```
Game $\mathrm{DLin}_{\mathsf{G},\mathsf{A}}(\lambda)$
$b \leftarrow\!\$\, \{0,1\}$ ; $\Gamma := (p, \mathbb{G}, \mathbb{G}_T, e, G) := \mathsf{G}(1^\lambda)$
$u, v, r, s \leftarrow\!\$\, \mathbb{Z}_p$
if $b = 1$ then $H := (r+s)G$
else  $H \leftarrow\!\$\, \mathbb{G}$
$b' \leftarrow \mathsf{A}(\Gamma, uG, vG, urG, vsG, H)$
return $(b = b')$
```

```
Game $\mathrm{DH\text{-}KE}_{\mathsf{G},\mathsf{A},\mathsf{Ext}}(\lambda)$
$\Gamma := (p, \mathbb{G}, \mathbb{G}_T, e, G) := \mathsf{G}(1^\lambda)$
$r \leftarrow\!\$\, \{0,1\}^{\mathsf{A}.\mathsf{rl}(\lambda)}$
$(X, Y, Z) := \mathsf{A}(\Gamma; r)$
$s \leftarrow \mathsf{Ext}(\Gamma, r)$
if $sG = X \lor sG = Y$ then return $0$
return $(e(X,Y) = e(Z,G))$
```

Fig. 3. Games for Assumptions 1 (DLin) and 2 (DH-KE).

Assumptions. Our protocol is based on the DH-KE assumption and the existence of a homomorphic extractable commitment scheme. Such schemes have been widely studied and there are constructions from standard assumptions such as the subgroup decision assumption or the decisional linear (DLin) assumption [BBS04]. For this work, we rely on the latter, which is also used in [GOS06a].

The DLin assumption [BBS04] for an abelian group $\mathbb{G} = \langle G \rangle$ of order p states that it is computationally difficult to distinguish $(uG, vG, urG, vsG, (r+s)G)$ with $u, v, r, s \leftarrow\!\$\, \mathbb{Z}_p$ from a uniformly random 5-tuple in $\mathbb{G}$.

Assumption 1 (DLin) *We say that the Decisional Linear Diffie-Hellman assumption holds for the group generator* G *if for all PPT adversaries* A *it holds:*

$$\mathsf{Adv}^{\mathrm{dlin}}_{\mathsf{G},\mathsf{A}}(\lambda) := \Pr\left[\mathrm{DLin}_{\mathsf{G},\mathsf{A}}(\lambda)\right] - 1/2 = \mathsf{negl}(\lambda)$$

where the game $\mathrm{DLin}_{\mathsf{G},\mathsf{A}}(\lambda)$ *is defined in Fig. 3.*

The intuition behind DH-KE [BFS16] is that it is difficult for some machine to produce a DH triple (xG, yG, xyG) in $\mathbb{G}$ without knowing at least x or y. The assumption is in the same spirit of earlier knowledge-of-exponent assumptions [Gro10, BCI+10], whose simplest form states that given $(G, xG) \in \mathbb{G}^2$ it is hard to return (yG, xyG) without knowing y.

Assumption 2 (DH-KE) *The Diffie-Hellman Knowledge of Exponent assumption holds for the bilinear group generator* G *if for any PPT adversary* A *there exists a PPT extractor* Ext *such that:*

$$\mathsf{Adv}^{\mathrm{dhke}}_{\mathsf{G},\mathsf{A},\mathsf{Ext}}(\lambda) := \Pr\left[\mathrm{DH\text{-}KE}_{\mathsf{G},\mathsf{A},\mathsf{Ext}}(\lambda)\right] = \mathsf{negl}(\lambda)$$

where the game $\mathrm{DH\text{-}KE}_{\mathsf{G},\mathsf{A},\mathsf{Ext}}(\lambda)$ *is defined in Fig. 3.*

In other variants of knowledge of exponent assumptions the adversary is provided with some auxiliary information, which amounts to a stronger assumption. This is typically required as in the security proofs the reduction obtains a challenge which it needs to embed somewhere. In our specific case, all the proof material is generated by the prover itself, including the CRS. Consequently, the game DH-KE presents an adversary that simply takes as input a group description, without any auxiliary information. Compared to [BFS16], where the adversary is provided with additional information, our variant is thus weaker.

3 An extractable commitment scheme from DLin

We recall the homomorphic commitment scheme based on linear encryption [BBS04] by Groth Ostrovsky and Sahai [GOS06a]. It defines two types of key generation: a perfectly hiding and perfectly binding one. Given a bilinear group $\mathit{\Gamma} := (p, \mathbb{G}, \mathbb{G}_T, e, G)$, it defines two key generation algorithms $\mathsf{Com.K}^{(\mathrm{b})}$ and $\mathsf{Com.K}^{(\mathrm{h})}$ producing binding and hiding keys, respectively:

```
Com.K^(h)
  $\tau := (r_u, s_v) \leftarrow\!\!\$\, (\mathbb{Z}_p^*)^2;\ (x, y) \leftarrow\!\!\$\, (\mathbb{Z}_p^*)^2$
  $F := xG,\quad H := yG$
  $(U, V, W) := (r_u F, s_v H, (r_u + s_v)G)$
  $\sigma := (F, H, U, V, W)$
  return $(\sigma, \tau)$
```

```
Com.K^(b)
  $\tau := (x, y, z) \leftarrow\!\!\$\, (\mathbb{Z}_p^*)^3;\ (r_u, s_v) \leftarrow\!\!\$\, (\mathbb{Z}_p^*)^2$
  $F := xG,\quad H := yG$
  $(U, V, W) := (r_u F, s_v H, (r_u + s_v + z)G)$
  $\sigma := (F, H, U, V, W)$
  return $(\sigma, \tau)$
```

In order to commit to a value $m \in \mathbb{Z}_p$, one samples $r, s \leftarrow\!\!\$\, \mathbb{Z}_p$ and returns:

$$C = \mathsf{Com.C}(m; r, s) = \big(mU + rF, mV + sH, mW + (r + s)G\big).$$

Since $\mathsf{Com.C}(m_0; r_0, s_0) + \mathsf{Com.C}(m_1; r_1, s_1) = \mathsf{Com.C}(m_0 + m_1; r_0 + r_1, s_0 + s_1)$, commitments are additively homomorphic. A committed value is opened by providing the randomness (r, s). Under a perfectly hiding key, a commitment to m can be opened to any value m', given trapdoor information $\tau = (r_u, s_v)$:

$$\begin{aligned}\mathsf{Com.C}(m; r, s) &= \big((mr_u + r)F, (ms_v + s)V, (mr_u + r + ms_v + s)G\big)\\ &= \mathsf{Com.C}\big(m'; r - (m' - m)r_u, s - (m' - m)s_v)\big).\end{aligned} \tag{1}$$

Under the DLin assumption, keys output by the perfectly hiding setup are computationally indistinguishable from ones output by the perfectly binding setup. For this reason, a the perfectly hiding setup leads to computationally binding commitments and vice-versa.

We say that a commitment tuple is *linear* w.r.t. (F, H, G) if it is of the form $(rF, sH, (r+s)G)$ for some $r, s \in \mathbb{Z}_p$. Commitments to 0 are linear tuples and all commitment under a *hiding* key is also a linear. Under a *binding* key we have:

$$\mathsf{Com.C}(m; r, s) = \big((mr_u + r)F,\ (ms_v + s)H,\ mzG + (mr_u + r + ms_v + s)G\big).$$

A commitment to m is thus a *linear encryption* [BBS04] of $mzG \in \mathbb{G}_1$ under randomness $(mr_u + r, ms_v + s)$. Given a commitment C and the trapdoor information $\tau = (x, y, z)$, one can *extract* the committed message. We denote by $\mathsf{Com.E}$ the extraction algorithm, which computes:

$$\mathsf{Com.E}\big(\tau,\ (C_0, C_1, C_2)\big) := \mathsf{dLog}\big(z^{-1}(C_2 - x^{-1}C_0 - y^{-1}C_1)\big), \tag{2}$$

where dLog can be efficiently computed if the message space is of logarithmic size; for instance assuming $m \in \{0, 1\}$, we define $\mathsf{Com.E}$ to return 0 if $(C_2 - x^{-1}C_0 - y^{-1}C_1)$ is the identity element, and 1 otherwise.

Theorem 4 ([GOS06a]). *Assuming DLin,* Com*, as defined above, is an extractable homomorphic commitment scheme that is:*

- *perfectly binding, computationally hiding when instantiated with* $\mathsf{Com.K}^{(\mathrm{b})}$*;*
- *computationally binding, perfectly hiding when instantiated with* $\mathsf{Com.K}^{(\mathrm{h})}$*.*

The "parameter switching" technique, which defines different types of keys that are computationally indistinguishable, has proved very useful and also applies to encryption schemes. The idea has been defined (and renamed) several times. What in [GOS06a] is called "parameter switching" is called "meaningful/meaningless encryption" in [KN08], "dual-mode encryption" in [PVW08] and "lossy encryption" in [BHY09].

```
ZAP.P(1^λ, ϕ, w)
    $\varGamma := \mathsf{G}(1^\lambda)$ ; $(\sigma_0, \tau) \leftarrow \mathsf{Circ.K}(\varGamma)$
    $\sigma_1 := \sigma_0 + (0,0,0,0,G)$
    $\pi_0 \leftarrow \mathsf{Circ.P}(\sigma_0, \phi, w)$; $\pi_1 \leftarrow \mathsf{Circ.P}(\sigma_1, \phi, w)$
    return $(\sigma_0, \pi_0, \pi_1)$

ZAP.V(ϕ, (σ_0, π_0, π_1))
    $\sigma_1 := \sigma_0 + (0,0,0,0,G)$
    return $\left(\bigwedge_{i\in\{0,1\}} \mathsf{Circ.V}(\sigma_i, \phi, \varPi_i)\right)$
```

Fig. 4. The (non-interactive) ZAP protocol of [GOS06a].

Proofs of binarity. As a building block for their zaps Groth, Ostrovsky and Sahai [GOS06a] first construct a witness-indistinguishable non-interactive proof system Bin. Given a commitment key $\sigma = (F, H, U, V, W)$ and a commitment $C \in \mathbb{G}^3$, it allows to prove that C commits to a value in $\{0, 1\}$ under σ. We detail their scheme in Appendix A.

4 Non-interactive zaps

To construct a non-interactive zap (i.e., a WI proof system without a CRS), Groth, Ostrovsky and Sahai [GOS06a] first construct a proof system for circuit satisfiability *with* a CRS, based on the commitment scheme from Section 3 and their proof of binarity. Then, in order to make their scheme CRS-less, they define the prover to to pick two CRS's that are correlated in a way that makes it impossible for the adversary to cheat under both of them.

As the commitment scheme described in Section 3 is homomorphic, it is possible to perform linear operations on commitments, and in particular prove logical relations between them.

First, proving that either C or $C' := C - (U, V, W)$ is linear proves that C is a commitment to a bit. In order to prove that committed values satisfy wire assignments of a NAND gate Groth et al. [GOS06b] observe that if $a, b \in \{0, 1\}$ then $c := \neg(a \wedge b)$ iff $t := a + b + 2c - 2 \in \{0, 1\}$. Reasoning with homomorphic commitments, we have that three commitments $A := (A_0, A_1, A_2)$, $B := (B_0, B_1, B_2)$, and $C := (C_0, C_1, C_2)$ are bound respectively to the values a, b, c, such that $c = \neg(a \ \wedge \ b)$, if and only if T is a commitment to either 0 or 1, where:

$$T := A + B + 2 \cdot C - 2 \cdot (U, V, W) \tag{3}$$

Thus, to prove that A, B, C are commitments to values in $\{0, 1\}$ and that C is a commitment to the NAND of the values in A and B, it is sufficient to prove that A, B, C and T are all bit commitments. With these observations, GOS construct a perfectly witness-indistinguishable proof system Circ for circuit satisfiability as follows:

The key generation algorithm Circ.K simply emulates $\mathsf{Com.K}^{(\mathrm{h})}$, that is, it generates a hiding commitment key. The prover $\mathsf{Circ.P}(\sigma, \mathtt{C}, w)$ takes as input a circuit $\mathtt{C}$ and a witness w satisfying $\mathtt{C}(w) = 1$, and does the following: represent the circuit evaluation $\mathtt{C}(w)$ in such a way that w_k is the value running in the k-th wire. For each w_k, produce a commitment $C_k \leftarrow \mathsf{Com.C}(\sigma, w_k)$ to w_k and prove it is to a bit under σ using proof system Bin. For each gate, construct T from the commitments corresponding to the ingoing and outgoing wires as above and prove that it too is a commitment to 0 or 1. For the output commitment, create a commitment C_{out} to 1 that can be easily reproduced and checked by the verifier: $C_{\mathrm{out}} := \mathsf{Com.C}(\sigma, 1; (0, 0))$. Let $\varPi$ be the collection of all other commitments together with the respective proofs of binarity generated. Return $\varPi$.

The verifier $\mathsf{Circ.V}(\sigma, \mathtt{C}, \varPi)$, computes $C_{\mathrm{out}} := \mathsf{Com.C}(\sigma, 1; (0, 0))$ and for every gate the value T as in Eq. (3); using Bin.V, it checks that all the wire commitments are to values in $\{0, 1\}$ and respect the gates (by checking the values T); if all verifications succeed, return **true**. Otherwise, return **false**.

Theorem 5 (**[GOS06a]**)**.** *Assuming DLin,* Circ *is a non-interactive, perfectly sound computationally witness-indistinguishable proof system.*

The reason why we cannot let the prover choose the CRS in Circ is that it could chose it as a perfectly hiding CRS and then simulate proofs. However, if the prover must construct two proofs under two different CRS's which are related in such a way that at least one of them is not linear (and thus binding), then the prover cannot cheat. In particular, note that given a 5-tuple $\sigma_0 \in \mathbb{G}^5$, and defining $\sigma_1 \coloneqq \sigma_0 + (0, 0, 0, 0, G)$ then at *most* one of σ_0, σ_1 is linear. At the same time, both of them are valid CRSs. With this last trick, it is straightforward to construct the zap scheme ZAP, as illustrated in Fig. 4.

Theorem 6 (**[GOS06a]**)**.** *Assuming DLin,* ZAP *is a non-interactive zap with perfect soundness and computational witness indistinguishability.*

Remark 7. We note that soundness of the system ZAP relies only on the fact that Γ is a bilinear group. In [GOS06a] the prover is allowed to generate Γ and it is required that Γ is verifiable. We presented a zap for deterministically generated groups, as considered by Bellare et al. [BFS16], which is also required for our construction of non-interactive zaps of knowledge in the next section.

5 ZAK: a non-interactive zap of knowledge

We present our NIWI argument of knowledge for circuit satisfiability.

The high-level idea of our protocol is to double the ZAP proof of [GOS06a] and link the two CRS's. Whereas ZAP used two Circ proofs to construct a zap from a proof that requires a CRS, we will use two zap proofs not only to prove circuit satisfiability, but to prove *knowledge* of a satisfying assignment. More specifically, knowledge soundness is obtained by generating two independent zap proofs, and then linking the two with 4 group elements in a matrix $\varDelta$. This additional matrix $\varDelta$, that we call *linking element*, guarantees the existence of an extractor that can recover the trapdoor from one of the CRS's contained in the two zap proofs, and use it to extract the witness from the commitments contained in a valid zap proof. Witness indistinguishability of the single proofs follows immediately from [GOS06a], but our proofs also contain the linking element $\varDelta$, which depend on the randomness of the CRS's. We thus need to argue that these addition elements do not harm witness indistinguishability.

Using an extractor to recover the trapdoor hidden in an adversarially generated CRS was also done by Bellare et al. [BFS16] to construct a scheme satisfying subversion-zero knowledge. Our protocol is detailed in Fig. 5, where by DH we denote the algorithm that checks that $\delta_{i,j}$ is the CDH of $(\sigma_{0,0})_i$ and $(\sigma_{1,0})_j$ (see below).

The trapdoor information $\tau_0 = (x_0, y_0)$ and $\tau_1 = (x_1, y_1)$ is correlated in $\varDelta$ to form the following products:

$$\varDelta \coloneqq [\delta_{i,j}]_{i,j\in\{0,1\}} = \begin{bmatrix} x_0x_1G & x_0y_1G \\ y_0x_1G & y_0y_1G \end{bmatrix} \tag{4}$$

Correctness of $\varDelta$ can be checked by the verification algorithm using the bilinear map. For $i = 0, 1$, let the CRS be $\sigma_i = (F_i, H_i, U_i, V_i, W_i)$, and let x_i, y_i be such that:

$$F_i \coloneqq x_iG, \qquad H_i \coloneqq y_iG,$$

in which case $\varDelta$ is constructed as in Eq. (4). The verifier checks that the following holds:

$$\begin{aligned} e(\delta_{0,0}, G) = e(F_0, F_1), \quad e(\delta_{0,1}, G) = e(F_0, H_1), \\ e(\delta_{1,0}, G) = e(H_0, F_1), \quad e(\delta_{1,1}, G) = e(H_0, H_1). \end{aligned} \tag{5}$$

```
$\mathsf{ZAK.P}(1^\lambda, \phi, w)$
$\mathit{\Gamma} := \mathsf{G}(1^\lambda)$
for $i = 0, 1$ do
    $(\sigma_{i,0}, \tau_i) \leftarrow \mathsf{Circ.K}(\mathit{\Gamma})$
    $\sigma_{i,1} := \sigma_{i,0} + (0, 0, 0, 0, G)$
    $\pi_{i,0} \leftarrow \mathsf{Circ.P}(\sigma_{i,0}, \phi, w)$
    $\pi_{i,1} \leftarrow \mathsf{Circ.P}(\sigma_{i,1}, \phi, w)$
Compute $\mathit{\Delta}$ from $\tau_0, \tau_1$ as in Eq. (4).
$\mathit{\Sigma} := [\sigma_{i,0}]_{i \in \{0,1\}}, \ \mathit{\Pi} = [\pi_{i,j}]_{i,j \in \{0,1\}}$
return $(\mathit{\Sigma}, \mathit{\Delta}, \mathit{\Pi})$
```

```
$\mathsf{ZAK.V}(\phi, (\mathit{\Sigma}, \mathit{\Delta}, \mathit{\Pi}))$
// Check if $\mathit{\Delta}$ is consistent with $\mathit{\Sigma}$
if not $\mathsf{DH}(\mathit{\Delta}, \mathit{\Sigma})$ return false
for $i$ in $\{0, 1\}$ do
    $\sigma_{i,1} := \sigma_0 + (0, 0, 0, 0, G)$
return $\left( \bigwedge_{i,j \in \{0,1\}} \mathsf{Circ.V}(\sigma_{i,j}, \phi, \pi_{i,j}) \right)$
```

Fig. 5. The ZAK protocol.

Let us denote by DH the algorithm that, given as input $\mathit{\Sigma}$ and $\mathit{\Delta}$ returns **true** if all equalities of Eq. (5) are satisfied, **false** otherwise. This procedure is used by the verification equation, as detailed in Fig. 5.

We now proceed with the proof of our main result, Theorem 1, which we rephrase here for completeness:

Theorem 1. *Assume that DLin and DH-KE hold for* G. *Then,* ZAK *as defined in Fig. 5 is a non-interactive zap that satisfies knowledge soundness and witness indistinguishability. In particular, we have*

$$\mathsf{Adv}^{\mathrm{ksnd}}_{\mathsf{ZAK}}(\lambda) \leq 4 \cdot \mathsf{Adv}^{\mathrm{dh\text{-}ke}}(\lambda) \quad \textit{and} \quad \mathsf{Adv}^{\mathrm{wi}}_{\mathsf{ZAK}}(\lambda) \leq 8 \cdot \mathsf{Adv}^{\mathrm{dlin}}(\lambda).$$

Completeness of the protocol is trivial: the prover (respectively, the verifier) simply performs 4 iterations of Circ proofs (respectively, verifications), and therefore correctness is implied by Theorem 5 and the fact that $\mathit{\Delta}$ as in Eq. (4) satisfies Eq. (5). We now prove knowledge soundness and witness indistinguishability.

Proof (of computational knowledge soundness). We show that for any adversary able to produce a valid proof we can construct a PPT extractor that can extract a witness from such a proof with overwhelming probability.

Let A be an adversarial prover in game KSND(λ) (Fig. 2, with $\mathsf{\Pi.K}$ void). On input 1^λ, A returns a proof consisting of $\sigma_{i,0} = (F_i, H_i, U_i, V_i, W_i)$ for $i \in \{0,1\}$, of $\mathit{\Delta} = [\delta_{i,j}]_{i,j \in \{0,1\}}$ and $\mathit{\Pi} = [\pi_{i,j}]_{i,j \in \{0,1\}}$. The game KSND$_{\mathsf{ZAK},\text{CIRC-SAT}}(\lambda)$ is given in Fig. 6. From A we construct four adversaries $\mathsf{A}_{i,j}$ (for $i,j \in \{0,1\}$) that execute A and output some components of the proof

```
Game KSND$_{\mathsf{ZAK},\text{CIRC-SAT},\mathsf{A},\mathsf{Ext_A}}(\lambda)$
$\mathit{\Gamma} := \mathsf{G}(1^\lambda)\,;\ r \leftarrow\!\!\$\ \{0,1\}^{\mathsf{A.rl}(\lambda)}$
$(\mathtt{C}, (\mathit{\Sigma}, \mathit{\Delta}, \mathit{\Pi})) := \mathsf{A}(1^\lambda; r)$
$w \leftarrow \mathsf{Ext}(1^\lambda, r)$
return $\mathsf{ZAK.V}(\mathtt{C}, (\mathit{\Sigma}, \mathit{\Delta}, \mathit{\Pi}))$ and $\mathtt{C}(w) \neq 1$
```

Fig. 6. Knowledge soundness game for the ZAK protocol.

produced by A, namely

$$\begin{aligned}(F_0,\, F_1,\, \delta_{0,0}) &= (x_0G,\, x_1G,\, x_0x_1G), && \text{(for } \mathsf{A}_{0,0}\text{)}\\ (F_0,\, H_1,\, \delta_{0,1}) &= (x_0G,\, y_1G,\, x_0y_1G), && \text{(for } \mathsf{A}_{0,1}\text{)}\\ (H_0,\, F_1,\, \delta_{1,0}) &= (y_0G,\, x_1G,\, y_0x_1G), && \text{(for } \mathsf{A}_{1,0}\text{)}\\ (H_0,\, H_1,\, \delta_{0,1}) &= (y_0G,\, y_1G,\, y_0y_1G), && \text{(for } \mathsf{A}_{1,1}\text{)}\end{aligned}$$

where x_i, y_i are such that $F_i = x_iG$, $H_i = y_iG$, and the four equations hold if $\mathsf{ZAK.V}(\mathsf{C}, (\varSigma, \varDelta, \varPi))$ returns **true**. By the DH-KE assumption there exist extractors $\mathsf{Ext}_{i,j}$ for each of the adversaries $\mathsf{A}_{i,j}$ that given its coins outputs:

$$\begin{aligned}&x_0 \text{ or } x_1, && x_0 \text{ or } y_1, && \text{(for } \mathsf{Ext}_{0,0},\, \mathsf{Ext}_{0,1}\text{)}\\ &y_0 \text{ or } x_1, && y_0 \text{ or } y_1 && \text{(for } \mathsf{Ext}_{1,0},\, \mathsf{Ext}_{1,1}\text{)}\end{aligned}$$

if the above equations hold. The statement $(x_0 \vee x_1) \wedge (y_0 \vee x_1) \wedge (x_0 \vee y_1) \wedge (y_0 \vee y_1)$ is logically equivalent to $(x_0 \wedge y_0) \vee (x_1 \wedge y_1)$. This means that together, these four extractors allow to recover either (x_0, y_0) or (x_1, y_1), that is, the extraction trapdoor for one of the CRS's. Let i^* be such that (x_{i^*}, y_{i^*}) is the extracted pair.

For $j \in \{0, 1\}$, denote with $\sigma_{i^*,j}$ the CRS $(F_{i^*}, H_{i^*}, U_{i^*}, V_{i^*}, W_{i^*} + jG)$, where $F_{i^*}, H_{i^*}, U_{i^*}, V_{i^*}, W_{i^*} \in \mathbb{G}$. Let $j^* \in \{0, 1\}$ be the smallest integer satisfying:

$$x_{i^*}^{-1}U_{i^*} + y_{i^*}^{-1}V_{i^*} - (W_{i^*} + j^*G) \neq 0G.$$

The above implies that σ_{i^*,j^*} is not a linear tuple, which means that it is a binding CRS. Let $C_{(i^*,j^*),k}$ denote the commitment to the k-th wire contained in π_{i^*,j^*}. Using the extraction algorithm described in Eq. (2) we can recover this witness:

$$w_k = \mathsf{Com.E}\big((x_{i^*}, y_{i^*}),\ C_{(i^*,j^*),k}\big).$$

It remains to prove that the extracted witness is indeed correct. Upon receiving a valid proof from adversary A, we know from the verification equation (the subroutine DH) that each $\mathsf{A}_{i,j}$ will output a valid DH triple. Therefore, extractors $\mathsf{Ext}_{i,j}$ together recover $\tau_{i^*} = (x_{i^*}, y_{j^*})$ with probability at least $1 - \sum_{i,j\in\{0,1\}} \mathsf{Adv}^{\mathrm{dhke}}_{\mathsf{G},\mathsf{A}_{i,j},\mathsf{Ext}_{i,j}}(\lambda)$, that is, by DH-KE, with overwhelming probability. Since the commitment scheme Com is perfectly binding if the CRS is not a linear tuple (Theorem 4), a message w_k is always successfully extracted. Correctness of w_k follows from the underlying proof system: by perfect soundness of Bin (Theorem 2) we are guaranteed that $w_k \in \{0, 1\}$; by perfect soundness of Circ (Theorem 5) that each gate evaluation is correct. The bound in the construction of the extractor is tight: we have $\mathsf{Adv}^{\mathrm{ksnd}}(\lambda) \leq 4 \cdot \mathsf{Adv}^{\mathrm{dhke}}(\lambda)$. □

Proof (of computational witness indistinguishability). Consider an adversary in the WI game (Fig. 1, where $\Pi.\mathsf{K}$ is void) and making $q = q(\lambda)$ queries to the PROVE oracle, each of the form $(\mathsf{C}^{(k)}, w_0^{(k)}, w_1^{(k)})$, for $0 \leq k < q$. Consider the following sequence of hybrid games where H_0 corresponds to $\mathrm{WI}_{\mathsf{ZAK},\mathsf{CIRC\text{-}SAT},\mathsf{A}}(\lambda)$ with $b = 0$ and H_{12} corresponds to $\mathrm{WI}_{\mathsf{ZAK},\mathsf{CIRC\text{-}SAT},\mathsf{A}}(\lambda)$ with $b = 1$. The games differ in how the PROVE oracle is implemented, which is specified in Fig. 7 for the first half of the hybrids (the second half is analogous). We give an overview of all hybrids in Table 2 below.

H_0 The challenger simulates an honest PROVE oracle, using (for every $k < q$) the first witness $w_0^{(k)}$ supplied by the adversary. It outputs $(\varSigma^{(k)}, \varDelta^{(k)}, \varPi^{(k)})$, where in particular we recall:

$$\varSigma^{(k)} = \begin{bmatrix} \sigma_{0,0}^{(k)} = (F_0^{(k)},\, H_0^{(k)},\, U_0^{(k)},\, V_0^{(k)},\, W_0^{(k)}) \\ \sigma_{1,0}^{(k)} = (F_1^{(k)},\, H_1^{(k)},\, U_1^{(k)},\, V_1^{(k)},\, W_1^{(k)}) \end{bmatrix} \quad \text{and} \quad \varPi^{(k)} = \begin{bmatrix} \pi_{0,0}^{(k)} & \pi_{0,1}^{(k)} \\ \pi_{1,0}^{(k)} & \pi_{1,1}^{(k)} \end{bmatrix}.$$

```
Oracle PROVE in $\mathsf{H}_1$, $\boxed{\mathsf{H}_2}$, and $\boxed{\mathsf{H}_3}$
$\Gamma := \mathsf{G}(1^\lambda)$
$(\sigma_{0,0}, \tau_i) \leftarrow \mathsf{Circ.K}(\Gamma)$
$\boxed{(\sigma_{0,0}, \tau_i) \leftarrow \mathsf{Com.K}^{(\mathrm{b})}(\Gamma)}$
$\sigma_{0,1} := \sigma_{0,0} + (0,0,0,0,G)$
$\boxed{(\sigma_{0,1}, \tau_i) \leftarrow \mathsf{Circ.K}(\Gamma)}$
$\boxed{\sigma_{0,0} := \sigma_{0,1} - (0,0,0,0,G)}$
$\pi_{0,0} \leftarrow \mathsf{Circ.P}(\sigma_{0,0}, \mathtt{C}, w_1)$
$\pi_{0,1} \leftarrow \mathsf{Circ.P}(\sigma_{0,1}, \mathtt{C}, w_0)$
// The second zap is as in $\mathsf{ZAK.P}$ using $w_0$.
$(\sigma_{1,0}, \pi_{1,0}, \pi_{1,1}) \leftarrow \mathsf{ZAP.P}(1^\lambda, \mathtt{C}, w_0)$
Compute $\Delta$ as in Eq. (4).
return $(\Sigma, \Delta, \Pi)$
```

```
Oracle PROVE in $\mathsf{H}_4$ and $\boxed{\mathsf{H}_5}$
$\Gamma := \mathsf{G}(1^\lambda)$
$(\sigma_{0,1}, \tau_i) \leftarrow \mathsf{Circ.K}(\Gamma)$
$\sigma_{0,0} := \sigma_{0,1} - (0,0,0,0,G)$
$\boxed{(\sigma_{0,1}, \tau_i) \leftarrow \mathsf{Com.K}^{(\mathrm{b})}(\Gamma)}$
$\pi_{0,0} \leftarrow \mathsf{Circ.P}(\sigma_{0,0}, \mathtt{C}, w_1)$
$\pi_{0,1} \leftarrow \mathsf{Circ.P}(\sigma_{0,1}, \mathtt{C}, w_1)$
// The second zap is as in $\mathsf{ZAK.P}$ using $w_0$.
$(\sigma_{1,0}, \pi_{1,0}, \pi_{1,1}) \leftarrow \mathsf{ZAP.P}(1^\lambda, \mathtt{C}, w_0)$
Compute $\Delta$ as in Eq. (4).
return $(\Sigma, \Delta, \Pi)$
```

Fig. 7. Overview of the simulations of the prove oracle in the first hybrid games for the proof of WI. Hybrids H_1 and H_4 are defined by ignoring all boxes (the light gray highlights the differences with respect to the previous hybrids), whereas $\boxed{\mathsf{H}_2}$ and $\boxed{\mathsf{H}_5}$ include the light boxes but not the gray one and $\boxed{\mathsf{H}_3}$ includes all boxes.

Recall that the two rows of $[\Sigma^{(k)}|\Pi^{(k)}]$ are independent zaps and that $\sigma_{0,0}^{(k)}$ and $\sigma_{1,0}^{(k)}$ are chosen to be *hiding*. The PROVE oracle computes $\sigma_{i,j}^{(k)}$ which are of the form $\sigma_{i,j}^{(k)} = \big(F_i^{(k)},\ H_i^{(k)},\ U_i^{(k)},\ V_i^{(k)},\ W_i^{(k)}+jG\big)$, for $i,j \in \{0,1\}$. Furthermore, $\pi_{i,j}^{(k)}$ is a Circ proof using $w_0^{(k)}$ under the CRS $\sigma_{i,j}^{(k)}$.

H_1 For every PROVE query, the simulator uses witness $w_1^{(k)}$ (instead of $w_0^{(k)}$) to produce $\pi_{0,0}^{(k)}$. As the respective CRS $\sigma_{0,0}^{(k)}$ was generated using the perfectly hiding commitment setup $\mathsf{Circ.K}$, the two hybrids are distributed equivalently (any commitment under a hiding key is a random linear triple; cf. Eq. (1)).

H_2 For every PROVE query, the simulator now generates CRS $\sigma_{0,0}^{(k)}$ as a *binding* key via $\mathsf{Com.K}^{(b)}$; $\sigma_{0,1}^{(k)}$ is generated as before (adding $(0,0,0,0,G)$), and so are all proofs. Note that the linking elements $\Delta^{(k)}$ can be constructed knowing only the trapdoor $(x_1^{(k)}, y_1^{(k)})$ of the CRS $\sigma_{1,0}^{(k)}$, which remained unchanged:

$$\Delta^{(k)} = \begin{bmatrix} y_1^{(k)} H_0^{(k)} & y_1^{(k)} F_0^{(k)} \\ x_1^{(k)} H_0^{(k)} & x_1^{(k)} F_0^{(k)} \end{bmatrix}. \tag{6}$$

H_1 and H_2 are computationally indistinguishable under the DLin assumption: given a DLin challenge (F,H,U,V,W), the reduction can exploit the random self-reducibility property of DLin to construct q instances of the DLin challenge: $\forall k < q$ select $\bar{x}^{(k)}, \bar{y}^{(k)}, \bar{r}^{(k)}, \bar{s}^{(k)}, \bar{z}^{(k)} \leftarrow_{\$} \mathbb{Z}_p$ and compute $\sigma_{0,0}^{(k)}$ as

$$\big(\bar{x}^{(k)}F,\ \bar{y}^{(k)}H,\ \bar{r}^{(k)}\bar{x}^{(k)}F + \bar{z}^{(k)}\bar{x}^{(k)}U,\ \bar{s}^{(k)}\bar{y}^{(k)}H + \bar{z}^{(k)}\bar{y}^{(k)}V,\ (\bar{r}^{(k)}+\bar{s}^{(k)})G + \bar{z}^{(k)}W\big).$$

Each $\sigma_{0,0}^{(k)}$ is a random linear tuple if and only if the DLin challenge is, and it is a uniformly random tuple if the DLin challenge is, as shown in [BFS16]. Computing $\sigma_{1,0}^{(k)}$ as in H_1 (hiding) and defining Δ as in Eq. (6), the simulator generates the rest of the game as defined. It returns the adversary's guess and thus breaks DLin whenever the adversary distinguishes H_1 and H_2.

Table 2. Overview of changes throughout the hybrids: (h) denotes hiding setup; (b) denotes binding setup; w_b identifies the witness used to produce the proof.

Hybrid	$\sigma_{0,0}^{(k)}$	$\pi_{0,0}^{(k)}$	$\sigma_{0,1}^{(k)}$	$\pi_{0,1}^{(k)}$	$\sigma_{1,0}^{(k)}$	$\pi_{1,0}^{(k)}$	$\sigma_{1,1}^{(k)}$	$\pi_{1,1}^{(k)}$
H_0	(h)	w_0	(b)	w_0	(h)	w_0	(b)	w_0
H_1		w_1						
H_2	(b)							
H_3			(h)					
H_4				w_1				
H_5			(b)					
H_6	(h)							
H_7						w_1		
H_8					(b)			
H_9							(h)	
H_{10}								w_1
H_{11}							(b)	
H_{12}	(h)	w_1	(b)	w_1	(h)	w_1	(b)	w_1

H_3 The simulator replaces each CRS $\sigma_{0,1}^{(k)}$ for all $k < q$ with a *hiding* commitment and defines $\sigma_{0,0}^{(k)} := \sigma_{0,1}^{(k)} - (0, 0, 0, 0, G)$, which is therefore (once again) binding. More specifically, the simulator creates a linear tuple invoking $\mathsf{Circ.K}$:

$$\sigma_{0,1}^{(k)} = \big(x_0^{(k)} G,\ y_0^{(k)} G,\ x_0^{(k)} r^{(k)} G,\ y_0^{(k)} s^{(k)} G,\ (r^{(k)} + s^{(k)}) G\big)$$

where $x_0^{(k)}, y_0^{(k)}, r^{(k)}, s^{(k)} \leftarrow_{\$} \mathbb{Z}_p$.
The two distributions are proven computationally indistinguishable under DLin by an argument analogous to the one for $\mathsf{H}_1 \to \mathsf{H}_2$. This time the challenger constructs all the various instances of the DLin challenge for $\sigma_{0,1}^{(k)}$, while $\sigma_{0,0}^{(k)}$ is derived. From there, the proof proceeds identically.

H_4 The simulator replaces each proof $\pi_{0,1}^{(k)}$ by using $w_1^{(k)}$ instead of $w_0^{(k)}$ $(\forall k < q)$.
This hybrid is equivalently distributed as the previous one; this is proved via the same argument as for $\mathsf{H}_0 \to \mathsf{H}_1$.

H_5 The simulator switches $\sigma_{0,1}^{(k)}$ from a hiding to a binding key. This game hop is analogous to the hop $\mathsf{H}_1 \to \mathsf{H}_2$ (which switched $\sigma_{0,0}^{(k)}$ from hiding to binding).

H_6 The simulator switches $\sigma_{0,0}^{(k)}$ from binding to hiding. Indistinguishability from the previous hybrid is shown analogously to the hop $\mathsf{H}_2 \to \mathsf{H}_3$. Note that in this hybrid the first zap $(\sigma_{0,0}^{(k)}, \pi_{0,0}^{(k)}, \pi_{0,1}^{(k)})$ is distributed according to the protocol specification, but using witness $w_1^{(k)}$.

Hybrids H_7 to H_{12} are now defined analogously to hybrids H_1 to H_6, except for applying all changes to $\sigma_1^{(k)}$ and $\pi_{1,0}^{(k)}$ and $\pi_{1,1}^{(k)}$. In hybrid H_{12} the adversary is then given arguments of knowledge for witness w_1.

As the difference between hybrids H_1 and H_{12} is bounded by 8 times the advantage of a DLin distinguisher, the adversary has total advantage

$$\mathsf{Adv}^{\mathrm{wi}}_{\mathsf{ZAK,C,A}}(\lambda) \leq 8 \cdot \mathsf{Adv}^{\mathrm{dlin}}_{\mathsf{ZAK,C,A}}(\lambda) = \mathsf{negl}(\lambda)\,.$$

The bound is thus tight. □

Acknowledgements. The authors would like to thank one of the anonymous reviewers of PKC 2018 for suggesting the construction of zaps of knowledge based on SXDH. The first author is supported by the French ANR EfTrEC project (ANR-16-CE39-0002). The second author is supported by ERC grant 639554 (project aSCEND).

References

BBS04. Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 41–55. Springer, 2004.

BCG+14. Eli Ben-Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *2014 IEEE Symposium on Security and Privacy*, pages 459–474. IEEE Computer Society Press, May 2014.

BCI+10. Eric Brier, Jean-Sébastien Coron, Thomas Icart, David Madore, Hugues Randriam, and Mehdi Tibouchi. Efficient indifferentiable hashing into ordinary elliptic curves. In Tal Rabin, editor, *CRYPTO 2010*, volume 6223 of *LNCS*, pages 237–254. Springer, 2010.

BFM88. Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications (extended abstract). In *20th ACM STOC*, pages 103–112. ACM Press, May 1988.

BFS16. Mihir Bellare, Georg Fuchsbauer, and Alessandra Scafuro. NIZKs with an untrusted CRS: Security in the face of parameter subversion. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *ASIACRYPT 2016, Part II*, volume 10032 of *LNCS*, pages 777–804. Springer, 2016.

BG93. Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In Ernest F. Brickell, editor, *CRYPTO'92*, volume 740 of *LNCS*, pages 390–420. Springer, 1993.

BHY09. Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In Antoine Joux, editor, *EUROCRYPT 2009*, volume 5479 of *LNCS*, pages 1–35. Springer, 2009.

BP15. Nir Bitansky and Omer Paneth. ZAPs and non-interactive witness indistinguishability from indistinguishability obfuscation. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 401–427. Springer, 2015.

BW06. Xavier Boyen and Brent Waters. Compact group signatures without random oracles. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 427–444. Springer, 2006.

BW07. Xavier Boyen and Brent Waters. Full-domain subgroup hiding and constant-size group signatures. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *PKC 2007*, volume 4450 of *LNCS*, pages 1–15. Springer, 2007.

DN00. Cynthia Dwork and Moni Naor. Zaps and their applications. In *41st FOCS*, pages 283–293. IEEE Computer Society Press, November 2000.

FLS90. Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st FOCS*, pages 308–317. IEEE Computer Society Press, October 1990.

FS90. Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *22nd ACM STOC*, pages 416–426. ACM Press, May 1990.

Fuc18. Georg Fuchsbauer. Subversion-zero-knowledge SNARKS. In Michel Abdalla, editor, *PKC 2018*, LNCS. Springer, 2018.

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

GO94. Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, 1994.

Gol93. Oded Goldreich. A uniform-complexity treatment of encryption and zero-knowledge. *Journal of Cryptology*, 6(1):21–53, 1993.

GOS06a. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for NIZK. In Cynthia Dwork, editor, *CRYPTO 2006*, volume 4117 of *LNCS*, pages 97–111. Springer, 2006.

GOS06b. Jens Groth, Rafail Ostrovsky, and Amit Sahai. Perfect non-interactive zero knowledge for NP. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 339–358. Springer, 2006.

Gro06. Jens Groth. Simulation-sound NIZK proofs for a practical language and constant size group signatures. In Xuejia Lai and Kefei Chen, editors, *ASIACRYPT 2006*, volume 4284 of *LNCS*, pages 444–459. Springer, 2006.

Gro10. Jens Groth. Short pairing-based non-interactive zero-knowledge arguments. In Masayuki Abe, editor, *ASIACRYPT 2010*, volume 6477 of *LNCS*, pages 321–340. Springer, 2010.

GS08. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 415–432. Springer, 2008.

KN08. Gillat Kol and Moni Naor. Cryptography and game theory: Designing protocols for exchanging information. In Ran Canetti, editor, *TCC 2008*, volume 4948 of *LNCS*, pages 320–339. Springer, 2008.

PVW08. Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *CRYPTO 2008*, volume 5157 of *LNCS*, pages 554–571. Springer, 2008.

Ràf15. Carla Ràfols. Stretching Groth-Sahai: NIZK proofs of partial satisfiability. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *TCC 2015, Part II*, volume 9015 of *LNCS*, pages 247–276. Springer, 2015.

A GOS's proof of binarity

Consider the CRS $\sigma \coloneqq (F, H, U, V, W)$ and $\Gamma \coloneqq (p, \mathbb{G}, \mathbb{G}_T, e, G)$ resulting from the execution of (one of the two types of) the key generation algorithm $\mathsf{Com.K}$. Note that F, H are two generators of $\mathbb{G}$ and (U, V, W) is a linear tuple w.r.t. (F, H, G) iff the key generation algorithm is chosen hiding. Groth, Ostrovsky and Sahai [GOS06a] presented a witness-indistinguishable non-interactive proof system Bin for proving that $C \in \mathbb{G}^3$ is a commitment to $\{0,1\}$ under σ. The intuition behind this construction is that, by the homomorphic property of Com, proving that C commits to a bit is equivalent to showing that either $C = (C_0, C_1, C_2)$ or $C' \coloneqq C - \mathsf{Com.C}(1; (0,0)) = C - (U, V, W)$ is a linear tuple with respect to (F, H, G). If we consider the discrete logarithms w.r.t (F, H, G) of the above commitments, i.e. letting $C = (r_0 F, s_0 H, t_0 G)$ and $C' = (r_1 F, s_1 H, t_1 G)$, we have that:

$$\begin{aligned}
&C \text{ or } C' \text{ is a linear tuple}\\
\iff\ & t_0 = r_0 + s_0 \text{ or } t_1 = r_1 + s_1\\
\iff\ & (r_0 + s_0 - t_0)(r_1 + s_1 - t_1) = 0\\
\iff\ & r_0 r_1 + r_0 s_1 + s_0 r_1 + s_0 s_1 + t_0 t_1 - (r_0 t_1 + t_0 r_1 + s_0 t_1 + t_0 s_1) = 0.
\end{aligned} \tag{7}$$

Consider a prover $\mathsf{Bin.P}$ holding the witness $(b, r, s) \in \{0,1\} \times \mathbb{Z}_p \times \mathbb{Z}_p$ for (C, C'), where b indicates which tuple is linear and r, s are its contained randomness. In order to convince a verifier, it proceeds as follows: choose $t \leftarrow_\$ \mathbb{Z}_p$ and let $\Pi = [\pi_{i,j}]_{i \in \{0,1\}, j \in \{0,1,2\}}$, where:

$$\begin{aligned}
\pi_{0,0} &\coloneqq r(2b-1)U + r^2 F & \pi_{1,0} &\coloneqq s(2b-1)U + (rs + t)F\\
\pi_{0,1} &\coloneqq r(2b-1)V + (rs - t)H & \pi_{1,1} &\coloneqq s(2b-1)V + s^2 H\\
\pi_{0,2} &\coloneqq r(2b-1)W + (r^2 + rs + t)G & \pi_{1,2} &\coloneqq s(2b-1)W + (s^2 + rs - t)G
\end{aligned} \tag{8}$$

A verifier $\mathsf{Bin.V}$, on input the CRS σ, the statement C and Π, computes $\pi_{2,j} \coloneqq \pi_{1,j} + \pi_{0,j}$ for $j = 0, 1, 3$ and returns **true** if all the following equations are satisfied:

$$\begin{aligned}
e(F, \pi_{0,0}) &= e(C_0, C_0 - U),\\
&\quad e(F, \pi_{0,1}) + e(H, \pi_{1,0}) = e(C_0, C_1 - V) + e(C_1, C_0 - U)\\
e(H, \pi_{1,1}) &= e(C_1, C_1 - V),\\
&\quad e(F, \pi_{0,2}) + e(G, \pi_{2,0}) = e(C_0, C_2 - W) + e(C_2, C_0 - U)\\
e(G, \pi_{2,2}) &= e(C_2, C_2 - W),\\
&\quad e(H, \pi_{1,2}) + e(G, \pi_{2,1}) = e(C_1, C_2 - W) + e(C_2, C_1 - V).
\end{aligned} \tag{9}$$

If we consider, as we did for the commitments, the discrete logarithms of the proof matrix w.r.t. (F, H, G), i.e. we put

$$m_{i,0} \coloneqq \log_F(\pi_{i,0}), \qquad m_{i,1} \coloneqq \log_H(\pi_{i,1}), \qquad m_{i,2} \coloneqq \log_G(\pi_{i,2}),$$

```
Bin.P(σ, C, (b, r, s))
  Construct $\Pi$ as per Eq. (8)
  return $\Pi$
```

```
Bin.V(σ, C, Π)
  $[\pi_{i,j}]_{i\in\{0,1\},j\in\{0,1,2\}} = \Pi$
  for $j = 0, 1, 2$ do $\pi_{2,j} := \pi_{1,j} + \pi_{0,j}$
  return (Eq. (9))
```

Fig. 8. The Bin protocol.

for $i = 0, 1$, then we observe that the verification equation sets: $m_{2,i} := m_{0,i} + m_{1,i}$, and then checks the following:

$$\begin{aligned} m_{0,0} &= r_0 r_1 & m_{0,1} + m_{1,0} &= r_0 s_1 + s_0 r_1 \\ m_{1,1} &= s_0 s_1 & m_{0,2} + m_{2,0} &= r_0 t_1 + t_0 r_1 \\ m_{2,2} &= t_0 t_1 & m_{1,2} + m_{2,1} &= s_0 t_1 + t_0 s_1 \end{aligned} \tag{10}$$

By substitution, this is exactly what Eq. (7) affirms.

As previously mentioned, the key generation algorithm is identical to Com.K. If the setup is perfectly binding, perfect completeness and perfect soundness follow immediately from Eq. (10). Perfect witness indistinguishability follows from the observation that a proof with a witness $(0, r_0, s_0)$ gives the same proof as using witness $(1, r_1, s_1)$ with randomness $t' = t + r_0 s_1 - s_0 r_1$. On the other hand, on a perfectly hiding key generation every commitment is a linear tuple, and thus there is nothing to prove.

Theorem 2 ([GOS06a]). *The protocol* Bin *is a non-interactive proof system with perfect completeness, perfect soundness, and perfect witness indistinguishability.*

B Non-interactive zaps of knowledge in asymmetric groups

In this section we show an alternative and more efficient approach to constructing non-interactive zaps of knowledge for circuit satisfiability. In contrast to symmetric bilinear groups used in Section 2, we will work with asymmetric pairings, that is, bilinear maps $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ (where $\mathbb{G}_1 = \langle G_1 \rangle$, $\mathbb{G}_2 = \langle G_2 \rangle$ and $\mathbb{G}_T$ are abelian additive groups of prime order p). We assume a deterministic algorithm G that outputs an (asymmetric) group description $\Gamma := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2)$.

By extending GOS proofs [GOS06b], Goth and Sahai [GS08] provide a general framework for non-interactive witness-indistinguishable (NIWI) proof systems, which can be based (among other computational assumptions) on SXDH. The SXDH assumption for an asymmetric pairing group generator G informally states that the decisional Diffie-Hellman assumption holds in both $\mathbb{G}_1$ and $\mathbb{G}_2$.

Assumption 3 (SXDH) *We say that the Symmetric External Diffie-Hellman assumption holds for the asymmetric bilinear group generator* G *if for all PPT adversaries* A *we have:*

$$\begin{aligned} \mathsf{Adv}^{\mathrm{xdh1}}_{\mathsf{G},\mathsf{A}}(\lambda) &:= \Pr\left[\mathrm{XDH}^{1}_{\mathsf{G},\mathsf{A}}(\lambda)\right] - 1/2 = \mathsf{negl}(\lambda) \quad \textit{and} \\ \mathsf{Adv}^{\mathrm{xdh2}}_{\mathsf{G},\mathsf{A}}(\lambda) &:= \Pr\left[\mathrm{XDH}^{2}_{\mathsf{G},\mathsf{A}}(\lambda)\right] - 1/2 = \mathsf{negl}(\lambda)\,, \end{aligned}$$

where $\mathrm{XDH}^{I}_{\mathsf{G},\mathsf{A}}(\lambda)$ *(for* $I = 1, 2$*) is defined in Fig. 9.*

In order to construct zaps of knowledge over asymmetric bilinear groups, we require the analogue of DH-KE for such groups, in particular for their first base group $\mathbb{G}_1$. We give a formal definition.

```
Game $\mathrm{XDH}^{I}_{\mathsf{G},\mathsf{A}}(\lambda)$
$\Gamma := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2) := \mathsf{G}(1^\lambda)$
$b \leftarrow\$ \{0,1\}$
$x, y \leftarrow \mathbb{Z}_p^*$
if $b = 1$ then $H := xyG_I$
else $H \leftarrow\$ \mathbb{G}_I$
$b' \leftarrow \mathsf{A}(\Gamma, xG_I, yG_I, H)$
return $(b = b')$
```

```
Game $\mathrm{ADH\text{-}KE}_{\mathsf{G},\mathsf{A},\mathsf{Ext}}(\lambda)$
$\Gamma := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2) := \mathsf{G}(1^\lambda)$
$r \leftarrow\$ \{0,1\}^{\mathsf{A.rl}(\lambda)}$
$(X, Y, Z) := \mathsf{A}(\Gamma; r)$
$s \leftarrow \mathsf{Ext}(\Gamma, r)$
if $sG_1 = X \vee sG_1 = Y$ then return 0
return $(Z = \log_{G_1}(X) \cdot Y)$
```

Fig. 9. Games for Assumptions 3 (SXDH) and 4 (ADH-KE).

Assumption 4 (ADH-KE) *The Asymmetric Diffie-Hellman Knowledge of Exponent assumption holds for (the first base group of) the asymmetric group generator* G *if for any PPT adversary* A *there exists a PPT extractor* Ext *such that:*

$$\mathsf{Adv}^{\text{adh-ke}}_{\mathsf{G},\mathsf{A},\mathsf{Ext}}(\lambda) := \Pr\left[\mathrm{ADH\text{-}KE}_{\mathsf{G},\mathsf{A},\mathsf{Ext}}(\lambda)\right] = \mathsf{negl}(\lambda)\,,$$

where the game $\mathrm{ADH\text{-}KE}_{\mathsf{G},\mathsf{A},\mathsf{Ext}}(\lambda)$ *is defined in Fig. 9.*

Groth-Sahai (GS) proofs [GS08] achieve improved efficiency by working for group-dependent languages, in contrast to the more elementary proof system Bin of "bit commitment" (given in Appendix A) used for circuit satisfiability. More recently, Ràfols [Ràf15] gave a construction of non-interactive zaps from GS proofs, which leads to more efficient non-interactive zaps (by a constant factor). Relying on the asymmetric variant of the DH-KE assumption, we show how to achieve knowledge soundness also for GS zaps. Interestingly, the scheme does not require any alteration to the protocol, that is, under ADH-KE we can show that a GS zap is already a GS zap of knowledge.

B.1 GS zaps

We first describe the GS-based zap and then argue that it satisfies knowledge soundness under ADH-KE.

SXDH commitments and proofs of binarity. The SXDH commitment scheme of Groth and Sahai [GS08] allows to commit to values in $\mathbb{Z}_p$ both in $\mathbb{G}_1$ and in $\mathbb{G}_2$ (here we parametrize the algorithm with $I \in \{1,2\}$ for compactness). The properties of the scheme are very similar to those of GOS's [GOS06a] DLin-based commitments (Section 3). Again, commitment keys can be generated in one of two possible "modes", one perfectly hiding and one perfectly binding.

```
$\mathsf{Com.K}^{(\mathrm{h})}_I$
$\tau := (x, y) \leftarrow\$ (\mathbb{Z}_p^*)^2$
$\vec{V} := (xG_I, G_I)^\top$
$\vec{W} := (xyG_I, yG_I)^\top$
$\sigma := (\vec{V}, \vec{W})$
return $(\sigma, \tau)$
```

```
$\mathsf{Com.K}^{(\mathrm{b})}_I$
$\tau := (x, z) \leftarrow\$ (\mathbb{Z}_p^*)^2;\ y \leftarrow\$ \mathbb{Z}_p^*$
$\vec{V} := (xG_I, G_I)^\top$
$\vec{W} := (xyG_I, (y+z)G_I)^\top$
$\sigma := (\vec{V}, \vec{W})$
return $(\sigma, \tau)$
```

The commitment key thus consists of vectors $\vec{V}, \vec{W} \in \mathbb{G}_I^2$, for $I = 1, 2$. Committing to a value $m \in \mathbb{Z}_p$ is performed by sampling $r \leftarrow_\$ \mathbb{Z}_p$ and computing:

$$\mathsf{Com.C}_I(m; r) := m\vec{W} + r\vec{V}.$$

The commitment scheme is additively homomorphic, since $\mathsf{Com.C}_I(m_0; r_0) + \mathsf{Com.C}_I(m_1; r_1) = \mathsf{Com.C}_I(m_0 + m_1; r_0 + r_1)$. The two setups $\mathsf{Com.K}_I^{(\mathrm{h})}$ and $\mathsf{Com.K}_I^{(\mathrm{b})}$ are computationally indistinguishable under DDH in $\mathbb{G}_I$: hiding setup returns a DH triple (V_0, W_1, W_0) with respect to $V_1 = G_I$, whereas binding setup returns random values (V_0, W_1, W_0).

If $\vec{V}, \vec{W}$ are linearly dependent, which is the case when generated by $\mathsf{Com.K}_I^{(\mathrm{h})}$, then the commitment is perfectly hiding; a commitment $\vec{C}$ to a value m can be opened to any value $m' \in \mathbb{Z}_p$ given the trapdoor information $\tau = (x, y)$:

$$\mathsf{Com.C}_I(m; r) = \begin{pmatrix} x(my + r)G_I \\ (my + r)G_I \end{pmatrix} = \mathsf{Com.C}_I\big(m'; r + (m - m')y\big).$$

If $\vec{V}, \vec{W}$ are linearly independent then the commitment is perfectly binding and for message spaces of logarithmic size the committed value can be *extracted* using the trapdoor information $\tau = (x, z)$ generated by $\mathsf{Com.K}_I^{(\mathrm{b})}$:

$$m = \mathsf{Com.E}_I(\tau,\ \vec{C}) := \mathsf{dLog}\big(z^{-1}(C_1 - x^{-1}C_0)\big).$$

A commitment in $\mathbb{G}_I$ can be shown to be bound to a bit via two quadratic equations in $\mathbb{Z}_p$, as introduced by Groth and Sahai [GS08]. To do so, we require another commitment in the opposite source group $\mathbb{G}_{3-I}$. Let b_1 be the value committed over $\mathbb{G}_1$ and b_2 the value committed over $\mathbb{G}_2$. Our goal is to prove that $b_1 \in \{0, 1\}$; at the same time we prove $b_1 = b_2$. This can be done by proving that the commitments satisfy the following two equations:

$$b_1(b_2 - 1) = 0 \qquad \text{and} \qquad b_2(b_1 - 1) = 0\ . \tag{11}$$

We refer the reader to [GS08, §9 pp. 28] for how to construct proofs for the above equations being satisfied by the committed values.[3] A proof for one such equation consists of one element from each source group. We can thus define a proof system Bin, which, given a commitment in $\mathbb{G}_1$ and another one in $\mathbb{G}_2$, proves that the committed values are bits using $2(|\mathbb{G}_1| + |\mathbb{G}_2|)$ group elements. The key generation algorithm $\mathsf{Bin.K}$ simply executes $\mathsf{Com.K}_1^{(b)}$ and $\mathsf{Com.K}_2^{(b)}$.

Proofs of circuit satisfiability. Now that we have a witness-indistinguishable system for proving that a commitment is bound to a bit $b \in \{0, 1\}$ over asymmetric bilinear groups under the SXDH assumption, we can construct a protocol Circ for proving circuit satisfiability analogously to scheme by GOS [GOS06b] given in Section 4: The prover commits to each wire in the circuit twice (once in $\mathbb{G}_1$ and once in $\mathbb{G}_2$), proves that the committed values are to either 0 or 1, and for each NAND gate with input wire values a, b and output wire value c it proves that $(a + b + 2c - 2) \in \{0, 1\}$. The output commitment is fixed again to $\mathsf{Com.C}(1; 0)$. This defines $\mathsf{Circ.P}((\sigma_1, \sigma_2), \phi, w)$ where σ_I is a CRS in group $\mathbb{G}_I$, ϕ is the statement, i.e., a circuit description and w is the witness, a satisfying assignment. A proof π consists thus of commitments $C_{1,k} \in \mathbb{G}_1^2$ and $C_{2,k} \in \mathbb{G}_2^2$ and proofs of binarity π_k for every wire w_k, and moreover proofs π_i for every gate g_i.

[3] In GS notation, the equations are defined by setting the parameters $\vec{a} := (0)$, $\vec{b} := (-1)$, $\Gamma := [1]$, and $t := 0$ for the first equation and $\vec{a} := (-1)$, $\vec{b} := (0)$, $\Gamma := [1]$, $t := 0$ for the second.

```
$\mathsf{ZAP.P}(1^\lambda, \phi, w)$
$\mathit{\Gamma} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2) := \mathsf{G}(1^\lambda)$
for $I = 1, 2$ do
    $\sigma_{I,0} \leftarrow \mathsf{Com.K}_I^{(h)}(\mathit{\Gamma})$
    $\sigma_{I,1} := \sigma_{I,0} - ((0,0)^\top, (0, G_I)^\top)$
Compute $(\vec{C}_{I,j,k})_{I,j,k}$ as per Eq. (14)
for $i, j \in \{0,1\}$ do
    $\pi_{i,j} \leftarrow \mathsf{Circ.P}((\sigma_{1,i}, \sigma_{2,j}), \phi, w)$
return $(\sigma_{1,0}, \sigma_{2,0}, (\vec{C}_{I,j,k})_{I,j,k}, [\pi_{i,j}]_{i,j})$
```

```
Procedure $\mathsf{Test\text{-}DH}(A, B, C, s)$
$\mathit{\Gamma} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, G_1, G_2) := \mathsf{G}(1^\lambda)$
if $sG_1 = A$ and $sB = C$
    return true
if $sG_1 = B$ and $sA = C$
    return true
else
    return false
```

Fig. 10. The non-interactive zap scheme based on SXDH and the procedure for testing DH triples used in the proof.

A zap from SXDH. Again, a ZAP is constructed from Circ analogously to the GOS zap [GOS06a] given in Section 4. There, a zap consisted of 2 Circ-proofs for two related CRS's of which one was guaranteed to be binding and thus lead to sound proofs. For SXDH we now create two related CRS's in *each* group $\mathbb{G}_1$ and $\mathbb{G}_2$, so we are guaranteed that for each group one of them is binding. Intuitively, we need to construct 4 Circ proofs, one for each combination of a CRS in $\mathbb{G}_1$ with one from $\mathbb{G}_2$. We are then guaranteed that one of the four proofs is under *two* binding CRS, which asserts that the prover cannot cheat. (Note that we do not actually need four full Circ proofs, as they can share the commitments.)

More specifically, the prover constructs two CRS $\sigma_{I,0}$, for $I = 1, 2$, for perfectly hiding SXDH commitments in $\mathbb{G}_1$ and $\mathbb{G}_2$. Then, it computes (again for $I = 1, 2$):

$$\sigma_{I,1} := \sigma_{I,0} - \big((0,0)^\top, (0, G_I)^\top\big). \tag{12}$$

As for the zap described in Section 4, $\sigma_{I,1}$ is deterministically generated from $\sigma_{I,0}$ and at least one of the two CRS leads to perfectly binding commitments. For simplicity, we will refer to the following matrix of CRS in order to perform Circ proofs:

$$\mathit{\Sigma} := [(\sigma_{1,i}, \sigma_{2,j})]_{0 \le i,j \le 1} := \begin{bmatrix} (\sigma_{1,0}, \sigma_{2,0}) & (\sigma_{1,0}, \sigma_{2,1}) \\ (\sigma_{1,1}, \sigma_{2,0}) & (\sigma_{1,1}, \sigma_{2,1}) \end{bmatrix}. \tag{13}$$

Then, the prover commits to every wire value w_k computing:

$$\vec{C}_{I,j,k} \leftarrow \mathsf{Com.C}_I(\sigma_{I,j}, w_k), \tag{14}$$

for each $I \in \{1,2\}$ and each $j \in \{0,1\}$. Reusing these commitments, the prover now computes four Circ proofs $\pi_{i,j}$ (with $i, j \in \{0,1\}$). This boils down to computing, for all $i \in \{0,1\}$ and all wire indices k:

$$\pi_{i,j,k} \leftarrow \mathsf{Bin.P}\big((\sigma_{1,i}, \sigma_{2,j}),\ (\vec{C}_{1,i,k}, \vec{C}_{2,j,k}),\ w_k\big), \tag{15}$$

and proceeding in the same way for all gates. With a slight abuse of notation, in the explicit construction of Figure 10 we denote this whole process with:

$$\pi_{i,j} \leftarrow \mathsf{Circ.P}\big((\sigma_{1,i}, \sigma_{2,j}),\ \phi,\ w\big),$$

keeping in mind that the commitments are not recomputed for each proof, and that instead we are using the commitments $\vec{C}_{I,i,k}$ in $\mathbb{G}_I$ to to wire w_k under the CRS $\sigma_{I,i}$.

The construction of $\mathsf{ZAP.V}(\phi, \pi)$ is straightforward: Upon receiving a proof

$$\big(\sigma_{1,0}, \sigma_{2,0},\ (\vec{C}_{1,j,k}, \vec{C}_{2,j,k})_{i,j\in\{0,1\}},\ [\pi_{i,j}]_{i,j\in\{0,1\}}\big)$$

the verifier computes the correlated CRS's $\sigma_{1,1}, \sigma_{2,1}$ according to Eq. (12) and verifies each of the proofs $\pi_{i,j}$ for $i, j \in \{0,1\}$ using $\mathsf{Circ.V}((\sigma_{1,i}, \sigma_{2,j}),\ \phi,\ \pi_{i,j})$ (using the respective commitments, as we described above). It returns true if all proofs verified.

Theorem 3. *Assume SXDH and ADH-KE holds for the asymmetric group generation* G*. Then* ZAP *as defined in Fig. 10 is a non-interactive zap that satisfies knowledge soundness and witness indistinguishability.*

Witness indistinguishability of the ZAP proof follows from an hybrid argument analogous to the proof of witness-indistinguishability of [GOS06a]. We now prove that the scheme also satisfies computational knowledge soundness.

Proof (of computational knowledge soundness). Let A be the PPT adversary in the game $\mathrm{KSND}_{\mathsf{A},\mathsf{ZAP}}(\lambda)$ able to produce a proof for which $\mathsf{ZAP.V}$ returned **true**. The proof is of the form:

$$\big(\sigma_{1,0},\ \sigma_{2,0},\ (\vec{C}_{1,j,k}, \vec{C}_{2,j,k})_{i,j\in\{0,1\}},\ [\pi_{i,j}]_{i,j})\big)$$

where $\pi_{i,j}$ is a valid Circ proof under the CRS $(\sigma_{1,i}, \sigma_{2,j})$ - with $\sigma_{1,1}$ and $\sigma_{2,1}$ derived from $\sigma_{1,0}$ and $\sigma_{2,0}$ as per Equation (12).

First, we claim that the extractor is able to find the index $i^* \in \{0,1\}$ of the perfectly binding CRS for $\mathbb{G}_1$. Consider the adversary $\mathsf{A}_{1,0}$ ($\mathsf{A}_{1,1}$, resp.) that behaves as A, but simply outputs the elements (V_0, W_1, W_0) contained in CRS $\sigma_{1,0}$ ($\sigma_{1,1}$, resp.). By ADH-KE there exists an extractor $\mathsf{Ext}_{1,0}$ ($\mathsf{Ext}_{1,1}$, resp.) that outputs a value s_0 (s_1, resp.) in $\mathbb{Z}_p$. If the triple the adversary outputs is a DH triple (which is the case for a perfectly hiding setup), the respective extractor will output the discrete logarithm of one of the first two elements (except with negligible probability). This can be efficiently tested: for a value s output by the extractor, either

$$\begin{aligned} sG_1 &= V_0 \qquad \text{and} \quad sW_1 = W_0\ , \quad \text{or} \\ sG_1 &= W_1 \qquad \text{and} \quad sV_0 = W_0\ , \end{aligned} \tag{16}$$

hold. Thus, let $i^* \in \{0,1\}$ be the first value s_{i*} for which Eq. (16) does *not* hold. There exists such an i^* because at most one of $\sigma_{1,0}$ and $\sigma_{1,1} = \sigma_{1,0} - ((0,0)^\top, (0,G_1)^\top)$ can be a DH triple and thus a hiding commitment key. In $\mathbb{G}_2$ can can also be at most one hiding commitment key; let j^* be the smallest index such that σ_{2,j^*} is binding. Note that our extractor will not know the value of j^*. The CRS $(\sigma_{1,i^*}, \sigma_{2,j^*})$ is thus of type "perfectly binding".

By soundness of the Bin proof associated to every pair $(\vec{C}_{1,i^*,k}, \vec{C}_{2,j^*k})$, the committed values $b_{1,k}$ and $b_{2,k}$ satisfy $b_{1,k} = b_{2,k}$ and $b_{1,k}, b_{2,k} \in \{0,1\}$. It now remains to show that these values contained in the commitments corresponding to input wires (which by perfect soundness of Circ constitute a satisfying assignment) can be extracted; in particular, we extract the value from $\vec{C}_{1,i^*,k}$ (note that the extractor knows i^*).

Using (once again) the initial adversary A, we can construct multiple adversaries $\mathsf{A}_k^{(b)}$, one for each commitment $\vec{C}_{1,i^*,k} = (C_{1,i^*,k,0}, C_{1,i^*,k,1})$ to an input wire, and each possible wire value $b = 0, 1$. The adversary $\mathsf{A}_k^{(b)}$ runs A and outputs:

$$(V_0,\ C_{1,i^*,k,1},\ C_{1,i^*,k,0}), \qquad \text{(for } \mathsf{A}_k^{(0)}\text{)}$$

$$(V_0,\ C_{1,i^*,k,1} - W_1,\ C_{1,i^*,k,0} - W_0) \qquad \text{(for } \mathsf{A}_k^{(1)}\text{)}$$

where $\sigma_{1,i^*} = (V_0, V_1, W_0, W_1)$. Note that if $\vec{C}_{1,i^*,k}$ is a commitment to 0 then $(V_0, \vec{C}_{1,i^*,k})$ it is of the form (xG_1, rG_1, rxG_1) for some $x, r \in \mathbb{Z}_p$, and thus a DH triple. If $\vec{C}_k$ is a commitment to one, then $\vec{C} - \vec{W}$ is a commitment to 0 and thus $(V_0, \vec{C}_{1,i^*,k} - \vec{W})$ is a DH triple as above.

By ADH-KE for each adversary $\mathsf{A}_k^{(b)}$ there exists an extractor $\mathsf{Ext}_k^{(b)}$ that outputs some value $s_k^{(b)}$ (with $b = 0, 1$), if $\mathsf{A}_k^{(b)}$ output a DH triple. Using the same reasoning of Eq. (16), we can test which of the two triples is a valid DH triple. To do so, we use the procedure $\mathsf{Test\text{-}DH}$ depicted in Fig. 10. For each commitment $\vec{C}_{1,i^*,k}$, there exists a single index b_k for which the sub-procedure $\mathsf{Test\text{-}DH}$ returned **true**: if there were more than one we would be violating the perfect binding property of the commitment scheme, if there were none we would be violating the perfect soundness of Bin (as the commitment would be bound to a value different from $0, 1$).

At this point, we are done: the extractor for knowledge soundness runs all above extractors and recovers the bit b_k from every commitment, which is the correct wire value because of soundness of the Circ protocol under a perfectly binding key generation.

As we needed to construct as many extractors as there are input wires in the circuit, the security loss depends on the size of the circuit. □