2013 IEEE 54th Annual Symposium on Foundations of Computer Science

Simultaneous Resettability from One-Way Functions

Kai-Min Chung
Academia Sinica
Taiwan

Rafail Ostrovsky
UCLA
USA

Rafael Pass
Cornell University
USA

Ivan Visconti
University of Salerno
ITALY

***Abstract*—Resettable-security, introduced by Canetti, Goldreich, Goldwasser and Micali (STOC'00), considers the security of cryptographic two-party protocols (in particular zero-knowledge arguments) in a setting where the attacker may "reset" or "rewind" one of the players. The strongest notion of resettable security, *simultaneous resettability*, introduced by Barak, Goldreich, Goldwasser and Lindell (FOCS'01), requires resettable security to hold for *both* parties: in the context of zero-knowledge, both the soundness and the zero-knowledge conditions remain robust to resetting attacks.**

To date, all known constructions of protocols satisfying simultaneous resettable security rely on the existence of ZAPs; constructions of ZAPs are only known based on the existence of trapdoor permutations or number-theoretic assumptions.

In this paper, we provide a new method for constructing protocols satisfying simultaneous resettable security while relying only on the minimal assumption of one-way functions. Our key results establish, assuming only one-way functions:

- **Every language in $\mathcal{NP}$ has an $\omega(1)$-round simultaneously resettable witness indistinguishable argument system.**
- **Every language in $\mathcal{NP}$ has a (polynomial-round) simultaneously resettable zero-knowledge argument system.**

The key conceptual insight in our technique is relying on *black-box impossibility results for concurrent zero-knowledge* to achieve resettable-security.

***Keywords*-proof systems; resettable WI/ZK/soundness;**

I. Introduction

Zero-knowledge (ZK) interactive proofs [GMR89] and arguments [BCC88] 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. The *soundness* condition of a zero-knowledge proof (resp. argument) stipulates that if $x \notin L$, the no matter what the Prover does (resp., no matter what a computationally bounded Prover does), the Verifier will only accept at the end of the interaction with negligible probability. The *zero-knowledge* condition, on the other hand, stipulates that no efficient malicious verifier can learn anything new from the prover. The zero-knowledge property is formalized using the so-called *simulation paradigm*: for every malicious verifier V^*, we require the existence of a "simulator" S that, given just the input x, can indistinguishably reproduce the view of V^* in an interaction with the honest prover. Beyond being fascinating in their own right, zero-knowledge proofs/arguments have numerous cryptographic applications and are one of the most fundamental cryptographic building blocks. Additionally, the simulation paradigm on which it is based extends well beyond the notion of zero-knowledge, and is a crucial component of modern definitions of protocol security and as such, the study of zero-knowledge proofs often provides insight into protocol security more widely.

Zero-knowledge protocols rely on both the Prover and the Verifier tossing random coins; furthermore, while for some protocols (called public-coin protocols) the Verifier's randomness can be public, it is crucial that the Prover's randomness is kept secret from the Verifier; additionally, though out the interaction, it is crucial that the Prover keeps a secret state. A natural question is whether zero-knowledge protocols can be made secure if an attacker may "reset" and "restart" his opponent, forcing him to return to an earlier state of the computation, and reusing the same random tape. (This model is particularly relevant for cryptographic protocols being executed on embedded devices, such as smart cards. Since these devices have neither a built-in power supply, nor a non-volatile re-writable memory, they can be "reset" by simply disconnecting and reconnecting the power supply.) Note that any stateless protocol (that is secure under multiple execution) is also directly secure under a reset-attack, but since cryptographic protocols typically aren't, achieving resettable security becomes demanding.

The notion of *resettable-zero knowledge (rZK)*, introduced by Canetti, Goldreich, Goldwasser and Micali [CGGM00] considers zero-knowledge protocols where the zero-knowledge property is retained under a resetting attack. That is, the Prover is protected even if the Verifier can reset him to his original state, while reusing the same randomness (but the Verifier is not necessarily protected against a resetting attack.) The following year, Barak, Goldreich, Goldwasser and Lindell [BGGL01] considered *resettably-sound zero-*

0272-5428/13 $26.00 © 2013 IEEE
DOI 10.1109/FOCS.2013.15

knowledge (rsZK) proofs, where instead the soundness condition holds against resetting attacks. That is, the Prover cannot convince the Verifier of any false statements even if he can reset the Verifier to its original state (but, now, the Prover is not necessarily protected against a resetting attack.) Both resettable zero-knowledge and resettably-sound zero-knowledge arguments were originally constructed based on the existence of collision-resistant hash functions [CGGM00], [BGGL01]; Bitansky and Paneth, more recently, provided a construction of resettably-sound zero-knowledge arguments based on the existence of an Oblivious Transfer protocol [BP12], and finally Chung, Pass and Seth [CPS13] provided a construction of both primitives based on the minimal assumption of one-way functions [CPS13] (as shown by Ostrovsky and Wigderson [OW93], one-way functions are also necessary for constructing zero-knowledge arguments for hard-on-the-average languages).

But resettable zero-knowledge, and resettably-sound zero-knowledge only consider resettably security for a single of the players (either the Prover or the Verifier). Can we achieve resettable security for *both*? Such a notion of resettable security was called *simultaneous resettability* by Barak et al [BGGL01]. While [BGGL01] left open the question of constructing simultaneously resettable zero-knowledge arguments, they provided a construction of a relaxation of zero-knowledge arguments that satisfy simultaneous resettable security: namely, witness-indistinguishable (WI) arguments [FS90] (rather than ensuring that the Verifier does not learn anything new, in a witness indistinguishable argument, the Prover is guaranteed that the Verifier cannot learn what witness the Prover is using.) Their construction was based on the existence of, so-called, ZAPs [DN00] (namely, the existence of two-round witness-indistinguishable proofs), which in turn can be based on enhanced trapdoor permutations, or number-theoretic assumptions.

More recently, Deng, Goyal and Sahai [DGS09] provided the first construction of simultaneously resettable zero-knowledge (i.e., resettable-sound resettable zero-knowledge (rsrZK)) for $\mathcal{NP}$. Their construction was based on the existence of one-to-one one-way functions, collision-resistant hash functions and ZAPs; more recently, the collision-resistant hash function assumption was removed in [CPS13], and the one-to-one one-way function assumption was removed in [BP13].[1] Thus, given the state-of-the art, constructions of both simultaneously resettable witness indistinguishability (rsrWI) and zero-knowledge can be constructed assuming the existence of ZAPs. We here focus on the question of whether ZAPs are necessary for achieving simultaneous resettable security.

> *Can we achieve simultaneous resettable security without assuming the existence of ZAPs? In particular, does the the existence of only one-way functions suffice?*

Before turning to our results, let us briefly explain the central role of ZAPs: as mentioned above, ZAPs are two-round witness-indistinguishable proofs. As such protocols are two-round (i.e., essentially non-interactive), they are *stateless*[2] which enables the property of simultaneous resettability. ZAPs are equivalent to the existence of non-interactive zero-knowledge proofs [DN00]; it is a long-standing open question whether non-interactive zero-knowledge proofs (with an efficient prover strategy) can be based only on the existence of one-way functions.

We stress that resettable security has also been studied in various relaxed models of security (e.g., with bare public keys [CGGM00], [MR01], [CPV04a], [YZ07], [CPV04b]) and "bounded"-resettable security ([Bar01], [BGGL01], [BOV12]), but despite the relaxations, all protocols achieving rsrWI and rsrZK rely on ZAPs.

A. Our Result

In this work, we answer the above questions in the affirmative. We show how to achieve simultaneously resettable witness-indistinguishable and zero-knowledge argument systems from one-way functions.

Theorem 1 (Main Theorem 1, informally stated)**.** *Assume the existence of one-way functions. Then there exists an $\omega(1)$-round simultaneously resettable witness-indistinguishable argument of knowledge for $\mathcal{NP}$.*

(An argument *of knowledge* is a stronger notion of an interactive argument where not only the Prover convinces the verifier that the statement x (to be proven) is part of some language L, but also that it "knows" a $\mathcal{NP}$-witness for $x \in L$.)

We next employ simultaneously resettable witness-indistinguishable arguments to obtain simultaneously resettable zero-knowledge.

Theorem 2 (Main Theorem 2, informally stated)**.** *Assume the existence of one-way functions. Then, for every $\epsilon > 0$, there exists an $O(k^\epsilon)$-round simultaneously resettable zero-knowledge argument of knowledge for $\mathcal{NP}$.*

In our approach, we abandon the "stateless" approach for achieving simultaneous resettability and demonstrate

[1] The authors of [DGS09] also claimed to have a variant of their protocol that does not need one-to-one one-way functions, but the results never appear in writing. In an earlier version of this paper, we provided our own variant of the [DGS09] protocol that dispensed of one-to-one one-way functions.

[2] Rather, although the prover need to keep some secret state, this state never changes.

how to keep a stateful protocol (which then can be based on one-way functions) without sacrificing resettable security.

B. Our Techniques

rsrWI from OWFs. The starting point of our technique is a connection between lower bounds for *black-box* zero-knowledge[3] and resettable soundness. Black-box zero-knowledge impossibility results typically demonstrate that certain classes of protocols (e.g., constant-round public-coin protocols), or *composition* (e.g., parallel repetition) of certain classes of underlying protocols (e.g., public-coin protocols), satisfy a weaker notion of *fixed-input* resettable soundness (where soundness only needs to hold as long as the resetting prover does not get to change the statement) if the verifier is slightly modified to appropriately generate its randomness using a pseudorandom function (PRF). This connection was first made by Pass, Tseng and Wikstrom [PTW11] where it was shown that the original black-box zero-knowledge impossibility result of [GK96] for constant-round public-coin protocols implicitly shows that any constant-round public-coin argument is fixed-input resettably-sound if the verifier generates its randomness by applying a PRF to the transcript. Additionally, the black-box zero-knowledge impossibility results for parallel composition of public-coin protocols of [PTW11] shows that repeating any (not necessarily constant-round) public-coin protocol sufficiently many times in parallel and generating the verifier's randomness in each session by applying a PRF to the transcript, yields a fixed-input resettably-sound protocol. [PTW11] also note that, following the technique used in [BGGL01], if the underlying protocol also is an *argument of knowledge*, then the resulting protocol actually satisfies the standard (unbounded) notion of resettable soundness.

However, although, the connection between impossibility results for black-box zero-knowledge and resettable-soundness was made in [PTW11], no new corollaries of this connection were provided. We here show that this connection can be taken even further, and by doing so, provide new (and improved) constructions of resettably-sound protocol. More precisely, we show that impossibility results for black-box *concurrent* zero-knowledge[4] can be appropriately interpreted as a way to transform a class of protocols into ones that are resettably sound. To explain this approach in a simple setting, consider the black-box impossibility result of 4-round concurrent zero-knowledge of Kilian, Petrank, Rackoff [KPR98]. This impossibility result can be interpreted as showing that if we take a 4-round protocol, run sufficiently many instances of it concurrently (according to some well-specified "nesting" schedule; see Figure 1) and have the verifier generate its randomness by applying a PRF to the transcript, then there exists at least one of the instances that remains sound, even in the presence of a (fixed-input) resetting attack. In other words, we can take any 4-round protocol and transform it into a fixed-input resettably-sound one by running sufficiently many copies of the protocol according to some pre-determined schedule. Additionally, if the starting 4-round protocol also is an argument of knowledge, the resulting protocol is (fully) resettably-sound. Now, they key observation is that this "concurrent-scheduling" transformation actually preserves (resettable) witness indistinguishability: if the original protocol is (resettable) witness indistinguishable, the new one is so as well—thus, the new one is simultaneously resettable witness indistinguishable!

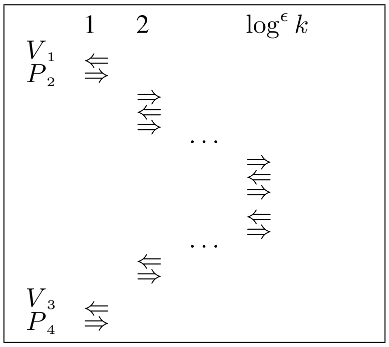

Figure 1. The nested scheduling of [KPR98].

So, if we could just come up with a 4-round resettable-witness indistinguishable argument of knowledge from one-way functions, we would be done. Unfortunately, no such protocols are known: Resettable witness-indistinguishable arguments of knowledge were first constructed based on the existence of collision-resistant hash functions in [BGGL01] and more recently based on one-way functions in [CPS13] but, while both constructions are constant-round, they have more than 4 rounds.

But, improved impossibility results for black-box concurrent zero-knowledge are known: Rosen [Ros00] presented an impossibility result for 7-round protocols and Canetti, Kilian, Petrank and Rosen [CKPR01] presented an impossibility result for $o(\log k/\log\log k)$ rounds; Chung, Pass and Tseng [CPT12] provide an alternative variant of the impossibility result of

[3] In a black-box zero-knowledge protocol we require the existence of a universal simulator S that, given only black-box access to *any* (efficient) V^*, can reproduce the view of V^* in an interaction with the honest prover.

[4] In a concurrent zero-knowledge protocol, the zero-knowledge property is required to remain intact even if the Verifier can, concurrently, start many interaction with the prover, and may arbitrarily schedule its messages between the different interactions.

[CKPR01]. Can we rely on these lower bounds instead? Indeed, Rosen's lower bound follows the same structure as the one from [KPR98] and thus it would suffice to come up with a 7-round resettably-sound witness-indistinguishable argument of knowledge.[5] The impossibility result of [CKPR01], on the other hand, doesn't simply present a scheduling that ensures resettable soundness. Rather, [CKPR01] consider a particular "aborting" verifier strategy to prove its lower-bound.

We note, however, that just as in the work of Haitner on parallel repetition [Hai09], such "random-termination" can be used as a protocol transformation. More precisely, take any (sub-logarithmic round) argument of knowledge, repeat it sufficiently many times in parallel where at each round, each of the parallel verifiers terminates, *accepting*, at random with some appropriately set probability. Each parallel verifier generates the randomness needed to decide whether to terminate or not, by applying a PRF to the current transcript. Relying on a result from Chung, Pass and Tseng [CPT12], it follows that if we appropriately fix the number of parallel repetitions and the termination probability, the resulting protocol is resettably sound as long as the number of resets queries is bounded.

A different result from [CPT12] can be used to show that the number of queries can be *amplified* to become super-polynomial by combining the above random-termination strategy with a concurrent scheduling (similar, but somewhat different, to the one in [CKPR01]).[6] We additionally observe that each of these transformations preserves (resettable) witness indistinguishability.

Combining the above, we thus have the following "soundness upgrade" lemma:

Lemma 3 (Soundness upgrade Lemma for WI)**.** *There exists a constructive protocol transformation that takes any (resettable) witness-indistinguishable argument of knowledge for $\mathcal{NP}$ with round-complexity $o(\log k/\log\log k)$ and outputs a resettable-sound, (resettable) witness-indistinguishable argument of knowledge for $\mathcal{NP}$. Additionally, if the original protocol has constant round-complexity, then resulting protocol will have round-complexity $\omega(1)$.*

Combining this theorem with the recent constant-round resettable witness-indistinguishable argument of knowledge of [CPS13] based on OWFs yields our Main Theorem 1.

rsrZK from OWF. Deng, Goyal and Sahai [DGS09] construction of rsrZK proceeded in two steps: 1) they first constructed a resettably-sound concurrent zero-knowledge protocol, and 2) they next present a generic transformation that takes any resettably-sound concurrent zero-knowledge argument and turns it into a rsrZK argument for the same language.[7] [DGS09] relied on ZAPs in both of the above steps. More recently, Bitansky and Paneth provided a construction of resettably-sound concurrent zero-knowledge for $\mathcal{NP}$ based on just one-way functions.[8] We here show how to perform the second step based on only one-way functions.

Lemma 4 (Soundness upgrade Lemma for ZK)**.** *Assume the existence of one-way functions. Then, there exists a constructive protocol transformation that takes any resettably-sound concurrent zero-knowledge argument of knowledge for $\mathcal{NP}$ and outputs a simultaneously resettable zero-knowledge argument of knowledge for $\mathcal{NP}$.*

Our protocol transformation closely follows the one in [DGS09]; our key observation is that we can simply replace ZAPs in the construction of [DGS09] with rsrWI arguments. Combining these observation with our Main Theorem 1 yields the lemma. Our Main Theorem 2 is now a direct consequence of the resettably-sound concurrent zero-knowledge protocol from [BP13] and Lemma 4.

II. Definitions and Tools

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 $\Pr_r[\ X\ ]$ the probability of an event X over coins r. The abbreviation "PPT" stands for probabilistic polynomial

[5] In an earlier version of this paper we showed that such a protocol can be constructed based on the existence of collision-resistant hash functions. As mentioned, in contrast, we here aim to provide a construction of rsrWI arguments of knowledge from just one-way functions. Another more subtle disadvantage of relying on the scheduling from [Ros00] is that it blows up the round-complexity to become polynomial; in contrast, in our solution we obtain a protocol with just $\omega(1)$ rounds.

[6] Although we haven't verifier the details, we believe we could also have relied on the [CKPR01] scheduling. However, doing so would have resulted in a polynomial (as opposed to slightly super-constant ($\omega(1)$)) round-complexity.

[7] We are slightly oversimplifying here. Their protocol from step 1 only satisfied a relaxed notion of concurrent zero-knowledge, and their second step applies also to protocols only satisfying this relaxed notion. In our context, we need not worry about this relaxed notion.

[8] In an earlier version of this paper, we also independently showed how to achieve resettably-sound "relaxed" concurrent zero-knowledge (as in [DGS09]) from one-way functions, by adapting the protocol from [CPS13]. Since the second step of the transformation applies also to such relaxed zero-knowledge protocol, this also sufficed to conclude our main result. Here, for simplicity of exposition, we instead directly appeal to the result of [BP13].

time. We will use the standard notion of computational indistinguishability [GM84].

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

Definition 5 (interactive proofs [GMR85])**.** 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 $R_L(x, w) = 1$,
$$\Pr[\ \langle P(w), V\rangle(x) = 1\ ] \geq 1 - \nu(|x|).$$
- Soundness: For every $x \notin L$ and for every interactive Turing machine P^* there exists a negligible function $\nu(\cdot)$ such that
$$\Pr[\ \langle P^*, V\rangle(x) = 1\ ] < \nu(|x|).$$

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 $\mathtt{view}^{P(w)}_{V^*(x,z)}$ 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.

Definition 6 (zero-knowledge arguments [GMR85])**.** 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 [BG92] 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 7 (arguments of knowledge [BGGL01])**.** 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,
$Pr[w \leftarrow K(\text{desc}(P^*), x) \wedge R_L(x, w) = 1] > Pr[\langle P^*, V\rangle(x) = \mathit{accept}] - \frac{1}{p(|x|)}$
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 8 (resetting adversary [CGGM00])**.** Let (P, V) be an interactive proof or argument system for a language L, $t = \mathtt{poly(k)}$, $\bar{x} = x_1, \ldots, x_t$ be a sequence of common inputs and $\bar{w} = w_1, \ldots, w_t$ the corresponding witnesses (i.e., $(x_i, w_i) \in R_L$) for $i = 1, \ldots, t$. We say that a PPT V^* is a *resetting* verifier if it concurrently interacts with an unbounded number of independent copies of P by choosing for each interaction the value i so that the common input will be $x_i \in \bar{x}$, and the prover will use witness w_i, and choosing j so that the prover will use r_j as randomness, with $i, j \in \{1, \ldots, t\}$. The scheduling or the messages to be sent in the different interactions with P are freely decided by V^*. Moreover we say that the transcript of such interactions consist of the common inputs $\bar{x}$ and the sequence of prover and verifier messages exchanged during the interactions. We refer to $\mathtt{view}^{P(\bar{w})}_{V^*(\bar{x},z)}$ as the random variable describing the content of the random tape of V^* and the transcript of the interactions between P and V^*, where z is an auxiliary input received by V^*.

Definition 9 (resettable zero knowledge [CGGM00])**.** Let (P, V) be an interactive argument system for a language L. We say that $\langle P, V\rangle$ is resettable zero knowledge (rZK) if, for any PPT resetting verifier V^* there exists a probabilistic polynomial-time algorithm S_{V^*} such that the for all pairs $(\bar{x}, \bar{w}) \in R_L$ the ensembles $\{\mathtt{view}^{P(\bar{w})}_{V^*(\bar{x},z)}\}$ and $\{S_{V^*}(\bar{x}, z)\}$ are computationally indistinguishable.

The definition of concurrent zero knowledge can be seen as a relaxation of the one of resettable zero knowledge. The adversarial concurrent verifier has the same power of the resetting verifier except it can not ask the prover to run multiple sessions with the same randomness.

Definition 10 (concurrent adversary)**.** Let (P, V) be an interactive proof or argument system for a language L, $t = \mathtt{poly(k)}$, $\bar{x} = x_1, \ldots, x_t$ be a sequence of common inputs and $\bar{w} = w_1, \dots, w_t$ the corresponding witnesses (i.e., $(x_i, w_i) \in R_L$) for $i = 1, \dots, t$. We say that a PPT V^* is a *concurrent* verifier if it concurrently interacts with an unbounded number of independent copies of P by choosing for each interaction the value i so that the common input will be $x_i \in \bar{x}$, and the prover will use witness w_i. Each copy of P runs with independent randomness. The scheduling or the messages to be sent in the different interactions with P are freely decided by V^*. Moreover we say that the transcript of such interactions consist of the common inputs $\bar{x}$ and the sequence of prover and verifier messages exchanged during the interactions. We refer to $\mathtt{view}^{P(\bar{w})}_{V^*(\bar{x},z)}$ as the random variable describing the content of the random tape of V^* and the transcript of the interactions between P and V^*, where z is an auxiliary input received by V^*.

Definition 11 (concurrent zero knowledge [DNS98])**.** Let (P, V) be an interactive argument system for a language L. We say that $\langle P, V\rangle$ is concurrent zero knowledge (cZK) if, for any PPT concurrent verifier V^* there exists a probabilistic polynomial-time algorithm S_{V^*} such that the for all pairs $(\bar{x}, \bar{w}) \in R_L$ the ensembles $\{\mathtt{view}^{P(\bar{w})}_{V^*(\bar{x},z)}\}$ and $\{S_{V^*}(\bar{x}, z)\}$ are computationally indistinguishable.

Definition 12 (witness indistinguishability [FS90])**.** 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:

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

Definition 13 (resettable WI [CGGM00])**.** Let L be a language in $\mathcal{NP}$ and R_L be the corresponding relation. An interactive argument $\langle P, V\rangle$ for L is *resettable* witness indistinguishable (rWI) if for every PPT resetting verifier V^* every $t = \mathtt{poly(k)}$, and every pair $(\bar{w}^0 = (w^0_1, \dots, w^0_t), \bar{w}^1 = (w^1_1, \dots, w^1_t))$ such that $(x_i, w^0_i) \in R_L$ and $(x_i, w^1_i) \in R_L$ for $i = 1, \dots, t$, and any auxiliary input z, the following ensembles are computationally indistinguishable:

$$\{\mathtt{view}^{P(\bar{w}^0)}_{V^*(\bar{x},z)}\} \quad and \quad \{\mathtt{view}^{P(\bar{w}^1)}_{V^*(\bar{x},z)}\}.$$

In [DN00], a construction of 2-round resettable witness-indistinguishable proof based on NIZK proofs has been shown, and then in [GOS06], a non-interactive resettable witness-indistinguishable proof has been shown by relying on specific number-theoretic assumptions.

Let us recall the definition of resettable soundness due to [BGGL01].

Definition 14 (resettably-sound arguments [BGGL01])**.** 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 $t = \mathtt{poly(k)}$ random-tapes, denoted $r_1, \dots, r_t$, for V, resulting in deterministic strategies $V^{(j)}(x) = V_{x,r_j}$defined by $V_{x,r_j}(\alpha) = V(x, r_j, \alpha)$,[9] 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 $\mathtt{poly(k)}$-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 15 (fixed-input resettably-sound arguments [PTW11])**.** 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 $x \notin L$,

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

Additionally, we will consider a further weaken notion of bounded-query fixed-input resettable soundness, where the cheating prover can reset the verifier, but is restricted to only learn a bounded number of verifier's messages in total (summed over all sessions).

Definition 16 (q-query fixed-input resettably-sound arguments [PTW11])**.** An interactive argument (P, V) for a $\mathcal{NP}$ language L with witness relation R_L is *q-query fixed-input resettably-sound* if it satisfies the following property: For all non-uniform polynomial-time adversarial resetting prover P^* that makes at most

[9] 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.

q queries to the verifier, 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 definitions of rsrWI (resp., rsrZK) simply consists in requiring both rWI (resp., rZK) and resettably sound WI (resp., rsZK) hold for the same proof system.

We stress that all our definitions and constructions do not impose any a priori bound on the number of resets and we only consider the *non-adaptive* attack models where input(s) and corrupted parties are fixed in advance before protocol executions begin.

III. $\omega(1)$-Round rsrWI AoK from OWFs

In this section, we show how to transform any constant-round argument of knowledge into an $\omega(1)$-round resettably-sound argument of knowledge by appropriately scheduling concurrent sessions of the underlying protocol, and letting the verifier generate its randomness using a PRF. Since, such concurrent scheduling preserve rWI and completeness, we have the following theorem.

Theorem 17. *Assume the existence of a constant-round rWI argument of knowledge for $\mathcal{NP}$. Then there exists an $\omega(1)$-round rsrWI argument of knowledge for $\mathcal{NP}$. More generally, the existence of a $o(\frac{\log k}{\log \log k})$-round rWI argument of knowledge for $\mathcal{NP}$ implies the existence of a* $\texttt{poly(k)}$*-round rsrWI argument of knowledge for $\mathcal{NP}$.*

Combined with the recent constant-round rWI argument of knowledge construction of Chung, Pass and Seth [CPS13] based on one-way functions, we get an $\omega(1)$-round rsrWI based on the minimal assumption of OWFs (proving our Main Theorem 1).

Theorem 18 (Main Theorem 1, restate)**.** *Assume the existence of one-way functions. Then there exists an $\omega(1)$-round rsrWI argument of knowledge for $\mathcal{NP}$.*

Proof of Theorem 17: We proceed proving Theorem 17 by combining and interpreting results proven in [CPT12], [PTW11], [BGGL01], [CPS13]. The key component of the transformation is a construction from [CPT12] (CPT), which modularizes (and improves in terms of round-complexity) the construction of [CKPR01]. The construction proceeds in two steps:

Step 1: Parallel Repetition With Random-Terminating Verifiers. Take any constant-round protocol (P, V). Repeat the protocol sufficiently many times in parallel with the follow exception: following [CKPR01], [Hai09], at each round, let each of the parallel verifier terminate, *accepting*, at random with some appropriately set probability; each parallel verifier generates the randomness needed to decide whether to terminate or not, by applying a PRF to the current transcript. CPT shows that by appropriately fixing the number of parallel repetitions and the termination probability, the resulting protocol $(\hat{P}, \hat{V})$ is k-query fixed-input resettable sound, where k is the security parameter.

Specifically (and more generally), let (P, V) be an m-round interactive argument for an $\mathcal{NP}$ language L. Define $(\hat{P}^k, \hat{V}^k)$ to be k parallel repetition of (P, V) where at each round, each of the parallel verifier terminates and accepts with probability $\rho = (1 - 1/q)$ with $q = k^{1/m}/\log^2 k$ (with randomness generated by applying a PRF to the current transcript). The following lemma is proved in CPT (but using a different language).[10]

Lemma 19 ("Lemma 7, generalized" in [CPT12])**.** *If (P, V) is sound, then $(\hat{P}^k, \hat{V}^k)$ is q-query fixed-input resettable sound.*

Step 2: Amplification Through Nesting. The second step shows how to amplify fixed-input bounded-query resettable soundness by "nesting" protocol executions. Roughly speaking, given an underlying m-round protocol $(\hat{P}, \hat{V})$ with fixed-input q-query resettable soundness, recursively define a d-level protocol $(P^{(d)}, V^{(d)})$ by executing $(\hat{P}, \hat{V})$ once, and in-between any two messages in $(\hat{P}, \hat{V})$ running an instance of $(P^{(d-1)}, V^{(d-1)})$ where $V^{(d-1)}$'s randomness is generated by applying a PRF to the transcript, and letting $(P^{(1)}, V^{(1)})$ be the "base" protocol $(\hat{P}, \hat{V})$ (without recursion calls in-between messages). The resulting protocol $(P^{(d)}, V^{(d)})$ has $O(m^d)$ rounds and CPT shows that $(P^{(d)}, V^{(d)})$ has fixed-input q^d-query resettable soundness.

More specifically, let $(\hat{P}, \hat{V})$ be an m-round interactive argument for an $\mathcal{NP}$ language L. Without loss of generality, we assume that $(\hat{P}, \hat{V})$ starts with a verifier message and end with a prover message. Also, we let $\hat{V}_r$ denote $\hat{V}$ with random tape r. For $d \in \mathbf{N}$, we define a d-level protocol $(P^{(d)}, V^{(d)})$ by defining the verifier $V^{(d)}$ recursively as follows.

- **Specifying the recursive schedule.** Simply put, each call to $V^{(d)}$ corresponds with an incarnation $\hat{V}_r$, with the additional modification that between every prover query and verifier response, $V^{(d)}$ nests a recursive call of itself, $V^{(d-1)}$, with decreased depth; since protocol $(\hat{P}, \hat{V})$ start with a

[10] There is also a slight semantical difference between the construction and lemma state here and that in CPT. More precisely, in the construction of [CPT12], the randomness used to determine termination is in fact generated by applying a q-wise independent hash function (as opposed to a PRF) to the transcript. However, as is well-known (and made explicitly in [PTW11], see e.g., Theorem 17), it follows by a standard hybrid argument that the verifier can use a PRF as described above, and the same results hold. We emphasize that this is true only since we consider *fixed-input* resettable soundness.

verifier message and end with a prover message, there will be $m-1$ nested calls.

Formally, $V^{(d)}$ starts by generating a "fresh random tape" r for $\hat{V}$ (we clarify this later) and invokes $\hat{V}_r$ with the following modification. After every prover query τ for $\hat{V}_r$ that expects a verifier response, $V^{(d)}$ delays the response from $\hat{V}_r$, and instead recursively calls in itself with decreased depth $V^{(d-1)}$. We call τ the *initiating query* for $V^{(d-1)}$. (Then $V^{(d-1)}$ invokes an incarnation $\hat{V}_{r'}$ and returns the first verifier message of $\hat{V}_{r'}$.)

When $V^{(d-1)}$ terminates, signaled by some the final prover query τ' meant for $V^{(d-1)}$ (we call τ' the *closing query* for $V^{(d-1)}$), $V^{(d)}$ makes sure that the prover has successfully "convinced" $V^{(d-1)}$ (as we will define later, this means that the prover has successfully convinced the incarnation of $\hat{V}$ invoked by $V^{(d-1)}$). At this point, $V^{(d)}$ generates a verifier response by forwarding the original initiating query τ to $\hat{V}_r$. Note that while this is $V^{(d)}$'s response to the closing query τ', the response depends only on the randomness r and the initiating query τ (because it is in fact $\hat{V}_r$'s response to τ).

When $V^{(d)}$ receives the closing query for $\hat{V}_r$, it accepts if and only if the prover has successfully convinced $\hat{V}_r$.

- **Specifying the base case of the recursion.** When $d = 1$, $V^{(1)}$ simply invokes a corresponding incarnation $\hat{V}_r$ without recursive calls (i.e., $V^{(1)}$ answers each prover query without delay).

To help understand the definition, we note that for an incarnation $\hat{V}$, the first $m-1$ round prover message queries are initiation queries, and the last round prover message queries are closing queries. Also, all except for the first round verifier messages of $\hat{V}$ are responses to initiating queries.

It remains to define how each recursive call of $V^{(d)}$ generates the randomness r for its own incarnation of $\hat{V}$. Just as before, $V^{(d)}$ generates r using a (global) PRF. More precisely, $V^{(d)}$ additionally takes as input a PRF f (which is sampled at the beginning of the protocol), and in each recursive call of $V^{(d)}$, corresponding to an initiating query τ, $V^{(d)}$ generates r by apply f to τ (note that the query τ is just the current global transcript of interaction right when $V^{(d)}$ is about to spawn $\hat{V}$, and that the whole random tape of $\hat{V}$ is generated).

This completes the description of the transformation in CPT, and the following lemma is proved there (again stated in a different language).[11]

Lemma 20 (Lemma 9 in [CPT12]). *For every $q \geq m$ and d, if $(\hat{P}, \hat{V})$ is q-query fixed-input resettably sound, then $(P^{(d)}, V^{(d)})$ is q^d-query fixed-input resettable sound.*

So, by combining the above two steps, if we let $d = \omega(1)$, we turn any constant-round argument (P, V) into an $\omega(1)$-round fixed-input (unbounded query) resettably sound argument. (More generally, as long as the original protocol has $m = o(\frac{\log k}{\log\log k})$ rounds, the parallel protocol obtained in Step 1 is q-query resettably sound for $q = O(k^{1/m}) = m^{\omega(1)}$. A $\texttt{poly}(\texttt{k})$-round single-instance (unbounded query) resettably sound argument can be obtained by properly choosing d so that $q^d = k^{\omega(1)}$ while $O(m^d) = \texttt{poly}(\texttt{k})$.)

We notice that the transformation preserves rWI since any attack of a resetting verifier on the transformed scheme is also a legitimate attack to the original scheme.

Finally, Theorem 17 follows by additionally applying the following lemma in [CPS13] (which relies on the technique from [BGGL01]), which shows that any rWI *argument of knowledge* satisfying fixed-input resettable soundness can be transformed into one that satisfies the full-fledged one, while preserving rWI (or any other secrecy property against malicious verifiers.

Lemma 21 ([CPS13]). *Let (P, V) be a fixed-input resettably sound rWI (resp., ZK or rZK) argument of knowledge for a language $L \in \mathcal{NP}$. Then there exists a protocol (P', V') that is a resettably-sound rWI (resp., ZK or rZK) argument of knowledge for L with the same round complexity as (P, V).*

■

IV. Simultaneously Resettable ZK from OWFs

In this section we prove the second main theorem of this work, namely: the existence of OWFs implies the existence of a rsrZK argument of knowledge for $\mathcal{NP}$. We start by recalling a protocol transformation from [DGS09], [GS08] (DGS).

Lemma 22 ([DGS09], [GS08]). *Assume the existence of ZAPs. Then, there exists a constructive protocol transformation* DGS *that takes any resettably-sound concurrent zero-knowledge argument $\Pi = (P, V)$ and outputs a protocol $\mathsf{DGS}^{\Pi} = (P', V')$ that is fixed-input resettably-sound, resettable zero-knowledge.*

We first observe that in their proof, the only properties they rely on from ZAPs is that they (can be made) to satisfy both resettable-soundness and resettable WI—that is, they are rsrWI. So we can simply replace the use of ZAPs with an rsrWI in their protocol.

Lemma 23 ([DGS09], [GS08]). *Assume the existence*

[11] Again, in the construction of [CPT12], the randomness of incarnations of $\hat{V}$ is in fact generated by a many-wise independent hash function (as opposed to a PRF). But, as mentioned above, it follows by a standard hybrid argument that the verifier can use a PRF instead.

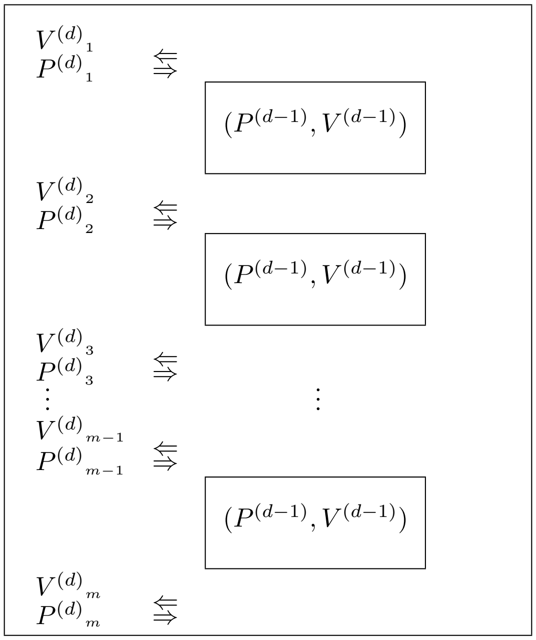

Figure 2. An execution of the $O(m^d)$-round protocol $(P^{(d)}, V^{(d)})$.

of an rsrWI for $\mathcal{NP}$. Then, there exists a constructive protocol transformation DGS *that takes any resettably-sound concurrent zero-knowledge argument* $\Pi = (P, V)$ *and outputs a protocol* $\mathsf{DGS}^{\Pi} = (P', V')$ *that is fixed-input resettably-sound, resettable zero-knowledge.*

Additionally, the soundness proof from DGS implicitly proves the following lemma.

Lemma 24 (implicit in [DGS09], [GS08])**.** *Given any efficient cheating prover P'^* for the transformed protocol* DGS^{Π}*, there exists an efficient cheating prover P^* for the original protocol Π that succeeds with negligibly close probability.*

As a consequence, if Π is an argument of knowledge, then so is DGS^{Π}; that is, the DGS transformation preserves the argument of knowledge property. We summarize the above two observation in the following lemma.

Combining the above two lemmas, we thus have:

Lemma 25. *Assume the existence of an rsrWI for $\mathcal{NP}$. Then, there exists a constructive protocol transformation* DGS *that takes any resettably-sound concurrent zero-knowledge argument $\Pi = (P, V)$ and outputs a protocol $\Pi' = \mathsf{DGS}^{\Pi}$ that is fixed-input resettably-sound, resettable zero-knowledge. Additionally, if Π is an argument of knowledge, so is Π'.*

Therefore, if we start with a resettably-sound concurrent ZK protocol Π that is an argument of knowledge, we can additionally appeal to Lemma 21 after applying the DGS transformation to obtain a "full-fledged" (as opposed to fixed-input) simultaneously resettable zero knowledge argument of knowledge. Combined with our Theorem 18 (i.e., the construction of rsrWI from one-way functions), we thus have the following theorem.

Theorem 26. *Assume the existence of one-way functions. Then, there exists a constructive protocol transformation that takes any resettably-sound concurrent zero-knowledge argument of knowledge and outputs a simultaneously resettable zero-knowledge argument of knowledge.*

Finally, our Main Theorem 2 follows directly by combining Theorem 26 with the resettably sound concurrent ZK argument of knowledge of Bitansky and Paneth [BP13] (which is based on one-way functions):

Theorem 27 (Main Theorem 2, restated)**.** *Assume the existence of one-way functions. Then there exists an rsrZK for $\mathcal{NP}$.*

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 Ostrovsky's NSF grants, by the EU through the FP7 programme under contract 216676 ECRYPT II, and 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.

We thank Yuval Ishai for several helpful discussions.

REFERENCES

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *FOCS '01*, pages 106–115, 2001.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *J. Comput. Syst. Sci.*, 37(2):156–189, 1988.

[BG92] Mihir Bellare and Oded Goldreich. On defining proofs of knowledge. In *CRYPTO '92*, pages 390–420, 1992.

[BGGL01] Boaz Barak, Oded Goldreich, Shafi Goldwasser, and Yehuda Lindell. Resettably-sound zero-knowledge and its applications. In *FOCS'02*, pages 116–125, 2001.

[BOV12] Joshua Baron, Rafail Ostrovsky, and Ivan Visconti. Nearly simultaneously resettable black-box zero knowledge. In *ICALP 2012s, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 88–99. Springer, 2012.

[BP12] Nir Bitansky and Omer Paneth. From the impossibility of obfuscation to a new non-black-box simulation technique. In *FOCS*, 2012.

[BP13] Nir Bitansky and Omer Paneth. On the impossibility of approximate obfuscation and applications to resettable cryptography. In *STOC*, 2013.

[CGGM00] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In *STOC '00*, pages 235–244, 2000.

[CKPR01] Ran Canetti, Joe Kilian, Erez Petrank, and Alon Rosen. Black-box concurrent zero-knowledge requires $\tilde{\omega}(\log n)$ rounds. In *STOC '01*, pages 570–579, 2001.

[CPS13] Kai-Min Chung, Rafael Pass, , and Karn Seth. Non-black-box simulation from one-way functions and applications to resettable security. In *STOC*. ACM, 2013.

[CPT12] Kai-Min Chung, Rafael Pass, and Wei-Lung Dustin Tseng. The knowledge tightness of parallel zero-knowledge. In *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, volume 7194 of *Lecture Notes in Computer Science*, pages 512–529. Springer, 2012.

[CPV04a] Giovanni Di Crescenzo, Giuseppe Persiano, and Ivan Visconti. Constant-round resettable zero knowledge with concurrent soundness in the bare public-key model. In *CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 237–253. Springer, 2004.

[CPV04b] Giovanni Di Crescenzo, Giuseppe Persiano, and Ivan Visconti. Improved setup assumptions for 3-round resettable zero knowledge. In *ASIACRYPT 2004*, volume 3329 of *Lecture Notes in Computer Science*, pages 530–544. Springer, 2004.

[DGS09] Yi Deng, Vipul Goyal, and Amit Sahai. Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 251–260. IEEE, 2009.

[DN00] Cynthia Dwork and Moni Naor. Zaps and their applications. In *In 41st FOCS*, pages 283–293. IEEE, 2000.

[DNS98] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. In *STOC '98*, pages 409–418. ACM, 1998.

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

[GK96] Oded Goldreich and Hugo Krawczyk. On the composition of zero-knowledge proof systems. *SIAM Journal on Computing*, 25(1):169–192, 1996.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In *STOC '85*, pages 291–304. ACM, 1985.

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

[GOS06] Jens Groth, Rafail Ostrovsky, and Amit Sahai. Non-interactive zaps and new techniques for nizk. In *Advances in Cryptology – CRYPTO 06*, volume 4117 of *Lecture Notes in Computer Science*, pages 97–111. Springer, 2006.

[GS08] Vipul Goyal and Amit Sahai. Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. cryptology eprint archive, report 2008/545, 2008. http://eprint.iacr.org/.

[Hai09] Iftach Haitner. A parallel repetition theorem for any interactive argument. In *FOCS*, pages 241–250, 2009.

[KPR98] Joe Kilian, Erez Petrank, and Charles Rackoff. Lower bounds for zero knowledge on the internet. In *FOCS '98*, pages 484–492, 1998.

[MR01] Silvio Micali and Leonid Reyzin. Soundness in the public-key model. In *Advances in Cryptology – Crypto '01*, volume 2139 of *Lecture Notes in Computer Science*, pages 542–565. Springer-Verlag, 2001.

[OW93] Rafail Ostrovsky and Avi Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *Theory and Computing Systems, 1993*, pages 3–17, 1993.

[PTW11] Rafael Pass, Wei-Lung Dustin Tseng, and Douglas Wikström. On the composition of public-coin zero-knowledge protocols. *SIAM J. Comput.*, 40(6):1529–1553, 2011.

[Ros00] Alon Rosen. A note on the round-complexity of concurrent zero-knowledge. In *CRYPTO '00*, pages 451–468, 2000.

[YZ07] Moti Yung and Yunlei Zhao. Generic and practical resettable zero-knowledge in the bare public-key model. In *EUROCRYPT*, pages 129–147, 2007.