On the Possibility of One-Message Weak Zero-Knowledge

Boaz Barak[1] and Rafael Pass[2]

[1] Institute for Advanced Study, Princeton, NJ ⋆⋆⋆
e-mail: boaz@ias.edu
[2] Royal Institute of Technology, Sweden. †
e-mail: rafael@nada.kth.se

Abstract. We investigate whether it is possible to obtain any meaningful type of zero-knowledge proofs using a *one-message* (i.e., *non-interactive*) proof system. We show that, under reasonable (although not standard) assumptions, there exists a one-message proof system for every language in **NP** that satisfies the following relaxed form of zero knowledge:

1. The soundness condition holds only against cheating provers that run in *uniform* (rather than non-uniform) probabilistic polynomial-time.
2. The zero-knowledge condition is obtained using a simulator that runs in *quasi-polynomial* (rather than polynomial) time.

We note that it is *necessary* to introduce both relaxations to obtain a one-message system for a non-trivial language. We stress that our result is in the plain model, and in particular we do *not* assume any setup conditions (such as the existence of a shared random string).
We also discuss the validity of our assumption, and show two conditions that imply it. In addition, we show that an assumption of a similar kind is *necessary* in order to obtain a one-message system that satisfies some sort of meaningful zero-knowledge and soundness conditions.

1 Introduction

The seminal 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 [15]. 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 idea behind this definition is the following: Assuming that a malicious verifier succeeds in doing something after having interacted with a

⋆⋆⋆ Supported by NSF grants DMS-0111298 and CCR-0324906. Most of this work done while studying in the Weizmann Institute of Science.

† Work done while visiting the Weizmann Institute of Science, Israel.

prover, then by running the simulator, he could have done it himself, without any interaction with a prover.

It has been shown that both *interaction* and *randomness* are necessary for zero-knowledge [14]. In this work, we investigate the possibility of a meaningful relaxation of zero-knowledge which does not require either interaction or randomness from the verifier. Somewhat surprisingly, we show that it *is* in fact possible to obtain a non-interactive proof system that satisfies a meaningful variant of zero-knowledge. Specifically, under reasonable (although non-standard) assumptions, for every $L \in \mathbf{NP}$, we construct a non-interactive system (P, V) (where V is a deterministic polynomial-time non-interactive algorithm) for proving membership in L that satisfies the following properties:

Perfect completeness For every $x \in L$ and w which is a witness for x, $V(x, P(x, w)) = 1$.

Soundness against Uniform Provers For every (possibly cheating) *uniform* probabilistic polynomial-time P^*, the probability that P^* outputs $x \notin L$ and a proof π such that $V(x, \pi) = 1$ is negligible. (Note that this is a relaxation of the standard soundness property for arguments, that require soundness against *non-uniform* polynomial-sized circuits.)

Quasi-polynomial time simulation There is a $n^{\mathrm{poly}(\log n)}$-time algorithm S such that for every $x \in L \cap \{0,1\}^n$, and w which is a witness for x, $S(x)$ is computationally indistinguishable (by polynomial-sized circuits) from $P(x, w)$. (Note that this is a relaxation of the standard zero-knowledge property, that requires simulation by a *polynomial-time* algorithm.)

Notes:

- As observed below, both relaxations are essential in order to obtain a non-interactive proof system for non-trivial languages. There do exist stronger models such as the Common Reference String (CRS) Model [4] where one-message zero-knowledge proofs and arguments can be constructed without these relaxations. However, in this paper we concentrate on the *plain* model, (i.e., without any set-up assumptions or random oracles).
- The quasi-polynomial time condition can be replaced with $T(n)$-time where $T(\cdot)$ can be any super-polynomial function.[3] In this paper, for simplicity, we restrict ourselves to quasi-polynomial time simulation. We note that if one allows larger simulation-time, one can obtain a one-message zero-knowledge argument under quantitatively weaker assumptions than the ones we use. We observe below that to obtain one-message systems, it is essential that the running time of the simulator be longer than the running time allowed to a cheating prover.
- As in the case of uniform (i.e., non-auxiliary input) zero-knowledge, the uniform soundness property is highly problematic when such a proof system

[3] However, note that if $T(n)$ is larger than the time it takes to compute a witness from a statement $x \in L \cap \{0,1\}^n$ then there is a trivial $T(n)$-time simulator that works as long as the system is witness indistinguishable.

is used as a subprotocol of a larger system. Also, the assumptions we use are somewhat non-standard, and so haven't been extensively studied. Therefore, we believe that this result serves more to clarify the boundaries of what can and cannot be done in zero-knowledge, than to provide a new practical proof system.

- As we show in Section 5, the non-standard assumption we use is essentially necessary to obtain a non-interactive zero-knowledge argument, even when allowing the two relaxations that we make.

1.1 Related works

Several relaxations of zero-knowledge have been suggested in the literature:

Witness Indistinguishability. The notion of *witness indistinguishability* was introduced by Feige and Shamir [12] as a relaxation of zero-knowledge. Intuitively, a witness indistinguishable proof is a proof where the view of the verifier is oblivious to the witness the honest prover uses. Recently the existence of one-message witness indistinguishable proofs with deterministic verifier was shown, under complexity theoretic assumptions [2]. Their result shows that, so called, **NP**-proofs, i.e. one-message proofs with deterministic verifiers, can be used to achieve certain security properties also for the prover.

Zero-knowledge arguments. Brassard, Chaum, and Crépeau [5] introduced the notion of *argument* systems, which is a relaxation of the [15] notion of *proof* systems. In an argument system, it may be possible for a cheating prover to convince the honest verifier of a false statement, but only if it makes use of a strategy that cannot be feasibly computed. The usual definition of "feasible computation" is computation by a *non-uniform* circuit family. We note that for one-message systems, this condition is equivalent to the definition of *proof* systems, since if there *exists* a prover message that can convince the verifier of a false statement, a non-uniform prover strategy can have this message "hard-wired" in to it. In this paper, we define "feasible computation" as computation by a *uniform* probabilistic polynomial-time Turing machine.

Weak Zero-knowledge. Recently *simulation in quasi-polynomial time* was explicitly proposed as a meaningful relaxation of zero-knowledge [19]. The notion of quasi-polynomial time simulatability implies that a malicious verifier will only be able to succeed in tasks that are easy for quasi-polynomial time after having interacted with a prover. Intuitively, quasi-polynomial time simulatable proofs only "leak" information that could be calculated in quasi-polynomial time. Since in most applications, the simulation condition is not the desired end result, but rather the means to prove the security of protocols,[4] it turns out that quasi-polynomial simulation suffices for most applications of zero-knowledge, provided

[4] An interesting exception to this rule is the case of deniable authentication [18, 8].

one is willing to make quantitatively stronger hardness assumptions. In the following we call proof systems that are simulatable in quasi-polynomial time *weak zero-knowledge.*[5]

Zero-knowledge with resource-bounded provers. Dwork and Stockmeyer investigate the possibility of two-round zero-knowledge proofs for provers that are resource-bounded (to, say, running time n^5) during the execution of the protocol [10]. Their relaxation of zero-knowledge proofs is somewhat orthogonal to ours. Whereas their definition considers a weaker form of adversaries (namely adversaries that are resource-bounded during the execution of the protocol), we consider a weaker form of zero-knowledge. Both relaxations have in common that the simulator is given a longer running time than the allowed running time of a cheating prover. We note that, as was observed in [10], one-message zero-knowledge proofs can not be obtained for time-bounded provers.

1.2 Impossibility results

Goldreich and Oren [14] showed that any auxiliary input zero-knowledge (i.e., a system that is zero-knowledge with respect to non-uniform verifiers) proof or argument system for a non-trivial language must have at least three rounds of interaction. Recently, Barak, Lindell and Vadhan [1] showed that, under certain computational assumptions, even uniform zero-knowledge perfect-completeness *proof* systems for **NP** must have at least three rounds of interactions. It can also be shown that (under reasonable computational assumptions) it is impossible to obtain one-message zero-knowledge proofs even if both the zero-knowledge and the soundness conditions are required to hold only with respect to uniform algorithms.[6] Thus to obtain one-message proof systems, one needs to allow the simulator to run in time which is long enough to break the soundness of the system (which we indeed do). As mentioned above (Section 1.1), this implies that the soundness property cannot hold against polynomial-sized non-uniform provers (since the existence of *any* cheating prover implies the existence of a *polynomial-sized* such prover).

1.3 On the Cryptographic Assumptions Used

Our construction relies on three assumptions:

Assumption 1 *There exists a one-message (i.e., non-interactive) WI proof system for every language $L \in$ **NP**.*

Recently, Barak, Ong and Vadhan [2] showed that such a system exists if there exist trapdoor permutations, and if $\mathbf{E} = \mathbf{Dtime}(2^{O(n)})$ contains a function of non-deterministic circuit complexity $2^{\Omega(n)}$. (See [2] for a discussion on

[5] Note that the notion of weak zero-knowledge used in this paper is different from the notion of weak zero-knowledge previously used in the literature (e.g. [16]).

[6] This can be proven in essentially the same way as the proof of Theorem 3.

the validity and reasonableness of this second condition). The protocol of [2] is obtained by derandomizing the Zaps construction of Dwork and Naor [9].[7]

Assumption 2 *There exists a non-interactive perfectly binding and computationally hiding commitment scheme, such that given a commitment $C(x)$, the plaintext x can be computed by a $n^{\log^c n}$-time algorithm, where n is the security parameter and c is some constant.*

Such a commitment can be constructed based on the existence of one-way permutations with subexponential hardness (using the well known commitment scheme of Blum [3] with a scaled-down security parameter, see [19] for more details). Alternatively, such a commitment scheme can be based on the assumption that there exists a subexponentially hard one-way *function*, and that $\mathbf{E} = \mathbf{Dtime}(2^{O(n)})$ contains a function of non-deterministic circuit complexity $2^{\Omega(n)}$, using the commitment scheme constructed by [2].

Assumption 3 *There exists a language $\Delta \in \mathbf{P}$ and constants $c_1 < c_2$ such that*

Δ is hard to sample in time $n^{\log^{c_1} n}$: *For every probabilistic $n^{\log^{c_1} n}$-time algorithm A, the probability that $A(1^n) \in \Delta \cap \{0,1\}^n$ and is negligible.*

Δ is easy to sample in time $n^{\log^{c_2} n}$: *There exists a $n^{\log^{c_2} n}$ algorithm S_Δ such that for every $n \in N$, $\Pr[S_\Delta(1^n) \in \Delta \cap \{0,1\}^n] > 1 - \mu(n)$, where $\mu(\cdot)$ is a negligible function (i.e., $\mu(n) = n^{\omega(1)}$).*[8]

As far as we are aware, this assumption is new, and therefore needs to be justified. We discuss its validity in Section 4.

2 Definitions and Preliminaries

Witness relations. Recall that a language L is in $\mathbf{NP}$ if there exists a polynomially-bounded and polynomial-time decidable relation R_L such that $L = \{x \mid \exists y \text{ s.t. } (x,y) \in R_L\}$. We call R_L the *witness relation* of L. We define $L(x)$ to be 1 if $x \in L$ and 0 otherwise.

Interactive proofs and arguments. We will use the notion of *interactive proofs* [15] (see [13] for the definitions). Interactive *arguments* [5] are defined in analogy with interactive proofs, with the only difference that the soundness condition only needs to hold against provers that can be implemented by a polynomial-sized circuit. A *uniform-soundness argument* is defined in an analogous way, where the soundness condition only needs to hold against provers that can be implemented by a *uniform* probabilistic polynomial-time Turing machine.

[7] As noted in [9], Zaps can, in fact, be seen as a *non-constructive, non-uniform* one-message witness indistinguishable proof (i.e., the honest prover and verifier algorithm are implemented by *non-uniform* circuits). Nevertheless, since we are interested in giving a constructive protocol in the plain model, without a shared random string or non-uniformity, we need to rely on the protocol of [2].

[8] Because $\Delta \in \mathbf{P}$, the probability of success can be amplified, and so this term can be replaced with anything between $1/\text{poly}(n)$ and $1 - 2^{-\text{poly}(n)}$.

Weak Zero-knowledge. Recall the standard notion of *zero-knowledge* proofs [15] (See [13] for exact definitions). We will use the following weaker form of zero-knowledge, following [19]:

Definition 1 *We say that an interactive proof (or argument)* (P, V) *for the language* $L \in \mathbf{NP}$*, with the witness relation* R_L*, is* $T(n)$-simulatable *if there for every probabilistic polynomial-time machine* V^* *exists a probabilistic simulator* S *with running time bounded by* $T(n)^{O(1)}$ *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*

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

We say that an interactive proof (or argument) is *weakly zero-knowledge* if it is $n^{\text{polylog}(n)}$-simulatable.

Remark 1. Note that the definition used only requires that the output of the simulator is indistinguishable by *polynomial*-sized circuits (as opposed to the quasi-polynomial running time of the simulator).

Extractable commitment scheme. As mentioned above, we define an *extractable commitment scheme* to be a (perfectly binding and computationally hiding) non-interactive commitment scheme, such that it is possible to extract the plain-text from the commitment scheme, in time $n^{\text{polylog}(n)}$.

Witness indistinguishable proof systems. A *witness indistinguishable* (WI) proof system [12] for a language L with witness relation R_L, is a proof system such that for every $x \in L$ and $w, w' \in R_L$, it is infeasible to distinguish between the view of any polynomial-sized verifier when interacting with the honest prover that gets w as auxiliary input, and between its view when it interacts with the honest prover that gets w' as auxiliary input. As mentioned above, we assume that there exists a one-message WI proof system for every $L \in \mathbf{NP}$.

3 One-message Weak Zero-Knowledge Argument for NP

In this section we show a construction of a one-message weak zero-knowledge argument for **NP** with uniform soundness.

The protocol which follows the Feige-Lapidot-Shamir paradigm [11], can be viewed as a derandomization of the two-round quasi-polynomial-time simulatable protocol of [19]. In order to do so we rely on the one-message witness indistinguishable protocol of [2].

3.1 The Protocol

Let $\varDelta$ be a language in $\mathbf{P}$ that is hard to sample in probabilistic time $n^{\log^{c_1} n}$, but easy to sample in time $n^{\log^{c_2} n}$ (where $c_1 < c_2$). Let Com be a commitment scheme extractable by a time $n^{\log^{c_0} n}$ algorithm, where we scale the parameters in such a way that $c_0 < c_1$. We define the following protocol:

Protocol $\varPi$ - One-message Weak ZK Argument for NP

Common Input: an instance x of a language L with witness relation R_L, 1^n: security parameter (we assume without loss of generality that both the witness size and the statement size are of length n).

The protocol: P $\rightarrow$ V: $\sigma = Com(0^n)$, a one-message WI argument z showing the statement

Either $x \in L$ *or* σ is a commitment to a member of $\varDelta$

More formally, the statement proven is that either $x \in L$ or that there exists y, r such that $\sigma = Com(y; r)$ and $y \in \varDelta$.

We have the following theorem:

Theorem 1 *Under Assumptions 1, 2 and 3, Protocol $\varPi$ is a one-message weak zero-knowledge argument with uniform soundness for* **NP**.

Proof We show that the above protocol in both sound against uniform probabilistic polynomial-time and simulatable in quasi-polynomial time.

Soundness. Let us start by the soundness condition. We prove this using complexity leveraging [6]. Assume, for contradiction, that there exist a uniform probabilistic machine P^* that produces an accepting proof c, z for a statement $x \notin L$. Let y be the plaintext that is committed to by c. By the perfect soundness condition of the WI system, either $x \in L$ or y is a member of $\varDelta$. Since the protocol uses extractable commitments, there exist a machine E that can extract y in time $n^{\log^{c_0} n}$. Furthermore, since $x \notin L$, it must hold that $y \in \varLambda$. Combining E with the prover P^*, we obtain a uniform machine that outputs a member of $\varDelta$ in time less than $n^{\log^{c_1} n}$, contradicting the hard to sample condition of $\varLambda$.

Simulation. Now, let us turn to quasi-polynomial time simulation. On input x, the simulator will obtain a member $y \in \varLambda$ in time $n^{\log^{c_2} n}$, compute a commitment σ to y and then prove in the WI system the true statement that either $(x, y) \in R_L$ or $y \in \varLambda$. It remains to show that the output of the simulator is indistinguishable from the output of the honest prover. This is done through a standard hybrid argument. That is, for every $(x, w) \in R_L$, we consider an intermediate hybrid $H = \{Com(y), z\}$ where y is the member of Λ obtained by the simulator, but z is a WI proof computed of the combined statement using the witness w for the fact that $x \in L$. The hybrid H is computationally indistinguishable from the simulator's output by the hiding property of the commitment scheme, and is computationally indistinguishable from the honest prover's output by the WI property of the WI system. ∎

Remark 2. We note that the output of the simulator is only polynomial-time indistingushable from a valid transcript. By using quantitatively stronger assumptions, such as the existence of WI proofs, where indistinguishability is guaranteed against quasi-polynomial time, the output of the simulator can be made indistinguishable for time $T'(n) = n^{\log^c n}$, for some constant c. Note, however, that in order to prove soundness, we require that the running time $T'(n)$ of the distinguisher is strictly smaller than the running time of the simulator. It is an interesting open problem to come up with a construction (under standard/reasonable assumptions) that allows running time of the distinguisher to be greater than the running time of the simulator.

4 On the New Complexity Theoretic Assumption

In this section we discuss the new complexity theoretic assumption that we use (Assumption 3). We show that Assumption 3 is implied by two different assumptions. Furthermore, in Section 5 we show that a variant of Assumption 3 is *necessary* to obtain a one-message weak zero-knowledge uniform-soundness argument.

4.1 Basing Assumption 3 on Uniform Hash Functions

In this section, we observe that Assumption 3 is implied by the existence of a hash function that is collision resistant against subexponential-time uniform algorithms. That is, if there exists a function H (computed by a polynomial-time algorithm) and a constant $\epsilon > 0$ such that $|H(x)| = \frac{|x|}{2}$, but for every 2^{k^ϵ} algorithm A, the probability that A outputs a pair $x \neq x' \in \{0,1\}^k$ such that $H(x) = H(x')$, is negligible. Note that H is a single function, and not a collection of functions, and so a non-uniform circuit *will* be able to output such a collision.

Define $\Lambda = \{(1^n, x, x') \mid x \neq x' \in \{0,1\}^{\log^{2/\epsilon} n}$ and $H(x) = H(x')\}$, and let $k = \log^{2/\epsilon} n$. We see that if A is an algorithm that runs in time less than $2^{k^\epsilon} = 2^{\log^2 n} = n^{\log n}$, then A will not be able to output a member of Δ. On the other hand, one can output a member of Δ by running the trivial collision finding algorithm that runs in time $2^k = n^{\mathrm{polylog}(n)}$.

We note that one candidate for such a uniform hash function may be obtained from the AES cipher [7], since (unlike DES), it uses algebraic components that can be scaled to arbitrarily large input lengths.

4.2 Basing Assumption 3 on the Hardness of $\mathbf{NE} \cap \mathbf{coNE}$

In this section, we show that Assumption 3 is implied by the existence of a *unary* language L in $\mathbf{NP} \cap \mathbf{coNP}$ that is hard for subexponential-time algorithms. Note that we only require *worst-case* hardness.[9] However, we do require that for every subexponential algorithm, the set of input lengths, for which the algorithm fails to decide the language, will be sufficiently "dense" in the sense that for every such algorithm A, and every large enough $n \in \mathrm{N}$, there exists $k \in (2^n, 2^{n+1}]$ such that $A(1^k)$ is different from $L(1^k)$. An equivalent way to formalize this requirement, is that there exists a (binary) language L in $\mathbf{NE} \cap \mathbf{coNE}$ (where $\mathbf{NE} = \mathbf{Ntime}(2^{O(n)})$ is the class of all languages decidable in non-deterministic exponential-time) that is worst-case hard for *doubly exponential-time* algorithms, in the sense that for every such algorithm A, and every large enough $n \in \mathrm{N}$, there exists an input $x \in \{0,1\}^n$ such that $A(x) \neq L(x)$. Thus, this can be looked up as a "scaling up" of the assumption that $\mathbf{NP} \cap \mathbf{coNP} \not\subseteq \mathbf{SUBEXP}$ (where $\mathbf{SUBEXP} = \cap_{\epsilon>0}\mathbf{Dtime}(2^{n^\epsilon})$ is the class of all languages having a subexonential algorithm).[10]

Theorem 2 *Suppose that there exists a unary language $L \in \mathbf{NP} \cap \mathbf{coNP}$ and $\epsilon > 0$ such that for every 2^{n^ϵ}-time probabilistic algorithm A, and every sufficiently large $i \in N$, there exists $k \in (2^i, 2^{i+1}]$ such that $A(1^k) \neq L(1^k)$.*

Then, there exists a hard-to-sample language Λ.

Proof Sketch: Let L be the assumed language, and assume (using padding if necessary) that for every k the witness, that 1^k is a member, or is not a member of L, is of length k. We define the language Λ in the following way: the tuple $\langle 1^m, 1^i, w_{2^i+1}, b_{2^i+1}, w_{2^i+2}, b_{2^i+2} \ldots, w_{2^{i+1}}, b_{2^{i+1}} \rangle$ is in Λ if

1. $i = \log(\log^{3/\epsilon} m)$
2. For every $k \in (2^i, 2^{i+1}]$, w_k is a witness that $L(1^k) = b_k$.

Firstly, note that Λ is indeed in $\mathbf{P}$. Also note, that an element of Λ can be obtained by finding each of the 2^i witnesses using exhaustive search (taking at most $2^{2^{i+1}}$ steps which is poly-logarithmic in m.)

Finally, we claim that every $m^{\log m}$-time algorithm A will fail to output a member of Λ starting with 1^m for all (sufficiently large) m's.[11] Indeed, any such algorithm can be converted into an 2^{n^ϵ}-time decision procedure B for the original language L in the following manner: On input 1^k, Algorithm B will find i such

[9] Unfortunately, there is no known complete language for $\mathbf{NP} \cap \mathbf{coNP}$, which means that, unlike the case in [17] and [2], we do not know of a fixed language $L_0 \in \mathbf{NP} \cap \mathbf{coNP}$ that satisfies this condition, as long as *some* language L satisfies it.

[10] Note that we assume hardness with respect to *probabilistic* algorithms. However, under standard complexity assumptions, probabilistic algorithms are equivalent to deterministic algorithms (c.f., [17]).

[11] Note that formally, A's job is to output a member of $\Lambda \cap \{0,1\}^m$. However, since any member of Λ starting with 1^m is of length $m + \mathrm{polylog}(m)$ (and this length is a fixed function of m), these two conditions are equivalent.

that $k \in (2^i, 2^{i+1}]$ and compute m such that $m = 2^{(2^i)^{\epsilon/3}}$. Then, it will run A to obtain a member $\langle 1^m, 1^i, w_{2^i+1}, b_{2^i+1}, \ldots, w_{2^{i+1}}, b_{2^{i+1}} \rangle$ of Λ, and then output b_k. Note that this takes at most $m^{logm} = 2^{\log^2 m}$ steps which is less than 2^{k^ϵ} steps. ■

Remark 3. Another condition that implies Assumption 3 is the existence of a language in $\mathbf{NE} = \mathbf{Dtime}(\mathbf{2^{O(n)}})$ that is hard on the average, in the sense that any doubly-exponential algorithm will succeed on at most a $\frac{1}{2} + \delta$ fraction of the inputs (with $\delta < \frac{1}{6}$). Loosely speaking, given such a language L, one can define a language Λ of witnesses for a $\frac{1}{2} - \delta$-fraction of the inputs of a particular length (note at least $\frac{1}{2} - \delta$-fraction of the inputs of any length must belong to L for it to be hard on the average). An algorithm to sample a member of Λ can be converted into an algorithm that decides L with a better than $\frac{1}{2} + \delta$ advantage. Again, this is equivalent to the existence of a hard on the average *unary* language in **NP**.

5 On the Necessity of the Assumption

In this section we show that the existence of one-message weak zero-knowledge arguments for **NP** implies a slightly weaker variant of Assumption 3.

Theorem 3 *Suppose that there exist one-to-one one-way functions hard against quasi-polynomial-time algorithms and that there exists a one-message weak zero-knowledge argument with uniform soundness for every $L \in$* **NP***. Then, there exists a language Λ that is hard to sample by polynomial-time algorithms, and that can be sampled by a quasi-polynomial-time algorithm.*

Before proving this theorem, note that its conclusion is only weaker from Assumption 3 in that that the language is hard to sample by polynomial-time algorithms, and not by $n^{\log^{c_1} n}$-time algorithms.

Proof Sketch: Let f be a one-to-one one-way function, and let h be its hard-core bit [20]. We define the following **NP** language L: $L = \{(f(x), h(x)) \mid x \in \{0,1\}^*\}$. Under the assumptions of the theorem, there exists a one-message weak zero-knowledge uniform-soundness argument system for L. Let V be the verifier algorithm for this system. We define the language Λ as follows

$$\Lambda = \{(y, b, \pi, x) \mid y = f(x), b \neq h(x), V(y, b, \pi) = 1\}$$

that is, Λ is the language of “false proofs” (i.e. proofs for false statements that pass verification). Clearly, the uniform soundness condition of the zero-knowledge system implies that it is infeasible for uniform probabilistic-time algorithms to sample a member of Λ. However, we claim that there is a $n^{\mathrm{polylog}(n)}$-time algorithm A to sample a member of Λ. On input 1^n, Algorithm A will choose x at random from $\{0,1\}^n$, and b at random from $\{0,1\}$, and output $(f(x), b, \pi, x)$ where π is obtained by applying the simulator of the system to the statement (y, b). We claim that

1. The probability that $V(f(x), b, \pi) = 1$ is very close to 1. Indeed, otherwise, the simulator combined with the verifier will be a distinguisher between the distribution $(f(x), b)$ and the distribution $(f(x), h(x))$.
2. The probability that $b \neq h(x)$ is equal to $\frac{1}{2}$ (since the choice of b is independent from the choice of x).

We see that A outputs a member of Λ with probability very close to $\frac{1}{2}$. Since membership in Λ can be verified, this probability can be amplified to $1 - 2^{\Omega(n)}$. (Actually, under computational assumptions, this can be derandomized and so A can output a member of Λ with probability 1.) ∎

Acknowledgments

We wish to thank Johan Håstad, Oded Goldreich, and Avi Wigderson for helpful discussions.

References

1. B. Barak, Y. Lindell and S. Vadhan. Lower Bounds for Non-Black-Box Zero-Knowledge. In *44th FOCS*, 2003.
2. B. Barak, S.J. Ong and S. Vadhan. Derandomization in Cryptography. In *Crypto2003*, Springer LNCS 2729, pages 299–315, 2003.
3. M. Blum. Coin Flipping by Telephone. In *Crypto81*, ECE Report 82-04, ECE Dept., UCSB, pages 11–15, 1982.
4. M. Blum, P. Feldman and S. Micali. Non-Interactive Zero-Knowledge and Its Applications. In *20th STOC*, pages 103–112, 1988.
5. G. Brassard, D. Chaum and C. Crépeau. Minimum Disclosure Proofs of Knowledge. *JCSS*, Vol. 37, No. 2, pages 156–189, 1988. Preliminary version by Brassard and Crépeau in *27th FOCS*, 1986.
6. R. Canetti, O. Goldreich, S. Goldwasser and S. Micali. Resettable Zero-Knowledge. In *32nd STOC*, pages 235–244, 2000.
7. J. Daemen and V.Rijmen. The Design of Rijndael: AES – The Advanced Encryption Standard Springer, ISBN 3-540-42580-2, 2002.
8. C. Dwork, M. Naor and A. Sahai. Concurrent Zero-Knowledge. In *30th STOC*, pages 409–418, 1998.
9. C. Dwork and M. Naor. Zaps and Their Applications. In *41th FOCS*, pages 283–293, 2000.
10. C. Dwork and L. Stockmeyer. 2-Round Zero Knowledge and Proof Auditors. In *34th STOC*, pages 332–331, 2002.
11. U. Feige, D. Lapidot and A. Shamir. Multiple Noninteractive Zero Knowledge Proofs under General Assumptions. *SIAM Jour. on Computing*, Vol. 29(1), pages 1–28, 1999.
12. U. Feige and A. Shamir. Witness Indistinguishability and Witness Hiding Protocols. In *22nd STOC*, pages 416–426, 1990.
13. O. Goldreich. *Foundations of Cryptography – Basic Tools.* Cambridge University Press, 2001.
14. O. Goldreich and Y. Oren. Definitions and Properties of Zero-Knowledge Proof Systems. *Jour. of Cryptology*, Vol. 7, No. 1, pages 1–32, 1994.

15. S. Goldwasser, S. Micali and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Jour. on Computing*, Vol. 18(1), pages 186–208, 1989.
16. O. Goldreich, S. Vadhan, A. Sahai. Honest Verifier Statistical Zero-Knowledge Equals General Statistical Zero-Knowledge. In *30th STOC*, pages 3999–408, 1998.
17. R. Impagliazzo, A. Wigderson. **P** = **BPP** if **E** requires exponential circuits: Derandomizing the XOR lemma. In *29th STOC*, pages 220–229, 1997.
18. M. Naor. Deniable Ring Authentication In *Crypto2002*, Springer LNCS 2442, pages 481–498, 2002.
19. R. Pass. Simulation in Quasi-polynomial Time and its Application to Protocol Composition. In *EuroCrypt2003*, Springer LNCS 2656, pages 160–176, 2003.
20. O. Goldreich, L. A. Levin. A Hard-Core Predicate for all One-Way Functions. In *21st STOC*, pages 25–32, 1989.