

# Concurrent Zero-Knowledge With Concurrent Soundness in the Bare Public-Key Model \*

Yunlei Zhao<sup>†</sup>

## Abstract

In this paper, we present a constant-round (specifically, 6-round) black-box concurrent zero-knowledge argument of knowledge with *concurrent soundness* for  $\mathcal{NP}$  in the bare public-key (BPK) model under standard intractability assumptions. To our knowledge, our result is the first work that achieves simultaneous concurrent security for zero-knowledge protocols in the BPK model. In comparison with the rZK protocols of [10, 41], the rZK protocols of [10, 41] guarantee concurrent (and more strongly, resettable) zero-knowledge (for concurrent prover security) but do not guarantee concurrent soundness (for concurrent verifier security). Furthermore, the works of [10, 41] are based on sub-exponential hardness assumptions. In comparison, our work provides both concurrent prover security and concurrent verifier security for zero-knowledge protocols in the BPK model under standard intractability assumptions without assuming any sub-exponential or super-polynomial hardness assumption.

Since the BPK model is very simple and also very reasonable and is in fact a weak version of the frequently used public-key infrastructure (PKI) model, which underlies any public-key cryptosystem or digital signature scheme, we suggest that zero-knowledge protocols with simultaneous concurrent security in the BPK model may be of independent interests and can be used as a building block in other applications in the BPK model (e. g. secure two-party and multi-party computation with registered public-keys).

**Keywords:** concurrent soundness, concurrent zero-knowledge, bare public-key (BPK) model

## 1 Introduction

The notion of zero-knowledge (ZK) was introduced in the seminal paper of Goldwasser, Micali and Rackoff [35] to illustrate situations where a prover reveals nothing other than the verity of a given statement to an even malicious verifier. Since their introduction, zero-knowledge proofs have proven to be very useful as a building block in the construction of cryptographic protocols, especially after Goldreich, Micali and Wigderson [34] have shown that all languages in  $\mathcal{NP}$  admit zero-knowledge proofs. By now, zero-knowledge has played a central role in the field of cryptography and is the accepted methodology to define and prove security of various cryptographic tasks.

With the emergence and far and wide sweeping popularity of the Internet, much recent research attention, initiated by Dwork, Naor and Sahai [26], has been paid on the security threats of zero-knowledge protocols when executing concurrently in an asynchronous network setting like the Internet. In this scenario, many concurrent executions of the same protocol take place in an asynchronous network setting. Honest players are assumed oblivious of each other's existence, nor do they generally know the

---

\*This research is supported by a research grant of City University of Hong Kong (Grant No.7001232). The first author is also supported by Microsoft Research Asia Fellowship 2003.

<sup>†</sup>Department of Computer Science, Fudan University, Shanghai, P. R. China.  
csylzhao@cityu.edu.hk

topology of the network, and thus cannot coordinate their executions. However, a malicious adversary controlling all the verifiers can schedule all the executions concurrently at its wish.

Although non-constant-round black-box concurrent zero-knowledge proofs for  $\mathcal{NP}$  exist in the plain model under standard intractability assumptions [45, 38, 44], but they cannot be constant-round in the black-box sense [11]. Constant-round non-black-box bounded concurrent zero-knowledge arguments and arguments of knowledge for  $\mathcal{NP}$  are achieved in [1, 4]. To achieve constant-round black-box concurrent zero-knowledge, several computational models are introduced: the timing model [26], the preprocessing model [17] and the bare public-key model [10].

The bare public-key (BPK) model is first introduced by Canetti, Goldreich, Goldwasser and Micali [10] to achieve round-efficient resettable zero-knowledge (rZK) that is a generalization and strengthening of the notion of concurrent zero-knowledge. But as stated by Micali and Reyzin [41], although introduced with a specific application in mind, the BPK model applies to interactive systems in general, regardless of their knowledge complexity. A protocol in BPK model simply assumes that all verifiers have deposited a public key in a public file before any interaction takes place among the users. This public file is accessible to all users at all times. Note that an adversary may deposit many (possibly invalid) public keys in it, particularly, without even knowing corresponding secret keys or whether such exist. That is, no trusted third party is assumed in the BPK model. The BPK model is very simple, and it is in fact a weak version of the frequently used public-key infrastructure (PKI) model, which underlies any public-key cryptosystem or digital signature scheme. Despite its apparent simplicity, the BPK model is quite powerful. While rZK protocols exist both in the standard and in the BPK models [10], only in the latter case can they be constant-round, at least in the black box sense.

Various soundness notions of cryptographic protocols in public-key models are noted and clarified by Micali and Reyzin [41]. In public-key models, a verifier  $V$  has a secret key SK, corresponding to its public-key PK. A malicious prover  $P^*$  could potentially gain some knowledge about SK from an interaction with the verifier. This gained knowledge might help him to convince the verifier of a false theorem in another interaction. In this paper we focus on concurrent soundness which roughly means, for zero-knowledge protocols, that a malicious prover  $P^*$  can not convince the honest verifier  $V$  of a false statement even  $P^*$  is allowed multiple interleaved interactions with  $V$ . Resettable zero-knowledge protocols with concurrent soundness are really desirable in most smart-card based applications over Internet. Unfortunately, up to now we do not know how to construct resettable zero-knowledge protocols with concurrent soundness for  $\mathcal{NP}$  in the BPK model and it is suggested as an interesting and important open problem in [10, 41]. Resettable zero-knowledge with concurrent soundness in some stronger version of the BPK model can be found in [42, 49].

## 1.1 Our contributions

In this paper, we present a constant-round (specifically, 6-round) black-box concurrent zero-knowledge argument of knowledge with *concurrent soundness* for  $\mathcal{NP}$  in the bare public-key (BPK) model under standard intractability assumptions. To our knowledge, our result is the first work that achieves simultaneous concurrent security for zero-knowledge protocols in the BPK model. Note that the rZK protocols of [10, 41] guarantee concurrent (and more strongly, resettable) zero-knowledge (for concurrent prover security) but do not guarantee concurrent soundness (for concurrent verifier security). Furthermore, the works of [10, 41] are under sub-exponential hardness assumptions. In comparison, our work provides both concurrent prover security and concurrent verifier security for zero-knowledge protocols in the BPK model under standard intractability assumptions without assuming any sub-exponential or super-polynomial hardness assumption. Our protocol is not resettable zero-knowledge because any argument of knowledge protocol cannot be resettable zero-knowledge. Since the BPK model is very simple and also very reasonable and is in fact a weak version of the frequently used public-key infras-

structure (PKI) model, which underlies any public-key cryptosystem or digital signature scheme, we suggest that zero-knowledge protocols with simultaneous concurrent security in the BPK model may be of independent interests and can be used as a building block in other applications in the BPK model (e. g. secure two-party and multi-party computation with registered public-keys).

## 1.2 Organization

In Section 2, we quickly recall the major cryptographic tools used. In Section 3, we present the definitions of concurrent zero-knowledge and concurrent soundness in the BPK model. In Section 4, we present the constant-round black-box concurrent zero-knowledge argument of knowledge with concurrent soundness for  $\mathcal{NP}$  in the BPK model and prove its security under standard intractability assumptions.

## 2 Preliminaries

In this section, we quickly recall the major cryptographic tools used.

We use standard notations and conventions below for writing probabilistic algorithms and experiments. If  $A$  is a probabilistic algorithm, then  $A(x_1, x_2, \dots; r)$  is the result of running  $A$  on inputs  $x_1, x_2, \dots$  and coins  $r$ . We let  $y \leftarrow A(x_1, x_2, \dots)$  denote the experiment of picking  $r$  at random and letting  $y$  be  $A(x_1, x_2, \dots; r)$ . If  $S$  is a finite set then  $x \leftarrow S$  is the operation of picking an element uniformly from  $S$ . If  $\alpha$  is neither an algorithm nor a set then  $x \leftarrow \alpha$  is a simple assignment statement.

### 2.1 $\Sigma$ -protocols for proving the knowledge of commitment trapdoors

$\Sigma$ -protocols are first introduced by Cramer, Damgard and Schoenmakers [14]. Informally, a  $\Sigma$ -protocol is itself a 3-round public-coin special honest verifier zero-knowledge protocol with special soundness in the knowledge-extraction sense. Since its introduction,  $\Sigma$ -protocols have been proved a very powerful cryptographic tool and are widely used in numerous important cryptographic applications including digital signatures (by using the famous Fiat-Shamir methodology [29] and efficient electronic payment systems [13]). For a good survey of  $\Sigma$ -protocols and their applications, readers are referred to [18, 13].

**Definition 2.1** ( $\Sigma$ -protocol [14]) *A 3-round public-coin protocol  $\langle P, V \rangle$  is said to be a  $\Sigma$ -protocol for relation  $R$  if the following hold:*

- *Completeness.* If  $P, V$  follow the protocol, the verifier always accepts.
- *Special soundness.* From any common input  $x$  and any pair of accepting conversations on input  $x$ ,  $(a, e, z)$  and  $(a, e', z')$  where  $e \neq e'$ , one can efficiently compute  $w$  such that  $(x, w) \in R$ . Here  $a, e, z$  stand for the first, the second and the third message respectively.
- *Special honest verifier zero-knowledge (SHVZK).* There exists a polynomial-time simulator  $S$ , which on input  $x$  and a random challenge string  $e$ , outputs an accepting conversation of the form  $(a, e, z)$ , with the same probability distribution as conversations between the honest  $P, V$  on input  $x$ .

**Definition 2.2** (trapdoor commitment scheme TC) *A trapdoor commitment scheme (TC) is a quintuple of probabilistic polynomial-time (PPT) algorithms  $TCGen, TCCom, TCVer, TCKeyVer$  and  $TCFake$ , such that*

- *Completeness.*  $\forall n, \forall v, \Pr[(TCPK, TCSK) \stackrel{R}{\leftarrow} TCGen(1^n); (c, d) \stackrel{R}{\leftarrow} TCCom(TCPK, v) : TCKeyVer(TCPK, 1^n) = TCVer(TCPK, c, v, d) = \text{YES}] = 1.$

- *Computational Binding.* For all sufficiently large  $n$  and for all PPT adversaries  $A$ , the following probability is negligible in  $n$ :  $\Pr[(TCPK, TCSK) \stackrel{R}{\leftarrow} \text{TGen}(1^n); (c, v_1, v_2, d_1, d_2) \stackrel{R}{\leftarrow} A(1^n, TCPK) : \text{TCVer}(TCPK, c, v_1, d_1) = \text{TCVer}(TCPK, c, v_2, d_2) = \text{YES} \text{ and } v_1 \neq v_2]$ .
- *Perfect Hiding.*  $\forall TCPK$  such that  $\text{TKeyVer}(TCPK, 1^n) = \text{YES}$  and  $\forall v_1, v_2$  of equal length, the following two probability distributions are identical:  $[(c_1, d_1) \stackrel{R}{\leftarrow} \text{TCom}(TCPK, v_1) : c_1]$  and  $[(c_2, d_2) \stackrel{R}{\leftarrow} \text{TCom}(TCPK, v_2) : c_2]$ .
- *trapdooriness.*  $\forall (TCPK, TCSK) \in \{\text{TGen}(1^n)\}$ ,  $\forall v_1, v_2$  of equal length, the following two probability distributions are identical:  
 $[(c, d_1) \stackrel{R}{\leftarrow} \text{TCom}(TCPK, v_1); d'_2 \stackrel{R}{\leftarrow} \text{TFake}(TCPK, TCSK, c, v_1, d_1, v_2) : (c, d'_2)]$  and  
 $[(c, d_2) \stackrel{R}{\leftarrow} \text{TCom}(TCPK, v_2) : (c, d_2)]$ .

The following is a construction of trapdoor commitment scheme based on DLP intractability assumption [8]: On a security parameter  $n$ , the receiver selects uniformly an  $n$ -bit prime  $p$  so that  $q = (p-1)/2$  is a prime, an element  $g$  of order  $q$  in  $\mathbf{Z}_p^*$ . Then the receiver uniformly selects  $w$  in  $\mathbf{Z}_q^*$  and sets  $h = g^w \text{ mod } p$ . The receiver publishes  $(p, q, g, h)$  as its public-key and keeps  $w$  as its secret-key (i. e. the trapdoor). To commit a bit  $\sigma$ , the sender first checks that  $(p, q, g, h)$  is of the right form (otherwise it halts announcing that the receiver is cheating), uniformly selects  $s \in \mathbf{Z}_q$ , and sends  $g^s h^\sigma \text{ mod } p$  as its commitment.

And the following is a  $\Sigma$ -protocol  $\langle P, V \rangle$  suggested by Schnorr [47] for proving the knowledge of trapdoor secret-key,  $w$ , for a public-key of the above form  $(p, q, g, h)$  such that  $h = g^w \text{ mod } p$ :

- $P$  chooses  $r$  at random in  $\mathbf{Z}_q$  and sends  $a = g^r \text{ mod } p$  to  $V$ .
- $V$  chooses a challenge  $e$  at random in  $\mathbf{Z}_{2^t}$  and sends it to  $P$ . Here,  $t$  is fixed such that  $2^t < q$ .
- $P$  sends  $z = r + ew \text{ mod } p$  to  $V$ , who checks that  $g^z = ah^e \text{ mod } p$ , that  $p, q$  are prime and that  $g, h$  have order  $q$ , and accepts iff this is the case.

**The OR-proof of  $\Sigma$ -protocols.** As shown in [14], any public-coin SHVZK protocol is itself witness indistinguishable (WI). Although for languages that each instance has a single witness,  $\Sigma$ -protocols for that languages is trivially WI, one basic construction with  $\Sigma$ -protocols allows a prover to show that given two inputs  $x_0, x_1$  that each of them has a single witness, he knows  $w$  such that either  $(x_0, w) \in R$  or  $(x_1, w) \in R$ , BUT without revealing which is the case.

So we assume we are given a  $\Sigma$ -protocol  $\langle P, V \rangle$  for  $R$  with random challenges of length  $t$ . Assume also that  $(x_0, x_1)$  are common input to  $P, V$ , and that  $w$  is private input to  $P$ , where  $(x_b, w) \in R$  for  $b = 0$  or  $1$ . Roughly speaking, the idea is that we will ask the prover to complete two instances of  $\langle P, V \rangle$ , with respect to  $x_0, x_1$  respectively. For  $x_b$ , he can do this for real, for  $x_{1-b}$  he will have to fake it using the SHVZK simulator. However, if we give him a little freedom in choosing the challenges to answer, he will be able to complete both instances. More precisely, consider the following protocol, which we call  $\Sigma_{OR}$ :

- $P$  computes the first message  $a_b$  in  $\langle P, V \rangle$ , using  $x_b, w$  as private inputs.  $P$  chooses  $e_{1-b}$  at random and runs the SHVZK simulator  $S$  on input  $x_{1-b}, e_{1-b}$ , let  $(a_{1-b}, e_{1-b}, z_{1-b})$  be the output.  $P$  finally sends  $a_0, a_1$  to  $V$ .
- $V$  chooses a random  $t$ -bit string  $s$  and sends it to  $P$ .

- $P$  sets  $e_b = s \oplus e_{1-b}$  and computes the answer  $z_b$  to challenge  $e_b$  using  $(x_b, a_b, e_b, w)$  as input. He sends  $(e_0, z_0, e_1, z_1)$  to  $V$ .
- $V$  checks that  $s = e_0 \oplus e_1$  and that conversations  $(a_0, e_0, z_0), (a_1, e_1, z_1)$  are accepting conversations with respect to inputs  $x_0, x_1$ , respectively.

**Theorem 2.1** [18] *The protocol  $\Sigma_{OR}$  above is a  $\Sigma$ -protocol for  $R_{OR}$ , where  $R_{OR} = \{(x_0, x_1), w \mid (x_0, w) \in R \text{ or } (x_1, w) \in R\}$ . Moreover, for any verifier  $V^*$ , the probability distribution of conversations between  $P$  and  $V^*$ , where  $w$  is such that  $(x_b, w) \in R$ , is independent of  $b$ . That is,  $\Sigma_{OR}$  is perfectly witness indistinguishable.*

## 2.2 Other cryptographic tools

We proceed to present other cryptographic tools used in this paper.

**Definition 2.3 (system for proof of knowledge)** *Let  $R$  be a binary relation and  $\kappa : N \rightarrow [0, 1]$ . We say that a probabilistic polynomial-time (PPT) interactive machine  $V$  is a **knowledge verifier for the relation  $R$**  with **knowledge error  $\kappa$**  if the following two conditions hold:*

- *Non-triviality: There exists an interactive machine  $P$  such that for every  $(x, y) \in R$  all possible interactions of  $V$  with  $P$  on common input  $x$  and auxiliary input  $y$  are accepting.*
- *Validity (with error  $\kappa$ ): There exists a polynomial  $q(\cdot)$  and a probabilistic oracle machine  $K$  such that for every interactive machine  $P^*$ , every  $x \in L_R$ , and every  $y, r \in \{0, 1\}^*$ , machine  $K$  satisfies the following condition:*

*Denote by  $p(x, y, r)$  the probability that the interactive machine  $V$  accepts, on input  $x$ , when interacting with the prover specified by  $P_{x,y,r}^*$  (where  $P_{x,y,r}^*$  denotes the strategy of  $P^*$  on common input  $x$ , auxiliary input  $y$  and random-tape  $r$ ). If  $p(x, y, r) > \kappa(|x|)$ , then, on input  $x$  and with oracle access to  $P_{x,y,r}^*$ , machine  $K$  outputs a solution  $s \in R(x)$  within an accepted number of steps bounded by*

$$\frac{q(|x|)}{p(x, y, r) - \kappa(|x|)}$$

*The oracle machine  $K$  is called a **knowledge extractor**.*

*An interactive proof system  $(P, V)$  such that  $V$  is a knowledge verifier for a relation  $R$  and  $P$  is a machine satisfying the non-triviality condition (with respect to  $V$  and  $R$ ) is called a **system for proof of knowledge for the relation  $R$** . The proof system  $(P, V)$  is a **system of zero-knowledge proof of knowledge (ZKPOK)** if it is also zero-knowledge.*

For more clarifications on the definition of proof of knowledge, readers are referred to [31]. More recent advances of zero-knowledge arguments of knowledge can be found in [39, 4].

**Definition 2.4 (witness indistinguishability WI)** *Let  $\langle P, V \rangle$  be an interactive proof system for a language  $L \in \mathcal{NP}$ , and let  $R_L$  be the fixed  $\mathcal{NP}$  witness relation for  $L$ . That is  $x \in L$  if there exists a  $w$  such that  $(x, w) \in R_L$ . We denote by  $\text{view}_{V^*(z)}^{P(w)}(x)$  a random variable describing the transcript of all messages exchanged between  $V^*$  and  $P$  in an execution of the protocol on common input  $x$ , when  $P$  has auxiliary input  $w$  and  $V^*$  has auxiliary input  $z$ . We say that  $\langle P, V \rangle$  is **witness indistinguishability for  $R_L$**  if for every PPT interactive machine  $V^*$ , and every two sequences  $W^1 = \{w_x^1\}_{x \in L}$  and  $W^2 = \{w_x^2\}_{x \in L}$ , so that  $(x, w_x^1) \in R_L$  and  $(x, w_x^2) \in R_L$ , the following two probability distributions are computationally indistinguishable:  $\{x, \text{view}_{V^*(z)}^{P(w_x^1)}\}_{x \in L, z \in \{0, 1\}^*}$  and  $\{x, \text{view}_{V^*(z)}^{P(w_x^2)}\}_{x \in L, z \in \{0, 1\}^*}$ .*

In this paper we use 3-round public-coin witness indistinguishability proofs of knowledge (WIPOK) for  $\mathcal{NP}$ . Such protocols exist under the existence of one-way functions, e. g. the parallel repetitions of Blum’s 3-round proof of knowledge for HC [7]. We remark that 2-round public-coin WI proofs for  $\mathcal{NP}$  do exist under the existence of one-way permutations [25].

**Definition 2.5 (non-interactive zero-knowledge NIZK)** *Let  $NIP$  and  $NIV$  be two interactive machines and  $NIV$  is also probabilistic polynomial-time, and let  $NI\sigma Len$  be a positive polynomial. We say that  $\langle NIP, NIV \rangle$  is an NIZK proof system for an  $\mathcal{NP}$  language  $L$ , if the following conditions hold:*

- *Completeness. For any  $x \in L$  of length  $n$ , any  $\sigma$  of length  $NI\sigma Len(n)$ , and  $\mathcal{NP}$ -witness  $w$  for  $x$ , it holds that*

$$\Pr[\Pi \xleftarrow{R} NIP(\sigma, x, w) : NIV(\sigma, x, \Pi) = YES] = 1.$$

- *Soundness.  $\forall x \notin L$  of length  $n$ ,*

$$\Pr[\sigma \xleftarrow{R} \{0, 1\}^{NI\sigma Len(n)} : \exists \Pi \text{ s.t. } NIV(\sigma, x, \Pi) = YES] \text{ is negligible in } n.$$

- *Zero-Knowledgeness.  $\exists$  a PPT simulator  $NIS$  such that,  $\forall$  sufficiently large  $n$ ,  $\forall x \in L$  of length  $n$  and  $\mathcal{NP}$ -witness  $w$  for  $x$ , the following two distributions are computationally indistinguishable:*

$$[(\sigma', \Pi') \xleftarrow{R} NIS(x) : (\sigma', \Pi')] \text{ and } [\sigma \xleftarrow{R} \{0, 1\}^{NI\sigma Len(n)}; \Pi \xleftarrow{R} NIP(\sigma, x, w) : (\sigma, \Pi)].$$

Non-interactive zero-knowledge proof systems for  $\mathcal{NP}$  can be constructed based on any one-way permutation [27]. An efficient implementation based on any one-way permutation is presented in [37] and readers are referred to [22] for recent advances of NIZK.

**Definition 2.6 (NIZK proof of knowledge [23])** *An NIZK proof system  $\langle NIP, NIV \rangle$  for a language  $L \in \mathcal{NP}$  with witness relation  $R_L$  (as defined above) is NIZK proof of knowledge (NIZKPOK) if there exists a pair of PPT machines  $(E_1, E_2)$  and a negligible function  $\varepsilon$  such that for all sufficiently large  $n$ :*

- *Reference-String Uniformity. The distribution on reference strings produced by  $E_1(1^n)$  has statistical distance at most  $\varepsilon(n)$  from the uniform distribution on  $\{0, 1\}^{NI\sigma Len(n)}$ .*
- *Witness Extractability. For all adversaries  $A$ , we have that  $\Pr[\mathbf{Expt}_A^E(n) = 1] \geq \Pr[\mathbf{Expt}_A(n) = 1] - \varepsilon(n)$ , where the experiments  $\mathbf{Expt}_A(n)$  and  $\mathbf{Expt}_A^E(n)$  are defined as follows:*

<b>Expt<sub>A</sub>(n):</b> $\sigma \xleftarrow{R} \{0, 1\}^{NI\sigma Len(n)}$ $(x, \Pi) \leftarrow A(\sigma)$ return $NIV(x, \sigma, \Pi)$	<b>Expt<sub>A</sub><sup>E</sup>(n):</b> $(\sigma, \tau) \leftarrow E_1(1^n)$ $(x, \Pi) \leftarrow A(\sigma)$ $w \leftarrow E_2(\sigma, \tau, x, \Pi)$ return 1 if $(x, w) \in R_L$
--	--

NIZK proofs of knowledge for  $\mathcal{NP}$  can be constructed assuming the existence of one-way permutations and dense secure public-key cryptosystems [23].

### 3 Definitions of concurrent zero-knowledge and concurrent soundness in the BPK model

In this section, we present the formal definitions of concurrent zero-knowledge and concurrent soundness in the BPK model.

**Concurrent zero-knowledge in the BPK model.** Let  $\bar{x} = \{x_1, x_2, \dots, x_q\}$ , where  $|x_1| = |x_2| = \dots = |x_q|$  and  $q$  is a polynomial in  $n$ . Upon  $\bar{x}$ , an adversary  $V^*$  in the BPK model firstly outputs an arbitrary public-file  $F$  that includes a list of (without loss of generality)  $q$  public-keys  $pk_1, \dots, pk_q$ . Then  $V^*$  concurrently interacts with  $q^2$  instances of the honest prover:  $P(x_i, pk_j)$ ,  $1 \leq i, j \leq q$ , and schedules all the concurrent executions at its wish. We remark that each instance of the honest prover uses independent random strings. Without loss of generality we also assume that messages from  $V^*$  are immediately answered by the honest prover instances.

**Definition 3.1** *We say that a proof or argument system  $\langle P, V \rangle$  for a language  $L$  in the BPK model is black-box concurrent zero-knowledge if there exists a probabilistic polynomial-time (PPT) oracle machine  $S$  (the simulator) such that for any polynomial  $q$  in  $n$  and for any PPT adversary  $V^*$ , the distributions  $\langle P, V^* \rangle(\bar{x})$  and  $S^{V^*}(\bar{x})$  are computationally indistinguishable for any sequence of common inputs  $\bar{x} = x_1, x_2, \dots, x_q \in L \cap \{0, 1\}^n$ .*

**Concurrent soundness in the BPK model.** For an honest verifier  $V$  with public-key  $PK$  and secret-key  $SK$ , an  $(s, t)$ -concurrent malicious prover  $P^*$  in the BPK model, for a pair positive polynomials  $(s, t)$ , be a probabilistic  $t(n)$ -time Turing machine that, on a security parameter  $1^n$  and  $PK$ , performs concurrently at most  $s(n)$  interactive protocols (sessions) with  $V$  as follows.

If  $P^*$  is already running  $i - 1$  ( $0 \leq i - 1 < s(n)$ ) sessions, it can select *on the fly* a common input  $x_i \in \{0, 1\}^n$  (which may be equal to  $x_j$  for  $1 \leq j < i$ ) and initiate a new session with  $V(SK, x_i)$ . We note that in different sessions  $V$  uses independent random-tapes.

We then say a protocol satisfies *concurrent soundness* in the BPK model if for any honest verifier  $V$ , for all positive polynomials  $(s, t)$ , for all  $(s, t)$ -concurrent malicious prover  $P^*$ , the probability that there exists  $i$  ( $1 \leq i \leq s(n)$ ) such that  $V(SK, x_i)$  outputs “accept  $x_i$ ” while  $x_i \notin L$  is negligible in  $n$ .

### 4 Constant-Round Concurrent Zero-Knowledge With Concurrent Soundness in the BPK model

In this section, we present the main result of this paper: a constant-round (specifically, 6-round) concurrent zero-knowledge argument of knowledge with concurrent soundness for  $\mathcal{NP}$  in the BPK model,  $\langle P, V \rangle$ , which is depicted in Figure 1 (page 7).

<b>The protocol</b> $\langle P, V \rangle$
<p><b>Key Generation.</b> For a security parameter <math>n</math>, let <math>(TCPK_0, TCSK_0) \stackrel{R}{\leftarrow} \text{TCGen}(1^n, r_0)</math>, <math>(TCPK_1, TCSK_1) \stackrel{R}{\leftarrow} \text{TCGen}(1^n, r_1)</math>, where <math>r_0</math> and <math>r_1</math> are two independent random strings used by TCGen. <math>(TCPK_0, TCPK_1)</math> is the public-key of the verifier <math>V</math>. But for its secret-key, the verifier <math>V</math> randomly selects a bit <math>b \stackrel{R}{\leftarrow} \{0, 1\}</math> and keeps <math>TCSK_b</math> in secret as its secret-key while discards <math>TCSK_{1-b}</math>.</p> <p>The public file <math>F</math> in the BPK model is a collection of records <math>(id, PK_{id})</math>, where <math>PK_{id} = (TCPK_0^{(id)}, TCPK_1^{(id)})</math> is the alleged public-key of the verifier with identity <math>id</math>, <math>V_{id}</math>. The secret-key of <math>V_{id}</math>, <math>SK_{id}</math>, is <math>TCSK_b^{(id)}</math> for a random bit <math>b</math> in <math>\{0, 1\}</math>.</p>
<p><b>Common input.</b> An element <math>x \in L \cap \{0, 1\}^n</math>. Denote by <math>R_L</math> the corresponding <math>\mathcal{NP}</math>-relation for <math>L</math>.</p>
<p><b>P private input.</b> An <math>\mathcal{NP}</math>-witness <math>y</math> for <math>x \in L</math>.</p>
<p><b>Stage 1.</b> The verifier <math>V</math> proves the knowledge that: he knows either <math>TCSK_0</math> or <math>TCSK_1</math> with respect to his public-key <math>(TCPK_0, TCPK_1)</math>, by using the <math>\Sigma_{OR}</math> protocol of Schnorr's <math>\Sigma</math>-protocol (described in Section 2.1) for proving commitment trapdoors. The witness used by <math>V</math> is its secret-key <math>TCSK_b</math>.</p>
<p><b>Stage 2.</b> The prover <math>P</math> uniformly selects two independent random strings <math>r_P^{(0)} \stackrel{R}{\leftarrow} \{0, 1\}^{NI\sigma Len(n)}</math> and <math>r_P^{(1)} \stackrel{R}{\leftarrow} \{0, 1\}^{NI\sigma Len(n)}</math>, and for two independent random strings <math>s^{(0)}, s^{(1)}</math>, computes <math>\alpha_0 = \text{TCCom}(TCPK_0, r_P^{(0)}, s^{(0)})</math> and <math>\alpha_1 = \text{TCCom}(TCPK_1, r_P^{(1)}, s^{(1)})</math> using the trapdoor commitment scheme TC. Finally, the left player sends <math>(\alpha_0, \alpha_1)</math> to the right player.</p>
<p><b>Stage 3.</b> The verifier <math>V</math> uniformly selects <math>r_V \stackrel{R}{\leftarrow} \{0, 1\}^{NI\sigma Len(n)}</math> and sends <math>r_V</math> to <math>P</math>.</p>
<p><b>Stage 4.</b> <math>P</math> sends <math>r = r_P^{(i)} \oplus r_V</math> to the right player for <math>i \stackrel{R}{\leftarrow} \{0, 1\}</math>.</p>
<p><b>Stage 5.</b> Using a 3-round public-coin WIPOK for <math>\mathcal{NP}</math>, <math>P</math> proves that either <math>\alpha_0</math> or <math>\alpha_1</math> commits to <math>r \oplus r_V</math>. That is, <math>P</math> proves the knowledge of <math>(i, r \oplus r_V, s)</math> such that <math>i \in \{0, 1\}</math> and <math>\alpha_i = \text{TCCom}(TCPK_i, r \oplus r_V, s)</math>. The witness used by <math>P</math> is <math>(i, r_P^{(i)}, s^{(i)})</math> for.</p>
<p><b>Stage 6.</b> Using <math>r</math> as the common input, <math>P</math> gives a NIZKPOK that he knows <math>y</math> such that <math>(x, y) \in R_L</math>.<sup>a</sup></p>
<hr style="width: 30%; margin-left: 0;"/> <p><sup>a</sup>We remark it is not necessarily to use NIZK proof of knowledge in Stage 6. Actually, a NIZK argument of knowledge for <math>\mathcal{NP}</math> does also work here. For example, we can adopt the robust NIZK (presented in [22]) that is a same-string unbounded non-malleable NIZK argument of knowledge for <math>\mathcal{NP}</math>. Here same-string NIZK means that the reference string generated by the simulator of robust NIZK is a uniformly random string rather than a pseudorandom one as usual. The same-string property can be used to simplify future security analyses.</p>

Figure 1. A constant-round concurrent zero-knowledge argument of knowledge with concurrent soundness for  $\mathcal{NP}$  in the BPK model

We remark the protocol depicted in Figure 1 runs in 8 rounds. But it can be reduced into 6 rounds by accordingly combining some rounds. Specifically, the Stage 2 and Stage 3 can be combined into the Stage 1.



**Theorem 4.1** *Under Discrete Logarithm assumption and the existence of trapdoor one-way permutations and dense secure public-key cryptosystems, the protocol depicted in Figure 1 is a constant-round concurrent zero-knowledge argument of knowledge with concurrent soundness for  $\mathcal{NP}$  in the BPK model.*

**Proof.** The completeness of the protocol can be easily checked. Below, we focus on the properties of concurrent zero-knowledge and concurrent soundness.

### Black-box concurrent zero-knowledge

For any adversary  $V^*$  described in Section 3.1, we need to construct a PPT simulator  $S$  such that the output of  $S^{V^*}$  is computationally indistinguishable from the view of  $V^*$  in its real concurrent interactions with honest-prover instances.

The simulation procedure is similar to (but simpler than) the simulation procedure presented in [10] for resettable zero-knowledge. Specifically,  $S$  works in at most  $q + 1$  rounds, where  $q$  is the number of public-keys registered by  $V^*$  in the public-key file. In each round,  $S$  either successfully gets a simulated transcript or “breaks” a new public-key in the sense that  $S$  can extract the corresponding secret-key  $TCSK_b$  (according to the special soundness of  $\Sigma$ -protocol, this is achieved by rewinding  $V^*$  to get two accepting conversations of Stage 1). Once a public-key is broken (that is  $S$  learns  $TCSK_b$ ), then in any session with respect to this broken public-key  $S$  works as follows:  $S$  runs accordingly just as a honest prover in Stage 1-3; Let  $(\alpha_0, \alpha_1)$  be the message sent by  $S$  in Stage 2 (that commit to two independent random values,  $r_P^{(0)}$  and  $r_P^{(1)}$ , respectively) and  $r_{V^*}$  be the message sent by  $V^*$  in Stage 3; In Stage 4,  $S$  runs the NIZK simulator to get a random string, denoted  $\sigma$ , and sends  $\sigma$  to  $V^*$ ; In Stage 5,  $S$  uses  $(b, \sigma \oplus r_{V^*}, s')$  as its witness to give a WIPOK, where  $s' \xleftarrow{R} \text{TCFake}(TCPK_b, TCSK_b, \alpha_b, r_P^{(b)}, s^{(b)}, \sigma \oplus r_{V^*})$ . Finally,  $S$  using the NIZK simulator to give a simulated NIZK (on  $\sigma$ ) in Stage 6.

Since  $S$  runs in at most  $q$  rounds, at during each round  $S$  also works in expected polynomial time, it is easy to see that  $S$  also runs in expected polynomial time in toto. Below, we show that the output of  $S$  is indistinguishable from the view of  $V^*$  in real interactions.

We consider four classes of transcripts: they differ according to the value sent in Stage 4 ( $r_P^{(i)} \oplus r_{V^*}$  or  $\sigma$  generated by NIZK simulator), the witness used in Stage 5, the Stage-6 message (real NIZK or simulated NIZK).

1. Stage-4 message is  $r_P^{(i)} \oplus r_{V^*}$  for  $i \xleftarrow{R} \{0, 1\}$ , Stage-5 witness is  $(\alpha_i, r^{(i)}, s^{(i)})$ . Stage-6 message is a real NIZK.
2. Stage-4 message is  $r_P \oplus r_{V^*}$ , Stage-5 witness is  $(\alpha_b, r_P, s_{r_P}^{(b)})$ , where  $b$  is the secret information of  $V^*$  in its secret-key  $TCSK_b$ . Stage-6 message is a real NIZK.
3. Stage-4 message is  $\sigma$ , Stage-5 witness is  $(\alpha_b, \sigma \oplus r_{V^*}, s')$ . Stage-6 message is a real NIZK.
4. Stage-4 message is  $\sigma$ , Stage-5 witness is  $(\alpha_b, \sigma \oplus r_{V^*}, s')$ . Stage-6 message is a simulated NIZK.

The real transcripts are the first class. The simulator outputs the fourth class. It is easy to see that Class 3 and Class 4 are computationally indistinguishable. Class 1 and Class 2 are computationally indistinguishable by the witness indistinguishability of Stage 5. We note that Class 3 and Class do not make (non-negligibly) noticeable distinguishability gap. Actually, if we adopt robust NIZK in Stage 6 as suggested in the protocol construction, the distributions of Class 2 and Class 3 are identical. The reason is that the  $\sigma$  generated by the simulator of robust NIZK is uniformly distributed. This means  $\sigma$ ,  $\sigma \oplus r_{V^*}$ ,  $r_P$  and  $r_P \oplus r_{V^*}$  are all uniformly random strings. Furthermore, according to the trapdoor property of the trapdoor commitment scheme used,  $s_{r_P}^{(b)}$  and  $s'$  are also independent random strings.

## Concurrent soundness and argument of knowledge

We first note that a computational power unbounded prover can easily convince the verifier of a false statement since he can get the secret-keys if his computational power is unbounded. Hence the protocol  $\langle P, V \rangle$  depicted in Figure 1 constitutes an argument system rather than a proof system.

We now proceed to prove that the protocol  $\langle P, V \rangle$  is concurrent soundness and argument of knowledge. The following proof uses standard reduction and knowledge-extraction techniques. That is, if the protocol  $\langle P, V \rangle$  does not satisfy concurrent soundness in the BPK model then we will construct a non-uniform algorithm  $S$  that breaks the discrete logarithm assumption in expected polynomial-time.

Suppose the protocol  $\langle P, V \rangle$  does not satisfy concurrent soundness in the BPK model, then according to the definition of concurrent soundness in the BPK model (described in Section 3), then in a concurrent attack issued by an  $(s, t)$ -concurrent malicious prover  $P^*$  against an honest verifier  $V$  with public-key  $(TCPK_0, TCSK_1)$  and secret-key  $TCSK_b$  for  $b \xleftarrow{R} \{0, 1\}$ , with non-negligible probability  $p(n)$  there exists an  $i$ ,  $1 \leq i \leq s(n)$ , such that  $V$  outputs “accept  $x_i$ ” while  $x_i \notin L$ . Then we will construct a non-uniform algorithm  $S$  that takes  $(TCPK_0, TCPK_1, TCSK_b)$  as input and outputs either a witness  $w$  such that  $(x_i, w) \in R_L$  or  $TCSK_{1-b}$  with probability  $\frac{p^A(n)}{4s(n)}$  in polynomial-time. For this purpose,  $S$  has oracle access to  $P^*$  and plays the role of  $V$  by running  $V(TCPK_0, TCPK_1, TCSK_b)$  to emulate the real concurrent interactions between  $P^*$  and  $V(TCPK_0, TCPK_1, TCSK_b)$ .

For any  $i$ ,  $1 \leq i \leq s(n)$ , denote by  $k_i \in \{0, 1\}$  the first component in the witness used by  $P^*$  in Stage-5 of  $i$ -th session. We first have the following lemma:

**Proposition 4.1** *For any  $i$ ,  $1 \leq i \leq s(n)$ ,  $k_i$  is independent of  $b$ . That is,  $\Pr[k_i = b] = \frac{1}{2}$ , where the probability is over the coin flips of  $P^*$  and  $V$ .*

**Proof.** We can view the honest  $V$  in two parts: the first part, denoted  $V_1$ , only works in Stage-1. That is,  $V_1$  takes  $TCSK_b$  as its secret input and proves that he knows one of the secret-key with respect to the public-key  $(TCPK_0, TCPK_1)$  in Stage-1. Note that  $V_1$  is itself perfectly witness indistinguishable; the second part, denote  $V_2$ , works in other stages and has no secret information.

Now, suppose there exists an  $i$ ,  $1 \leq i \leq s(n)$ , such that  $k_i$  is not independent of  $b$ , then we can construct a PPT algorithm  $A$  that works as follows to violate the perfect witness indistinguishability of  $V_1$ :  $A$  interacts with  $V_1$  on common input  $(TCPK_0, TCPK_1)$  and runs  $P^*$  and  $V_2$  to emulate the concurrent interactions between  $P^*$  and  $V$ . After the simulation,  $A$  randomly guess the “bad”  $i$  and outputs  $k_i$  as its output. Suppose  $P^*$  distinguishes  $b$  with probability  $p$  then  $A$  will distinguish  $b$  with probability  $p/s(n)$ , which violates the perfect WI property of  $V_1$ .  $\square$

For future reference convenience, below we denote by Stage 5.1, 5.2 and 5.3 the first, the second and the third message of Stage 5 in protocol  $\langle P, V \rangle$  respectively, where Stage 5.2 is supposed to be a random string. Let  $(E_1, E_2)$  be the pair of PPT algorithms guaranteed in the definition of NIZKPOK. Now, consider the following algorithm  $S^{P^*}(1^n, TCPK_0, TCPK_1, TCSK_b)$  depicted in Figure 2 (page 11).

First, we note that  $P^*$  cannot distinguish (except with statistically negligible probability) whether he is interacting with honest verifier  $V$  or with  $S$  since the distribution of  $\sigma$  generated by  $E_1$  is statistically distinguishable from the uniform distribution on  $\{0, 1\}^{NI\sigma Len(n)}$ . This means that, in the concurrent interactions between the  $(s, t)$ -concurrent malicious  $P^*$  and  $S$ , with the same non-negligible probability  $p(n)$  there exists an  $i$ ,  $1 \leq i \leq s(n)$ , such that  $S$  outputs “accept  $x_i$ ” while  $x_i \notin L$ . Furthermore, conditioned on  $P^*$  always succeeds in completing its interactions with  $S$ , then  $\Pr[k_i = k'_i = 1 - b] = \frac{1}{4}$  according to the proposition 4.1. Since  $i$  is uniformly selected by  $S$  from  $\{1, 2, \dots, s(n)\}$ , we conclude

**The algorithm  $S^{P^*}(1^n, TCPK_0, TCPK_1, TCSK_b)$**

$i \xleftarrow{R} \{1, 2, \dots, s(n)\}$ .

Runs  $P^*$  and acts accordingly by running  $V(TCPK_0, TCPK_1, TCSK_b)$  in any session other than the  $i$ -th session. In the  $i$ -th session, denote by  $(\alpha_0^{(i)}, \alpha_1^{(i)})$  the Stage-2 message of the  $i$ -th session,  $S$  acts as follows:

- Uniformly selects  $r_V \xleftarrow{R} \{0, 1\}^{NI\sigma Len(n)}$  and sends  $r_V$  to  $P^*$  as the Stage-3 message of the  $i$ -th session.
- When running into the WIPOK phase (Stage 4-5) of the  $i$ -th session, denoted by  $r$  the Stage-4 message (from  $P^*$ ) of the  $i$ -th session,  $S$  uses the knowledge-extractor of WIPOK to extract the witness used by  $P^*$  in Stage-5 of the  $i$ -th session. This needs to rewind  $P^*$  once and such a rewinding is called the first *knowledge rewinding*. If the extracted value is  $(k_i, t_i, s_i)$  such that  $\alpha_{k_i}^{(i)} = TCCom(TCPK_{k_i}, t_i, s_i)$  and  $t_i = r \oplus r_V$ , where  $t_i$  is of length  $NI\sigma Len(n)$  and  $k_i \in \{0, 1\}$ , then  $S$  does the following:
  1. runs  $E_1$  to get  $(\sigma, \tau) \leftarrow E_1(1^n)$ .
  2. rewinds  $P^*$  to the point  $P^*$  just sent  $(\alpha_0^{(i)}, \alpha_1^{(i)})$  and sends back  $\sigma \oplus t_i$  to  $P^*$  as a new Stage-3 message. Such a rewinding is called the *major rewinding*.
  3. Runs  $P^*$  further (from the major rewinding point). When running into again the WIPOK phase (Stage 4-5), *if the Stage-4 message from  $P^*$  is not  $\sigma$*  then  $S$  uses the knowledge-extractor of WIPOK again to extract the witness used by  $P^*$  in this WIPOK phase. This needs to knowledge-rewind  $P^*$  once more. Denote by  $(k'_i, t'_i, s'_i)$  the witness extracted in the second knowledge-rewinding.
  4. If  $P^*$  successfully gives an NIZKPOK  $\Pi$  on  $\sigma$  for  $x_i$  at Stage-6,  $S$  runs  $E_2(\sigma, \tau, x_i, \Pi)$  to get a witness  $w$ .

Figure 2. The algorithm  $S^{P^*}(1^n, TCPK_0, TCPK_1, TCSK_b)$

that with probability at least  $\frac{p^4}{4s(n)}$ ,  $S$  either get a witness  $w$  such that  $(x_i, w) \in R_L$  or two different decommitments to  $\alpha_{1-b}^{(i)}$  from which the secret-key  $TCSK_{1-b}$  can be easily extracted. This contradicts to the assumption that  $x_i \notin L$  and discrete logarithm is hard. Thus the protocol  $\langle P, L \rangle$  is concurrently sound in the BPK model.

That the system is an argument of knowledge is immediately from the extraction procedure of  $S$ .  $\square$

**Acknowledgments.** The author is deeply indebted to Yehuda Lindell for his deep comments on motivations, numerous kindly clarifications and very valuable discussions and suggestions. The author is full of gratitude to Boaz Barak, Oded Goldreich, Wenbao Mao and Leonid Reyzin for their kindly clarifications and warm helps.

## References

- [1] B. Barak. How to Go Beyond the Black-Box Simulation Barrier. In *IEEE Symposium on Foundations of Computer Science*, pages 106-115, 2001.
- [2] B. Barak. Constant-Round Coin-Tossing With a Man in the Middle or Realizing the Shared Random String Model. In *IEEE Symposium on Foundations of Computer Science*, pages , 2002.
- [3] B. Barak and O. Goldreich. Universal Arguments and Their Applications. In *IEEE Conference on Computational Complexity*, pages 194-203, 2002.
- [4] B. Barak and Y. Lindell. Strict Polynomial-Time in Simulation and Extraction. In *ACM Symposium on Theory of Computing*, pages 484-493, 2002.
- [5] M. Bellare, R. Canetti and H. Krawczyk. A Modular Approach to the Design and Analysis of Authentication and Key-Exchange Protocols. In *ACM Symposium on Theory of Computing*, pages 419-428, 1998.
- [6] M. Blum. Coin Flipping by Telephone. In *proc. IEEE Spring COMPCOM*, pages 133-137, 1982.
- [7] M. Blum. How to Prove a Theorem so No One Else can Claim It. In *Proceedings of the International Congress of Mathematicians*, 1 (2): 1444-1451, 1987.
- [8] Brassard, D. Chaum and C. Crepeau. Minimum Disclosure Proofs of Knowledge. *Journal of Computer Systems and Science*, 37(2): 156-189, 1988.
- [9] R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In *IEEE Symposium on Foundations of Computer Science*, pages 136-145, 2001.
- [10] R. Canetti, O. Goldreich, S. Goldwasser and S. Micali. Resettable Zero-Knowledge. In *ACM Symposium on Theory of Computing*, pages 235-244, 2000.
- [11] R. Canetti, J. Kilian, E. Petrank and A. Rosen. Black-Box Concurrent Zero-Knowledge Requires  $\tilde{\Omega}(\log n)$  Rounds. In *ACM Symposium on Theory of Computing*, pages 570-579, 2001.
- [12] R. Canetti, Y. Lindell, R. Ostrovsky and A. Sahai. Universally Composable Two-Party and Multi-Party Secure Computation. In *ACM Symposium on Theory of Computing*, pages 494-503, 2002.
- [13] R. Cramer and I. Damgard. On Electronic Payment Systems. A lecture note for the course of Cryptographic Protocol Theory at Aarhus University, 2003. Available from: <http://www.daimi.au.dk/~ivan/CPT.html>
- [14] R. Cramer, I. Damgard and B. Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *Y. Desmedt (Ed.): Advances in Cryptology-Proceedings of CRYPTO 1994, LNCS 839*, pages 174-187. Springer-Verlag, 1994.
- [15] G. D. Crescenzo, Y. Ishai and R. Ostrovsky. Non-Interactive and Non-Malleable Commitment In *ACM Symposium on Theory of Computing*, pages 141-150, 1998.

- [16] G. D. Crescenzo, J. Katz, R. Ostrovsky and A. Smith. Efficient and Non-Interactive Non-Malleable Commitments. In *B. Pfitzmann (Ed.): Advances in Cryptology-Proceedings of EUROCRYPT 2001, LNCS 2045*, pages 40-59. Springer-Verlag, 2001.
- [17] G. D. Crescenzo and R. Ostrovsky. On Concurrent Zero-Knowledge with Pre-Processing. In *M. J. Wiener (Ed.): Advances in Cryptology-Proceedings of CRYPTO 1999, LNCS 1666*, pages 485-502. Springer-Verlag, 1999.
- [18] I. Damgard. On  $\Sigma$ -protocols. A lecture note for the course of Cryptographic Protocol Theory at Aarhus University, 2003. Available from: <http://www.daimi.au.dk/~ivan/CPT.html>
- [19] I. Damgard and J. Groth. Non-interactive and reusable non-malleable commitment schemes. In *ACM Symposium on Theory of Computing*, pages 426-437, 2003.
- [20] I. Damgard and J. B. Nielsen. Perfect Hiding and Perfect Binding Universally Composable Commitment Schemes with Constant Expansion Factor. In *M. Yung (Ed.): Advances in Cryptology-Proceedings of CRYPTO 2002, LNCS 2442*, pages 581-596. Springer-Verlag, 2002.
- [21] I. B. Damgard, T. P. Pedersen and B. Pfitzmann. On the Existence of Statistically Hiding Bit Commitment Schemes and Fail-Stop Signatures. *Journal of Cryptology*, 10(3): 163-194, 1997.
- [22] A. D. Santis, G. D. Crescenzo, R. Ostrovsky, G. Persiano and A. Sahai. Robust Non-Interactive Zero-Knowledge. In *J. Kilian (Ed.): Advances in Cryptology-Proceedings of CRYPTO 2001, LNCS 2139*, pages 566-598. Springer-Verlag, 2001.
- [23] A. D. Santis and G. Persiano. Zero-Knowledge Proofs of Knowledge Without Interaction. In *IEEE Symposium on Foundations of Computer Science*, pages 427-436, 1992.
- [24] D. Dolev, C. Dwork and M. Naor. Non-Malleable Cryptography. In *ACM Symposium on Theory of Computing*, pages 542-552, 1991.
- [25] C. Dwork and M. Naor. Zaps and Their Applications. In *IEEE Symposium on Foundations of Computer Science*, pages 283-293, 2000. Available on-line from: <http://www.wisdom.weizmann.ac.il/~naor/>
- [26] C. Dwork, M. Naor and A. Sahai. Concurrent Zero-Knowledge. In *ACM Symposium on Theory of Computing*, pages 409-418, 1998.
- [27] U. Feige, D. Lapidot and A. Shamir. Multiple Non-Interactive Zero-Knowledge Proofs Under General Assumptions. *SIAM Journal on Computing*, 29(1): 1-28, 1999.
- [28] M. Fischlin and R. Fischlin. Efficient Non-Malleable Commitment Schemes. In *M. Bellare (Ed.): Advances in Cryptology-Proceedings of CRYPTO 2000, LNCS 1880*, pages 413-431. Springer-Verlag, 2000.
- [29] A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *A. Odlyzko (Ed.): Advances in Cryptology-Proceedings of CRYPTO'86, LNCS 263*, pages 186-194. Springer-Verlag, 1986.
- [30] J. A. Garay, P. MacKenzie and K. Yang. Strengthening Zero-Knowledge Protocols Using Signatures. In *E. Biham (Ed.): Advances in Cryptology-Proceedings of EUROCRYPT 2003, LNCS 2656*, pages 177-194. Springer-Verlag, 2003.
- [31] O. Goldreich. *Foundation of Cryptography-Basic Tools*. Cambridge University Press, 2001.
- [32] O. Goldreich and H. Krawczyk. On the Composition of Zero-Knowledge Proof Systems. *SIAM Journal on Computing*, 25(1): 169-192, 1996.
- [33] O. Goldreich, S. Micali and A. Wigderson. How to Play any Mental Game-A Completeness Theorem for Protocols with Honest Majority. In *ACM Symposium on Theory of Computing*, pages 218-229, 1987.
- [34] O. Goldreich, S. Micali and A. Wigderson. Proofs that Yield Nothing But Their Validity or All language in  $\mathcal{NP}$  Have Zero-Knowledge Proof Systems. *Journal of the Association for Computing Machinery*, 38(1): 691-729, 1991.
- [35] S. Goldwasser, S. Micali and C. Rackoff. The Knowledge Complexity of Interactive Proof System. *SIAM Journal on Computing*, 18(1): 186-208, 1989.

- [36] S. Halevi and S. Micali. Practical and Provably-Secure Commitment Schemes From Collision-Free Hashing. In *N. Kobitz (Ed.): Advances in Cryptology-Proceedings of CRYPTO 1996, LNCS 1109*, pages 201-215. Springer-Verlag, 1996.
- [37] J. Kilian, E. Petrank. An Efficient Non-Interactive Zero-Knowledge Proof System for  $\mathcal{NP}$  with General Assumptions. *Journal of Cryptology*, 11(2): 24, 1998.
- [38] J. Kilian, E. Petrank, R. Richardson. Concurrent and Resettable Zero-Knowledge in Poly-Logarithmic Rounds. In *ACM Symposium on Theory of Computing*, pages 560-569, 2001.
- [39] Y. Lindell. Parallel Coin-Tossing and Constant-Round Secure Two-Party Computation. In *J. Kilian (Ed.): Advances in Cryptology-Proceedings of CRYPTO 2001, LNCS 2139*, pages 171-189. Springer-Verlag, 2001.
- [40] Y. Lindell. Composition of Secure Multi-Party Protocols - A Comprehensive Study. LNCS 2815, Springer-Verlag, 2003.
- [41] S. Micali and L. Reyzin. Soundness in the Public-Key Model. In *J. Kilian (Ed.): Advances in Cryptology-Proceedings of CRYPTO 2001, LNCS 2139*, pages 542-565. Springer-Verlag, 2001.
- [42] S. Micali and L. Reyzin. Min-Round Resettable Zero-Knowledge in the Public-Key Model. In *B. Pfitzmann (Ed.): Advances in Cryptology-Proceedings of EUROCRYPT 2001, LNCS 2045*, pages 373-393. Springer-Verlag, 2001.
- [43] D. Micciancio and E. Petrank. Simulatable Commitments and Efficient Concurrent Zero-Knowledge. In *E. Biham (Ed.): Advances in Cryptology-Proceedings of EUROCRYPT 2003, LNCS 2656*, pages 140-159. Springer-Verlag, 2003.
- [44] M. Prabhakaran, A. Rosen and A. Sahai. Concurrent Zero-Knowledge With Logarithmic Round Complexity. In *IEEE Symposium on Foundations of Computer Science*, pages 366-375, 2002.
- [45] R. Richardson and J. Killian. On the Concurrent Composition of Zero-Knowledge Proofs. In *J. Stern (Ed.): Advances in Cryptology-Proceedings of EUROCRYPT 1999, LNCS 1592*, pages 415-423. Springer-Verlag, 1999.
- [46] A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen Ciphertext Security. In *IEEE Symposium on Foundations of Computer Science*, pages 543-553, 1999.
- [47] C. Schnorr. Efficient Signature Generation by Smart Cards. *Journal of Cryptology*, 4(3): 24, 1991.
- [48] A. C. Yao. How to Generate and Exchange Secrets. In *IEEE Symposium on Foundations of Computer Science*, pages 162-167, 1986.
- [49] Y. Zhao, X. Deng, C. H. Lee and H. Zhu. Resettable Zero-Knowledge in the Weak Public-Key Model. In *E. Biham (Ed.): Advances in Cryptology-Proceedings of EUROCRYPT 2003, LNCS 2656*, pages 123-140. Springer-Verlag, 2003.