Resettable Statistical Zero Knowledge

Sanjam Garg[1], Rafail Ostrovsky[1,2], Ivan Visconti[3], and Akshay Wadia[1]

[1] Department of Computer Science, UCLA, USA
{sanjamg,rafail,awadia}@cs.ucla.edu
[2] Department of Mathematics, UCLA, USA
[3] Dipartimento di Informatica, University of Salerno, Italy
visconti@dia.unisa.it

Abstract. Two central notions of Zero Knowledge that provide strong, yet seemingly incomparable security guarantees against malicious verifiers are those of Statistical Zero Knowledge and Resettable Zero Knowledge. The current state of the art includes several feasibility and impossibility results regarding these two notions *separately*. However, the question of achieving Resettable Statistical Zero Knowledge (i.e., Resettable Zero Knowledge and Statistical Zero Knowledge *simultaneously*) for non-trivial languages remained open. In this paper, we show:

- Resettable Statistical Zero Knowledge with unbounded prover: under the assumption that sub-exponentially hard one-way functions exist, $\mathrm{r}\mathcal{SZK} = \mathcal{SZK}$. In other words, every language that admits a Statistical Zero-Knowledge ($\mathcal{SZK}$) proof system also admits a Resettable Statistical Zero-Knowledge ($\mathrm{r}\mathcal{SZK}$) proof system. (Further, the result can be re-stated unconditionally provided there exists a sub-exponentially hard language in $\mathcal{SZK}$). Moreover, under the assumption that (standard) one-way functions exist, all languages L such that the complement of L is random self reducible, admit a $\mathrm{r}\mathcal{SZK}$; in other words: co-$\mathcal{RSR} \subseteq \mathrm{r}\mathcal{SZK}$.
- Resettable Statistical Zero Knowledge with efficient prover: efficient-prover Resettable Statistical Zero-Knowledge proof systems exist for all languages that admit hash proof systems (e.g., QNR, QR, $\mathcal{DDH}$, DCR). Furthermore, for these languages we construct a two-round resettable statistical witness-indistinguishable argument system.

The round complexity of our proof systems is $\tilde{O}(\log \kappa)$, where κ is the security parameter, and all our simulators are *black-box*.

1 Introduction

The notion of a Zero-Knowledge (ZK, for short) Proof System introduced by Goldwasser, Micali and Rackoff [19] is central in Cryptography. Since its introduction, the concept of a ZK proof has been extremely influential and useful for many other notions and applications (e.g., multi-party computation [18], CCA encryption [27]). Moreover, the original definition has been then extended under several variations, trying to capture additional security guarantees. Well

R. Cramer (Ed.): TCC 2012, LNCS 7194, pp. 494–511, 2012.

known examples are the notions of non-malleable ZK [14] introduced by Dolev, Dwork and Naor, which concerns security against man-in-the-middle attacks, of ZK arguments introduced by Brassard, Chaum and Crepeau [4] where soundness is guaranteed only with respect to probabilistic polynomial-time adversarial provers, and of concurrent ZK [16] introduced by Dwork, Naor and Sahai, which concerns security against concurrent malicious verifiers. Another important variant is that of Statistical Zero Knowledge [19,3,33], where it is guaranteed that a transcript of a proof will remain zero knowledge even against computationally unbounded adversaries.

An important model of security against malicious verifiers, known as *Resettable Zero-Knowledge*, was introduced by Canetti, Goldreich, Goldwasser and Micali in [5]. In this setting, the malicious verifier is allowed to *reset* the prover, and make it re-use its randomness for proving new theorems. Indeed, one of the main motivations for studying resettable ZK was to understand the consequences of re-using limited randomness on the zero-knowledge property. In [5], it was shown that *computational* zero-knowledge for all of $\mathcal{NP}$ is possible even in this highly adversarial setting. Although resettable zero knowledge has received considerable attention since its inception (see for example [1,24,13,39,12,8,35]), almost all the work has been focused on the computational setting.

In this work, we continue the line of research on resettable ZK by investigating the question of resettability when the zero-knowledge property is required to be statistical, i.e., Resettable Statistical Zero Knowledge. This model constrains the prover strategy severely: not only should the prover somehow re-use its limited randomness, it must do so in a way that makes the transcript of the proof statistically secure. Known solutions in the setting of computational resettable ZK involve converting prover's bounded randomness to unbounded pseudo-randomness by using pseudo-random functions (PRF). However, this approach fails in our case, as an unbounded adversary can break the PRF and gain critical information, breaking zero knowledge. In this paper, we develop a new technique to handle this problem. Using this technique, we study resettable statistical zero knowledge in the form of following two *distinct* questions.

- Do there exist *efficient-prover* resettable statistical ZK proofs? This question is motivated by practical applications of resettable ZK, for example, in smart cards. If a prover is to be implemented in a small device like a smart card, it is essential that the prover strategy is polynomial-time.
- What languages in $\mathcal{SZK}$ have resettable statistical ZK proofs? The class $\mathcal{SZK}$ is the class of problems which admit statistical zero-knowledge proofs. This question is purely theoretical in nature, and tries to ascertain the difficulty of achieving resettability where statistical zero-knowledge already exists. In this setting we consider prover's which are forced into giving multiple proofs using the same *limited* random coins. This work can be thought of a natural extension of the recent work on Concurrent Statistical Zero-Knowledge (c$\mathcal{SZK}$) [25,30].

1.1 Our Contribution

In this paper we address the above questions and present the following results. We stress that our techniques may be of independent interest.

Resettable Statistical Zero Knowledge with efficient prover. We show the existence of *efficient-prover* resettable statistical ZK proof systems for all languages in $\mathcal{SZK}$ that admit hash proof systems [10] (e.g., Quadratic Non-Residuosity (QNR), Decisional Diffie-Hellman ($\mathcal{DDH}$), Decisional Composite Residuosity (DCR)). Therefore, our techniques show that *efficient-prover* resettable statistical ZK proof systems also exist for non-trivial languages (like $\mathcal{DDH}$) where each instance is associated to more than one witness, where intuitively reset attacks are harder to deal with.[1] Furthermore, using our techniques, for these languages we also construct a two-round resettable statistical witness-indistinguishable argument system.

Resettable Statistical Zero Knowledge with unbounded prover. We show that if a family of sub-exponentially hard one-way functions exists then r$\mathcal{SZK}$ = $\mathcal{SZK}$, i.e., all languages that admit a statistical ZK proof systems also admit a resettable statistical ZK proof system. If there exists an $\mathcal{SZK}$ language L which is (worst-case) sub-exponentially hard for all input length[2] then r$\mathcal{SZK}$ = $\mathcal{SZK}$ without any additional assumptions, as it already implies the existence of sub-exponentially hard one-way functions [29]. Informally, a sub-exponentially hard one-way function is a one-way function that is secure against sub-exponential (2^{κ^ϵ} for some $0 < \epsilon < 1$) size circuits. Moreover, we show that if a family of (standard) one-way functions exists (or, if there are languages which are hard on the average and admit statistical zero-knowledge proofs [29]) then co-$\mathcal{RSR} \subseteq$ r$\mathcal{SZK}$. Our results are achieved through a novel use of instance-dependent (ID, for short) commitment schemes, a new simulation technique, and a coin-tossing protocol that is secure under reset attacks that we build on top of a new ID commitment for all $\mathcal{SZK}$.

Our simulators are *black-box* and the round complexity of all our constructions is $\tilde{O}(\log \kappa)$ which is optimal considering the lower bounds achieved so far for black-box concurrent ZK [6,26].

We stress that since the very introduction in [5] of the notion of resettable ZK, our results are the first in establishing Resettable *Statistical* Zero Knowledge.

[1] When there are multiple witnesses that can prove membership of an instance in a language, in a reset attack we allow the adversarial verifier to force the prover to reuse the same randomness for proving the same instance but using a different witness. We therefore achieve a stronger definition of resettability than the one used in previous work.

[2] If there exists a language $L \in \mathcal{SZK}$ such that for infinite sequence of input lengths, the worst-case decision problem for L is sub-exponentially-hard, Ostrovsky showed that there exists a non-uniform sub-exponentially hard one-way functions for that sequence of input length [29].

We finally leave open an interesting question of proving that $\mathcal{SZK} = \mathrm{r}\mathcal{SZK}$ unconditionally or under relaxed complexity-theoretic assumptions and of establishing whether resettable statistical ZK *arguments* are achievable for all $\mathcal{NP}$.

As a final note, we remark upon the complexity of the verifiers in our protocols. Historically, the notion of $\mathcal{SZK}$ was developed with bounded verifiers (and unbounded distinguishers), for example, see [3,37]. Moving in the same direction, we obtain our results in this model, where the verifiers are computationally bounded. In subsequent literature on $\mathcal{SZK}$, the stronger notion of statistical zero-knowledge against *unbounded* verifiers was developed. In this scenario, the notion of resettability seems hard to achieve: unbounded verifiers can compute statistical correlations on the fly by making multiple reset queries to the prover. We leave the question of constructing such protocols or showing impossibility in a setting with unbounded verifiers as an open problem for future work.

1.2 Technical Difficulties and New Techniques

We begin by asking the general question: "Why is the problem of constructing resettable statistical zero-knowledge proof systems hard?" The problem lies in the fact that the prover has limited randomness and can be reset. Therefore, prover's messages are essentially a deterministic function of the verifier's messages, and the verifier can probe this function by resetting the prover and thereby obtaining information that might be useful for an unbounded distinguisher. We highlight the issues by demonstrating why existing techniques fail. The most well studied way of achieving resettable *computational* zero-knowledge proofs [5], is by using a pseudorandom function. In particular, very informally, using this technique the prover applies a pseudorandom function on the common input and the verifier's first messages (this message is called the *determining message*), which fixes all future messages of verifier, and uses the output as its random tape. Now, when the verifier resets and changes its determining message, prover's random tape changes, and thus, intuitively, the verifier does not gain any advantage by resetting the prover. However, for our goal of obtaining resettable *statistical* zero knowledge, this approach is not sufficient. In fact, intuitively, any protocol (as far as we know) in which there exists a message computed using both the witness and the randomness, where the randomness is fixed but the witnesses could change with theorem statements, can not be statistically "secure" in presence of reset attacks. Indeed, an adversarial verifier could interact multiple times with provers that use a fixed randomness but different statements and witnesses. This information can be used by an unbounded distinguisher to establish certain correlations among the values used in different executions, ultimately breaking the statistical ZK property. Because of these restrictions, previously known techniques, which were sufficient for resettable [5,1] and statistical ZK [25,20,21] independently, turn out to be insufficient for achieving both of them simultaneously.

In light of the intuition above resettable statistical ZK for non-trivial languages at first sight might be considered impossible to achieve. But, on the contrary we develop a new technique that overcomes the above problems.

We demonstrate this new technique by first considering a *toy version* of our protocol. The protocol consists of three phases. In the first phase the verifier sends a "special" *instance-dependent non-interactive* (ID, for short) *commitment* of a random string m to the prover. (In this commitment, if the prover is lying and $x \notin L$, then m will be undefined, while if $x \in L$, then m will be unique.) The second phase consists of a *PRS preamble* [32]. Very roughly, in the PRS preamble the verifier commits to random shares of m, which are opened depending on the provers challenges. Finally, the prover is required to send m to the verifier. The prover can obtain m by extracting it from the commitment either efficiently using a witness in case of efficient-prover proofs, or running in exponential time in case of unbounded-prover proofs. We stress that when the theorem being proved is true the message m that can be extracted is unique.

First, the protocol just described has the following property: every message sent by the prover is *public coin*[3] except its last message, which is *uniquely* determined by the first message of the verifier (we use [5] terminology and refer to it as the *determining message*). Most importantly, no message depends on the witness of the prover. It is this property that allows a simulator to generate a transcript that is statically close to the transcript generated in the interaction with a real prover. An honest prover uses a pseudorandom function on the common input and the determining message and uses the output as its random tape. A simulator can sample the messages from the *same* distribution as the real prover. Finally, the simulator will be able to obtain m by using rewinding capabilities, through a variation of a PRS rewinding strategy [32]. The need for the variation arises from the fact that a simulator that uses pseudo-random coins does not gain anything by rewinding (i.e., after a rewind it would re-send the same message). We deal with this problem by having the simulator use pseudorandom coins for some messages while using pure random coins for others. We elaborate on this in § 4. This toy version, described above, illustrates the key ideas that we use in achieving simultaneously both resettable and statistical zero knowledge. To transform our toy version into a full proof system, for even the most basic languages that we consider in this paper, we need an extra instance-dependent primitive. But we defer this discussion to § 3 and § 5.

Second, our protocol also has the property that if the theorem is false then the prover has almost no chance (in the information-theoretic sense) of sending an accepting last message. This follows from the fact that the ID commitment from verifier is statistically hiding. This property guarantees soundness.

Unfortunately, the above ideas are insufficient to prove that r$\mathcal{SZK} = \mathcal{SZK}$. This is because statistically hiding non-interactive ID commitments, introduced by Chailloux, Ciocan, Kerenidis and Vadhan [7] for $\mathcal{SZK}$ are "honest-sender." To force the sender into using purely random coins we need a coin-flipping protocol secure against resetting senders. For this coin-flipping protocol an *ID commitment* scheme which is *computationally* binding with respect to a resetting sender for instances in the language and statistically hiding for instances not in the language, suffices. We will use some techniques introduced by Barak, Goldreich,

[3] Looking ahead, we will use a pseudorandom function to generate these messages.

Goldwasser and Lindell in [1] on top of a previous result of Ong and Vadhan [28] for obtaining such an ID commitment scheme.

However the more subtle problem arises in the use of pseudorandom functions. To obtain security against reset attacks, the coin-flipping message played by the receiver of the commitment must be computed by using a pseudorandom function. This again turns out to be insufficient for our analysis since the use of the pseudorandom function does not guarantee that the outcome of the coin-flipping protocol is a uniform string to be used in the honest-sender non-interactive ID commitment scheme. In order to solve this additional problem, we use sub-exponentially hard pseudorandom functions (constructed from sub-exponentially hard one-way functions). These stronger primitives have the additional property that they are secure against sub-exponential size circuits. This technique is referred to as *complexity leveraging*, and has been previously used in various applications (e.g., [5,23,2,11,9,31,38]). However, we stress that in all our constructions, the simulator runs in expected polynomial time, and the above assumptions play a role only inside our security proof.

Before concluding this section, we point out an important difference between our approach and ideas developed by Micciancio, Ong, Sahai and Vadhan in [25], where the authors give *unconditional* constructions of concurrent statistical zero-knowledge proofs for many non-trivial problems. Like their construction we use similar ID commitments but our general approach and overall protocol is different from their approach. In [25], a compiler is constructed that (using ID commitments) provides a generic way to construct statistical zero-knowledge protocols. But, as pointed earlier, such a compiling technique along with standard resettability techniques [5] is not sufficient for us. Therefore, we develop our zero-knowledge protocol from scratch. This is needed because obtaining resettability along with statistical zero knowledge is different and (as pointed earlier) harder than obtaining concurrent statistical zero knowledge. We further note that in fact our techniques imply that $\mathcal{SZK} = \mathrm{c}\mathcal{SZK}$ unconditionally. We refer the reader to the full version [17] for further discussion on this.

Road map. We start by giving some preliminary definitions in § 2. We use three ID primitives in this paper. We elaborate on those in § 3. In § 4 we construct a resettable statistical ZK proof secure against partially honest verifiers. Then in § 5 we remove this limitation for certain classes of languages. In § 6, we construct the proof system that works for all language in $\mathcal{SZK}$.

2 Notation and Tools

We say that a function is *negligible* in the security parameter κ if it is asymptotically smaller than the inverse of any fixed polynomial. Otherwise, the function is said to be *non-negligible* in κ. We say that an event happens with *overwhelming* probability if it happens with a probability $p(\kappa) = 1 - \nu(\kappa)$ where $\nu(\kappa)$ is a negligible function of κ. In this section, we provide an overview of the primitives used in this paper. Formal definitions can be found in the full version [17].

Resettable/Statistical Zero Knowledge. In this paper we consider resettable [5] and statistical [19,3,33] notions of zero-knowledge. The notion of resettability requires that a protocol remains zero-knowledge even if the verifier can reset the prover. The notion of statistical zero knowledge provides security guarantees against unbounded distinguishers. This paper constructs resettable statistical zero-knowledge proof systems. In other words we try to achieve both the resettability and the statistical guarantees simultaneously.

PRS Preamble from [32]. A PRS preamble is a protocol between a committer $\mathcal{C}$ and a receiver $\mathcal{R}$ that consists of two main phases, namely: 1) the commitment phase, and 2) the challenge-response phase.

Let k be a parameter that determines the round-complexity of the protocol. Then, in the commitment phase, very informally, the committer commits to a secret string σ and k^2 pairs of its 2-out-of-2 secret shares. The challenge-response phase consists of k iterations, where in each iteration, very informally, the committer "opens" k shares, one each from k different pairs of secret shares as chosen by the receiver.

The goal of this protocol is to enable the simulator to be able to rewind and extract the "preamble secret" σ with overwhelming probability. In the concurrent setting, rewinding can be difficult since one may rewind to a time step that precedes the start of some other protocol [16]. However, as it has been demonstrated in [32], there is a fixed "time-oblivious" rewinding strategy that the simulator can use to extract the preamble secrets from every concurrent cheating committer, except with negligible probability. Moreover this works as long as $k = \tilde{\Omega}(\log \kappa)$ for some positive ϵ. We refer to this as the PRS rewinding strategy or the PRS simulation strategy. We refer the reader to [32] for more details.

Sub-exponentially hard one-way functions. A sub-exponentially hard one-way function is a one-way function that is hard to invert even by sub-exponential (2^{κ^ϵ} for some $1 > \epsilon > 0$) size circuits. They imply the existence of *sub-exponentially hard pseudorandom functions*. We stress that we need this assumption only for proving that $\mathcal{SZK} = \mathrm{r}\mathcal{SZK}$.

3 Instance-Dependent Commitments and Proofs

In this section we construct three instance-dependent primitives, that we use in this paper: (1) a non-interactive instance-dependent commitment scheme, (2) an interactive instance-dependent commitment scheme, and finally (3) an instance-dependent argument system.

Non-Interactive Instance-Dependent Commitment Scheme. An important tool that we will re-define, construct and use in our proof systems, is that of "special" *non-interactive instance-dependent* (ID, for short) *commitment schemes*. A commitment scheme allows one party (referred to as the *sender*) to commit to a value while keeping it *hidden*, with the ability to *reveal* the committed value

later. Commitments also have the property that once the sender commits to a value, it can not change its mind later. This property is refereed to as the *binding* property. In certain settings, commitment schemes for which these properties are not required to hold simultaneously, suffice. Such schemes are parameterized by a value x and a language L and either the binding or the hiding property holds depending upon the membership of x in L. These schemes are referred to as ID commitment schemes [7]. Typically, the ID commitment schemes that have been considered in the literature require hiding property to hold when $x \in L$ and binding property to hold otherwise. We actually need the reverse properties, i.e., we need hiding property when $x \notin L$ and binding property otherwise.

In particular we consider an ID commitment scheme with further special properties. We require that the commitment scheme be statistically binding for $x \in L$ and statistically hiding otherwise. In other words we want binding and hiding properties to hold against unbounded adversaries. Also we require that our ID commitment scheme be secure against a resetting sender. This always holds when the commitment scheme is *non-interactive*. All the non-interactive ID commitments that we consider are statistically hiding. So to simplify notation we refer to a non-interactive instance-dependent commitment scheme with perfect (binding holds with probability 1) binding and statistical hiding as a *perfect non-interactive ID commitment*. Similarly, we refer to a non-interactive instance-dependent commitment scheme with statistical binding and statistical hiding as a *statistical non-interactive ID commitment*.

Since the commitment is statistically binding, when $x \in L$, the committed value can always (with overwhelming probability) be extracted in exponential time. Extractability instead becomes tricky when the extractor has to run in polynomial time. We will call an ID commitment scheme *efficiently extractable* if when $x \in L$ then there exists an extractor that takes as input a witness for the membership of x in L and the commitment, and outputs in polynomial-time the committed message.

It turns out that perfect non-interactive ID commitment schemes are actually known to exist for all languages in co-$\mathcal{RSR}$ [22,36,34]. co-$\mathcal{RSR}$ is the class of languages such that the complement of each of these languages is random self-reducible. Another class of languages that is amenable to our techniques is the class of languages that are in $\mathcal{SZK}$ and that admit a hash proof system. Observing that these languages imply instance-dependent primitives that are analogous to ID commitments described above, we get efficient-prover resettable statistical ZK proof systems for this interesting class. In particular, for $\mathcal{DDH}$ (the language that consists of all Diffie-Hellman quadruples and that admits two different witnesses for proving the membership of a quadruple to the language), we give a separate ID commitment scheme highlighting how our techniques work with multiple witnesses.

We notice that for the whole $\mathcal{SZK}$ we only know a weak form of statistical non-interactive ID commitment scheme where statistical binding holds with respect to honest senders only. The details have been provided in the full version.

We will denote the extractable perfect (or, statistical) non-interactive ID commitment scheme by $\mathcal{COM}$. The commitment function for a value x and language L will be denoted by $\mathsf{C}_{L,x}$. Also we use the notation $\mathsf{C}_{L,x}(m;r)$ for the function used to generate the commitment to message $m \in \{0,1\}^{\ell_0}$ using random coins $r \in \{0,1\}^{\ell_1}$. The extractability property of these commitments is very important for our constructions.

Interactive ID Commitment Scheme. We use an *interactive ID commitment scheme* $\mathcal{COM}_{L,x} = (S_x, R_x)$, where S_x and R_x are the sender and the receiver respectively, with common input x. This ID commitment scheme is *computationally binding* against a *resetting* sender when the instance x is in the language, and is *statistically hiding* otherwise. Very roughly, we construct such a scheme by using the constant-round public-coin ID commitment scheme of [28]. This scheme has statistical binding and statistical hiding properties. We make it secure under resetting senders by having the receiver determine its messages by applying a pseudo-random function (similarly to Proposition 3.1 in [1]) to the transcript so far. Because of this, the statistical binding property is degraded to computational[4] binding. We stress that unlike the non-interactive ID commitment described earlier, we will not need any extractability from these commitments. We obtain this new ID commitment scheme for all of $\mathcal{SZK}$ under the assumption that one-way functions exist. The details have been provided in the full version.

Instance-Dependent Argument System $\langle \mathsf{PrsSWI}_x, \mathsf{VrsSWI}_x \rangle$. We will need an instance-dependent argument system $\langle \mathsf{PrsSWI}_x, \mathsf{VrsSWI}_x \rangle$ where PrsSWI_x and VrsSWI_x are the prover and the verifier respectively, with common inputs x and a statement ξ. When[5] x is in the language, we want that $\langle \mathsf{PrsSWI}_x, \mathsf{VrsSWI}_x \rangle$ be a resettably sound argument of knowledge for $\mathcal{NP}$. In this case, very roughly, $\langle \mathsf{PrsSWI}_x, \mathsf{VrsSWI}_x \rangle$ has the additional property that the soundness holds even when the prover can reset the verifier. If instead x is not in the language then $\langle \mathsf{PrsSWI}_x, \mathsf{VrsSWI}_x \rangle$ must be statistical witness indistinguishable. We construct this argument system by instantiating Blum's Hamiltonicity protocol with the constant-round public-coin ID commitment scheme of [28]. We make it resettably sound by using a pseudorandom function [1]. Details, definition and constructions are given in the full version.

[4] However, looking ahead we note that, computational binding will be sufficient for our applications since the role of the sender will be played by a polynomially bounded party.

[5] In general, in proof systems when an ID commitment is used, it is parameterized by the theorem statement ξ being proven. In our case the ID commitment is actually parameterized by a different value x. Looking ahead, x would be the theorem statement of an interactive proof system that uses the sub-protocol $\langle \mathsf{PrsSWI}_x, \mathsf{VrsSWI}_x \rangle$ to prove the $\mathcal{NP}$ statement ξ.

4 Resettable Partially Honest-Verifier Statistical Zero Knowledge

We aim at constructing a resettable statistical zero-knowledge proof system. We start by building a simpler protocol which is resettable statistical zero knowledge only against a restricted class of adversarial verifiers. In subsequent sections, we build upon this simpler protocol to achieve our general results. The adversarial verifiers that we consider here are restricted to "act honestly" but only in a limited manner. We call such verifiers *partially honest*. As pointed out in § 3, we use a non-interactive ID commitment scheme. Looking ahead, in our protocol this commitment is used by the verifier to commit to certain messages. A partially honest verifier is required to behave honestly when computing the commitment function, using pure randomness to commit to messages. Besides this it can cheat in any other way. We state this restriction more concretely after we have described the protocol.

We begin by construction a *concurrent* statistical zero-knowledge proof system secure against such partially honest adversaries, and then transform it into a resettable statistical zero-knowledge proof system under the same restricted class of adversarial verifiers.

Concurrent Partially Honest-Verifier SZK. We start by informally describing the protocol cpHSZK of Fig. 1. It consists of three phases. The first phase, called the *Determining Message Phase*, consists of the verifier sending a commitment to a string m to the prover. We use the extractable non-interactive ID commitment scheme described earlier. The second phase is roughly a PRS preamble [32] and we refer to it as the *PRS Phase*. Note that some commitments are made in the PRS preamble, but we lump these with the commitment to m, in the Determining Message Phase itself. Finally the prover sends to the verifier the value m. This is referred to as the *Final Message*. An adversarial verifier, denoted by V^*, is called a partially honest verifier if it generates the non-interactive ID commitments of the Determining Message Phase "honestly." This requires that these ID commitments are: (1) "well-formed" and (2) have unique[6] openings (except with negligible probability).

We begin by briefly sketching why cpHSZK is a concurrent statistical zero-knowledge proof system for L. Full details of the proof are in the full version. Completeness follows from binding property of $\mathcal{COM}$: when $x \in L$, the commitments in the Determining Message Phase are statistically binding with unique openings with overwhelming probability. Thus, the prover can extract the unique message m and make the verifier accept in the Final Message Phase. For soundness, note that when $x \notin L$, the commitments in the first phase are statistically hiding. Thus, m committed to in the Determining Message Phase is information theoretically hidden from a cheating prover (also shares received during the

[6] It follows from the description in the full version, that a perfectly non-interactive ID commitment always has a unique opening. On the other hand an honest sender statistical non-interactive ID commitment, has a unique opening with overwhelming probability, for honest senders only.

Common Input: $x \in L \cap \{0,1\}^n$, $k = \omega(\log \kappa)$ and $n = \mathsf{poly}(\kappa)$, for a security parameter κ..
Secret Input to P: Witness w such that $(x, w) \in R_L$ (not needed in case of unbounded prover).

1. **Determining Message** $(V \rightarrow P)$ V chooses message m randomly from $\{0,1\}^{\ell_0}$, and computes $\alpha = \mathsf{C}_{L,x}(m; \rho_0)$ for some random $\rho_0 \in \{0,1\}^{\ell_1}$. For $1 \leq i \leq k$ and $1 \leq j \leq k$, V randomly chooses $\sigma^0_{i,j}$ and $\sigma^1_{i,j}$ such that $\sigma^0_{i,j} \oplus \sigma^1_{i,j} = m$. For each (i, j, b), where $1 \leq i \leq k$, $1 \leq j \leq k$ and $b \in \{0,1\}$, V randomly chooses $\rho^b_{i,j} \in \{0,1\}^{\ell_1}$ computes the commitment $\alpha^b_{i,j} := \mathsf{C}_{L,x}(\sigma^b_{i,j}; \rho^b_{i,j})$. Finally, V sends all the commitments $\alpha, \alpha^0_{1,1}, \alpha^1_{1,1}, \ldots, \alpha^1_{k,k}$ to the prover.
2. **PRS Phase** $(V \Leftrightarrow P)$ For $1 \leq l \leq k$:
   (a) P sends b_l chosen randomly in $\{0,1\}^k$ to V.
   (b) Let b^i_l be the i^{th} bit of b_l. V sends the openings of $\alpha^{b^1_l}_{l,1}, \ldots, \alpha^{b^k_l}_{l,k}$.
   (c) If the opening sent by the verifier is invalid, then P sends ABORT to verifier, and aborts the protocol.
3. **Final Message** $(P \rightarrow V)$ P runs the extractor associated to the ID commitment of the Determining Message Phase. If the extractor aborts then P aborts, else P sends the output of the extractor m' to V, who accepts if $m' = m$.

Fig. 1. Concurrent Partially Honest-Verifier Statistical Zero-Knowledge Proof System: cpHSZK

preamble do not give any information), and therefore, it can convince the verifier only with negligible probability.

To argue zero knowledge, we use the rewinding strategy of [32]. Using the PRS rewinding strategy we can construct a simulator that obliviously rewinds the verifier and is guaranteed (except with negligible probability) to obtain the opening m committed to in the Determining Message Phase, before the end of the PRS Phase for every session (except with negligible probability) initiated by the cheating verifier. Once the cpHSZK simulator knows the message m committed in the Determining Message Phase, it can play it back to the verifier in the Final Phase.

Note that to prove zero knowledge, we crucially use the fact that the verifier is partially honest. First, we need that the commitment sent by the verifier are correctly formed. This is to make sure that the commitments are done in accordance with the specifications for the first message of PRS preamble. Secondly, we need that these commitments have unique openings with overwhelming probability. If, for example, verifier's commitment to m in the Determining Message Phase has two openings, then the simulation would fail. Indeed, an unbounded prover unable to decide which is the right opening, would always abort while the simulator would still extract some message from the PRS Phase and send that to the verifier in the Final Phase. The case of an efficient prover instead would result in extracting a message that could depend on the witness used, while the one obtained by the simulator would not depend on the witness, therefore potentially generating a distinguishable deviation in the transcript.

Resettable Partially Honest-Verifier SZK. We now exploit a key property of cpHSZK and transform it into a resettable statistical zero-knowledge proof system secure against partially honest verifiers. We note that the final message of cpHSZK depends only on the first message of the verifier. In particular, it depends neither on the random tape of the prover, nor on its witness. Also messages of the prover in the PRS phase are just random strings. Thus, very informally, an adversarial verifier can not obtain any advantage by resetting the prover, as after every reset, the verifier will get the same message back in the final round. This is a crucial fact that allows us to achieve resettability.

Common Input: $x \in L \cap \{0,1\}^n, k = \omega(\log \kappa), n = \mathsf{poly}(\kappa)$ for a security parameter κ.
Secret Input to P: Witness w such that $(x, w) \in R_L$ (not needed in case of unbounded prover).

1. **Determining Message** Same as in Fig. 1.
2. **PRS Phase** $(V \Leftrightarrow P)$ P chooses a random seed s, and sets $\omega = f_s(x, \alpha, \alpha^0_{1,1}, \ldots, \alpha^1_{k,k})$. Now P divides ω into k blocks of k-bits each, i.e., $\omega = \omega^1 \circ \ldots \circ \omega^k$. For $1 \leq l \leq k$,
   (a) P sends ω^l to V.
   (b) Same as Fig. 1 Step 2b.
   (c) Same as in Fig. 1 Step 2c.
3. **Final Message** Same as in Fig. 1.

Fig. 2. Resettable Statistical Partially Honest Verifier Zero-Knowledge Proof System rpHSZK

The transformed protocol, called rpHSZK (Fig. 2), is the same as cpHSZK, except for one difference: in the PRS Phase, instead of sending random challenges in Step 2(a), the prover uses pseudorandom challenges. The prover chooses a random seed s for selecting a function from a PRF family $\{f_s\}_{s\in\{0,1\}^*}$, and sets ω as the output of $f_s()$ evaluated on the message received during the Determining Message Phase. The prover uses this ω as its random tape for the PRS phase. A modification of the PRS simulation where the simulator uses both pseudorandom and random messages during the preamble, along with other known tricks [5] allows us to prove that this protocol is a resettable statistical zero-knowledge proof system with respect to partially honest verifiers.

5 Resettable Statistical ZK from Perfect Non-interactive ID Commitments

In this section we consider languages that admit perfect non-interactive ID commitments and we construct a resettable statistical ZK proof system which is secure against *all* malicious verifiers.

Let L be a language that admits a perfect non-interactive ID commitment scheme. We extend the proof system rpHSZK for L to handle arbitrary malicious verifiers. The main idea is to enforce "partially honest behavior" on the malicious verifier. We recall that the partially honest restriction on a verifier required that the verifier generates commitments honestly. More specifically, we required that these commitments have unique openings and are correctly constructed. A fully malicious verifier however can deviate and compute commitments that do not have the prescribed form. Therefore, the only concern we have is to make sure that commitments are correctly generated. We enforce this by modifying rpHSZK and adding an extra step to it. This step requires that the verifier proves to the prover that shares constructed in Step 1 (as part of the Determining Message) are correct. If this proof is accepted then the prover can conclude that the first message of the verifier is indeed honestly generated and the malicious verifier is forced into following the desired partially honest behavior. In our protocol the verifier uses an instance-dependent argument system $\langle \mathsf{PrsSWI}_x, \mathsf{VrsSWI}_x \rangle$ such that: when $x \in L$, $\langle \mathsf{PrsSWI}_x, \mathsf{VrsSWI}_x \rangle$ is a resettably sound argument of knowledge, while when $x \notin L$, $\langle \mathsf{PrsSWI}_x, \mathsf{VrsSWI}_x \rangle$ is statistically witness indistinguishable. Since the protocol is resettably sound the malicious verifier can not go ahead with incorrect commitments even when it can reset the prover. For the protocol see Fig. 3.

Sub-protocol: $\langle \mathsf{PrsSWI}_x, \mathsf{VrsSWI}_x \rangle$ is a resettably sound argument of knowledge when $x \in L$ and a statistical witness indistinguishable argument when $x \notin L$.
Common Input: $x \in L \cap \{0,1\}^n, k = \omega(\log \kappa), n = \mathsf{poly}(\kappa)$ for a security parameter κ.
Secret Input to P[a]**:** Witness w such that $(x, w) \in R_L$ (not needed in case of unbounded prover).

1. **Determining Message:** Same as in Fig. 2.
2. **Proof of Consistency:** $(V \Leftrightarrow P)$ V and P run $\langle \mathsf{PrsSWI}_x, \mathsf{VrsSWI}_x \rangle$, where V plays the role of PrsSWI_x, and P plays the role of VrsSWI_x. V proves to P knowledge of $m, \sigma^b_{i,j}, \rho_0, \rho^b_{i,j}$ for $1 \leq i, j, \leq k$, $b \in \{0,1\}$, such that:
   (a) $\alpha = \mathsf{C}_{L,x}(m, \rho_0)$, and,
   (b) $\alpha^b_{i,j} = \mathsf{C}_{L,x}(\sigma^b_{i,j}; \rho^b_{i,j})$ for each $1 \leq i, j \leq k$ and $b \in \{0,1\}$, and,
   (c) $\sigma^0_{i,j} \oplus \sigma^1_{i,j} = m$ for $1 \leq i, j \leq k$.
3. **PRS Phase:** Same as in Fig. 2.
4. **Final Message:** Same as in Fig. 2.

[a] P aborts the protocol in case any proof from the verifiers does not `accept` or some message is not well formed.
Notice that P uses two different seeds for the PRF f (one in Step 2 and the other one in Step 3).

Fig. 3. Resettable Statistical Zero-Knowledge from Perfect Non-Interactive ID Commitments: rSZK

Application to co-$\mathcal{RSR}$ and hash proof systems. Languages in co-$\mathcal{RSR}$ and $\mathcal{DDH}$ have perfect non-interactive ID commitment schemes. Thus, from the discussion above, it follows that these languages have resettable statistical zero-knowledge proofs. For languages in $\mathcal{SZK}$ that admit hash proof systems, a minor modification of our resettable statistical zero-knowledge protocol suffices. The details are provided in the full version [17].

6 Resettable Statistical ZK for All Languages in $\mathcal{SZK}$

In this section we construct the general proof system which is actually resettable statistical zero knowledge for all languages that have a statistical zero knowledge proof. Just like in previous section, we start with a resettable partially honest verifier statistical ZK proof system. But we look at all languages in $\mathcal{SZK}$ and construct a resettable statistical ZK proof system which is secure against *all* malicious verifiers.

Let L be a language that admits an honest sender statistical non-interactive ID commitment scheme $\mathcal{COM}$. We extend the proof system rpHSZK for L to handle arbitrary malicious verifiers. The main idea is to enforce "partially honest behavior" on the malicious verifier. We recall that the partially honest restriction on a verifier required that the verifier uses $\mathcal{COM}$ to generate commitments honestly. More specifically, we required that these commitments are correctly constructed and have unique openings. The first requirement can be handled in a way just like in previous section, i.e. by having the verifier prove to the prover that shares constructed in Step 1 (as part of the Determining Message) are correct. We use the ID argument system $\langle \mathsf{PrsSWI}_x, \mathsf{VrsSWI}_x \rangle$ to achieve this. The problem of uniqueness is more tricky, and we discuss that next.

The difficulty lies in the fact that the statistical non-interactive ID commitment scheme for all languages in $\mathcal{SZK}$ [7], only works with respect to honest senders. Indeed, if the sender chooses the randomness for the commitment uniformly, then, with overwhelming probability, the computed commitment has a unique valid opening. However a malicious sender could focus on a set of negligible size, $\mathcal{B}$, of *bad* random strings r, such that $\mathsf{C}_{L,x}(m;r)$ does not have a unique opening. If a malicious verifier (that plays as sender of this commitment scheme) is able to pick random strings from $\mathcal{B}$, then the real interaction and the simulation can be easily distinguished. In the real protocol, the prover tries to invert the commitment α, finds it does not have a unique opening, and aborts. In the simulation, the simulator extracts some message m from the PRS phase, and sends m as the final message. As the simulator is polynomially bounded, it *can not* detect if the commitment has a unique opening or not. To use this commitment scheme, we must somehow ensure that the verifier does not use bad randomness for its commitments. We do this by adding a special coin-flipping subprotocol at the beginning of the protocol. However, because of reset attacks, the coin-flipping subprotocol introduces several technical problems.

We begin by describing our coin-flipping protocol. The coin-flipping protocol requires a commitment scheme such that computational binding holds against

resetting senders when $x \in L$ and statistical hiding holds when $x \notin L$. We use the interactive ID commitment scheme $\mathcal{COM}_{L,x} = (S_x, R_x)$. The coin flipping proceeds as follows: first the verifier commits to a random string r_1. Let the transcript of the interactive commitment be c. Then prover applies the sub-exponentially hard PRF $f_s(c)$ and obtains r_2 that is sent to the verifier. The randomness that the verifier will use for the non-interactive ID commitment is $r_1 \oplus r_2$. For technical reasons, the verifier also needs to prove knowledge of r_1 after it has committed to r_1. We use the interactive ID argument system $\langle \mathsf{PrsSWI}_x, \mathsf{VrsSWI}_x \rangle$ for this.

Next we highlight the reasons behind the use of sub-exponentially hard pseudorandom (PRF) functions for our construction. Let α be the statistical non-interactive ID commitment of some message m sent by the verifier. There are two ways in which α might not have a unique opening. In the first case, a malicious V^*, after looking at prover's response r_2, might use an opening of c such that $r_1 \oplus r_2 \in \mathcal{B}$. This however would violate the computational binding of the interactive ID commitment scheme secure against resetting senders used in the coin flipping, thus this event occurs with negligible probability. The second case is more subtle. It might be possible that performing reset attacks, the verifier can study the behavior of the PRF, and then can be able to succeed in obtaining that $r_1 \oplus r_2 \in \mathcal{B}$ with non-negligible probability (even though the polynomial-time V^* does not know the two openings). In this case, we can not construct a polynomial-time adversary that breaks f_s, as we can not efficiently decide if $r \in \mathcal{B}$. This is where we need the sub-exponential hardness of the one-way function and in turn of the PRF. As $|\mathcal{B}|$ is only 2^{ℓ} while the size of the set of all random strings is 2^L, where $l = \mathrm{o}(L)$, we can give the entire set $\mathcal{B}$ as input to the sub-exponential size circuit that aims at breaking the PRF. The circuit can now check if the string r is a bad string or not, by searching through its input. Notice that one can give as input to the circuit the whole $\mathcal{B}$ for each of the polynomial number of statements (since for each x there can be a different $\mathcal{B}$) on which the reset attack is applied. This sub-exponential size circuit has still size $\mathrm{o}(L)$ and breaks the PRF which contradicts the sub-exponential hardness of the PRF.

Completeness follows from the fact that when $x \in L$, with overwhelming probability, the commitment α in the determining message will have a unique opening. Thus, the prover will be able to extract the committed message and send it as the final message to the verifier, that will accept.

Statistical resettable zero knowledge property of our protocol also follows the same argument. Indeed, when $x \in L$ even a resetting verifier can not cheat during the proofs in Steps 1(c) and 3. Moreover, the above discussion about the security of the coin-flipping protocol implies that a resetting adversarial verifier is forced into following partially honest behavior when computing the non-interactive ID commitments.

Finally, we look at soundness. Note that when $x \notin L$ non-interactive ID commitments are statistically hiding and the protocol $\langle \mathsf{PrsSWI}_x, \mathsf{VrsSWI}_x \rangle$ is statistical WI in Steps 1(c) and 3. Also note that only a single share is revealed in the PRS phase. From this it follows that the prover's view when verifier commits

message m is statistically close to its view when verifier commits to m', where $m \neq m'$. Thus, the probability that it replies with the correct final message is negligible. The complete protocol and proof appear in the full version [17].

7 2-Round Statistical Witness Indistinguishability

In this section, we highlight the applicability of our techniques, and construct a simple two-round resettable statistical witness-indistinguishable argument for languages that have efficiently extractable perfectly binding instance-dependent commitment schemes. As discussed before, this class contains, in particular, all languages that admit hash proof systems. We note that all results in this section hold in the stronger model of statistical zero-knowledge where the verifier is computationally unbounded.

Informally, the two-round WI argument consists of the verifier committing to a randomly chosen message m using the instance-dependent commitment scheme for that language. The prover, using the witness and the efficient extractor, extracts a message m' from the commitment and sends it to the verifier. The verifier accepts if $m = m'$. Intuitively, as long as verifier's commitment is well-formed, this protocol is a perfect WI, as irrespective of the witness and randomness, the prover always extracts the same message (in fact, prover's strategy is deterministic). Thus, the only complication is to ensure that verifier's commitment is well-formed in a round efficient manner. We enforce this by making the verifier provide a non-interactive WI proof (i.e., a one-round ZAP [21,15]) of "well-formedness" in the first round.

For lack of space, details of the protocol and proof of security can be found in the full version of this paper.

Acknowledgments. We thank Giuseppe Persiano for suggesting the open problem of achieving resettable statistical zero knowledge and Omkant Pandey for several discussions on our techniques.

Research supported in part by NSF grants 0830803, 09165174, 1065276, 1118126 and 1136174, US-Israel BSF grant 2008411, B. John Garrick Foundation, OKAWA Foundation Award, IBM Award, Lockheed-Martin Corporation and the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0392. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. The work of the third author was partially done while visiting UCLA and has also been supported in part by the European Commission through the FP7 programme under contract 216676 ECRYPT II, and in part by a research grant of Provincia di Salerno.

References

1. Barak, B., Goldreich, O., Goldwasser, S., Lindell, Y.: Resettably-sound zero-knowledge and its applications. In: FOCS, pp. 116–125 (2001), full version, `http://eprint.iacr.org/2001/063`

2. Barak, B., Lindell, Y., Vadhan, S.: Lower bounds for non-black-box zero knowledge. In: FOCS 2003, pp. 384–393 (2003)
3. Bellare, M., Micali, S., Ostrovsky, R.: The (true) complexity of statistical zero knowledge. In: STOC, pp. 494–502 (1990)
4. Brassard, G., Chaum, D., Crépeau, C.: Minimum disclosure proofs of knowledge. J. Comput. Syst. Sci. 37(2), 156–189 (1988)
5. Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge (extended abstract). In: STOC, pp. 235–244 (2000)
6. Canetti, R., Kilian, J., Petrank, E., Rosen, A.: Black-box concurrent zero-knowledge requires (almost) logarithmically many rounds. SIAM J. Comput. 32(1), 1–47 (2002)
7. Chailloux, A., Ciocan, D.F., Kerenidis, I., Vadhan, S.P.: Interactive and Noninteractive Zero Knowledge are Equivalent in the Help Model. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 501–534. Springer, Heidelberg (2008)
8. Cho, C., Ostrovsky, R., Scafuro, A., Visconti, I.: Simultaneously Resettable Arguments of Knowledge. In: Cramer, R. (ed.) TCC 2012. LNCS, pp. 530–547. Springer, Heidelberg (2012)
9. Cook, J., Etesami, O., Miller, R., Trevisan, L.: Goldreich's One-Way Function Candidate and Myopic Backtracking Algorithms. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 521–538. Springer, Heidelberg (2009)
10. Cramer, R., Shoup, V.: Universal Hash Proofs and a Paradigm for Adaptive Chosen Ciphertext Secure Public-Key Encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 45–64. Springer, Heidelberg (2002)
11. Damgård, I., Fazio, N., Nicolosi, A.: Non-interactive Zero-Knowledge from Homomorphic Encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 41–59. Springer, Heidelberg (2006)
12. Deng, Y., Goyal, V., Sahai, A.: Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. In: FOCS (2009)
13. Di Crescenzo, G., Persiano, G., Visconti, I.: Constant-Round Resettable Zero Knowledge with Concurrent Soundness in the Bare Public-Key Model. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 237–253. Springer, Heidelberg (2004)
14. Dolev, D., Dwork, C., Naor, M.: Non-Malleable Cryptography. SIAM J. on Computing 30(2), 391–437 (2000)
15. Dwork, C., Naor, M.: Zaps and their applications. In: FOCS, pp. 283–293 (2000)
16. Dwork, C., Naor, M., Sahai, A.: Concurrent zero-knowledge. In: STOC, pp. 409–418 (1998)
17. Garg, S., Ostrovsky, R., Visconti, I., Wadia, A.: Resettable statistical zero knowledge. Cryptology ePrint Archive, Report 2011/457 (2011), http://eprint.iacr.org/
18. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game - a completeness theorem for protocols with honest majority. In: STOC, pp. 218–229 (1987)
19. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof-systems. SIAM J. on Computing 18(6), 186–208 (1989)
20. Goyal, V., Moriarty, R., Ostrovsky, R., Sahai, A.: Concurrent Statistical Zero-Knowledge Arguments for NP from One Way Functions. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 444–459. Springer, Heidelberg (2007)
21. Groth, J., Ostrovsky, R., Sahai, A.: Non-interactive Zaps and New Techniques for NIZK. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 97–111. Springer, Heidelberg (2006)
22. Itoh, T., Ohta, Y., Shizuya, H.: A language-dependent cryptographic primitive. J. Cryptology 10(1), 37–50 (1997)

23. Lindell, Y.: Bounded-concurrent secure two-party computation without setup assumptions. In: STOC, pp. 683–692. ACM (2003)
24. Micali, S., Reyzin, L.: Soundness in the Public-Key Model. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 542–565. Springer, Heidelberg (2001)
25. Micciancio, D., Ong, S.J., Sahai, A., Vadhan, S.P.: Concurrent Zero Knowledge Without Complexity Assumptions. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 1–20. Springer, Heidelberg (2006)
26. Micciancio, D., Yilek, S.: The Round-Complexity of Black-Box Zero-Knowledge: A Combinatorial Characterization. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 535–552. Springer, Heidelberg (2008)
27. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC 1990, pp. 427–437 (1990)
28. Ong, S.J., Vadhan, S.P.: An Equivalence Between Zero Knowledge and Commitments. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 482–500. Springer, Heidelberg (2008)
29. Ostrovsky, R.: One-way functions, hard on average problems, and statistical zero-knowledge proofs. In: Structure in Complexity Theory Conference, pp. 133–138 (1991)
30. Pass, R., Tseng, W.L.D., Venkitasubramaniam, M.: Concurrent zero knowledge: Simplifications and generalizations. Technical Report (2008), `http://hdl.handle.net/1813/10772`
31. Pass, R., Wee, H.: Constant-Round Non-malleable Commitments from Sub-exponential One-Way Functions. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 638–655. Springer, Heidelberg (2010)
32. Prabhakaran, M., Rosen, A., Sahai, A.: Concurrent zero knowledge with logarithmic round-complexity. In: FOCS, pp. 366–375 (2002)
33. Sahai, A., Vadhan, S.P.: A complete problem for statistical zero knowledge. J. ACM 50(2), 196–249 (2003)
34. Santis, A.D., Crescenzo, G.D., Persiano, G., Yung, M.: On monotone formula closure of szk. In: FOCS, pp. 454–465 (1994)
35. Scafuro, A., Visconti, I.: On round-optimal zero knowledge in the bare public-key model. In: EUROCRYPT. LNCS. Springer, Heidelberg (2012)
36. Tompa, M., Woll, H.: Random self-reducibility and zero knowledge interactive proofs of possession of information. In: FOCS, pp. 472–482 (1987)
37. Vadhan, S.: A Study of Statistical Zero-Knowledge Proofs. Ph.D. thesis. MIT (1999)
38. Wee, H.: Black-box, round-efficient secure computation via non-malleability amplification. In: FOCS (2010)
39. Yung, M., Zhao, Y.: Generic and Practical Resettable Zero-Knowledge in the Bare Public-Key Model. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 129–147. Springer, Heidelberg (2007)