4-Round Resettably-Sound Zero Knowledge

Kai-Min Chung[1] Rafail Ostrovsky[2], Rafael Pass[3], and Muthuramakrishnan Venkitasubramaniam[4] Ivan Visconti[5]

[1] Academia Sinica, Taiwan,
kmchung@iis.sinica.edu.tw
[2] UCLA, Los Angeles, CA, USA,
rafail@cs.ucla.edu
[3] Cornell University, Ithaca, NY 14850, USA,
chung@cs.cornell.edu
[4] University of Rochester, Rochester, NY 14627, USA
muthuv@cs.rochester.edu
[5] University of Salerno, Italy,
visconti@unisa.it

Abstract. While 4-round constructions of zero-knowledge arguments are known based on the existence of one-way functions, constuctions of *resettably-sound* zero-knowledge arguments require either stronger assumptions (the existence of a fully-homomorphic encryption scheme), or more communication rounds. We close this gap by demonstrating a 4-round resettably-sound zero-knowledge argument for NP based on the existence of one-way functions.

1 Introduction

Zero-knowledge (ZK) interactive protocols [18] are paradoxical constructs that allow one player (called the Prover) to convince another player (called the Verifier) of the validity of a mathematical statement $x \in L$, while providing *zero additional knowledge* to the Verifier. We are here interested in a stronger notion of zero-knowledge arguments known as *resettably-sound zero-knowledge*. This notion, first introduced by Barak, Goldwasser, Goldreich and Lindell (BGGL)[2], additionally requires the soundness property to hold even if the malicious prover is allowed to "reset" and "restart" the verifier. This model is particularly relevant for cryptographic protocols being executed on embedded devices, such as smart cards. BGGL provided a construction of a resettably-sound zero-knowledge argument for NP based on the existence of collision-resistant hash-functions. More recently, Bitansky and Paneth [5] presented a resettably-sound zero-knowledge argument based on the existence of an *oblivious transfer (OT) protocol*. Finally, Chung, Pass and Seth (CPS) [10] show how to construct such protocol based on the minimal assumption of one-way functions (OWFs).[6]

[6] As shown by Ostrovsky and Wigderson, one-way functions are also "essentially" necessary for non-trivial zero-knowledge [25]. In [9] one-way functions have been shown to suffice also when resettable zero knowledge is desired.

Our focus here is on the *round-complexity* of resettably-sound zero-knowledge arguments. All the above protocols only require a constant number of rounds; but what is the exact round-complexity? The original BGGL protocol requires 8 rounds and collision-resistant hash functions (CRHs); an implementation in 6 rounds of the BGGL construction has been shown in [24]. More recently, Bitansky and Paneth in [6] improved the round complexity of resettably-sound zero knowledge to 4 rounds but additionally requiring the existence of a fully homomorphic encryption (FHE) schemes [13, 8]. Additionally they showed a 6-round protocol based on trapdoor permutations. In contrast, for "plain" (i.e., not resettably-sound) zero-knowledge, Bellare, Jakobsson and Yung [4] show how to obtain a 4-round zero-knowledge argument for NP based on the existence of the existence of one-way functions. This leaves open the question of whether round-efficient (namely 4-round) resettably-sound arguments can be based on weaker assumptions than FHE.

1.1 Our Results

We close the gap between resettably-sound and "plain" zero-knowledge arguments, demonstrating a 4-round resettably sound zero-knowledge argument (of knowledge) based solely on the existence of OWFs.

Theorem 1 (Informal). *Assume the existence of one-way functions. Then there exists a 4-round resettably-sound zero-knowledge argument of knowledge for every language in NP.*

Our starting point is the constant-round resettably-sound zero-knowledge argument for NP due to CPS. Our central contribution is a method for "collapsing" rounds in this protocol. A key feature of the CPS protocol is that, although the protocol consist of many rounds, the honest prover actually just sends commitments to 0 in all but *two* of these rounds. These "commitment to 0" preamble messages are only used by the simulator; roughly speaking, the simulator uses these message to come up with a "fake witness" that it can use in the remaining part of the protocol. On a very high-level, we show that all these preamble messages can be run in parallel, if appropriately adjusting the remaining two messages. An initial observation is that if we simply run all the preamble rounds in parallel—in a single "preamble slots"—then both completeness and zero-knowledge will still hold; the problem is that soundness no longer holds. In fact, soundness of the CPS protocol relies on the fact that the preamble messages are executed in sequence. Our key-idea for dealing with this issue is to have the verifier additionally provide a *signature* on the message-response pair for the "preamble" slot, and we now modify the "fake witness" part of the protocol to be a *chain of signatures* of the preamble messages *in the right order*. Soundness is now restored, and zero-knowledge simulation can be re-established by having the simulator *rewind* the preamble slot to get a signed sequence of messages in the right order.

1.2 Techniques

To explain our techniques in more detail, let us first recall Barak's non-black-box zero knowledge protocol on which BGGL is based, and then recall how CPS modify this protocol to only rely on OWF. We finally explain how to "collapse" rounds in this protocol.

Barak's Protocol and the BGGL transformation. Recall that Barak's protocol relies on the existence of a family of collision-resistant hash function $h : \{0,1\}^* \to \{0,1\}^n$; note that any such family of collision-resistant hash functions can be implemented from a family of collision-resistant hash functions mapping n-bit string into $n/2$-bit strings using *tree hashing* [21]. Roughly speaking, in Barak's protocol, on common input 1^n and $x \in \{0,1\}^{\mathtt{poly(n)}}$, the Prover P and Verifier V, proceed in two stages. In Stage 1, V starts by selecting a function h from a family of collision-resistant hash function and sends it to P; P next sends a commitment $c = \mathsf{Com}(0^n)$ of length n, and finally, V next sends a "challenge" $r \in \{0,1\}^{2n}$; we refer to this as the "commit-challenge" round. In Stage 2, P shows (using a witness indistinguishable argument of knowledge) that either x is true, or that c is a commitment to a "hash" (using h) of a program M (i.e., $c = \mathsf{Com}(h(M))$ such that $M(c) = r$.

Roughly speaking, soundness follows from the fact that even if a malicious prover P^* tries to commit to (the hash of) some program M (instead of committing to 0^n), with high probability, the string r sent by V will be different from $M(c)$ (since r is chosen independently of c). To prove ZK, consider the non-black-box simulator S that commits to a hash of the code of the malicious verifier V^*; note that, by definition, it thus holds that $M(c) = r$, and the simulator can use c as a "fake" witness in the final proof. To formalize this approach, the witness indistinguishable argument in Stage 2 must actually be a witness indistinguishable *universal argument* (WIUARG) [22, 1] since the statement that c is a commitment to a program M of *arbitrary* polynomial-size, and that proving $M(c) = r$ within some *arbitrary* polynomial time, is not in $\mathcal{NP}$. WIUARGs are known based on the existence of CRH and those protocols are constant-round public-coin; as a result, the whole protocol is constant-round and public-coin.

Finally, BGGL show that any constant-round public-coin zero-knowledge argument of knowledge can be transformed into a resettable-sound zero-knowledge argument, by simply having the verifier generate its (random) message by applying a pseudorandom function to the current partial transcript.[7]

The CPS Protocol. We now turn to recall the ideas from CPS for removing the use of CRHs in Barak's protocol. Note that hash functions are needed in two locations in Barak's protocol. First, since there is no *a-priori* polynomial upper-bound of the length of the code of V^*, we require the simulator to commit to the

[7] Strictly speaking, Barak's protocol is not a argument of knowledge, but rather a "weak" argument of knowledge (see [1, 2] for more details), but the transformation of [2] applies also to such protocol.

hash of the code of V^*. Secondly, since there is no *a-priori* polynomial upper-bound on the running-time of V^*, we require the use of universal arguments (and such constructions are only known based on the existence of collision-resistant hash functions).

The main idea of CPS is to notice that digital signature schemes—which can be constructed based on one-way functions—share many of the desirable properties of CRHs, and to show how to appropriately instantiate (a variant of) Barak's protocol using signature schemes instead of using CRHs. More precisely, CPS show that by relying on strong fixed-length signature schemes, which can be constructed based on one-way functions, one can construct *signature tree* analogous to the tree hashing that could be used to compress arbitrary length messages into a signature of length n and satisfies an analogue collision-resistance property. A strong fixed-length signature scheme allows signing messages of arbitrary polynomial-length (e.g length $2n$) using a length n signature and satisfies that no polynomial time attacker can obtain a *new* signature even for messages that it has seen a signature on [14].

CPS then show how to replace tree hashing by signature trees by appropriately modifying Barak's protocol. Firstly, CPS adds a signature slot at the beginning of the protocol. More precisely, in an initial stage of the protocol, the verifier generates a signature key-pair SK, VK and sends only the verification key VK to the prover. Next, in a "signature slot", the prover sends a commitment c of some message to the verifier, and the verifier computes and returns a valid signature σ of c (using SK). This is used by the simulator to construct a signature tree through rewinding the (malicious) verifier as a fake witness for WIUARG in an analogous way as before. Note that the commitment is used to hide the message to be signed from the malicious verifier, and as such, the signature tree is constituted by signatures of commitments of signatures...etc—this is referred to as a *Sig-com tree*. On the other hand, soundness follows in a similar way to Barak's protocol by relying on the fact that Sig-com tree satisfy a strong "collision-resistance" property—namely, no attacker getting the VK can find collisions, even given access to a signing oracle.

Secondly, CPS use a variant of Barak's protocol due to Pass and Rosen [26], which relies on a special-purpose WIUARG, in which the honest prover never needs to perform any hashing.[8] More precisely, the WIUARG consist two phases: a first phase where the honest provery simply sends commitments to 0^n, and a second phase where it proves that either $x \in L$ or the messages it committed to consistutes a valid UARG proving that the prover knows a fake witness.

While this protocol is not public-coin, CPS nevertheless shows that it suffices to apply the PRF transformation of BGGL to just the the public-coin part of the protocol to obtain a resettably soundness protocol; recall that the only part of the protocol that is not public-coin is the "signature slot" and, thus, intuitively, the only "advantages" a resetting prover gets is that it may rewind the signature slot, and thus get an arbitrary polynomial number of signatures on messages of its choice. But, as noted above, signature trees are collision-resistant even with

[8] In fact, an early version of Barak's protocol also had this property.

respect to an attacker that gets an arbitrary polynomial number of queries to a signing oracle and thus resettable-soundness follows in exactly the same way as the (non-resetting) soundness property.

Formalizing this intuition, however, is subtle. CPS first introduce an "oracle-aided" model where both players have access to a signing oracle, and construct a public-coin zero knowledge argument of knowledge in this model. Then the transformation of [2] is applied to this protocol to obtain an oracle-aided resettably-sound zero-knowledge argument of knowledge. CPS then show a general transformation for turning the protocol into a "fixed-input" resettably-sound zero-knowledge argument (of knowledge) in the "plain" model (i.e. without any oracle); fixed-input resettable-soundness means that resettable soundness is only required to hold with respect to a single fixed input. Finally, CPS show another general transformation that turns any fixed-input resettable soundness *argument of knowledge* into "full-fledged" resettable sound argument (or knowledge). Combining all these steps leads to constant-round resettably-sound zero-knowledge argument of knowledge for $\mathcal{NP}$ based on one-way functions.

Collapsing rounds for the CPS protocol. We are now ready to explain our method for collapsing rounds in the CPS protocol. Note that, although the CPS protocol consists of many rounds, the honest prover actually just sends commitments to 0, in all but the final two rounds, where the prover shows that it either has a "fake witness" or that $x \in L$. More precisely, in the final "proof phase" of the protocol (where the prover only sends two messages), the prover shows the verifier that either $x \in L$ or that the "committed UARG" transcript is accepting. The key idea is to modify the protocol to let the prover show in the "proof phase" that either $x \in L$ or it *knows* a "commit-challenge" pair (c, r) and a committed UARG transcript showing that the commit-challenge pair was successful. This, alone, clearly does not work: soundness no longer hold if the prover can come up with its own "invented transcript". Inspired by the work of Lin and Pass [20], we instead require the prover to show that it knows a transcript—that has been signed, message-by-message, by the verifier through a "signature-chain". A similar approach was used also in [11, 19]. Once we have done this change, we can simply remove all messages in the preamble phase (where the honest prover commits to 0) and just replace them with a signature slot. More precisely, we modify the CPS protocol in the following way:

- We start by running *two* signature slots in parallel: the first one is used for the signature-trees as in the original CPS protocol; the second one is used for the "signature-chain".
- In parallel with the signature slots, we start running the modified "proof phase" where the prover is requested to (using a WI argument of knowledge) prove that either $x \in L$ or it knows a "successful" transcript for the preamble phase that has been signed, message-by-message, in the right sequence using the second signature key.

Intuitively, simulation can be performed similarly to CPS, except that instead of simply providing the UARG messages in the protocol, the simulator rewinds the

signature slot to get an appropriately signed transcript of the UARG protocol. (Proving this is a bit delicate since the CPS simulator is already providing its own rewindings, so we need to be careful to ensure that the composition of these rewindings does not blow up the expected running-time.)

The key challenge, however, is proving resettable-soundness of the resulting protocol. On a very high-level, we shows that how to transform any resetting attacker to a "stand-alone" (i.e., non-resetting) attacker for oracle-aided CPS protocol (recall that the CPS protocol was first constructed in an oracle-model where the prover and verifier have access to signature oracles, and then the oracle-aided protocol was transformed into a protocol in the "plain" model by adding the the signature slots).[9] Roughly speaking, we show how to extract out the implicit transcript messages from any successful resetting prover and we can then use these messages in the (oracle-aided) CPS protocol. This is not entirely trivial, since in the CPS protocol these messages need to be provided one-by-one, whereas we can only extract out a full transcript. Our key technical contribution consist of showing how to appropriately rewind the resetting attacker to make it provide accepting transcript that are consistent with a current partial transcript of the CPS protocol. We here rely on the properties of signature-chains, and the fact the the protocol only has a constant number of rounds.

2 Definitions

We now give definitions for interactive proof/argument systems with all variants that are useful in this work.

Definition 1 (interactive proofs [17]). *A proof system for the language L, is a pair of interactive Turing machines (P, V) running on common input x such that:*

- *Efficiency: P and V are* PPT.
- *Completeness: There exists a negligible function $\nu(\cdot)$ such that for every pair (x, w) such that $R_L(x, w) = 1$,*

$$\text{Prob}[\ \langle P(w), V\rangle(x) = 1\ ] \geq 1 - \nu(|x|).$$

- *Soundness: For every $x \notin L$ and for every interactive Turing machine P^* there exists a negligible function $\nu(\cdot)$ such that*

$$\text{Prob}[\ \langle P^*, V\rangle(x) = 1\ ] < \nu(|x|).$$

In the above definition we can relax the soundness requirement by considering P^* as PPT. In this case, we say that (P, V) is an argument system.

We denote by $\texttt{view}^{P(w)}_{V^*(x,z)}$ the view (i.e., its private coins and the received messages) of V^* during an interaction with $P(w)$ on common input x and auxiliary input z.

[9] This is a slight oversimplification; we actually need to slightly modify the oracle-aided CPS protocol. See Section 3 for more details.

Definition 2 (zero-knowledge arguments [17]). *Let (P, V) be an interactive argument system for a language L. We say that (P, V) is zero knowledge (ZK) if, for any probabilistic polynomial-time adversary V^* receiving an auxiliary input z, there exists a probabilistic polynomial-time algorithm S_{V^*} such for all pairs $(x, w) \in R_L$ the ensembles $\{\mathtt{view}^{P(w)}_{V^*(x,z)}\}$ and $\{S_{V^*}(x, z)\}$ are computationally indistinguishable.*

Arguments of knowledge are arguments where there additionally exists an expected PPT *extractor* that can extract a witness from any successful prover, and this is a stronger notion of soundness. We will give now a definition that is slightly weaker than the standard definition of [3] but is useful for our constructions.

Note, also, that in the following definition, the extractor is given non-black box access to the prover. This is an essential property for our techniques.

Definition 3 (arguments of knowledge [2]). *Let R be a binary relation. We say that a probabilistic, polynomial-time interactive machine V is a* knowledge verifier *for the relation R with* negligible knowledge error *if the following two conditions hold:*

- *Non-triviality: There exists a probabilistic polynomial-time interactive machine P such that for every $(x, w) \in R$, all possible interactions of V with P on common input x, where P has auxiliary input w, are accepting, except with negligible probability.*
- *Validity (or knowledge soundness) with negligible error: There exists a probabilistic polynomial-time machine K such that for every probabilistic polynomial-time machine P^*, every polynomial $p(\cdot)$ and all sufficiently large x's, $Pr[w \leftarrow K(desc(P^*), x) \wedge R_L(x, w) = 1] > Pr[\langle P^*, V\rangle(x) = accept] - \frac{1}{p(|x|)}$ where $\langle P^*, V\rangle(x)$ denotes V 's output after interacting with P^* upon common input x and $desc(P^*)$ denotes the description of P^*'s strategy.*

Further, (P, V) is an argument of knowledge for relation R.

Definition 4 (witness indistinguishability [12]). *Let L be a language in $\mathcal{NP}$ and R_L be the corresponding relation. An interactive argument (P, V) for L is* witness indistinguishable *(WI) if for every verifier V^*, every pair (w_0, w_1) such that $(x, w_0) \in R_L$ and $(x, w_1) \in R_L$ and every auxiliary input z, the following ensembles are computationally indistinguishable:*

$$\{\mathtt{view}^{P(w_0)}_{V^*(x,z)}\} \quad \textit{and} \quad \{\mathtt{view}^{P(w_1)}_{V^*(x,z)}\}.$$

2.1 Resettably-Sound Proofs

A polynomial-time relation R is a relation for which it is possible to verify in time polynomial in $|x|$ whether $R(x, w) = 1$. Let us consider an $\mathcal{NP}$-language L and denote by R_L the corresponding polynomial-time relation such that $x \in L$ if and only if there exists w such that $R_L(x, w) = 1$. We will call such a w a

valid witness for $x \in L$. A *negligible* function $\nu(k)$ is a non-negative function such that for any constant $c < 0$ and for all sufficiently large k, $\nu(k) < k^c$. We will denote by $\text{Prob}_r[\ X\ ]$ the probability of an event X over coins r. The abbreviation "PPT" stands for probabilistic polynomial time. We will use the standard notion of computational indistinguishability [16].

Let us recall the definition of resettable soundness due to [2].

Definition 5 (resettably-sound arguments [2]). *A resetting attack of a cheating prover* P^* *on a resettable verifier* V *is defined by the following two-step random process, indexed by a security parameter* k*.*

1. *Uniformly select and fix* $t = \texttt{poly(k)}$ *random-tapes, denoted* $r_1, \ldots, r_t$*, for* V*, resulting in deterministic strategies* $V^{(j)}(x) = V_{x,r_j}$ *defined by* $V_{x,r_j}(\alpha) = V(x, r_j, \alpha)$,[10] *where* $x \in \{0,1\}^k$ *and* $j \in [t]$*. Each* $V^{(j)}(x)$ *is called an incarnation of* V *.*
2. *On input* 1^k*, machine* P^* *is allowed to initiate* $\texttt{poly(k)}$*-many interactions with the* $V^{(j)}(x)$*'s. The activity of* P^* *proceeds in rounds. In each round* P^* *chooses* $x \in \{0,1\}^k$ *and* $j \in [t]$*, thus defining* $V^{(j)}(x)$*, and conducts a complete session with it.*

Let (P, V) *be an interactive argument for a language* L*. We say that* (P, V) *is a* resettably-sound argument *for* L *if the following condition holds:*

– Resettable-soundness*: For every polynomial-size resetting attack, the probability that in some session the corresponding* $V^{(j)}(x)$ *has accepted and* $x \notin L$ *is negligible.*

We will also consider a slight weakening of the notion of resettable soundness, where the statement to be proven is fixed, and the verifier uses a single random tape (that is, the prover cannot start many independent instances of the verifier).

Definition 6 (fixed-input resettably-sound arguments [27]). *An interactive argument* (P, V) *for a* $\mathcal{NP}$ *language* L *with witness relation* R_L *is* fixed-input resettably-sound *if it satisfies the following property: For all non-uniform polynomial-time adversarial resetting prover* P^**, there exists a negligible function* $\mu(\cdot)$ *such that for every all* $x \notin L$*,*

$$\Pr[R \leftarrow \{0,1\}^\infty; (P^{*V_R(x)}, V_R)(x) = 1] \leq \mu(|x|)$$

The following theorem was proved in [10]

Claim 1 *Let* (P, V) *be a fixed-input resettably sound zero-knowledge (resp. witness indistinguishable) argument of knowledge for a language* $L \in \mathcal{NP}$ *. Then there exists a protocol* (P', V') *that is a (full-fledged) resettably-sound zero-knowledge (resp. witness indistinguishable) argument of knowledge for* L*.*

As a result, in the sequel, we only focus on proving fixed-input resettable-soundness.

[10] Here, $V(x, r, \alpha)$ denotes the message sent by the strategy V on common input x, random-tape r, after seeing the message-sequence α.

2.2 Commitment Schemes

We now give a definition for a commitment scheme. For readability we will use "for all m" to mean any possible message m of length polynomial in the security parameter.

Definition 7. $(\mathsf{Gen}, \mathsf{Com}, \mathsf{Ver})$ *is a* commitment scheme *if:*

- **efficiency:** Gen*,* Com *and* Ver *are polynomial-time algorithms;*
- **completeness:** *for all* m *it holds that* $\Pr[h_{com} \leftarrow \mathsf{Gen}(1^n); (\mathrm{COM}, \mathsf{dec}) \leftarrow \mathsf{Com}(h_{com}, m) : \mathsf{Ver}(h_{com}, \mathrm{COM}, \mathsf{dec}, m) = 1] = 1$;
- **binding:** *for any polynomial-time algorithm* $\mathtt{committer}^*$ *there is a negligible function* ν *such that for all sufficiently large* k *it holds that:*
  $\Pr[h_{com} \leftarrow \mathsf{Gen}(1^n); (\mathrm{COM}, m_0, m_1, \mathsf{dec}_0, \mathsf{dec}_1) \leftarrow \mathtt{committer}^*(h_{com}) :$
  $m_0 \neq m_1$ and $\mathsf{Ver}(h_{com}, \mathrm{COM}, \mathsf{dec}_0, m_0) = \mathsf{Ver}(h_{com}, \mathrm{COM}, \mathsf{dec}_1, m_1) = 1] \leq \nu(k)$;
- **hiding:** *for any algorithm polynomial-time* $\mathtt{receiver}^*$ *there is a negligible function* ν *such that for all* m_0, m_1 *where* $|m_0| = |m_1|$ *and all sufficiently large* k *it holds that*

$$\Pr\left[(h_{com}, \mathsf{aux}) \leftarrow \mathtt{receiver}(1^n); b \leftarrow \{0,1\}; (\mathrm{COM}, \mathsf{dec}) \leftarrow \mathsf{Com}(h_{com}, m_b) : b \leftarrow \mathtt{receiver}^*(\mathrm{COM}, \mathsf{aux})\right] \leq \frac{1}{2} + \nu(n)$$

When h_{com} *is clear from context, we often say "*m, dec *is a valid opening for* COM*" to mean that* $\mathsf{Ver}(h_{com}, \mathrm{COM}, \mathsf{dec}, m) = 1$.

Collision-resistant hash functions. We will use hash functions as defined below.

Definition 8. *Let* $\mathcal{H} = \{h_\alpha\}$ *be an efficiently sampleable hash function ensemble where* $h_\alpha : \{0,1\}^* \to \{0,1\}^\alpha$*. We say that* $\mathcal{H}$ *is* collision-resistant against polynomial size circuits *if for every (non-uniform) polynomial-size circuit family* $\{A_n\}_{n \in N}$*, for all positive constants* c*, and all sufficiently large* k*, it holds that*

$$\mathrm{Prob}[\ \alpha \stackrel{R}{\to} \{0,1\}^k : A_n(\alpha) = (x, x') \wedge h_\alpha(x) = h_\alpha(x')\ ] < n^{-c}.$$

2.3 Signature Trees

Constructions of universal arguments (defined later) rely on Merkle-trees and collision-resistant hash-functions to be able to commit to a program of arbitrary polynomial length where no apriori-bound is known. In [10], they construct an analog to Merkle-trees, called signature trees, while relying only on one-way functions. Below, we recall definitions from [10]. Some of the text in this section, is copied verbatim from [10]

Definition 9 (Strong Signatures). *A* strong, length-ℓ, signature scheme SIG *is a triple* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *of* PPT *algorithms, such that*

1. *for all* $n \in \mathbf{N}, m \in \{0,1\}^*$,

$$\Pr[(\mathrm{SK}, \mathrm{VK}) \leftarrow \mathsf{Gen}(1^n), \sigma \leftarrow \mathsf{Sign}_{\mathrm{SK}}(m); \mathsf{Ver}_{\mathrm{VK}}(m, \sigma) = 1 \wedge |\sigma| = \ell(n)] = 1$$

2. *for every non-uniform* PPT *adversary* A*, there exists a negligible function* $\mu(\cdot)$ *such that*

$$\Pr\Big[(\mathrm{SK}, \mathrm{VK}) \leftarrow \mathsf{Gen}(1^n), (m, \sigma) \leftarrow A^{\mathsf{Sign}_{\mathrm{SK}}(\cdot)}(1^n) : \mathsf{Ver}_{\mathrm{VK}}(m, \sigma) = 1 \wedge (m, \sigma) \notin L\Big] \leq \mu(n),$$

where L *denotes the list of query-answer pairs of* A*'s queries to its oracle.*

Strong, length-ℓ, deterministic signature schemes with $\ell(n) = n$ are known based on the existence of OWFs; see [23, 28, 14] for further details.

Definition 10 (Signature Trees). *Let* $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *be a strong, length-n signature scheme. Let* $(\mathrm{SK}, \mathrm{VK})$ *be a key-pair of* SIG*, and* s *be a string of length* 2^d*. A* signature tree *of the string* s *w.r.t.* $(\mathrm{SK}, \mathrm{VK})$ *is a complete binary tree of depth* d*, defined as follows.*

- *A leaf* l_γ *indexed by* $\gamma \in \{0,1\}^d$ *is set as the bit at position* γ *in* s*.*
- *An internal node* l_γ *indexed by* $\gamma \in \bigcup_{i=0}^{d-1}\{0,1\}^i$ *satisfies that* $\mathsf{Ver}_{\mathrm{VK}}((l_{\gamma 0}, l_{\gamma 1}), l_\gamma) = 1$.

To *verify* whether a Γ is a valid signature-tree of a string s w.r.t. the signature scheme SIG and the key-pair $(\mathrm{SK}, \mathrm{VK})$ knowledge of the secret key SK is not needed. However, to *create* a signature-tree for a string s, the secret key SK is needed.

Definition 11 (Signature Path). *A* signature path *w.r.t.* SIG, VK *and a root* l_λ *for a bit* b *at leaf* $\gamma \in \{0,1\}^d$ *is a vector* $\boldsymbol{\rho} = ((l_0, l_1), ((l_{\gamma_{\leq 1}0}, l_{\gamma_{\leq 1}1}), \ldots (l_{\gamma_{\leq d-1}0}, l_{\gamma_{\leq d-1}1}))$ *such that for every* $i \in \{0, \ldots, d-1\}$, $\mathsf{Ver}_{\mathrm{VK}}((l_{\gamma_{\leq i}0}, l_{\gamma_{\leq i}1}), l_{\gamma_{\leq i}}) = 1$.

Let $\mathsf{PATH}^{\mathsf{SIG}}(\boldsymbol{\rho}, b, \gamma, l_\lambda, \mathrm{VK}) = 1$ *if* ρ *is a signature path w.r.t.* SIG, VK, l_λ *for* b *at* γ.

2.4 Sig-Com Schemes

Definition 12 (Sig-Com Schemes). *Let* $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *be a strong, length-n, signature scheme, and let* COM *be a non-interactive commitment schemes. Define* $\mathsf{SIG}' = (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Ver}')$ *to be a triple of* PPT *machines defined as follows:*

- $\mathsf{Gen}' = \mathsf{Gen}$.
- $\mathsf{Sign}'_{\mathrm{SK}}(m)$: *compute a commitment* $c = \mathrm{COM}(m; \tau)$ *using a uniformly selected* τ*, and let* $\sigma = \mathsf{Sign}_{\mathrm{SK}}(c)$*; output* (σ, τ)
- $\mathsf{Ver}'_{\mathrm{VK}}(m, \sigma, \tau)$: *Output* 1 *iff* $\mathsf{Ver}_{\mathrm{VK}}(\mathrm{COM}(m, \tau), \sigma) = 1$.

We call SIG' *the* Sig-Com Scheme *corresponding to* SIG *and* COM.

Definition 13 (Sig-Com Trees). *Let* $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathbf{SHVer}_{h_{com}})$ *be a strong, length-n signature scheme, let* COM *be a non-interactive commitment and let* $\mathsf{SIG}' = (\mathsf{Gen}', \mathsf{Sign}', \mathbf{SHVer}'_{h_{com}})$ *be the sig-com scheme corresponding to* SIG *and* COM. *Let* (SK, VK) *be a key-pair of* SIG', *and* s *be a string of length* 2^d. *A* signature tree *of the string* s *w.r.t.* (SK, VK) *is a complete binary tree of depth* d, *defined as follows.*

- *A leaf* l_γ *indexed by* $\gamma \in \{0,1\}^d$ *is set as the bit at position* γ *in* s.
- *An internal node* l_γ *indexed by* $\gamma \in \bigcup_{i=0}^{d-1}\{0,1\}^i$ *satisfies that there exists some* τ_γ *such that* $\mathsf{Ver}'_{\text{VK}}((l_{\gamma 0}, l_{\gamma 1}), l_\gamma, \tau_\gamma) = 1$.

Definition 14 (Sig-Com Path). *Let* $\mathsf{SIG}' = (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Ver}')$ *be a sig-com scheme. A* sig-com path *w.r.t.* SIG', VK *and a root* l_λ *for a bit* b *at leaf* $\gamma \in \{0,1\}^d$ *is a vector* $\boldsymbol{\rho} = ((l_0, l_1, \tau_\lambda), ((l_{\gamma_{\leq 1}0}, l_{\gamma_{\leq 1}1}, \tau_{\gamma_{\leq 1}}), \ldots, (l_{\gamma_{\leq d-1}0}, l_{\gamma_{\leq d-1}0}, \tau_{\gamma_{\leq d-1}})$ *such that for every* $i \in \{0, \ldots, d-1\}$, $\mathsf{Ver}'_{\text{VK}}((l_{\gamma_{\leq i}0}, l_{\gamma_{\leq i}1}), l_{\gamma_{\leq i}}, \tau_{\gamma_{\leq i}})) = 1$. *Let* $\mathsf{PATH}^{\mathsf{SIG}'}(\boldsymbol{\rho}, b, \gamma, l_\lambda, \text{VK}) = 1$ *if* $\boldsymbol{\rho}$ *is a signature path w.r.t.* SIG', VK, l_λ *for* b *at* γ.

2.5 Oracle-Aided Zero Knowledge Protocols

In this section we recall definitions of oracle-aided protocols from [10].

Let $\mathcal{O}$ be a probabilistic algorithm that on input a security parameter n, outputs a polynomial-length (in n) public-parameter pp, as well as the description of an oracle O. The oracle-aided execution of an interactive protocol with common input x between a prover P with auxiliary input y and a verifier V consist of first generating $\mathsf{pp}, O \leftarrow \mathcal{O}(1^{|x|})$ and then letting $P^O(x, y, \mathsf{pp})$ interact with $V(x, \mathsf{pp})$.

Definition 15 (Oracle-aided Interactive Arguments). *A pair of oracle algorithms* (P, V) *is an* $\mathcal{O}$-oracle aided argument *for a* $\mathcal{NP}$ *language* L *with witness relation* R_L *if it satisfies the following properties:*

- *Completeness: There exists a negligible function* $\mu(\cdot)$, *such that for all* $x \in L$, *if* $w \in R_L(x)$,

$$\Pr[\mathsf{pp}, O \leftarrow \mathcal{O}(1^{|x|}); (P^O(w), V)(x, \mathsf{pp}) = 1] = 1 - \mu(|x|)$$

- *Soundness: For all non-uniform polynomial-time adversarial prover* P^*, *there exists a negligible function* $\mu(\cdot)$ *such that for every all* $x \notin L$,

$$\Pr[\mathsf{pp}, O \leftarrow \mathcal{O}(1^{|x|}); (P^{*O}, V)(x, \mathsf{pp}) = 1] \leq \mu(|x|)$$

Additionally, if the following condition holds, (P, V) *is an* $\mathcal{O}$-oracle aided argument of knowledge*:*

- *Argument of knowledge: There exists a expected* PPT *algorithm* E *such that for every polynomial-size* P^*, *there exists a negligible function* $\mu(\cdot)$ *such that for every* x,

$$\begin{aligned}&\Pr[\mathsf{pp}, O \leftarrow \mathcal{O}(1^{|x|}); w \leftarrow E^{P^{*O}(x,\mathsf{pp})}(x, \mathsf{pp}); w \in R_L(x)] \\ &\geq \Pr[\mathsf{pp}, O \leftarrow \mathcal{O}(1^{|x|}); (P^{*O}, V)(x, \mathsf{pp}) = 1] - \mu(|x|)\end{aligned}$$

Definition 16 (Oracle-aided Resettably-sound Interactive Arguments). *An $\mathcal{O}$-oracle aided resetting attack of a cheating prover P^* on a resettable verifier V is defined by the following three-step random process, indexed by a security parameter n.*

1. *An initial setup where a public parameter and an oracle are generated:* $\mathsf{pp}, O \leftarrow \mathcal{O}(1^n)$. *$P^*$ is given* pp *and oracle access to O.*
2. *Uniformly select and fix $t = poly(n)$ random-tapes, denoted $r_1, \ldots, r_t$, for V, resulting in deterministic strategies $V^{(j)}(x) = V_{x,r_j}$ defined by $V_{x,r_j}(\alpha) = V(x, r_j, \alpha)$, where $x \in \{0,1\}^n$ and $j \in [t]$. Each $V^{(j)}(x)$ is called an incarnation of V .*
3. *On input 1^n, machine P^* is allowed to initiate $poly(n)$-many interactions with the $V^{(j)}(x)$'s. The activity of P^* proceeds in rounds. In each round P^* chooses $x \in \{0,1\}^n$ and $j \in [t]$, thus defining $V^{(j)}(x)$, and conducts a complete session with it.*

Let (P, V) be an $\mathcal{O}$-oracle aided interactive argument for a language L. We say that (P, V) is an $\mathcal{O}$-oracle aided resettably-sound argument *for L if the following condition holds:*

- $\mathcal{O}$-oracle aided resettable soundness*: For every polynomial-size resetting attack, the probability that in some session the corresponding $V^{(j)}(x)$ has accepted and $x \notin L$ is negligible.*

Oracle-aided Universal Arguments Universal arguments (introduced in [1] and closely related to CS-proofs [22]) are used in order to provide "efficient" proofs to statements of the form $y = (M, x, t)$, where y is considered to be a true statement if M is a non-deterministic machine that accepts x within t steps. The corresponding language and witness relation are denoted $L_{\mathcal{U}}$ and $\mathbf{R}_{\mathcal{U}}$ respectively, where the pair $((M, x, t), w)$ is in $\mathbf{R}_{\mathcal{U}}$ if M (viewed here as a two-input deterministic machine) accepts the pair (x, w) within t steps. Notice that every language in $\mathcal{NP}$ is linear time reducible to $L_{\mathcal{U}}$. Thus, a proof system for $L_{\mathcal{U}}$ allows us to handle all $\mathcal{NP}$-statements. In fact, a proof system for $L_{\mathcal{U}}$ enables us to handle languages that are beyond $\mathcal{NP}$, as the language $L_{\mathcal{U}}$ is $\mathcal{NE}$-complete (hence the name universal arguments).[11]

Definition 17 (Oracle-aided Universal Argument). *An oracle-aided protocol (P, V) is called an* $\mathcal{O}$-oracle-aided universal argument *system if it satisfies the following properties:*

- Efficient verification: *There exists a polynomial p such that for any $y = (M, x, t)$, and for any pp, O generated by $\mathcal{O}$, the total time spent by the (probabilistic) verifier strategy V, on common input y, pp, is at most $p(|y| + |\mathsf{pp}|)$. In particular, all messages exchanged in the protocol have length smaller than $p(|y| + |\mathsf{pp}|)$.*

[11] Furthermore, every language in $\mathcal{NEXP}$ is polynomial-time (but not linear-time) reducible to $L_{\mathcal{U}}$.

– Completeness with a relatively efficient oracle-aided prover: *For every* $(y = (M, x, t), w)$ *in* $\mathbf{R}_{\mathcal{U}}$,

$$\Pr[\mathsf{pp}, O \leftarrow \mathcal{O}(1^{|y|}); (P^O(w), V)(y, \mathsf{pp}) = 1] = 1.$$

Furthermore, there exists a polynomial q *such that the total time spent by* $P^O(w)$*, on common input* $y = (M, x, t)$*,* pp*, is at most* $q(T_M(x, w) + |\mathsf{pp}|) \leq q(t + |\mathsf{pp}|)$*, where* $T_M(x, w)$ *denotes the running time of* M *on input* (x, w)*.*

– Weak proof of knowledge for adaptively chosen statements: *For every polynomial* p *there exists a polynomial* p' *and a probabilistic polynomial-time oracle machine* E *such that the following holds: for every non-uniform polynomial-time oracle algorithm* P^**, if*

$$\Pr[\mathsf{pp}, O \leftarrow \mathcal{O}(1^n); R \leftarrow \{0,1\}^\infty; y \leftarrow P_R^{*O}(\mathsf{pp}) :$$

$$(P_R^{*O}(\mathsf{pp}), V(y, \mathsf{pp})) = 1] > 1/p(n)$$

then

$$\Pr[\mathsf{pp}, O \leftarrow \mathcal{O}(1^n); R, r \leftarrow \{0,1\}^\infty; y \leftarrow P_R^{*O}(\mathsf{pp}) : \exists w = w_1, \ldots w_t \in \mathbf{R}_{\mathcal{U}}(y)$$

$$s.t.\ \forall i \in [t], E_r^{P_R^{*O}}(\mathsf{pp}, y, i) = w_i] > \frac{1}{p'(n)}$$

where $\mathbf{R}_{\mathcal{U}}(y) \stackrel{\text{def}}{=} \{w : (y, w) \in \mathbf{R}_{\mathcal{U}}\}$.

Let SIG' be a canonical sig-com scheme with $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ and COM being its underlying signature scheme and commitment scheme.

Definition 18 (Signature Oracle). *Given* $\mathsf{SIG} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *a signature scheme , we define a signature oracle* $\mathcal{O}^{\mathsf{SIG}}$ *as follows: On input a security parameter* n*,* $\mathcal{O}^{\mathsf{SIG}}(1^n)$ *generates* $(\text{VK}, \text{SK}) \leftarrow \mathsf{Gen}(1^n)$ *and lets* $\mathsf{pp} = \text{VK}$ *and* $O(m) = \mathsf{Sign}_{\text{SK}}(m)$ *for every* $m \in \{0,1\}^{\mathtt{poly(n)}}$*.*

Definition 19 (Valid Sig-com Oracle). *An oracle* $\mathcal{O}'$ *is a* valid (SIG', ℓ) oracle *if there is a negligible* $\mu(\cdot)$ *such that for every* $n \in N$*, the following holds with probability* $1 - \mu(n)$ *over* $\mathsf{pp}, O \leftarrow \mathcal{O}'(1^n)$*: for every* $m \in \{0,1\}^{\ell(n)}$*,* $O(m)$ *returns* (σ, τ) *such that* $\mathsf{Ver}'_{\text{VK}}(m, \sigma, \tau) = 1$ *with probability at least* $1 - \mu(n)$*.*

Definition 20. *An* $\mathcal{O}^{\mathsf{SIG}}$*-aided universal arg.* (P, V) *has* (SIG', ℓ)-completeness *if there exists a prover* P' *such that the completeness condition holds for* (P', V) *when the oracle* $\mathcal{O}^{\mathsf{SIG}}$ *is replaced by any valid* (SIG', ℓ) *oracle* $\mathcal{O}'$*.*

The following theorem was proved in [10] (relying on Barak and Goldreich [1])

Theorem 2. *Let* SIG' *be a canonical sig-com scheme with* SIG *and* COM *being its underlying signature scheme and commitment scheme. Then there exists a* (SIG', ℓ)*-complete* $\mathcal{O}^{\mathsf{SIG}}$*-aided universal argument with* $\ell(n) = 2n$*.*

Common Input: An instance x of a language $L \in \mathcal{NP}$ with witness relation $\mathbf{R}_L$.
Auxiliary input to P: A witness w such that $(x, w) \in \mathbf{R}_L$.
Primitives Used: A canonical sig-com scheme SIG' with SIG and COM as the underlying signature and commitment schemes, and a (SIG', ℓ)-complete $\mathcal{O}^{\mathsf{SIG}}$-aided universal argument $(P_{\mathtt{UA}}, V_{\mathtt{UA}})$ with $\ell(n) = 2n$.
Set Up: Run $(\mathsf{pp}, O) \leftarrow \mathcal{O}^{\mathsf{SIG}}(1^n)$, add pp to common input for P and V. Furthermore, allow P oracle access to O.
Stage One (Trapdoor—Commit-Challenge):
P_1: Send $c_0 = \text{COM}(0^{2n}, \tau_0)$ to V with uniform τ_0
V_1: Send $r \xleftarrow{\$} \{0,1\}^{4n}$ to P
Stage Two (Encrypted Universal Argument):
P_2: Send $c_1 = \text{COM}(0^{2n}, \tau_1)$ for uniformly selected τ_1
V_3: Send r', uniformly chosen random tape for $V_{\mathtt{UA}}$
P_3: Send $c_2 = \text{COM}(0^k, \tau_2)$ for uniformly selected τ_2, where k is the length of $P_{\mathtt{UA}}$'s second message.
Stage Three: (Main Proof)
$P \Leftrightarrow V$: A WI-AOK $\langle P_{\mathsf{WI}}, V_{\mathsf{WI}} \rangle$ proving the OR of the following statements:
1. $\exists\ w \in \{0,1\}^{\mathtt{poly}(|\mathtt{x}|)}$ s.t. $(x, w) \in \mathbf{R}_L$.
2. $\exists\ \langle p_1, p_2, \tau_1, \tau_2 \rangle$ s.t. $(\langle c_0, r, c_1, c_2, r', \mathsf{pp} \rangle, \langle p_1, p_2, \tau_1, \tau_2 \rangle) \in \mathbf{R}_{L_2}$ (defined in Fig. 2).

Fig. 1. $\mathcal{O}^{\mathsf{SIG}}$-aided ZK Argument of Knowledge.

3 A Variant of the Signature Oracle-aided ZK Protocol from CPS

In this section, we provide a formal protocol description and theorem statement for a slight variant of the CPS protocol in a signature oracle-aided model. We will show in the next section how to collapse rounds of this protocol, and prove resettable soundness of the collapsed protocol by reducing the resetting attacker to a stand-alone (i.e., non-resetting) adversary that breaks soundness of this protocol.

We refer the readers to Section 1.2 for the ideas and intuition behind the CPS protocol. A formal description of the protocol can be found in Figure 1 and 2, where we make a slight modification to the language proved in the UA where we require the committed program either output the string r when fed a commitment to its own description or output r as the second component of 4-tuple output when fed by a string of length shorter than r. This modification is inconsequential to the soundness property of the protocol, but will be useful for us to prove soundness of the collapsed protocol in the next section. The following theorem follows by [10].

Theorem 3. *Assume the existence of one-way functions. The protocol defined in Figure 1 and 2 is a signature oracle-aided zero-knowledge argument of knowledge for NP.*

Relation 1: Let SIG' a sig-com scheme, with underlying signature scheme SIG and commitment scheme COM. Let $\mathtt{ECC}$ be a binary error-correcting code with constant min-distance and efficient encoding algorithm. We say that $\langle c_0, r, \mathsf{pp}\rangle \in L_1$ if $\exists\langle \tau_0, d, l_\lambda, C, \{\boldsymbol{\rho_i}\}_{i\in[2^d]}, y\rangle$ such that

- $c_0 = \text{COM}((d, l_\lambda), \tau_0)$
- (d, l_λ) are the depth and root of a sig-com tree for C w.r.t. pp
- Each $\boldsymbol{\rho_i}$ is a valid sig-com path for leaf i of this sig-com tree. That is, $\mathsf{PATH}^{\mathsf{SIG}'}(\boldsymbol{\rho_i}, C_i, i, l_\lambda, \mathsf{pp}) = 1$ for each i.
- $C = \mathtt{ECC}(\Pi)$ for some circuit Π
- $\Pi(c_0) = r$ or $|y| < 2n$ and $\Pi(y) = (s_1, r, s_2, s_3)$ for some strings s_1, s_2, s_3 and appropriate encoding of the 4-tuple.

We let $\mathbf{R}_{L_1}$ be the witness relation corresponding to L_1.

Relation 2: Let L_1 be described as above, with respect to SIG' and $\mathtt{ECC}$. Let $(P_{\mathtt{UA}}, V_{\mathtt{UA}})$ be a (SIG', ℓ)-complete $\mathcal{O}^{\mathsf{SIG}}$-aided universal argument with $\ell(n) = 2n$. We say that $\langle c_0, r, c_1, c_2, r', \mathsf{pp}\rangle \in L_2$ if $\exists\langle p_1, p_2, \tau_1, \tau_2\rangle$ such that

- $c_1 = \text{COM}(p_1, \tau_1)$, $c_2 = \text{COM}(p_2, \tau_2)$.
- (p_1, r', p_2) constitutes an accepting $(P_{\mathtt{UA}}, V_{\mathtt{UA}})$ transcript for $\langle c_0, r, \mathsf{pp}\rangle \in L_1$.

We let $\mathbf{R}_{L_2}$ be the witness relation corresponding to L_2.

Fig. 2. Relations used in the $\mathcal{O}^{\mathsf{SIG}}$-aided ZK protocol in Fig. 1.

4 4-Round Resettably-Sound Zero Knowledge

We are now ready to describe our 4-round protocol. Our protocol relies on Blum's 4-round Hamiltonicity WI-AOK, $\langle P_{\mathsf{WI}}, V_{\mathsf{WI}}\rangle$ [7]. Our protocol is obtained by first constructing a "basic" protocol where the verifier uses "fresh" randomness in each round, and then applying the BGGL transformation to this protocol (i.e., having the verifier pick its randomness by applying a PRF to the current transcript). The "basic" protocol proceeds as follows.

1. The verifier V picks two signature key pairs (vk, sk) and (vk', sk') using $\mathsf{Gen}(1^k)$. V also generates the first message $\mathtt{BH}_1$ for the WI AoK.
   The language considered for the WI argument of knowledge is identical to one used in the protocol presented in the previous section, i.e. $(x, vk, vk') \in L^*$ iff
   (a) $\exists\ w \in \{0,1\}^{\mathtt{poly}(|\mathtt{x}|)}$ s.t. $(x, w) \in \mathbf{R}_L$.
   (b) $\exists\ \langle c_0, r, c_1', r', p_1, p_2, \tau_1, \tau_2, \sigma_1, \sigma_2\rangle$ s.t. $(\langle vk, vk'\rangle, \langle c_0, r, c_1, r', p_1, p_2, \tau_1, \sigma_1, \sigma_2\rangle) \in \mathbf{R}_{L_3}$ (defined in Fig. 4).
   V sends $vk, vk', \mathtt{BH}_1$ to the prover P.
2. P responds with a commitment c to the all 0's string of length k and the second message for the WI AoK, $\mathtt{BH}_2$.
3. V sends $r, \sigma, \sigma', \mathtt{BH}_3$ to the prover where $r \leftarrow \{0,1\}^{3k}$, σ and σ' are signatures of messages $c|r$ and c under keys sk and sk' respectively and $\mathtt{BH}_3$ is the third message of WI AoK .

4. P finally sends $\mathtt{BH}_4$, the fourth message of the WI AoK. The verifier accepts if the transcript $(\mathtt{BH}_1, \mathtt{BH}_2, \mathtt{BH}_3, \mathtt{BH}_4)$ is accepting for $(x, h, vk) \in L^*$.

We finally modify the basic protocol by having the verifier first pick a random seed s for a PRF f and then, at each round, generating the randomness it needs by applying the f_s to the current transcript.

A formal description of the protocol is presented in Figure 3.

Common Input: An instance x of a language $L \in \mathcal{NP}$ with witness relation $\mathbf{R}_L$.
Auxiliary input to P: A witness w such that $(x, w) \in \mathbf{R}_L$.
$_1$: $V \to P$: Send $\mathtt{BH}_1, vk, vk'$ where $(vk, sk) \leftarrow \mathsf{Gen}(1^n)$ and $(vk', sk') \leftarrow \mathsf{Gen}(1^n)$.
$_2$: $P \to V$: Send $\mathtt{BH}_2, c = \text{COM}(0^{2n}, \tau)$ for a randomly chosen τ.
$_3$: $V \to P$: Send $\mathtt{BH}_3, r, \sigma, \sigma'$ where $r \leftarrow \{0,1\}^{3n}$, $\sigma \leftarrow \text{SIGN}(sk, c|r)$ and $\sigma' \leftarrow \text{SIGN}(sk', c)$.
$_4$: $P \to V$: Send $\mathtt{BH}_4$.

We finally modify the above protocol by having the verifier first pick a random seed s for a PRF f and then, at each round, generating the randomness it needs by applying the f_s to the current transcript.

Fig. 3. Our 4-round rsZK Argument of Knowledge $\pi = (P, V)$.

Relation 3: Let L_1 be as described in Fig 2, with respect to SIG' and $\mathtt{ECC}$. Let $(P_{\mathtt{UA}}, V_{\mathtt{UA}})$ be a (SIG', ℓ)-complete $\mathcal{O}^{\mathsf{SIG}}$-aided universal argument with $\ell(n) = 2n$. We say that $\langle vk, vk' \rangle \in \mathbf{R}_{L_3}$, if $\exists\ \langle c_0, r, c_1, r', p_1, p_2, \tau_1, \sigma_1, \sigma_2 \rangle) \in \mathbf{R}_{L_3}$ such that

- $\mathsf{Ver}_{vk}(c_0|r, \sigma_1) = 1$,$c_1 = \text{COM}(p_1|\sigma_1, \tau_1)$ and $\mathsf{Ver}_{vk}(c_1|r', \sigma_2) = 1$.
- (p_1, r', p_2) constitutes an accepting $(P_{\mathtt{UA}}, V_{\mathtt{UA}})$ transcript for $\langle c_0, r, vk' \rangle \in L_1$.

We let $\mathbf{R}_{L_3}$ be the witness relation corresponding to L_3.

Fig. 4. Relations used in the protocol in Fig. 3.

Theorem 4. *Assume the existence of OWFs, then protocol in Fig. 3 is a 4-round resettably sound zero knowledge argument of knowledge.*

Proof. We prove completeness and resettable-soundness of the protocol. As proved in [10], it suffices to prove fixed-input resettable-soundness.

Completeness. Completeness of $\langle P, V \rangle$ follows directly from the completeness of the WI-AOK protocol.

Soundness. To prove the fixed-input resettable-soundness of $\langle P, V\rangle$, we show how to convert a malicious prover P^* for $\langle P, V\rangle$ into an *oracle-aided* malicious prover B that violates the *stand-alone* soundness of $\langle P_{\mathtt{zk}}, V_{\mathtt{zk}}\rangle$ (from the previous section).

First, we consider the experiment $\text{HYB}_1^A(n, z)$ where we run an adversary A on input (n, z) by supplying the messages of an honest verifier, with the exception that the verifier challenges, i.e. r and $\mathtt{BH}_3$ in the third message are chosen uniformly at random even in the rewindings instead of applying the PRF. Upon completion, we run the extractor of the WI AoK in a random session to obtain witness w. If this witness is not a real witness, output the transcript along with w. Otherwise output $\perp$.

From the pseudo-randomness of $\mathcal{F}$, we know that if P^* convinces an honest verifier of a false statement with non-negligible probability in the original experiment, then it will succeed in proving a false statement with non-negligible probability in HYB_1 as well. Since there are only polynomially many sessions, $\text{HYB}_1^{P^*}(n, z)$ outputs the second (or fake) witness with non-negligible probability.

More precisely, for a statement $(x, vk, vk') \in L^*$ the fake witness contains $\langle c_0, r, c_1', r', p_1, p_2, \tau_1, \sigma_1, \sigma_2\rangle$. From the unforgeability of the signature scheme under verifier key vk, it follows that, if P^* proves using the fake witness, then P^* must have obtained σ_1, σ_2 by querying the verifier with the appropriate commitment as part of the second message of the protocol. Let $\mathcal{J}_1$ (and $\mathcal{J}_2$) be the random variable representing the message index where the commitment c_0 and the corresponding signature σ_1 (resp., c_1' and σ_2) were sent in the experiment $\text{HYB}_1^{P^*}(n, z)$. We also denote by $\mathcal{J}_3$ the message index where P^* sends (the same) $\mathtt{BH}_2$ of the convincing session. We set each of them to $\perp$ if no such session exists. From the unforgeability of the signature scheme and the binding property of the commitment, we have the following claims.

Claim 2 *For every adversary A there exists a negligible function $\nu_1()$ such that for all $n \in \mathbf{N}, z \in \{0,1\}^*$, the probability that the output of $\text{HYB}_1^A(n, z)$ is not $\perp$ and any of $\mathcal{J}_1, \mathcal{J}_2$ or $\mathcal{J}_3$ is $\perp$ is at most $\nu_1(n)$.*

Claim 3 *For every adversary A there exists a negligible function $\nu_2()$ such that for all $n \in \mathbf{N}, z \in \{0,1\}^*$, the probability that the output of $\text{HYB}_1^A(n, z)$ is not $\perp$, $\mathcal{J}_1, \mathcal{J}_2, \mathcal{J}_3 \neq \perp$ and $\mathcal{J}_1 \geq \mathcal{J}_2$ or $\mathcal{J}_2 > \mathcal{J}_3$ is at most $\nu_2(n)$.*

Before proving Claims 2 and 3, we prove soundness using these claims. Consider $B^O(1^n, \mathsf{pp})$ that internally incorporates P^* and begins an internal emulation by supplying the verifier messages internally and proceeds as follows:

1. B picks three integers i_1, i_2, i_3 at random such that $i_1 < i_2 < i_3$.
2. B selects keys $(vk, sk) \leftarrow \mathsf{Gen}(1^n)$. It then internally feeds P^* with $(\mathtt{BH}_1, vk, \mathsf{pp})$ where $\mathtt{BH}_1$ is the first message of the WI-AOK proving language L^*. To generate the third message as the verifier, B^* first queries the oracle with the commitment c received in the second message of that session and obtains σ'. Then it generates a random string r and obtains a signature for $c|r$, σ under

key sk. B then feeds P^* with $\mathtt{BH}_3, r, \sigma, \sigma'$ where $\mathtt{BH}_3$ is honestly generated. In this manner B continues with the emulation internally.

3. B continues the emulation until the partial transcript has i_1 messages. If this is not a second message of any session, it halts. Otherwise, it takes the commitment c as part of this message and forwards it to $V_{\mathtt{zk}}$ as the first message in the external execution. Upon receiving the challenge r from the external verifier, it forwards that challenge internally as part of the third message corresponding to the same session; it generates σ, σ' as before. It then continues the emulation until the partial transcript has i_2 messages. If this is not a second message of any session, it halts. Otherwise, let β be the partial transcript and α be its session number.
4. Next, it continues the emulation from β until the partial transcript has totally i_3 messages. If the last message is not the third message of session α it halts. Otherwise, let the partial transcript be $(\beta :: \beta_1)$ (where :: denotes concatenating transcripts). Now, it runs two random continuations from i_3 to completion and extracts a witness use in the WI-AOK using the special-sound property. Let the two transcripts be $(\beta :: \beta_1 :: \beta_{11})$ and $(\beta :: \beta_1 :: \beta_{12})$. If it fails to extract a fake witness internally then it halts. If it obtains a fake witness but the witness does not contain c, r from the previous step it halts. Otherwise, it takes p_1 from the witness and sends $\textsc{com}(p_1, \tau_1)$ where τ_1 is randomly chosen externally to $V_{\mathtt{zk}}$.
5. Upon receiving the challenge r' from $V_{\mathtt{zk}}$, B internally rewinds P^* to the prefix β. B starts a new continuation from this point and feeds r' as part of the third message in the current session. B then continues the internal emulation until the partial transcript $(\beta :: \beta_2)$ has i_3 messages. Once again B extracts the witness in the WI-AOK by emulating two random continuations to completion from $(\beta :: \beta_2)$, say $(\beta :: \beta_2 :: \beta_{21})$ and $(\beta :: \beta_2 :: \beta_{22})$. If c, r, p_1, r' are not part of the witness B aborts. Otherwise it takes p_2 from the witness and sends $\textsc{com}(p_2, \tau_2)$ where τ_2 is randomly chosen externally to $V_{\mathtt{zk}}$.
6. B stops the internal emulation and proceeds to complete the external execution with $V_{\mathtt{zk}}$ by using $(p_1, p_2, \tau_1, \tau_2)$ as the witness for the proof phase.

It follows from the soundness of the WI AOK and the way $\mathbf{R}_{L_3}$ is defined, that if B succeeds in extracting the fake witness that contains the appropriate previous messages, then, except with negligible probability, B succeeds in convincing $V_{\mathtt{zk}}$ in the external execution. It suffices to argue that B is able to achieve this with non-negligible probability. Recall that P^* succeeds in convincing a false statement to V with non-negligible probability, say $\frac{1}{p(n)}$.

By Claims 2 and 3, it holds for sufficiently large n that with probability at least $\frac{1}{p(n)} - \nu_1(n) - \nu_2(n)$ that P^* cheats and $\mathcal{J}_1, \mathcal{J}_2, \mathcal{J}_3 \neq \perp$ and $\mathcal{J}_1 < \mathcal{J}_2 < \mathcal{J}_3$ in $\textsc{hyb}_1^{P^*}(n, z)$. Since there are only polynomially many sessions we can further assume that there exists a polynomial $p_1(n)$ and functions $i_1(), i_2(), i_3()$ such that, for sufficiently large n, with probability $\frac{1}{p_1(n)}$ over the experiment $\textsc{hyb}_1^{P^*}(n, z)$, it holds that $\mathcal{J}_1 = i_1(n)$, $\mathcal{J}_2 = i_2(n)$ and $\mathcal{J}_3 = i_3(n)$. For a complete

transcript β of an interaction with P^*, we say event $\mathsf{WO}(\beta)$ occurs if $\mathcal{J}_1(\beta) = i_1(n)$, $\mathcal{J}_2(\beta) = i_2(n)$ and $\mathcal{J}_3(\beta) = i_3(n)$ (for *well-ordered*).

We now analyze the success probability of B. We do this by analyzing the probability that B succeeds in each of the steps iteratively.

Event E_1: We say E_1 holds if $i_1 = i_1(n), i_2 = i_2(n)$ and $i_3 = i_3(n)$. Since there are only polynomially many sessions, this happens with polynomial probability, say $\frac{1}{p_2(n)}$.

Event E_2: We say that E_2 holds for a partial transcript β, i.e. $\mathsf{E}_2(\beta) = 1$, if β is of length i_2 and WO holds in random continuation from β with probability $\frac{1}{2p_1(n)}$. Since WO holds with probability $\frac{1}{p_1(n)}$, using an averaging argument, we can conclude that with probability at least $\frac{1}{2p_1(n)}$ over partial transcripts of length i_2, WO holds in a random continuation with probability at least $\frac{1}{2p_1(n)}$. So conditioned on E_1, $\mathsf{E}_2(\beta)$ holds with probability $\frac{1}{2p_1(n)}$ over β.

Event E_3: We say that E_3 holds for a partial transcript β, i.e. $\mathsf{E}_3(\beta) = 1$, if β is of length i_3 and WO holds in random continuation from β with probability $\frac{1}{4p_1(n)}$. We estimate the probability E_3 holds conditioned on E_2 and E_1. If E_1 and E_2 holds for transcript β, we know a random continuation from β yields a transcript where WO holds with probability at least $\frac{1}{2p_1(n)}$. So using another averaging argument, we get that, $\Pr[\mathsf{E}_3(\beta :: \beta_1)|\mathsf{E}_2(\beta) \wedge \mathsf{E}_1] \geq \frac{1}{4p_1(n)}$

B succeeds if it extracts the correct witness in Steps 4 and 5. More precisely, B will succeed except with negligible probability, if WO holds in all of $(\beta :: \beta_1 :: \beta_{11})$, $(\beta :: \beta_1 :: \beta_{12})$, $(\beta :: \beta_1 :: \beta_{21})$ and $(\beta :: \beta_1 :: \beta_{21})$ as the witness will be correct and the special-sound extractor will succeed. This probability can be written as

$$\Pr[B \text{ succeeds}] = \Pr\left[\mathsf{WO}(\beta :: \beta_1 :: \beta_{11}) \wedge \mathsf{WO}(\beta :: \beta_1 :: \beta_{12}) \wedge \mathsf{WO}(\beta :: \beta_1 :: \beta_{21}) \wedge \mathsf{WO}(\beta :: \beta_1 :: \beta_{22})\right] - 2\nu(n) \quad (1)$$

where $\nu(\cdot)$ is the probability that the special-sound extractor fails. From the description of the events, it holds that

$$\begin{aligned}
\Pr[\mathsf{WO}(\beta :: \beta_1 :: \beta_{11})|\mathsf{E}_3(\beta :: \beta_1) \wedge \mathsf{E}_1] &\geq \frac{1}{4p_1(n)} \\
\Pr[\mathsf{E}_3(\beta :: \beta_1)|\mathsf{E}_2(\beta) \wedge \mathsf{E}_1] &\geq \frac{1}{4p_1(n)} \\
\Pr[\mathsf{E}_2(\beta)|\mathsf{E}_1] &\geq \frac{1}{2p_1(n)} \\
\Pr[\mathsf{E}_1] &\geq \frac{1}{p_2(n)}
\end{aligned}$$

And similar bounds hold for the other transcripts as well. Therefore, simplifying Equation 1, we get that

$$\Pr[B \text{ succeeds}] \geq \frac{1}{p_2(n)} \frac{1}{2p_1(n)} \left(\frac{1}{4p_1(n)}\right)^2 \left(\frac{1}{4p_1(n)}\right)^4 - 2\nu(n)$$

which is non-negligible

We remark that the transformation works only for a constant-round protocol since B makes a guess for each round (i.e., i_1, i_2 and i_3) each correct only with polynomial probability.

It only remains to prove Claims 2 and 3. This on a high-level will follow from the binding property of the commitment and the unforgeability of the signature scheme.

Proof of Claim 2. Since the output of HYB_1 is not $\perp$, it immediately follows that $\mathcal{J}_3 \neq \perp$. We now show that P^* must have obtained the signature σ_1, σ_2 obtained from the witness by sending the commitment and receiving the corresponding random string with the signature in some session. Suppose not, then we can violate the unforgeability of the signature scheme by constructing an adversary C that receives a verification key vk as input conducts the HYB_1 experiment by supplying vk to P^* and forwarding all signing queries to the signing oracle. Finally upon extracting a fake witness, C simply outputs either $(c_0|r, \sigma_1)$ or $(c_1'|r', \sigma_2)$ which ever is valid.

Proof of Claim 3. Using the preceding argument, we can conclude that the signatures must be obtained before P^* convinces the verifier in some session, i.e. $\mathcal{J}_1 < \mathcal{J}_3$ and $\mathcal{J}_2 < \mathcal{J}_3$.[12] It only remains to argue that $\mathcal{J}_3 > \mathcal{J}_1 > \mathcal{J}_2$ does not happen. Assume for contradiction that with non-negligible probability $\mathcal{J}_1, \mathcal{J}_2, \mathcal{J}_3 \neq \perp$ and $\mathcal{J}_3 > \mathcal{J}_1 > \mathcal{J}_2$. This means that P^* was able to commit to a signature σ_1 as part of $p_1|\sigma_1$ in session $\mathcal{J}_2$ before it obtained the signature σ_1 from the verifier in session $\mathcal{J}_1$. We construct an adversary C that violates the collision-resistance property of the signature scheme.

C on input (n, vk) and oracle access to a signing oracle $\text{Sign}_{sk}()$ first selects i_1, i_2 and i_3 at random. Then it internally incorporates $P^*(n, z)$ and begins an internal emulation of an execution of P^* as follows. It forwards the verification-key vk internally to P^* as part of the first message and generates all the verifier messages honestly except the signatures corresponding to vk which it obtains by feeding the corresponding message to the signing oracle. C then runs the emulation until the partial transcript, say β, has i_2 messages. If this is not the second message of a session, C halts. Otherwise, it spawns two random continuation from β until the partial transcripts, say $(\beta :: \beta_1)$ and $(\beta :: \beta_2)$ of both threads has i_3 messages. If in either of the thread the current message is not the second message of a session C halts. Otherwise, it runs two random continuations from both $(\beta :: \beta_1)$ and $(\beta :: \beta_2)$ to obtain $(\beta :: \beta_1 :: \beta_{11})$, $(\beta :: \beta_1 :: \beta_{12})$, $(\beta :: \beta_1 :: \beta_{21})$ and $(\beta :: \beta_1 :: \beta_{21})$ and run the special-sound extractor of the WI-AOK protocol to obtain two witnesses. If the extractor succeeds in extracting a fake witness from both these sessions and σ_1 is the same in both these witnesses, then the message signed will be different with high-probability. This is because the message being signed has a random string r of length $O(n)$ and for two threads to have the same challenge is exponentially

[12] Consider C that proceeds as in Claim 2, but stops at a random session, extracts the witness and outputs the signature obtained from the witness.

small, say $\nu_1(n)$. Therefore, by the soundness of the WI-AOK protocol we have two different messages with the same signature. C outputs them as a collision.

To argue that C succeeds with non-negligible probability we proceed exactly as in the previous argument. We know that with non-negligible probability, there exists $i_1(n), i_2(n), i_3(n)$ such that $\mathcal{J}_1 = i_1(n), \mathcal{J}_2 = i_2(n), \mathcal{J}_3 = i_3(n)$ and $\mathcal{J}_2 > \mathcal{J}_1 > \mathcal{J}_3$ with probability $\frac{1}{p_1(n)}$. Lets call this event WO as before. Define events E_1, E_2 and E_3 exactly as before. Following the same approach we can conclude that C succeeds with probability at least

$$\frac{1}{p_2(n)} \frac{1}{2p_1(n)} \left(\frac{1}{4p_1(n)}\right)^2 \left(\frac{1}{4p_1(n)}\right)^4 - 2\nu(n) - \nu_1(n)$$

which is non-negligible and thus we arrive at a contradiction.

Argument of Knowledge Since the $\mathcal{O}^{\mathsf{SIG}}$-oracle aided $\langle P_{\mathtt{zk}}, V_{\mathtt{zk}} \rangle$ protocol is also a argument of knowledge, from the proof of soundness, it holds that our 4-round protocol is also an argument of knowledge.

Zero Knowledge. Before we describe the simulator, we need the following definition of a valid SIG''-oracle similar to Definition 19.

Definition 21 (Valid SIG'' Oracle). *An oracle $\mathcal{O}''$ is a* valid (SIG'', ℓ) oracle *if there is a negligible $\mu(\cdot)$ such that for every $n \in N$, the following holds with probability $1-\mu(n)$ over $\mathsf{pp}, O \leftarrow \mathcal{O}''(1^n)$: for every $m \in \{0,1\}^{\ell(n)}$, $O(m)$ returns $(\mathtt{BH}_2, c, r, \sigma, \tau)$ such that $\mathsf{Ver}_{vk}(c|r, \sigma) = 1$, $c = \mathrm{COM}(m, \tau)$ and r is the second string in the tuple output by V^* when fed $\mathtt{BH}_2, c$ with probability at least $1-\mu(n)$.*

Consider some malicious (w.l.o.g. deterministic) verifier $\tilde{V}^*$ for (P, V) of size $T_{\tilde{V}^*}$. We remark that while the simulator for the resettably-sound ZK protocol in [10] had one signing slot, here we have a slot that serves as a signing slot for two different keys sk and sk'. We use two signing keys for simplicity. We use two keys for simplicity. We construct a simulator S for $\tilde{V}^*$ that starts simulating $(P, \tilde{V}^*)$ until it receives $\mathtt{BH}_1$ and two verification keys vk, vk'. Let V^* be the "residual" verifier after the first message is sent. It then proceeds as follows.

1. S prepares a valid $(\mathsf{SIG}', 2n)$ oracle $\mathcal{O}'$ and $(\mathsf{SIG}'', 2n)$ oracle $\mathcal{O}''$ by rewinding V^* and using the second and third message of the protocol as a Signing Slot for both sk and sk'. This step is essentially the same as what the simulator does in the protocol presented in [10] which in turn is inspired by Goldreich-Kahan [15]),
2. S will convince V^* in the WI-AOK using the second witness. Towards this, S will first use oracle $\mathcal{O}'$ to produce a Sig-com tree for $C = \mathtt{ECC}(\Pi)$ where $\Pi = V^*$. Let d and l_λ be the depth and root of the Sig-com tree. Using the oracle $\mathcal{O}''$, S obtains (c_0, r, σ_1, τ) where $(c_0, r, vk') \in \mathbf{R}_{L_1}$ and $\mathsf{Ver}_{vk}(c_0|r, \sigma_1) = 1$.
3. S then generates the first prover message p_1 using the witness for $(c_0, r, vk') \in \mathbf{R}_{L_1}$. Using the oracle $\mathcal{O}''$ again, S generates $c_1, r', \sigma_2, \tau_1$ such that $c_1 = \mathrm{COM}(p_1|\sigma_1, \tau_1)$ and $\mathsf{Ver}_{vk}(c_1|r', \sigma_2) = 1$. S now generates the second prover message p_2 for the UA using r' as the challenge message for the UA.

4. Finally, S rewinds V^* to the top and completes the interaction with V^* by using $\langle c_0, r, c_1, r', p_1, p_2, \tau_1, \sigma_1, \sigma_2 \rangle$ as the second witness in the WI-AOK.

The correctness of S follows essentially using the same proof as in [10]. First, we argue that S can prepare valid oracles for both the keys. Given valid oracles, S obtains a valid second witness for the WI-AOK. It then runs V^* in a straight-line manner by generating messages for the WI-AOK protocol using the second witness and all the other messages as the honest prover. Indistinguishability of the output of the simulator follows directly from the witness-indistinguishability property of the WI-AOK protocol. It only remains to argue that S can prepare valid $\mathcal{O}^{\mathsf{SIG}'}$ and $\mathcal{O}^{\mathsf{SIG}''}$ oracles. We remark that the approach we take is similar to [10], with the exception that the preamble phase of the oracle preparation is executed only once for both oracles. First S executes the following preamble.

- S sends $c, \mathtt{BH}_2$ to V^* where $c = \mathrm{COM}(0^{2n}; \tau)$ with uniform τ and $\mathtt{BH}_2$ is a random dummy second message of the Blum-Hamiltonicity protocol[13], and then receives $\mathtt{BH}_3, r, \sigma, \sigma'$ from V^*. If σ is not a valid signature of $c|r$ under verification vk or σ' is not a valid signature of c under vk', then the simulation halts immediately and outputs the transcript up to that point.
- S repetitively queries V^* with fresh commitments $\mathrm{COM}(0^{2n}; \tau)$ at the Signing Slot along with dummy $\mathtt{BH}_2$ messages until it collects $2n$ valid signatures. Let t be the number of queries $\tilde{S}$ makes.

Preparing $\mathcal{O}^{\mathsf{SIG}''}$ Oracle: Define $\mathcal{O}''$ that outputs $\mathsf{pp} = vk$, and an oracle O that on input a message $m \in \{0,1\}^{2n}$, proceeds as follows: O repetitively queries V^* at the Signing Slot with fresh commitments $c_m = \mathrm{COM}(m; \tau)$ with dummy $\mathtt{BH}_2$ messages for at most t times. If V^* ever replies $\mathtt{BH}_3, r, \sigma, \sigma'$ where $\mathsf{Ver}_{vk}(c_m|r, \sigma) = 1$, then O outputs $(\mathtt{BH}_2, c_m, r, \sigma, \tau)$. Otherwise, O returns $\perp$.

Preparing $\mathcal{O}^{\mathsf{SIG}'}$ Oracle: Define $\mathcal{O}'$ that outputs $\mathsf{pp} = vk'$, and an oracle O that on input a message $m \in \{0,1\}^{2n}$, proceeds as follows: O repetitively queries V^* at the Signing Slot with fresh commitments $\mathrm{COM}(m; \tau)$ for at most t times. If V^* ever replies a valid signature σ' for $\mathrm{COM}(m, \tau)$, then O outputs (σ', τ). Otherwise, O returns $\perp$.

We now analyze the running time. If $t \geq 2^{n/2}$, then S aborts. To analyze this part of the simulator S, we introduce some notation. Let $p(m)$ be the probability that V^* on query $\mathtt{BH}_2, c_m$ where $\mathtt{BH}_2$ is the specific second message of the Blum-Hamiltonicity protocol and $c_m = \mathrm{COM}(m, \tau)$ of $m \in \{0,1\}^{2n}$ a random commitment returns a valid signature of $c_m|r$ under sk where r is part of V^*'s output when fed $\mathtt{BH}_2, c_m$ and a valid signature of c_m under sk'. Let $p = p(0^{2n})$.

We first show that S runs in expected polynomial time. To start, note that S aborts at the end of the Signature Slot with probability $1 - p$, and in this case, S runs in polynomial time. With probability p, S continues to invoke a strictly

[13] Recall that, in the second message of the Blum-Hamiltonicity protocol, the prover sends a set of commitments. Hence, to generate a dummy message, the simulator can simply commit to the all 0's string.

polynomial-time simulator S for the residual V^*, which has size bounded by $T_{\tilde{V}^*}$. Thus, S runs in some $T = \mathtt{poly}(\mathtt{T}_{\tilde{\mathtt{V}}^*})$ time and makes at most T queries to both its oracles, which in turn runs in time $t \cdot \mathtt{poly}(\mathtt{n})$ to answer each query. Also note that S runs in time at most 2^n, since S aborts when $t \geq 2^{n/2}$. Now, we claim that $t \leq 10n/p$ with probability at least $1 - 2^{-n}$, and thus the expected running time of S is at most

$$(1-p) \cdot \mathtt{poly}(\mathtt{n}) + \mathtt{p} \cdot \mathtt{T} \cdot (\mathtt{10n}/\mathtt{p}) \cdot \mathtt{poly}(\mathtt{n}) + 2^{-\mathtt{n}} \cdot 2^{\mathtt{n}} \leq \mathtt{poly}(\mathtt{T}_{\tilde{\mathtt{V}}^*}, \mathtt{n}).$$

To see that $t \leq 10n/p$ with overwhelming probability, let $X_1, \ldots, X_{10n/p}$ be i.i.d. indicator variables on the event that V^* returns a valid signature for the message 0^{2n}, and note that $t \leq 10n/p$ implies $\sum_i X_i \leq 2n$, which by a standard Chernoff bound, can only happen with probability at most 2^{-n}.

Finally, we argue indistinguishability. First, the computational hiding property of COM implies that there exists some negligible $\nu(\cdot)$ such that $|p(m) - p| \leq \nu(n)$ for every $m \in \{0,1\}^{2n}$. Now we consider two cases. If $p \leq 2\nu$, then the indistinguishability trivially holds since the interaction aborts at the end of the Signature Slot (in this case, the view is perfectly simulated) with all but negligible probability. On the other hand, if $p \geq 2\nu$, we show that $\mathcal{O}''$ generated by S is a valid $(\mathsf{SIG}'', 2n)$ oracle for SIG'' and $\mathcal{O}'$ generated by S is a valid $(\mathsf{SIG}', 2n)$ oracle for SIG' with overwhelming probability, and thus the indistinguishability of S follows by the indistinguishability of S.

To see that $\mathcal{O}''$ is a valid $(\mathsf{SIG}'', 2n)$ oracle for SIG'' with overwhelming probability, note again by a Chernoff bound that $n/p \leq t \leq 2^{n/2}$ with probability at least $1 - 2^{-\Omega(n)}$. In this case, for every $m \in \{0,1\}^{2n}$, $p(m) \geq p - \nu \geq p/2$ implies that $t \geq n/2p(m)$, and thus $O(m)$ learns a valid signature of $\mathrm{COM}(m;\tau)$ from V^* with probability at least $1 - 2^{-\Omega(n)}$. A similar argument establishes that $\mathcal{O}'$ is a valid $(\mathsf{SIG}', 2n)$ oracle for SIG' with overwhelming probability. This concludes the proof of correctness.

Acknowledgment

Ostrovsky's research is supported in part by NSF grants CCF-0916574; IIS-1065276; CCF-1016540; CNS-1118126; CNS-1136174; US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, Lockheed-Martin Corporation Research Award, Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0392.

Pass is supported in part by a Alfred P. Sloan Fellowship, Microsoft New Faculty Fellowship, NSF Award CNS-1217821, NSF CAREER Award CCF-0746990, NSF Award CCF-1214844, AFOSR YIP Award FA9550-10-1-0093, and DARPA and AFRL under contract FA8750-11-2- 0211.

Chung is supported by NSF Award CNS-1217821, NSF Award CCF-1214844 and Pass' Sloan Fellowship.

Visconti's research is supported in part by the MIUR Project PRIN "GenData 2020".

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies or positions, either expressed or implied, of the Department of Defense, the Defense Advanced Research Projects Agency or the U.S. Government.

References

1. Barak, B., Goldreich, O.: Universal arguments and their applications. In: Computational Complexity. pp. 162–171 (2002)
2. Barak, B., Goldreich, O., Goldwasser, S., Lindell, Y.: Resettably-sound zero-knowledge and its applications. In: FOCS'02. pp. 116–125 (2001)
3. Bellare, M., Goldreich, O.: On defining proofs of knowledge. In: CRYPTO '92. pp. 390–420 (1992)
4. Bellare, M., Jakobsson, M., Yung, M.: Round-optimal zero-knowledge arguments based on any one-way function. In: Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding. Lecture Notes in Computer Science, vol. 1233, pp. 280–305. Springer (1997)
5. Bitansky, N., Paneth, O.: From the impossibility of obfuscation to a new non-black-box simulation technique. In: FOCS (2012)
6. Bitansky, N., Paneth, O.: On the impossibility of approximate obfuscation and applications to resettable cryptography. In: STOC (2013)
7. Blum, M.: How to prove a theorem so no one else can claim it. Proc. of the International Congress of Mathematicians pp. 1444–1451 (1986)
8. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (leveled) fully homomorphic encryption without bootstrapping. In: ITCS. pp. 309–325. ACM (2012)
9. Chung, K.M., Ostrovsky, R., Pass, R., Visconti, I.: Simultaneous resettability from one-way functions. In: 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013. pp. 60–69. IEEE Computer Society (2013)
10. Chung, K.M., Pass, R., , Seth, K.: Non-black-box simulation from one-way functions and applications to resettable security. In: STOC. ACM (2013)
11. Di Crescenzo, G., Persiano, G., Visconti, I.: Improved setup assumptions for 3-round resettable zero knowledge. In: Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3329, pp. 530–544. Springer (2004)
12. Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: STOC '90. pp. 416–426 (1990)
13. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: STOC. pp. 169–178. ACM (2009)
14. Goldreich, O.: Foundations of Cryptography — Basic Tools. Cambridge University Press (2001)
15. Goldreich, O., Kahan, A.: How to construct constant-round zero-knowledge proof systems for NP. Journal of Cryptology 9(3), 167–190 (1996)
16. Goldwasser, S., Micali, S.: Probabilistic encryption. J. Comput. Syst. Sci. 28(2), 270–299 (1984)

17. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. In: STOC '85. pp. 291–304. ACM (1985), http://doi.acm.org/10.1145/22145.22178
18. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. 18(1), 186–208 (1989)
19. Goyal, V., Jain, A., Ostrovsky, R., Richelson, S., Visconti, I.: Constant-round concurrent zero knowledge in the bounded player model. In: Advances in Cryptology - ASIACRYPT 2013, 19th International Conference on the Theory and Application of Cryptology and Information Security, 2004, Proceedings. Lecture Notes in Computer Science, vol. 8279, pp. 21–40. Springer (2013)
20. Lin, H., Pass, R.: Constant-round non-malleable commitments from any one-way function. In: STOC. pp. 705–714 (2011)
21. Merkle, R.C.: A digital signature based on a conventional encryption function. In: CRYPTO. pp. 369–378 (1987)
22. Micali, S.: Computationally sound proofs. SIAM Journal on Computing 30(4), 1253–1298 (2000)
23. Naor, M., Yung, M.: Universal one-way hash functions and their cryptographic applications. In: STOC '89. pp. 33–43 (1989)
24. Ostrovsky, R., Visconti, I.: Simultaneous resettability from collision resistance. Electronic Colloquium on Computational Complexity (ECCC) 19, 164 (2012)
25. Ostrovsky, R., Wigderson, A.: One-way fuctions are essential for non-trivial zero-knowledge. In: ISTCS. pp. 3–17 (1993)
26. Pass, R., Rosen, A.: New and improved constructions of non-malleable cryptographic protocols. In: STOC '05. pp. 533–542 (2005)
27. Pass, R., Tseng, W.L.D., Wikström, D.: On the composition of public-coin zero-knowledge protocols. SIAM J. Comput. 40(6), 1529–1553 (2011)
28. Rompel, J.: One-way functions are necessary and sufficient for secure signatures (1990)