On Input Indistinguishable Proof Systems

Rafail Ostrovsky[1], Giuseppe Persiano[2,⋆], and Ivan Visconti[2,⋆]

[1] University of California at Los Angeles
rafail@cs.ucla.edu
[2] Università di Salerno
giuper@gmail.com, visconti@unisa.it

Abstract. We study *Input Indistinguishable Computation* (IIC), a security notion proposed by Micali, Pass, and Rosen in [14] and recently considered also by Garg, Goyal, Jain and Sahai in [9]. IIC aims at generalizing the notion of a *Witness Indistinguishable (WI)* proof system to general two-party functionalities and in its concurrent version (cIIC) also considers security against man-in-the-middle (MiM) attacks.

In this paper, we focus on the proof system functionality and compare IIC with two other security notions for proof systems: WI and Non-Malleability (NM). We address the following two questions.

1. Since IIC is a generalization of WI from proof systems to general 2PC, are all WI proofs also IIC secure?
2. Are cIIC proofs also NM?

We show, somewhat surprisingly, that *both* answers to the above questions are negative. Indeed, we show that there exists a WI proof system that is not IIC secure. We then show that a large class of WI proof systems, including the classical Blum's proof system for NP, are concurrently secure in the IIC sense. This answers the second question in the negative, since Blum's proofs are known to be malleable.

The consequence of our results is three-fold. 1) IIC is a too stringent notion and this leaves the possibility of security notions weaker than IIC with a satisfying level of security. 2) For important functionalities, such as the proof system functionality, classical constructions like Blum's protocol are cIIC secure. 3) cIIC security should be carefully evaluated when used as a security guarantee to model real-world concurrent attacks to protocols, as our results show that cIIC security does not guarantee non-malleability of proof systems. In contrast, standard simulation-based security [5,2] and concurrent non-malleable WI (a game-based security notion introduced by [15,16]) are secure against MiM attacks (the latter even in constant rounds).

1 Introduction

Proof systems were introduced in [12] and their security was defined using the *simulation paradigm* through the notion of *Zero Knowledge* (ZK). *Witness Indistinguishability* (WI[1]) introduced by Feige and Shamir [8] is instead a game-based

[⋆] Work partially done while visiting University of California at Los Angeles.

[1] We will use WI to mean both witness indistinguishability and indistinguishable.

J. Esparza et al. (Eds.): ICALP 2014, Part I, LNCS 8572, pp. 895–906, 2014.

security notion for proof systems requiring that the adversarial verifier be not able to distinguish which of two given witnesses has been used by the prover. WI is easily seen to be implied by ZK and, under plausible complexity assumptions, there exist WI proof systems that are not ZK [6].

It was later observed that ZK is not preserved if more sophisticated attacks are considered. Dwork et al. [7] initiated the study of security under *concurrent* composition. That is, the adversarial verifier can play multiple concurrent sessions keeping control over the scheduling of the messages. It is easy to see that ZK is not closed under concurrent composition whereas WI is.

In a man-in-the-middle (MiM) attack an adversary $\mathcal{A}$ acts as a verifier interacting with a honest prover and, at the same time, as a prover interacting with a honest verifier. Security against MiM attacks is called *non-malleability* [5]. Concurrency and MiM can be combined by considering an adversary playing as prover and verifier in multiple sessions. Formalizing security under such attacks in the simulation paradigm gives the notion of *concurrent non-malleable ZK* [2] (cNMZK) and guarantees non-transferability of proofs. The existence of a cNMZK protocol in the plain model with sub-logarithmic round complexity is a major open problem. Ostrovsky et al. [15,16] proposed a game-based security notion for proof systems, *concurrent non-malleable WI* (cNMWI), that implies security against concurrent MiM attacks. cNMWI guarantees non-transferability of proofs and is achievable in constant rounds, avoiding the complications of the simulation paradigm. Proofs (rather than arguments) have been achieved in [4].

Beyond Proof Systems: Concurrently-Secure 2PC. Security of two-part computation (2PC) has been traditionally formalized within the simulation paradigm. When concurrent attacks are considered though a series of impossibility results hinted at the fact that this notion might be too stringent. Indeed, Lindell [13] proved that concurrently secure 2PC can not be achieved for several interesting functionalities. This first impossibility result relied on the use of adaptive inputs through concurrent executions of protocols for some specific functionalities. The result has been then strengthened to the static input case by [2], and later broader impossibility results have been proved in [1,11].

Input-Indistinguishable Computation. Given the above limitations of simulation-based notions capturing security against concurrent MiM attacks, Micali et al. [14] proposed a game-based notion. Informally, the notion of *Input Indistinguishability Computation* (*IIC*, in short) tries to formalize the following security goal for 2PC: suppose there is more than one input for player P_1 that is consistent with the output obtained by player P_2; then, even a malicious P_2 cannot distinguish which of the possible consistent inputs has been actually used by P_1 in an execution. This is very similar in flavor to what WI requires from a proof system. The notion of IIC can be extended by considering the concurrent setting, where several sessions can be concurrently played and the adversary is allowed to play different roles in different sessions. The goal of cIIC is to guarantee that the output of the honest players in some sessions is not affected by the inputs used by honest players in other sessions, and this must hold for all inputs of the honest players that would produce the same outputs in those other sessions. The goal of the adversary is to play concurrently in different sessions to create correlations among their inputs/outputs.

Interestingly, in [14] it is first shown that IIC is not closed under concurrent composition. In particular, this is proved by showing a successful concurrent MiM attack on a protocol implementing the coin-flipping functionality. In the same paper [14], by building on top of some powerful non-malleable subprotocols, the authors showed a constant-round cIIC protocol for any two-party functionality. This is a major result since for the first time a meaningful security notion is shown to be feasible in the concurrent setting without relying on set-up assumptions.

The Recent Work of Garg et al. [9]. Very recently, Garg, Goyal, Jain and Sahai [9][2] gave a different notion of IIC with a simulation-based flavor, that we refer to as sIIC. Their notion also applies to randomized functionalities and is therefore more general. The proposed definition however is unsatisfactory if there exists a "splitting input" as discussed in [14]. Then [9] proposed another formulation referred to as exdIIC, that implies both IIC and sIIC. Of course the concepts of sIIC and exdIIC are naturally extended to the concurrent setting.

IIC: The Impact on Proof Systems. IIC is of major importance for the following two reasons: a) there are several popular computations as the Millionaire Problem, where the goal is simply to keep the input hidden among all other possible inputs that produce the same output, and IIC seems to be sufficient to achieve such a type of security; b) constant-round cIIC for any 2-party functionality has been achieved, while the same result under the standard simulation-based notion of secure computation is impossible to achieve, regardless of the round complexity. cIIC is therefore an appealing security notion to model security under concurrent composition in the plain model.

In this work we will focus on relations among IIC and two well-studied security notions for proof systems: WI and NM. The reason is two-fold. First, IIC has been proposed as a generalization of WI. Second, a MiM attacks to a protocol implementing the coin-flipping functionality proved that IIC is not closed under concurrent MiM attacks. Ignoring details of definitions one would expect at least one of the following two implications be true.

1. Since IIC is a generalization of WI to general two-party functionalities when the proof system functionality is taken into account then IIC and WI should coincide. In other words, every WI proof system should be IIC secure and any IIC-secure proof system should be WI.
2. Since a cIIC-secure protocol must defeat some forms of MiM attacks, then any cIIC-secure proof system should be non-malleable too.

1.1 Our Results

In this paper we analyze IIC-security in proof systems. Indeed IIC has been defined with the goal of lifting up the notion of WI from the proof system

[2] When referring to [9], we will actually refer to the full version available at [10].

functionality to general 2-party functionalities. Therefore any subtlety in IIC for the proof system functionality is potentially reflected on many other 2-party functionality (in particular to the functionalities that are similar to the proof system functionality). We embark on the task of finding answers to the above two questions that try to relate IIC to WI and non-malleable of proof systems.

We show that there exists a non-conversation-based[3] WI system that does not enjoy IIC. This is indeed surprising since the notion of IIC has been introduced to generalize the notion of WI to any other 2-party functionality, and thus one should expect that WI proofs of knowledge be IIC secure too. We then show that most of WI proofs of knowledge found in the literature are also secure in the IIC sense for the proof-system functionality even under concurrent composition. Specifically, this holds for *conversation-based* proofs, therefore contradicting the second claim. Indeed this class of WI proofs of knowledge contains several malleable protocols such as Blum's protocol.

We consider the notions of sIIC and exdIIC proposed in [9] and show that in contrast to IIC, every WI PoK is also sIIC. Of course since exdIIC implies IIC, there exists a (non-conversation-based) WI PoK that is not exdIIC secure. We will also prove that any conversation-based WI PoK is also exdIIC secure.

Consequences. In addition to the conceptual relevance of showing somewhat unexpected relations among security notions, our results have the following three consequences in applications of IIC.

First, IIC does not generalize WI but only a stronger form of it. The impact of this is that it is still possible to give a weaker definition of IIC that still captures the desired flavor, but that is easier to achieve. sIIC goes in this direction.

Second, there are important functionalities (e.g., the proof system functionality) such that classic constructions (e.g., Blum's protocol) are already cIIC secure. Therefore depending on the functionality in question, cIIC-security could come for free, without resorting to the general and inefficient (based on the use of expensive non-malleable subprotocols) constructions shown in previous work.

Third, when relying on cIIC for some 2-party functionalities, the actual meaning of cIIC for the given functionality should be carefully evaluated. Indeed while simulation-based secure 2PC provides strong enough guarantees, the security of cIIC can be unsatisfying. Such a decreased security *does not* depend on the fact that non-transferability of proofs requires simulation. Indeed, for the case of proof systems, it is still possible to obtain non-malleability (i.e., non-transferability of proofs) under an indistinguishability notion (e.g., NMWI [15]). We show that the formulation of cIIC does not give such a guarantee.

2 Definitions

A polynomial-time relation R is a relation for which it is possible to verify in time polynomial in $|x|$ whether $R(x, w) = 1$. We consider $\mathbb{NP}$-languages L and

[3] In a conversation-based proof the transcript identifies the common instance x and with overwhelming probability whether the verifier accepted or rejected.

denote by R_L the corresponding polynomial-time relation such that $x \in L$ if and only if there exists w such that $R_L(x, w) = 1$. We call such a w a *valid witness for* $x \in L$ and denote by $W_L(x)$ the set of valid witnesses for $x \in L$. We slightly abuse notation and, whenever L is clear from the context, we simply write $W(x)$ instead of $W_L(x)$. For sequences $X = (x_1, \cdots, x_m)$ and $W = (w_1, \cdots, w_m)$, by the writing "$W \in W(X)$" we mean that $w_i \in W(x_i)$ for $i = 1, \cdots, m$.

For a language L we will denote by L_n^m the set of sequences of m elements of L each of length at most n. A *negligible* function $\nu(k)$ is a function such that for any constant $c < 0$ and for all sufficiently large k, $\nu(k) < k^c$.

We stress that we will always refer to polynomial-time adversaries, therefore when we say proof systems or PoK, we actually refer to arguments.

We will use the standard definitions of proof system, WI and and the definition of IIC given in [14].

3 Input Indistinguishability vs WI

In this section we consider the notion of IIC [14] for the proof system functionality and compare it with the notion of a WI proof system [8]. While it is trivial to see from the definitions that any IIC proof is also WI, we show that the opposite implication does not hold.

We first show that a large class of WI proof systems (that includes all WI proof systems in the literature) also enjoys cIIC. However we will also show that this does not hold for all WI proof systems. The above large class consists of all WI proofs of knowledge that are *conversation-based*; that is, one can guess the output of the verifier (that is, whether the verifier accepts) by looking at the transcript of the protocol and, possibly, running in super-polynomial time. It is easy to see that all WI proof systems in the literature enjoy this property even in a very stringent sense, since the sole transcript is usually sufficient to efficiently guess whether the verifier accepts.

In this section, we denote the prover P by P_1 and the verifier V by P_2 in order to keep notation consistent with [14].

The Proof System Functionality $\mathcal{F}^L_{\mathcal{PK}}$. The input of (the prover) P_1 for functionality $\mathcal{F}^L_{\mathcal{PK}}$ for $\mathbb{NP}$ language L consists of a pair (x, y) whereas (the verifier) P_2 has in input only x. The output f_1 of P_1 in $\mathcal{F}^L_{\mathcal{PK}}((x, y), x)$ is defined as $f_1((x, y), x) = \perp$ (i.e., P_1 does not get any output); output f_2 of P_2 instead is defined as $f_2((x, y), x) = 1$ if y is a valid $\mathbb{NP}$ witness for $x \in L$; and $f_2((x, y), x) = 0$, otherwise.

Notice that $\mathcal{F}^L_{\mathcal{PK}}$ is defined with the two players having common input x. Whenever L is clear from the context, we will simply write $\mathcal{F}_{\mathcal{PK}}$.

Remark 1. One could think of using a different definition for the ideal functionality of a proof system where prover and verifier can have different statements as input. With such a different definition, then it happens that even the standard zero-knowledge PoK of Blum (e.g., sequential repetition of the classical 3-round protocol with a 1-bit challenge) would not be a secure instantiation (in the classical 2PC sense) of such a proof system functionality. The reason is that with such a functionality, the input statement x of the prover should remain private when playing with a V^* that runs on input a statement x' different from x. Instead in Blum's protocol the statement proven by the prover is not private at all. In general implementing such a functionality could require techniques/assumptions taken from general 2-party computation. Therefore we find such a definition of an ideal functionality less intuitive than the one that we use in the paper and that follows in spirit the formulation of [12].

Conversation-Based WI Proofs. We say that a WI proof is *conversation-based* if, given a transcript of the protocol it is possible to identify the common instance x and to compute with overwhelming probability the output of the honest verifier. We stress that no time bound is imposed on the decision procedure. All standard WI proofs (including Blum's protocol [3]) are in this category.

3.1 Conversation-Based WI $\Rightarrow$ IIC

In this section we prove that all conversation-based WI proof systems with perfect completeness are also cIIC for the proof system functionality $\mathcal{F}_{\mathcal{PK}}$. Thus, following Definition 1 and 2 of [14], we exhibit, for any conversation-based WI proof, a first-party and second-party implicit input functions $\mathsf{IN}_1, \mathsf{IN}_2$ that fulfill all requirements of IIC.

Defining the Implicit-Input Functions for $\mathcal{F}_{\mathcal{PK}}$. We remind the reader that, according to the definition of IIC, implicit-input functions are not necessarily efficiently computable.

IN_1: Let $\mathsf{View}_1^*(\mathbf{e})$ be the full view of P_1^* of an execution $\mathbf{e}$ of (P_1^*, P_2) (this includes the private coins and the input of P_1^*). For each session i of $\mathbf{e}$, the output of IN_1 on input $\mathsf{View}_1^*(\mathbf{e})$ is defined as follows.

If $\mathtt{OUTPUT}_1^i(\mathbf{e}) = 1$ and the verifier P_2 accepted the proof (this can be decided because of the conversation-based property), then IN_1 outputs a pair consisting of the instance x that is obtained from $\mathsf{View}_1^*(\mathbf{e})$ (as it is the i-th common input), and the lexicographically first witness y for $x \in L$. Instead, IN_1 outputs $\perp$ for all sessions i in which $\mathtt{OUTPUT}_1^i(\mathbf{e}) = 0$ or the verifier P_2 did not accept the proof (again, this is can be decided using the conversation-based property).

IN_2: Let $\mathsf{View}_2^*(\mathbf{e})$ be the full view of P_2^* of an execution $\mathbf{e}$ of (P_1, P_2^*). For each session i of $\mathbf{e}$, if $\mathtt{OUTPUT}_2^i(\mathbf{e}) = 1$, IN_2 on input $\mathsf{View}_2^*(\mathbf{e})$ outputs the statement x that is obtained from $\mathsf{View}_2^*(\mathbf{e})$ since it is the i-th common input, and outputs $\perp$ otherwise.

First of all, notice that IN_1 and IN_2 are implicit functions (i.e., they both output $\perp$ in case of aborts).

Completeness. For any session i, $\text{Prob}\left[\ P_1(\mathsf{View}_1^i(\mathbf{e})) = f_1((x_i, y_i), x_i)\ \right] = 1$ and $\text{Prob}\left[\ P_2(\mathsf{View}_2^i(\mathbf{e})) = f_2((x_i, y_i), x_i)\ \right] = 1$. Indeed, for the former, notice that f_1 always outputs $\perp$, and honest prover P_1 never outputs a value different than $\perp$; for the latter, notice that P_2 outputs precisely a bit denoting accept or reject and this coincides with the output of f_2. The perfect completeness property of the WI proof is required to prover the IIC completeness.

Implicit Computation. Let P_2^* be the adversary. W $\text{Prob}\left[\ P_1(\mathsf{View}_1^i(\mathbf{e})) = \perp\ \right] = 1$ in session i, and also $f_1((x_i, y_i), x_i^*) = \perp$ where $x_i^* =$ is the i-th component of the output of $\mathsf{IN}_2(\mathsf{View}_2^*(\mathbf{e}))$. Here notice that regardless of the value of $\mathtt{OUTPUT}_1^i(\mathbf{e})$, both $P_1(\mathsf{View}_1^i(\mathbf{e})) = \perp$ and $f_1((x_i, y_i), x_i^*)$ are always equal to $\perp$.

Let P_1^* be the adversary. If $\mathtt{OUTPUT}_2^i$ is false then $\text{Prob}\left[\ P_2(\mathsf{View}_2^i(\mathbf{e})) = \perp\ \right] = 1$ since the output delivery message of the i session is not in the view of $\mathbf{e}$. In case $\mathtt{OUTPUT}_2^i$ is true, we have that the i-th component (x_i, y_i^*) of the output of $\mathsf{IN}_1(\mathsf{View}_1^*(\mathbf{e}))$, is a valid theorem-witness pair for the i-th component x_i of the input of P_2 in the i-th session, only if P_2 gives in output 1. Therefore the implicit function IN_1 always outputs a value that makes the evaluation of f_2 consistent with the output of P_2[4].

Input Indistinguishability and Independence. We next show that for any adversary P_2^* it holds that $\{\mathsf{EXPT}^{P_1,P_2^*}((\mathbf{x},\mathbf{y}^1),(\mathbf{x},\mathbf{y}^2),\mathbf{x};1^n)\}$ is indistinguishable from $\{\mathsf{EXPT}^{P_1,P_2^*}((\mathbf{x},\mathbf{y}^2),(\mathbf{x},\mathbf{y}^1),\mathbf{x};1^n)\}$.

Indeed, the input function IN_2 selects x_i from $\mathsf{View}_2^*(\mathbf{e})$ independently of the other inputs. Since those other inputs are the only differences between the two experiments we have that by the WI of the views, the outputs $(\mathbf{x}^*, \mathsf{View}_2^*(\mathbf{e}))$ of both experiments are computationally indistinguishable.

Let us now consider adversary P_1^*. In this case we have that, since the verifier has as input only $\mathbf{x}$, both experiments correspond to $\{\mathsf{EXPT}^{P_1^*,P_2}((\mathbf{x},\mathbf{y}),\mathbf{x},\mathbf{x};1^n)\}$. The input function IN_1 defined above selects the instance-witness pair (x_i, y_i) for the i-session from the view of the i-th execution independently of the witness that has been actually used (as long as the transcript is accepting), since IN_1 considers the first witness in lexicographic order. Therefore both experiments produce the same output $(\mathbf{y}^*, \mathsf{View}_1^*(\mathbf{e}))$.

From Fixed Roles to a General Adversarial Behavior. Notice that in the discussion above, we have considered a fixed-role adversay only; that is, an adversary that either plays the role of the prover in all sessions or it plays the role of verifier. Intuitively, we used the following two facts: 1) when the adversary is a verifier, it has as input only x and the output by EXPT is a tuple of pairs (y, View), one for each session, where y is actually independent of the witness used by the prover, and View is a witness indistinguishable transcript; 2) when the adversary is a prover, the honest verifier has no private input and thus the two experiments EXPT of the definition collapse in one experiment only, so that indistinguishability of the output is trivial.

[4] The fact that IN_1 uses the conversation-based property is critical. Indeed we will exploit this to show that there exists a WI proof system that does not enjoy IIC.

In general, the adversary could play a man-in-the-middle attack; that is, it could play the role of the prover in some sessions and the role of the verifier in other sessions. We next argue that the above analysis still works. Indeed, based on the two possible sequences of inputs of the honest provers, we have that:

1) in the sessions where the adversary played as verifier, the output of EXPT contains witnesses unrelated to the ones used by honest provers and views that do not allow one to distinguish which witness has been used; 2) in the sessions where the adversary played as prover, we will have statements and views that again can only vary according to the sequence of witnesses used by honest provers in other sessions, which in turn means that, by the witness indistinguishability of those proofs, these views are indistinguishable as well.

Thus we proved the following theorem.

Theorem 1. *Any conversation-based WI proof system for an* $\mathbb{NP}$ *language* L *is IIC (even under concurrent composition) for* $\mathcal{F}^L_{\mathcal{PK}}$.

3.2 WI $\not\Rightarrow$ IIC for $\mathcal{F}_{\mathcal{PK}}$

Here, we show that there exist WI proof systems that do not enjoy IIC.

Theorem 2. *There exists a WI proof system* Π *for* $\mathbb{NP}$ *language* L *that is not IIC for functionality* $\mathcal{F}^L_{\mathcal{PK}}$.

Proof. Consider the classical proof system pBLUM that consists of parallel executions of Blum's protocol for the $\mathbb{NP}$-complete language of the Hamiltonian graphs [3]. More precisely, in the first round of pBLUM with security parameter k and input graph G, the prover selects k random permutations $\pi_1, \ldots, \pi_k$, computes graphs $G_1, \ldots, G_k$ where $G_i = \pi_i(G)$ and sends the commitments of the adjacency matrices of the k graphs. The verifier picks random bits $b_1, \ldots, b_k$ and sends them to the verifier. Finally, for each i for which $b_i = 0$, the prover opens all the commitments of the adjacency matrix of G_i and sends π_i; instead, for each i for which $b_i = 1$, the prover opens the commitments of the adjacency matrix of G_i that correspond to edges in a Hamiltonian cycle. The verifier accepts if and only if all the k answers obtained are correct.

It is easy to see that pBLUM is a WI proof system with perfect completeness and negligible (in k) soundness error for the language of the Hamiltonian graphs. Also, pBLUM is conversation-based since the final decision of the verifier is solely based on the transcript.

To prove Theorem 2, we artificially modify pBLUM by requiring the honest verifier V to randomly select $j \in \{1, \ldots, k\}$ and to neglect the answer of the prover in the j-th parallel execution in deciding whether to accept or not. The resulting protocol, mBLUM, enjoys perfect completeness and negligible soundness error and is still WI. However, mBLUM is not conversation-based. Indeed, a malicious prover P^* could use a string s hardwired in its code to decide to play wrongly in exactly one of the k parallel executions while playing honestly in the remaining ones. Notice that the honest verifier for mBLUM will accept the proof of P^* with non-negligible probability $1/k$. Indeed, it will accept exactly when the randomly selected parallel execution of the protocol corresponds to the one specified by s. Therefore, by looking at the transcript one can not guess and be correct with overwhelming probability if the honest verifier accepted or not. Notice that this holds unconditionally, even when the private input and coins of the prover are known.

Formally, assume that the instance is x and the witness is y. We have that the Implicit Computation property does now hold when P_1^* is the above adversarial prover and P_2 is the above honest verifier. Indeed in the above execution it happens that $P_2(\mathsf{View}_2^1) = 1$ with probability $1/k$ and $P_2(\mathsf{View}_2^1) = 0$ with probability $1 - 1/k$. The probability is over (a subset of) the private coins of P_2 that are independent from the transcript. Therefore to satisfy the Implicit Computation property one should have an implicit function IN_1 that on input View_1^* guesses with overwhelming probability the output of P_2. However, View_1^* does not contain the private coins used by P_2 to discard one of the parallel executions of Blum's protocol, therefore any implicit function IN_1 fails with non-negligible probability[5].

The existence of a WI proof system that is not IIC proves that IIC is a generalization to general functionalities of some special forms of WI only.

4 Simulation-Based IIC: sIIC and exdIIC

Here we study some relations among WI, IIC and sIIC and exdIIC [10]. We stress that even though we focus on the proof-system functionality, it is expected that our results extend to several other functionalities.

We next briefly review the notions of sIIC and exdIIC and refer the reader to Definition 5 and Definition 7 of [10] for formal definitions.

Classical (simulation-based) concurrently-secure 2PC requires the existence of a simulator S for every real-world adversary A so that the views of the adversary in the real world and in the ideal world are indistinguishable. Roughly speaking, sIIC (called IIC in [10]) relaxes this requirement by allowing the simulator S to depend also on the pair of input vectors of the honest party and by only requiring that the distributions of the outputs of the two party to be indistinguishable. The definition of *extended Input Indistinguishable Computation* (exdIIC, for short) strengthens the notion of sIIC by requiring indistinguishability between the ideal and real world of the pair consisting of the output of the parties (so far, it is similar to sIIC) and the input of the adversary which for the real world is defined by means of an implicit function IN that extracts the input from the view. For further details on sIIC and exdIIC, see Definition 5 and Definition 7 of [10].

[5] Indeed even in case one can have a randomized implicit function IN_1 that with probability $1-1/k$ outputs $\perp$, it would work against the above P_1^*, but it would fail against another adversary P_1^{**} that just plays as honest prover and always convinces the verifier.

4.1 WI and IIC vs sIIC

Theorem 3. *Any WI PoK for an* $\mathbb{NP}$ *language* L *is sIIC for* $\mathcal{F}^{L}_{\mathcal{PK}}$.

Proof. First of all remember that in the definition of [9], the simulator can be different for different pairs of inputs obtained by the honest player.

When the real-world prover is honest, the simulator plays as verifier in the ideal world and simulates a prover against the malicious verifier. The simulator internally has two witnesses (sIIC allows a different simulator for each pair of inputs for the honest player) for the theorem corresponding to the input statement and picks one of them to be used with the malicious verifier, playing then the protocol of the honest prover.

When the real-world verifier is honest, the simulator plays as prover in the ideal world and simulates a verifier against the malicious prover. Since the honest player (i.e., the verifier) has no witness, the simulator will have no witness as well. However the PoK property guarantees that the simulator can extract a witness from the malicious prover and can then play it in the ideal world.

It is easy to see that if the output of the ideal-world experiment differs form the one of the real world, one can easily break the WI of the proof system. Indeed notice that for the sessions where the simulator is an ideal-world verifier, the only deviation with respect to the real world consists in the fact that the simulator might use a different witness. For the sessions where the simulator is a real-world verifier, the only deviation with respect to the real world consists in the fact that the simulator has to extract a witness from P^* in order to play it in the ideal world. The PoK property guarantees that this can be done.

Notice that rewinding the adversary in the concurrent setting is often dangerous and can blow up the running time of the simulator. Nevertheless, since here the simulator rewinds only the malicious prover, there is no issue with its running time. The reason is that rewinds are related to extractions and can therefore be done sequentially, applying the extractor to the final transcript. During each extraction, there are no rewinds related to other sessions.

4.2 WI and IIC vs exdIIC

With the purpose of having a definition that also captures the security goals of IIC, Garg et al. in [9] defined exdIIC and proved that it implies both IIC of [14] and sIIC. One might think that exdIIC is a strengthening of IIC that requires stronger security properties (indeed it is a simulation-based notion) and it could be possible that several protocols that are IIC are not exdIIC. We show that for the proof system functionality this is not the case as we prove that any conversation-based WI PoK is also secure in the exdIIC sense.

Theorem 4. *Any conversation-based WI PoK for* $\mathbb{NP}$ *language* L *is exdIIC for functionality* $\mathcal{F}^{L}_{\mathcal{PK}}$.

Proof. The definition of exdIIC considers the indistinguishability of ideal and real world experiments also including 1) in the distribution of the ideal-world experiment, the inputs sent by the adversary to the trusted party and 2) in the distribution of the real-world experiment, the outputs of implicit functions $\mathsf{IN}_1, \mathsf{IN}_2$ on inputs the views of the adversaries.

The proof given for the case of sIIC is not sufficient here because we need to define implicit functions first, and then we must make sure that the inputs sent by the simulator to the ideal functionality are consistent with the outputs of the implicit functions.

Since the honest verifier of a PoK runs on input just the statement of the theorems, the implicit function IN_2 can just output the list of theorems (accepting or not) belonging to the view received in input. Therefore the only problem is to define the implicit function IN_1 that receives as input the view of the adversarial prover. Our choice is to have IN_1 to output, for each theorem in the view received in input, the first valid witness in lexicographic order provided that the transcript of the session is accepting[6], otherwise it will output $\perp$.

In order to have that the inputs sent to the functionality by the ideal-world adversary be consistent with the output of IN_1 we consider a simulator that first runs internally as verifier against the real-world adversary and checks if it gets a convincing proof. If this is the case, the simulator send to the functionality the witness that it has hardwired in its code. Notice that since in the definition of [9] there is a simulator for any pair of inputs, we have that there always exists a simulator that contains hardwired in its code the first witness in lexicographic order corresponding to the theorem specified in the input of the prover.

The case of an ideal-world adversarial verifier is simpler. As soon as a proof starts the simulator sends the theorem to the ideal functionality, and if it gets as output 1, it runs internally the honest prover procedure using the witness that it has hardwired in its code. If instead it receives 0, it just sends and abort message to the real-world adversarial verifier (the same things is of course done by a prover when the theorem to be proved is different from the one expected by the verifier). As for the case of sIIC, this proposed simulation is indistinguishable by the WI of the underlying proof system. Therefore the theorem holds.

It is worthy to notice the point in which the above theorem would fail in case of non-conversation based WI proof systems. Indeed we have that IN_1 could output a witness even when the verifier, because of his private coins, does not accept that transcript. Therefore in the real-world experiment, the verifier would output 0 while the output of IN_1 would be a witness. Then in the ideal-world experiment the simulator would be required to send the same witness, which of course allows the honest verifier of the ideal world to obtain 1 as output. This would clearly make ideal and real worlds distinguishable.

Acknowledgments. Work supported in part by NSF grants 09165174, 1065276, 1118126 and 1136174, US-Israel BSF grant 2008411, OKAWA Foundation Research Award, IBM Faculty Research Award, Xerox Faculty Research Award, B. John Garrick Foundation Award, Teradata Research Award, and Lockheed-Martin Corporation Research Award. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S.

[6] This restricts the statement of our theorem to conversation-based proofs only.

Office of Naval Research under Contract N00014 -11 -1-0392. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

References

1. Agrawal, S., Goyal, V., Jain, A., Prabhakaran, M., Sahai, A.: New impossibility results for concurrent composition and a non-interactive completeness theorem for secure computation. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 443–460. Springer, Heidelberg (2012)
2. Barak, B., Prabhakaran, M., Sahai, A.: Concurrent non-malleable zero knowledge. In: 47th FOCS. IEEE Computer Society Press (2006)
3. Blum, M.: How to Prove a Theorem So No One Else Can Claim It. In: Proceedings of the International Congress of Mathematicians, pp. 1444–1451 (1986)
4. Cao, Z., Visconti, I., Zhang, Z.: On constant-round concurrent non-malleable proof systems. Inf. Process. Lett. 111(18), 883–890 (2011)
5. Dolev, D., Dwork, C., Naor, M.: Non-malleable cryptography. In: 23rd ACM STOC, pp. 542–552. ACM Press (1991)
6. Dwork, C., Naor, M.: ZAPs and their applications. In: 41st FOCS, pp. 283–293. IEEE Computer Society Press (2000)
7. Dwork, C., Naor, M., Sahai, A.: Concurrent zero-knowledge. In: 30th ACM STOC, pp. 409–418. ACM Press (1998)
8. Feige, U., Shamir, A.: Witness indistinguishable and witness hiding protocols. In: 22nd ACM STOC, pp. 416–426. ACM Press (1990)
9. Garg, S., Goyal, V., Jain, A., Sahai, A.: Concurrently secure computation in constant rounds. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 99–116. Springer, Heidelberg (2012)
10. Garg, S., Goyal, V., Jain, A., Sahai, A.: Concurrently secure computation in constant rounds (full version) (2012), http://goo.gl/iPXSbe
11. Garg, S., Kumarasubramanian, A., Ostrovsky, R., Visconti, I.: Impossibility results for static input secure computation. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 424–442. Springer, Heidelberg (2012)
12. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM Journal on Computing 18(1), 186–208 (1989)
13. Lindell, Y.: Lower Bounds for Concurrent Self Composition. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 203–222. Springer, Heidelberg (2004)
14. Micali, S., Pass, R., Rosen, A.: Input-indistinguishable computation. In: 47th FOCS, pp. 136–145. IEEE Computer Society Press (2006)
15. Ostrovsky, R., Persiano, G., Visconti, I.: Constant-round concurrent non-malleable zero knowledge in the bare public-key model. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 548–559. Springer, Heidelberg (2008)
16. Ostrovsky, R., Persiano, G., Visconti, I.: Concurrent non-malleable witness indistinguishability and its applications. Electronic Colloquium on Computational Complexity (ECCC) 13(95) (2006)