Concurrent Non-Malleable Witness Indistinguishability and its Applications

Rafail Ostrovsky[*] Giuseppe Persiano[†] Ivan Visconti[‡]

Abstract

One of the central questions in Cryptography today is proving security of the protocols "on the Internet", i.e., in a concurrent setting where there are multiple interactions between players, and where the adversary can play so called "man-in-the-middle" attacks, forwarding and modifying messages between two or more unsuspecting players. Indeed, the main challenge in this setting is to provide security with respect to *adaptive concurrent composition* of protocols and also the *non-malleability* property, where the "man-in-the-middle" attacks are prevented. Despite much research effort, we do not know how to implement many basic tasks in this setting (which features concurrent composition and man-in-the-middle attacks). Indeed, even for tasks such as zero-knowledge proofs, which play an essential role in Cryptography, it is not known how to construct a protocol in a way that satisfies both security guarantees simultaneously.

In this paper, we consider a slightly weaker notion than zero-knowledge, namely *witness indistinguishability* of proofs, which never-the-less is an extremely important building block in Cryptography. Despite its importance, neither formulations nor constructions that satisfy both concurrent composition and resiliency against man-in-the-middle attacks were known.

The first contribution of this paper is to put forward the definition of *concurrent non-malleable* witness indistinguishability (in fact, we show two different definitions) and show a *constant-round* construction using non-black-box techniques. Furthermore, we show that this construction allow us to solve some important open problems.

More specifically, based on our construction of a constant-round input-adaptive concurrent non-malleable witness-indistinguishable argument of knowledge, we construct a *constant-round* input-adaptive concurrent non-malleable zero-knowledge argument of knowledge in the Bare Public-Key Model (the BPK model in short) that has been first proposed in [Canetti et al., STOC 2000]. The BPK model makes very minimal set-up assumptions, therefore our result improves the current state-of-the-art as previous results required either the existence of trusted third parties (trusted PKI, common reference string), or made physical assumptions (common reference string) or achieved only quasi security (simulation in super-polynomial time) or quasi concurrency (timing assumptions, bounded concurrency) or non-adaptiveness.

By plugging our results into known constructions, we achieve constant-round zero-knowledge and then $(n-1)$-secure multi-party computation under general concurrent composition in the BPK model.

[*] UCLA, USA. E-mail: rafail@cs.ucla.edu.

[†] Università di Salerno, Italy. E-mail: giuper@dia.unisa.it.

[‡] Università di Salerno, Italy. E-mail: visconti@dia.unisa.it.

1 Introduction

Interactive proof systems play a major role in the area of foundations of cryptography. Starting with the seminal paper of [GMR89], the notion of zero knowledge and the simulation paradigm have been adopted in order to prove security of interactive proof systems. Feasibility results show that zero-knowledge proof systems exist under the assumption that any one-way function exists [GMW91].

A related secure notion for interactive proof systems is that of witness indistinguishability introduced in [FS90]. This notion is weaker than zero knowledge as it only requires that the adversarial verifier does not distinguish the witness used by the prover among the set of all possible witnesses. The impact of this new notion has revolutionized the next research on zero-knowledge proof systems, as for non-interactive zero knowledge [FLS99, DDO$^+$01] non-black-box zero knowledge [Bar01] and concurrent zero knowledge [RK99, KP01].

Since its introduction, the basic (stand-alone) notion of a zero-knowledge proof systems has been showed to be inadequate to guarantee security with respect to stronger attacks mounted by malicious players. In [DDN91], Dolev et al. proposed the notion of a non-malleable zero knowledge proof system where security must be preserved even in case the adversary can play the role of a man-in-the-middle. This stronger attack allows the adversary to act as a prover in a session and as a verifier in another session with full control over the scheduling of the messages and with the capability of aborting sessions. Feasibility results have been showed by using either black-box techniques and a super-constant number of rounds in [DDN00] or by using non-black-box techniques and a constant number of rounds in [Bar02, PR05b] and requiring only computational soundness. A related notion is that of simulation soundness introduced in [Sah99], where the man-in-the-middle receives *simulated proofs* and tries to prove a false statement. These notions have been used for constructing protocols for bounded-concurrent secure two party and multi-party computation [Lin03, PR03, Pas04] and for designing CCA2 secure public-key encryption schemes [DDN00, Sah99, DDO$^+$01].

Motivated by challenging scenarios as the Internet, the possibility of concurrently running any polynomial number of sessions has been first considered in [DNS98] in the context of zero knowledge thus introducing the notion of *concurrent zero knowledge*. It is known that zero knowledge is not closed under concurrent composition [KPR98] and this contrasts with the notion of witness indistinguishability since, as showed in [FS90, FLS99], any witness indistinguishable proof system with respect to non-uniform adversaries is also concurrent witness indistinguishable. This strong closure property of witness indistinguishability has been crucially used for achieving concurrent zero knowledge in [RK99, KP01] using black-box techniques and a logarithmic number of rounds. The existence of a *constant-round* concurrent zero-knowledge argument system for $\mathcal{NP}$ (which would require non-black-box techniques in light of an impossibility result proved in [CKPR01]) is a major open problem.

Unfortunately, the feasibility results disappear when we consider the strong (but actually more realistic) case of concurrent non-malleable zero knowledge, where the man-in-the-middle can open polynomially many sessions playing as verifier and polynomially many sessions playing as prover.

If one restricts to the *plain* model for interactive proofs [GMR89] where no extra set-up assumption is made, no construction has been given so far for concurrent non-malleable zero knowledge. Moreover, in [Lin03, Lin04], Lindell showed that *input-adaptive* concurrent non-malleable zero-knowledge arguments of knowledge, where the adversary can adaptively choose the statements to be proved in the sessions, is impossible for $\mathcal{NP}$-complete languages.

The challenging task of designing protocols that are secure in this strong and realistic setting and the lack of feasibility results in the plain model, motivated the introduction of several set-up assumptions and security definitions. So far, concurrent non-malleable zero knowledge (or even stronger notions) has been achieved by assuming the existence of trusted third parties [DDO$^+$01, CLOS02, BCNP04] (i.e., the existence of a common reference string or of a certified public-key infrastructure), by weakening the security requirement [PS04, BS05] (i.e., simulation in super-polynomial time), by making assumptions on the underlying network [KLP05], and by bounding the ability of the adversary of opening concurrent sessions [PR05b]. Input-non-adaptive concurrent non-malleable zero knowledge in the plain model with a super-constant number of rounds has been recently (and independently from our work) achieved by [BPS06]. Seen the current state of knowledge, achieving concurrent non-malleable zero knowledge with a constant number of rounds and with respect to input-adaptive adversaries is an important open problem.

Recently, several construction have been shown to be secure in the Bare Public-Key model (the BPK model, for short) introduce by Canetti et al. [CGGM00]. This model makes very minimal set-up and network assumptions. More specifically, the BPK model requires that verifiers register their public keys in a public file during a set-up stage. This stage does not require any interaction among the players. Moreover, no trusted third party is assumed, nor physical assumptions are made, and the underlying communication network is assumed to be (adversarially) asynchronous. The BPK model only assumes that no prover-verifier interaction starts during the set-up stage.

Constant-round concurrent zero-knowledge (actually, resettable zero-knowledge, a stronger notion from [CGGM00]) protocols in the BPK model were presented in [CGGM00, MR01] and the verifier's security (that is, the soundness of the argument) is only guaranteed with respect to provers that are active in only one session at time. In [DPV04] a constant-round concurrently sound (that is, the argument remains sound even if the prover concurrently opens several sessions with the verifier) concurrent zero-knowledge argument system in the BPK model is presented under non-standard assumptions on the hardness of computational problems against sub-exponential-time adversaries. The use of such non-standard assumptions is referred to as "complexity leveraging" and is very related to the notion of super-polynomial-time simulation used in [Pas03] that corresponds to a relaxed notion of security that we refer to as *quasi-security* . Recently, concurrent zero-knowledge and soundness in the BPK model have been achieved in [CV05] with polynomial-time assumptions. Since concurrent non-malleability has been achieved with quasi-concurrency in [KLP05] and with quasi-security in [BS05] after obtaining the simpler notion of concurrent zero-knowledge in the same models ([DNS98] and [Pas03]), a very challenging open problem is the design of concurrent non-malleable zero knowledge arguments of knowledge in the BPK model. This would improve either the set-up assumption or the network assumption or the security of all previous constructions.

Adaptive corruption and adaptive inputs selection. In general, an adversary can be adaptive in the following two senses.

- **Adaptive corruption**: the adversary can dynamically choose the corrupted parties, thus *during* the execution of the protocols he can select the parties that will then run under is control. This contrasts with static corruption where the adversary has to choose the corrupted parties of a protocol before the protocol starts. In the context of concurrent non-malleable proof systems, adaptive corruption means that during the executions of the protocols the adversary can choose for any running proof to get the control of either the prover or the verifier, possibly choosing the prover in some proofs and the verifier in other proofs. Instead, static corruption in concurrent non-malleable proof systems means that before a given proof starts, the adversary has to choose the corrupted party, i.e., the prover or the verifier, possibly choosing the prover in some proofs and the verifier in other proofs.
- **Adaptive input selection**: the adversary can dynamically choose the inputs of the parties, thus *during* the execution of the protocols he can adaptively choose the inputs to be used in new protocols. This contrasts with the non-adaptive case where all inputs are established in advance, before protocols start. In the context of concurrent non-malleable proof systems, adaptive input selection means that during the execution of the protocols, the adversary can adaptively choose the common inputs for new proofs. Instead, non-adaptive input selection in concurrent non-malleable proof systems means that the adversary has to choose all common inputs in advance, before any protocol starts.

Notice that the impossibility result by Lindell [Lin03, Lin04] concerns concurrent non-malleable zero knowledge with respect to *adaptive input selection.* Consequently, the setting with adaptive input selection is much more challenging than the corresponding non-adaptive one. In all our results, we will consider static corruption and in the main results we will achieve adaptive input selection. Therefore, for the sake of simplifying the notion, we will use the word *input-adaptive*, to refer to adaptive input selection.

1.1 Our Results

In this paper we introduce the notion of *non-malleable witness indistinguishability in the plain model.* In particular, we focus on a specific class of argument systems referred to as *commit-and-prove* argument systems, introduced in [Kil90] and also considered in [CLOS02]. Informally, the transcript of a *commit-and-prove* argument system encodes, typically through a commitment, the witness used by the prover. We consider an adversary A mounting a *concurrent* man-in-the-middle attack (an extension of the attack introduced by [DDN91]). In a concurrent man-in-the-middle attack A acts as a verifier interacting with a honest prover in polynomially many *left* interactions and acts as a prover interacting with honest verifiers in polynomially many *right* interactions. Our notion of non-malleable witness indistinguishability requires the *witnesses* encoded in the right arguments (in which A acts as a prover) to be independent from the witnesses used (by honest provers) in the left arguments.

We present two different definitions of non-malleable witness indistinguishability. The first definition (see Def. 4.1) which we call the basic one extends the original definitions for stand-alone witness indistinguishability of [FS90] and requires that the distribution of the witnesses (and not simply the views as in [FS90]) encoded in the right arguments is independent of the witnesses used by the honest provers in left interactions. This requirement forces us to consider only computationally indistinguishable distributions.

The second definition (see Def. 4.2), which we call *simulation-based* witness indistinguishability, requires that, for each successful man-in-the-middle adversary A, there exists a stand-alone prover S (that is, S does not have access to the honest prover) that has access to A and succeeds in proving to a honest verifier the same statements proved by A. Moreover, the arguments proved by S to the honest verifier encode the same witnesses encoded in the proofs given by A. This second notion of non-malleable witness indistinguishability is actually stronger than non-malleable zero knowledge (NMZK, in short). Indeed, the stand-alone prover S' of NMZK is only guaranteed to prove the same statements proved by A to a honest verifier. However, there is no guarantee that the distribution of the witnesses encoded in the successful arguments of S' is computationally indistinguishable from the distribution of the witnesses encoded in the proofs given by A.

We will consider static corruption but adaptive input selection, thus A can adaptively choose its statements. Consequently, the statements proved in the right sessions by A are not necessarily the same statements proved by A when interacting with the simulator. However, as in [PR05b] the distributions of the statements proved in the two cases and of the witnesses encoded in the proofs are computationally indistinguishable.

Constructions in the plain model. We give the following results in the plain model: 1) we construct *constant-round* one-left many-right simulation-based concurrent non-malleable witness indistinguishable (cNMWI, for short) argument systems for all $\mathcal{NP}$ (see Theorem 4.4); this construction relies upon the recent work by Pass and Rosen [PR05a]; 2) we prove that any one-left many-right simulation-based cNMWI argument system is actually a many-left many-right concurrent non-malleable with respect to the basic definition (see Theorem 4.5 and Theorem 4.7).

We will formally state and prove the following theorem.

Theorem 1.1 (informal) *Under the assumption that a family of collision resistant hash functions exists, in the plain model there exists a constant-round input-adaptive concurrent non-malleable witness indistinguishable argument of knowledge for all languages in $\mathcal{NP}$.*

Constructions in the bare public-key model. The main construction we give in this paper works in the Bare Public Key model (`BPK` model, in short) introduced by [CGGM00]. Indeed, based on our construction of cNMWI arguments of knowledge in the plain model, we construct a *constant-round* concurrent non-malleable zero-knowledge argument of knowledge for all $\mathcal{NP}$ in the `BPK` model (see Section 5). The `BPK` model is, at the best of our knowledge, the weakest model in which concurrent non-malleability has been achieved. Previous results required either the existence of trusted third parties (trusted PKI [BCNP04], common reference string [DDO$^+$01, CLOS02]), or achieved only quasi security (simulation in super-polynomial time [PS04, BS05]) or quasi concurrency (timing assumptions [KLP05], bounded concurrency [Pas04]). Moreover, our results clarify the importance of the newly-introduced notion of non-malleable witness indistinguishability that can have other important applications in the design of secure protocols.

We stress that concurrent non-malleable zero knowledge [DDO$^+$01] and universally composable zero knowledge [CLOS02] have been achieved in the common reference string model. In this model a randomly chosen string (called the *reference* string) is available to both the prover and verifier. There are several ways of actually implementing this model each of which has some drawback. One possibility would be to postulate the existence of a trusted party that selects the reference string at random and then disappears. Alternatively, one might think of obtaining the reference string as the output of some natural process which is equivalent to making physical assumptions. A different approach has been taken by [Bar02] and consists in constructing the reference string by means of a coin tossing protocol. Here we stress though that, in order to obtain a constant-round concurrent non-malleable zero knowledge, we need to design a constant-round concurrently non-malleable coin-tossing protocol. On the other hand the BPK model does not make any of the assumptions above. Indeed, no trusted party is needed to manage the repository of public keys which needs not to be authenticated and can be in total control of an adversary. The only (rather mild) assumption made by the BPK model consists in requiring that the registration and proof stages do not interleave.

Adaptive input selection and general concurrent composition. We show that all our constructions are secure even if the adversary is allowed to adaptively choose the inputs. In particular, this holds for our construction of concurrent non-malleable zero-knowledge arguments of knowledge in the BPK model. We stress that concurrent non-malleable zero-knowledge with adaptively chosen inputs is not possible in the plain model (and thus a set-up assumption is necessary).

We will formally state and prove the following theorem.

Theorem 1.2 (informal) *Under the assumption that a family of collision resistant hash functions exists, in the* BPK *model there exists a constant-round input-adaptive concurrent non-malleable zero-knowledge argument of knowledge for all languages in $\mathcal{NP}$.*

Then since our construction is secure with respect to input-adaptive adversaries, we can use the compiler by Lindell [Lin04, Lin03] to obtain a constant-round protocol for realizing the zero-knowledge argument of knowledge functionality under general concurrent composition in the BPK model.

Finally, we use the compiler from zero-knowledge to multi-party computation [GMW87, BMR90, Rog91, CLOS02, GL02] to obtain constant-round $(n-1)$-secure multi-party computation under general concurrent composition in the BPK model. We stress that we consider static corruption, assume the existence of authenticated channels and do not consider fairness issues.

Comparison with the protocol of [BPS06]. Input-non-adaptive concurrent non-malleable zero knowledge in the plain model with a super-constant number of rounds has been recently (and independently from our work) achieved by [BPS06]. We stress that even though we use a set-up assumption (i.e., the BPK model), we achieve a better round complexity and, moreover, in our case security is preserved against adversaries that adaptively choose the inputs. Obtaining such a stronger result in the plain model is proved to be impossible for $\mathcal{NP}$-complete languages by Lindell [Lin04].

2 Definitions

We will consider $\mathcal{NP}$-languages L and denote by R_L the corresponding poly-time relation R_L such that $x \in L$ if and only if there exists w such that $R_L(x, w) = 1$. Also a *negligible* function $\nu(n)$ is a function such that for any constant $c < 0$ and for all sufficiently large n, $\nu(n) \leq n^c$.

We assume that the reader is familiar with the notion of an *interactive Turing machine* (see [GMR89]) and of *computationally indistinguishable* and *statistically close* ensemble of random variables.

Interactive Argument/Proof Systems. An *interactive proof (resp., argument) system* [GMR89] for a language L is a pair of interactive Turing machines $\langle P, V\rangle$, satisfying the requirements of *completeness* and *soundness*. Informally, completeness requires that for any $x \in L$, at the end of the interaction between P and V, V rejects with negligible probability. Soundness requires that for any $x \notin L$, for any computationally unbounded (resp., probabilistic polynomial-time) $P^\star$, at the end of the interaction between $P^\star$ and V, V accepts with negligible probability. We denote by $\langle P, V\rangle(x)$ the output of the verifier V when interacting on common input x with prover P. Also, sometimes we will use the notation $\langle P(w), V\rangle(x)$ to stress that prover P receives as additional input witness w for $x \in L$.

Formally, we have the following definition.

Definition 2.1 *A pair of interactive Turing machines $\langle P, V\rangle$ is an* interactive proof system *for the language L, if V is probabilistic polynomial-time and there exists a negligible function $\nu(\cdot)$ such that*

1. Completeness: *For every $x \in L$ and valid witness w,*

$$\texttt{Prob}[\langle P(w), V\rangle(x) = 1] \geq 1 - \nu(|x|).$$

2. Soundness*: For every $x \notin L$ and for every $P^\star$*

$$\texttt{Prob}[\langle P^\star, V\rangle(x) = 1] \leq \nu(|x|).$$

If the soundness condition holds only with respect to probabilistic polynomial time $P^\star$ then $\langle P, V\rangle$ is called an argument*.*

Since all constructions we give are actually argument (rather than proof) systems, we will now focus on argument systems only. Also from now on we assume that all interactive Turing machines are probabilistic polynomial-time.

Arguments of knowledge. Informally, an argument system is an argument of knowledge if for any prover that convinces with non-negligible probability a honest verifier, there exists an expected polynomial-time algorithm, called extractor, that has access to the prover and outputs a witness for the statement proved by the prover. Formally, we have the following definition.

Definition 2.2 *An argument system $\langle P, V\rangle$ for the $\mathcal{NP}$-language L and* knowledge error *function $\delta : \{0,1\}^\star \rightarrow [0,1]$ is an* argument system of knowledge *if for any interactive Turing machine $P^\star$ there exists a constant c and an algorithm E such that for any $x \in L$, if $\texttt{Prob}[\langle P^\star, V\rangle(x) = 1] = \texttt{POLY}(x) > \delta(x)$, then $\texttt{Prob}[w \leftarrow E^{P^\star(x)}(x) : R_L(x,w) = 1] = 1$ and the expected running time of algorithm E is bounded by $\frac{|x|^c}{\texttt{POLY}(x)-\delta(x)}$.*

Zero Knowledge. The classical notion of a zero-knowledge argument/proof system has been introduced in [GMR89]. In a zero-knowledge proof system a prover can prove to a verifier the validity of a statement without releasing any additional information. This concept is formalized by requiring the existence of an expected polynomial-time algorithm, referred to as *Simulator*, whose output is indistinguishable from the view of the verifier.

We start by defining the concept of a view of an interactive Turing machine. Let A and B be two interactive Turing machines that run on common input x and assume that A and B have additional information z_A and z_B. We denote by $\texttt{View}^A_B(x, z_A, z_B)$ the random variable describing the *view of* B; that is, B's random coin tosses and the sequence of messages received by B.

We are now ready to present the notion of a zero-knowledge argument.

Definition 2.3 *Let L be an $\mathcal{NP}$-language. An interactive argument $\langle P, V\rangle$ for L is* zero-knowledge *if for all $z \in \{0,1\}^\star$ and for all polynomial-time verifiers $V^\star$, there exists an expected algorithm S running in expected polynomial-time such that for all (x,w) such that $R_L(x,w) = 1$ it holds that $\texttt{View}^P_{V^\star}(x,w,z)$ and $S(x,z)$ are indistinguishable.*

If the two probability distributions are statistically *close then $\langle P, V\rangle$ is a* statistical zero-knowledge argument *system.*

Witness Indistinguishability. The notion of witness-indistinguishability applies to interactive proof/argument systems for $\mathcal{NP}$ languages and requires that no information is revealed to malicious verifiers about which witness is being used during the execution of the argument.

Definition 2.4 *Let L be an $\mathcal{NP}$-language. An interactive argument $\langle P, V\rangle$ for L is* witness indistinguishable *if for every pair (w_1, w_2) such that $R_L(x, w_1) = R_L(x, w_2) = 1$, and all polynomial-time verifiers $V^\star$ running on input any auxiliary information z (that can include w_1 and w_2) the probability distributions $\texttt{View}^P_{V^\star}(x, w_1, \cdot)$ and $\texttt{View}^P_{V^\star}(x, w_2, \cdot)$ are indistinguishable.*

If the two probability distributions are statistically *close then $\langle P, V\rangle$ is* statistical witness indistinguishable argument*.*

Commitment Schemes. A commitment scheme can be seen as the digital equivalent of a sealed envelope. If a party A, called the committer, wants to commit to some message m she just puts it into the sealed envelope, so that whenever A wants to reveal the message, she opens the envelope. This primitive must meet some basic requirements. First of all the digital envelope should *hide* the message: no party other than A should be able to learn m from the commitment (this is often referred in the literature as the hiding property). Second, the digital envelope should be *binding* in the sense that once the commitment has been produced A cannot change his mind about m.

Here we only focus on non-interactive commitment schemes where there is one message from the sender to the receiver to commit to a message and another message from the sender to the receiver to open the previous commitment. More specifically, a commitment scheme is a pair of probabilistic polynomial-time algorithms: the commitment algorithm C and the decommitment algorithm D. The committer obtains a pair (c, d) of commitment and decommitment keys by running C on input the message m. The commitment c is published by the committer. It is useful to think of c as the sealed envelop containing the message m. To reveal the commitment, the committer publishes the triple (c, d, m). The decommitter algorithm D then is run on input the triple to verify that the commitment c was properly opened as m.

The hiding property requires that the commitment c does not reveal any information on the committed message m to an adversary that has access to c. In a *statistically hiding commitment scheme* this property holds regardless of the computational power of the adversary.

The binding property requires that an adversary can not produce a commitment c for which there exists two messages m_0 and m_1 and two decommitment keys (d_0, d_1) such that c can be opened as m_0 using d_0 and as m_1 using d_1 (that is, $D(c, d_0, m_0) = 1$ and $D(c, d_1, m_1) = 1$). In a *statistically binding commitment scheme* this property holds regardless of the computational power of the adversary.

Here we give the formal definition of non-interactive statistically binding commitment scheme.

Definition 2.5 $(\mathtt{Com}, \mathtt{Ver})$ *is a non-interactive statistically binding* commitment scheme *if:*

- **efficiency:** $\mathtt{Com}$ *and* $\mathtt{Ver}$ *are probabilistic polynomial-time algorithms;*
- **completeness:** *for all m it holds that*

$$\mathtt{Prob}\left((\mathtt{com}, \mathtt{dec}) \leftarrow \mathtt{Com}(m) : \mathtt{Ver}(\mathtt{com}, \mathtt{dec}, m) = 1\right) = 1;$$

- **binding:** *for any algorithm* $\mathtt{sender}^*$ *there is a negligible function ν such that*

$$\begin{aligned}\mathtt{Prob}\,(&(\mathtt{com}, m_0, m_1, \mathtt{dec}_0, \mathtt{dec}_1) \leftarrow \mathtt{sender}(1^n) \wedge m_0 \neq m_1 \\ &\mathtt{Ver}(\mathtt{com}, \mathtt{dec}_0, m_0) = \mathtt{Ver}(\mathtt{com}, \mathtt{dec}_1, m_1) = 1) \leq \nu(n);\end{aligned}$$

- **hiding:** *for all m_0, m_1 where $|m_0| = |m_1|$ the probability distributions:*

$$\{(\mathtt{com}_0, \mathtt{dec}_0) \leftarrow \mathtt{Com}(m_0) : \mathtt{com}_0\} \quad \textit{and} \quad \{(\mathtt{com}_1, \mathtt{dec}_1) \leftarrow \mathtt{Com}(m_1) : \mathtt{com}_1\}$$

are computationally indistinguishable.

3 The MIM Setting

We now review the man-in-the-middle setting in which we give our new definitions and constructions.

Man-In-the-Middle (MIM) Attacks. The notion of non-malleability has been first consider in [DDN91]. Non-malleability is concerned with an adversary $\mathcal{A}$ that mount a so-called *man-in-the-middle attack* on two concurrently executed sessions (called the left and the right session) of a protocol Π. Even though non-malleability can be considered with respect to any protocol, in this paper we will consider it mainly in relation to argument systems. Let $\Pi = \langle P, V \rangle$ be an argument system for the language L. In a man-in-the-middle attack on Π, the adversary $\mathcal{A}$ acts as a verifier in the left session (and thus $\mathcal{A}$ verifies the validity of a statement $x_L \in L$ being proved by P) and as a prover in the right session (trying to convince V of the validity of a statement $x_R \in L$). It is assumed that $\mathcal{A}$ has complete control of the communication channel and therefore decides the scheduling of the messages. Very informally, Π is non-malleable if the interaction in the left session does not help A in the right session. In a more powerful man-in-the-middle attack, the adversary $\mathcal{A}$ is not restricted to one session on the left and one session on the right but instead $\mathcal{A}$ is allowed to concurrently play polynomially many left and right sessions. An argument Π is said *concurrent non-malleable* if the left sessions do not help $\mathcal{A}$ in the right sessions. Let us now proceed more formally.

Concurrent Non-Malleable Zero Knowledge. The notion of concurrent non-malleable zero knowledge is formalized by considering two scenarios: the MIM scenario and the Ideal scenario.

In the MIM scenario, adversary $\mathcal{A}$ mounts the concurrent man-in-the-middle attack.

We denote by $\{\mathtt{view}^{P,\mathcal{A}}_{\mathcal{A},V}(x_1, \ldots, x_{\mathtt{POLY}(n)})\}_{x_1,\ldots,x_{\mathtt{POLY}(n)}}$ the distribution of the output of V after a concurrent man-in-the-middle attack of $\mathcal{A}$.

In the Ideal scenario, we consider an ideal adversary S that has access to $\mathcal{A}$ and produces the same attack to prove statements $(x'_1, \ldots, x'_{\mathtt{POLY}(n)})$ to V.

We denote by $\{S^{\mathcal{A}}_V(x_1, \ldots, x_{\mathtt{POLY}(n)})\}_{x_1,\ldots,x_{\mathtt{POLY}(n)}}$ the random variable of the output of V when interacting with S. We stress that S does not interact with P. Therefore, the requirement that the outputs of V in the two scenarios to be indistinguishable, captures the idea that the proofs played by $\mathcal{A}$ with P did not help.

Definition 3.1 *Let $\Pi = \langle P, V \rangle$ be an argument system for a language L. We say that Π is a* concurrent non-malleable zero-knowledge *argument if, for any adversary $\mathcal{A}$, there exists an expected polynomial-time algorithm $S_{\mathcal{A}}$ such that the ensembles*

$$\{\mathtt{view}^{P,\mathcal{A}}_{\mathcal{A},V}(x_1, \ldots, x_{\mathtt{POLY}(n)})\}_{x_1,\ldots,x_{\mathtt{POLY}(n)}} \quad \mathtt{and}$$

$$\{S^{\mathcal{A}}_V(x_1, \ldots, x_{\mathtt{POLY}(n)})\}_{x_1,\ldots,x_{\mathtt{POLY}(n)}}$$

are computationally indistinguishable.

Notice that if we consider only one left and one right session we obtain the notion of (stand-alone) non-malleable zero knowledge.

Non-malleability in the extraction sense. A stronger notion of non-malleable zero-knowledge, is obtained by requiring that a non-malleability machine M on input statements $(x_1, \ldots, x_l)$ and access to any MIM adversary A, outputs statements $(x'_1, \ldots, x'_{l'})$ along with corresponding valid witnesses such that the distribution of these statements is computationally indistinguishable from that of the statements successfully proved by A in the MIM scenario when P proves statements $(x_1, \ldots, x_l)$. This notion was introduced by [DDO+01] for NIZK and clearly implies non-malleability.

4 Non-Malleable Witness Indistinguishability

The notion of a witness indistinguishable proof system was introduced in [FS90] and required that the *view* of the (adversarial) verifier when interacting with a prover must be independent of the witness used by the prover among all valid witnesses for the statement. This notion therefore concerns $\mathcal{NP}$ statements such that there exist more than one valid witness.

The standard notion of WI is defined even if the verifier is non-uniform and has access to the witnesses. Using hybrid arguments, it follows that witness indistinguishability is preserved under concurrent self composition [FS90]. This notion does not seem to capture any interesting property in a man-in-the-middle setting. Indeed, even though an adversarial man-in-the-middle A can not distinguish the witness w used by the prover, he could succeed in proving theorems as a prover that uses a witness w' that is related to w. Moreover, the proof given by A is still witness indistinguishable and therefore a succeeding attack can not be detected by only looking at the transcripts of the interactions and at the coin tosses of the honest parties. Therefore witness indistinguishability does not seem to be closed under these man-in-the-middle attacks, i.e., witness indistinguishability does not seem to imply non-malleable witness indistinguishability.

On top of this intuitive failure we study witness indistinguishability with respect to man-in-the-middle attacks and give, to the best of our knowledge, the first definition of non-malleable witness indistinguishability. In particular we focus on a specific class of argument systems referred to as commit-and-prove argument systems, previously considered in [CLOS02, Kil90]. Informally, the transcript of an instance of this class of argument systems encodes the witness used by the prover even though it can not be efficiently detected by the adversary. Our notion of non-malleable witness indistinguishability requires that the witnesses encoded in the proofs computed by the adversary have to be independent of the witnesses used (by honest provers) in arguments in which the adversary acts as a verifier.

4.1 Two Definitions of NMWI

A commit-and-prove argument system $\Pi_R = \langle P, V \rangle$ is an argument system parameterized by a relation R that is composed by two stages and a common input x of length n. In the first stage the verifier receives a commitment $\mathtt{com}$ to a message w from the prover. In the second stage the prover proves to the verifier that $R(x, w) = 1$. This notion is a generalization of a commitment scheme since R can be the identity relation and x can be set equal to w. We focus on the implementation of this gadget in which the commitment scheme used in the first stage enjoys the statistically-binding property. We refer to the witness encoded in the proof as the message committed in the first stage[1]. We denote by $n \in N$ the security parameter.

We deal with an adversary A that is capable of mounting a concurrent man-in-the-middle attack; i.e., A can open any polynomial number m of sessions choosing to play for each of them either as a prover or as a verifier. We assume that A has complete control over the scheduling of the messages in the concurrent sessions and can abort any session in any step of the protocol. For the sessions in which A plays as a verifier (these are called the *left* sessions), A interacts with a prover on input $X_L = (x_1, \ldots, x_l)$ and we denote by $W_L = (w_1, \ldots, w_l)$ the witnesses encoded in the proofs given by the prover. Moreover, A takes an auxiliary input ω that may contain an a-priori information on W_L. Instead, for the sessions in which A plays as a prover (these are the *right* sessions), A will produce proofs for inputs $Y_R = (y_1, \ldots, y_r)$ where each element of Y_R is not in X_L. With each such proof we associate a witness (that is the message committed in the first round of the protocol). We denote by $\mathtt{mim}^A_{X_L}(W_L, \omega)$ the random variable that describes the witnesses $Z_R = (z_1, \ldots, z_r)$ encoded in the proofs for Y_R given by $\mathcal{A}$ in the right interactions with ω as auxiliary information.

In this section we give two definitions for the concept of a *concurrent non-malleable witness indistinguishable commit-and-prove* argument system. In the definitions we only address the input-non-adaptive case where all common inputs are known in advance. The other two-cases that we will discuss later are the input-semi-adaptive case in which the adversary is allowed to adaptively choose the statement he proves (i.e., Y_R), and the input-adaptive case where the adversary adaptively decides all statements, thus providing correct witnesses to the provers for X_L.

We now give the definition of a *concurrent non-malleable witness indistinguishable* commit-and-prove argument system. We focus on the computational case since it is the only notion that we use in the remaining part of the paper.

Definition 4.1 *For any security parameter n and any $l = \mathtt{POLY}(n)$, a commit-and-prove argument system $\Pi = \langle P, V \rangle$ for an $\mathcal{NP}$-language L is* Concurrent Non-Malleable Witness Indistinguishable *(in short, cNMWI) if, for all PPT A, and for all auxiliary information ω it holds that $\{\mathtt{mim}^A_{X_L}(W_L, \omega)\}$ and $\{\mathtt{mim}^A_{X_L}(W'_L, \omega)\}$ are computationally indistinguishable, where the ensembles are taken over all l-tuples X_L of n-bit elements of L and all pairs W_L, W'_L of sequences of valid witnesses for X_L.*

Informally this notion says that in case of a concurrent man-in-the-middle attack, the witnesses encoded in the proofs given by the adversary in the right sessions do not *depend* on the witnesses encoded by the prover in the left sessions. Here the independence that we consider is about which witnesses are chosen among all possible witnesses. More concretely, for $i = 1, \ldots l$, the choice of a prover in using a given witness w_i instead of w'_i in the i-th proof does not affect the distribution of the witnesses (i.e., which witness is encoded in each proof among the set of possible witnesses) encoded in the proofs given in the right sessions by the adversary.

We also consider the restricted notion of concurrent non-malleability referred to as one-left many-right (resp., many-left one-right) concurrent non-malleability where the adversary is

[1] When the proof system used in the second stage is sound with respect to the attack of the adversary, then the witness encoded in the proof corresponds to an $\mathcal{NP}$-witness for x.

allowed to run only one session as verifier (resp., as prover) while it is still unrestricted on the number of right (resp., left) sessions.

Moreover, we say that A is 1) input-adaptive if it is allowed to adaptively choose all the statements to be proved, thus providing the correct witnesses to the honest provers; 2) input-semi-adaptive if it is allowed to adaptively choose only the statements he proves; 3) input-non-adaptive, otherwise.

The notion we have introduced and defined has similarities with the notions of concurrent non-malleable commitments where the messages encoded in the commitments computed by the $\mathtt{mim}$ have to be independent of the ones encoded in the commitments computed by the honest sender.

A simulation-based definition. Following the traditional simulation paradigm, we can give a different definition for cNMWI commit-and-prove arguments that considers a simulator that has access to A (and simulates proofs for X_L) and outputs proofs for Y_R. We require that the distribution of the statements and the witnesses encoded in the proofs of the adversary in the right interactions when interacting on the left on input X_L with provers using witnesses W_L is the same as the one produced by S with access to A. We denote by $\mathtt{sis}^S_{X_L}(\omega)$ the random variable that describes the witnesses $Z_R = (z_1, \ldots, z_{m_r})$ encoded in the proofs given by S in the right interactions on input ω.

Definition 4.2 *A commit-and-prove argument system* $\Pi = \langle P, V \rangle$ *is* simulation-based concurrent non-malleable witness indistinguishable *(in short, SBcNMWI) for the language L if for every PPT MIM A for any auxiliary input ω there exists an expected polynomial-time simulator S such that the ensembles $\{\mathtt{mim}^A_{X_L}(W_L, \omega)\}$ and $\{\mathtt{sis}^S_{X_L}(\omega)\}$ are computationally indistinguishable, where the ensembles are taken over all l-tuple X_L of elements of L and all l-tuple W_L of valid witnesses for X_L.*

Tag-based argument systems. The previous definitions can be extended by labeling each proof with a tag. In this case, we require that the indistinguishability between $\mathtt{mim}^A_{X_L}$ and $\mathtt{sis}^S_{X_L}$ holds only for the right sessions that are labeled with tags never used in the left sessions. Note that, when the tag is the common input, this prevents the relying attacks where the $\mathtt{mim}$ simply forwards messages from left to right and vice versa.

4.2 Constructions in the Plain Model

The main tool that with almost cosmetic variations allows us to give constructions for non malleable witness-indistinguishability in the plain model is a recent result by Pass and Rosen [PR05b, PR05a].

Theorem 4.3 (Pass and Rosen [PR05b, PR05a]) *Under the assumption that a family of collision resistant hash functions exists, there exists a constant-round input-semi-adaptive tag-based one-left many-right concurrent non-malleable statistical zero-knowledge argument of knowledge for any $\mathcal{NP}$-language L.*

This result says that for any one-left many-right efficient $\mathtt{mim}$ A there exists an efficient simulator S that by accessing to A achieves the following results:

1. the view of both the left session and the right sessions given in output by S is statistically indistinguishable from the interaction of A with a honest prover and any polynomial number of honest verifiers;
2. the $\mathtt{mim}$ has to use in the right sessions, tags that are different from the one seen in the left session (all right sessions with a tag equal to the one used in the left session are ignored);
3. S also outputs all witnesses for the accepting proofs given by A, this means that the capability of any one-left many-right $\mathtt{mim}$ adversary A in proving a statement in a right session is owned by S (that is stand-alone) for a computationally indistinguishable statements.

The tool discussed above and constructed by Pass and Rosen in [PR05b, PR05a] actually is a tag-based simulation-based constant-round non-malleable witness-indistinguishable arguments of knowledge.

Theorem 4.4 *Under the assumption that a family of collision resistant hash functions exists, there exists a constant-round tag-based input-semi-adaptive one-left many-right SBcNMWI commit-and-prove argument of knowledge for any polynomial-time relation R.*

The construction and the proof follow those of non-malleable commitments of [PR05a].

PROOF. Let $\Pi_{\mathtt{tag}} = \langle P_{\mathtt{tag}}, V_{\mathtt{tag}}\rangle$ be the input-semi-adaptive tag-based constant-round one-left many-right concurrent non-malleable statistical zero-knowledge argument of knowledge of [PR05b, PR05a], $SS = (\mathtt{SG}, \mathtt{Sig}, \mathtt{SVer})$ be a one-time secure signature scheme of [Rom90] and $(\mathtt{Com}, \mathtt{Ver})$ be the non-interactive statistically binding commitment scheme of [Blu82]. All these primitives exist under the assumption that a family of collision-resistant hash functions exists. We now show the construction of a tag-based constant-round one-left many-right SBcNMWI commit-and-prove argument of knowledge $\Pi = \langle P, V\rangle$. The parties run the following steps on input x and tag t.

1. P sets $\mathtt{com} \leftarrow \mathtt{Com}(w, r)$ where w is his private input such that (i.e., $R(x, w) = 1$) and r is a random string.
   P sets $(\mathtt{sk}, \mathtt{pk}) \leftarrow \mathtt{SG}(1^n)$.
   P sends the pair $(\mathtt{com}, \mathtt{pk})$ to V.
2. P and V run protocol $\Pi_{\mathtt{pk}} = \langle P_{\mathtt{pk}}, V_{\mathtt{pk}}\rangle$ where the prover proves knowledge of w, r such that $\mathtt{com} = \mathtt{Com}(w, r)$ and $R(x, w) = 1$.
3. The prover computes a signature $\sigma \leftarrow \mathtt{Sig}(t \circ \mathtt{trans}, \mathtt{sk})$ where $\mathtt{trans}$ is the transcript exchanged so far and sends it to V.
4. The verifier accepts the proof iff $\mathtt{SVer}(t \circ \mathtt{trans}, \sigma, \mathtt{pk}) = 1$ and $V_{\mathtt{pk}}$ outputs 1.

A few observations.

1. This construction is similar to the construction of non-malleable commitments of [PR05a]. The only difference is that in [PR05a] the statement used in the underlying input-semi-adaptive tag-based one-left many-right concurrent non-malleable statistical zero-knowledge argument of knowledge is about knowledge of the decommitment while now we also check that the committed message satisfies R.

2. The simulator S runs exactly as the one of [PR05a]. Indeed the simulator does not use any *real* witness and therefore the extra check that the witness satisfies the relation R does not affect the simulation.

The proof follows the ones of [PR05b, PR05a] that we discuss as follows. The simulator required by the definition of SBcNMWI consists in committing to 0 and then invoking the simulator for $\Pi_{\mathtt{tag}}$.

1. Assume by contradiction the existence of a successful mim A and consider the arguments correctly proved by A. If A runs the subprotocol $\Pi_{\mathtt{tag}}$ in the right interactions (i.e., A uses the same tag used in the left session) then A easily allows us to break the signature scheme unless it copies all messages and use the same tag t for the proof (since the signature is computed on both t and the transcript). In this case, however, tag-based non-malleability is not violated by definition. Thus we assume from now on, that subprotocols $\Pi_{\mathtt{tag}'}$ run by A are such that $\mathtt{tag}' \neq \mathtt{tag}$.
   1. Consider the experiment exp_0 in which A interacts with the real prover.
   2. Consider the experiment exp_1 where a real witnesses is committed to but then the simulator for $\Pi_{\mathtt{tag}}$ is used in the left session (instead of the real prover) and in the right sessions instead of real verifiers; since this simulation is statistical indistinguishable (including both the transcripts of left sessions and the ones of the right sessions) then exp_0 and exp_1 are computationally (actually statistically) indistinguishable.
   3. Consider the experiment exp_2 where the commitment is a commitment to 0 and the simulator for $\Pi_{\mathtt{tag}}$ is used to produce the transcript (instead of the real prover and verifiers); since the commitment scheme enjoys the hiding property, exp_1 and exp_2 are computationally indistinguishable. Here, the argument of knowledge property is used to obtain the witnesses encoded in the proofs of the adversary A that allow one to reduce a distinguisher for the NMWI property to a distinguisher for the hiding property of the commitment scheme.
4. From the previous observations, the proof given by a mim A can be run by a stand-alone adversary S that has access to A, thus contradicting the claim that A is a successful mim.

□

The next theorem shows that the property of being SBcNMWI implies being cNMWI for the case of one-left many-right concurrency. The proof follows the one of [FS90] where witness indistinguishability is derived from zero knowledge.

Theorem 4.5 *Any input-semi-adaptive one-left many-right SBcNMWI commit-and-prove argument of knowledge is an input-semi-adaptive one-left many-right cNMWI commit-and-prove argument of knowledge.*

Proof. Let S be the simulator of the input-semi-adaptive one-left many-right SBcNMWI commit-and-prove argument of knowledge. Assume by contradiction that the claim does not hold. Therefore, there exists a successful mim A for the cNMWI commit-and-prove notion. Now fix two sequences W_L and W'_L of witnesses for X_L (here $|W_L| = |W'_L| = 1$ since we are considering the one-left case) and consider distributions $\mathtt{mim}^A_{X_L}(W_L, \omega)$ and $\mathtt{mim}^A_{X_L}(W'_L, \omega)$. Then each of them is computationally indistinguishable from $\mathtt{sis}^S_{X_L}(\omega)$, where S is the simulator associated with A (which exists since the argument is a input-semi-adaptive one-left many-right SBcNMWI commit-and-prove argument of knowledge). The existence of A therefore, allows one to distinguish either between $\mathtt{mim}^{A}_{X_L}(W_L, \omega)$ and $\mathtt{sis}^{S}_{X_L}(\omega)$ or between $\mathtt{mim}^{A}_{X_L}(W'_L, \omega)$ and $\mathtt{sis}^{S}_{X_L}(\omega)$ which contradicts the hypothesis that S is a simulator for the SBcNMWI notion. □

The next theorem shows that input-adaptive SBcNMWI arguments of knowledge do not exist for $\mathcal{NP}$-complete languages. The proof is derived from an impossibility result by Lindell [Lin03, Lin04].

Theorem 4.6 *Input-adaptive SBcNMWI commit-and-prove arguments of knowledge in the plain model do not exist for $\mathcal{NP}$-complete languages.*

Proof. Any input-adaptive concurrent SBcNMWI commit-and-prove argument of knowledge is also a concurrent non-malleable zero-knowledge arguments of knowledge. This follows by observing that the simulator for input-adaptive SBcNMWI commit-and-prove argument of knowledge is also a simulator for concurrent non-malleable zero knowledge. Indeed, assume by contradiction that there exists a succeeding mim adversary A for an input-adaptive concurrent non-malleable zero-knowledge argument of knowledge. This implies that with non-negligible probability, A succeeds in proving some distinguishable statements in a mim attack that instead no stand-alone simulator S for non-malleable zero knowledge with access to A is able to indistinguishably prove. Therefore there exists no stand-alone simulator S' with access to A for non-malleable WI since S' with access to A can not indistinguishably prove some distinguishable statements, otherwise we contradict the existence of a succeeding A. This shows that any input-adaptive concurrent SBcNMWI commit-and-prove argument of knowledge is also an input-adaptive concurrent NMZK argument of knowledge. By noticing that input-adaptive concurrent non-malleable zero-knowledge arguments of knowledge do not exist in the plain model [Lin03, Lin04], the claim holds. □

The next theorem follows the previous results of [FS90] that show that witness indistinguishability is preserved under concurrent composition. Here we consider the man-in-the-middle setting.

Theorem 4.7 *Under the assumption that a family of collision resistant hash functions exists, there exists an input-semi-adaptive cNMWI commit-and-prove arguments of knowledge in the plain model for all $\mathcal{NP}$ languages.*

Proof. We know via theorems 4.4 and 4.5, that input-semi-adaptive cNMWI commit-and-prove arguments of knowledge exist in the plain model if we only have one left interaction. We now prove that any input-semi-adaptive one-left many-right cNMWI commit-and-prove argument of knowledge is actually an input-semi-adaptive many-left many-right cNMWI commit-and-prove argument of knowledge.

Let Π be a one-left many-right cNMWI commit-and-prove argument of knowledge and assume by contradiction that there exists a successful concurrent mim adversary A. Consider input sequence X_L for the left sessions and two witness sequences for the left sessions W_L and W'_L

for which A is successful. That is, there exists a distinguisher D that can distinguish the distributions of witnesses encoded in the proofs given by A when interacting with a prover using W_L and when interacting with a prover using W'_L. Using standard hybrid arguments we can assume that W_L and W'_L only differ in one component which we call the *special* component and let w and w' be the witnesses of the special components of W_L and W'_L, respectively. We use A to construct a `mim` A' that contradicts the hypothesis that Π is an input-semi-adaptive one-left many-right cNMWI commit-and-prove argument of knowledge.

Essentially A' has as auxiliary information the witnesses of W_L and runs the prover's algorithm using these witnesses for all interactions except for the special one. For the special left interaction and for all right interactions, A' performs a relay of messages respectively with external honest verifiers and an external honest prover P that can use either w or w' as witness. The case in which P uses w corresponds to the game played using W_L for X_L while the case in which P uses w' corresponds to the game played using W'_L for X_L. Since D is a distinguisher between these two games, by giving in output the same value given by D, A' breaks Π with non-negligible probability which is a contradiction. □

4.3 Security Against Input-Adaptive Adversaries

An input-adaptive adversary A is allowed to choose new statements while running other proof systems. This feature holds both in case A plays as prover and in case A plays as verifier. The aim of A is therefore to exploit this feature in order to succeed in a man-in-the-middle attack, thus proving a statement that he could not prove otherwise (therefore attacking non-malleable zero knowledge) or proving statements that encode related witnesses (therefore attacking non-malleable witness indistinguishability).

Following the previous approach of [FLS99], we consider two input-adaptive indistinguishability tests. In the first test, that we refer to as the non-malleable witness indistinguishability test, the input-adaptive adversary A plays a MIM attack either with a real prover P_0 and a real verifier V or with a real prover P_1 and a real verifier V. These two provers, differ in the witnesses they use in their proofs. During the man-in-the-middle attack, A is allowed to generate triplets (x, w_0, w_1) that are sent to a black-box B, where both w_0 and w_1 are legal witnesses for proving $x \in L$. If $B = (P_0, V)$, then A will receive a proof that encodes w_0 as witness, while if $B = (P_1, V)$, then A will receive a proof that encodes w_1 as witness. Moreover, A will give proofs to B proving statements of its choice. This procedure is repeated any polynomial number of times and A succeeds if the distribution of the witnesses encoded in the proofs he gives (i.e., right sessions) when interacting with $B = (P_0, V)$ is computationally distinguishable from the one of the witnesses encoded in the proofs he gives when interacting with $B = (P_1, V)$.

In the second test, that we refer to as the non-malleable zero-knowledge test, the adversary A plays a MIM attack either with real prover and verifier algorithms P, V (i.e., $B = (P, V)$) or a simulator $S = (S_P, S_V)$ (i.e., $B = (S_P, S_V)$). Each time A wants to start a new left session, it sends a pair $(x, w) \in R$ to the black-box B. If $B = (P, V)$, A interacts with the honest prover algorithm on input a valid witness. Instead, if $B = (S_P, S_V)$, A interacts with a simulator on input x. In this case, input-adaptive non-malleability means that for all input-adaptive adversaries A, the distribution of the statements and the witnesses encoded in the proofs played in the right sessions when $B = (P, V)$ is computationally indistinguishable from the one given in output when $B = (S_P, S_V)$.

We now state again some of our previous theorems extending the claims to the input-adaptive case.

Theorem 4.8 *Under the assumption that a family of collision resistant hash functions exists, there exists a constant-round tag-based input-adaptive one-left many-right SBcNMWI commit-and-prove argument of knowledge for any polynomial-time relation R.*

Proof. The construction given for Theorem 4.4 shows a simulator that only cares about just one left session. The proof of Theorem 4.4 never uses the assumption that the statement proved by the simulator was fixed before the concurrent man-in-the-middle attack started, therefore our construction is an input-adaptive one-left many-right simulation-based concurrent NMWI argument of knowledge. □

Theorem 4.9 *Any input-adaptive one-left many-right SBcNMWI commit-and-prove argument of knowledge is an input-adaptive one-left many-right cNMWI commit-and-prove argument of knowledge.*

Proof. The proof of Theorem 4.5 instead fixes the statements before the MIM attack starts. However, it can be easily seen that since we just proved that for the simulation-based notion, input-adaptiveness is enjoyed by the construction, then the proof of Theorem 4.5 can be straightforwardly adapted to show that input-adaptiveness is enjoyed even for one-left many-right basic concurrent NM witness indistinguishability. □

Theorem 4.10 *Under the assumption that a family of collision resistant hash functions exists, there exists an input-adaptive cNMWI commit-and-prove arguments of knowledge in the plain model for all $\mathcal{NP}$ languages.*

Proof. Theorem 4.7 shows fully concurrency by assuming that all statements proved in the left sessions are fixed before the MIM attack starts. Notice that the proof of Theorem 4.7 shows how to contradict the proved one-left many-right case using hybrid arguments. When the statements are adaptively chosen by the adversary, the same hybrid arguments can be applied. Indeed, the only difference is that now the indistinguishability is obtained with a reduction to the input-adaptive one-left many-right basic concurrent NMWI argument of knowledge, already proved in Theorem 4.9. □

The impossibility results of Lindell [Lin03, Lin04]. As already proved in Theorem 4.6, we can not achieve in the plain model input-adaptive simulation-based concurrent NMWI arguments of knowledge and neither input-adaptive concurrent NMZK arguments of knowledge. Since a set-up assumption is necessary for achieving these results, in Section 5.1 we consider the BPK model that seems to be more realistic than the previously proposed set-up assumptions (trusted third parties, quasi-security, quasi-concurrency) and show how to achieve input-adaptiveness. Moreover, all constructions we give only use a constant-number of communication rounds. Our results therefore contrast with the recent results of [BPS06]. Indeed, they achieve input-non-adaptive concurrent non-malleable zero knowledge in the plain model with a super-constant number of rounds.

5 Concurrent NMZK in the BPK model

In this section we give a construction of a concurrent non-malleable ZK argument of knowledge in the BPK model.

5.1 The BPK Model

In the BPK model, the verifiers register their public keys in a directory during a preprocessing stage. In this stage, we assume no restriction on the power of the adversary that therefore has complete control of the directory and at the end of this stage outputs the (possibly maliciously adapted) list of public keys. After the preprocessing the power of the adversary is precisely the same of a fully concurrent mim.

The BPK model for interactive argument systems. We now review the definition of concurrently-secure interactive argument systems in the BPK model that were previously given in [MR01, Rey01] and the extension to the concurrent man-in-the-middle case. Formally, in the BPK model:

1. there exists a public file F that is a collection of records, each containing a public key;
2. a (honest) prover P is an interactive deterministic polynomial-time Turing machine that takes as input a security parameter 1^n, F, an n-bit string x, such that $x \in L$, for some language L, an auxiliary input y, a reference to an entry of F and a random tape;
3. a (honest) verifier V is an interactive deterministic polynomial-time Turing machine that works in the following two stages: (a) on input a security parameter 1^n and a random tape, V generates a key pair $(\mathtt{pk}, \mathtt{sk})$ and stores the public key pk in one entry of the file F; (b) later, V takes as input the secret key sk, a statement $x \in L$ and a random string, and outputs "accept" or "reject" after performing an interactive protocol with a prover;
4. the first interaction between a prover P and a verifier V starts after all verifiers have completed their first stage.

Concurrent malicious provers in the BPK model. Let s be a positive polynomial. We say that $P^\star$ is an *s-concurrent malicious* prover if it is a probabilistic polynomial-time Turing Machine that, on input 1^n and PK, can perform the $s(n)$ interactive protocols with V as follows: 1) if $P^\star$ is already running i protocols $0 \leq i < s(n)$ he can choose a new statement "$x_i \in L$" to be proved and start a new protocol with V with $x_i \in L$ as statement; 2) he can output a message for any running protocol, receive immediately the response from V and continue.

Given an s-concurrent malicious prover $P^\star$ and a honest verifier V, an *s-concurrent attack* is performed as follows: 1) the first stage of V is run on input 1^n and a random string to obtain pair $(\mathtt{pk}, \mathtt{sk})$; 2) $P^\star$ is run on input 1^n and pk so to obtain an n-bit string x_1; 3) whenever $P^\star$ starts a new protocol choosing an n-bit string x_i, V uses inputs x_i, a new random string r_i and sk, and interacts with $P^\star$.

Concurrent malicious verifiers in the BPK model. Let s be a positive polynomial. We say that $V^\star$ is an *s-concurrent malicious* verifier if it is a probabilistic polynomial-time Turing

Machine that, on input 1^n and PK, can perform the following $s(n)$ interactive protocols with P: 1) if $V^\star$ is already running i protocols $0 \leq i < s(n)$ he can decide the i-th protocol to be started with P; 2) he can output a message for any running protocol, receive immediately the next message from P and continue.

Given an s-concurrent malicious verifier $V^\star$ and a honest prover P, an *s-concurrent attack* is performed as follows: 1) in its first stage, $V^\star$, on input 1^n and a random string, generates a public file F; 2) $V^\star$ is run on input 1^n and F so to start the first protocol with P; 3) whenever $V^\star$ starts a new protocol, P uses a new statement, a new random string, and interacts with $V^\star$.

Concurrent malicious man-in-the-middle adversaries in the BPK model. Here we consider a man-in-the-middle adversary A that has all the power of a concurrent man-in-the-middle adversary in both stages. First, he receives the public keys of the honest verifiers, then he is allowed to compute both new independent keys and possibly related keys and outputs the public file F along with an auxiliary information τ. Therefore, A has complete control of F, the only restriction is the standard one of the BPK model, i.e., once F is announced, it can not be changed anymore. The second stage is precisely the game played in the plain model with the addition that A has access to τ, all parties know F and the verifiers know their secret keys. Playing this stage A joins the power of a concurrent malicious prover with that of a concurrent malicious verifier and, moreover he can exploit its adaptive choices of the first stage that allow him to use public keys that are related to the ones of the honest verifiers. More formally, A is a pair of algorithms (A_0, A_1) where A_0 receives as input a set of public keys and output a new set of public keys (i.e., the public file) along with an auxiliary information τ. A_1 is the concurrent man-in-the-middle adversary of the plain model with the difference that he receives τ as input and is forced to declare the public-key entry of the public file that has to be used in each concurrent session.

Definition 5.1 *Let $\Pi = \langle P, V \rangle$ be an argument system for a language L in the* BPK *model. Let $\bar{V}$ be the probabilistic polynomial-time algorithm that corresponds to the work of honest verifiers $V_1, \ldots, V_{\mathtt{POLY}(n)}$. We say that Π is a* concurrent non-malleable zero-knowledge *argument system in the* BPK *model if, for any man-in-the-middle adversary $A = (A_0, A_1)$ in the* BPK *model and any auxiliary information ω, there exists an expected polynomial-time algorithm $S_A = (S_A^P, S_A^V)$ such that the distributions of the outputs of the following experiments are computationally indistinguishable:*

$\mathtt{Expt}_{\mathcal{A}}(1^n, \omega)$*:* $(\mathtt{PK}, \mathtt{SK}) \leftarrow \bar{V}(1^n)$ $(\mathtt{PK}', \tau) \leftarrow \mathcal{A}_0(\mathtt{PK}, \omega)$ $\{\mathtt{mim}_{X_L}^{A_1}(X_L, \tau)\}_{X_L}$	$\mathtt{Expt}^{S(1^n,\omega)_A}$: $(\mathtt{PK}, \mathtt{SK}) \leftarrow S_A^V(1^n)$ $(\mathtt{PK}', \tau) \leftarrow \mathcal{A}_0(\mathtt{PK}, \omega)$ $\{\mathtt{sis}^{S_A^P(\mathtt{PK}',\mathtt{SK})}(X_L)\}_{X_L}$

Multi-party computation in the BPK model. The BPK model has been proposed by [CGGM00] for achieving round-efficient resettable zero knowledge. We stress that the BPK model requires a public key for any possible verifier, thus only verifiers run the set-up stage and upload their public keys to the public file. This asymmetry between provers and verifiers has sense in the context of zero-knowledge proof systems where the two parties have specific and different roles.

Unlike zero knowledge, when we consider the BPK for achieving the multi-party computation of a generic functionality, it is not clear how to specify an asymmetry between one party and another party. Therefore we will actually extend the set-up stage to all parties in the natural way. Each player during the set-up stage has to upload its public key, then at the end of the set-up stage, all parties dowload the public file and then second stage starts.

5.2 High-Level Overview of the Result

We now discuss the protocol for constant-round input-semi-adaptive concurrent non-malleable zero-knowledge arguments of knowledge for any $\mathcal{NP}$ language in the BPK model. In the preprocessing stage each verifier computes a pair of public keys along with the corresponding secret keys. He 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, randomly picking two messages $\mathtt{sk}_0, \mathtt{sk}_1$ in the domain of f, computing public keys $\mathtt{pk}_0 = f(\mathtt{sk}_0), \mathtt{pk}_1 = f(\mathtt{sk}_1)$ and using $\mathtt{sk}_b$ as secret key for a randomly chosen bit b.

The protocol for the relation R and common input x is a sequential composition of the input-semi-adaptive cNMWI commit-and-prove argument of knowledge presented above. In the first execution the verifier proves knowledge of one of the two secret keys, this is obviously done by $\mathcal{NP}$-reducing this instance to the $\mathcal{NP}$-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 w such that $R(x, w) = 1$ or of one of the two secret keys. 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.

Informally, since this protocol is only a double execution of the same subprotocol, the concurrent man-in-the-middle execution is directly reduced to the concurrent man-in-the-middle execution of the subprotocol. We stress that the main problem of the concurrent man-in-the-middle execution of the subprotocol is that it is concurrent non-malleable witness indistinguishable but not simulatable. Therefore, the goal is to let the simulator of the protocol know all witnesses he needs for running the subprotocol. The bare public-key model allows us to solve this problem. Indeed, in the BPK model when the simulator plays the role of verifier, he plays honestly the first stage and therefore knows the witness (i.e., one of two secret keys) for running the honest prover algorithm of the first execution of the subprotocol. Instead, when the simulator plays the role of prover, he needs the witness for the second subprotocol. He can therefore use the extractor of the first subprotocol to get the secret key (which is a valid witness for the second subprotocol) of the man-in-the-middle. The use of rewinds is dangerous in general but not in the BPK model as shown in [CGGM00]. Indeed the number of extraction procedures that have to be successfully run is independent of the number of concurrent sessions, since it is bounded by the size of the public file. Once the simulator knows the secret keys of the man-in-the-middle, the simulation is straight-line. The BPK model therefore, allows for any polynomial number of proofs played by a bounded number of verifiers.

5.3 The Protocol

The formal protocol is depicted in Figure 1.

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

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

V's private input: secret key $\mathtt{sk}$ such that either $\mathtt{pk}_0 = f(\mathtt{sk}_0)$ or $\mathtt{pk}_1 = f(\mathtt{sk}_1)$.

$V \longrightarrow P$: V and P engage in a cNMWI commit-and-prove argument of knowledge with tag $x \circ 0$ where V proves to P knowledge of $\mathtt{sk}$ such that either $\mathtt{pk}_0 = f(\mathtt{sk}_0)$ or $\mathtt{pk}_1 = f(\mathtt{sk}_1)$.

$P \longrightarrow V$: P and V engage in a cNMWI commit-and-prove argument of knowledge with tag $x \circ 1$ where P proves to V knowledge of w such that either $R(x, w) = 1$ or $\mathtt{pk}_0 = f(w)$ or $\mathtt{pk}_1 = f(w)$.

Figure 1: The constant-round cNMZK argument of knowledge for any $\mathcal{NP}$-language L with corresponding R in the BPK model.

Given the intuition discussed above, we now formally state and prove the result.

Theorem 5.2 *Under the assumption that a family of collision resistant hash functions exists, there exists a constant-round input-semi-adaptive concurrent NMZK (in the extraction sense) argument of knowledge for any polynomial-time computable relation R in the* BPK *model.*

Proof. Consider a $\mathtt{mim}$ A that opens any polynomial number of both left and right sessions, receives proofs of true statements in the left sessions and succeeds in proving a statements in the right sessions.

We now describe the efficient simulator S requested by the definition of concurrent NM zero knowledge. First of all we will use S to show that all new statements proved by A are true.

Indeed, assume by contradiction that A proves a false statement and assume that S knows the index i of the session in which A manages to convince a honest verifier of the validity of a false statement and denote by $\mathtt{tag}_i$ the tag used in this session. This assumption is wlog as S can correctly guess the right session in which the adversary succeeds with some non-negligible probability.

The simulator S performs the preprocessing stage and then interacts with A in the proof stage. The session in which A proves a false statement is relative to an entry of the public file (each entry contains two public keys). According to the definition of concurrent NMZK in the BPK model, we assume that all right interactions are performed relatively to entries constructed by S and for which S knows one of the associated secret keys (i.e., the other interactions are performed inside A since they correspond to entries that it generated/updated).

During the proof stage S interacts both in left sessions and right sessions. For each left session, we have two possible cases:

1. S knows the secret key corresponding to one of the two public keys relative to this session. In this case, S uses the secret key as witness to run the honest prover algorithm of the second subprotocol.
2. S does not know a secret key for the corresponding entry of the public file. In this case, S starts the extraction procedure relatively to the first subprotocol in which the adversary A proves knowledge of one of the two secret keys of that entry. Once one of the two secret keys is obtained, we have reduced ourselves to the first case.
   The extraction procedure involves rewinding the adversary. We stress though that, since entries of the public file cannot be changed once the proof stage has started, each time S obtains a secret key that can be used for all sessions relative to the same entry of the public file and that the number of entries is polynomially bounded.

Moreover, we stress that each time S obtains a new secret key, it can start the simulation of the proof stage from scratch, this achieves at some point, a straight-line simulation.

For the right sessions we distinguish between the i-th right session (the *special* session) and the other sessions (which we call the *regular* sessions). In a regular right session, S executes the code of the honest verifier. This is possible since S knows one of the secret keys associated with the corresponding entry of the public file. For the special right session instead, S executes the code of the honest verifier for the first subprotocol. Then, S executes the extraction procedure for the second protocol and obtains the witness ω used by the adversary. This is the combined simulation-extraction procedure already proposed in [PR05a] in the context of concurrent non-malleable commitments. In our case however, there is a subtle potential problem that arises. Indeed, the extraction procedures on the proofs given by A playing as prover could be concurrent with the extraction procedures on the proofs given by A playing as verifier. This could potentially introduce nested chains of rewinds that could affect the running time of the simulator. However, note that only one extraction procedure on the proofs given by A playing as prover is considered at any time[2]. Moreover, the extraction procedures on the proofs given by A playing as verifier have the property that they are bounded by the number of entries in the public file. The key observation is that at any moment, S focuses on one extraction procedure only, i.e., the first one that would not allow S to go ahead.

Once S has extracted the witness from the proof given by A, three cases are possible:

Case 1. ω is the secret key $\mathtt{sk}_b$ used by S in the first subprotocol;

Case 2. ω is the secret key $\mathtt{sk}_{1-b}$;

Case 3. ω is a witness for $x \in L$.

Suppose Case 1 above happens with overwhelming probability and $\mathtt{tag}_i$ has never been used by the simulator. We use A to construct an adversary A' that breaks the concurrent NM witness indistinguishability of the underlying protocol in the plain model. First of all, A' relays from an external prover P' to A all executions of the first subprotocol. P' uses a sequence of witnesses W over two different sequences of witnesses W_L, W'_L. The aim of A' is then to succeed in giving

[2] We stress that the extraction procedure of a witness starts only in case there is no session with an unfinished extraction procedure of a secret key.

to an external verifier V' a proof that encodes a witness that is related to the specific sequence of witnesses used by P'.

A' guesses i and therefore relies with V' the messages of the second subprotocol proved by A in the i-th session, while A' plays as honest verifier each other execution of the second subprotocol. Since case 1 happens with overwhelming probability, A encodes $\mathtt{sk}_0$ when $W = (\mathtt{sk}_0, \ldots, \mathtt{sk}_0)$ and encodes $\mathtt{sk}_1$ when $W = (\mathtt{sk}_1, \ldots, \mathtt{sk}_1)$, thus it holds that there exist two sequences of witnesses W_L, W'_L that differ only in one position, where either $\mathtt{sk}_0$ or $\mathtt{sk}_1$ is used. Letting P' choose between W_L and W'_L, we have that A succeeds with a non-negligible advantage over $1/2$ in giving a proof in which in the i-th session $\mathtt{sk}_0$ is encoded if P' uses W_L, while $\mathtt{sk}_1$ is encoded if P' uses W'_L. Therefore, with the same non-negligible advantage A' wins its game with P' and V'.

Suppose case 1 happens with overwhelming probability and $\mathtt{tag}_i$ has already been used by the simulator-extractor. Notice that since the tags used in the first subprotocols always end with 0, it must be the case that the simulator-extractor used this tag in the second subprotocol. However, in case the two tags are equal then A is proving the same statement proved by S and this contradicts the fact that A proved a new theorem.

Suppose case 1 does not happen with overwhelming probability. Since case 3 is impossible for false statements, it must be the case that case 2 happens with some non-negligible probability. In this case, A can be easily used to compute a secret key from a public key (e.g., in the discussed implementation, it can be used to invert a one-way function). This holds since an algorithm A', on input as challenge a public key, generates the other public key and plays with A the same game played by the simulator extractor. Since with non-negligible probability it obtains the secret key that corresponds to $\mathtt{pk}_{1-b}$, A' wins its game.

Since we have proved that all new statements proved by A are true, we now consider a $\mathtt{mim}$ adversary A that succeeds in proving new true statements Y_R. We show a stand-alone simulator A' that by accessing to A outputs the witnesses for all statements proved by A. The work of A' must be for A indistinguishable from that of honest P and V. A' during the preprocessing stage generates the pair of public/secret keys by running the honest verifier's algorithm, and thus he can later use the secret keys thus simulating perfectly the work of V. The work of P is instead more complex. Indeed, A' does not know the witnesses for running the second cNMWI commit-and-prove argument of knowledge of each left session. In order to get a witness, the strategy of A' is the same one discussed above, i.e., it first runs the extractor of the cNMWI commit-and-prove argument of knowledge given by the verifier. Since the number of possible verifiers is bounded by the size of the public file, the number of extraction procedures and thus the running time of A' are polynomial in the number of concurrent sessions played. The standard technique of extracting one-by-one the witnesses encoded in the proofs given by A' can be applied here. The only difference in the distribution of the messages played in the $\mathtt{mim}$ attack and in the simulation is that in a $\mathtt{mim}$ attack the provers use in the second subprotocol different witnesses with respect to the ones used by the simulator. However, as discussed above, the cNMWI property of the subprotocol guarantees that the distribution of the statements and of the witnesses extracted in the right sessions during the simulation are computationally indistinguishable from the statements and the witness encoded in the $\mathtt{mim}$ attacks in the right sessions. □

5.4 Security Against Input-Adaptive Adversaries

As previously discussed, in the plain model it is possible to construct a constant-round *input-adaptive* concurrent NMWI argument of knowledge. This is possible only with respect to the basic notion of witness indistinguishability since for the stronger simulation-based notion and in general for input-adaptive concurrent NMZK arguments of knowledge, the impossibility results of Lindell [Lin03, Lin04] hold.

We now show that in the BPK model we can actually overcome these impossibility results. We stress that the simulation-based notion of NM witness indistinguishability, requires the existence of a simulator S which has access to a MIM A and outputs witnesses with the same distribution as the ones encoded in the proofs given by A to a honest verifier V in a concurrent MIM attack. In the *input-adaptive* scenario we allow A to adaptively choose the statements for which he wants to see a proof. We next ask whether one can construct constant-round *input-adaptive* concurrent NMZK argument of knowledge in the BPK model. We positively answer this question in two steps.

Theorem 5.3 *Under the assumption that a family of collision resistant hash functions exists, there exists in the* BPK *model a constant-round* input-adaptive *concurrent NMZK argument of knowledge for any polynomial-time relation R.*

Proof. First we observe that the simulator of the proof of security of our construction in the BPK model (Theorem 5.2) does not require the statements to be proved to the MIM A to be fixed from the beginning. Indeed, once the simulator has managed to extract the secret keys of the MIM A, it can simulate the proof of *any* statement in a straight-line manner by using a proper witness (the secret keys of the MIM A) even though not the same witness that would have been used by a honest prover (which uses a witness for $x \in L$). If the MIM A behaves differently in the two cases (simulated proofs and real proofs) then we can easily use A for constructing a successful adversary for the input-adaptive concurrent NM witness indistinguishability of the subprotocol. This leads to a contradiction.

In the second step we observe that the simulator also works when asked a proof for $x \notin L$ which could be the case if we allow the MIM A to choose the theorems. Some extra attention is needed in arguing the case in which A tries to prove a false statement. However, the proof of Theorem 5.2 for the input-semi-adaptive case still works. Indeed we break our argument in three cases as in the input-semi-adaptive case: if Case 1 occurs, then we break the non-malleability of the subprotocol unless the tag was already used in some previous proof; but then the theorem is not new and thus the adversary is not considered successful; Case 2 contradicts the security of the one-way function; Case 3 is impossible. □

6 Zero Knowledge Under General Concurrent Composition in the BPK Model

In [Lin03, Lin04], Lindell proved that for a large class of functionalities, a protocol is secure under self-concurrent composition if and only if it is secure under concurrent *general* composition. This equivalence crucially needs that parties are allowed to choose their inputs adaptively, basing their choices on the output of the previous evaluations of the functionality. In the remaining part of this section we will only refer to this more realistic setting.

We start by reviewing the notion of a two-party functionality.

Definition 6.1 *A* two-party functionality $f : \{0,1\}^\star \times \{0,1\}^\star \rightarrow \{0,1\}^\star \times \{0,1\}^\star$ *maps pairs of inputs to pairs of output.*

It is convenient to write $f = (f_1, f_2)$ and thus, for each pair of inputs x, y, the output $f(x, y)$ can be written as $(f_1(x, y), f_2(x, y))$.

The generalization of two-party functionality to n parties is straightforward.

The zero-knowledge argument of knowledge functionality for a relation R can be seen as the functionality mapping pairs $((x, w), \lambda)$ to pairs $(\lambda, (x, R(x, w)))$.

The following definition is from [Lin03, Lin04].

Definition 6.2 *A two-party deterministic functionality $f = (f_1, f_2)$* enables bit transmission from P_1 to P_2 *if there exists an input y for P_2 and inputs x and x' for P_1 such that $f_2(x, y) \neq f_2(x', y)$.*

The zero-knowledge argument of knowledge functionality enables bit transmission from prover to verifier but not from verifier to prover. The "non-malleable zero-knowledge argument of knowledge functionality" is the functionality in which each party can play the role of the prover and thus it allows bit transmission in *both directions*.

We have the following theorem.

Theorem 6.3 *[Lin04] Let f be a two-party functionality that enables bit transmission in both directions and let* Π *be a protocol. Then* Π *securely computes f under concurrent self composition* if and only if Π *securely computes f under concurrent general composition.*

By combining Theorem 6.3 and Theorem 5.3 we have the following theorem.

Theorem 6.4 *Under the assumption that a family of collision resistant hash functions exists, there exists in the* BPK *model a constant-round protocol that securely realizes the ZK argument of knowledge functionality for any polynomial-time relation R under general concurrent composition.*

7 Secure Multi-Party Computation in the BPK Model

In this section we show how to use our implementation of the ZK argument of knowledge functionality that is secure under general concurrent composition (Theorem 6.4 above) for constructing a protocol for securely realizing multi-party functionalities under general concurrent composition.

7.1 Multi-Party Computation Under General Concurrent Composition

In this section we present a formal definition of a protocol realizing a multi-party functionality under general concurrent composition (the presentation follows [Lin04, Lin03]).

We start by presenting the *hybrid* model where we have a protocol π that utilizes ideal interactions with a trusted party computing a multi-party functionality f. This means that π contains two types of messages: standard messages and ideal messages. A standard message is one that is sent between two parties that are participating in the execution, using the point-to-point network (or broadcast channel, if assumed). An ideal message is one that is sent by a participating party to the trusted third party, or from the trusted third party to a participating party. This trusted party runs the code for f and associates all ideal messages with f. Notice that the computation is a "hybrid" between the ideal model (where a trusted party carries out the entire computation) and the real model (where the parties interact with each other only). Specifically, the messages are sent directly between the parties, and the trusted party is only used in the ideal call to f.

Computation in the hybrid model proceeds as follows. In the static corruption model, the computation begins with the adversary receiving the inputs and random tapes of the corrupted parties. Throughout the execution, the adversary controls these parties and can instruct them to send standard and ideal messages that it wishes. In the adaptive corruption model, the adversary can choose to corrupt parties throughout the computation. Upon corruption, the adversary receives the party's internal state and then controls the party for the remainder of the computation. In addition to controlling the corrupted parties, the adversary delivers all the standard and ideal messages by copying them from outgoing communication tapes to incoming communication tapes. The honest parties always follow the specification of protocol π. Specifically, upon receiving a message (delivered by the adversary), the party reads the message, carries out a local computation as instructed by π, and writes standard and/or ideal messages to its outgoing communication tape, as instructed by π. At the end of the computation, the honest parties write the output value on their output tapes, the corrupted parties output a special "corrupted" symbol and the adversary outputs an arbitrary function of its view.

Let k be the security parameter, let S be an adversary for the hybrid model with auxiliary input ω, and let $\bar{x}$ be the vector of the parties' inputs. Then, the hybrid execution of π with ideal functionality f, denoted $\textsc{hybrid}^{f}_{\pi,S}(k, \bar{x}, \omega)$, is defined as the output vector of all parties and S from the above hybrid execution.

In the real model instead the composition of protocol π with ρ is such that ρ takes the place of the ideal call to f. Formally, each party holds a separate probabilistic interactive Turing machine (ITM) that works according to ρ. When π instructs a party to send an ideal message α to the ideal functionality f, the party writes α on the input tape of its ITM for ρ and invokes the machine. Any message that it receives that is marked for ρ is forwarded to this ITM, while all other messages are answered according to π. Finally, when the execution of ρ concludes, the output of ρ is copied to the incoming communication tape for π (pretending it came from the ideal functionality f). This composition of π with ρ is denoted π^{ρ} and takes place without any trusted help. Thus, the computation proceeds in the same way as in the hybrid model, except that all messages are standard.

Let k be the security parameter, let A be an adversary for the real model with auxiliary input ω, and let $\bar{x}$ be the vector of the parties' inputs. Then, the real execution of π with ρ, denoted $\textsc{real}_{\pi^\rho,A}(k, \bar{x}, \omega)$, is defined as the output vector of all the parties and A from the above real execution.

Now we say that protocol ρ realizes f *under concurrent general composition* if for every protocol π in the f-hybrid model that utilizes ideal calls to f, every probabilistic polynomial-time real-model adversary A for π^ρ and for any $\omega \in \{0,1\}^\star$, there exists an expected polynomial-time hybrid-model adversary S such that: $\textsc{hybrid}^f_{\pi,S}(k, x, \omega)$ and $\textsc{real}_{\pi^\rho,A}(k, \bar{x}, \omega)$ are computationally indistinguishable.

That is, for any adversary A in the real model there exists another one S in the ideal world such that the two outputs are indistinguishable. In other words, if A gains any advantage then A could have gained also in the ideal execution where f is implemented ideally and thus the advantage of A does not derive from the specific protocol ρ implementing f.

We stress that we consider static corruption, assume the existence of authenticated channels and do not consider fairness issues.

7.2 Realizing F_{ZK} and Multi-Party Computation

We base our construction on compiler due to [CLOS02] that compiles any protocol for semi-honest parties into a protocol for malicious adversaries. The compiler of [CLOS02] assumes that access to zero-knowledge functionality F_{ZK} is available to all parties and that parties can communicate using a broadcast channel. The latter has been implemented by Goldwasser and Lindell [GL02]. The above chain of results can be summarized in the following theorem.

Theorem 7.1 *[GMW87, BMR90, Rog91, CLOS02, GL02] Assume that enhanced trapdoor permutations exist. Then for any n-party functionality f there exists a constant-round protocol that $(n-1)$-securely realizes f in the F_{ZK}-hybrid model with respect to static adversaries.*

By combining Theorem 6.4 (which shows how to securely realized in a constant number of rounds the zero knowledge argument of knowledge functionality under general concurrent composition) and Theorem 7.1 (that shows that a secure implementation of zero knowledge is sufficient for secure multi-party computation) we obtain the following theorem.

Theorem 7.2 *Assume that enhanced trapdoor permutations and collision-resistant hash functions exist. Then for any n-party functionality f there exists a constant-round protocol that $(n-1)$-securely realizes f under general concurrent composition in the* `BPK` *model with respect to static adversaries.*

8 Conclusions

In this paper we have introduced the notion of non-malleable witness indistinguishability. We have shown that the recent results and techniques of [PR05a] allow us to construct concurrent non-malleable witness-indistinguishable arguments of knowledge for any $\mathcal{NP}$-language in the plain model.

We used NMWI to construct constant-round input-adaptive concurrent non-malleable zero-knowledge arguments of knowledge in the BPK model. By plugging in our construction into general compilers for secure multi-party computation, we achieved constant-round zero-knowledge and $(n-1)$-secure multi-party computation under general concurrent composition in the BPK model.

The significance of our work lies in the wide applicability of the models in which the results have been obtained (namely the plain model and the BPK model) and the strong notions of security (general concurrent composition under polynomial-time simulation) achieved. We wonder whether other applications of NMWI can be used to obtain additional results in these hostile realistic settings.

9 Acknowledgments

We would like to thank Alon Rosen for useful discussions on non-malleable commitments and the anonymous reviewers of FOCS '06 for their comments. The work of the first author has been supported in part by Intel equipment grant, NSF Cybertrust grant No. 0430254, Xerox Innovation group Award and IBM Faculty Award. The work of the last two authors has been supported in part by the European Commission through the IST program under Contract IST-2002-507932 ECRYPT and through the FP6 program under contract FP6-1596 AEOLUS.

References

[Bar01] Boaz Barak. How to go beyond the black-box simulation barrier. In *42nd Annual Symposium on Foundations of Computer Science*, pages 106–115, Las Vegas, Nevada, USA, October 14–17, 2001. IEEE Computer Society Press.

[Bar02] Boaz Barak. Constant-round coin-tossing with a man in the middle or realizing the shared random string model. In *43rd Annual Symposium on Foundations of Computer Science*, pages 345–355, Vancouver, British Columbia, Canada, November 16–19, 2002. IEEE Computer Society Press.

[BCNP04] Boaz Barak, Ran Canetti, Jesper Buus Nielsen, and Rafael Pass. Universally composable protocols with relaxed set-up assumptions. In *45th Annual Symposium on Foundations of Computer Science*, pages 186–195. IEEE Computer Society Press, 2004.

[Blu82] Manuel Blum. Coin Flipping by Phone. In *24th IEEE Computer Conference (CompCon)*, pages 133–137, 1982.

[BMR90] Donald Beaver, Silvio Micali, and Phil Rogaway. The Round Complexity of Secure Protocols. In *22nd Annual ACM Symposium on Theory of Computing*, pages 503–513, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press.

[BPS06] Boaz Barak, Manoj Prabhakaran, and Amit Sahai. Concurrent non-malleable zero knowledge. In *47th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society Press, 2006.

[BS05] Boaz Barak and Amit Sahai. How to play almost any mental game over the net - concurrent composition via super-polynomial simulation. In *46th Annual Symposium on Foundations of Computer Science*, pages 543–552. IEEE Computer Society Press, 2005.

[CGGM00] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge. In *32nd Annual ACM Symposium on Theory of Computing*, pages 235–244, Portland, Oregon, USA, May 21–23, 2000. ACM Press.

[CKPR01] Ran Canetti, Joe Kilian, Erez Petrank, and Alon Rosen. Black-box concurrent zero-knowledge requires omega (log n) rounds. In *33rd Annual ACM Symposium on Theory of Computing*, pages 570–579, Crete, Greece, July 6–8, 2001. ACM Press.

[CLOS02] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In *34th Annual ACM Symposium on Theory of Computing*, pages 494–503, Montral, Qubec, Canada, May 19–21, 2002. ACM Press.

[CV05] Giovanni Di Crescenzo and Ivan Visconti. Concurrent zero knowledge in the public-key model. In *32nd International Colloquium on Automata, Languages, and Programm ing (ICALP 05)*, volume 3580 of *Lecture Notes in Computer Science*, pages 816–827. Springer-Verlag, 2005.

[DDN91] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. In *23rd Annual ACM Symposium on Theory of Computing*, pages 542–552, New Orleans, Louisiana, USA, May 6–8, 1991. ACM Press.

[DDN00] Danny Dolev, Cynthia Dwork, and Moni Naor. Nonmalleable cryptography. *SIAM Journal on Computing*, 30(2):391–437, 2000.

[DDO$^+$01] Alfredo De Santis, Giovanni Di Crescenzo, Rafail Ostrovsky, Giuseppe Persiano, and Amit Sahai. Robust non-interactive zero knowledge. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 566–598, Santa Barbara, CA, USA, August 19–23, 2001. Springer-Verlag, Berlin, Germany.

[DNS98] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. In *30th Annual ACM Symposium on Theory of Computing*, pages 409–418, Dallas, Texas, USA, May 23–26, 1998. ACM Press.

[DPV04] Giovanni Di Crescenzo, Giuseppe Persiano, and Ivan Visconti. Constant-round resettable zero knowledge with concurrent soundness in the bare public-key model. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 237–253, Santa Barbara, CA, USA, August 15–19, 2004. Springer-Verlag, Berlin, Germany.

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple NonInteractive Zero Knowledge Proofs under General Assumptions. *SIAM Journal on Computing*, 29:1–28, 1999.

[FS90] Uriel Feige and Adi Shamir. Witness indistinguishable and witness hiding protocols. In *22nd Annual ACM Symposium on Theory of Computing*, pages 416–426, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press.

[GL02] Shafi Goldwasser and Yehuda Lindell. Secure Computation without Agreement. In *Proc. of DISC 2002*, volume 2508 of *Lecture Notes in Computer Science*, pages 17–32, 2002.

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

[GMW87] O. Goldreich, S. Micali, and A. Wigderson. How to Play any Mental Game - A Completeness Theor em for Protocols with Honest Majority. In *19th ACM Symposium on Theory of Computing (STOC '87)*, pages 218–229, 1987.

[GMW91] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *Journal of the ACM*, 38(1):691–729, 1991.

[Kil90] Joe Kilian. *Uses of randomness in Algorithms and Protocols.* MIT Press, Cambridge, MA, 1990.

[KLP05] Yael Tauman Kalai, Yehuda Lindell, and Manoj Prabhakaran. Concurrent general composition of secure protocols in the timing model. In *37th Annual ACM Symposium on Theory of Computing*, pages 644–653. ACM Press, 2005.

[KP01] Joe Kilian and Erez Petrank. Concurrent and resettable zero-knowledge in polyloalgorithm rounds. In *33rd Annual ACM Symposium on Theory of Computing*, pages 560–569, Crete, Greece, July 6–8, 2001. ACM Press.

[KPR98] Joe Kilian, Erez Petrank, and Charles Rackoff. Lower bounds for zero knowledge on the internet. In *39th Annual Symposium on Foundations of Computer Science*, pages 484–492, Palo Alto, California, USA, November 8–11, 1998. IEEE Computer Society Press.

[Lin03] Yehuda Lindell. Bounded-Concurrent Secure Two-Party Computation With out Setup Assumptions. In *Proceedings of the 35th ACM Symposium on Theory of Computing (STOC '03)*, pages 683–692. ACM, 2003.

[Lin04] Yehuda Lindell. Lower Bounds for Concurrent Self Composition. In *1st Theory of Cryptography Conference (TCC '04)*, volume 2951 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, 2004.

[MR01] Silvio Micali and Leonid Reyzin. Soundness in the public-key model. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 542–565, Santa Barbara, CA, USA, August 19–23, 2001. Springer-Verlag, Berlin, Germany.

[Pas03] Rafael Pass. Simulation in quasi-polynomial time, and its application to protocol composition. In Eli Biham, editor, *Advances in Cryptology – EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 160–176, Warsaw, Poland, May 4–8, 2003. Springer-Verlag, Berlin, Germany.

[Pas04] Rafael Pass. Bounded-concurrent secure multi-party computation with a dishonest majority. In *36th Annual ACM Symposium on Theory of Computing*, pages 232–241. ACM Press, 2004.

[PR03] Rafael Pass and Alon Rosen. Bounded-concurrent secure two-party computation in a constant number of rounds. In *44th Annual Symposium on Foundations of Computer Science*, pages 404–413, Cambridge, Massachusetts, USA, October 11–14, 2003. IEEE Computer Society Press.

[PR05a] Rafael Pass and Alon Rosen. Concurrent non-malleable commitments. In *46th Annual Symposium on Foundations of Computer Science*, pages 563–572. IEEE Computer Society Press, 2005.

[PR05b] Rafael Pass and Alon Rosen. New and improved constructions of non-malleable cryptographic protocols. In *37th Annual ACM Symposium on Theory of Computing*, pages 533–542. ACM Press, 2005.

[PS04] Manoj Prabhakaran and Amit Sahai. New notions of security: achieving universal composability without trusted setup. In *36th Annual ACM Symposium on Theory of Computing*, pages 242–251. ACM Press, 2004.

[Rey01] Leonid Reyzin. *Zero-Knowledge with Public Keys, Ph.D. Thesis.* MIT, 2001.

[RK99] Ransom Richardson and Joe Kilian. On the concurrent composition of zero-knowledge proofs. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 415–431, Prague, Czech Republic, May 2–6, 1999. Springer-Verlag, Berlin, Germany.

[Rog91] Phil Rogaway. *The Round Complexity of Secure Protocols–MIT PhD Thesis.* 1991.

[Rom90] John Rompel. One-Way Functions are Necessary and Sufficient for Digital Signatures. In *22nd ACM Symposium on Theory of Computing (STOC '90)*, pages 12–19, 1990.

[Sah99] Amit Sahai. Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security. In *40th Annual Symposium on Foundations of Computer Science*, pages 543–553, New York, New York, USA, October 17–19, 1999. IEEE Computer Society Press.