Secure Computation Under Network and Physical Attacks

Alessandra Scafuro
Università degli Studi di Salerno

Abstract

Secure computation enables many parties to jointly compute a function of their private inputs. The security requirement is that the input privacy of any honest party is preserved even if other parties participating in the protocol collude or deviate from the protocol. In concurrent and physical attacks, adversarial parties try to break the privacy of honest parties by exploiting the network connection or physical weaknesses of the honest parties' machine.

This article provides an overview of the results for achieving secure computation in presence of concurrent and physical attacks contained in the phD thesis:"Secure Computation under concurrent and physical attacks", with emphasis to the relation of such results with the state of the art.

1 Introduction

The setting of secure computation is the following. There are many entities, that we call parties. Parties are mutually distrustful, and want to jointly compute a function of their inputs while keeping such inputs secret. As an example of this setting, consider a voting system. Each party has as secret input a preference, i.e., the name of the person that the party wants to vote. The joint function run by the parties takes in input all the secret preferences (the votes) and computes the majority. As it is already apparent from this example, besides the correctness of the result, an important property that we require from a voting system is that the preference of each player remains private.

Informally, a protocol for function evaluation is secure if each player does not learn anything more than what is leaked from the knowledge of the output and the knowledge of its own private input, even if the party actively deviate from the protocol.

How can we formally model the concept of not learning anything in this setting? In the seminal work [39] Goldwasser et al. introduce the concept of the simulator. The idea is that anything that an adversary can learn by interacting with the actual parties participating in the protocol, it can be learned also by a simulator, that is not interacting with any real party but it simulates the protocol executions in his head using fake inputs (for the honest parties). Very informally, to prove that a protocol is secure, is sufficient to show that for any adversarial party attacking the protocol in a real world execution run with the honest parties, there exists a simulator that successfully performs the very same attack, but executing the protocol in his head, without knowing the input of the honest players, and having access to an oracle that computes the function.

The above definition of secure computation is a good starting point, but it is still very far from capturing real-life setting. Indeed, in the setting above we have considered a very restricted adversary, that participates to one protocol execution only, and that can misbehave by corrupting a subset of parties. Thus, proving security in this setting, implies that a protocol will be secure as long as it will be run in isolation, namely parties cannot be involved in the execution of other protocols.

However, in real-life scenarios many functionalities are run over the internet. Thus, an adversary does not necessarily run one execution of the protocol in isolation, but it can activate and interleave many executions of other (or the same) protocols. Hence, in the formal security definition one should model the adversary as an entity that initiates and actively participates in many protocol executions, that can be run concurrently.

Such general security definition was introduced in [12] and is called Universal Composability. In the Universal Composability framework, besides the simulator and the adversary seen before, a new entity called "environment" is introduced. While the simulator and the adversary still model one execution of the target protocol for which one wants to prove security, the purpose of the environment is to model the concurrent executions of other (possibly different) protocols, that are run together with the target protocol. The environment can of course communicate with the adversary. This models the fact that while attacking the target protocol, the adversary can exploit the information gained from the concurrent executions. The Universal Composability (UC) framework is very general, and thus provides very strong security guarantees. Therefore, is desirable to have protocols that can be proved secure in this model. Unfortunately, it has been shown in [17] that it is impossible to design a protocol that achieves such definition, without the help of some setup assumption. A setup can be seen as some tool that parties can use to run the protocol. A setup is *trusted* if it is assumed that the adversary cannot participate in the generation of such setup. For example, a trusted setup can be that each party is given a smart card, and the adversary cannot participate in the process of generating/delivering such smart cards. Therefore, the behavior of such smart card is never adversarial.

Given that the UC security cannot be achieved without assuming some setup, two lines of research have been explored. One line of research investigated on the possibility of relaxing the need of trust in the setup assumptions required to achieve UC-security. For example, in the setup that we discussed above, the assumption was that each party receives a trusted smart-card. However, assuming that all the smart cards, even the ones used by the adversary, are trusted seems unrealistic, and it seems indeed natural to ask whether we can *reduce* the amount of trust needed to achieve UC-security. This line of research, started with [46], explores the use of hardware boxes (formally, tamper-proof hardware tokens) in the protocol. The idea is that, each party must trust only its own hardware box, and not the boxes generated/delivered by the other players.

The second line of research instead investigates on relaxing the UC-definition so that it is possible to design protocols without using any trusted setup. Such relaxed definitions capture a restricted scenario in which an adversary activates many executions of the *same* protocol simultaneously, playing always the *same* role (in contrast in the UC-definition the adversary can execute arbitrary protocols concurrently, and play different roles). We refer to this more relaxed definition as security in the concurrent setting, and to the adversary playing in this setting, as a concurrent adversary. Besides the mere feasibility results, this line of research develops on understanding the minimal requirements, such as round and communication complexity, computation complexity, black-box uses of cryptographic primitive, for a protocol which is secure in the concurrent setting.

Overview of the Results

In the thesis we provide contributions for both lines of research.

First, we discuss how to achieve UC-security based on physical setup assumptions while removing the trust in any third party, and the trust on the physical devices used by the adversary. We explore the use of Physically Uncloneable Functions (PUFs) as setup assumption for achieving UC-secure computations. PUF are physical noisy source of randomness. The use of PUFs in the UC-framework has been proposed already in [10]. However, [10] assumes that all PUFs in the system are *trusted*. This means that, each party has to trust the PUFs generated by the other parties. We focus on reducing the trust involved in the use of such PUFs and we introduce the Malicious PUFs model in which only PUFs generated by honest parties are assumed to be trusted. Thus the security of each party relies on its own PUF only and holds regardless of the goodness of the PUFs generated/used by the adversary. We are able to show that, under this more realistic assumption, one can achieve UC-secure computation, under computational assumptions. Moreover, we show how to achieve *unconditional* UC-secure commitments with (malicious) PUFs and with stateless tamper-proof hardware tokens. These results are discussed in more details in Section 2.

Secondly, we restrict our attention to the concurrent setting. We investigate on protocols which are concurrently secure and enjoy *round optimality* and *black-box* access to a cryptographic primitive. We study two fundamental functionalities: commitment scheme and zero knowledge, and we study two concurrent attack models, as explained below.

Commitment scheme is a two-stage (commitment, decommitment stage) functionality run between a committer and a receiver. The committer has a secret value in input, and it commits to such value running the commitment stage. When the committer is ready to reveal the value to the receiver, it runs the decommitment stage, also called opening stage. The security properties of a commitment scheme are hiding and binding. Hiding preserves the security of the committer, and is the property of the commitment stage. A commitment stage is hiding if the secrecy of the input committed is preserved against any adversarial receiver. Binding is a correctness property of the commitment scheme, and it requires that for a given transcript of the commitment stage, there exists only one value that can be revealed by any, possible adversarial, committer. We consider the following attack model for commitment schemes. An adversarial receiver can interleave arbitrarily many commitment sessions, and once the commitment stages are over, ask for the decommitment (the opening) of some of them, adaptively on the transcript observed from the sessions played so far and in any order. Note that this adversary is not purely concurrent, since it can ask to open a session only if the commitment stage of any other sessions is already completed. Namely, there is a barrier between the commitment phase and the openings. We refer to this type of concurrency as ”concurrency with barrier". This attack model is referred in literature as Selective Opening Attack (SOA, in short) and was introduced in [32]. A commitment scheme secure in this model is said to be SOA-secure.

A Zero Knowledge protocol is run between two parties, a prover and a verifier. Both parties have as common input an instance x of an **NP** language L. The prover has as input a witness w for the statement $x \in L$, and he wants to use knowledge of the witness to convince the verifier that the statement is true, however, he does not want to reveal any information about the witness. Security here means that any (concurrent) adversarial verifier running a protocol execution with the prover on input x should not gain any information besides the fact that $x \in L$. The correctness requirement establishes that if the statement is false, i.e., $x \notin L$, any adversarial (possibly concurrent) prover should not be able to convince a honest verifier of the truthfulness of the statement. In this thesis we consider the following attack model. An adversarial verifier can initiate any polynomial number of protocol sessions in concurrency, but it is bounded on the number of identities that it can play with. More precisely, we consider a setting in which any verifier that wishes to run the zero knowledge protocol with the prover, has to register its identity in a public file before any proof begins. Thus, there is a registration phase, in which no interaction between prover and verifier takes place, but each verifier upload its public identity in a common public file. After the registration has been completed, the “proof phase” starts. In this phase prover and registered verifiers run concurrently many proofs. The restriction is that the proof phase can be run only by registered verifiers. This model is called Bare Public Key (BPK, for short) model, and was introduced in [15].

Our results concern some of the round-optimal constructions and lower bounds presented in the literature for both SOA-secure commitments and zero-knowledge protocols in the BPK model. We show that some of the proposed constructions present subtle issues. Then we propose new protocols that meet the security guarantees promised by protocols proposed in literature. We describe these results in more details in Section 3.

Physical Attacks

So far we have discussed about network attacks. Namely, we have considered an adversary which exploits the network to run concurrent executions of different protocols. We now turn to physical attacks. Namely, we consider an adversary that can physically tamper with the machine of the honest party while running the protocol. In particular, in this thesis we consider a very specific attack in which the adversary is able to *reset* the memory of the machine of an honest party, in fact forcing the machine to run many executions of the same protocol *reusing* the same randomness. At first sight, this attack might seem too unrealistic, as one can imagine that the honest party can physically protect its own machine from the adversary. Therefore, considering such attack when designing a protocol, might seem an overkill. However, as nowadays tiny and weak computational devices are used in cryptographic protocols, such attack has become a real threat. Smart cards are the canonical example of tiny devices computing (sensitive) cryptographic protocol. Indeed, we use smart card everyday to perform bank transactions or for identification purposes. The way computations with smart card work, is that we put our smart card in some more powerful computing device, like a smart-card reader, or a PC, and they run a protocol. In this scenario resetting attacks seem very plausible, since a smart card is a very weak device compared to a reader or to a computer.

The security definition that formally captures this kind of attack has been introduced in [15]. This definition focuses only on the zero knowledge functionality (and the weaker notion of witness indistinguishability), and considers only the case in which only one party is reset. More precisely, the party which is subject to reset is always the prover. Thus, the constructions shown in [15] are secure if the prover is subject to reset attack, but is not secure if the verifier is instead the victim of reset. Later in [6] resettable security has been formulated for the opposite case. Namely, [6] provides constructions that are secure if the party that is subject to reset is only the verifier. Clearly, an interesting question is whether one can design a protocol in which the party that might be subject to reset attack is not known in advance. Therefore, such protocol should be secure in both cases. Such notion is called security under *simultaneous resettability*. The work of [22] provides a protocol that is secure in such setting. Namely, it presents a construction for simultaneously resettable zero knowledge system. However this result does not close the gap between what we are able to achieve in presence of a fixed resetting party, and what we can achieve when both parties can reset. Indeed, in case of one-side resetting (either only the prover can reset, or only the verifier can reset), we know how to achieve protocols that are *arguments of knowledge*. Roughly, a protocol is an argument of knowledge, if the prover can convince the verifier of a truthfulness of a statement, if and only if it knows the witness.

In the thesis we provide the first construction of a witness indistinguishable argument system that is *simultaneous resettable* and *argument of knowledge*. We discuss about this contribution in Section 4.

2 UC-security from Malicious PUFs

In this section we discuss our results in the Universal Composability framework. Such results can be found in the articles [59] and [82]. In the following, we first discuss the state of the art of Universally Composable secure computation and then we outline our contribution.

Universal Composability using Physical Assumptions. The universal composability framework was introduced by Canetti in [14] and captures the most general security notion considering an adversary that run many concurrent execution of arbitrary protocols. Such general security notion is, unfortunately, impossible to achieve in the plain model, as it was first proved in Canetti and Fischlin [14] and then strengthened by Canetti et al. in [17]. As a consequence, several setup assumptions, and relaxations of the UC framework have been proposed to achieve UC security [18, 5, 64, 45].

In recent years, researchers have started exploring the use of secure hardware in protocol design. The idea is to achieve protocols with strong security guarantees (like UC) by allowing parties to use hardware boxes that have certain security properties. An example of the kind of security required from such a hardware box is that of *tamper-proofness*; i.e., the receiver of the box can only observe the input/output behaviour of the functionality that the box implements. This property was formalized by Katz in [46], and it was shown that UC security is possible by relying on the existence of tamper-proof programmable hardware tokens, and computational assumptions. Smart cards are well understood examples of such tokens, since they have been used in practice in the last decades. Several improvements and variations of Katz's model have been then proposed in follow up papers (e.g., [19, 56, 38, 40, 29, 21, 30]).

Spurred by technological advances in manufacturing, recently a new hardware component has gained a lot of attention: Physically Uncloneable Functions (PUFs) [61, 60]. A PUF is a hardware device generated through a special physical process that implements a "random" function[1] that depends upon the physical parameters of the process. These parameters can not be "controlled", and producing a clone of the device is considered infeasible. Once a PUF has been constructed, there is a physical procedure to query it, and to measure its answers. The answer of a PUF depends on the physical behavior of the PUF itself, and it is assumed to be unpredictable, or to have high min-entropy. Namely, even after obtaining many challenge-response pairs, it is infeasible to predict the response to a new challenge.

Since their introduction by Pappu in 2001, PUFs have gained a lot of attention for cryptographic applications like anti-counterfeiting mechanisms, secure storage, RFID applications, identification and authentication protocols [71, 43, 69, 34, 50]. More recently PUFs have been used for designing more advanced cryptographic primitives. In [67] Rührmair shows the first construction of Oblivious Transfer, the security proof of which is later provided in [68]. In [4], Armknecht et al. deploy PUFs for the construction of memory leakage-resilient encryption schemes. In [51] Maes et al. provide construction and implementation of PUFKY, a design for PUF-based cryptographic key generators. There exist several implementations of PUFs, often exhibiting different properties. The work of Armknecht et al. [3] formalizes the security features of physical functions in accordance to existing literature on PUFs and proposes a general security framework for physical functions. A survey on PUF implementations is given in [52]. Very recently in [47] Katzenbeisser et al. presented the first large scale evaluation of the security properties of some popular PUFs implementations (i.e., intrinsic electronic PUFs).

Modeling PUFs in the UC framework. Only very recently, Brzuska et al. [10] suggested a model for using PUFs in the UC setting that aims at abstracting real-world implementations. The unpredictability and uncloneability properties are modeled through an ideal functionality. Such functionality allows only the creation of trusted PUFs. In [10] PUFs are thought as non-PPT setup assumptions. As such, a PPT simulator cannot simulate a PUF, that is, PUFs are non-

[1] Technically, a PUF does not implement a function in the mathematical sense, as the same input might produce different responses.

programmable. Although non-programmable, PUFs are not modeled as global setup [13]. [10] shows how to achieve unconditional UC secure Oblivious Transfer, Bit Commitment and Key Agreement with trusted PUFs.

PUFs vs tamper-proof hardware tokens. The apparent similarity of PUFs with programmable tamper-proof tokens [46] vanishes immediately when one compares in detail the two physical devices. Indeed, PUFs are non programmable and thus provide unpredictability only. Instead tokens are programmable and can run sophisticated code. Moreover, PUFs are stateless, while tokens can be stateful. When a PUF is not physically available, it is not possible to know the output of new queries it received. Instead the answer of a stateless token to a query is always known to its creator[2], since it knows the program embedded in the token. Tamper-proof tokens are realized through ad-hoc procedures that model them as black boxes, their internal content is protected from physical attacks and thus the functionalities that they implement can be accessed only through the prescribed input/output interface provided by the token designer. Instead, PUFs do not necessarily require such a hardware protection and their design is associated to recommended procedures to generate and query a PUF, guaranteeing uncloneability and unpredictability. Finally, in contrast to tokens that correspond to PPT machines, PUFs are not simulatable since it is not clear if one can produce an (even computationally) indistinguishable distribution.

2.1 Our contribution

We continue the line of research started by Brzuska et al. investigating more on the usability of PUFs to obtain UC-secure computation. We observe that the UC formulation of PUFs proposed by Brzuska et al. considers only *trusted* PUFs. This means that, it is assumed that an adversary is be unable to produce fake/malicious PUFs. We believe that making such assumption might be unrealistic. Given that the study of PUFs is still in its infancy, it is risky to rely on assumptions on the impossibility of the adversaries in generating PUFs adversarially.

Our main contribution consists in studying the security of protocols in presence of adversaries that can create malicious PUFs. We present a modification of the model of Brzuska et al. that formalizes security with respect to such stronger adversary and we give positive answers to the question of achieving universally composable secure computation with PUFs. More in details, our contributions are listed below.

[2] This is true for stateful tokens too, provided that one knows the sequence of inputs received by the token.

The Malicious PUFs Model. We generalize the model of Brzuska et al. so to enable the adversary to create untrusted (malicious) PUFs. But what exactly are malicious PUFs? In real life, an adversary could tamper with a PUF in such a way that the PUF loses any of its security properties. Or the adversary may introduce new behaviours; for example, the PUF could keep a state. To keep the treatment of malicious behaviour as general as possible, we allow the adversary to send as PUF any hardware token that meets the syntactical requirements of a PUF. Thus, an adversary is assumed to be able to even produce fake PUFs that might be stateful and programmed with malicious code. We assume that a malicious PUF however cannot interact with its creator once is sent away to another party. If this was not the case, then we are back in the standard model, where UC security is impossible to achieve has argued below.

The impossibility is straight forward. Consider any functionality that protects the privacy of the input of a player P_1. Comparing to the plain model (where UC is impossible), the only advantage of the simulator to extract the input of the real-world adversary P_1^*, is to read the challenge/answer pairs generated by P_1^* when using the honest PUF created by the simulator that plays on behalf of P_2. If such a simulator exists, then an adversary P_2^* can generate a malicious PUF that just plays as proxy and forwards back and forth what P_2^* wishes to play. P_2^* can locally use one more honest PUF in order to compute the answers that the (remote) malicious PUF is supposed to give. Clearly P_2^* will have a full view of all challenge/response pairs generated by honest P_1 and running the simulator's code, P_2^* will extract the input of P_1, therefore contradicting input privacy.

General UC-secure computation in the Malicious PUFs Model. The natural question is whether UC security can be achieved in such a much more hostile setting. We give a positive answer to this question by constructing a *computational* UC-secure commitment scheme in the malicious PUFs model. Our commitment scheme needs two PUFs that are transferred only once (PUFs do not go back-and-forth), at the beginning of the protocol and it requires computational assumptions. We avoid that PUFs go back-and-forth by employing a technique that requires OT. The results of Canetti, et al. [18] shows how to achieve general UC computation from computational UC commitments.

Hardness assumptions with PUFs. Notice that as correctly observed in [10], since PUFs are not PPT machines, it is not clear if standard complexity-theoretic assumptions still hold in presence of PUFs. We agree with this observation. However the critical point is that even though there can exist a PUF that helps to break in polynomial time a standard complexity-theoretic assumptions, it is still unlikely that a PPT adversary can find such a PUF. Indeed a PPT machine can only generate a polynomial number of PUFs, therefore obtaining the one that allows to break complexity assumptions is an event that happens with negligible probability and thus it does not effect the concrete security of the protocols. In light of the above discussion, only one of the following two cases is possible. 1) Standard complexity-theoretic assumptions still hold in presence of PPT adversaries that generate PUFs; in this case our construction is secure. 2) There exists a PPT adversary that can generate a PUF that breaks standard assumptions; in this case our construction is not secure, but the whole foundations of complexity-theoretic cryptography would fall down (which is quite unlikely to happen) with respect to real-world adversaries.

Unconditional UC-secure Commitment Scheme with Malicious PUFs. We furthermore provide a tool for constructing UC-secure commitments given any straight-line extractable commitment. This tool allows us to prove feasibility of unconditional UC-secure protocols (for a non-trivial functionality) in the malicious PUF model. More precisely, we provide a compiler that transforms any ideal extractable commitment – a primitive that we define – into a UC-secure commitment. An ideal extractable commitment is a statistically hiding, statistically binding and straight-line extractable commitment. The transformation uses the ideal extractable commitment as black-box and is unconditional. The key advantage of such compiler is that, one can implement the ideal extractable commitment with the setup assumption that is more suitable with the application and the technology available.

We then construct an extractable commitment scheme in the malicious PUFs model. By plugging such scheme into the compiler we obtain the first unconditional UC-secure commitment with malicious PUFs. However, whether *general* secure computation with unconditional UC security is possible with (malicious) PUFs, remains an interesting open question.

3 Round-optimal Concurrently Secure Protocols

In this section we discuss our results concerning achieving round-optimal protocols for concurrent Zero Knowledge and Commitment Scheme secure under Selective Opening Attacks in the concurrent setting. Such results can be found in the articles [70] and [58] respectively. In the following, we first discuss the state of the art of the respective problems and then we outline our contribution.

3.1 Round-Optimal Concurrent ZK in the Bare Public Key model

The notion of concurrent zero knowledge (*cZK*, for short) introduced in [33] deals with proofs given in asynchronous networks controlled by the adversary.

In [15] Canetti et al. studied the case of an adversary that can reset the prover, forcing it to re-use the same randomness in different executions. They defined as resettable zero knowledge ($\mathsf{r}ZK$, for short) the security of a proof system against such attacks. Very interestingly, $\mathsf{r}ZK$ is proved to be stronger than *cZK*.

Motivated by the need of achieving round-efficient $\mathsf{r}ZK$, in [15] the Bare Public-Key (BPK, for short) model has been introduced, with the goal of relying on a setup assumption that is as close as possible to the standard model. Indeed, round-efficient *cZK* and $\mathsf{r}ZK$ are often easy to achieve in other models (e.g., with trusted parameters) that unfortunately are hard to justify in practice.

The BPK model. The sole assumption of the BPK model is that when proofs are played, identities of (polynomially many) verifiers interacting with honest provers are fixed. For instance, identities could be posted to a public directory so that players can download the content of the directory before proofs start. This registration phase is non-interactive, does not involve trusted parties or other assumptions, and can be fully controlled by any adversary. When proofs start, it is assumed that honest provers interact with registered verifiers only.

The BPK model is very close to the standard model, indeed the proof phase does not have any requirement beyond the availability of the directory to all provers, and for each verifier, of a secret key associated to his identity. Moreover, in both phases the adversary has full control of the communication network, and of corrupted players.

Round-optimal *cZK* in the BPK model from $\mathsf{r}ZK$. The first constant-round $\mathsf{r}ZK$ (and thus *cZK*) argument for **NP** in the BPK model has been given in [15]. Then in [54] it is pointed out the subtle separations among soundness notions in the BPK model. Indeed, in contrast to the standard model, the notions of one-time, sequential and concurrent soundness, are distinct in the BPK model. In [54] it is then showed that the proof of [15] is actually sufficient for sequential soundness only. Moreover in [54] it is proved that 4 rounds are necessary for concurrent soundness and finally, they showed a 4-round $\mathsf{r}ZK$ (and thus *cZK*) argument with sequential soundness. The protocol is "conversation based", i.e., by simply observing the transcript one can compute the output of the verifier. In light of the impossibility proved by [1] (i.e., there exists no 3 round sequentially sound *cZK* conversation-based argument in the BPK model for non-trivial languages) the above 4-round rZK (and thus cZK) argument is round optimal.

Concurrent soundness along with rZK (and thus cZK) was achieved in [24], requiring 4 rounds. Further improvements on the required complexity assumptions have been showed in [79] where a 4-round protocol under generic assumptions and an efficient 5-round protocol under number-theoretic assumptions are shown.

The above mentioned results on constant-round rZK/cZK in the BPK model rely on the assumptions that some cryptographic primitives are secure against sub-exponential time adversaries (i.e., complexity leveraging) and obtained black-box simulation.

Round-optimal cZK in the BPK model under standard assumptions. The question of achieving a constant-round black-box cZK in the BPK model without relying on complexity leveraging has been first addressed in [80] and then in [26]. The protocol of [80] needs 4 rounds and achieves sequential soundness only. The protocol given in [26] also needs 4 rounds and achieves concurrent soundness. A follow up result of [72] showed an efficient transformation that starting from a language admitting a Σ-protocol produces a cZK argument with concurrent soundness needing only 4 rounds and adding only a constant number of modular exponentiations. A more recent result [23] obtains both round optimality and optimal complexity assumptions (i.e., the need of One-way Functions only) for concurrently sound cZK. The notion of "knowledge extraction" has been studied in [27] and in [77, 76] where in particular concurrent knowledge extraction (under different formulations) is considered.

All above results achieve cZK and are based on hardness assumptions with respect to polynomial-time adversaries.

3.1.1 Our Contribution

In the thesis we show subtle problems concerning security proofs of various cZK and rZK arguments in the BPK model [54, 80, 24, 26, 72, 79, 23, 77], including *all* round-optimal constructions published so far.

The source of the problem: parallel execution of different sub-protocols. In order to achieve round efficiency, various known protocols, including all round-optimal protocols, consist in parallel executions of sub-protocols that are useful in different ways in the proofs of soundness and $cZK/$rZK. Roughly speaking, there is always a sub-protocol π_0 where in 3 rounds the verifier is required to use a secret related to its identity. Then there is a 3-round sub-protocol π_1 in which the prover convinces the verifier about the validity of the statement and the simulator can do the same by using knowledge of a secret information obtained by rewinding π_0

(in the current session or in other sessions corresponding to the same identity). To obtain a 4-round protocol[3], π_1 starts during the second round of π_0. Such round combination yields one of the following two cases.

The first case is when the simulator needs the secret information already to compute the first message of π_1 so that such a message can appear in the final transcript of the simulation. In this case when the simulator runs protocol π_1 for the first time with a given identity, it first needs to extract the secret related to such identity, used in π_0 by the verifier. The use of look-ahead threads (i.e., trying to go ahead with a virtual simulation with the purpose of obtaining the required information needed in the main thread of the simulation) would not help here since only a limited polynomial amount of work can be invested for them, and there is always a non-negligible probability that look-ahead threads fail, while still in the main thread the verifier plays the next message. Given the above difficulty, the simulator needs to play a *bad* first round in π_1 so that later, when the needed secret information is extracted from π_0, the simulator can play again such first round of π_1, this time with a *good* message. However, this approach suffers of a problem too. Indeed, aborting the main thread and starting a new thread leads to a detectable deviation in the final transcript that the simulator will output. Indeed, the fact that the simulator gives up with a thread each time it is stuck, and then starts a new one, skews the distribution of the output of the simulator, since the output will then include with higher probability threads that are "easier" to complete (e.g., where the simulator does not get stuck because new sessions for new identities do not appear). Notice that this issue motivates the simulation strategies adopted in previous work on *cZK* (e.g., [66, 63]) where the main thread corresponds to the construction of the view that will be given in output, while other threads are started with the sole purpose of extracting secrets useful to complete the main thread.

We now consider the second case where the simulator does not need any secret to compute the first round of π_1. We observe that this approach could hurt the proof of concurrent soundness, when the latter is proved by means of witness extraction[4] from π_1. Indeed, a malicious concurrent prover can exploit the execution of π_0 in a session j, for completing the execution of π_1 in another concurrent session $j' \neq j$ by playing a man-in-the-middle attack such that, when (in the proof of concurrent soundness) one tries to reach a contradiction by extracting the witness from the proof π_1 given in session j', it instead obtains the secret used to run π_0 in session j. Instead, if the secret to be extracted from π_1 is fixed from the very first round of π_1, then one can show that it is either independent from the one used in session

[3] Similar discussions hold for some 5-round protocols when π_0 requires 4 rounds.

[4] We note that all constructions of *cZK* in the BPK model under standard assumptions prove soundness by means of witness extraction.

j (this happens when the secret is used in π_0 of session j after the first round of π_1 in session j' is played, and the secret used by the verifier can not be predicted with non-negligible probability), or is dependent but not affected by the rewind of session j' (this happens when the secret is used in π_0 of session j before the first round of π_1 in session j' is played).

The use of the secret in the last round of π_1 only, could instead be helpful in the following three cases: I) when one is interested in r*ZK* since in this case soundness is proved through a reduction based on complexity leveraging (no need for rewinding); II) when *cZK* with sequential soundness only is desired; III) when the secret needed by the simulator when running π_1 in a session j' is different from the witness used by the verifier in the execution of π_0 in other sessions. In these three cases the above discussion does not necessarily apply. Indeed some proposed round-optimal protocols that fall in one of such cases, might still be secure even though their security proofs seem to ignore at least in part the problems that we are pointing out.

Because of the above case I), we believe that achieving 4-round *cZK* with concurrent soundness in the BPK model under standard assumptions is definitively harder than obtaining 4-round r*ZK* with concurrent soundness in the BPK model through complexity leveraging. Therefore, in this thesis we focus on achieving *cZK* and this will require a new technique.

We stress that in all previous constructions, one could obtain a different protocol that satisfies the desired soundness and zero-knowledge properties by simply running π_0 and π_1 sequentially. Indeed, in this case the simulator can complete π_0 in the main thread, then run the extractor in another thread, and finally continue the main thread running π_1 having the secret information. We also stress that all papers that we revisit in the thesis, achieved also other results that are not affected by our analysis when round optimality is not desired.

We finally note that we did not investigate other round-efficient results in variations of the BPK model [53, 81, 25], and other results in the BPK model that do not focus on optimal round complexity [57, 78, 20].

New techniques for round-optimal *cZK* in the BPK model. In this thesis we show a protocol and a security proof that close the gap in between lower and upper bounds for the round complexity of concurrently sound *cZK* in the BPK model under standard assumptions. The result is achieved by using a new technique where in addition to the (permanent) secret associated to the identity of the verifier, there is a *temporary* secret per session, that enables the simulator to proceed in two modes as follows. Knowledge of the permanent secret of the verifier allows the simulator to proceed in straight-line in the main thread in sessions started after the extraction of the permanent secret. Knowledge of the temporary secret allows the simulator to solve the sessions started before the extraction of the permanent secret, by launching rewinding threads but without changing the main thread. The temporary key is extracted through rewinding threads, and it is used only when computing the last prover message of a session of the main thread, i.e., only after the extraction has been completed. This allows to keep the main thread unchanged. In the rewinding threads the simulator is always straight-line. We implement both the permanent and the temporary keys by means of trapdoor commitments. The proof of *cZK* will be tricky since it requires the synergy of the two above simulation modes. In our case the number of extraction procedures required to carry on the simulation is not bounded by the number of identities registered in the directory (in contrast with the main technique used in the past in the BPK model), but by the number of sessions. The proof of concurrent soundness also requires special attention. Indeed while the interplay of temporary and permanent secrets helps the simulator, it could also be exploited by the malicious prover.

Finally, we show that our *cZK* protocol admits a transformation that starting from any language admitting a perfect Σ-protocol, produces round-optimal concurrently-sound *cZK* protocol. Such transformation requires a constant number of modular exponentiations only, and the final protocol is secure under standard number-theoretic assumptions.

3.2 Round-Optimal SOA-secure Commitments

Commitment schemes are a fundamental building block in cryptographic protocols. By their usual notion, they satisfy two security properties, namely, hiding and binding. While the binding property guarantees that a committed message can not be opened to two distinct messages, the hiding property ensures that before the decommitment phase begins, no information about the committed message is revealed. Binding and hiding are preserved under concurrent composition, in the sense that even a concurrent malicious sender will not be able to open a committed message in two ways, and even a concurrent malicious receiver will not be able to detect any relevant information about committed messages as long as only commitment phases have been played so far.

In [32], Dwork et al. pointed out a more subtle definition of security for hiding where the malicious receiver is allowed to ask for the opening of only some of the committed messages, with the goal of breaking the hiding property of the remaining committed messages. This notion was captured in [32] via a simulation-based security definition, and is referred to as hiding in presence of selective opening attack (SOA, for short). [32] shows that, in a *trusted setup* setting, it is possible to construct a non-interactive SOA-secure commitment scheme from a trapdoor commitment scheme. Indeed, in the trusted setup the simulator sets the parameters of the trapdoor commitment, thus obviously it knows the trapdoor. However, the fundamental question of whether there exist SOA-secure commitment schemes in the *plain model*, is left open in [32]. We stress that the question is particularly important since commitments are often used in larger protocols, where often only some commitments are opened but the security of the whole scheme still relies on hiding the unopened commitments. For instance, the importance of SOA-secure commitments for constructing zero-knowledge sets is discussed in [36][5].

The SOA-security experiment put forth in [32] considers a one-shot commitment phase, in which the receiver gets all commitments in one-shot, picks adaptively a subset of them, and obtains the opening of such subset. Such definition implicitly considers non-interactive commitments and only parallel composition. Subsequent works have explored several extensions/variations of this definition showing possibility and impossibility results. Before proceeding to the discussion of the related work, it is useful to set up the dimensions that will be considered. One dimension is *composition*. As commitment is a two-phase functionality, other than parallel composition, one can consider two kinds of concurrent composition. Concurrent-with-barrier composition (considered in [9, 44]), refers to the setting in which the adversarial receiver can interleave the execution of several commitments, and the execution of decommitments, with the restriction that all commitment phases are played before any decommitment phase begins. Thus, there is a barrier between commitment and decommitment stage. Fully-concurrent composition (considered in [73]) refers to the setting in which the adversary can arbitrarily interleave the execution of the commitment phase of one session with the decommitment of another session (and vice-versa).

Next dimension is the *access to primitive*, namely, if the construction uses a cryptographic primitive as a black-box (in short, BB), or in a non black-box way (in short, NBB).

Another dimension is *simulation*. In this discussion we consider always black-box simulators (if not otherwise specified).

The question of achieving SOA-secure commitments without any set-up was solved affirmatively in [9] by Bellare, Hofheinz, and Yilek, and in Hofheinz [44], who presented an interactive SOA-secure scheme based on non-black-box use of any one-way permutation and with a commitment phase requiring a super-constant number of rounds. The security of such construction is proved in the concurrent-with-barrier setting. [9, 44] also shows that non-interactive SOA-secure commitments which use cryptographic primitives in a black-box way do not exist. The same work introduces the notion of *indistinguishability* under selective opening attacks, that we do not consider in the thesis. The results of [9, 44]

[5] In [36] some forms of zero-knowledge sets were proposed, and their strongest definition required SOA-secure commitments.

left open several other questions on round optimality and black-box use of the underlying cryptographic primitives.

In TCC 2011 [73], Xiao addressed the above open questions and investigated on how to achieve nearly optimal schemes where optimality concerns both the round complexity and the black-box use of cryptographic primitives. In particular, Xiao addressed SOA-security of commitment schemes for both parallel composition and fully-concurrent composition and provided both possibility and impossibility results, sticking to the simulation-based definition. Concerning positive results, [73] shows a 4-round (resp., $(t + 3)$-round for a t-round statistically-hiding commitment) computationally binding (resp., statistically binding) SOA-secure scheme for parallel composition. Moreover, [73] provides a commitment scheme which is "strong" (the meaning of strong is explained later) SOA-secure in the fully-concurrent setting and requires a logarithmic number of rounds. All such constructions are fully black-box. Concerning impossibility results, [73] shows that 3-round (resp., 4-round) computationally binding (resp., statistically binding) parallel SOA-secure commitment schemes are impossible to achieve. As explained later, in our work we present some issues affecting the proof of security of the constructions shown in [73]. We also show that, the strong security claimed for the construction suggested for the fully-concurrent setting, is actually impossible to achieve, regardless of the round complexity. We contradict the lower bounds claimed in [73] by providing a 3-round fully black-box commitment scheme which is SOA-secure under concurrent-with-barrier composition, which implies parallel composition.

In a subsequent work [75] Xiao showed a black-box construction of 4-round statistically binding commitment which is SOA-secure under parallel composition.

The work [74] provides an updated version of [73]. Concerning positive results, [74] includes the $(t + 3, 1)$-round construction of [73] and shows a new simulation strategy for it. Concerning impossibility results, [74] includes the lower bounds of [73] that are still valid for 2-round (resp., 3 round) computationally hiding and computationally (resp., statistically) binding, parallel SOA-secure commitment scheme with black-box simulators. [74] contains also other contributions of [73] that are not contradicted by our results.

In [7], Bellare et al. proves that existence of CRHFs implies impossibility of non-interactive SOA-secure commitments (regardless of the black-box use of the cryptographic primitives). In fact, they show something even stronger; they show that this impossibility holds even if the simulator is non-black-box and knows the distribution of the message space. An implication of such results is that, standard security does not imply SOA-security. Previous results in [9, 44] only showed the impossibility for the case of black-box reductions.

In [62], Pass and Wee provide several black-box constructions for two-party protocols. Among other things, they provide constructions for look-ahead trapdoor commitments (in a look-ahead commitment, knowledge of the trapdoor is necessary already in the commitment phase in order for the commitment to be equivocal), and trapdoor commitments. Such constructions have not been proved to be SOA-secure commitment schemes, as SOA-security is proven in presence of (at least) parallel composition, while security of the trapdoor commitment of [62] is proved only in the stand-alone setting.

3.2.1 Our Contribution

We focus on simulation-based SOA-secure commitment schemes, and we restrict our attention to black-box simulation, and (mainly) black-box access to cryptographic primitives (like in [73]). Firstly, we point out various issues in the claims of [73]. These issues essentially re-open some of the open questions that were claimed to be answered in [73]. We next show how to solve (in many cases in a nearly optimal way) all of them. Interestingly, our final claims render quite a different state-of-the-art from (and in some cases also in contrast to) the state-of-the-art set by the claims of [73].

In detail, by specifying as (x, y) the round complexity of a commitment scheme when the commitment phase takes x rounds and the decommitment phase takes y rounds, we revisit some claims of [73] and re-open some challenging open questions as follows.

1. The proof in [73] of the non-existence of $(3, 1)$-round schemes assumes implicitly that the sender sends the last message during the commitment phase. We show that surprisingly this assumption is erroneous, and that one round might be saved in the commitment phase if the receiver goes last. This re-opens the question of the achievability of $(3, 1)$-round SOA-secure schemes, even for just parallel composition.

2. The proof of binding and SOA-security of the $(4, 1)$-round scheme of [73] for parallel composition presents a sublet issue, and it is currently unknown whether the scheme is secure. The same issue in the SOA-security proof exists for the $(t + 3, 1)$-round statistically binding scheme of [73] which is based on any t-round statistically-hiding commitment. Indeed, for both constructions, SOA-security is claimed to follow from the simulation technique of Goldreich-Kahan [37]. The problem is that the simulator of [37] was built for a *stand-alone* zero-knowledge protocol where an atomic sub-protocol is repeated several times in parallel, and the verifier cannot *selectively* abort one of the sub-protocols. Instead in the SOA-setting the adversarial receiver interacts with multiple senders and can decide to abort only a subset of the sessions of its choice adaptively based on the commitment-phase transcript.

3. The proof of security of the fully-concurrent SOA-secure commitment proposed in [73] presents a subtle issue. The security of such construction is claimed even for the case in which the simulator cannot efficiently sample from the distribution of messages committed to by the honest sender (but needs to query an external party for it). This notion is referred in [73] as "strong" security. This issue in [73] re-opens the possibility of achieving schemes that are strong SOA-secure under fully concurrent composition (for any round complexity).

In the thesis we solve the above open problems (still sticking to the notion of black-box simulation as formalized in [73]) as follows.

1. We present a $(3, 1)$-round scheme based on BB use of any trapdoor commitment (TCom, for short), which contradicts the lower bound claimed in [73].

   We also show a $(4, 1)$-round scheme based on BB use of any weak trapdoor commitment (wTCom, for short)[6].

2. We show that when the simulator does not know the distribution of the messages committed to by the honest sender, there exists no scheme that achieves fully concurrent SOA-security, regardless of the round complexity and of the BB use of cryptographic primitives. Thus contradicting the claimed security of the construction given in [73].

3. As a corollary of our $(3, 1)$-round scheme based on BB use of any TCom, there exists a $(3, 1)$-round scheme based on NBB use of any one-way function (OWF). This improves the round complexity in [9] from logarithmic in the security parameter to only 3 rounds and using minimal complexity-theoretic assumptions. Moreover, we observe that (as a direct consequence from proof techniques in [73]) a $(2, 1)$-round SOA-secure scheme is impossible regardless of the use of the underlying cryptographic primitive (for black-box simulation only). Thus, our $(3, 1)$-round scheme for black-box simulation is essentially round-optimal.

Notice that both our $(3, 1)$-round protocols – the one based on BB use of TCom and the other based on NBB use of OWFs – contradict the impossibility given in [73], that was claimed to hold regardless of the access to the cryptographic primitives.

All the constructions that we present are secure under concurrent-with-barrier composition, which obviously implies parallel composition. Our simulators work

[6] This result indeed requires a relaxed definition of trapdoor commitment where the trapdoor is required to be known already during the commitment phase in order to later equivocate. We call it "weak" because any TCom is also a wTCom.

for any message distributions, and do not need to know the distribution of the messages committed to by the honest sender. In light of our impossibility for the fully concurrent composition (see Item 2 of the above list), the concurrency achieved by our schemes seems to be optimal for this setting.

4 Simultaneously Resettable Arguments of Knowledge

Interaction and private randomness are the two fundamental ingredients in Cryptography. They are crucial for achieving zero-knowledge proofs [39]. In [15] Canetti, Goldreich, Goldwasser and Micali showed that when private randomness is limited and re-used in multiple instances of a proof system, it is still possible to preserve the zero-knowledge requirement. The setting proposed by [15] is of a malicious verifier that resets the prover, therefore forcing the prover to run several protocol executions using the same randomness. This setting applies to protocols where the prover is implemented by a stateless device. Therefore, a prover can only count on the limited (hardwired) randomness while it can be adaptively reset any polynomial number of times. The resulting security notion against such powerful verifiers is referred to as *resettable zero knowledge* ($\mathsf{r}ZK$) and is provably harder to achieve than concurrent zero knowledge. Feasibility results have been achieved in [15, 49] in the standard model with the following round-complexity: polylogarithmic for $\mathsf{r}ZK$ and constant for resettable witness indistinguishability ($\mathsf{r}WI$, in short). Since then, it was also shown how to achieve resettable zero knowledge in the Bare Public-Key (BPK) model, introduced by Canetti et al. [15], where one can obtain better round complexity and assumptions [54, 24, 1, 79]. Very recently, it has been shown [35] that resettable statistical zero knowledge for non-trivial languages is possible.

The "reverse" of the above question has been considered by Barak, Goldreich, Goldwasser and Lindell [6] where a malicious prover resets a verifier, called *resettable soundness*. In [6], it has been shown how to obtain resettable soundness along with ZK in a constant number of rounds.

Barak et al. [6] proposed the challenging *simultaneous resettability conjecture*, where one would like to prove that a protocol is secure against both a resetting malicious prover and a resetting malicious verifier. The existing machinery turned out to be insufficient, and a definitive answer required almost a decade. In the work of Deng, Goyal and Sahai [22] they showed a resettably sound $\mathsf{r}ZK$ argument for **NP** with polynomial round complexity. Very recently, results in the BPK model for simultaneous resettability have been obtained in [78, 2] with a constant number of rounds.

Arguments of knowledge under simultaneous resettability. Argument systems are often used with a different goal than proving membership of an instance in a language. Indeed, it is commonly required to prove knowledge (possession) of a witness instead of the truthfulness of a statement. Since arguments of knowledge serve as major building blocks in Cryptography (e.g., in identification schemes[7]), it is an interesting question whether the previous results for arguments of membership extend to arguments of knowledge. Unfortunately, arguments of knowledge have been achieved so far only when one party can reset. That is, we have $\mathsf{r}ZK$ arguments of knowledge [15] and, separately, resettably sound ZK arguments of knowledge [6]. Instead, when reset attacks are possible in both directions, no result is known even when only $\mathsf{r}WI$ with resettable argument of knowledge is desired.

In [31] Dwork and Naor present ZAPs, which are simultaneously resettable WI proofs. It is important to note that resettable security for ZAPs comes almost for free because of the minimal round complexity (1 or 2 rounds). However, it is not known how to accommodate for knowledge extraction, unless one relies on non-standard (e.g., non-falsifiable) assumptions. For the case of resettably sound $\mathsf{r}ZK$, all the above results [22, 78, 2] critically use an instance-dependent technique along with ZAPs: when the statement is true (i.e., when proving $\mathsf{r}ZK$), the prover/simulator can run ZAPs which allow the use of multiple witnesses. Such use of multiple witnesses gives some flexibility that turns out to be very useful to prove resettable zero knowledge. Instead, when the statement is false, the protocols are designed so that adversarial malicious prover must stick with some fixed messages during the execution of protocol. Therefore, rewinding capabilities do not help the resetting malicious prover since he can not change those fixed messages. This is critically used in the proofs of resettable soundness in order to reach a contradiction when a prover proves a false statement. It is easy to see that the above approach fails when arguments of knowledge are considered. Indeed, when the malicious resetting prover proves a true statement, the same freedom that allows one to prove $\mathsf{r}ZK/\mathsf{r}WI$, also gives extra power to the malicious prover. Consequently, designing an extractor appears problematic and new techniques seem to be needed so that the simultaneous resettability conjecture is resolved even when we consider knowledge extraction.

Our main result is the first construction of a constant-round simultaneously resettable witness-indistinguishable argument of knowledge for any **NP** language. Our protocol is based on the novel use of ZAPs and resettably sound zero knowledge arguments, which improves over the techniques previously used in [22, 78] as well as concurrent and independent work[8] of [42].

[7] Bellare et al. in [8] gave various definitions for identification schemes when the adversary can also reset the proving device.

[8] In an independent work [42], Goyal and Maji achieved simultaneously resettable secure com-

As application of our main protocol, we also consider the question of secure identification under simultaneous resettability and show how to use the above simultaneous resettable WI argument of knowledge to obtain the first simultaneously resettable identification scheme which follows the knowledge extraction paradigm.

Acknowledgment

The results presented in this survey are contained in the phD thesis: ”Secure computation under concurrent and physical attacks", supervised by Prof. Ivan Visconti.

References

[1] Joël Alwen, Giuseppe Persiano, and Ivan Visconti. Impossibility and feasibility results for zero knowledge with public keys. In *Advances in Cryptology – Crypto ’05*, volume 3621 of *Lecture Notes in Computer Science*, pages 135–151. Springer Verlag, 2005.

[2] Seiko Arita. A constant-round resettably-sound resettable zero-knowledge argument in the bpk model. Cryptology ePrint Archive, Report 2011/404, 2011. `http://eprint.iacr.org/`.

[3] Frederik Armknecht, Roel Maes, Ahmad-Reza Sadeghi, François-Xavier Standaert, and Christian Wachsmann. A formalization of the security features of physical functions. In *IEEE Symposium on Security and Privacy*, pages 397–412. IEEE Computer Society, 2011.

[4] Frederik Armknecht, Roel Maes, Ahmad-Reza Sadeghi, Berk Sunar, and Pim Tuyls. Memory leakage-resilient encryption based on physically unclonable functions. In Mitsuru Matsui, editor, *ASIACRYPT*, volume 5912 of *Lecture Notes in Computer Science*, pages 685–702. Springer, 2009.

[5] Boaz Barak, Ron Canetti, Jesper B. Nielsen, and Rafael Pass. Universally composable protocols with relaxed set-up assumptions. In *Foundations of Computer Science (FOCS’04)*, pages 394–403, 2004.

[6] Boaz Barak, Oded Goldreich, Shafi Goldwasser, and Yehuda Lindell. Resettably-sound zero-knowledge and its applications. In *FOCS*, pages 116–125, 2001.

[7] Mihir Bellare, Rafael Dowsley, Brent Waters, and Scott Yilek. Standard security does not imply security against selective-opening. In David Pointcheval and

putation. Their work achieves (with simulation-based security) simultaneous resettability with polynomial round complexity assuming also the existence of lossy trapdoor encryption.

Thomas Johansson, editors, *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 645–662. Springer, 2012.

[8] Mihir Bellare, Marc Fischlin, Shafi Goldwasser, and Silvio Micali. Identification protocols secure against reset attacks. In Birgit Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 495–511. Springer, 2001.

[9] Mihir Bellare, Dennis Hofheinz, and Scott Yilek. Possibility and impossibility results for encryption and commitment secure under selective opening. In *EUROCRYPT*, pages 1–35, 2009.

[10] Christina Brzuska, Marc Fischlin, Heike Schröder, and Stefan Katzenbeisser. Physically uncloneable functions in the universal composition framework. In Phillip Rogaway, editor, *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 51–70. Springer, 2011.

[11] Christina Brzuska, Marc Fischlin, Heike Schröder, and Stefan Katzenbeisser. Physically uncloneable functions in the universal composition framework. *IACR Cryptology ePrint Archive*, 2011:681, 2011.

[12] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Foundations of Computer Science (FOCS'01)*, pages 136–145, 2001.

[13] Ran Canetti, Yevgeniy Dodis, Rafael Pass, and Shabsi Walfish. Universally composable security with global setup. In Salil P. Vadhan, editor, *TCC*, volume 4392 of *Lecture Notes in Computer Science*, pages 61–85. Springer, 2007.

[14] Ran Canetti and Marc Fischlin. Universally composable commitments. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 19–40. Springer, 2001.

[15] Ran Canetti, Oded Goldreich, Shafi Goldwasser, and Silvio Micali. Resettable zero-knowledge (extended abstract). In *STOC*, pages 235–244, 2000.

[16] Ran Canetti, Joe Kilian, Erez Petrank, and Alon Rosen. Black-Box Concurrent Zero-Knowledge Requires $\omega(\log n)$ Rounds. In *33st ACM Symposium on Theory of Computing (STOC '01)*, pages 570–579. ACM, 2001.

[17] Ran Canetti, Eyal Kushilevitz, and Yehuda Lindell. On the limitations of universally composable two-party computation without set-up assumptions. In Eli Biham, editor, *EUROCRYPT*, volume 2656 of *Lecture Notes in Computer Science*, pages 68–86. Springer, 2003.

[18] Ran Canetti, Yehuda Lindell, Rafail Ostrovsky, and Amit Sahai. Universally composable two-party and multi-party secure computation. In John H. Reif, editor, *STOC*, pages 494–503. ACM, 2002.

[19] Nishanth Chandran, Vipul Goyal, and Amit Sahai. New constructions for uc secure computation using tamper-proof hardware. In Nigel P. Smart, editor, *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 545–562. Springer, 2008.

[20] Chongwon Cho, Rafail Ostrovsky, Alessandra Scafuro, and Ivan Visconti. Simultaneously resettable arguments of knowledge. In Ronald Cramer, editor, *TCC*, volume 7194 of *Lecture Notes in Computer Science*, pages 530–547. Springer, 2012.

[21] Seung Geol Choi, Jonathan Katz, Dominique Schröder, Arkady Yerukhimovich, and Hong-Sheng Zhou. (efficient) universally composable two-party computation using a minimal number of stateless tokens. *IACR Cryptology ePrint Archive*, 2011:689, 2011.

[22] Yi Deng, Vipul Goyal, and Amit Sahai. Resolving the simultaneous resettability conjecture and a new non-black-box simulation strategy. In *FOCS*, pages 251–260. IEEE Computer Society, 2009.

[23] Giovanni Di Crescenzo. Minimal assumptions and round complexity for concurrent zero-knowledge in the bare public-key model. In *COCOON*, pages 127–137, 2009.

[24] Giovanni Di Crescenzo, Giuseppe Persiano, and Ivan Visconti. Constant-round resettable zero knowledge with concurrent soundness in the bare public-key model. In *Advances in Cryptology – Crypto '04*, volume 3152 of *Lecture Notes in Computer Science*, pages 237–253. Springer-Verlag, 2004.

[25] Giovanni Di Crescenzo, Giuseppe Persiano, and Ivan Visconti. Improved Setup Assumptions for 3-Round Resettable Zero Knowledge. In *Asiacrypt '04*, volume 3329 of *Lecture Notes in Computer Science*, pages 530–544. Springer-Verlag, 2004.

[26] Giovanni Di Crescenzo and Ivan Visconti. Concurrent zero knowledge in the public-key model. In *Proc. of ICALP '05*, volume 3580 of *Lecture Notes in Computer Science*, pages 22–33. Springer, 2005.

[27] Giovanni Di Crescenzo and Ivan Visconti. On defining proofs of knowledge in the bare public key model. In *ICTCS*, pages 187–198, 2007.

[28] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[29] Nico Döttling, Daniel Kraschewski, and Jörn Müller-Quade. Unconditional and composable security using a single stateful tamper-proof hardware token. In Ishai [**?**], pages 164–181.

[30] Nico Döttling, Daniel Kraschewski, and Jörn Müller-Quade. David & goliath oblivious affine function evaluation - asymptotically optimal building blocks for universally composable two-party computation from a single untrusted stateful tamper-proof hardware token. Cryptology ePrint Archive, Report 2012/135, 2012. `http://eprint.iacr.org/`.

[31] Cynthia Dwork and Moni Naor. Zaps and their applications. In *In 41st FOCS*, pages 283–293. IEEE, 2000.

[32] Cynthia Dwork, Moni Naor, Omer Reingold, and Larry Stockmeyer. Magic functions. In *Foundations of Computer Science (FOCS'99)*, pages 523–534, 1999.

[33] Cynthia Dwork, Moni Naor, and Amit Sahai. Concurrent zero-knowledge. In *Proceedings of the 20th annual ACM symposium on Theory of computing*, STOC '98, pages 409–418. ACM, 1998.

[34] Ilze Eichhorn, Patrick Koeberl, and Vincent van der Leest. Logically reconfigurable pufs: memory-based secure key storage. In *Proceedings of the sixth ACM workshop on Scalable trusted computing*, STC '11, pages 59–64, New York, NY, USA, 2011. ACM.

[35] Sanjam Garg, Rafail Ostrovsky, Ivan Visconti, and Akshay Wadia. Resettable statistical zero knowledge. In *TCC*, Lecture Notes in Computer Science. Springer-Verlag, 2012.

[36] Rosario Gennaro and Silvio Micali. Independent zero-knowledge sets. In *ICALP*, volume 4052 of *Lecture Notes in Computer Science*, pages 181–234. Springer, 2006.

[37] Oded Goldreich and Ariel Kahan. How to construct constant-round zero-knowledge proof systems for np. *J. Cryptology*, 9(3):167–190, 1996.

[38] Shafi Goldwasser, Yael T. Kalai, and Guy. N. Rothblum. One-time programs. In *Advances in Cryptology – CRYPTO'08*, volume 5157 of *Lecture Notes in Computer Science*, pages 39–56. Springer, Berlin, Germany, 2008.

[39] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In Robert Sedgewick, editor, *STOC*, pages 291–304. ACM, 1985.

[40] Vipul Goyal, Yuval Ishai, Amit Sahai, Ramarathnam Venkatesan, and Akshay Wadia. Founding cryptography on tamper-proof hardware tokens. In Daniele Micciancio, editor, *TCC*, volume 5978 of *Lecture Notes in Computer Science*, pages 308–326. Springer, 2010.

[41] Vipul Goyal, Chen-Kuei Lee, Rafail Ostrovsky, and Ivan Visconti. Constructing non-malleable commitments: A black-box approach. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012*, pages 51–60. IEEE Computer Society, 2012.

[42] Vipul Goyal and Hemanta K. Maji. Stateless cryptographic protocols. In *FOCS*, 2011.

[43] Jorge Guajardo, Sandeep S. Kumar, Geert Jan Schrijen, and Pim Tuyls. Fpga intrinsic pufs and their use for ip protection. In Pascal Paillier and Ingrid Verbauwhede, editors, *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pages 63–80. Springer, 2007.

[44] Dennis Hofheinz. Possibility and impossibility results for selective decommitments. *J. Cryptology*, 24(3):470–516, 2011.

[45] 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, 2005.

[46] Jonathan Katz. Universally composable multi-party computation using tamper-proof hardware. In Moni Naor, editor, *Advances in Cryptology – EUROCRYPT 2007*, volume 4515 of *Lecture Notes in Computer Science*, pages 115–128, Barcelona, Spain, May 20–24, 2007.

[47] Stefan Katzenbeisser, Ünal Koçabas, Vladimir Rozic, Ahmad-Reza Sadeghi, Ingrid Verbauwhede, and Christian Wachsmann. Pufs: Myth, fact or busted? a security evaluation of physically unclonable functions (pufs) cast in silicon. In Prouff and Schaumont [65], pages 283–301.

[48] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, STOC '92, pages 723–732, New York, NY, USA, 1992. ACM.

[49] Joe Kilian and Erez Petrank. Concurrent and resettable zero-knowledge in poly-loalgorithm rounds. In *Proceedings of the 33rd annual ACM symposium on Theory of computing*, STOC '01, pages 560–569. ACM, 2001.

[50] Ünal Koçabas, Ahmad-Reza Sadeghi, Christian Wachsmann, and Steffen Schulz. Poster: practical embedded remote attestation using physically unclonable functions. In Yan Chen, George Danezis, and Vitaly Shmatikov, editors, *ACM Conference on Computer and Communications Security*, pages 797–800. ACM, 2011.

[51] Roel Maes, Anthony Van Herrewege, and Ingrid Verbauwhede. Pufky: A fully functional puf-based cryptographic key generator. In Prouff and Schaumont [65], pages 302–319.

[52] Roel Maes and Ingrid Verbauwhede. Physically unclonable functions: A study on the state of the art and future research directions. In Ahmad-Reza Sadeghi and David Naccache, editors, *Towards Hardware-Intrinsic Security*, Information Security and Cryptography, pages 3–37. Springer Berlin Heidelberg, 2010.

[53] Silvio Micali and Leonid Reyzin. Min-round resettable zero-knowledge in the public-key model. In *Advances in Cryptology – Eurocrypt '01*, volume 2045 of *Lecture Notes in Computer Science*, pages 373–393. Springer-Verlag, 2001.

[54] Silvio Micali and Leonid Reyzin. Soundness in the public-key model. In *Advances in Cryptology – Crypto '01*, volume 2139 of *Lecture Notes in Computer Science*, pages 542–565. Springer-Verlag, 2001.

[55] Daniele Micciancio and Erez Petrank. Simulatable commitments and efficient concurrent zero-knowledge. In *Advances in Cryptology – Eurocrypt '03*, volume 2045 of *Lecture Notes in Computer Science*, pages 140–159. Springer-Verlag, 2003.

[56] Tal Moran and Gil Segev. David and Goliath commitments: UC computation for asymmetric parties using tamper-proof hardware. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 527–544, Istanbul, Turkey, April 13–17, 2008. Springer, Berlin, Germany.

[57] Rafail Ostrovsky, Giuseppe Persiano, and Ivan Visconti. Constant-round concurrent non-malleable zero knowledge in the bare public-key model. In *Proc. of ICALP '08*, volume 5126 of *Lecture Notes in Computer Science*, pages 548–559. Springer, 2008.

[58] Rafail Ostrovsky, Vanishree Rao, Alessandra Scafuro, and Ivan Visconti. Revisiting lower and upper bounds for selective decommitments. In submission to TCC 2013.

[59] Ostrovsky, R., Scafuro, A., Visconti, I., Wadia, A.: Universally composable secure computation with (malicious) physically uncloneable functions. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT. Lecture Notes in Computer Science, vol. 7881, pp. 702–718. Springer (2013)

[60] Ravikanth S. Pappu, Ben Recht, Jason Taylor, and Niel Gershenfeld. Physical one-way functions. *Science*, 297:2026–2030, 2002.

[61] Ravikanth Srinivasa Pappu. *Physical One-Way Functions*. PhD thesis, MIT, 2001.

[62] Rafael Pass and Hoeteck Wee. Black-box constructions of two-party protocols from one-way functions. In *TCC*, pages 403–418, 2009.

[63] Manoj Prabhakaran, Alon Rosen, and Amit Sahai. Concurrent zero knowledge with logarithmic round-complexity. In *In 43rd FOCS*, pages 366–375, 2002.

[64] 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, 2004.

[65] Emmanuel Prouff and Patrick Schaumont, editors. *Cryptographic Hardware and Embedded Systems - CHES 2012 - 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings*, volume 7428 of *Lecture Notes in Computer Science*. Springer, 2012.

[66] Ransom Richardson and Joe Kilian. On the Concurrent Composition of Zero-Knowledge Proofs. In *Advances in Cryptology – Eurocrypt '99*, volume 1592 of *Lecture Notes in Computer Science*, pages 415–431. Springer-Verlag, 1999.

[67] Ulrich Rührmair. Oblivious transfer based on physical unclonable functions. In Alessandro Acquisti, Sean W. Smith, and Ahmad-Reza Sadeghi, editors, *TRUST*, volume 6101 of *Lecture Notes in Computer Science*, pages 430–440. Springer, 2010.

[68] Ulrich Rührmair, Stefan Katzenbeisser, and H. Busch. Strong pufs: Models, constructions and security proofs. In A. Sadeghi and P. Tuyls, editors, *Towards Hardware Intrinsic Security: Foundations and Practice*, pages 79–96. Springer, 2010.

[69] Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. Enhancing rfid security and privacy by physically unclonable functions. In Ahmad-Reza Sadeghi and David Naccache, editors, *Towards Hardware-Intrinsic Security*, Information Security and Cryptography, pages 281–305. Springer Berlin Heidelberg, 2010.

[70] Alessandra Scafuro and Ivan Visconti. On round-optimal zero knowledge in the bare public-key model. In *EUROCRYPT*, Lecture Notes in Computer Science. Springer-Verlag, 2012.

[71] Pim Tuyls and Lejla Batina. Rfid-tags for anti-counterfeiting. In David Pointcheval, editor, *CT-RSA*, volume 3860 of *Lecture Notes in Computer Science*, pages 115–131. Springer, 2006.

[72] Ivan Visconti. Efficient zero knowledge on the internet. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, pages 22–33. Springer, 2006.

[73] David Xiao. (Nearly) round-optimal black-box constructions of commitments secure against selective opening attacks. In Ishai [**?**], pages 541–558.

[74] David Xiao. On the round complexity of black-box constructions of commitments secure against selective opening attacks. Cryptology ePrint Archive, Report 2009/513 - Revision May 29, 2012, 2012. `http://eprint.iacr.org/`.

[75] David Xiao. Round-optimal black-box statistically binding selective-opening secure commitments. In *Progress in Cryptology - AFRICACRYPT 2012 - 5th International Conference on Cryptology in Africa, Ifrance, Morocco, July 10-12, 2012. Proceedings*, volume 7374 of *Lecture Notes in Computer Science*, pages 395–411. Springer, 2012.

[76] Andrew Chi-Chih Yao, Moti Yung, and Yunlei Zhao. Concurrent knowledge-extraction in the public-key model. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(002), 2007.

[77] Andrew Chi-Chih Yao, Moti Yung, and Yunlei Zhao. Concurrent knowledge extraction in the public-key model. In *ICALP (1)*, pages 702–714, 2010.

[78] Deng Yi, Dengguo Feng, Vipul Goyal, Dongdai Lin, Amit Sahai, and Moti Yung. Resettable cryptography in constant rounds - the case of zero knowledge. In *ASIACRYPT*, 2011.

[79] Moti Yung and Yunlei Zhao. Generic and practical resettable zero-knowledge in the bare public-key model. In *EUROCRYPT*, pages 129–147, 2007.

[80] Yunlei Zhao. Concurrent/resettable zero-knowledge with concurrent soundness in the bare public-key model and its applications. Cryptology ePrint Archive, Report 2003/265, 2003. `http://eprint.iacr.org/`.

[81] Yunlei Zhao, Xiaotie Deng, Chan H. Lee, and Hong Zhu. Resettable zero-knowledge in the weak public-key model. In *Advances in Cryptology – Eurocrypt '03*, volume 2045 of *Lecture Notes in Computer Science*, pages 123–139. Springer-Verlag, 2003.

[82] Damgård, I., Scafuro, A.: Unconditionally secure and universally composable commitments from physical assumptions. To appear in Asiacrypt 2013.