Resettable Zero-Knowledge in the Weak Public-Key Model

Yunlei Zhao[1,3], Xiaotie Deng[2], C. H. Lee[2], and Hong Zhu[3]

[1] Software School
Fudan University, Shanghai, China
CSYLZHAO@cityu.edu.hk

[2] Department of Computer Science
City University of Hong Kong, Hong Kong
{CSDENG, CHLEE}@cityu.edu.hk

[3] Department of Computer Science
Fudan University, Shanghai, China

Abstract. A new public-key model for resettable zero-knowledge (rZK) protocols, which is an extension and generalization of the upper-bounded public-key (UPK) model introduced by Micali and Reyzin [EuroCrypt'01, pp. 373-393], is introduced and is named weak public-key (WPK) model. The motivations and applications of the WPK model are justified in the distributed smart-card/server setting and it seems more preferable in practice, especially in E-commerce over Internet. In this WPK model a 3-round (optimal) black-box resettable zero-knowledge argument with concurrent soundness for $\mathcal{NP}$ is presented assuming the security of RSA with large exponents against subexponential-time adversaries. Our result improves Micali and Reyzin's result of resettable zero-knowledge argument with concurrent soundness for $\mathcal{NP}$ in the UPK model. Note that although Micali and Reyzin' protocol satisfies concurrent soundness in the UPK model, but it does not satisfy even sequential soundness in our WPK model.

Our protocol works in a somewhat "parallel repetition" manner to reduce the error probability and the black-box zero-knowledge simulator works in strict polynomial time rather than expected polynomial time. The critical tools used are: verifiable random functions introduced by Micali, Rabin and Vadhan [FOCS'99, pp. 120-130], zap presented by Dwork and Naor [FOCS'00, pp.283-293] and complexity leveraging introduced by Canetti, Goldreich, Goldwasser and Micali [STOC'00, pp. 235-244].

1 Introduction

The strongest notion of zero-knowledge to date, resettable zero-knowledge (rZK), was recently put forward by Canetti, Goldreich, Goldwasser and Micali [8]. Roughly speaking, an rZK protocol is an interactive system in which a verifier learns nothing (except for the verity of a given statement) even if he can interact with the prover polynomial many times, each time restarting an interaction with the prover using the same configuration and random tape. rZK enlarges the number of physical ways to implement zero-knowledge protocols while guaranteeing security is preserved. For example, rZK makes it possible to implement the zero-knowledge prover by using those devices that may be possibly (maliciously) resetted to their initial conditions or can not afford to generate fresh randomness for each new invocation. An example of those devices is the ordinary smart card. rZK is also guaranteed to preserve the prover's security when the protocol is executed concurrently in an asynchronous network like the Internet. Actually, rZK is a generalization and strengthening of the notion of concurrent zero-knowledge introduced by Dwork, Naor and Sahai [12].

1.1 Previous results

Under standard complexity assumptions, non-constant-round resettable zero-knowledge proof for $\mathcal{NP}$ was constructed in [8, 22] by properly modifying the concurrent zero-knowledge protocol of Richardson and Killian [28]. Unfortunately, there are no constant-round rZK protocols in the standard model, at least for the black-box case, as shown by Canetti, Killian, Petrank and Rosen [9]. To get constant-round resettable zero-knowledge protocols Canetti, Goldreich, Goldwasser and Micali [8] introduced an appealingly simple model, the *bare public-key* (BPK) model, and presented a 5-round rZK argument for $\mathcal{NP}$ in this model. The round complexity was further reduced to four by Micali and Reyzin [24].

A protocol in the BPK model simply assumes that all verifiers have deposited a public key in a public file before any interaction 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. We remark that the BPK model is a weak version of the frequently used Public-Key Infrastructure (PKI) model, which underlies any public key cryptosystem or digital signature.

Resettable zero-knowledge protocols also shed hope on finding ID schemes secure against resetting attack. Feige, Fiat and Shamir [16, 14] introduced a paradigm for ID schemes based on the notion of zero-knowledge proof of knowledge. In essence, a prover identifies himself by convincing the verifier of knowing a given secret. Almost all subsequent ID schemes followed this paradigm, and were traditionally implemented by the prover being a smart card. However, up to the emergence of rZK all the previous Fiat-Shamir like ID schemes fail to secure whenever the prover is resettable. Using constant-round rZK protocols in the BPK model above, Bellare, et al. [3] provided identification protocols secure against resetting attack. Unfortunately, there is a main disadvantage of this rZK-based solution since it only preserves the identity prover's security but does not guarantee to preserve any security of the identity verifier when the identification protocol is concurrently executed in an asynchronous setting like the Internet. Actually, if an adversary is allowed to concurrently interact with the identity verifiers then the adversary can easily impersonate the identity prover. The reason is just that the underlying resettable zero-knowledge protocols in the

BPK model [8, 24] do not guarantee to preserve verifier's security when they are concurrently executed.

The various security notions of the verifier in public-key models were first noted and clarified by Micali and Reyzin [24, 27]. 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 [24] four soundness notions in public-key models were defined in which each implies the previous one: *one-time soundness, sequential soundness, concurrent soundness, resettable soundness.* In this paper we focus on concurrent soundness which roughly means 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. As discussed above, resettable zero-knowledge protocols with concurrent soundness are really desirable in most smart-card and Internet based applications. 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. In a stronger version of BPK model introduced by Micali and Reyzin [25] in which each public-key of an honest verifier is restricted to be used at most a priori bounded polynomial times, the *upper-bounded public-key* (UPK) model, Micali and Reyzin gave a 3-round black-box rZK argument with sequential soundness for $\mathcal{NP}$ in the UPK model [25]. And Reyzin [27] further proved that it also satisfies concurrent soundness in the UPK model.

Regarding the round-complexity of resettable zero-knowledge protocols for $\mathcal{NP}$ in public-key models, Micali and Reyzin [24, 25] showed that any (resettable or not) black-box zero-knowledge protocol in public-key models for a language outside of $\mathcal{BPP}$ requires at least three rounds (using an earlier result of Goldreich and Kraczwyck [20]). For efficient 4-round zero-knowledge protocols for $\mathcal{NP}$, readers are referred to [7]. We also note that 2-round public-coin black-box and concurrent zero-knowledge protocols for $\mathcal{NP}$ do exist under the assumption that the prover is *resource bounded*[13]. Here, resource bounded prover means that during protocol execution the prover uses certain limited amount of (say, a-priori polynomial bounded) time or non-uniform advice.

1.2 Our contributions

In this paper, we introduce a new public-key model for resettable zero-knowledge (rZK) protocols, which we name it *weak public-key* (WPK) model. Roughly speaking, in the WPK model the public-key of an honest verifier V can be used by an (even malicious) prover P^* for any polynomial times just as allowed in the BPK model. But for each theorem statement x selected by P^* *on the fly* x is restricted to be used at most a priori bounded polynomial times with respect to the same verifier's public key. Note that if the same verifier's public-key is restricted to be used at most a priori bounded polynomial times just as required in the UPK model then for each common input x selected by P^* x is also restricted to be used at most a priori bounded polynomial times with respect to the same verifier's public key. It means the WPK model is an extension and generalization of the UPK model. Really, the WPK model just lies between the BPK model and the UPK model. That is, the WPK model is stronger than the BPK model but *weaker* than the UPK model.

The main result of this paper is a 3-round black-box resettable zero-knowledge argument with concurrent soundness for $\mathcal{NP}$ in the WPK model. The round complexity is optimal according to Micali and Reyzin's result. In comparison with Micali and Reyzin's 3-round rZK argument with concurrent soundness for $\mathcal{NP}$ in the UPK model [25], we remark that our protocol in the WPK model is also an rZK argument with concurrent soundness for $\mathcal{NP}$ in the UPK model since the WPK model is an extension and generalization of the UPK model. But, the concurrent soundness of Micali and Reyzin's protocol is not preserved in our WPK model. The reason is that the concurrent soundness of Micali and Reyzin's protocol relies on the restriction that the public-key of V can not be used by P^* more than a priori bounded polynomial times and without this restriction P^* can easily cheat V with non-negligible probability (even with probability 1). Since this restriction is removed in our WPK model, it means that Micali and Reyzin's protocol not only does not satisfy concurrent soundness in our WPK model but also even does not satisfy sequential soundness in the WPK model. Our protocol can be viewed as an improvement and extension of Micali and Reyzin' result.

Motivations, applications and implementation of the WPK model As an extension and generalization of the UPK model, roughly speaking, almost all the ways to implement the UPK model [25] can also be used to implement our WPK model. A simple way is to just let the honest verifier to keep a counter for each common input on which he has been invoked. This is an extension of the implementation of the UPK model in which an honest verifier keeps a single counter for all common inputs (selected on the fly by a malicious prover) on which he has been invoked.

Note that one of the major applications of resettable zero-knowledge is that it makes it possible to implement zero-knowledge prover by those devices that may be possibly maliciously resetted to their initial conditions or can not afford to generate fresh randomness for each invocation. The most notable example of such devices is the widely used smart card. Also as argued above resettable zero-knowledge provides the basis for identification protocols secure against resetting attacks [3]. Then we consider the distributed client/server setting in which the clients are implemented by smart cards. We remark that this setting is widely used in practice, especially in E-commerce over Internet. When a resettable identification protocol is executed in this distributed smart-card/server setting we view the identity of each smart-card as the common input. An adversary may hold many (any polynomial number of) smart-cards but in our WPK model we require that each smart-card can be used by the adversary at most a priori polynomial times. Note that in practice each smart-card has an expiry date that corresponds to in some level the a-priori bounded polynomial restriction required in our WPK model. We remark that in this distributed smart-card/server setting there usually exists a central server that may be located in a central bank or other organizations and plays the verifier's role. In practice the central server keeps a record for each smart card and dynamically updates its information. It is easy for this central server to keep a counter in each record to remember how many times the corresponding smart-card has been used. We stress that in this distributed smart-card/server setting since the server (verifier) may be invoked and interacted concurrently with many smart-cards, the design of rZK protocols with concurrent soundness in the WPK model is really desirable.

1.3 Organization of this paper

In Section 2, we recall the tools we will use in this paper. In Section 3, we provide the formal description of the WPK model. In Section 4, we present the 3-round black-box resettable zero-knowledge argument with concurrent soundness for $\mathcal{NP}$ in the WPK model.

2 Preliminaries

In this section, we present some main tools used in this paper. However, one critical tool, zap presented in [11], is absent from this section and is provided in Section 3 together with the definition of resettable witness indistinguishability. We remark that all these tools can be constructed assuming the security of RSA with large prime exponents against subexponential-time adversaries.

Definition 1 (one-round perfect-binding commitments). *A one-round perfect-binding commitment scheme is a pair of probabilistic polynomial-time (PPT) algorithms, denoted* (C, R)*, satisfying:*

- *Completeness.* $\forall k$, $\forall v$, *let* $c = C_{s_v}(1^k, v)$ *and* $d = (v, s_v)$, *where* C *is a PPT commitment algorithm while using* s_v *as its randomness and* d *is the corresponding decommitment to* c, *it holds that* $\Pr[(c, d) \xleftarrow{R} C(1^k, v) : R(1^k, c, v, d) = YES] = 1$, *where* k *is security parameter.*
- *Computational hiding. For every* v, u *of equal* $p(k)$*-length, where* p *is a positive polynomial and* k *is security parameter, the random variables* $C_{s_v}(1^k, v)$ *and* $C_{s_u}(1^k, u)$ *are computationally indistinguishable.*
- *Perfect binding. For every* v, u *of equal* $p(k)$*-length, the random variables* $C_{s_v}(1^k, v)$ *and* $C_{s_u}(1^k, u)$ *have disjoint support. That is, for every* v, u *and* m, *if* $\Pr[C_{s_v}(1^k, v) = m]$ *and* $\Pr[C_{s_u}(1^k, u) = m]$ *are both positive then* $u = v$ *and* $s_v = s_u$.

A one-round perfect-binding commitment scheme can be constructed based on any one-way permutation [17].

Definition 2 (Pseudorandom Functions PRF [19]). *A pseudorandom function family is a keyed family of efficiently computable functions, such that a* *function picked at random from the family is indistinguishable (via oracle access) from a truly random function with the same domain and range. Formally, a function PRF:* $\{0,1\}^n \times \{0,1\}^* \rightarrow \{0,1\}^n$ *is a pseudorandom function if for all* 2^{n^α}*-size adversaries ADV, the following difference is negligible in* n*:*

$$\left| \Pr\left[PRFKey \xleftarrow{R} \{0,1\}^n : ADV^{PRF(PRFKey,\,\cdot)} = 1\right] - \Pr\left[F \xleftarrow{R} (\{0,1\}^n)^{\{0,1\}^*} : ADV^{F(\cdot)} = 1\right] \right|$$

The value α *is called the pseudorandomness constant.*

Definition 3 (non-interactive zero-knowledge NIZK [2, 4]). *Let* NIP *and* NIV *be two probabilistic interactive machines, and let* $NI\sigma Len$ *be a positive polynomial. We say that* $< NIT, NIV >$ *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* σ *of length* $NI\sigma Len(n)$*, and* $\mathcal{NP}$*-witness* y *for* $x \in L$*, it holds that* $\Pr[\Pi \xleftarrow{R} NIP(\sigma, x, y) : NIV(\sigma, x, \Pi) = YES] = 1$.
- *Soundness.* $\forall x \notin L$ *of length* n*,* $\Pr[\sigma \xleftarrow{R} \{0,1\}^{NI\sigma Len(n)} : \exists\, \Pi$ *s. t.* $NIV(\sigma, x, \Pi) = YES]$ *is negligible in* n.
- *Zero-Knowledge.* $\exists\, \alpha > 0$ *and a PPT simulator* NIS *such that,* $\forall$ *sufficiently large* n*,* $\forall x \in L$ *of length* n *and* $\mathcal{NP}$*-witness* y *for* $x \in L$*, the following two distributions are indistinguishable by any* 2^{n^α}*-gate adversary:*
  $[(\sigma', \Pi') \xleftarrow{R} NIS(x) : (\sigma', \Pi']$ *and*
  $[\sigma \xleftarrow{R} \{0, 1\}^{NI\sigma Len(n)}; \Pi \xleftarrow{R} NIP(\sigma, x, y) : (\sigma, \Pi)]$.
  The value α *is called the NIZK constant.*

Non-interactive zero-knowledge proof systems for $\mathcal{NP}$ can be constructed based on any one-way permutation [15] and one-way permutations can be constructed in turn under RSA assumption [18]. An efficient implementation based on any one-way permutation can be found in [21]. For more recent advances in NIZK readers are referred to [10].

2.1 Verifiable random functions

A family of verifiable random functions (VRF), first introduced in [26], is essentially a pseudorandom function family with an additional property that the correct value of a function on an input can not only be computed by the owner of the seed, but also be proven to be the unique correct value. The proof can be verified by anyone who knows the public-key corresponding to the seed.

Definition 4 (Verifiable Random Functions). *Let* `VRFGen`*,* `VRFEval`*,* `VRFProve` *and* `VRFVer` *be polynomial-time algorithms (the first and the last are probabilistic, and the middle two are* deterministic*). Let* $a : N \rightarrow N \cup \{0,1\}^*$

and $b : N \to N$ *be any two functions that are computable in time* $poly(k)$ *and bounded by a polynomial in* k *(except when* a *takes on* $\{0,1\}^*$*).*

We say that (VRFGen, VRFEval, VRFProve, VRFVer) is a verifiable pseudorandom function (VRF) with input length $a(k)$ *and output length* $b(k)$ *under a security parameter* k *if the following properties hold:*

1. *The following two conditions hold with probability* $1 - 2^{-\Omega(k)}$ *over the choice of*
   $(VRFPK, VRFSK) \xleftarrow{R} \mathtt{VRFGen}(1^k)$:
   (a) *(Domain-Range Correctness):*
   $$\forall x \in \{0,1\}^{a(k)}, \mathtt{VRFEval}(VRFSK, x) \in \{0,1\}^{b(k)}.$$
   (b) *(Complete Probability):* $\forall x \in \{0,1\}^{a(k)}$*, if* $v = \mathtt{VRFEval}(VRFSK, x)$ *and* $pf = \mathtt{VRFProve}(VRFSK, x)$*, then*
   $$\Pr[\mathtt{VRFVer}(VRFPK, x, v, pf) = \mathit{YES}] > 1 - 2^{-\Omega(k)}$$
   (This probability is over the coin tosses of VRFVer).
2. *(Unique Probability) For every* $VRFPK, x, v_1, v_2, pf_1, pf_2$ *such that* $v_1 \neq v_2$*, the following holds for either* $i = 1$ *or* $i = 2$*:*
   $$\Pr[\mathtt{VRFVer}(VRFPK, x, v_i, pf_i) = \mathit{YES}] < 2^{-\Omega(k)}$$
   (This probability is over the coin tosses of VRFVer).
3. *(Residual Pseudorandomness): Let* $\alpha > 0$ *be a constant. Let* $T = (T_E, T_J)$ *be any pair of algorithms such that* $T_E(\cdot,\cdot)$ *and* $T_J(\cdot,\cdot)$ *run for a total of at most* 2^{k^α} *steps when their first input is* 1^k*. Then the probability that* T *succeeds in the following experiment is at most* $1/2 + 1/2^{k^\alpha}$*:*
   (a) *Run* $\mathtt{VRFGen}(1^k)$ *to obtain* $(VRFPK, VRFSK)$*.*
   (b) *Run* $T_E^{\mathtt{VRFEval}(VRFSK,\,\cdot),\mathtt{VRFProve}(VRFSK,\,\cdot)}(1^k, VRFPK)$ *to obtain the pair* $(x, state)$*.*
   (c) *Choose* $r \xleftarrow{R} \{0,1\}$*.*
      – *if* $r = 0$*, let* $v = \mathtt{VRFEval}(VRFSK, x)$
      – *if* $r = 1$*, choose* $v \xleftarrow{R} \{0,1\}^{b(k)}$
   (d) *Run* $T_J^{\mathtt{VRFEval}(VRFSK,\,\cdot),\mathtt{VRFProve}(VRFSK,\,\cdot)}(1^k, VRFPK, v, state)$ *to obtain a guess.*
   (e) $T = (T_E, T_J)$ *succeeds if* $x \in \{0,1\}^{a(k)}$*,* $guess = r$*, and* x *was not asked by either* T_E *or* T_J *as a query to* $\mathtt{VRFEval}(VRFSK,\cdot)$ *or* $\mathtt{VRFProve}(VRFSK,\cdot)$*.*
   We call α *the pseudorandomness constant.*

The above verifiable pseudorandom functions can be constructed assuming RSA with large prime exponents against subexponential-time adversaries [26]. Very recently, a new construction of VRF was provided by Lysyanskaya on an assumption about groups in which decisional Diffie-Hellman is easy, but computational Diffie-Hellman is hard [23]. We remark that up to now the first application of VRF, as suggested by Micali and Reyzin, is the simple construction of an rZK argument with one-time soundness for $\mathcal{NP}$ in the BPK model [24]. Our result can be viewed as another major application of VRF.

3 The Weak Public-Key (WPK) Model

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

3.1 Honest players in the WPK model

The WPK model consists of the following:

- F be a public-key file that is a polynomial-size collection of records (id, PK_{id}), where id is a string identifying a verifier and PK_{id} is its (alleged) public-key.
- $P(1^n, x, y, F, id, w)$ be an honest prover that is a *polynomial-time* interactive machine, where 1^n is a security parameter, x is an n-bit string in L, y is an auxiliary input, F is a public-file, id is a verifier identity, and w is his random-tape.
- V be an honest verifier that is an polynomial-time interactive machine working in two stages.
  1. Key generation stage. V, on a security parameter 1^n and a random-tape r, outputs a public-key PK and remembers the corresponding secret key SK.
  2. Verification stage. V, on inputs SK, $x \in \{0,1\}^n$ and a random tape ρ, performs an interactive protocol with a prover and outputs "accept x" or "reject x". We stress that in our WPK model for each common input x on which the verification stage of V has been invoked the honest verifier V keeps a counter in secret with upperbound $U(n)$, a priori bounded polynomial, to remember how many times the verification stage has been invoked on the same x and refuses to participate in other interactions with respect to the same x once the counter reading reaches its upperbound $U(n)$. It means that each common input x can not be used (even by a malicious prover) more than $U(n)$ times with respect to the same PK_{id}, where id is the identity of the *honest* verifier V.

3.2 The malicious resetting verifier and resettable zero-knowledge

A malicious (s,t)-resetting malicious verifier V^*, where t and s are positive polynomials, is a $t(n)$-time TM working in two stages so that, on input 1^n,

Stage 1 V^* receives $s(n)$ *distinct* strings $x_1, \cdots, x_{s(n)}$ of length n each, and outputs an arbitrary public-file F and a list of (without loss of generality) $s(n)$ identities $id_1, \cdots, id_{s(n)}$.

Stage 2 Starting from the final configuration of Stage 1, $s(n)$ random tapes, $w_1, \cdots, w_{s(n)}$, are randomly selected and then fixed for P, resulting in $s(n)^3$ deterministic prover strategies $P(x_i, id_j, w_k)$, $1 \leq i, j, k \leq s(n)$. V^* is then given oracle access to these $s(n)^3$ provers, and finally outputs its "view" of the interactions (i. e. its random tape and messages received from all his oracles).

Definition 5 (Black-box Resettable Zero-Knowledge). *A protocol $< P, V >$ is black-box resettable zero-knowledge for a language $L \in \mathcal{NP}$ if there exists a black-box simulator M such that for every (s,t)-resetting verifier V^*, the following two probability distributions are indistinguishable. Let each distributions be indexed by a sequence of common* distinct *inputs $\bar{x} = x_1, \cdots, x_{s(n)} \in L \cap \{0,1\}^n$ and their corresponding NP-witnesses $aux(\bar{x}) = y_1, \cdots, y_{s(n)}$:*

Distribution 1 *The output of V^* obtained from the experiment of choosing $w_1, \cdots, w_{s(n)}$ uniformly at random, running the first stage of V^* to obtain F, and then letting V^* interact in its second stage with the following $s(n)^3$ instances of P: $P(x_i, y_i, F, id_j, w_k)$ for $1 \leq i, j, k \leq s(n)$. Note that V^* can oracle access to these $s(n)^3$ instances of P.*

Distribution 2 *The output of $M^{V^*}(x_1, \cdots, x_{s(n)})$.*

Remark 1. In Distribution 1 above, since V^* oracle accesses to $s(n)^3$ instances P: $P(x_i, y_i, F, id_j, w_k)$, $1 \leq i, j, k \leq s(n)$, it means that V^* may invoke and interact with the same $P(x_i, y_i, F, id_j, w_k)$ multiple times, where each such interaction is called a session. We remark that there are two versions for V^* works in Distribution 1.

1. Sequential version. In this version, a session must be terminated (either completed or aborted) before V^* initiating a new session. That is, V^* is required to terminate its current interaction with the current oracle $P(x_i, y_i, F, id_j, w_k)$ before starting an interaction with any $P(x_{i'}, y_{i'}, F, id_{j'}, w_{k'})$, regardless of $(i, j, k) = (i', j', k')$ or not. Thus, the activity of V^* proceeds in rounds. In each round it selects one of his oracles and conducts a complete interaction with it.
2. Interleaving version. In this version the above restriction is removed and so V^* may initiate and interact with $P(x_i, y_i, F, id_j, w_k)$s concurrently in many sessions. That is, we allow V^* to send arbitrary messages to each of the $P(x_i, y_i, F, id_j, w_k)$ and obtain the response of $P(x_i, y_i, F, id_j, w_k)$ to such message.

However, these two versions are equivalent as shown in [8]. In other words, interleaving interactions do not help the malicious verifier get more advantages on learning knowledge from his oracles than he can do by sequential interactions. Without loss of generality, in the rest of this paper we assume the resetting malicious verifier V^* works in the sequential version.

Definition 6 (Resettable Witness Indistinguishability rWI). *A protocol $< P, V >$ is said to be resettable witness indistinguishable for an $L \in \mathcal{NP}$ if for every pair of positive polynomials (s,t), for every (s,t)-resetting malicious verifier V^*, two distribution ensembles of Distribution 1 (defined in Definition 5), which are indexed by the same $\bar{x}$ but possibly different sequence of prover's $\mathcal{NP}$-witnesses : $aux^{(1)}(\bar{x}) = y_1^{(1)}, \cdots, y_{s(n)}^{(1)}$ and $aux^{(2)}(\bar{x}) = y_1^{(2)}, \cdots, y_{s(n)}^{(2)}$, are computationally indistinguishable.*

In [8] Canetti et al. first gave a 4-round rWI for $\mathcal{NP}$. The round-complexity is drastically reduced to 2 by Dwork and Naor [11], where they named such a 2-round WI a zap.

Dwork and Naor's 2-round rWI proof for $\mathcal{NP}$ [11] The prover P has a private random string s that determines a pseudorandom function f_s. Let L be an $\mathcal{NP}$-Complete language and R_L be its corresponding $\mathcal{NP}$ relation. Under a security parameter n, let p be a positive polynomial and $x \in \{0,1\}^n$ be the common input and y be the corresponding $\mathcal{NP}$-witness (kept in secret by the prover) for $x \in L$.

Step 1 The verifier V uniformly selects (fixes once and for all) $p(n)$ random strings $R_V = (R_{V_1}, R_{V_2}, \cdots, R_{V_{p(n)}})$ with length $NI\sigma Len(n)$ each and sends them to P.

Step 2 Let $f_s(x, y, R_V) = (r_1, r_2, \cdots, r_{p(n)}, R_P)$, where the length of R_P is also $NI\sigma Len(n)$. For each i, $1 \leq i \leq p(n)$, on x and y, P uses r_i as its randomness to compute an NIZK proof Π_i with respect to common random string $R_P \oplus R_{V_i}$. In the rest of this paper we denote by Π_i $NIZK(x, R_P \oplus R_{V_i})$, $1 \leq i \leq p(n)$. Finally P sends R_P along with all the $p(n)$ NIZK proofs to V.

An interesting property of Dwork and Naor's 2-round rWI is that R_V in Step 1 can be fixed once and for all and applied to any instance of length n [11]. It means R_V can be posted in one's public key in the public-key model. We will use this property in our construction later. We also note that using the general result of existence of zaps for $\mathcal{NP}$ (rather than the above specific NIZK-based construction) may further simplify the construction of the protocol presented in Section 4. We will investigate it in a late full version.

3.3 Concurrent soundness in the WPK model

For an honest verifier V with public-key PK and secret-key SK, an (s,t)-concurrent malicious prover P^* in our WPK 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 $i-1$ $(1 \leq i-1 \leq s(n))$ 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 the verification stage of $V(SK, x_i, \rho_i)$ on the restriction that the same x_i can not be used by P^* in more than $U(n)$ sessions, where $U(n)$ is the a priori bounded polynomial indicating the upper-bound of the corresponding counter kept in secret by V for x_i. We stress that in different sessions V uses independent random-tapes in his verification stage (that is, $\rho_1, \cdots, \rho_i$ $(1 \leq i \leq s(n))$ are independent random strings).

We then say a protocol satisfies *concurrent soundness* in our WPK 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 an i $(1 \leq i \leq s(n))$ such that $V(SK, x_i, \rho_i)$ outputs "accept x_i" while $x_i \notin L$ and x_i is not used in more than $U(n)$ sessions is negligible in n.

4 3-Round Resettable Zero-Knowledge Argument for $\mathcal{NP}$ with Concurrent Soundness in the WPK model

In this section, we present the main result of this paper: a 3-round resettable zero-knowledge argument for $\mathcal{NP}$ with concurrent soundness in the WPK model. As discussed before, the design of such a protocol is really desirable in practice. Three tools are crucial to our construction: verifiable pseudorandom functions [26], Dwork and Naor's 2-round rWI [11] and "complexity leveraging" [8].

4.1 Complexity leveraging

The "complexity leveraging" is used as follows. Let α be the pseudorandom constant of a VRF (that is, the output of `VRFEval` is indistinguishable from random for circuit of size 2^{k^α}, where k is the security parameter of the VRF). Let γ_1 be the following constant: for all sufficiently large n, the length of the $\mathcal{NP}$-witness y for $x \in L \cap \{0,1\}^n$ is upper-bounded by n^{γ_1}. Let γ_2 be the following constant: for all sufficiently large n, the length of the NIZK proof Π for an $\mathcal{NP}$-statement $x' \in L'$ of length $poly(n)$ is upper-bounded by n^{γ_2}. We then set $\gamma = max\{\gamma_1, \gamma_2\}$ and $\varepsilon > \gamma/\alpha$. We use a VRF with a *larger* security parameter $k = n^\varepsilon$. This ensures that one can enumerate all potential $\mathcal{NP}$-witnesses y, or all potential NIZK proofs for x', in time 2^{n^γ}, which still lesser than the time it would take to break the residual pseudorandomness of the VRF (because $2^{n^\gamma} < 2^{k^\alpha}$).

4.2 The VRF used

Let x be the common input of length n, and U be an a-priori bounded polynomial indicating the upper-bound of the corresponding counter kept by an honest verifier for x. That is x is allowed to be used at most $U(n)$ times by a malicious prover with the same honest verifier. We need a verifiable pseudorandom function with input length n and output length $2U(n) \cdot n^2$. We denote by $\mathtt{VRFEval}(VRFSK, x) = R_1^1 R_2^1 \cdots R_{2U(n)}^1 R_1^2 R_2^2 \cdots R_{2U(n)}^2 \cdots R_1^n R_2^n \cdots R_{2U(n)}^n$ the output of VRF on input x of length n, where for each i $(1 \leq i \leq n)$ and each j $(1 \leq j \leq 2U(n))$, the length of R_j^i is n.

4.3 Key generation of *V*

Under a system security parameter n, each verifier with identity id, V_{id} , generates a key pair $(VRFSK, VRFPK)_{id}$ for a VRF with security parameter k. V_{id} then uniformly selects $p(n)$ random strings $(R_{V_1}, R_{V_2}, \cdots, R_{V_{p(n)}})_{id}$ to be used as the first message of Dwork and Naor's 2-round rWI, where p is a positive polynomial. $VRFSK_{id}$ is V_{id}'s secret key and $VRFPK_{id}$ along with the $p(n)$ random strings, $(R_{V_1}, R_{V_2}, \cdots, R_{V_{p(n)}})_{id}$, is its public key. We remark that in comparison with the key generation stage of Micali and Reyzin's protocol [25], the key generation stage of our protocol is greatly simplified.

4.4 The full protocol

Common input An element $x \in L \cap \{0,1\}^n$. Denote by R_L the corresponding $\mathcal{NP}$-relation for L.

System Security parameter n. (Note that the security parameter of the VRF is k that is much larger than n).

Public file A collection F of records (id, PK_{id}), where $PK_{id} = (VRFPK_{id}, (R_{V_1}, R_{V_2}, \cdots, R_{V_{p(n)}})_{id})$.

P **private input** An $\mathcal{NP}$-witness y for $x \in L$; V's id and the file F; and a random string w that determines a PRF f_w.

V **private input** A secret key $SK_{id} = VRFSK_{id}$.

P**-step-one**

1. Using the PRF f_w, P generates R_P and $(s_1^1, s_2^1, \cdots, s_{2U(n)}^1, s_1^2, s_2^2, \cdots, s_{2U(n)}^2, \cdots, s_1^n, s_2^n, \cdots, s_{2U(n)}^n)$ from the inputs x, y, and PK_{id}. R_P will be served as the first part of the second message of Dwork and Naor's 2-round rWI and the other $2U(n) \cdot n$ pseudorandom strings will be served as the randomnesses used in the one-round perfect binding commitment scheme defined in Definition 1.
2. Selects $2U(n) \cdot n$ arbitrary strings of length $2U(n) \cdot n^2$ each: $(t_1^1, t_2^1, \cdots, t_{2U(n)}^1, t_1^2, t_2^2, \cdots, t_{2U(n)}^2, \cdots, t_1^n, t_2^n, \cdots, t_{2U(n)}^n)$. Let $Com = \{c^{(i,j)} = C_{s_j^i}(t_j^i), 1 \le i \le n$ and $1 \le j \le 2U(n)\}$, where C is the one-round perfect binding commitment scheme defined in Definition 1.
3. P sends (R_P, Com) to V.

V**-step one**

1. Computes $VR_x = \texttt{VRFEval}(SK_{id}, x) = R_1^1 R_2^1 \cdots R_{2U(n)}^1 R_1^2 R_2^2 \cdots R_{2U(n)}^2 \cdots R_1^n R_2^n \cdots R_{2U(n)}^n$, and $pf_x = \texttt{VRFProve}(SK_{id}, x)$. Note that $SK_{id} = VRFSK$. We call each R_j^i, $1 \le i \le n$ and $1 \le j \le 2U(n)$, a *block* with respect to the pair (x, id).
2. Randomly selects $(j_1, j_2, \cdots, j_n)$, where j_i, $1 \le i \le n$, is uniformly distributed over $\{1, 2, \cdots, 2U(n)\}$. For each i, $1 \le i \le n$, computes $VR_{R_{j_i}^i} = \texttt{VRFEval}(SK_{id}, R_{j_i}^i)$ and $pf_{R_{j_i}^i} = \texttt{VRFProve}(SK_{id}, R_{j_i}^i)$.
3. V sends $(VR_x, pf_x, (j_1, j_2, \cdots, j_n), (VR_{R_{j_1}^1}, VR_{R_{j_2}^2}, \cdots, VR_{R_{j_n}^n}), (pf_{R_{j_1}^1}, pf_{R_{j_2}^2}, \cdots, pf_{R_{j_n}^n}))$ to the prover P.

P-step-two

1. Verifies that VR_x is correct by invoking $\texttt{VRFVer}(VRFPK, x, VR_x, pf_x)$. If not, aborts.
2. For each i, $1 \le i \le n$, verifies that $VR_{R_{j_i}^i}$ is correct by invoking $\texttt{VRFVer}(VRFPK, R_{j_i}^i, VR_{R_{j_i}^i}, pf_{R_{j_i}^i})$. If not, aborts.
3. Constructs another $\mathcal{NP}$-statement: x'="there exists an $\mathcal{NP}$-witness y such that $(x, y) \in R_L$ *OR* for each i, $1 \le i \le n$, there exists a $j \in \{1, 2, \cdots, 2U(n)\}$ and a string s_j^i such that $c^{i,j} = C_{s_j^i}(VR_{R_{j_i}^i})$".

4. As does in the second round of Dwork and Naor's 2-round rWI, on the statement x' while using y as the witness P generates and sends to V $p(n)$ NIZK proofs $\{NIZK(x', R_P \oplus R_{V_i}), 1 \leq i \leq p(n)\}$. The randomness used by P is got by applying his PRF f_w on the transcript so far. In the rest of this paper, we denote by $\{NIZK(x', R_P \oplus R_{V_i}), 1 \leq i \leq p(n)\}$ a $p(n)$-NIZK-proof sequence.

Verifier's decision If the $p(n)$ NIZK proofs above are all acceptable then accepts, otherwise, rejects.

Theorem 1. *Assuming the security of RSA with large exponents against subexponential-time adversaries, the above protocol is a 3-round black-box resettable zero-knowledge argument with concurrent soundness for $\mathcal{NP}$ in the WPK model.*

Proof. (sketch)

The completeness and the optimal round-complexity of our protocol can be easily checked. In the rest we focus on proofs of black-box resettable zero-knowledge and concurrent soundness.

(1). Black-box resettable zero-knowledge

The rZK property can be shown in a way similar to (and simpler than) the way shown in [8].

Specifically, for any (s,t)-resetting malicious verifier V^*, suppose the outputs of the first stage of V^* are: $s(n)$ distinct strings $x_1, x_2, \cdots, x_{s(n)} \in L$ of length n each, the public file F and a list of $s(n)$ identities $id_1, id_2, \cdots, id_{s(n)}$. Intuitively, if for each *block*, R^i_j ($1 \leq i \leq n$ and $1 \leq j \leq 2U(n)$), with respect to (x_k, id_t), $1 \leq k, t \leq s(n)$, the simulator can learn the output of `VRFEval` on R^i_j before his commitments in P-step-one then it is easy for the simulator to generate a transcript that is computationally indistinguishable from the real interactions between P and V^*. That is, the simulator simulates the P-step-one by just setting $t^i_j = \texttt{VRFEval}(VRFSK, R^i_j)$, $1 \leq i \leq n$ and $1 \leq j \leq 2U(n)$. Since for an (s,t)-resetting verifier V^*, there are at most $s(n)^2 \cdot 2U(n) \cdot n$ blocks in total, the simulator works as follows to generate a simulated transcript while oracle accessing to V^*.

The simulator works in $s(n)^2 \cdot 2U(n) \cdot n + 1$ phases. Each phase corresponds to an attempt to simulate the real interactions between P and V^* and so each phase may consist of multiple sessions. In each phase the simulator uses an independent random-tape to try to simulate the real interactions between P and V^* except that at the current session V^* invokes P on the same x and PK_{id} that has been used in a previous session. In this case, the simulator simulates the P-step-one of current session by just copy the P-step-one messages sent in the previous session. In each phase, suppose V^* invokes P on x and PK_{id} at the current session then the simulator simulates the P-step-one of the current session by committing to the outputs of *VRFEval* on the *blocks* with respect to (x, id) he has learnt previously, together with committing to some garbage values if he has not yet leant the outputs of *VRFEval* on all the blocks with respect to the pair (x, id). We remark that at any point in the simulation if the simulator detects cheating (e. g. the V-step-one messages do not pass through the *VRFVer* test correctly) then the simulator aborts the simulation and outputs the transcript so far. It is easy to see that in each phase if V^* does not select a new block in this phase then the simulator succeeds in generating a simulated transcript that is indistinguishable from the real interactions between P and V^* due to the pseudorandomness of the PRF used and the computational hiding of the commitment scheme and the witness indistinguishability property of the underlying Dwork and Naor's 2-round rWI. Otherwise, the simulator will learn the outputs of `VRFEval` on at least one new block and in this case the simulator goes to the next phase. Here we have ignored the probability that a malicious verifier may give different outputs of `VRFEval` on the same block. But according to the unique probability of the VRF this probability is indeed exponentially small.

We stress that in each phase of above simulation the simulator does not rewind V^* and so he can proceed in strict polynomial-time in each phase. Also note that the total number of phases is also a polynomial. It means that the black-box simulator works in strict polynomial time rather than expected polynomial time. We remark that this result does not hold for black-box zero-knowledge in the standard model. Indeed, expected polynomial time is necessary for black-box zero-knowledge simulation in the standard model [6] and the first *non-black-box* zero-knowledge argument for $\mathcal{NP}$ with strict polynomial time simulation was presented in [1].

(2). Concurrent soundness

We first note that a computational power unbounded prover can easily convince the verifier of a false statement since he can get the $VRFSK$ if his computational power is unbounded. Hence the above protocol constitutes an argument system rather than a proof system.

To deal with the soundness of the above protocol in the WPK model we stress that we should be very careful since our argument system works in a somewhat "parallel repetition" manner to reduce the error probability. The reason is that Bellare et al. have proven that for a 3-round argument system if the verifier has secret information regarding historical transcripts then parallel repetition does not guarantee to reduce the error probability [5]. Note, however, that in our argument protocol the verifier indeed has secret information, the SK.

The following proof uses a standard reduction technique. That is, if the above protocol does not satisfy concurrent soundness in the WPK model then we will construct a machine $T = (T_E, T_J)$ to break the residual pseudorandomness of the VRF.

Suppose the above protocol does not satisfy concurrent soundness in the WPK model then in a concurrent attack issued by an (s,t)-concurrent malicious prover P^* against an honest verifier with identity id, V_{id}, with non-negligible probability there exists an i, $1 \leq i \leq s(n)$, such that V_{id} outputs "accept x_i" while $x_i \notin L$ and x_i has not been used by P^* in more than $U(n)$ sessions. Now, T_E first guesses this "i" and then simulates the *concurrent* multiple interactions between P^* and V_{id} while running P^*. Note that in his simulation T_E does not need to rewind P^* since he has oracle access to both $\texttt{VRFEval}(VRFSK, \cdot)$ and $\texttt{VRFProve}(VRFSK, \cdot)$ and the j_i, $1 \le i \le n$, in V-step-one is uniformly distributed over $\{1, 2, \cdots, 2U(n)\}$. So, T_E can simulate the multiple concurrent interactions between P^* and V_{id}. When it is the time to simulate the i-th session T_E first determines whether $x_i \in L$ or not by just enumerating all the $\mathcal{NP}$-witnesses of x_i. Note that with non-negligible probability this is the case that $x_i \notin L$ since T_E can correctly guess the i with non-negligible probability. If $x_i \notin L$ then T_E runs P^* to get the P-step-one messages from P^*. Then T_E uniformly selects $(j_1, j_2, \cdots, j_n)$ from $\{1, 2, \cdots, 2U(n)\}$ and computes n blocks $(R^1_{j_1}, R^2_{j_2}, \cdots, R^n_{j_n})$ with respect to (x_i, id) just as V_{id} does in V-step-one. Since x_i has been used at most $U(n)$ times and for each i, $1 \le i \le n$, j_i is uniformly distributed over $\{1, 2, \cdots, 2U(n)\}$, then with probability at least $1-2^{-n}$ T_E will select a new block from all the $2U(n) \cdot n$ blocks with respect to the pair (x_i, id), on which the output of $\texttt{VRFEval}$ is unknown to P^* up to now. Denote by $R^k_{j_k}$, $1 \le k \le n$, the new block selected. T_E then outputs $(R^k_{j_k}, state)$, where *state* is the historical view of T_E.

Now, T_J receives v and T_J's job is to find whether v is a truly random value or $\texttt{VRFEval}(VRFSK, R^k_{j_k})$. For this purpose T_J first constructs the new $\mathcal{NP}$-statement x' (defined in P-step-two) with respect to $(VR_{R^1_{j_1}}, VR_{R^2_{j_2}}, \cdots, VR_{R^{k-1}_{j_{k-1}}}, v, VR_{R^{k+1}_{j_{k+1}}}, \cdots, VR_{R^n_{j_n}})$. The key observation is that if v is a truly random value then most likely there are no $p(n)$-*NIZK-proof sequences* in which the $p(n)$ NIZK proofs are all acceptable on the statement x' since $x_i \notin L$ and the commitment scheme used by P^* is perfect binding and v is completely unpredictable for P^*. However, if $v = \texttt{VRFEval}(VRFSK, R^k_{j_k})$, then (according to our assumption) with non-negligible probability there exists a $p(n)$-*NIZK-proof sequence* in which the $p(n)$ NIZK proofs are all acceptable on the statement x'. Note that T_J can enumerate all the NIZK proofs for x' in time $p(n) \cdot 2^{n^\gamma}$. Then T_J checks that if there exists a $p(n)$-*NIZK-proof sequence* in which the $p(n)$ NIZK proofs are all acceptable. If find such a sequence then T_J decides that $v = \texttt{VRFEval}(VRFSK, R^k_{j_k})$, otherwise, T_J decides that v is a truly random value. Note that $p(n) \cdot 2^{n^\gamma} < 2^{n^\alpha}$ which violates the residual pseudorandomness of the VRF. $\square$

Acknowledgment. The first author is indebted to Leonid Reyzin for his enlightening discussions and numerous kindly clarifications. The first author is full of gratitude to Boaz Barak, Cynthia Dowrk, Yehuda Lindell and Oded Goldreich for their kindly clarifications. Moti Yung referred the first author to [23] for recent advances of VRF. We are grateful to anonymous referees for their many valuable comments. This research is supported by a research grant of City University of Hong Kong (No. 7001232) and a research grant of Natural Science Funds of China (No. 60003007).

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. M. Blum, A. D. Santis, S. Micali and G. Persiano. Non-interactive Zero-Knowledge. *SIAM Journal on Computing*, 20(6): 1084-1118, 1991.
3. M. Bellare, M. Fischlin, S. Goldwasser and S. Micali. Identification protocols secure against reset attacks. In *B. Pfitzmann (Ed.): Advances in Cryptology-Proceedings of EUROCRYPT 2001, LNCS 2045*, pages 495–511. Springer-Verlag, 2001.
4. M. Blum, P. Feldman and S. Micali. Non-interactive Zero-Knowledge and Its Applications. In *ACM Symposium on Theory of Computing*, pages 103-112, 1988.
5. M. Bellare, R. Impagliazzo and M. Naor. Does Parallel Repetition Lower the Error in Computationally Sound Protocols. In *IEEE Symposium on Foundations of Computer Science*, pages 374-383, 1997.
6. B. Barak and Y. Lindell. Strict Polynomial-Time in Simulation and Extraction. In *ACM Symposium on Theory of Computing*, pages 484-493, 2002.
7. R. Cramer and I. Damgard. Linear Zero-knowledge: A Note on Efficient Zero-Knowledge Proofs and Arguments. In *ACM Symposium on Theory of Computing*, pages 436-445, 1997.
8. R. Canetti, O. Goldreich, S. Goldwasser and S. Micali. Resettable Zero-Knowledge. In *ACM Symposium on Theory of Computing*, pages 235-244, 2000.
9. R. Canetti, J. Kilian, E. Petrank and A. Rosen. Black-Box Concurrent Zero-Knowledge Requires $\tilde{\Omega}$ Rounds. In *ACM Symposium on Theory of Computing*, pages 570-579, 2001.
10. 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.
11. C. Dwork and M. Naor. Zaps and Their Applications. In *IEEE Symposium on Foundations of Computer Science*, pages 283-293, 2000.
12. C. Dwork, M. Naor and A. Sahai. Concurrent Zero-Knowledge. In *ACM Symposium on Theory of Computing*, pages 409-418, 1998.
13. C. Dwork and L. Stockmeyer. 2-Round Zero-Knowledge and Proof Auditors. In *ACM Symposium on Theory of Computing*, pages 322-331, 2002.
14. U. Feige, A. Fiat and A. Shamir. Zero-knowledge Proof of Identity. *Journal of Cryptology*, 1(2): 77-94, 1988.
15. 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.
16. 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.
17. O. Goldreich. *Foundation of Cryptography-Basic Tools*. Cambridge University Press, 2001.
18. S. Goldwasser and M. Bellare. *Lecture Notes on Cryptography*. 2001.
19. O. Goldreich, S. Goldwasser and S. Micali. How to Construct Random Functions. *Journal of the Association for Computing Machinery*, 33(4):792–807, 1986.
20. O. Goldreich and H. Krawczky. On the Composition of Zero-Knowledge Proof Systems. *SIAM Journal on Computing*, 25(1): 169-192, 1996.
21. 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.

22. 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.
23. A. Lysyanskaya. Unique Signatures and Verifiable Random Functions from the DH-DDH Separation. In *M. Yung (Ed.): Advances in Cryptology-Proceedings of CRYPTO 2002, LNCS 2442* , pages 597-612. Springer-Verlag, 2002.
24. 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.
25. 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.
26. S. Micali, M. Rabin and S. Vadhan. Verifiable Random Functions. In *IEEE Symposium on Foundations of Computer Science*, pages 120-130, 1999.
27. L. Reyzin. *Zero-Knowledge with Public Keys*. Ph. D Thesis, MIT, 2001.
28. 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.