APPEARED IN ICALP (2) 2008: 548-559

Constant-Round Concurrent Non-Malleable Zero Knowledge in the Bare Public-Key Model

Rafail Ostrovsky* Giuseppe Persiano† Ivan Visconti‡

Abstract

One of the central questions in Cryptography is the design of round-efficient protocols that are secure under concurrent man-in-the-middle attacks. In this paper we present the first *constant-round concurrent non-malleable zero-knowledge* argument system for NP in the Bare Public-Key model [Canetti et al. STOC 2000], resolving one of the major open problems in this area. To achieve our result, we introduce and study the notion of non-malleable witness indistinguishability, which is of independent interest. Previous results either achieved relaxed forms of concurrency/security or needed stronger setup assumptions or required a non-constant round complexity.

Keywords: non-malleable zero knowledge, witness indistinguishability.

1 Introduction

Interactive proof systems play a central role in cryptography. Starting with the seminal paper of Goldwasser, Micali and Rackoff [1], the notion of zero knowledge and the simulation paradigm have been adopted in order to prove security of interactive proof systems. Dolev, Dwork and Naor [2] proposed the notion of a non-malleable zero-knowledge (NMZK, in short) proof systems where security must be preserved even under a man-in-the-middle attack. This stronger attack allows the adversary to act as a prover in a proof and as a verifier in another proof with full control over the scheduling of the messages. The notion of NMZK is proved to be extremely important in cryptography, since it captures the notion of *proof independence*, and led to multiple applications. Feasibility results for NMZK have been shown by using either black-box techniques and a super-constant number of rounds by Dolev et al. [2] or by using non-black-box techniques and obtaining computational soundness in a constant number of rounds by Barak [3] and Pass and

*Department of Computer Science and Department of Mathematics, UCLA, Los Angeles, CA, Email: rafail@cs.ucla.edu. Supported in part by IBM Faculty Award, Xerox Innovation Group Award, NSF grants 0430254, 0716835, 0716389 and U.C. MICRO grant.

†Dipartimento di Informatica ed Applicazioni, Università di Salerno 84084 Fisciano (SA), ITALY. Email: giuper@dia.unisa.it

‡Dipartimento di Informatica ed Applicazioni, Università di Salerno 84084 Fisciano (SA), ITALY. Email: visconti@dia.unisa.it

Rosen [4]. Another stronger variation of zero knowledge is *concurrent* zero knowledge, introduced by Dwork, Naor and Sahai [5], where security has to work against adversaries that are involved in many concurrent executions of a proof system.

In this paper we consider an adversary $\mathcal{A}$ mounting a *concurrent* man-in-the-middle attack in which $\mathcal{A}$ acts as a verifier interacting with a honest prover in polynomially many *left* proofs and acts as a prover interacting with honest verifiers in polynomially many *right* proofs. The problem of designing protocols that combine concurrent security with security against man-in-the-middle adversaries has received a lot of attention; several questions still remain open, though. In particular, constant-round concurrent non-malleable zero-knowledge (cNMZK, for short) proof systems have been shown to exist by assuming the existence of trusted third parties or a trusted common reference string [6, 7] or by using relaxed security notions [8] or relaxed concurrency [9]. A construction with poly-logarithmic round complexity for concurrent NMZK in the plain model has been given by Barak, Prabhakaran, and Sahai [10]. The possibility of constructing constant round cNMZK proof system in the plain model or under weaker setup assumptions is an open problem.

Non-malleable witness indistinguishability. A weaker but still useful security notion for proof systems is that of witness indistinguishability [11], where it is required that the adversarial verifier does not distinguish the witness used by the prover. Despite the tremendous applicability of witness indistinguishability, while a lot of attention has been given to zero knowledge with respect to man-in-the-middle attacks, very little attention has been given to witness indistinguishability in case of man-in-the-middle attacks.

We first show the definition and construction of a new concurrent non-malleable primitive that extends the notion of witness indistinguishability to the setting in which the adversary is a concurrent man-in-the-middle. For defining this new primitive, we focus on a specific class of argument systems referred to as *commit-and-prove*[1] functionality introduced in [12] and also considered in [6]. We then construct a *constant-round* concurrent non-malleable witness indistinguishable (cNMWI, for short) argument of knowledge (under Def. 2) for all $\mathtt{NP}$ in the plain model (see Theorem 4). This construction relies upon the work by Pass and Rosen [13] where constant-round concurrent non-malleable (NM, for short) commitments have been achieved. In a next work we also show that the notions of NMWI and NMZK argument systems are incomparable, this is surprising since all previously introduced notions of witness indistinguishability were implied by the corresponding notions of zero knowledge.

Non-malleable zero knowledge. We show the construction of a *a constant-round cNMZK argument system* under standard complexity theoretic assumptions and security notions in the Bare Public-Key model, a set-up assumption introduced in [14] that does not require any trusted third party. So far this has been achieved only under stronger setup assumptions. On the other hand, constant-round concurrent zero knowledge has been obtained in the BPK model in [14] (in [15]

[1] We restrict our study to this class of argument systems as: 1) they allow us to define the notion of witness encoded in a proof; 2) they suffice for our constructions and applications. It is possible however to generalize this notion.

with a concurrent soundness guarantee, and in [16, 17] under standard assumptions). Given our results, the BPK model is, at the best of our knowledge, the weakest model in which *constant-round* cNMZK has been achieved.

Corruption model and adaptive inputs. In all our results we consider the static corruption model where the adversary has to choose the corrupted parties before the protocols start. Following the previous work on NMZK, in the proof of our concurrent NMZK argument of knowledge in the BPK model we assume that the inputs (i.e., statements) for honest parties are fixed according to some predetermined distribution while the adversary can choose its inputs adaptively. Instead, for our cNMWI argument of knowledge in the plain model, following [18] we also allow the adversary to choose the inputs of the prover by giving it both the statements and the witnesses.

Work related to witness indistinguishability and cNMZK zero knowledge in the plain model. Recently and independently from our work Micali, Pass and Rosen [19] presented an extension of the notion of witness indistinguishability for achieving a relaxed notion of secure computation that does not resort to the simulation paradigm. Their techniques are similar to ours but in this work, in contrast to [19], we achieve arguments of knowledge and focus on the use of these strong notions of witness indistinguishability for achieving a notion of security based on simulation (i.e., concurrent NMZK). Moreover, achieving input-indistinguishability involves significantly more complicated protocols; furthermore, it is not clear how easy this notion is to work with when used as a "sub-protocol". The power of our simple and specific definition of non-malleable witness indistinguishability is that it can achieved essentially directly by relying on the non-malleable commitment protocol of [13] and it is easy to work with.

We observe that in the plain model constant-round (non-concurrent) NMZK has been recently obtained [3, 4] whereas obtaining constant-round concurrent zero knowledge in the plain model has been open for quite some time. The only constant-round concurrent zero-knowledge arguments known in the plain model impose a bound on the number of concurrent executions that the adversary can perform [20]. If we do not insist on constant-round protocols, non-malleability and security in a concurrent setting have been achieved by [10] which present a protocol with logarithmic round complexity.

2 Non-Malleable Witness Indistinguishability

For lack of space, the definition of standard tools and the ones about non-malleability can be found in the full version of this work [21, 22].

We now start by discussing and defining the new non-malleable notion of proof systems. In our definition of NM witness indistinguishability we shall require that the witness *encoded in the proof* given by the man-in-the-middle adversary $\mathcal{A}$ is independent from the witness used by the honest prover in the left proof. Notice that $\mathcal{A}$ might be unaware of the witness it has used in the right proof. More specifically, we focus on a specific class of argument systems referred to as *commit-and-prove* argument systems (previously considered in [6, 12]). Informally, the transcript

of a commit-and-prove argument encodes in an unambiguous way the witness used by the prover (even though it might not be efficiently extracted from the transcript). In a NMWI commit-and-prove argument we require the witness encoded in the proof produced by the man-in-the-middle adversary to be independent of the witness used (by the honest prover) in the proof in which the adversary acts as a verifier.

For general argument systems it is not clear whether the notion of witness encoded is well defined as there could be more than one. Therefore, we focus on commit-and-prove argument systems for which the notion of the witness encoded is well defined and commit-and-prove arguments actually suffice for proving our next result (i.e., cNMZK n the BPK model).

Commit-and-prove argument systems. A commit-and-prove argument system $\Pi = \langle P, V \rangle$ for a language L is a two-stage protocol. On input x, in the first stage the prover and the verifier execute a commitment protocol by which the prover commits to a string w. In the second stage, the prover proves to the verifier that the committed string w is a valid witness for "$x \in L$". We study commit-and-prove argument systems in which the commitment scheme used in the first stage is non-interactive and statistically binding, therefore the notion of *witness encoded in the proof* is well defined and it corresponds to the string committed to by the *first* prover-to-verifier message. If the proof is not accepted by the verifier, we consider the witness to be encoded in the proof to be the string $\perp$. We shall require that in a NMWI commit-and-prove argument system the man-in-the-middle adversary encodes in the right proof a witness that is independent from the one that the honest prover has used in the left proof.

Tag-based NMWI commit-and-prove arguments. We consider a man-in-the-middle adversary $\mathcal{A}$ interacting in the left proof with tag $\mathtt{tag}$ with the honest prover P that is running on input instance x and witness w. In the right proof, $\mathcal{A}$ is interacting with the honest verifier V on common input $\tilde{x}$ and tag $\tilde{\mathtt{tag}}$ of its choice. We denote by z the auxiliary information available to $\mathcal{A}$.

The notion of tag-based NM witness indistinguishability is defined in terms of the random variable $\mathsf{wmim}^{\mathcal{A}}(\mathtt{tag}, x, w, z)$ that is the distribution of the output of the following process: a transcript $\mathtt{trans}$ of an interaction of $\mathcal{A}$, including the left and the right proof, is picked according to distribution $\mathsf{View}^{P}_{\mathcal{A}}(\mathtt{tag}, x, w, z)$ (i.e., the view of $\mathcal{A}$ when running with z as auxiliary input and playing with P that runs on input (x, w) and tag $\mathtt{tag}$) and the output of a procedure wit applied to $\mathtt{trans}$ is returned. The procedure wit returns $\perp$ if the right proof is not accepting (i.e., V outputs 0) or $\mathtt{tag}$ is the tag of the right proof. Otherwise it returns the witness encoded in the right proof.

Definition 1 (tag-based NMWI argument) *A family of commit-and-prove argument systems $\Pi = \{\langle P_{\mathtt{tag}}, V_{\mathtt{tag}} \rangle\}_{\mathtt{tag}}$ for an $\mathtt{NP}$-language L is a* tag-based non-malleable witness indistinguishable (tag-based *NMWI*, in short) argument with tags of length ℓ *if, for all probabilistic polynomial-time man-in-the-middle adversaries $\mathcal{A}$, for all probabilistic polynomial-time algorithms D, there exists a negligible function ν such that for all $x \in L$, for all tags $\mathtt{tag} \in \{0,1\}^{\ell}$, for all pairs (w, w') of witnesses for x, and for all auxiliary information z it holds that*

$$|\text{Prob}[\ D(x, w, w', \mathsf{wmim}^{\mathcal{A}}(\mathtt{tag}, x, w, z), z) = 1\ ] -$$
$$\text{Prob}[\ D(x, w, w', \mathsf{wmim}^{\mathcal{A}}(\mathtt{tag}, x, w', z), z) = 1\ ]| < \nu(|x|).$$

A NMWI argument system is an argument of knowledge when for any prover that proves a given statement with probability p, there exists an efficient extractor that outputs a valid witness with essentially the same probability p (see the definition of [23]).

Comparison with NMZK. We stress here that NMZK requires the existence of a simulator while NM witness indistinguishability does not. Instead, NM witness indistinguishability crucially considers the possible witnesses that are encoded in the proofs given by the man-in-the-middle while NMZK requirements are satisfied when a valid witness is given in output by the simulator-extractor.

Comparison with NM commitments. The notion of NM witness indistinguishability is similar to the notion of NM commitment with respect to commitment [2, 4]. Indeed, both notions concern the security of a primitive against man-in-the-middle attacks by considering a string that is encoded in the messages sent by the adversary. This string is a committed message in case of NM commitments while it is an encoded witness in case of NM witness indistinguishability.

2.1 Concurrent and Simulation-Based NMWI Arguments

We extend the notion of non-malleable witness indistinguishability to the concurrent setting by considering a concurrent man-in-the-middle adversary $\mathcal{A}$ that opens $m = \mathsf{poly}(k)$ left and right proofs each with a common input of length $n = \mathsf{poly}(k)$. Here k refers to the security parameter. $\mathcal{A}$ interacts in the i-th left proof with an instance of the honest prover P on common input "$x_i \in L$" and private prover's input $w_i \in W(x_i)$. In the j-th right proof $\mathcal{A}$ is interacting with the honest verifier V on common input $\tilde{x}_j$ of its choice.

To define concurrent non-malleable witness indistinguishability, we extend $\mathsf{wmim}^{\mathcal{A}}(X, W, z)$ to sequences of inputs and witnesses in the following way. The distribution $\mathsf{wmim}^{\mathcal{A}}(X, W, z)$ is the distribution of the output of the following procedure. First a transcript $\mathtt{trans}$ is sampled according to the view $\mathsf{View}^{P}_{\mathcal{A}}(X, W, z)$ of $\mathcal{A}$. Then the output of the following extension of the procedure wit applied to $\mathtt{trans}$ is returned. Procedure wit returns a sequence $(\tilde{w}_1, \cdots, \tilde{w}_m)$ where m is the number of right proofs and it holds that: if the j-th right proof is non-accepting or has the same common input as one of the left proofs then $\tilde{w}_j = \perp$; otherwise, $\tilde{w}_j$ is the witness encoded in the j-th right proof.

As done for non-malleable witness indistinguishability, we can obtain a tag-based definition of concurrent non-malleable witness indistinguishability and we define $\mathsf{wmim}^{A}(T, X, W, z)$ so to take into account the tags and not the inputs of the right proofs. We stress again that $\mathcal{A}$ is allowed to choose the inputs and the tags for the right proofs.

Definition 2 (tag-based cNMWI argument) *A family of commit-and-prove argument systems* $\Pi = \{\langle P_{\mathtt{tag}}, V_{\mathtt{tag}}\rangle\}_{\mathtt{tag}}$ *for the language* L *is a* tag-based concurrent non-malleable witness indistinguishable argument *(a* tag-based cNMWI*) with tags of length* ℓ *if, for all probabilistic polynomial-time concurrent man-in-the-middle adversaries* $\mathcal{A}$*, for all* $m = \mathsf{poly}(k)$*, for all* $n = \mathsf{poly}(k)$ *and for all probabilistic polynomial-time algorithms* D*, there exists a negligible function* ν *such that for all* k*, for all sequences* X *of* m *elements of* L *of length* n*, for all sequences* T *of*

tags of length ℓ, for all sequences W and W' of witnesses for X, and for all auxiliary information z it holds that

$$|\text{Prob}[\ D(X, W, W', \mathsf{wmim}^{\mathcal{A}}(T, X, W, z), z) = 1\ ] -$$
$$\text{Prob}[\ D(X, W, W', \mathsf{wmim}^{\mathcal{A}}(T, X, W', z), z) = 1\ ]| < \nu(k).$$

We stress that the two above definitions can be adapted by requiring that each statement to be proved is adaptively chosen by the adversary (that will also provide valid witnesses to the provers) before the corresponding proof starts, as discussed in [24]. Our constructions will enjoy this extra property.

We will also consider a relaxed notion of concurrent non-malleable witness indistinguishability where the adversary is allowed to run only one left proof. We denote this restricted notion of concurrent NM witness indistinguishability as *one-left many-right* concurrent NM witness indistinguishability.

Simulation-based cNMWI Arguments. We also give a simulation-based definition of non-malleable witness indistinguishability. We consider only the tag-based case. Let $\mathcal{A}$ be a concurrent man-in-the-middle adversary and consider the following two executions. The first execution is the *man-in-the-middle* execution where the concurrent man-in-the-middle adversary $\mathcal{A}$ interacts with several copies of the honest prover in the left proofs and with several copies of the honest verifier in the right proofs. For this execution we define distribution $\mathsf{wmim}^{\mathcal{A}}(T, X, W, z)$ as done in the previous section. Also, we stress that $\mathcal{A}$ can choose the inputs for the right proofs as well as the tags. In the second execution, called the *stand-alone* execution, we consider a simulator S that, without receiving any witness for the inputs X of the left instances and without interacting with a honest prover, manages to output the transcripts of the left and the right proofs. We denote by $\mathsf{wsta}^{S}(T, X, z)$ the random variable that describes output of the following procedure. First a transcript $\texttt{trans}$ is sampled according to the distribution of the output of $S(T, X, z)$. Then the procedure wit is applied to $\texttt{trans}$ and the output is returned.

Definition 3 (tag-based SBcNMWI argument) *A family of commit-and-prove argument system $\Pi = \{\langle P_{\texttt{tag}}, V_{\texttt{tag}}\rangle\}_{\texttt{tag}}$ is a* tag-based simulation-based concurrent non-malleable witness indistinguishable *(*tag-based SBcNMWI*, in short) argument for the language L, if for all polynomials $m = \mathsf{poly}(k)$ and $n = \mathsf{poly}(k)$, for all probabilistic polynomial-time concurrent man-in-the-middle adversaries $\mathcal{A}$, there exists a simulator S running in expected polynomial time, such that the following distributions are computationally indistinguishable:*

$$\{\mathsf{wmim}^{\mathcal{A}}(T, X, W, z)\}_{T\in\{0,1\}^{ml}, X\in L_n^m, W\in W(X), z\in\{0,1\}^\star} \text{ and}$$
$$\{\mathsf{wsta}^{S}(T, X, z)\}_{T\in\{0,1\}^{ml}, X\in L_n^m, z\in\{0,1\}^\star}.$$

The notion of a simulation-based non-malleable witness indistinguishable commit-and-prove *argument of knowledge* can be obtained by further requiring that S is able to extract witnesses from the right proofs whenever they use tags different from the left proofs.

The notion of one-left many-right SBcNMWI argument can be obtained by restricting the adversary to be involved only in one left proof.

Theorem 4 *Assume that there exists a family of claw-free permutations. Then there exists a constant-round tag-based cNMWI commit-and-prove argument of knowledge for all* NP *in the plain model.*

The proof of this theorem is obtained by first noticing that a variation of the commitment scheme of [4] actually allows one to obtain a one-left many-right SBcNMWI argument of knowledge, then by noticing that any one-left many-right SBcNMWI argument of knowledge is a one-left many-right cNMWI argument of knowledge, and finally by noticing that any one-left many-right cNMWI argument of knowledge is a many-left many-right cNMWI argument of knowledge (see the full version of this work [21, 22] for the protocol and the security proof.)

We finally stress that the above theorem still holds in case the adversary chooses the inputs of the honest prover, by feeding it also valid witnesses.

3 cNMZK in the BPK Model

In the BPK model [14], each verifier registers some public information (called the *public key*) in a public file during a preprocessing stage. Each public key is associated with some secret information (called the *secret key*) that is known only to the owner of the public key. After the preprocessing is completed, parties engage in the proof stage where proofs are run.

We will define and construct in the BPK model constant-round arguments for any NP-language that are secure with respect to a BPK concurrent man-in-the-middle adversary $\mathcal{A}$ which during the preprocessing stage has complete control over the public file where keys are registered (that is, $\mathcal{A}$ can modify, omit and, add new adaptively chosen keys to the public file) and, once the preprocessing stage is completed, $\mathcal{A}$ acts as a concurrent man-in-the-middle adversary. We stress that no form of key-authentication is required thus making the BPK model a setting very close to the plain model.

The BPK model for interactive argument systems. We now review the definition of an interactive argument system in the BPK model that were previously given in [25] and the extension to the concurrent man-in-the-middle attack case.

Formally, a BPK *pair* is a pair $\langle P, V\rangle$ where P is a probabilistic polynomial-time algorithm and V is a pair $V = (V_0, V_1)$ of probabilistic polynomial-time algorithms. The interaction between provers and verifiers takes place in two stages. In the first stage, called the *set-up* stage, verifiers run algorithm V_0, on input a security parameter 1^k, to obtain a pair $(\mathsf{pk}, \mathsf{sk})$ consisting of a public and a secret key. Each verifier publishes his public key pk in a public file F. The second stage, called the *proof* stage, consists of polynomially (in the security parameter) many proofs. In each of them a prover interacts with a verifier; specifically, the prover runs algorithm P on input x (of length polynomial in the security parameter), some auxiliary information w (typically w is a witness for x to be member of some fixed language L) and the public key pk chosen by the verifier. The verifier instead runs algorithm V_1 on input x and sk.

Definition 5 *A* BPK *pair* $\langle P, V\rangle$ *is* complete *for the language* L *if in any interaction on common*

input $x \in L$ and pk *constructed by V_0, where P receives as additional input $w \in W(x)$, and V_1 secret key* sk *associated with* pk*, V_1 accepts except with negligibly probability.*

The definitions of argument systems in the BPK model can be found in [14], in particular in [25, 26] the notions of concurrent zero-knowledge and concurrent soundness have been defined. We will focus on cNMZK arguments of knowledge in the BPK model that implies both concurrent zero knowledge and concurrent soundness. Indeed, concurrent zero-knowledge corresponds to a special case where the man-in-the-middle does not run any right proof. Instead, concurrent soundness corresponds to the special case where the man-in-the-middle does not run any left proof and is implied by the fact that we require that a legal NP witness is obtained for any accepting proof given by the adversary (i.e. proofs where V outputs 1).

3.1 cNMZK in the BPK Model

We next define *cNMZK argument of knowledge* in the BPK model.

A BPK concurrent man-in-the-middle adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ is a pair of probabilistic algorithms. $\mathcal{A}_0$ on input an auxiliary information z receives the public file F containing the public keys as computed by the honest verifiers and outputs a modified public file F'. In computing F', $\mathcal{A}_0$ is allowed to add new adaptively chosen keys and to remove some of the keys of the honest verifiers. $\mathcal{A}_0$ also outputs some secret auxiliary information Z relative to F'. Once F' is made public by $\mathcal{A}_0$, it cannot be changed and the control passes to $\mathcal{A}_1$ that runs on input F' and Z. In the proof stage, $\mathcal{A}_1$ behaves like a concurrent man-in-the-middle adversary with the only restriction that he can start right proofs in which he plays as a prover with honest verifiers only with respect to entries of F' that were chosen by the honest verifiers and not modified by $\mathcal{A}_0$.

We define the view $\mathsf{BView}_{\mathcal{A}}(X, W, z)$ of a BPK concurrent man-in-the-middle adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ with respect to the vector X of left inputs with witnesses W as consisting of the initial public file received by $\mathcal{A}_0$, of all messages received by $\mathcal{A}_1$ in the proof stage both in the left proofs run on input X and right proofs run on inputs adaptively chosen by $\mathcal{A}_1$, along with the sequence of internal states of $\mathcal{A}_0$ and $\mathcal{A}_1$ and coin tosses, and the output of the honest verifiers.

Definition 6 *(cNMZK arguments of knowledge in the* BPK*) A* BPK *pair $\Pi = \langle P, V \rangle$ complete for the language L is a* BPK cNMZK argument of knowledge *if for every probabilistic polynomial-time* BPK *concurrent man-in-the-middle adversary $\mathcal{A}$, there exists a probabilistic algorithm S running in expected polynomial time such that, for all $m = \mathsf{poly}(k)$ and $n = \mathsf{poly}(k)$, by denoting with $S(X, z) = (S_0(X, z), S_1(X, z))$ the output of S on input (X, z), we have*

1. *$\{S_0(X, z)\}_{X \in L_n^m, z \in \{0,1\}^\star}$ and $\{\mathsf{BView}_{\mathcal{A}}(X, W, z)\}_{X \in L_n^m, W \in W(X), z \in \{0,1\}^\star}$ are computationally indistinguishable.*

2. *Writing the second component of S's output as $S_1(X, z) = (\tilde{w}_1, \ldots, \tilde{w}_m)$, we have that, for all accepting right proofs j of $S_0(X, z)$ with common input $\tilde{x}_j \notin X$, $\tilde{w}_j \in W(\tilde{x}_j)$ except with negligible probability.*

We stress that the adversary can always see the output of the verifier since this is what concretely happens when protocols are executed in the real world. This is an important issue for proof systems in which the internal state of the verifier is needed to decide whether a proof is accepted or not.

As a concurrent verifier and a concurrent prover are both special cases of a concurrent man-in-the-middle adversary, then it is obvious that a cNMZK argument of knowledge in the BPK model is both concurrent zero-knowledge and concurrently sound.

3.2 The Constant-Round Protocol

The main idea is to use the FLS paradigm by having the prover prove knowledge of either a legal witness of the input statement or of the secret key of the verifier. The goal is to design a simulator that runs the honest verifier algorithm and plays the role of the prover by first extracting the secret keys used by the adversary and then by using them as witness by running in a straight-line fashion the honest prover algorithm. In order to make this possible, we have the verifier first prove knowledge of his secret key so that the simulator will first extract the secret keys of the adversary. To withstand concurrent man-in-the-middle attack, we employ the cNMWI argument of knowledge we have developed in the previous section along with the two-key technique by [11].

More in details, in the preprocessing stage, each verifier computes a pair of public keys along with the corresponding secret keys. He then randomly chooses one of the two secret keys and discards the other one. This step can be implement by using a one-way function f in the following way: randomly pick two messages $\mathsf{sk}_0, \mathsf{sk}_1$ in the domain of f; compute public keys $\mathsf{pk}_0 = f(\mathsf{sk}_0), \mathsf{pk}_1 = f(\mathsf{sk}_1)$; randomly select $b \leftarrow \{0,1\}$; set $\mathsf{sk} = (b, \mathsf{sk}_b)$ and $\mathsf{pk} = (\mathsf{pk}_0, \mathsf{pk}_1)$.

The actual argument on input x consists of a sequential composition of two instances of the tag-based constant-round cNMWI commit-and-prove argument of knowledge we have constructed. First the verifier proves knowledge of one of the two secret keys associated to his entry in the public file (this is obviously done by NP-reducing this instance to the NP-complete language used by the subprotocol). This subprotocol is run using $x \circ 0$ as tag. Obviously the honest verifier uses his knowledge of one of two secret keys to successfully complete this subprotocol. In the second execution the prover proves knowledge of either w such that $R(x,w) = 1$ or of one of the two secret keys associated with the two public keys of the verifier. The tag used in this subprotocol is $x \circ 1$. Obviously the honest prover uses knowledge of a witness w for $R(x,\cdot)$ to complete the protocol.

Let us explain how we plan to perform simulation of the protocol. Simulation is easy for right proofs where the simulator plays the role of the honest verifier. Indeed right proofs are executed relatively to entry of the public file that have been constructed by the simulator itself and thus it knows one of the secret keys to perform the first subprotocol of a right proof. Simulating the second subprotocol of right proofs and the first subprotocol of the left proofs is trivial as the simulator can simply play the honest verifier algorithm of the subprotocol. In order to simulate the second subprotocol of left proofs instead the simulator needs to know either a witness for "$x \in L$" or one of the secret keys associated with the corresponding entries of the public file that are *used* by the adversary. However, the adversary has just proved knowledge of at least one of the two keys in the first subprotocol of the same proof. Therefore we plan on extracting one of these keys from

the adversary and then use it to perform the second subprotocol. The use of rewinds is dangerous in concurrent setting but not in the BPK model as shown in [14]. Indeed the number of extraction procedures that have to be successfully run is independent of the number of concurrent proofs, since it is bounded by the size of the public file. Once the simulator knows at least one secret key for each of the entries of the public file used by the adversary, the simulation is straight-line.

Let us now explain why we can also extract valid witness for all theorems proved by the adversary. We know that in all succeeding proofs for $x \in L$ given by the adversary, there is a cNMWI argument of knowledge for proving that $x \in L$ or that the adversary knows one of the two secret keys of the verifier. During the simulated game we can run the extractor for all these proofs in order to obtain the valid witnesses thus satisfying definition 6. If instead we extract as witnesses the secret keys of the verifier, we distinguish two cases. In the former case we extract a secret key that was not used by the simulator; we show how to reduce this case to an adversary that inverts the one-way function used for generating the public keys. In the latter case we always extract the same secret keys used by the simulator; this last case means that the adversary succeeded in encoding in the cNMWI arguments of knowledge that it proved, the same witness encoded by the simulator in the cNMWI arguments of knowledge where the adversary played as verifier. This last case contradicts the NM witness indistinguishability of the cNMWI arguments of knowledge.

The protocol in details. Let L be an NP-language with polynomial-time relation R and let f be a one-way function. Associated with L and f, we consider two auxiliary NP-languages L_1 and L_2 with polynomial-time relations R_1 and R_2 defined as follows:

- $(\mathsf{pk}_0, \mathsf{pk}_1) \in L_1$ iff there exist b and sk such that $\mathsf{pk}_b = f(\mathsf{sk})$;
- $(x, \mathsf{pk}_0, \mathsf{pk}_1) \in L_2$ iff $x \in L$ or $(\mathsf{pk}_0, \mathsf{pk}_1) \in L_1$.

In the description of our BPK cNMZK argument of knowledge (P, V) for any NP-language L we will use a tag-based cNMWI argument of knowledge $\Pi = \{\langle \mathcal{P}_{\mathtt{tag}}, \mathcal{V}_{\mathtt{tag}} \rangle\}_{\mathtt{tag}}$ for an NP-complete language Λ. When we say that we execute Π for proving that $\tau \in L_1$ (or $\sigma \in L_2$) we actually mean that τ (or σ) is reduced to an instance of Λ and $\mathcal{P}_{\mathtt{tag}}$ and $\mathcal{V}_{\mathtt{tag}}$ are executed on input this instance. We also remark that known reductions have the property that, if a witness for $\tau \in L_1$ (or for $\sigma \in L_2$) is known then a witness for the new instance can be constructed in polynomial time. (The protocol is formally described in Figure 1.)

Lemma 7 *If f is a one-way function and Π is a cNMWI argument of knowledge then the protocol $(\mathcal{P}, \mathcal{V})$ of Figure 1 is a cNMZK argument of knowledge in the* BPK *model for any* NP *language.*

For lack of space, the formal can be found in the full version of the paper available at [22, 21].

Theorem 8 *Assume that there exists a family of claw-free permutations. Then in the* BPK *model there exists a constant-round cNMZK argument of knowledge for all* NP.

The proof follows by Theorem 7, and by the observation that claw-free permutations imply the existence of one-way functions.

Input: security parameter 1^k.

PREPROCESSING STAGE:

Entry l of the public file is constructed by V_0 as follows:

pick $\mathsf{sk}_0^l, \mathsf{sk}_1^l \leftarrow \{0,1\}^k$, compute $\mathsf{pk}_0^l = f(\mathsf{sk}_0^l)$ and $\mathsf{pk}_1^l = f(\mathsf{sk}_1^l)$,

randomly pick $b^l \leftarrow \{0,1\}$, set $\mathsf{pk}^l = (\mathsf{pk}_0^l, \mathsf{pk}_1^l)$ and $\mathsf{sk}^l = (b_l, \mathsf{sk}_{b_l}^l)$.

output: $(\mathsf{pk}, \mathsf{sk})$.

PROOF STAGE:

Sub-protocol: tag-based cNMWI argument of knowledge $\Pi = \{\langle \mathcal{P}_{\mathtt{tag}}, \mathcal{V}_{\mathtt{tag}} \rangle\}_{\mathtt{tag}}$ for a NP-complete language Λ.

Common input: the public file F, entry $\mathsf{pk}^l = (\mathsf{pk}_0^l, \mathsf{pk}_1^l)$ of F, $n = \mathsf{poly}(k)$-bit string $x \in L$.

P**'s private input:** a witness w for $x \in L$.

V_1**'s private input:** secret key $\mathsf{sk}^l = (b_l, \mathsf{sk}_{b_l}^l)$;

$V_1 \longrightarrow P$**:** V_1 and P engage in an execution of Π with tag $x \circ 0$ where V_1 runs $\mathcal{P}_{x\circ 0}$ to prove to P (running $\mathcal{V}_{x\circ 0}$) knowledge of a witness (b_l, sk^l) for $\sigma = (\mathsf{pk}_0^l, \mathsf{pk}_1^l) \in L_1$.

$P \longrightarrow V_1$: P and V_1 engage in an execution of Π with tag $x \circ 1$ where P runs $\mathcal{P}_{x\circ 1}$ to prove to V_1 (running $\mathcal{V}_{x\circ 1}$) knowledge of a witness for $\tau = (x, \mathsf{pk}_0^l, \mathsf{pk}_1^l) \in L_2$.

Figure 1: The constant-round BPK cNMZK argument of knowledge $\langle P, V \rangle$ for NP.

References

[1] Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. **18** (1989) 186–208

[2] Dolev, D., Dwork, C., Naor, M.: Nonmalleable cryptography. **30** (2000) 391–437

[3] Barak, B.: Constant-round coin-tossing with a man in the middle or realizing the shared random string model. (2002) 345–355

[4] Pass, R., Rosen, A.: New and Improved Constructions of Non-Malleable Cryptographic Protocols. (2005) 533–542

[5] Dwork, C., Naor, M., Sahai, A.: Concurrent zero-knowledge. (1998) 409–418

[6] Canetti, R., Lindell, Y., Ostrovsky, R., Sahai, A.: Universally composable two-party and multi-party secure computation. (2002) 494–503

[7] Barak, B., Canetti, R., Nielsen, J.B., Pass, R.: Universally composable protocols with relaxed set-up assumptions. (2004) 186–195

[8] Barak, B., Sahai, A.: How to play almost any mental game over the net - concurrent composition via super-polynomial simulation. (2005) 543–552

[9] Kalai, Y.T., Lindell, Y., Prabhakaran, M.: Concurrent general composition of secure protocols in the timing model. (2005) 644–653

[10] Barak, B., Prabhakaran, M., Sahai, A.: Concurrent non-malleable zero knowledge. (2006)

[11] Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. (1990) 416–426

[12] Kilian, J.: Uses of randomness in Algorithms and Protocols. MIT Press, Cambridge, MA (1990)

[13] Pass, R., Rosen, A.: Concurrent non-malleable commitments. (2005) 563–572

[14] Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge. (2000) 235–244

[15] Di Crescenzo, G., Persiano, G., Visconti, I.: Constant-round resettable zero knowledge with concurrent soundness in the bare public-key model. (2004) 237–253

[16] Di Crescenzo, G., Visconti, I.: Concurrent zero knowledge in the public-key model. (2005) 816–827

[17] Visconti, I.: Efficient zero knowledge on the internet. (2006) 22–33

[18] Feige, U., Lapidot, D., Shamir, A.: Multiple NonInteractive Zero Knowledge Proofs under General Assumptions. SIAM Journal on Computing **29** (1999) 1–28

[19] Micali, S., Pass, R., Rosen, A.: Input-indistinguishable computation. (2006) 136–145

[20] Barak, B.: How to go beyond the black-box simulation barrier. (2001) 106–115

[21] Ostrovsky, R., Persiano, G., Visconti, I.: Constant-round concurrent nmwi and its relation to nmzk. Technical Report ECCC Report TR06-095, ECCC (2006)

[22] Ostrovsky, R., Persiano, G., Visconti, I.: Constant-round concurrent nmwi and its relation to nmzk. Technical Report 2006-256, Cryptology ePrint Archives (2006)

[23] Bellare, M., Goldreich, O.: On defining proofs of knowledge. (1992) 390–420

[24] Sahai, A.: Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. (1999) 543–553

[25] Micali, S., Reyzin, L.: Soundness in the public-key model. (2001) 542–565

[26] Reyzin, L.: Zero-Knowledge with Public Keys, Ph.D. Thesis. MIT Press, Cambridge, MA (2001)