Simulation in Quasi-polynomial Time, and its Application to Protocol Composition

Rafael Pass

Department of Numerical Analysis and Computer Science,
Royal Institute of Technology, 100 44 Stockholm, Sweden.
rafael@nada.kth.se

Abstract. We propose a relaxation of zero-knowledge, by allowing the simulator to run in quasi-polynomial time. We show that protocols satisfying this notion can be constructed in settings where the standard definition is too restrictive. Specifically, we construct constant-round straight-line concurrent quasi-polynomial time simulatable arguments and show that such arguments can be used in advanced composition operations without any set-up assumptions. Our protocols rely on slightly strong, but standard type assumptions (namely the existence of one-to-one one-way functions secure against subexponential circuits).

1 Introduction

The ground-breaking notion of *zero-knowledge proofs*, i.e. proofs that yield no knowledge except the validity of the assertion proved, was introduced by Goldwasser, Micali and Rackoff [27] in 1982. Although the notion in itself is very beautiful, zero-knowledge is often not a goal but is rather used as a tool or technique for proving security of other cryptographic protocols. The definition of zero-knowledge is captured through the simulation paradigm. Namely, an interactive proof is said to be zero-knowledge if there exist a simulator that can simulate the behavior of every, possibly malicious, verifier, without having access to the prover, in such a way that its output is indistinguishable from the output of the verifier after having interacted with an honest prover. The logic behind the simulation paradigm is straightforward: Assuming that a malicious verifier succeeds in doing something after having interacted with a prover, then by running the simulator, he could have done it himself, without any interaction with a prover.

Nevertheless, it seems that current simulation techniques do not allow for advanced composition of protocols [22] [10]. The problem has been adressed in a positive way by Canetti [6], and Canetti and Fischlin [7] where the use of a stronger model, namely the Common Reference String model, was suggested to achieve zero-knowledge protocols that are universally composable [6]. Their approach is, however, not entirely problem free as was recently pointed out in [30]. In this paper we, instead, suggest a meaningful relaxation of the zero-knowledge definition that allows us to construct protocols in the plain model without any set-up assumption, on which advanced composition operations can be performed. We start by outlining the definitions and then motivate why they are meaningful.

Simulation in quasi-polynomial time. As the verifier in an interactive proof often is modeled as a probabilistic polynomially bounded turing machine, a PPT, the zero-knowledge definition requires that the simulator should be a PPT as well. Here, we weaken that specific assumption on the simulator, allowing it to be a probabilistic turing machine with running time bounded by $n^{poly(logn)}$, i.e. with quasi-polynomial running time. Loosely speaking, we say that an interactive proof is $n^{O(\sigma(n))}$-simulatable (or $n^{O(\sigma(n))}$-strongly simulatable) if there exist a simulator, running in time $n^{O(\sigma(n))}$, where n is the size of the statement being proved, whose output is indistinguishable, by polynomial size (or $n^{\sigma(n)}$-size) circuits, from a verifier's output after interacting with an honest prover.

Extraction in quasi-polynomial time. Another useful tool for proving security of cryptographic protocol is the concept of *proofs of knowledge*, introduced in [27]. A proof of knowledge is a proof whereby the verifier gets, not only convinced of the validity of the statement proved, but also that the prover has an (NP) witness to the statement. The definition is captured through the introduction of an extractor, i.e. a machine that is able to extract a witness from a prover that convinces an honest verifier. Thus a prover that convinces an honest verifier of a statement, could have run the extractor (on himself) and obtained a witness to the statement. Since the extractor models the power of the prover, and the prover usually is modeled by a PPT, the extractor is so as well. We weaken this assumption on the extractor and only require that extraction can be performed in time $n^{poly(logn)}$. Loosely speaking, we say that an interactive proof is $n^{O(\sigma(n))}$-witness extractable if there exist a probabilistic turing machine with running time bounded by $n^{O(\sigma(n))}$ that succeeds in extracting witnesses.

1.1 Practical motivation behind the proposed notions

Generalization. The intuitive meaning of $n^{O(\sigma(n))}$-simulatable proofs are proofs where the verifier might learn something from the prover, but not more than can be calculated by a machine running in time $n^{O(\sigma(n))}$. Our notion is thus a generalization of zero-knowledge where the "knowledge leaked" is quantified in a natural way.

Relaxation. Our work is in the vein of [19], where the notions witness indistinguishability and witness hiding were introduced. Like our notions, these notions are weaker than zero-knowledge, but are sufficient in many applications. Since our notions, however, still build on the standard simulation paradigm they can often be directly used to replace zero-knowledge/proof of knowledge protocols. In fact, the logic behind zero-knowledge and proofs of knowledge holds in many settings with $n^{O(\sigma(n))}$-simulatable, and extractable proofs, respectively. If a malicious verifier succeeds with a task after having interacted with an honest prover using an $n^{O(\sigma(n))}$-simulatable protocol, then there exists a $n^{O(\sigma(n))}$-time machine that succeeds with the task without interacting with the prover. In the case when $\sigma = poly(logn)$, this means that the task is solvable in quasi-polynomial time. Yet most natural problems that we believe are hard on average for polynomial time are also believed hard for quasi-polynomial time. In such a case, $n^{O(\sigma(n))}$-simulatable proofs can be directly used to replace zero-knowledge proofs. The situation is analogous in the case of proofs of knowledge. Proofs of knowledge are often used to show that if an adversary successfully convinces an honest verifier, then the adversary can find a solution to an intractable problem. Now, if using $n^{poly(logn)}$-extractable proofs, we arrive at a contradiction as well, assuming that the underlying problem is intractable for quasi-polynomial time.

Guarantee security in the On-line/Off-line model. In many settings it seems reasonable to assume that parties are given a certain on-line time and a certain, longer, off-line time. Such an *on-line/off-line model* can be modeled by letting parties to run in polynomial time while being on-line, and $n^{poly(logn)}$ time off-line.

An important property of $n^{poly(logn)}$-*strongly* simulatable arguments is that they are zero-knowledge in the on-line/off-line model. If a protocol is strongly quasi-polynomial time simulatable, that means that there exist an off-line simulator for every on-line verifier such that the output of of the simulator is indistinguishable in quasi-polynomial time, i.e. off-line, from the output of the verifier after interaction with a real on-line prover. Strongly quasi-polynomial time simulatable protocols thus guarantee that anything that a verifier can calculate after interaction with a prover, he could have calculated by himself off-line.

Allow for advanced composition. As zero-knowledge protocols normally are quite easy to construct there is not really a need for a relaxed notion in the synchronous setting. In asynchronous settings it seems harder to achieve efficient zero-knowledge protocols. Indeed, Canetti, Kilian, Petrank, Rosen [10] have shown the impossibility of constant-round black-box concurrent zero-knowledge, i.e. zero-knowledge under concurrent executions. In fact, it in general seems hard to compose protocols that are proven secure using standard rewinding techniques. We show how to construct constant-round protocols that are concurrent quasi-polynomial time simulatable without the use of rewinding, i.e. straight-line simulatable. We also show that such straight-line protocols can be used in fully asynchronous composition operations.

1.2 Theoretical motivation behind the proposed notions

In the standard model several strong impossibility results are known:

1. The impossibility of non-trivial straight-line simulatable expected-polynomial-time black-box zero-knowledge arguments. (folklore)
2. The impossibility of non-trivial 2-round expected-polynomial-time zero-knowledge arguments. (straight-forward extension, from proofs to arguments, of the result of [25])

3. The impossibility of non-trivial 3-round expected-polynomial-time black-box zero-knowledge arguments. [22]
4. The impossibility of non-trivial strict polynomial-time constant-round black-box zero-knowledge arguments and proofs of knowledge. [2]
5. The impossibility of non-trivial constant-round concurrent black-box zero-knowledge arguments. [10]

Our main theoretical motivation is to investigate if these impossibility results still are valid if relaxing the running time of the simulator/extractor to $n^{poly(log n)}$.[1] We show that all these impossibility result can be overcome when allowing the simulator/extractor to run in quasi-polynomial time.

1.3 Our contribution

On top of the definitional efforts, we demonstrate a powerful composition theorem for straight-line concurrent $n^{poly(log n)}$-simulatable arguments showing that in settings where proving the security of a fixed protocol, straight-line concurrent $n^{poly(log n)}$-simulatable arguments can be used as sub-protocols to guarantee security against concurrent and man-in-the-middle attacks. We note that it is the straight-line simulation property that allows us to achieve such a composition theorem.

Results. We construct a 2-round argument for $\mathcal{NP}$ that is straight-line concurrent black-box strict-$n^{poly(log n)}$-simulatable[2] and straight-line strict-$n^{poly(log n)}$-extractable, under the assumption that one-to-one one-way functions secure against subexponential circuits exists, and the existence of zaps [15] . We have thus shown that all the above mentioned impossibility results in the standard model no longer hold in our relaxed setting. We further show the impossibility of 2-round $n^{poly(log n)}$-simulatable proofs with negligible soundness error for languages that are not decidable in quasi-polynomial time.

On the practical side, we construct an *efficient* 4-round straight-line concurrent strict-$n^{poly(log n)}$-perfectly simulatable argument, under the assumption of one-to-one one-way functions secure against subexponential circuits, and perfectly hiding commitments. In analogy with perfect zero-knowledge, perfect simulation here means that the simulator's output has the same distribution as the verifier's output, after interaction with a real prover. The 4-round argument for $\mathcal{NP}$ is thus concurrent zero-knowledge in the on-line/off-line model. We also mention that the protocol is constructed through an efficient generic

[1] We note that even in the standard definition of zero-knowledge, the verifier is modeled as a strict polynomial time machine, whereas the simulator is allowed to run in expected polynomial time. Thus, already in the standard definition there is a slight gap between the power of the verifier and the power of the simulator. The same concerns are also valid in the case of proofs of knowledge.

[2] In this section we emphasize that our protocols are simulatable in *strict* $n^{poly(log n)}$-time, as opposed to expected time. In the rest of the paper we do not emphasize this fact.

transformation from 3-round special-sound public-coin honest-verifier perfect zero-knowledge arguments.

1.4 Related research

Our on-line/off-line model is quite similar to the timing model introduced by Dwork, Naor and Sahai [14] in the context of concurrent zero-knowledge. We mention that the concurrent zero-knowledge protocol presented in [14] relies on both *time-out* and *delay* mechanism, whereas our protocol only relies on *time-outs*, which drastically improves the efficiency. The on-line/off-line model, however, relies on stronger assumptions than the timing model as it explicitly bounds the on-line running time of malicious parties.

Recently, Dwork and Stockmeyer [13], investigated the possibility of 2-round zero-knowledge protocols in a model where the prover has bounded resources. The intuition and the structure of our 2-round protocol is similar to that of [13], however both the security definitions and the techniques used to instantiate the intuition are very different. Indeed the results of Dwork and Stockmeyer are quite limited in the setting where the prover's running time is bounded, while we are able to prove security under standard type assumptions. We note, however, that the definition used in [13] is more restrictive than that of simply quasi-polynomial time simulatable arguments.

Canetti et al have, in [9], used the technique of complexity leveraging. The proof of security of our 2-round protocol relies on the same technique.

2 Definitions and Preliminaries

2.1 One-way functions and hard-core predicates

Intuitively one-way functions are functions that are easy to compute, but hard to invert. Here "easy" means, achievable in polynomial time, and "hard" normally means not achievable in polynomial time. In this paper we rely on stronger assumptions than the most commonly used, namely we assume the existence of one-way functions where inverting the function is hard for subexponential circuits. The assumptions is, nevertheless, very plausible and has for example been used to construct resettable zero-knowledge in [9]. More formally, (borrowing notation from [20])

Definition 1. *A function* $f : \{0,1\}^* \to \{0,1\}^*$ *is called* one-way against 2^{n^κ} adversaries *if the following two conditions hold:*

- *Easy to compute: There exist a (deterministic) polynomial-time algorithm* A *such that on input* x*,* A *outputs* $f(x)$*.*
- *Hard to invert: For every probabilistic algorithm* A' *with running time bounded by* 2^{n^κ}*, every positive polynomial* p*, and all sufficiently large* n*'s, every auxiliary input* $z \in \{0,1\}^{poly(n)}$*,*

$$Pr[A'(f(U_n), z) \in f^{-1}(f(U_n))] < 2^{-n^\kappa}$$

where U_n is a random variable uniformly distributed in $\{0,1\}^n$.

Definition 2. *A function $f : \{0,1\}^* \rightarrow \{0,1\}^*$ is called* one-way against subexponential circuits *iff there exist a κ such that f is one-way against 2^{n^κ} adversaries.*

A predicate b is a called a hard-core of a function f if an adversary given $f(x)$ can guess $b(x)$ with success probability only negligibly higher than one half. Here again, the adversary is normally limited to polynomial time. We use a stronger definition, allowing the adversary to be a subexponential circuit:

Definition 3. *A polynomial-time-computable predicate $b : \{0,1\}^* \rightarrow \{0,1\}$ is called a* hard-core against 2^{n^κ} adversaries *of a function f if for every probabilistic algorithm A' with running time bounded by 2^{n^κ}, every positive polynomial p, and all sufficiently large n's, every auxiliary input $z \in \{0,1\}^{poly(n)}$,*

$$Pr[A'(f(U_n), z) = B(U_n)] < \frac{1}{2} + 2^{-n^\kappa}$$

where U_n is a random variable uniformly distributed in $\{0,1\}^n$.

Definition 4. *A polynomial-time-computable predicate $b : \{0,1\}^* \rightarrow \{0,1\}$ is called a* hard-core against subexponential circuits *of a function f if there exist a κ such that b is a hard-core against 2^{n^κ} adversaries for f.*

Goldreich and Levin [23] have shown that a simple hard-core predicate can be constructed assuming the existence of one-way functions. We note that the Goldreich-Levin predicate is also a hard-core against subexponential circuits of a function that is one-way against subexponential circuits.

Theorem 1 (Goldreich-Levin). *If there exist a one-way function $f : \{0,1\}^* \rightarrow \{0,1\}^*$ against subexponential circuits, then there exist a pair f',b', where $f' : \{0,1\}^* \rightarrow \{0,1\}^*$ is a one-way function against subexponential circuits, and $b' : \{0,1\}^* \rightarrow \{0,1\}$ is a hard-core predicate against subexponential circuits for f'. Furthermore, if f is one-to-one, then f' is so as well.*

2.2 Commitment schemes

We will give an informal definition of a commitment scheme. For a formal definition we refer the reader to [20]. Informally a commitment scheme between a PPT committer C and a PPT receiver R is a protocol in two phases, a commit phase and a reveal phase, such that C commits to a string (or bit) during the commit phase, that it can thereafter reveal (or "decommit to") during the reveal phase. The commitment scheme should also have the two properties: hiding and binding. The hiding property means that the receiver should have no knowledge of the string, that C has committed to, before the reveal phase. In other words, this means that there should not exist a non-uniform distinguisher that can distinguish between two commitments. The binding property means that the committer should not be able to successfully pass the commit phase and still be able to decommit to two different values during the reveal phase.

2.3 Witness indistinguishabilty

The notion witness indistinguishability was introduced by Feige and Shamir in [19] as a weaker alternative to zero-knowledge. It has later proved to be an excellent tool to achieve zero-knowledge [17], [18], [32], [1]. Intuitively an interactive proof of an $\mathcal{NP}$ relation, in which the prover uses one of several secret witnesses is witness indistinguishable if the verifier can not tell what witness the prover has used. We further say that an interactive proof is witness independent if the verifier's view is equally distributed independently of what witness the prover has used.

2.4 Zaps

Zaps, two round witness indistinguishable public-coin protocols, where the first message can be fixed once and for all, were introduced in [15] by Dwork and Naor. They presented the following construction of a zap for proving membership of x in the language L:
Suppose that there exist a non-interactive zero-knowledge proof for the language L using a CRS string consisting of l bits. Then the following protocol is a witness indistinguishable proof.

$V \rightarrow P$: Sends a random k-bit string $\rho = b_1 ... b_k$ which is interpreted as $B_1 ... B_m$, where B_i denotes the i'th block of l consecutive bits.
$P \rightarrow V$: The prover chooses and sends a random l-bit string $C = c_1 ... c_l$. For $j = 1$ to m the prover sends a non-interactive zero-knowledge argument that $x \in L$ using $B_j \oplus C$ as CRS string.

2.5 Proofs of knowledge

Informally an interactive proof is a proof of knowledge if there exists an oracle machine, a so called extractor, such that if a prover can convince the verifier with non-negligible probability, then the extractor having oracle access to the prover can extract a witness to the statement being proved in polynomial time. Proofs of knowledge are of special interest to us since the existence of an extractor directly implies soundness. See [20] for a formal definition.

In the following we will also be needing a restricted form of proofs of knowledge, namely special-sound proofs.

Definition 5. *Suppose Π is a three round interactive proof (or argument) for the language $L \in \mathcal{NP}$, with witness relation R_L. We say that the protocol Π is* special sound *if there exist a PPT extractor machine E, such that for all $x \in L$ and all pairs of accepting transcripts for proving x, $T_1 = (a, b_1, c_1), T_2 = (a, b_2, c_2)$, where $b_1 \neq b_2$, $E(T_1, T_2) \in R_L(x)$.*

It can be seen that if an interactive proof is special sound it is also a proof of knowledge. [12]

3 Definition of the relaxed notions and some consequences

As argued in the introduction, for many applications, it is often sufficient to use interactive proofs (or arguments) with simulators, or extractors, running in quasi-polynomial time.

3.1 Simulation

We start by defining $n^{O(\sigma(n))}$-simulatable interactive proofs.

Definition 6. *We say that an interactive proof (or argument)* (P, V) *for the language* $L \in \mathcal{NP}$*, with the witness relation* R_L*, is* $n^{O(\sigma(n))}$-simulatable *if there for every PPT machine* V^* *exists a probabilistic simulator* S *with running time bounded by* $n^{O(\sigma(n))}$ *such that the following two ensembles are computationally indistinguishable (when the distinguishing gap is a function in* $n = |x|$*)*

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

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* $y \in R_L(x)$ *and all auxiliary inputs* $z \in \{0,1\}^*$ *it holds that*

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

Notation: As in the literature, $\langle A, B\rangle(x)$ is the random variable representing B's output when interacting with machine A on common input x.

We note that in this definition, the distinguisher, as well as the distinguishing gap is polynomial, even though the simulator is allowed to run in time $n^{O(\sigma(n))}$. We show, in the composition theorem in the next section, that in many cases this is a sufficient requirement. Intuitively this stems from the fact that both the adversary, i.e. the malicious verifier, and parties that it will interact with, outside of the interactive proof, are polynomial time. Thus, an PPT adversary that succeeds with a protocol execution, communicating with PPT machines, after having interacted with a prover using a $n^{O(\sigma(n))}$-simulatable proof, could have done so by itself in time $n^{O(\sigma(n))}$.

We also introduce two stronger and more robust notions, namely *strongly simulatable*, and *perfectly simulatable* proofs. We say that an interactive proof (or argument) is $n^{O(\sigma(n))}$*-strongly simulatable* if the two above ensembles are indistinguishable by $n^{O(\sigma(n))}$-sized circuits, with distinguishing gap $n^{O(\sigma(n))}$. In analogy with standard definitions of perfect zero-knowledge, we say that an interactive proof (or argument) is $n^{O(\sigma(n))}$*-perfectly simulatable* if there exist a simulator that fails with probability smaller than $\frac{1}{2}$ and the two above ensembles have the same distribution, when the simulator's output is conditioned on not failing. It is an easy excercise to show that $n^{poly(logn)}$-perfectly simulatable proofs are $n^{poly(logn)}$-strongly simulatable.

$n^{poly(logn)}$-strongly simulatable arguments are trivially zero-knowledge in the on-line/off-line model introduced in the introduction [3]. In the on-line/off-line model all parties are allowed to run in polynomial time when being on-line, i.e. in communication with other parties, but are allowed to use $n^{poly(logn)}$ time when being off-line. If a protocol in such a setting is $n^{poly(logn)}$-strongly simulatable then it means that the verifier does not learn anything that it could not have learned by itself when being off-line.

The definitions can be restricted to straight-line simulators:

Definition 7. *We say that an interactive argument (proof) (P, V) for the language $L \in \mathcal{NP}$, with the witness relation R_L, is* straight-line $n^{O(\sigma(n))}$-simulatable *if there for every PPT machine V^* exists a probabilistic simulator S with running time bounded by $n^{O(\sigma(n))}$ such that the following two ensembles are computationally indistinguishable (when the distinguishing gap is a function in $n = |x|$)*

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

We note that the above definition is very restrictive. In fact, the simulator is supposed to act a cheating prover, with its only advantage being the possibility of running in time $n^{O(\sigma(n))}$, instead of polynomial time. Trivially, there therefore do not exist any straight-line $n^{O(\sigma(n))}$-simulatable proofs.

We also generalize the definition to guarantee security under concurrent runs:

Definition 8. *We say that an interactive argument (proof) (P, V) for the language $L \in \mathcal{NP}$, with the witness relation R_L, is* straight-line concurrent $n^{O(\sigma(n))}$-simulatable *if there for every PPT oracle machine A that is not allowed to restart or rewind the oracle it has access to, every polynomial $g(n)$, exists a probabilistic simulator $S(i, x)$ with running time bounded by $n^{O(\sigma(n))}$ such that the following two ensembles are computationally indistinguishable (when the distinguishing gap is a function in n)*

- $\{A^{P(x_1,y_1),P(x_2,y_2),..P(x_{g(n)},y_{g(n)})}(z, x_1, x_2, .., x_{g(n)})\}_{z\in\{0,1\}^*, x_1,x_2,..,x_{g(n)}\in L}$ *for arbitrary* $y_i \in R_L(x_i)$
- $\{A^{S(1,x_1),S(2,x_2),..S(g(n),x_{g(n)})}(z, x_1, x_2, .., x_{g(n)})\}_{z\in\{0,1\}^*, x_1,x_2,..,x_{g(n)}\in L}$

We end this section by noting that the notion of strongly simulatable proof (and also perfectly simulatable proofs) is more robust than that of simply simulatable proofs, as is shown in the following useful lemma:

Lemma 1. *If the interactive argument (P, V) is straight-line $n^{O(\sigma(n))}$-strongly simulatable (or perfectly simulatable), then it is also straight-line concurrent $n^{O(\sigma(n))}$-strongly simulatable (or perfectly simulatable).*

Proof. Let S be the straight-line simulator for (P, V). Then by a standard hybrid argument [26] it follows that $S'(i, x) = S(x)$ is a straight-line concurrent $n^{O(\sigma(n))}$-strong simulator for (P, V). ⋄

[3] This is not true for simply $n^{poly(logn)}$-simulatable arguments.

We note that the same proof can not be applied in the case of simply simulatable proofs, as the simulators in the hybrid argument will cause the distinguisher to run in time $n^{O(\sigma(n))}$.

3.2 Extraction

Analogously, we extend the definition of proofs of knowledge to $n^{O(\sigma(n))}$-witness extractable interactive proofs:

Definition 9. *We say that an interactive proof (or argument) with negligible soundness (P, V) for the language $L \in \mathcal{NP}$, with the witness relation R_L, is $n^{O(\sigma(n))}$-*witness extractable *if there for every PPT machine P^* exists a probabilistic witness extractor oracle machine E with running time bounded by $n^{O(\sigma(n))}$, 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, $E^{P^*_{x,y,r}}(x) \in R_L(x)$ 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.*

Notation: As in the literature, E^{P^*} signifies the machine E given oracle access to the machine P^*.

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 $n^{O(\sigma(n))}$-witness extractable* if the extractor finds a witness without rewinding the prover. In analogy with straight-line simulation, we here note that a straight-line extractor machine is, in fact, a cheating verifier running in time $n^{O(\sigma(n))}$ instead of in polynomial time.

We will, in the following, also be needing $n^{O(\sigma(n))}$-extractable commitment schemes, i.e. commitment scheme where an extractor running in time $n^{O(\sigma(n))}$ can extract the committed value. For simplicity we only state the definition for one-round commitment schemes.

Definition 10. *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 $n^{O(\sigma(n))}$-extractable if there exists a probabilistic extractor machine E with running time bounded by $n^{O(\sigma(n))}$, such that for all c, if C succeeds in decommiting to x with non-negligible probability, then $E(c) = x$ with overwhelming probability.*

4 Applications to general composition of protocols

In this section we show our main theorem regarding the use of straight-line concurrent $n^{poly(logn)}$-simulatable interactive arguments. This section formalizes and extends some of the intuition described in the motivation. Very loosely speaking the theorem states that when proving the security of a fixed protocol, straight-line concurrent $n^{poly(logn)}$-simulatable interactive arguments can be used as sub-protocols to provide security under concurrent, and man-in-the-middle attacks.

Suppose that we have a cryptographic system (an environment) that an adversary is trying to break. Suppose further that we are able to prove that a stand-alone adversary will not be able to break the system. We say that such an environment is hard. A central problem in cryptography is to guarantee that an adversary that is participating in other protocol executions simultaneously will still not be able to break the system.

Naively, one could expect that an adversary participating as a verifier of zero-knowledge proofs should not be able to break an environment that he is not able to break stand-alone. It is, however, known that the definition of zero-knowledge only guarantees security under sequential composition [22].

In fact, it is quite easy to see the problem can not be resolved in its most general form. Consider the environment consisting of the verifier of an interactive proof of knowledge for a hard-instance language. A stand-alone adversary that does not have a witness will not be able to succeed with such an environment. However, the man-in-the-middle adversary that simultaneously is participating as a verifier in a different execution of the same interactive proof will, by simply forwarding messages between the environment and the prover it is communicating with, succeed in breaking the environment.

Thus, in order to give a solution to the problem it is inevitable to, in some way, put restrictions on the environment. Recently a solution was proposed in [7] by resorting to the Common Reference String (CRS) model. The solution implicitly restricts the environments to those where CRS string are not reused. See [30] for further discussion. We here, instead, show that this problem can be resolved in the plain model, if restricting the environments to only those that are hard against $n^{O(\sigma(n))}$-adversaries.

We start by defining the notion of an environment. We see an environment as a system that an adversary is trying to break. The environment outputs 1 if the adversary succeeds and 0 otherwise. Intuitively, we say that an environment is hard if an adversary can not make the environment output 1, i.e. break the system. More formally,

Definition 11. *We say that an interactive PPT machine E, called environment, is* hard for the language L and the generator Gen_L against $n^{O(\sigma(n))}$-adversaries*, if for all interactive probabilistic machine A' with running time bounded by $n^{O(\sigma(n))}$, every $z \in \{0,1\}^{poly(n)}$*

$$Pr[x \leftarrow Gen_L, \langle A'(z), E\rangle(x) = 1]$$

is negligible as a function of n, where Gen_L is a machine that chooses an element $x \in L \cap \{0,1\}^n$ according to some distribution.

Our composition theorem now, informally, states that a PPT adversary, that is allowed concurrent access to different provers, communicating using straight-line concurrent $n^{O(\sigma(n))}$-simulatable interactive arguments, and an environment, will not be able to succeed in an attack against the environment if the environment is hard against $n^{O(\sigma(n))}$ adversaries. We note that since the environment is a polynomial time machine, it is enough that the arguments that the adversary is allowed to participate in are simply simulatable and not strongly simulatable, i.e. that the simulator's output only is indistinguishable in polynomial time instead of in time $n^{O(\sigma(n))}$. More formally,

Theorem 2. *Let Π be a straight-line concurrent $n^{O(\sigma(n))}$-simulatable interactive argument for the language L, such that $\sigma(n) \in \omega(1)$. Let p(n) be a polynomial, $Gen_{L'^{p(n)}}$ be a generator for $(L \cap \{0,1\}^n)^{p(n)}$, i.e. a machine that chooses an element $(x_1, x_2, ..x_{p(n)}) \in (L \cap \{0,1\}^n)^{p(n)}$ according to some distribution, such that the environment E is hard for the language $L^{p(n)}$ and the generator $Gen_{L^{p(n)}}$ against $n^{O(\sigma(n))}$-adversaries. Let $P(x)$ be the honest prover in Π for $x \in L$. Now let the adversary A be a PPT oracle machine that can not restart or rewind the oracle it gets access to. Then for all $z \in \{0,1\}^{poly(n)}$,*

$$Pr[\bar{x} = (x_1, x_2, .., x_{p(n)}) \leftarrow Gen_{L^{p}(n)}, y_i \in R_L(x_i),$$
$$\langle A^{P(x_1,y_1),P(x_2,y_2),..P(x_{g(n)},y_{g(n)})}(z), E\rangle(\bar{x}) = 1]$$

is negligible.

Informally, the theorem follows from the fact that a straight-line simulator is a cheating prover running in time $n^{O(\sigma(n))}$. If a man-in-the-middle attacker succeeds with a specific environment, then the attacker with access to the straight-line simulator running in time $n^{O(\sigma(n))}$, instead of a real prover, would succeed as well. See the full version of the paper for a formal proof.

The theorem shows that straight-line $n^{O(\sigma(n)}$-simulation is a sufficient condition for security when integrating a sub-protocol in an environment that is hard against $n^{O(\sigma(n)}$-adversaries. This yields an efficient way of constructing protocols with strong security properties by the use of telescopic composition of protocols, i.e. using protocols that are successively harder and harder. Indeed, the key to the theorem is the fact that the interactive arguments, that are run by the adversary trying to cheat an environment, are *easy* against $n^{O(\sigma(n))}$ adversaries (since they are straight-line $n^{O(\sigma(n))}$-simulatable) while the environment is hard against $n^{O(\sigma(n))}$ adversaries.

5 A two round simulatable and extractable argument

We combine $n^{poly(logn)}$-extractable commitments with the zaps of [15] to achieve a 2-round straight-line concurrent $n^{poly(logn)}$-simulatable and straight-line $n^{poly(logn)}$-witness extractable argument for $\mathcal{NP}$.

5.1 Extractable commitments under general assumptions

We start by showing that assuming the existence of one-to-one one-way functions against subexponential circuits, there exists $n^{poly(logn)}$-extractable commitment schemes. Specifically, we construct a $n^{poly(logn)}$-extractable bit commitment scheme using a modified version of Blum's commitments scheme [4] (using the notation of [20]). The idea behind the construction is to create commitments that are large enough to guarantee the hiding property against polynomial adversaries, but small enough for a quasi-polynomial adversary to perform a brute-force attack and thus extracting the commitments.

The construction. Let $f : \{0,1\}^* \rightarrow \{0,1\}^*$ be a function, and let $b : \{0,1\}^* \rightarrow \{0,1\}$ be a predicate.

Commit phase
To commit to value $v \in \{0,1\}$, the sender uniformly selects $s \in \{0,1\}^{log^k n}$ and sends the pair $(f(s), b(s) \oplus v)$.
Reveal phase
The sender reveals v and s. The receiver accepts if $f(s) = \alpha$ and $b(s) \oplus v = \beta$ where (α, β) is the receiver's view of the commit phase.

Lemma 2. *Let $f : \{0,1\}^* \rightarrow \{0,1\}^*$ be a one-to-one one-way function against subexponential circuits, and let $b : \{0,1\}^* \rightarrow \{0,1\}$ be a hard-core predicate against subexponential circuits for f, i.e. there exists a κ such that b is a hardcore against probabilistic non-uniform adversaries with running time bounded by 2^{n^κ}. Then for $k = \frac{1}{\kappa} + 1$, the protocol presented constitutes a $n^{O(log^k n)}$-extractable bit-commitment scheme.*

The proof of the lemma is given in the full version of the paper. We note that in order to construct this, and the following protocols, the constant κ needs to be known.

5.2 The protocol

We are now ready to show the protocol. The protocol builds on the Feige-Lapidot-Shamir construction [17], where a witness indistinguishable proof (argument) is used, in such a way that the simulator can perform the simulation using a "fake" witness. In our case, the verifier starts by sending a random image $b = f(r)$ through a one-to-one one-way function against subexponential adversaries f. The prover thereafter proves that he either has a witness to the statement x or that he has the pre-image to b. The size of b is chosen in such a way that a polynomial time adversary will not be able to find a pre-image to b, but a pre-image can be found in quasi-polynomial time by performing a brute-force attack. Now, intuitively, soundness follows from the proof of knowledge property. The simulator, on the other hand, is able to find a pre-image and can thus use it as a fake witness. The protocol follows:

Suppose that f is a one-to-one one-way function against adversaries running in time 2^{n^κ}, C is a committing machine for a commitment scheme extractable in time $n^{log^{k'} n}$, $k = \frac{1}{\kappa} + \frac{2k'}{\kappa}$. Now, consider the following protocol:

Protocol Π
V uniformly chooses $r \in \{0,1\}^{log^k n}$, $B \in \{0,1\}^{poly(n)}$

V $\rightarrow$ P: $b = f(r)$, B
P $\rightarrow$ V: $c = C(r'||w)$, a zap using B as randomness, showing the statement "$\exists r''$, w' s.t $c = C(r''||w') \wedge (b = f(r'') \vee w' \in R_L(x))$".

In the full version of the paper we show that Π is a straight-line concurrent $n^{O(log^k n)}$-simulatable, and straight-line $n^{O(log^{k'} n)}$-extractable argument. Thus,

Theorem 3. *Assuming the existence of one-to-one one-way functions against subexponential circuits, and the existence of zaps, there exist a two round interactive argument that is straight-line concurrent $n^{poly(logn)}$-simulatable and straight-line $n^{poly(logn)}$-extractable.*

Remark 1. Zaps can be constructed based on the existence of non-interactive zero-knowledge proofs in the common random string model, which in turn can be based on the existence of trapdoor permutations. [15]

5.3 On the round-complexity of quasi-polynomial time simulatable protocols

We do not know if there exists a one-round argument that is $n^{poly(logn)}$-simulatable. In a weakened random oracle model, as recently defined in [29], where the simulator is not allowed to choose the random oracle but should be able to perform a simulation for an overwhelming part of random oracles, we are, however, able to construct one-round expected-$n^{\omega(1)}$-time simulatable and extractable arguments without any further complexity assumptions by using a protocol very similar to the moderately hard function of [16][4]. This result does however *not* give us any concrete indication of the possibility of achieving a similar result in the standard model as the two models are incomparable [8].

Changing perspective and considering proofs instead of arguments, we show the impossibility of two-round $n^{poly(logn)}$-simulatable proofs with negligible soundness for languages that are not decidable in quasi-polynomial time:

Theorem 4. *If there exist a two-round interactive proof with negligible soundness for the language L that is $n^{poly(logn)}$-simulatable, then L is decidable in quasi-polynomial time.*

Proof. Recall the proof of the impossibility result for non-trivial two-round auxiliary-input zero-knowledge of Goldreich-Oren [25]. They show how that the simulator for the cheating verifier that simply forwards its auxiliary input as its message, can be used to decide the language. The same transformation can be used in our setting, yielding a distinguisher running in quasi-polynomial time, since our simulator runs in quasi-polynomial time. $\diamond$

We mention that the lower bound can be matched: Consider the 3-round interactive proof consisting of $log^2 n$ parallel repetitions of the graph hamiltonicity protocol of [5]. The protocol has both negligible soundness error, and is simulatable in quasi-polynomial time, by simple rewinding.

[4] We mention that the impossibility of non-trivial one-round zero-knowledge arguments using the strengthened definition of zero-knowledge in the random oracle model is proved in [29].

6 An efficient perfectly simulatable argument

In this section we show an efficient transformation from a 3-round special-sound public-coin honest-verifier perfect zero-knowledge argument into a 4-round straight-line $n^{poly(logn)}$-perfectly simulatable argument. The protocol is thus zero-knowledge in the on-line/off-line model. [5]

On a high level, the idea behind the protocol is very similar to the protocol described in section 5.2 with the main difference that a witness independent argument is used instead of a witness indistinguishable argument. Since we use 4 rounds instead of 2 we are also able to construct a more efficient protocol.

Let f be a one-to-one one-way function against adversaries running in time 2^{n^κ}. Let the witness relation $R_{L'}$, where $(x, y) \in R_{L'}$ if $f(x) = y$, characterize the language L'.

Let the language $L \in \mathcal{NP}$, and $k = \frac{1}{\kappa} + 1$. Consider the following protocol for proving that $x \in L$:

Protocol Π
V uniformly chooses $r \in \{0, 1\}^{log^k n}$.
V $\rightarrow$ P: $c = f(r)$
P $\leftrightarrow$ V: a 3-round witness independent argument of knowledge 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 3-round witness independent argument we start by noting that honest-verifier perfect zero-knowledge arguments are witness independent [11]. It is known that every language in $\mathcal{NP}$ has a special-sound public-coin honest-verifier perfect zero-knowledge argument (consider for example Blum's proof for the Hamiltonian Cycle problem [5], using perfectly hiding commitments). We can thereafter combine the argument for L and the argument for L' using the efficient OR-transformation of [11] yielding a special-sound honest-verifier zero-knowledge public-coin argument for $L \vee L'$ and the witness relation $R_{L \vee L'}$ [6]. We remark that if a specific one-way function is used, the protocol for proving knowledge of instances in L', i.e. the knowledge of a pre-image to f, can be implemented efficiently. Examples of such protocols are the Guillou-Quisquater scheme [28] for the RSA function, and the Schnorr scheme [33] for the discrete logarithm.

In the full version of the paper we show that Π is a straight-line concurrent $n^{O(log^k n)}$-perfectly simulatable argument. Thus,

[5] The 2-round protocol presented earlier is not zero-knowledge in the on-line/off-line model, since a witness to the statement proved can be straight-line extracted in quasi-polynomial time.

[6] Since the transformation in [11] uses that the second messages of the two protocols have the same length, we need to run several parallel versions of the protocol for L'. The resulting argument then uses less communication than the the argument for L plus the (parallelized) argument for L'.

Theorem 5. *Assuming the existence of one-to-one one-way functions against subexponential circuits, and the existence of perfectly hiding commitments, there exist a four round interactive argument of knowledge that is straight-line concurrent $n^{poly(log n)}$-perfectly simulatable.*

Remark 2. Perfectly hiding commitments can be constructed based on claw-free collections [21].

7 Extensions

If assuming one-to-one one-way functions against exponential circuits, our protocols can be modified in a straight-forward way to become $n^{\omega(1)}$-simulatable. In fact, if assuming provers that are computationally bounded below a specific polynomial $f(n)$, then our protocols can be modified to become simulatable in time $g(n)$, where $g(n) > f(n)$ is another polynomial. It would be interesting to extend this analysis to exact security.

8 Acknowledgments

I wish to thank Johan Håstad for his invaluable help and comments. Special thanks to Boaz Barak for suggesting the use of complexity leveraging in a similar way to [9] to achieve 2 round protocols. I am also very grateful to Alon Rosen for helpful comments. Finally, I would like to thank the anonymous referees.

References

1. Boaz Barak, "How to Go Beyond the Black-Box Simulation Barrier", 42nd FOCS, pp. 106-115, 2001
2. Boaz Barak, Yehuda Lindell "Strict polynomial-time in simulation and extraction", 34th STOC, pp. 484-493, 2002
3. Dan Boneh, Moni Naor, "Timed Commitments", Crypto00, Springer LNCS 1880, pp. 236-254, 2000
4. Manuel Blum, "Coin Flipping by Telephone", Crypto81, ECE Report 82-04, ECE Dept., UCSB, pp. 11-15, 1982
5. M. Blum, "How to prove a theorem so no one else can claim it", Proc. of the International Congress of Mathematicians, pp. 1444-1451, 1986
6. Ran Canetti, "Universally Composable Security: A New Paradigm for Cryptographic Protocols", 42nd FOCS, pp. 136-145, 2001
7. Ran Canetti, Marc Fischlin, "Universally Composable Commitments", Crypto01, Springer LNCS 2139, pp. 19-40, 2001
8. Ran Canetti, Oded Goldreich, Shai Halevi, "The Random Oracle Methodology, Revisited (Preliminary Version)", 30th STOC, pp. 209-218, 1998
9. R. Canetti, O. Goldreich, S. Goldwasser and S. Micali, "Resettable Zero-Knowledge", 32nd STOC, pp. 235-244, 2000
10. Ran Canetti, Joe Kilian, Erez Petrank, Alon Rosen, "Black-box concurrent zero-knowledge requires Omega (log n) rounds", 33rd STOC, pp. 570-579, 2001

11. Ronald Cramer, Ivan Damgård, Berry Schoenmakers, “Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols”, Crypto94, Springer LNCS 839, pp. 174-187, 1994
12. Y. Damgård, “Efficient Concurrent Zero-Knowledge in the Auxilliary String Model”, EuroCrypt00, Springer LNCS 1807, pp 418-430, 2000
13. Cynthia Dwork, Larry J. Stockmeyer, “2-round zero knowledge and proof auditors”, 34th STOC, pp 322-331, 2002
14. C. Dwork, M. Naor, A. Sahai, “Concurrent zero-knowledge ”, 30th STOC pp. 409-418, 1998
15. Cynthia Dwork, Moni Naor, “Zaps and Their Applications”, 40th FOCS, pp. 283-293, 2000
16. Cynthia Dwork, Moni Naor, “Pricing via Processing or Combatting Junk Mail”, Crypto92, Springer LNCS 740, pp. 139-147, 1992
17. U. Feige, D. Lapidot, A. Shamir, “Multiple Noninteractive Zero Knowledge Proofs under General Assumptions”, Siam J. Comput. 29(1), pp. 1-28, 1999
18. Uriel Feige, Adi Shamir, “Zero Knowledge Proofs of Knowledge in Two Rounds”, Crypto89, Springer LNCS 435, pp. 526-544, 1989
19. Uriel Feige, Adi Shamir, “Witness Indistinguishable and Witness Hiding Protocols”, 22nd STOC, pp. 416-426, 1990
20. O. Goldreich, “Foundations of Cryptography”, Cambridge University Press, 2001
21. Oded Goldreich, Ariel Kahan, “How to Construct Constant-Round Zero-Knowledge Proof Systems for NP”, Jour. of Cryptology 9(3), pp. 167-190, 1996
22. Oded Goldreich, Hugo Krawczyk, “On the Composition of Zero-Knowledge Proof Systems”, SIAM J. Comput. 25(1), pp. 169-192, 1996
23. Oded Goldreich, Leonid A. Levin, “A Hard-Core Predicate for all One-Way Functions”, 21st STOC, pp. 25-32, 1989
24. Oded Goldreich, Silvio Micali, Avi Wigderson, “Proofs that Yield Nothing But Their Validity or All Languages in NP Have Zero-Knowledge Proof Systems” , JACM 38(3), pp. 691-729, 1991
25. Oded Goldreich, Yair Oren, “Definitions and Properties of Zero-Knowledge Proof Systems”, Jour. of Cryptology 7(1), pp. 1-32, 1994
26. Shafi Goldwasser, Silvio Micali, “Probabilistic Encryption”, JCSS 28(2), pp. 270-299, 1984
27. Shafi Goldwasser, Silvio Micali, Charles Rackoff, “The Knowledge Complexity of Interactive Proof Systems”, SIAM J. Comput. 18(1), pp. 186-208, 1989. Preliminary version in STOC 1985. Earlier versions date to 1982.
28. Louis C. Guillou, Jean-Jacques Quisquater, “A Practical Zero-Knowledge Protocol Fitted to Security Microprocessor Minimizing Both Trasmission and Memory”, EuroCrypt88, Springer LNCS 330, pp. 123-128, 1988
29. Rafael Pass, “How to Use Random Oracles Properly”, Submitted.
30. Rafael Pass, “On the Limits of the Common Reference String Model, and Implications on Universal Composability”, Submitted.
31. Charles Rackoff, Daniel R. Simon, “Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack”, Crypto91, Springer LNCS 576, pp. 433-444, 1991
32. Ransom Richardson, Joe Kilian, “On the Concurrent Composition of Zero-Knowledge Proofs”, EuroCrypt99, Springer LNCS 1592 , pp. 415-431, 1999
33. C.P. Schnorr, “Efficient Identification and Signatures for Smart Cards”, Crypto89, Springer LNCS 435, pp. 235-251, 1989