On Deniability in the Common Reference String and Random Oracle Model

Rafael Pass

Department of Numerical Analysis and Computer Science
Royal Institute of Technology, Stockholm, Sweden
rafael@nada.kth.se

Abstract. We revisit the definitions of zero-knowledge in the Common Reference String (CRS) model and the Random Oracle (RO) model. We argue that even though these definitions syntactically mimic the standard zero-knowledge definition, they loose some of its spirit. In particular, we show that there exist a specific natural security property that is not captured by these definitions. This is the property of *deniability*. We formally define the notion of *deniable zero-knowledge* in these models and investigate the possibility of achieving it. Our results are different for the two models:

- Concerning the CRS model, we rule out the possibility of achieving deniable zero-knowledge protocols in "natural" settings where such protocols cannot already be achieved in plain model.
- In the RO model, on the other hand, we construct an efficient 2-round deniable zero-knowledge argument of knowledge, that preserves both the zero-knowledge property and the proof of knowledge property under concurrent executions (concurrent zero-knowledge and concurrent proof-of knowledge).

1 Introduction

Zero-knowledge proofs, i.e., interactive proofs that yield no other knowledge than the validity of the assertion proved, were introduced by Goldwasser, Micali and Rackoff [26] in 1982. Intuitively, the verifier of a zero-knowledge proof should not be able to do anything it could not have done before the interaction. Knowledge, thus, in this context means the ability to perform a task. The intuition is captured through a simulation definition: We say that a protocol is zero-knowledge if there exists a simulator (that does not have access to a prover) that can simulate a malicious verifier's output after interaction with a prover. The existence of such a simulator implies that if an adversary succeeds in a task after having communicated with a prover, the adversary could just as well have reached the same results without a prover by first running the simulator. This feature has made zero-knowledge a very powerful and useful tool for proving the security of cryptographic protocols.

For some applications, such as signature schemes [18] [39], voting systems, non-interactive zero-knowledge [5] [25], concurrent zero-knowledge [14], [9] etc., it however seems hard, or is even impossible, to achieve efficient and secure schemes in the standard model. Stronger models, such as the Common Reference String (CRS) model [5], where a random string is accessible to the players, or the Random Oracle (RO) model [2], where a random function is accessible through oracle calls to the players, were therefore introduced to handle even those applications. Recently the CRS model has been extensively used in interactive settings to prove universal composability (e.g. [6] [7] [10]).

We note that an important part of the intuition behind zero-knowledge is lost in these two models in a multi-party scenario, if the CRS string or the random oracle may be reused. An easy way of seeing this is simply by noting that non-interactive zero-knowledge proofs are possible in both these model. A player having received a non-interactive proof of an assertion, it could not have proved before the interaction, can definitely do something new: it can simply send the same proof to someone else. This fact may seem a bit counter-intuitive since the intuition tells us that the simulation paradigm should take care of this. We note, however, that the simulator is much "stronger" in these models than in the plain model. As it is, the simulator is allowed to *choose* the CRS string, or random oracle, and this fact jeopardizes the zero-knowledge intuition. In fact the zero-knowledge property in these model only guarantees that the verifier will not be able to do anything *without* referring to the CRS string or the random oracle, it could not have done before. In the non-interactive setting, this problem has lead to the definition of *non-malleable non-interactive zero-knowledge* [37], and very recently *robust non-interactive zero-knowledge* [13]. In this paper we examine the problem in the more general interactive setting.

Deniable Zero-knowledge. In many interactive protocols (e.g. undeniable signatures [11], or deniable authentification [14]) it is essential that the transcript of the interaction does not yield any evidence of the interaction. We say that such protocols are *deniable*. We use the standard simulation paradigm to formalize this notion:

Definition 1. *[Informal meta-definition] A protocol is* deniable *if it is zero-knowledge and the zero-knowledge simulator can be run by the verifier.*[1]

The standard definition of zero-knowledge in the plain model certainly satisfies deniability, however this is *no longer* the case with the definitions of zero-knowledge in the CRS/RO models. This stems from the fact that in the real world the public information in the model, i.e., the CRS string or the random oracle, is fixed once and for all at start-up. When proving security, however, the simulator in these models is allowed to choose this public information in anyway it pleases as long as it "looks" ok. Thus, even though there exists a simulator for a protocol, there is no guarantee that a player can actually simulate a transcript using a certain predefined public information. Non-interactive proofs of a

[1] Strictly speaking, the simulator is an algorithm and can therefore always be run by the verifier. What we mean here is that the output of the verifier when running this simulator algorithm should be "correctly" distributed.

statement x are trivially proofs of an interaction with a party that can prove the assertion of the statement x, or else the soundness condition of the proof would be broken.

Indeed, the idea behind the simulation paradigm, and the reason for its widespread applicability, is that a verifier should be able to run the simulator by himself instead of interacting with a prover. The standard definitions of zero-knowledge in the CRS and RO models have not retained this spirit (since the simulator in these model is allowed to choose the public information, which evidently the verifier is not allowed to do), but only syntactically mimic the original zero-knowledge definition.

In the following we give formal definitions of deniable zero-knowledge in the CRS (see section 3) and RO (see section 4) models and investigate the possibility of achieving protocols satisfying the definitions.

When Does Deniability Matter. For some settings zero-knowledge and deniability is the goal (e.g. deniable authentification [14]). In such settings the standard definitions of zero-knowledge in the CRS/RO models clearly are not sufficient, since they do not guarantee deniability.

The issue of deniability also arises when a zero-knowledge protocol is used as a sub-protocol in a larger context where the CRS string or random oracle may be reused. In such a scenario it is no longer clear what security properties are guaranteed by the standard definitions of zero-knowledge in the CRS/RO models. More technically, general protocol composition becomes problematic since the simulator cannot be run when a specific CRS string or random oracle already has been selected.

Nevertheless, we mention that when "plugging-in" zero-knowledge protocols in the CRS/RO models into certain specific protocols, the standard definitions (that do not guarantee deniability) can in some cases be sufficient. For example in the construction of encryption schemes secure against chosen-ciphertext attacks [34], zero-knowledge protocols that do not satisfy deniability have been successfully used as sub-protocols.[2] (Looking ahead, the notion "unreplayability" introduced in section 1.1 is another example where zero-knowledge definitions that do not satisfy deniability can be sufficient).

Implications on the Framework for Universal Composability. A framework for universal composability (UC) was introduced by Canetti in [6]. The idea behind the framework is to put forward security definitions such that the security of a stand-alone component implies the security of a larger system where the component is plugged in, if the outer system is proven secure when having access to an "ideal" component. The UC framework allows for a modular design

[2] We mention that in the more complicated case of encryption schemes secure against *adaptive* chosen-cipher text attacks, the standard definition of zero-knowledge in the CRS model is not sufficient, but needs to be strengthened to guarantee *simulation-soundness.* [37]

of cryptographic protocols, which facilitates the design of secure solutions, e.g. [7] [10].

The ideal zero-knowledge functionality was first defined in [6] and has later been used in several subsequent works. Due to the impossibility of implementing the ideal zero-knowledge functionailty in the plain model [6], the functionality was implemented in the CRS model [7] [13]. We note that the implementation of [13] is non-interactive, i.e., only a single message is send. Their protocol is, thus, not deniable and therefore constitutes an evidence that the ideal zero-knowledge functionality *does not* capture the concerns for deniability in the framework.

The example given shows the non-triviality of the task of defining ideal functionalities in the UC framework. At a first glance it seemed like the definition given of the ideal zero-knowledge functionality would satisfy deniability. Closer inspection of the framework shows, however, that the concern for transferability/deniability is not taken into account in the framework *when introducing public objects*, such as the CRS string. This can be seen as follows: The UC framework only guarantees security if a CRS string is not reused. A transferability/deniability attack, however, relies on the fact that an honest-party reuses a CRS that has been used in a different execution. In other words, such attacks are not ruled-out by the composition theorem of [6], since they involve honest-parties deviating from their prescribed protocols by reusing a CRS string.

A serious concern is born out of this discussion: Since the zero-knowledge proof functionality is both relatively simple and quite well understood, it should be easy to define an ideal functionality that satisfies the real spirit behind the concept. In particular, the ideal zero-knowledge functionality should be deniable. Given our understanding of the concept of zero-knowledge, the definition of the ideal zero-knowledge functionality given in [6] also seems to be the right one. However, as shown, this definition does not satisfy our expectations in the UC framework. We conclude that, in order to capture the spirit behind natural definitions of ideal functionalities, the introduction of public objects in the UC framework needs to be adapted. See section 3.3 for additional discussions.

1.1 Results Concerning the CRS Model

There could have been hope that the CRS model might be used to implement deniable zero-knowledge protocols in settings where the plain model is not sufficient. We show that in natural settings, where the usage of the CRS model seems meaningful, the demand for deniability makes the CRS model collapse down to the plain model:

- We show that known black-box impossibility result concerning zero-knowledge in the plain model also hold in the CRS model, with respect to deniable zero-knowledge. That is, we show the impossibility of non-trivial deniable black-box zero-knowledge arguments in the CRS model with either of the following extra requirements:
  - straight-line simulatatable (i.e., non-rewinding)
  - non-interactive

  - constant-round strict polynomial-time simulatable
  - constant-round public-coin
  - constant-round concurrent zero-knowledge
  - 3-round
- We show an efficient transformation from deniable zero-knowledge protocols in CRS model to zero-knowledge protocols in the plain model using small overhead. This result thus rules out the possibility of constructing deniable zero-knowledge protocols in the CRS model that are much more efficient than protocols in the plain model.

Achieving a Weaker Form of Deniability. Although our results rule out the possibility of "interesting" deniable zero-knowledge protocols in many natural settings, we show that a limited form of deniability can be achieved in the CRS model by restricting the communication to a certain class of pre-specified protocols where the CRS string may be reused. Very loosely speaking, we say that a class of protocols is closed under *unreplayability* if an adversary cannot prove anything using a protocol in the class, after having interacted with a prover using a protocol in the class, that it could not have done before interacting with the prover. We show that a natural class of protocols is closed under unreplayability in the CRS model : If C is a class of interactive proofs (or arguments) *of knowledge*, with negligible soundness error, that are zero-knowledge in the CRS model, then C is closed under unreplayabilty. This result shows that restricting the communication to only arguments of knowledge that are zero-knowledge, eliminates the concern for deniability in the CRS model. We postpone these results to the full version of the paper.

1.2 Results Concerning the RO Model

While the results in the CRS model were mostly negative in nature, the situation in the RO model is rather different. Indeed we are able to construct "interesting" deniable zero-knowledge protocols.

More precisely, we show that 2 rounds are necessary and sufficient to construct deniable black-box zero-knowledge arguments for $\mathcal{NP}$ in the RO model. In fact, we construct an efficient 2-round deniable zero-knowledge argument for $\mathcal{NP}$ in the RO model that is both straight-line simulatable and witness extractable. This implies that both simulation of polynomially many concurrent executions (concurrent zero-knowledge) and simultaneous extraction of polynomially many witnesses under concurrent executions (concurrent proof of knowledge) can be performed. It was previously unknown how to simultaneously extract witnesses from polynomially many proofs in the RO model (let alone the question of deniability).

1.3 Other Models

We mention briefly that there are other models that are stronger than the plain model, such as the timing model of [14], or the on-line/off-line model of [35], that do not suffer from problems with deniability. We also note that in a public-key model, methods similar to those of designated verifiers [30] can be used to successfully implement non-trivial zero-knowledge protocols that are deniable. Indeed, the method of designated verifier shows how to convert zero-knowledge protocols that are not deniable into zero-knowledge protocols in a stronger model (namely the public-key model) that satisfy deniability.

1.4 A Computational Separation Between the RO and CRS Model

An interesting, and (as far as we know) until now, open question has been to investigate if a plausibility result in the RO model implies a plausibility result in the CRS model. An information theoretical separation between the models follows from the difference in entropy of the random oracle and CRS string. However, the computational case, which is the relevant one when considering cryptographic applications, seems more complicated.

The existence of the powerful tool of pseudo-random functions [21] has shown that in some applications an object with low-entropy (the seed to the pseudo-random function) can be used to "simulate" the behavior of a high-entropy object (namely a random function). It, thus, might seem conceivable that methods of "stretching" randomness could be used to transform protocols in the RO model to protocols in the CRS model that achieve the same task.

A natural candidate to perform such a transformation would be to substitute the random oracle with a (hash) function chosen from a class of function according to the CRS string [2]. However, it was shown by Canetti, Goldreich and Halevi [8] that there exist schemes for which every transformations of this type results in an insecure schemes.

The question of the existence of other (more complicated) transformation has, nevertheless, remained open. A side-effect of our results settles this question by showing a computational separation between the RO model and the CRS model.[3]

In fact, by combining our negative results for the CRS model and the positive results in the RO model, we obtain applications (like for example 2-round deniable black-box zero-knowledge arguments) that can be achieved in the RO model but cannot be achieved in the CRS model.

1.5 Techniques

Although this paper is mostly conceptual in nature, we believe that some of the techniques used in the proofs might be of independent interest.

Tools for Constructing Protocols in the RO Model. In order to construct our 2-round deniable zero-knowledge argument in the RO model we define and construct efficient straight-line extractable (i.e., the extraction can be performed without rewinding) commitments and straight-line witness extractable

[3] We note that this is done without resorting to "heavy" machinery like for example the PCP theorem that is needed in [8].

arguments. We mention that the straight-line extraction feature implies two strong properties that were (as far as we know) previously unattained in the RO model:

- **Simultaneous extraction of polynomially many witnesses**. Previous methods to extract witnesses [39] relied on rewinding and could therefore not be used to extract witnesses under concurrent executions.
- **Tight security reductions for non-interactive proofs of knowledge**. Standard extraction techniques for non-interactive proofs of knowledge in the RO model [39] result in "loose" security reductions (see [27] for a discussion).[4] Using straight-line extraction, on the other hand, we obtain a linear and optimal security reduction.

We mention that this technique can be used also for standard zero-knowledge proofs in RO model that do not satisfy the stronger requirement of deniability.

Proofs of Protocol Security without the Simulation Paradigm. In the proof of Lemma 3 (in section 3.1) we show that a parallelized version of Blum's coin-tossing protocol [4] can be used to generate a pseudo-random string. The interesting part of the proof is that we show this *without resorting* to the standard simulation based definition of secure computation [24]. Previously, the only known constant-round coin-tossing protocol for generating a "random" string (and not a bit) is the protocol of Lindell [31] which relies on zero-knowledge proofs and is therefore not practical. (The protocol of Lindell is, however, simulatable). More details can be found in the full version.

1.6 Preliminaries

Due to lack of space in this abstract, we assume familiarity with the following notions: Zero-knowledge in the RO model (see [2]), Zero-knowledge in the CRS model, Witness relations, Commitment schemes, Hard instance ensembles, Witness Indistinguishability (WI), Witness Hiding (WH), Proofs of knowledge (see [19] for definitions), Special soundness (see [12]), Concurrent zero-knowledge (see [20] for a survey). Formal definitions are given in the full paper.

2 ZK in the CRS/RO Model Implies WH and WI

In this section, we show two lemmas concerning the witness hiding (WH) and witness indistinguishable (WI) properties of standard (not deniable) zero-knowledge proofs, or arguments, in the CRS and RO models. Due to lack of space the proofs are omitted and can be found in the full version of the paper.

[4] Roughly, in order to break the underlying assumption the "cheating prover" has to be run $O(q)$ times, where q is the running time of the cheating prover, thus resulting in a total running time of $O(q^2)$.

Lemma 1. *Suppose that Π is a zero-knowledge proof (argument), in the CRS/RO model, for the language L. Then, for all witness relations R_L for L, Π is witness hiding in the CRS/RO model.*

Remark 1. The lemma was proven for the plain model in [16].

Lemma 2. *Let the language $L \in \mathcal{NP}$, R_L be a witness relation for L, and Π be a zero-knowledge proof (argument) in the CRS/RO model for L with efficient prover for R_L. Then Π is witness indistinguishable for R_L in the CRS/RO model.*

Remark 2. The lemma was proven for the plain model in [16], and for non-interactive proofs in the CRS model in [15].

We note that in the case of WH, the proof of the lemma is a straight-forward adaptation of the proof in the plain model [16], but concerning WI such a simple adaptation can no longer be done, as was pointed out already for the non-interactive setting using a CRS model in [15]. The problem stems from the fact that WI in the CRS/RO model considers what happens when the prover uses different witnesses, but the *same* CRS string/random oracle.

Thus, although the lemmas show positive results concerning the security of protocols satisfying the standard definition of zero-knowledge in these models, the non-triviality of the adaptation needed in the case of WI, by itself, shows that special care has to be taken in models where the simulator is allowed to choose the public information.

Nevertheless, the essence of Lemma 1 and 2 is that in settings where only WH or WI is required as a security requirement, the standard definitions of zero-knowledge in the CRS or RO model are sufficient. Looking ahead, we will use the WH and WI properties of zero-knowledge proofs in the RO model in the construction of a deniable zero-knowledge protocol in the RO model.

3 On Deniable Zero-Knowledge Proofs in the CRS Model

To be able to obtain deniable zero-knowledge in the CRS model, we restrict the power of the simulator in the definition of zero-knowledge in the CRS model. The key to the problem seems to be the fact that the simulator in the CRS model *chooses* the CRS string. In fact, if the simulator was able to perform a simulation without choosing the CRS string, we would be sure that the verifier had not learnt anything, except the assertion of the statement being proved, even with respect to the CRS string. This leads us to a new zero-knowledge definition.

Definition 2. *We say that an interactive proof (P, V) for the language $L \in \mathcal{NP}$, with the witness relation R_L, is* deniable zero-knowledge in the CRS model *if for every PPT machine V^* there exists an expected polynomial time probabilistic simulator S such that the following two ensembles are computationally indistinguishable (when the distinguishing gap is a function in $|x|$)*

– $\{(r, \langle P(y_x), V^*(z)\rangle(x, r))\}_{z\in\{0,1\}^*, x\in L}$ *for arbitrary* $y_x \in R_L(x)$
– $\{(r, S(x, z, r)\}_{z\in\{0,1\}^*, x\in L}$

where r *is a random variable uniformly distributed in* $\{0, 1\}^{poly(|x|)}$.
That is, for every probabilistic algorithm D *running in time polynomial in the length of its first input, every polynomial* p*, all sufficiently long* $x \in L$*, all* $y_x \in R_L(x)$ *and all auxiliary inputs* $z \in \{0, 1\}^*$ *it holds that*

$$|Pr[D(x, z, r, \langle P(y_x), V^*(z)\rangle(x, r))) = 1] \\ -Pr[D(x, z, r, S(x, z, r)) = 1]| < \frac{1}{p(|x|)}$$

where r *is a random variable uniformly distributed in* $\{0, 1\}^{poly(|x|)}$.

3.1 On the Impossibility of More Efficient Deniable ZK Protocols

We show that if there exist an interactive deniable zero-knowledge proof (or argument) with negligible soundness error for a language L in the CRS model then there exists an interactive zero-knowledge proof (or argument) with negligible soundness error for L in the plain model using essentially the same communication complexity. In fact, we show a general transformation that only uses an overhead of twice the length of the CRS string plus the length of a statistically binding commitment to a string of the same length as the CRS string.

The construction. Suppose that protocol Π is an interactive deniable zero-knowledge proof (or argument) with negligible soundness error, for the language L, in the CRS model. Suppose further that a CRS string of length $p(n)$, where $p(n)$ is a polynomial, is used for proving membership of instances in L of size n, using Π. Now consider the protocol Π' in the plain model (without a CRS string), for proving membership of instances in L of length n, constructed by simply adding a coin-tossing phase to the protocol Π:

Protocol Π'
Phase one:
P → V: Commits, using a statistically binding commitment scheme, that is non-uniformly computationally hiding, to a random string of length $p(n)$.
V → P: Sends a random string of length $p(n)$.
P → V: Opens up the commitment.
Phase two:
P ↔ V: Both parties thereafter use the XOR of the strings as a CRS string and execute the protocol Π.

In the full version of the paper we show the following lemma:

Lemma 3. *If Π is an interactive deniable zero-knowledge proof (or argument) with negligible soundness error, for the language L, in the CRS model, then the protocol Π', resulting from the above transformation, is an interactive zero-knowledge proof (or argument) with negligible soundness error for the language L.*

Remark 3. The existence of statistically binding commitment schemes that are non-uniformly computationally hiding is implied by the existence of non-uniform one-way functions by combining the results of [29] and [33].

Remark 4. We note that we do not show that the coin-tossing protocol in phase one is simulatable. Indeed, for our construction to work we simply have to show that the output of the coin-tossing is pseudo-random.

This result, thus, rules out the possibility of finding deniable zero-knowledge protocols that can be implemented much more efficiently in the CRS model than in the plain model.

3.2 On the Impossibility of "Non-trivial" Deniable ZK Protocols

In this section we show that known black-box impossibility result concerning zero-knowledge in the plain model also hold in the CRS model with respect to deniable zero-knowledge. That is we show that for known settings where it seems interesting to resort to the CRS model the demand for deniability makes the CRS model collapse down to the plain model. (The proofs that are left out are given in the full version).

Theorem 1. *If Π is a straight-line black-box simulatable deniable zero-knowledge proof (or argument), in the CRS model, for the language L with negligible soundness error, then $L \in \mathcal{BPP}$.*

We continue with two impossibility results that follow from Lemma 3:

Theorem 2. *Assume the existence of statistically binding commitment schemes that are non-uniformly computationally hiding. If Π is a constant-round strict polynomial-time black-box simulatable deniable zero-knowledge proof (or argument) with negligible soundness in the CRS model for the language L, then $L \in \mathcal{BPP}$.*

Theorem 3. *Assume the existence of statistically binding commitment schemes that are non-uniformly computationally hiding. If Π is a constant-round black-box simulatable public-coin deniable zero-knowledge proof (or argument) with negligible soundness in the CRS model for the language L, then $L \in \mathcal{BPP}$.*

Proof. It is clear from the construction that the transformation in section 3.1 preserves the public-coin property of the protocol. Now, since Goldreich and Krawczyk [22] have shown the impossibility of non-trivial constant-round black-box public-coin zero-knowledge arguments, $L \in \mathcal{BPP}$. □

As a sanity check to the definition we also note the impossibility of non-trivial non-interactive zero-knowledge arguments,

Theorem 4. *If Π is a non-interactive deniable zero-knowledge argument, in the CRS model, for the language L with negligible soundness error, then $L \in \mathcal{BPP}$.*

Proof. Follows directly from Theorem 1 since non-interactive arguments need to be black-box straight-line simulatable. □

Indeed, non-interactive proofs are the most obvious violation of deniability, in the CRS model, since they can be passed on.

Goldreich-Krawczyk Reductions. In 1990, Goldreich and Krawczyk [22] showed that if a language L has an interactive zero-knowledge argument with negligible soundness, using less than 4 rounds, with a blackbox simulator, then $L \in \mathcal{BPP}$. The method of Goldreich-Krawczyk has later been used to show black-box impossibility results in the case of constant-round concurrent zero-knowledge [9], and very recently in the case of strict polynomial time simulatable zero-knowledge [1]. On a high-level, the Goldreich-Krawczyk method is a constructive reduction from a machine deciding the language L to a simulator of the zero-knowledge argument. That is, the existence of a simulator implies the existence of a machine deciding the language, which in turn implies that the language is in $\mathcal{BPP}$.

Indeed, since the reduction is black-box and constructive, the same reduction can be used for protocols that are deniable zero-knowledge in the CRS model. The machine deciding the language, would simply first choose a random string and thereafter run the original deciding machine using the random string as a CRS string. Careful examination of the proofs of [22] and [9] thus gives:

Theorem 5. *If Π is a 3-round black-box simulatable deniable zero-knowledge proof (or argument) in the CRS model, for the language L, with negligible soundness error, then $L \in \mathcal{BPP}$.*

Theorem 6. *If Π is a constant-round black-box simulatable deniable concurrent zero-knowledge argument in the CRS model, for the language L, with negligible soundness error, then $L \in \mathcal{BPP}$.*

3.3 Conclusions and Directions for Future Research

We have shown that for currently known settings, the CRS model cannot be used to implement deniable black-box zero-knowledge protocols for languages in $\mathcal{NP}$, that cannot already be implemented in the plain model. In the full version of the paper we, nevertheless, show that a limited form of deniability (called *unreplayability*) can be achieved by restricting the communication of honest-parties to a certain class of protocols (see section 1.1).

Concerning the UC framework [6], we have shown that the ideal zero-knowledge functionality is not deniable. Thus, in order to be able to model universally composable deniable zero-knowledge, either a new definition has to be given or the incorporation of public objects in the framework modified.

A possible approach would be to only allow composition with ideal functionalities that physically cannot be reused, thus ruling out the use of the ideal CRS functionality and other functionalities that model public information. However, since the plain model is too weak to construct even universally composable commitment [7], some extra set-up assumptions need to be incorporated into the security definitions, in such a way that the simulator can be run by the parties themselves. For example, if incorporating a CRS string in the framework, the simulator should be able to carry out the simulation for all but a negligible fraction of CRS strings, in analogy with Definition 2. We note, however, that Theorem 1, which states the impossibility of straight-line simulatable deniable zero-knowledge in the CRS model, yields the impossibility of universally composable zero-knowledge in this setting, since a protocol implementing the ideal zero-knowledge functionality must have a straight-line simulator.[5] On the other hand, if incorporating a public-key infrastructure in the framework, methods similar to those of designated verifier [30] could possibly be used to achieve universally composable deniable zero-knowledge.

An altogether different approach was taken in [32] [36] where it is shown how to realize the ideal zero-knowledge functionality *without* resorting to set-up assumptions (such as a CRS string), by trading universal composability for the weaker notion of concurrent composability.

Open Problems. An interesting open problem is to find a type of deniable zero-knowledge protocol that can be achieved in the CRS but not in the plain model. Since most of our results only apply in the black-box setting, a direction would be to investigate the non-black-box setting.

4 On Deniable Zero-knowledge Proofs in the RO Model

As in the CRS model, in order to obtain interactive proofs and arguments, with random oracles, that capture the spirit of zero-knowledge, we need to resort to a weaker simulation model, where the simulator no longer is allowed to choose the random oracle, but should be able to perform the simulation for all but a negligible fraction of random oracles. Such a simulator can therefore be run by a verifier, assuring that the intuitive interpretation of zero-knowledge holds, i.e., that the verifier cannot do anything except to assert the validity of the statement proved, that it could not have done before the interaction with a prover.

Definition 3. *We say that an interactive proof* (P, V) *for the language* $L \in \mathcal{NP}$*, with witness relation* R_L*, is* deniable zero-knowledge in the RO model *if for every*

[5] For those familiar with the UC framework, this is due to the fact that the environment cannot be rewound. Now, supposing a real-life adversary that simply forwards messages between the environment and the simulator, shows that the simulator needs to be straight-line. More details in [6].

PPT verifier V^* there exists an expected polynomial time probabilistic simulator S such that the following two ensembles are computationally indistinguishable (when the distinguishing gap is a function in $|x|$):

- $\{(RO, \langle P^{RO}(y_x), V^{*RO}(z)\rangle(x))\}_{z\in\{0,1\}^*, x\in L}$ *for arbitrary* $y_x \in R_L(x)$
- $\{RO, S^{RO}(z, x))\}_{z\in\{0,1\}^*, x\in L}$

where $RO : \{0,1\}^{poly(|x|)} \to \{0,1\}^{poly(|x|)}$ *is a uniformly distributed random variable.*
That is, for every probabilistic algorithm D running in time polynomial in the length of its first input, every polynomial p, all sufficiently long $x \in L$, all $y_x \in R_L(x)$ and all auxiliary inputs $z \in \{0,1\}^*$ it holds that

$$|Pr[D^{RO}(x, z, \langle P^{RO}(y_x), V^{*RO}(z)\rangle(x))) = 1]$$
$$-Pr[D^{RO}(x, z, S^{RO}(x, z)) = 1]| < \frac{1}{p(|x|)}$$

where $RO : \{0,1\}^{poly(|x|)} \to \{0,1\}^{poly(|x|)}$ *is a uniformly distributed random variable.*

We note that when proving security according to the standard zero-knowledge definition in the RO model, the simulator has two advantages over a plain model simulator, namely,

- The simulator can see what values parties query the oracle on.
- The simulator can answer these queries in whatever way it chooses as long as the answers "look" ok.

The definition of deniable zero-knowledge in the RO model restricts the power of the simulator and *only* allows it to see on what value the parties query the oracle (thus out of the two advantages only the first remains). This is due to the fact that in the definition of deniable zero-knowledge in the RO model, the distinguisher is given access to the random oracle and can thus verify if the simulator has answered the oracle queries in accordance to the pre-specified oracle. We, however, use this first property in a novel fashion, and show that it alone is an extremely powerful tool. Looking ahead, we use the random oracle to construct commitment schemes where the simulator, gaining access to all oracle calls, will be able to extract the committed values, without rewinding the committer.

As a sanity check to the definition we start by noting: (proof is given in the full version)

Theorem 7. *If Π is a one-round deniable zero-knowledge argument, in the RO model, for the language $L \in \mathcal{NP}$ with negligible soundness error, then $L \in \mathcal{BPP}$.*

On the positive side we show that 2 rounds are necessary to construct efficient and "robust" deniable zero-knowledge protocols for $\mathcal{NP}$. In fact we construct a protocol that is both concurrent zero-knowledge and concurrent proof of knowledge through a transformation from any special-sound honest-verifier zero-knowledge (HVZK) public-coin argument. We here briefly outline the construction.

Outline of the Construction of 2-round Deniable ZK Arguments. On a very high level the protocol follows the paradigm of Feige-Shamir [17]. The verifier start by sending a "challenge" and a witness hiding proof of knowledge of the answer to the challenge, to the prover. The prover thereafter shows using a WI argument that either it has a witness for the statement it wishes to prove or that it has the answer to the challenge.

The difficulty in constructing such a protocol relies in the fact that each of these steps must be implemented in a single message.[6]

The main technical ingredient that allows us to achieve this goal is the introduction of straight-line extractable commitments in the RO model (see section 4.1). On a high level, these are commitments where the value committed to can be extracted by a simulator without the use of rewinding techniques. We construct such commitment schemes by letting the committer use the random oracle to commit. It follows from the random properties of the oracle that the committer, in order to succeed in opening a commitment must have applied the oracle on it, which means that by simply observing all the queries the adversary makes, the committed values can be extracted without rewinding.

Having established this powerful tool, in section 4.2 we construct a one-round straight-line witness extractable zero-knowledge arguments for Graph-3-Coloring in the RO model, by implementing the commitment scheme in the GMW protocol [23] with straight-line extractable commitments and thereafter applying the Fiat-Shamir transformation [18] [2] to "collapse" it down to a one-round zero-knowledge argument in the RO model (see Lemma 6). Straight-line witness extraction here means that a witness to the statement proved can be extracted without rewinding the prover. Lemma 1 and 2 can now be applied to show that the one-round protocol, which is zero-knowledge in the RO model, is both WH and WI in the RO model.

In order to achieve an efficient protocol, in Lemma 7, we show how to construct a WH and WI one-round straight-line witness extractable argument from any special-sound HVZK public-coin argument. Essentially this is done by transforming the special-sound HVZK argument into a cut-and-choose argument and thereafter applying the same transformation as was done for Graph-3-Coloring.

In section 4.3 we finally put everything together to achieve the 2-round deniable zero-knowledge argument (see Theorem 8). We here rely on the efficient OR transformation of [12] to implement the second message of the protocol.

We mention that some technical problems related to the malleability of the commitments arise in the security proof. Nevertheless, since we have access to

[6] Technically, it is actually sufficient that the first step is implemented with a single message. The second step could conceivable be implemented using 2 rounds (see [35]). Nevertheless, our construction implements both steps using one-round solutions.

a random oracle these problems can be resolved in a rather straightforward manner.

4.1 Straight-line Extractable Commitments

We construct efficient commitment schemes with strong properties, without allowing the simulator to choose the random oracle. We start by defining the notion of straight-line extractable commitments schemes in the RO model. For simplicity we only state the definition for non-interactive commitment schemes.

Definition 4. *Let a PPT committer C commit to a string using a non-interactive commitment scheme, sending c to the receiver, where $|c| = poly(n)$. We say that the non-interactive commitment scheme is* straight-line extractable in the RO model *if there exists a PPT extractor machine E such that for all c, if C succeeds in decommitting to x with non-negligible probability, then $E(c, l) = x$ with overwhelming probability, where l is a list of all the random oracle queries and answers performed by C during and before the commit phase.*

Remark 5. We note that the extractor E is not given access to the random oracle, but instead receives both the queries and the answers to those queries.

When having access to a random oracle it is easy to construct efficient commitment schemes that are straight-line extractable. Let l be a super-logarithmic polynomially bounded function, i.e., $\omega(log(n)) \leq l(n) \leq poly(n)$, and RO : $\{0,1\}^{2n} \rightarrow \{0,1\}^{l(n)}$ be a random oracle. Consider the following commitment scheme:

SLCom
Commit phase (A sends a commitment, to $x \in \{0,1\}^n$, to B)
A randomly picks $r \in \{0,1\}^n$
A $\rightarrow$ B : $c = RO(x, r)$

Reveal phase
A $\rightarrow$ B : x, r
B checks that $c = RO(x, r)$

Lemma 4. *SLCom is a straight-line extractable non-interactive commitment scheme in the RO model.*

Proof. The hiding and binding properties are proven in the full version.

Straight-line extraction follows: The extractor simply goes through the list $\{(x_i, r_i), c_i\}_{i=1..poly(n)}$ and checks if there is an i such that $c_i = c$. If so it returns x_i, and otherwise nothing. Since the committer, in order to succeed in opening the commitment with probability that is non-negligible, must have used the random oracle on the value it committed to, the extractor always succeeds if the committer succeeds with probability that is non-negligible. (A cheating committer that has not used the random oracle on the value committed to has a probability of $\frac{T(n)}{2^{l(n)}}$, where $T(n)$ is the number of oracle calls during the decommit phase, of decommitting.) □

In fact, SLCom can be used either as a statistically binding or statistically hiding commitment scheme depending on the parameter l: (proof is given in the full version)

Lemma 5. *If $l(n) = 4n$ then SLCom is a statistically binding non-interactive commitment scheme in the RO model. If $l(n) = n/8$ then SLCom is a statistically hiding non-interactive commitment scheme in the RO model.*

Extractable Commitments with Oracle Restrained to a Prefix. To be able to construct multiple commitments that are non-malleable with respect to each other we generalize the notion of straight-line extractability. We say that a commitment scheme in the RO model is *straight-line extractable with oracle restrained to the prefix s* if the commitment scheme is straight-line extractable and there exists an extractor that succeeds in extracting witnesses using only oracle queries that begin with the prefix s. We will in the following let different parties use different prefixes allowing for individual extraction of the committed values.

We note that SLCom can be changed in a straight-forward manner to become straight-line extractable with oracle restrained to the prefix s, by simply concatenating the string s to the oracle queries, i.e., $RO(s, x, r)$ becomes a commitment to the string x, where $RO : \{0,1\}^{2n+|s|} \rightarrow \{0,1\}^{l(n)}$.

4.2 Straight-line Witness Extractable Proofs

All previously known proofs of knowledge in the RO model (e.g. [39]) relied on rewinding and could therefore not be applied to simultaneously extract polynomially many witnesses. We introduce a stronger notion of proofs of knowledge, namely proofs where witnesses can be extracted without rewinding the prover. More formally,

Definition 5. *We say that an interactive proof with negligible soundness (P, V) for the language $L \in \mathcal{NP}$, with the witness relation R_L, is* straight-line witness extractable in the RO model *if for every PPT machine P^* there exists a PPT witness extractor machine E such that for all $x \in L$, all $y, r \in \{0,1\}^*$, if $P^*_{x,y,r}$ convinces the honest verifier with non-negligible probability, on common input x, then $E(\text{view}_V[(P^*_{x,y,r}, V(x))], l) \in R_L(x)$ with overwhelming probability, where $P^*_{x,y,r}$ denotes the machine P^* with common input fixed to x, auxiliary input fixed to y and random tape fixed to r,* $\text{view}_V[(P^*_{x,y,r}, V(x))]$ *is V's view including its random tape, when interacting with $P^*_{x,y,r}$, and l is a list of all oracle queries and answers posed by $P^*_{x,y,r}$ and V.*

We show two constructions to achieve efficient straight-line witness extractable arguments in the RO model. First, we show how the GMW [23] protocol for proving the existence of a 3 coloring to a graph directly can be turned into a straight-line witness extractable, WH and WI, one-round argument in the RO model, by applying the Fiat-Shamir transformation [18] to "collapse" it down to one round, and using straight-line extractable commitments. Secondly we show how to transform any three round special-sound HVZK public-coin argument into a straight-line witness extractable, WH and WI, one-round argument. The second construction is of interest as it allows us to construct efficient protocols without going through Cook's transformation.

An Argument System for Graph-3-Coloring. We start off with the three round protocol of GMW (Goldreich, Micali, Widgerson) [23]:

Protocol Π (GMW's Graph 3-coloring proof):
Common input: a directed graph $G = (V_G, E_G)$, with $n = |V_G|$
Auxiliary input to the prover: a 3-coloring of G, $c_0, c_1, .., c_n \in \{1, 2, 3\}$.

P uniformly chooses a permutation π over 1,2,3.
P $\rightarrow$ V: Commits to $\pi(c_0), \pi(c_1), .., \pi(c_n)$ using any statistically binding commitment scheme.
V $\rightarrow$ P: Uniformly selects an edge $(i, j) \in E$.
P $\rightarrow$ V: Reveals c_i, c_j.
V checks that c_i and c_j are different colors.

As is shown in [2] the protocol can be collapsed down to a one-round zero-knowledge argument, Π', in the RO model by running $t = 2n * |E_G|$ parallel versions of the protocol and applying the random oracle to all the t first messages, to "simulate" the honest verifier. This transformation is called the Fiat-Shamir transformation [18].

Protocol Π':
P $\rightarrow$ V: $a' = a'_1, a'_2, .., a'_t$, $c' = c'_1, c'_2, .., c'_t$.
V checks that for all $1 \leq i \leq t$, $(a'_i, RO(a')_i, c'_i)$ is an accepting execution of the protocol Π, where $RO(a')_i$ signifies the i'th part of the random oracle's reply, such that each part has the appropriate size of the verifier's challenge in protocol Π.

Since Π' is zero-knowledge in the RO model it is, by Lemma 1 and 2, also WH and WI. Now, if the commitment scheme chosen has the property of being straight-line extractable, the resulting protocol is straight-line witness extractable. (proof is given in the full version)

Lemma 6. *If the protocol Π' is instantiated with a straight-line extractable commitment scheme the resulting protocol is straight-line witness extractable, witness hiding and witness indistinguishable in the RO model.*

A Transformation from HVZK Protocols. Suppose $\Pi = (a, b, c)$ is a three round special-sound HVZK public-coin argument for the language $L \in \mathcal{NP}$. In order to achieve a one-round witness extractable, WH and WI argument for L we transform the protocol Π into a cut-and-choose protocol Π' and thereafter use the same transformation as was done in the case of the proof of Graph-3-Coloring. Consider the following protocol:

Protocol Π':
P $\rightarrow$ V: a, two different random numbers $b_0, b_1 \in B$, commitments to c_0, and c_1 where c_i is the answer to the query b_i with a as first message in the protocol Π
V $\rightarrow$ P: chooses q randomly from $\{0, 1\}$
P $\rightarrow$ V: Decommits to c_q
V checks that (a, b_q, c_q) is a consistent execution of the protocol Π

Now, let Π'' be the protocol obtained after applying the Fiat-Shamir transformation on Π', i.e., running $2n$ versions of the protocol in parallel, and simulating the verifier's challenge by applying the random oracle to the first message:

Protocol Π'':
P $\rightarrow$ V: $a' = a'_1, a'_2, .., a'_t$, $c' = c'_1, c'_2, .., c'_t$.
V checks that for all $1 \leq i \leq 2n$, $(a'_i, RO(a')_i, c'_i)$ is an accepting execution of the protocol Π', where $RO(a')_i$ signifies the i'th bit of the random oracle's reply.

In the full version of the paper we show,

Lemma 7. *If the protocol Π'' is instantiated with a straight-line extractable commitment scheme the resulting protocol is a straight-line witness extractable, witness hiding and witness indistinguishable argument for L in the RO model.*

Witness Extraction by an Oracle Restrained to a Prefix. As with the commitments schemes, the above mentioned protocols can easily be turned into arguments that are witness extractable by an oracle restrained to a certain prefix, by using commitment schemes that are straight-line witness extractable by oracle restrained to the prefix.

4.3 Deniable Concurrent Zero-knowledge Proofs of Knowledge

In this section we use the witness extractable, WH and WI, one-round arguments in a way similar to the Feige-Shamir construction [17] to construct a 2-round straight-line simulatable deniable zero-knowledge argument of knowledge for $\mathcal{NP}$ in the RO model. Since the protocol is straight-line simulatable it is also deniable concurrent zero-knowledge:

Theorem 8. *Assuming the existence of polynomially computable one-way functions, there exists a two round deniable black-box concurrent zero-knowledge argument for languages in $\mathcal{NP}$ in the RO model. Furthermore the argument is both straight-line witness extractable, and straight-line simulatable.*

Proof. Let $f : \{0,1\}^n \to \{0,1\}^{poly(n)}$ be a one-way function, and let Π' be a special-sound HVZK public-coin argument for proving the knowledge of a pre-image to f. Such argument systems exists for every one-way function, by reducing the one-way function to an instance of the graph hamiltonicity problem, using Cook's theorem, and thereafter using Blum's protocol [3]. We emphasize, however, that if a specific one-way function is used, the HVZK argument can be tailored for the function to get an efficient implementation. Examples of such protocols are the Guillou-Quisquater scheme [28] for the RSA function, and the Schnorr scheme [38] for the discrete logarithm.

Let the witness relation $R_{L'}$, where $(x, y) \in R_{L'}$ if $f(x) = y$, characterize the language L'.

Let $RO : \{0,1\}^{poly(n)} \to \{0,1\}^{poly(n)}$ be a random oracle, and the language $L \in \mathcal{NP}$. Consider the following protocol for proving that $x \in L$:

Protocol SLZK
V chooses a random number $r \in \{0,1\}^n$.
V $\to$ P: $c = f(r)$, a one-round WH, straight-line witness extractable, by oracle restrained to prefix "0", argument of the statement "$\exists r'$ s.t $c = f(r')$" for the witness relation $R_{L'}$.
P $\to$ V: a one-round WI, straight-line witness extractable, by oracle restrained to prefix "1", argument of the statement "$\exists r'$ s.t $c = f(r') \vee x \in L$" for the witness relation $R_{L \vee L'}(c, x) = \{(r', w) | r' \in R_{L'}(c) \vee w \in R_L(x)\}$.

To implement the first message, we use the transformation described in section 4.2 to turn Π', i.e., the special-sound HVZK zero-knowledge argument for L', into the needed one-round argument for L'.
The second message is implemented as follows: Assuming that we have a special-sound HVZK public-coin argument for L, we can use the efficient OR transformation in [12] to yield a special-sound HVZK public-coin argument for $L \vee L'$ and the witness relation $R_{L \vee L'}$.[7] We can thereafter apply the transformation in section 4.2.

Completeness of the protocol is clear. In order to prove soundness, we start by noting that the prover sends an argument that is straight-line witness extractable by oracle restrained to prefix "1". But since the honest verifier has not used the oracle with prefix "1", a witness can be extracted using only the prover's oracle queries. If a malicious prover succeeds in convincing the honest verifier, he must thus have either an r' s.t $c = f(r')$ or a witness for $x \in L$. We will show that the prover needs to have a witness for x: Let the probability ensemble U be uniform on $\{0,1\}^n$, and let $X = f(U)$ be a probability ensemble for the language L'. Then since f is a one-way function, X is a hard instance ensemble. Now, if the prover, after having received the verifier's first message was able to a find a witness to a randomly chosen instance in the hard-instance

[7] The resulting argument uses less communication than the argument for L plus the argument for L'.

ensemble X, this would violate the witness hiding property of the verifier's message. The claim that the prover must have a witness for x follows. The protocol is thus straight-line witness extractable for the statement $x \in L$. Soundness follows automatically.

Straight-line zero-knowledge: The simulator simply extracts r from the verifier's first message and then uses it as a "fake" witness to send its proof. Since the prover's message is a WI argument, the simulator's output is indistinguishable from the honest prover's. □

Remark 6. We note that since the protocol is straight-line witness extractable it is also witness extractable under concurrent executions, i.e., witnesses to all concurrent executions can be simultaneously extracted. Indeed, this feature is of great importance in, for example, authentification schemes. We note that it was previously unknown how to simultaneously extract witnesses from polynomially many proofs in the RO model.

Remark 7. Even though we have access to a random oracle we need to rely on the existence of one-way functions since our protocol uses the one-way function in a non-blackbox way. In fact, we either apply Cook's transformation on the function, or use specially tailored protocols for specific one-way functions.

A Note on the Efficiency of SLZK. Although the protocol SLZK is constructed through an efficient transformation from any special-sound HVZK argument, the transformation turns the HVZK protocol into a cut-and-choose protocol inducing a blow up in communication complexity of n. In the full version of the paper we also show the existence of a more efficient protocol considering communication complexity, which in particular is not cut-and-choose. The protocol that is also a proof and not an argument, as SLZK, however uses four rounds instead of the optimal two.

4.4 Conclusions and directions for future research

We have shown the facility of constructing efficient and powerful protocols that are deniable zero-knowledge in the RO model.

Open problems. The most urgent open problem is to find a more efficient construction of one-round witness extractable arguments that do not rely on cut-and-choose techniques. Secondly, our 2-round protocol relies on the existence of one-way functions, while our 4-round protocol (given in the full version) does not. We wonder if it is possible to construct 2-round straight-line simulatable deniable zero-knowledge protocols without any further assumptions than the random oracle.

5 Acknowledgments

First, I wish to thank Johan Håstad for his invaluable help and comments. I am also very grateful to Ran Canetti for helpful discussions. Thanks also to Shafi Goldwasser, Tal Rabin, Alon Rosen, Victor Shoup and the anonymous referees for helpful comments.

References

1. B. Barak and Y. Lindell. Strict Polynomial-Time in Simulation and Extraction. In *34th STOC*, pages 484–493, 2002.
2. M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *1st ACM Conf. on Computer and Communications Security*, pages 62–73, 1993.
3. M. Blum. How to prove a Theorem So No One Else Can Claim It. *Proc. of the International Congress of Mathematicians,* Berekeley, California, USA, pages 1444–1451, 1986.
4. M. Blum. Coin Flipping by Telephone. In *Crypto81*, ECE Report 82-04, ECE Dept., UCSB, pages 11–15, 1982
5. M. Blum, P. Feldman and S. Micali. Non-Interactive Zero-Knowledge and Its Applications. In *20th STOC*, pages 103–112, 1988
6. R. Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In *34th STOC*, pages 494–503, 2002.
7. R. Canetti and M. Fischlin. Universally Composable Commitments. In *Crypto2001*, Springer LNCS 2139, pages 19–40, 2001.
8. R. Canetti, O. Goldreich and S. Halevi. The Random Oracle Methodology, Revisited. In *30th STOC*, pages 209–218, 1998
9. R. Canetti, J. Kilian, E. Petrank and A. Rosen. Black-Box Concurrent Zero-Knowledge Requires (almost) Logarithmically Many Rounds. *SIAM Jour. on Computing*, Vol. 32(1), pages 1–47, 2002.
10. R. Canetti, Y. Lindell, R. Ostrovsky and A. Sahai. Universally Composable Two-Party and Multy-Party Computation. In *34th STOC*, pages 494–503,2002.
11. D. Chaum and H. van Antwerpen. Undeniable Signatures. In *Crypto89*, Springer LNCS 435, pages. 212–216, 1989.
12. R. Cramer, I. Damgård and B. Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *Crypto94*, Springer LNCS 839, pages. 174–187, 1994
13. A. De Santis, G. Di Crescenzo, R. Ostrovsky, G. Persiano and A. Sahai. Robust Non-interactive Zero Knowledge. In *Crypto2001*, Springer LNCS 2139, pages 566–598, 2001.
14. C. Dwork, M. Naor and A. Sahai. Concurrent Zero-Knowledge. In *30th STOC*, pages 409–418, 1998.
15. U. Feige, D. Lapidot and A. Shamir. Multiple Noninteractive Zero Knowledge Proofs under General Assumptions. *Siam Jour. on Computing 1999*, Vol. 29(1), pages 1–28.
16. U. Feige and A. Shamir. Witness Indistinguishability and Witness Hiding Protocols. In *22nd STOC*, pages 416–426, 1990.
17. U. Feige and A. Shamir. Zero Knowledge Proofs of Knowledge in Two Rounds. In *Crypto89*, Springer LNCS 435, pages. 526–544, 1989.

18. A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In *Crypto86*, Springer LNCS 263, pages 181–187, 1987
19. O. Goldreich. *Foundations of Cryptography – Basic Tools*. Cambridge University Press, 2001.
20. O. Goldreich. Zero-knowledge twenty years after their invention. Weizmann Institute, 2002.
21. O. Goldreich, S. Goldwasser and S. Micali. How to Construct Random Functions. *JACM*, Vol. 33(4), pages 210–217, 1986.
22. O. Goldreich and H. Krawczyk. On the Composition of Zero-Knowledge Proof Systems. *SIAM Jour. on Computing*, Vol. 25(1), pages 169–192, 1996.
23. O. Goldreich, S. Micali and A. Wigderson. Proofs that Yield Nothing But Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems. *JACM*, Vol. 38(1), pp. 691–729, 1991.
24. O. Goldreich, S. Micali and A. Wigderson. How to Play any Mental Game – A Completeness Theorem for Protocols with Honest Majority. In *19th STOC*, pages 218–229, 1987.
25. O. Goldreich and Y. Oren. Definitions and Properties of Zero-Knowledge Proof Systems. *Jour. of Cryptology*, Vol. 7, No. 1, pages 1–32, 1994.
26. S. Goldwasser, S. Micali and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Jour. on Computing*, Vol. 18(1), pp. 186–208, 1989.
27. E. Goh and S. Jarecki. A Signature Scheme as Secure as the Diffie-Hellman Problem. In *EuroCrypt2003*, Springer LNCS 2656, pages 401–415, 2003.
28. L.C. Guillou and J. Quisquater. A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Trasmission and Memory. In *EuroCrypt88*, Springer LNCS 330, pages 123–128, 1988.
29. J. Håstad, R. Impagliazzo, L.A. Levin and M. Luby. Construction of Pseudorandom Generator from any One-Way Function. *SIAM Jour. on Computing*, Vol. 28 (4), pages 1364–1396, 1999.
30. M. Jakobsson, K. Sako and R. Impagliazzo. Designated Verifier Proofs and Their Applications. In *EuroCrypt96*, Springer LNCS 1070, pages 143–154.
31. Y. Lindell. Parallel Coin-Tossing and Constant-Round Secure Two-Party Computation. In *Crypto2001*, Springer LNCS 2139, pages 171–189, 2001.
32. Y. Lindell. Bounded-Concurrent Secure Two-Party Computation Without Setup Assumptions. To appear in *34th STOC*, 2003.
33. M. Naor. Bit Commitment using Pseudorandomness. *Jour. of Cryptology*, Vol. 4, pages 151–158, 1991.
34. M. Naor and M. Yung. Universal One-Way Hash Functions and their Cryptographic Applications. In *21st STOC*, pages 33–43, 1989.
35. R. Pass. Simulation in Quasi-polynomial Time and its Application to Protocol Composition. In *EuroCrypt2003*, Springer LNCS 2656, pages 160–176, 2003.
36. R. Pass and A. Rosen. Bounded-Concurrent Two-Party Computation in Constant Number of Rounds. Submitted.
37. A. Sahai. Non-Malleable Non-Interactive Zero Knowledge and Adaptive Chosen-Ciphertext Security. In *40th FOCS*, pages 543–553, 1999.
38. C.P. Schnorr. Efficient Identification and Signatures for Smart Cards. In *Crypto89*, Springer LNCS 435, pages 235–251, 1989.
39. J. Stern and D. Pointcheval. Security Arguments for Digital Signatures and Blind Signatures. *Jour. of Cryptology*, Vol. 13, No. 3, pages 361–396, 2000.