Composition of Zero-Knowledge Proofs with Efficient Provers*

Eleanor Birrell† Salil Vadhan‡

February 18, 2010

Abstract

We revisit the composability of different forms of zero-knowledge proofs when the honest prover strategy is restricted to be polynomial time (given an appropriate auxiliary input). Our results are:

1. When restricted to efficient provers, the original Goldwasser–Micali–Rackoff (GMR) definition of zero knowledge (STOC '85), here called *plain zero knowledge*, is closed under a constant number of sequential compositions (on the same input). This contrasts with the case of unbounded provers, where Goldreich and Krawczyk (ICALP '90, SICOMP '96) exhibited a protocol that is zero knowledge under the GMR definition, but for which the sequential composition of 2 copies is not zero knowledge.
2. If we relax the GMR definition to only require that the simulation is indistinguishable from the verifier's view by uniform polynomial-time distinguishers, with no auxiliary input beyond the statement being proven, then again zero knowledge is not closed under sequential composition of 2 copies.
3. We show that auxiliary-input zero knowledge with efficient provers is not closed under *parallel* composition of 2 copies under the assumption that there exist nontrivial languages in $\mathcal{NP}$ with uniquely determined, secret feature functions. This result generalizes the previous work by Feige and Shamir (STOC '90), who gave similar results under the assumption that (a) the discrete logarithm problem is hard, or (b) $\mathcal{UP} \not\subseteq \mathcal{BPP}$ and one-way functions exist. It also implies a new result under a seemingly incomparable assumption that there exist a secure key agreement protocol in which it is easy to recognize valid transcripts.

---

*These results first appeared in the first author's undergraduate thesis [5] and an extended abstract will appear in *TCC 2010* [6].

†Department of Computer Science, Cornell University, Ithaca, NY 14853. `mailto:eleanor@cs.cornell.edu`.

‡School of Engineering and Applied Sciences and Center for Research on Computation and Society, Harvard University, 33 Oxford Street, Cambridge, MA 02138. `mailto:salil@seas.harvard.edu`. `http://people.seas.harvard.edu/~salil/`. Supported by NSF grant CNS-0831289.

1 Introduction

Composition has been one of the most active subjects of research on zero-knowledge proofs. The goal is to understand whether the zero-knowledge property is preserved when a zero-knowledge proof is repeated many times. The answers vary depending on the variant of zero knowledge in consideration and the form of composition (e.g. sequential, parallel, or concurrent). The study of composition was first aimed at reducing the soundness error of basic constructions of zero-knowledge proofs (via sequential or parallel composition), but was later also motivated by considering networked environments in which an adversary might be able to open several instances of a protocol (even concurrently).

Soon after Goldwasser, Micali, and Rackoff introduced the concept of zero-knowledge proofs [20], it was realized that composability is a subtle issue. In particular, this motivated a strengthening of the GMR definition, known as *auxiliary-input zero knowledge* [21, 19, 9], which was shown to be closed under sequential composition [19]. The need for this stronger definition was subsequently justified by a result of Goldreich and Krawczyk [16], who showed that the original GMR definition is not closed under sequential composition. Specifically, they exhibited a protocol that is *plain zero knowledge* when executed once, but fails to be zero knowledge when executed twice sequentially.

The starting point for our work is the realization that the Goldreich–Krawczyk protocol is not an entirely satisfactory counterexample, because the prover strategy is inefficient (i.e. superpolynomial time). Most cryptographic applications of zero-knowledge proofs require a prover strategy that can be implemented efficiently given an appropriate auxiliary input (e.g. NP witness). Prover efficiency can intuitively have an impact on the composability of zero-knowledge proofs, because an adversarial verifier may be able to use the extra computational power of one prover copy to "break" the zero-knowledge property of another copy. Indeed, known positive results on the parallel and concurrent composability of witness-indistinguishable proofs (a weaker variant of zero-knowledge proofs) rely on prover efficiency [9] .

Thus, we revisit the sequential composability of plain zero knowledge, but restricted to efficient provers. Our first result is positive, and shows that such proofs *are* closed under any constant number of sequential compositions (in contrast to the Goldreich–Krawczyk result with unbounded provers). The case of a superconstant or polynomial number of compositions remains an interesting open question. This positive result refers to the standard formulation of plain zero knowledge, where the simulation and the verifier's view are required to be indistinguishable by nonuniform polynomial-time distinguishers (or distinguishers that are given the prover's auxiliary input in addition to the statement being proven).

We then consider the case where the distinguishers are uniform probabilistic polynomial-time algorithms, whose only additional input is the statement being proven. In this case, we obtain a negative result analogous to the one of Goldreich and Krawczyk, showing that zero knowledge is not closed under sequential composition of even 2 copies (assuming that $\mathcal{NP} \not\subseteq \mathcal{BPP}$). Informally, these two results say that plain zero knowledge is closed under a constant number of sequential compositions if and only if the distinguishers are at least a powerful as the prover.

We also examine the *parallel* composability of *auxiliary-input* zero knowledge. Here, too, Goldreich and Krawczyk [16] gave a negative result that utilizes an inefficient prover. Feige and Shamir [9], however, gave a negative result with an efficient prover, under the assumption that the discrete logarithm is hard, or more generally under the assumptions that $\mathcal{UP} \not\subseteq \mathcal{BPP}$ and one-way functions exist. We are interested in whether the complexity assumption used by Feige and Shamir can be weakened. To this end, we provide a general negative result (assuming the existence of nontrivial languages in $\mathcal{NP}$ with uniquely determined, secret feature functions as introduced by Haitner, Rosen, and Shaltiel [22]) and observe that this implies both the previous work and a new construction under a seemingly incomparable assumption, namely that there exists a key agreement protocol (in which it is easy to recognize valid transcripts).

2 Definitions and Preliminaries

2.1 Interactive Proofs

Given two interactive Turing machines – a prover P and a verifier V – we consider two types of interactive protocols: proofs of language membership (interactive proofs) and proofs of knowledge. In each case, both parties receive a common input x, and P is trying to convince V that $x \in L$ for some language L. We will allow P to have an extra "auxiliary input" or "witness" y. We use the notation (P, V) to denote an interactive protocol and the notation $\langle P(x, y), V(x) \rangle$ to denote the verifier V's view of that protocol with inputs (x, y) and x respectively. The choices for y will be given by a relation of the following kind:

Definition 2.1 (Poly-balanced Relation)**.** A binary relation R is *poly-balanced* if there exists a polynomial p such that for all $(x, y) \in R$, $|y| \leq p(|x|)$. The language generated by such a relation is denoted $L_R = \{x : (x, y) \in R\}$.

Observe that we don't require R to be polynomial-time verifiable, so *every* language L is generated by such a relation, for example the relation $R = \{(x, y) : |y| = |x| \text{ and } x \in L\}$.

Definition 2.2 (Interactive Proof)**.** We say that an interactive protocol (P, V) is an *interactive proof* system for a language L if there exists a poly-balanced relation R such that $L = L_R$ and the following properties hold:

- (Verifier Efficiency): The verifier V runs in time at most $\text{poly}(|x|)$ on input x.
- (Completeness): If $(x, y) \in R$ then the verifier $V(x)$ accepts with probability 1 after interacting with the prover $P(x, y)$ on common input x and prover auxiliary input y.
- (Soundness): There exists a function $s(n) \leq 1 - 1/\text{poly}(n)$ (called the *soundness error*) for which it holds that for all $x \notin L$ and for all prover strategies P^*, the verifier $V(x)$ accepts with probability at most $s(|x|)$ after interacting with P^* on common input x and prover auxiliary input y.

Definition 2.3 (Proof of Knowledge)**.** Let R be a poly-balanced relation. Given an interactive protocol (P, V), we let $p(x, y, r)$ be the probability that V accepts on common input x when y is P's auxiliary input and r is the random input generated by P's random coin flips. Let $P_{x,y,r}$ be the function such that $P_{x,y,r}(\overline{m})$ is the message sent by P after receiving messages $\overline{m}$. An interactive protocol $(P(x, y), V(x))$ is an *interactive proof of knowledge* for the relation R if the following three properties hold:

- (Verifier Efficiency): The verifier V runs in time at most $\text{poly}(|x|)$ on input x.
- (Completeness): If $(x, y) \in R$, then V accepts after interacting with P on common input x.
- (Extraction): There exists a function $s(n) \leq 1 - 1/\text{poly}(n)$ (called the *soundness error*), a polynomial q, and a probabilistic oracle machine K such that for every $x, y, r \in \{0, 1\}^*$, K satisfies the following condition: if $p(x, y, r) > s(|x|)$ then on input x and with access to oracle $P_{x,y,r}$ machine K outputs w such that $(x, w) \in R$ within an expected number of steps bounded by $q(|x|)/(p(x, y, r) - s(|x|))$.

Observe that extraction implies soundness, so a proof of knowledge for R is also an interactive proof for L_R.

Although the above definitions require a polynomial-time verifier, neither places any restriction on the computational power of the prover P. In keeping with the standard model of "realistic" computation, we sometimes prefer to limit the computational resources of both parties to polynomial time. Specifically, we add the additional requirement that there exists a polynomial p such that the prover $P(x, y)$ runs in time $p(|x|, |y|)$ where x is the common input and y is the prover's auxiliary input. We refer to such protocols as *efficient* or *efficient-prover* proofs.

2.2 Zero Knowledge

In keeping with the literature, we define zero knowledge in terms of the indistinguishability of the output distributions.

Definition 2.4 (Uniform/Nonuniform Indistinguishability)**.** Two ensembles of probability distributions $\{\Pi_1(x)\}_{x\in S}$ and $\{\Pi_2(x)\}_{x\in S}$ are *uniformly (resp. nonuniformly) indistinguishable* if for every uniform (resp. nonuniform) probabilistic polynomial-time algorithm D, there exists a negligible function μ such that for every $x \in S$,

$$\left|\Pr[D(1^{|x|}, \Pi_1(x)) = 1] - \Pr[D(1^{|x|}, \Pi_2(x)) = 1]\right| \leq \mu(|x|),$$

where the probability is taken over the samples of $\Pi_1(x)$ and $\Pi_2(x)$ and the coin tosses of D.

Often, definitions of computational indistinguishability give the distinguisher the index x (not just its length). This makes no difference for nonuniform distinguishers – since they can have x hardwired in – but it does matter for uniform distinguishers. Indeed, we will see that zero-knowledge proofs demonstrate different properties under composition depending on how much information the distinguisher is given about the inputs.

Also, uniform indistinguishability is usually not defined with a universal quantifier over $x \in S$, but instead with respect to all polynomial-time samplable distributions on $x \in S$ (e.g. [2][12]). We use the above definition for simplicity, but our results also extend to the usual definition.

For the purposes of this paper, we consider two different definitions of zero knowledge. The first, which has primarily been of interest for historical reasons, is the one originally introduced by Goldwasser, Micali, and Rackoff [20]:

Definition 2.5 (Plain Zero Knowledge)**.** An interactive proof system (P, V) for a language $L = L_R$ is *plain zero knowledge* (with respect to nonuniform distinguishers) if for all probabilistic polynomial-time machines V^*, there exists a probabilistic polynomial-time algorithm M_{V^*} that on input x produces an output probability distribution $\{M_{V^*}(x)\}$ such that $\{M_{V^*}(x)\}_{(x,y)\in R}$ and $\{\langle P(x, y), V^*(x)\rangle\}_{(x,y)\in R}$ are nonuniformly indistinguishable.

As is standard, the above definition refers to *nonuniform* distinguishers (which can have x, y and any additional information depending on x, y hardwired in as nonuniform advice). However, it is also natural to consider *uniform* distinguishers. In this setting, it is important to differentiate between the case where the distinguisher is only given the single verifier input x and the case where the distinguisher is given both x and the prover's auxiliary input y.

Definition 2.6. An interactive proof system (P, V) for a language $L = L_R$ is *plain zero knowledge with respect to V-uniform distinguishers* if for all probabilistic polynomial-time machines V^*, there exists a probabilistic polynomial-time algorithm M_{V^*} that on input x produces an output probability distribution $\{M_{V^*}(x)\}$ such that $\{(x, M_{V^*}(x))\}_{(x,y)\in R}$ and $\{(x, \langle P(x, y), V^*(x)\rangle)\}_{(x,y)\in R}$ are uniformly indistinguishable.

Definition 2.7. An interactive proof system (P, V) for a language $L = L_R$ is *plain zero knowledge with respect to P-uniform distinguishers* if for all probabilistic polynomial-time machines V^*, there exists a probabilistic polynomial-time algorithm M_{V^*} that on input x produces an output probability distribution $\{M_{V^*}(x)\}$ such that $\{(x, y, M_{V^*}(x))\}_{(x,y)\in R}$ and $\{(x, y, \langle P(x, y), V^*(x)\rangle)\}_{(x,y)\in R}$ are uniformly indistinguishable.

The next definition of zero knowledge that we will consider is the more standard definition which incorporates an auxiliary input for the verifier.

Definition 2.8 (Auxiliary-Input Zero Knowledge)**.** An interactive proof system (P, V) for a language L is *auxiliary-input zero knowledge* if for every probabilistic polynomial-time machine V^* and every polynomial p there exists a probabilistic polynomial-time machine M_{V^*} such that the probability ensembles $\{\langle P(x, y), V^*(x, z)\rangle\}_{(x,y)\in R, z\in\{0,1\}^{p(|x|)}}$ and $\{M_{V^*}(x, z)\}_{(x,y)\in R, z\in\{0,1\}^{p(|x|)}}$ are nonuniformly indistinguishable.

Observe that although this last definition is given only in terms of nonuniform indistinguishability, this is actually equivalent to requiring only uniform indistinguishability; any nonuniform advice used by the distinguisher can instead be incorporated into the verifier's auxiliary input z.

2.3 Composition

In this section, we explicitly state the definitions of sequential and parallel composition that will be used throughout this paper. These definitions can be applied to any of the definitions of zero knowledge given in the previous section.

Definition 2.9. Given an interactive proof system (P, V) and a polynomial $t(n)$, we consider the *$t(n)$-fold sequential composition* of this system to be the interactive system consisting of $t(n)$ copies of the proof executed in sequence. The i^{th} copy of the protocol is initialized after the $(i-1)^{th}$ copy has concluded. All copies of the protocol are initialized with the same inputs.

We can extend our notion of zero knowledge to this setting in the natural way.

Definition 2.10. An interactive proof (P, V) for the language L is *sequential zero knowledge* if for all polynomials $t(n)$, the $t(n)$-fold sequential composition of (P, V) is a zero knowledge proof for L.

Note that although the verifiers in the different proof copies may be distinct entities and may in fact be honest, this definition implicitly assumes the worst case in which a single adversary controls all verifier copies. That is, it considers a sequential adversary (verifier) to be an interactive Turing machine V^* that is allowed to interact with $t(n)$ independent copies of P (all on common input x) in sequence.

Our definition of parallel composition is analogous to the above definition:

Definition 2.11. Given an interactive proof system (P, V) and a polynomial $t(n)$, we consider the *$t(n)$-fold parallel composition* of this system to be the interactive system consisting of $t(n)$ copies of the proof executed in parallel. Each message in the i^{th} round of a copy of the protocol must be sent before any message from the $(i+1)^{th}$ round. All copies of the protocol are initialized with the same inputs.

We can again extend our notion of zero knowledge to this setting:

Definition 2.12. An interactive proof (P, V) for the language L is *parallel zero knowledge* if for all polynomials $t(n)$ the $t(n)$-fold parallel composition of (P, V) is a zero-knowledge proof for L.

Thus a parallel adversary (verifier) is an interactive Turing machine V^* that is allowed to interact with $t(n)$ independent copies of P (all on common input x) in parallel. That is the i^{th} message in each copy is sent before the $(i+1)^{th}$ message of any copy of the protocol.

3 Sequential Zero Knowledge

3.1 Previous Results

In the area of sequential zero knowledge, there are two major results. The first is a negative result concerning the composition of plain zero-knowledge proofs.

Theorem 3.1 (Goldreich and Krawczyk [16])**.** *There exists a plain zero-knowledge proof (with respect to nonuniform distinguishers) whose 2-fold sequential composition is not plain zero-knowledge.*

The second significant result to emerge from the area concerns the composition of auxiliary-input zero-knowledge proofs. In this case it is possible to show that the zero-knowledge property is retained under sequential composition.

Theorem 3.2 (Goldreich and Oren [19])**.** *If* (P, V) *is auxiliary-input zero knowledge, then* (P, V) *is auxiliary-input sequential zero knowledge.*

These two results provide a context for our new results on sequential composition.

3.2 New Results

While Theorem 3.1 demonstrates that the original definition of zero knowledge is not closed under sequential composition, it relies on the fact that the prover can be computationally unbounded. In this section, we address the question: what happens when you compose *efficient-prover* plain zero-knowledge proofs? We obtain two results that partially characterize this behavior.

First we show that the Goldreich and Krawczyk result (Theorem 3.1) cannot be extended to efficient-prover plain zero-knowledge proofs. Indeed, we show that such proofs are closed under a *constant* number of compositions.

Theorem 3.3. *If* (P, V) *is an efficient-prover plain zero-knowledge proof system with respect to nonuniform (resp.,* P*-uniform) distinguishers then for every* constant k*, the* k*-fold sequential composition of* (P, V) *is also plain zero knowledge w.r.t. nonuniform (resp.,* P*-uniform) distinguishers.*

We leave the case of a super-constant number of compositions as an intriguing open problem.

Next we consider the case of V-uniform distinguishers, and we show that such protocols are *not* closed under 2-fold sequential composition with efficient provers.

Theorem 3.4. *If* $\mathcal{NP} \not\subseteq \mathcal{BPP}$ *then there exists an efficient-prover plain zero-knowledge proof with respect to* V*-uniform distinguishers whose 2-fold composition is not plain zero knowledge with respect to* V*-uniform distinguishers.*

Informally, Theorems 3.3 and 3.4 say that plain zero knowledge is closed under a constant number of sequential compositions if and only if the distinguishers are at least as powerful as P.

3.2.1 Proof of Theorem 3.3.

We now prove that efficient-prover plain zero-knowledge is closed under $O(1)$-fold sequential composition.

Proof. Let (P_k, V_k) denote the sequential composition of k copies of (P, V). We prove by induction on k that (P_k, V_k) is plain zero knowledge with respect to nonuniform (resp., P-uniform) distinguishers.

(P_1, V_1) is zero knowledge by assumption.

Assume for induction that (P_{k-1}, V_{k-1}) is zero knowledge, and consider the interactive protocol (P_k, V_k). Let V_k^* be some sequential verifier strategy for interacting with P_k, and let V_{k-1}^* denote the sequential verifier that emulates V_k^*'s interactions with the first $k-1$ copies of the the proof system (P, V) and then halts. Since (P_{k-1}, V_{k-1}) is zero knowledge, there exists a simulator M_{k-1} that successfully simulates V_{k-1}^*.

Define H_k^* to be the "hybrid" verifier strategy (for interaction with P) that consists of running the simulator M_{k-1} to obtain a simulated view v of the first $k-1$ interactions, and then emulates V_k^* (starting from the simulated view v) in the kth interaction. Since (P, V) is plain zero knowledge, there exists a polynomial-time simulator M_k for this verifier strategy.

We now show that M_k is also a valid simulator for (P_k, V_k^*). Since by induction (P_{k-1}, V_{k-1}) is plain zero knowledge versus nonuniform (resp., P-uniform) distinguishers, the ensembles $\Pi_1(x, y) = (x, y, \langle P_{k-1}(x, y), V_{k-1}^*(x)\rangle)$ and $\Pi_2(x, y) = (x, y, M_{k-1}(x))$ are nonuniformly (resp., uniformly) indistinguishable when $(x, y) \in R$. Consider the function $f(x, y, v) = (x, y, v')$ that emulates V_k^* starting from view v in one more interaction with $P(y)$ to obtain view v'. Since f is polynomial-time computable, we have that $f(\Pi_1(x, y))$ and $f(\Pi_2(x, y))$ are also nonuniformly (resp., uniformly) indistinguishable. Observe that $f(\Pi_1(x, y)) =$
$(x, y, \langle P_k(x, y), V_k^*(x)\rangle)$ and $f(\Pi_2(x, y)) = (x, y, M_k(x))$ therefore M_k is a valid simulator for (P_k, V_k^*) and hence (P_k, V_k) is plain zero knowledge with respect to nonuniform (resp., P-uniform) distinguishers. □

In this proof, we implicitly rely on the fact that the number of copies k is a constant. It is possible that the running time of the simulation is $\Theta(n^{g(k)})$ for some growing function g, and hence super-polynomial for nonconstant k.

Note that this result doesn't conflict with either Theorem 3.1 (in which the prover was allowed to use exponential time and was therefore able to distinguish between a simulated interaction and a real interaction) or Theorem 3.4 (in which the prover is polynomial time but the distributions are only indistinguishable to a V-uniform distinguisher, so the prover was still able to distinguish between a simulated interaction and a real interaction). Instead, it demonstrates that when neither party has more computational resources than the distinguisher, it is possible to prove a sequential closure result for plain zero knowledge, albeit restricted to a constant number of compositions.

3.2.2 Proof of Theorem 3.4.

We now prove Theorem 3.4, showing that plain zero knowledge with respect to V-uniform distinguishers is *not* closed under sequential composition. Our proof of Theorem 3.4 is a variant of the Goldreich-Krawczyk [16] proof of Theorem 3.1, so we be begin by reviewing their construction.

Overview of the Goldreich-Krawczyk Construction [16]. In the proof of Theorem 3.1, the key to constructing a zero-knowledge protocol that breaks under sequential composition lies in taking advantage of the difference in computational power between the unbounded prover and the polynomial-time verifier. The proof requires the notion of an *evasive pseudorandom ensemble*. This is simply a collection of sets $S_i \subseteq \{0, 1\}^{p(i)}$ such that each set is pseudorandom and no polynomial-time algorithm can generate an element of S_i with non-negligible probability. The existence of such ensembles was proven by Goldreich and Krawczyk in [17]. Using this, Goldreich and Krawczyk [16] construct a protocol such that in the first sequential copy, the verifier learns some element $s \in S_{|x|}$. In the second iteration, the verifier uses this s (whose membership in $S_{|x|}$ can be confirmed by the prover) to extract information from P. A polynomial-time prover would be unable to generate or verify $s \in S_{|x|}$, therefore the result inherently relies on the super-polynomial time allotted to the prover.

Overview of our Construction. As in the Goldreich-Krawczyk construction, we take advantage of the difference in computational power between the two parties. However, since both are required to be polynomial-time machines, the only advantage that the prover has over the verifier is in the amount of nonuniform input each machine receives. The prover is allowed $\text{poly}(|x|)$ bits of auxiliary input y whereas the verifier receives only the $|x|$ bits from the common input x. In order to take advantage of this difference, we define *efficient bounded-nonuniform* evasive pseudorandom ensembles. Using the newly defined ensembles, we construct an analogous protocol; in the first iteration, the verifier learns some element of an efficient bounded-nonuniform evasive pseudorandom ensemble, and in the second it uses this information to extract otherwise unobtainable information from P.

Definition 3.5. Let q be a polynomial and let $S = \{S_1, S_2, \ldots\}$ be a sequence of (non-empty) sets such that each $S_n \subseteq \{0,1\}^n$. We say that S is a *efficient $q(n)$-nonuniform evasive pseudorandom ensemble* if the following three properties hold:

(1) For all probabilistic polynomial-time machines A with at most $q(n)$ bits of nonuniformity, S_n is indistinguishable from the uniform distribution on strings of length n. That is, there exists a negligible function ϵ such that for all sufficiently large n,

$$\left| \Pr_{x \in S_n}[A(x) = 1] - \Pr_{x \in U_n}[A(x) = 1] \right| \leq \epsilon(n).$$

(2) For all probabilistic polynomial-time machines B with at most $q(n)$ bits of nonuniformity, it is infeasible for B to generate any element of S_n except with negligible probability. That is, there exists a negligible function ϵ such that for all sufficiently large n,

$$\Pr_{r \in \{0,1\}^{q(n)}}[B(x, r) \in S_n] \leq \epsilon(n).$$

(3) There exists a polynomial $p(n)$ and a sequence of strings $\{\pi_n\}_{n \in \mathbb{N}}$ of length $|\pi_n| = p(n)$ such that:

   (a) There exists a probabilistic polynomial-time machine D such that for all $n \in \mathbb{N}$ and $x \in \{0,1\}^n$, $D(\pi_n, x) = 1$ if $x \in S_n$ and $D(\pi_n, x) = 0$ else.

   (b) There exists an expected probabilistic polynomial-time machine E such that for all n $E(\pi_n)$ is a uniformly random element of S_n.

That is there exist efficient algorithms with polynomial-length advice for checking membership in the ensemble and for choosing an element uniformly at random.

This definition is similar in spirit to the notion of an evasive pseudorandom ensemble used by Goldreich and Krawczyk in the proof of Theorem 3.1. However, we add the additional requirement that a polynomial-time machine with an appropriate advice string π_n can identify and generate elements of the ensemble. In order for this to be possible, we relax the pseudorandomness and evasiveness requirements to only hold with respect to distinguishers with bounded nonuniformity rather than with respect to nonuniform distinguishers.

The introduction of this definition begs the question of whether or not such ensembles exist. Fortunately it turns out that they do.

Theorem 3.6. *There exists an efficient $n/4$-nonuniform evasive pseudorandom ensemble.*

The proof of this theorem is in Appendix A. It shows that if we select a hash function $h_n : \{0,1\}^n \to \{0,1\}^{5n/16}$ from an appropriate pairwise independent family then with high probability $S_n = h_n^{-1}(0^{5n/16})$ is an $n/4$-nonuniform evasive pseudorandom set. The pseudorandomness and evasiveness conditions (items (1) and (2)) are obtained by using pairwise independence and taking a union bound over all algorithms with $n/4$ bits of nonuniformity. The efficiency condition (item (3)) is obtained by taking h_n to be from a standard family (e.g., $h_n(x) =$ the first $5n/16$ bits of $a \cdot x + b$) and taking π_n to be the descriptor of h_n (e.g., (a, b)).

We use this result to demonstrate that efficient-prover plain zero-knowledge proofs with respect to V-uniform distinguishers are not closed under sequential composition. The construction is analogous to the one by Goldreich and Krawczyk.

Proof. Let $S_1, S_2, \ldots$ be an efficient $n/4$-nonuniform evasive pseudorandom ensemble (the existence of which is guaranteed by Theorem 3.6) and let $\pi_1, \pi_2, \ldots$ be the sequence of polynomial-length strings that enable testing membership in and sampling random elements of the S_n's.

We now construct an interactive-proof protocol (P, V) for the trivial language $L = \{0,1\}^*$. First we define the relation R which will specify the possible auxiliary inputs for P, specifically $R = \{(x, (\pi_{4|x|}, w)) : |w| \le |x|\}$. Notice that $L_R = L$. The string w plays no role in the relation; we will use it as "secret" information that the verifier can learn from two sequential executions.

Let x be the common input for P and V, let $n = |x|$, and let (π_{4n}, w) be P's auxiliary input. The verifier V begins by sending to the prover a random string s of length $4n$. The prover P checks whether $s \in S_{4n}$ (the $(4n)^{th}$ set in the sequence). If this is the case (i.e., $s \in S_{4n}$) then P sends to V the secret value w. Otherwise, P sends to V a string s_0 randomly selected from S_{4n}. V then always accepts.

Step	$P(x, \pi_{4n} \circ w)$		$V(x)$
1		$\leftarrow$	$s \in_R \{0,1\}^{4n}$
2	if $s \in S_{4n}$: $c = w$		
	else $c \in_R S_{4n}$		
	c	$\rightarrow$	

Figure 1: A plain zero-knowledge proof with respect to V-uniform distinguishers.

Unlike the prover in the Goldreich-Krawczyk protocol, this prover runs in polynomial time given P's witness (π_{4n}, w). The prover need only check if an element is in S_{4n} and produce a uniformly random element of S_{4n}; the existence of efficient algorithms for both is guaranteed by Property (3) of the definition of an efficient $n/4$-nonuniform evasive pseudorandom ensemble.

On one hand, the protocol is zero knowledge (when executed once). To show this, we present for any verifier V^*, a polynomial-time simulator M_{V*} that can simulate the conversations between V^* and the prover P. There is only one prover message that needs to be simulated, namely Step 2. P sends the value of w in case that the string s sent by the verifier in Step 1 belongs to the set S_{4n}, and a randomly selected element of S_{4n} otherwise. By Property (2) of Definition 3.5, there is only a negligible probability that the first case holds. Indeed, no probabilistic polynomial-time machine (in our case, the verifier V) with n bits of nonuniformity (namely the input x) can find a string $s \in S_{4n}$, except with negligible probability. Therefore, the simulator can succeed by always simulating the second possibility, i.e. sending a random element c from S_{4n}. This step is simulated by randomly choosing c from $\{0,1\}^{4n}$ rather than from S_{4n}. By Property (1) of Definition 3.5, a machine with n bits of nonuniform input (namely the V-uniform distinguisher with input x) cannot distinguish such a string from one chosen from S_{4n}.

On the other hand, this protocol fails to remain zero knowledge when composed with itself twice in sequence. We stress that the same efficient $n/4$-nonuniform evasive pseudorandom ensemble is used in all the executions of the protocol. Therefore, the string c, obtained by a verifier in the first execution of the protocol, enables him to deviate from the protocol during a second execution in order to obtain the value of w. Specifically, consider the verifier strategy V^* that behaves correctly in the first iteration of the protocol, but in the first step of the second iteration sends the string c obtained from the previous iteration instead of a random element of $\{0,1\}^{4n}$. Since $c \in S_{4n}$, V^* now learns w which, by assumption, he could not calculate (or simulate) on its own.

We now use the fact that $\mathcal{NP} \nsubseteq \mathcal{BPP}$ to show that any efficient simulator M_{V^*} for this strategy V^* will produce an output that is distinguishable from the verifier's view by V-uniform distinguishers. Define $D(x,t)$ to output 1 if the transcript contains a message that is a satisfying assignment to x (when interpreted as a circuit). Thus for every satisfiable circuit x and satisfying assignment w $\Pr[D(x, \langle P(\pi_{4n}, w), V^* \rangle) = 1] = 1$. Hence if M_{V^*} were a good simulator with respect to V-uniform distinguishers, then $\Pr[D(x, M_{V^*}(x)) = 1] \geq \frac{1}{2}$, i.e. M_{V^*} finds a satisfying assignment to x with probability at least 1/2. This contradicts the assumption that $\mathcal{NP} \nsubseteq \mathcal{BPP}$, therefore there is no simulator for V^*, and hence the 2-fold sequential composition of (P, V) is not zero knowledge with respect to V-uniform distinguishers. □

4 Parallel Zero Knowledge

4.1 Previous Results

There are two classic results that provide context for our new result concerning the parallel composition of efficient-prover zero-knowledge proof systems. In both cases, the result applies to auxiliary-input (as well as plain) zero knowledge, and both results are negative.

The first result establishes the existence of non-parallelizable zero-knowledge proofs independent of any complexity assumptions.

Theorem 4.1 (Goldreich and Krawczyk [16]). *There exists an auxiliary-input zero knowledge proof whose 2-fold parallel composition is not auxiliary-input zero knowledge (or even plain zero knowledge with respect to nonuniform distinguishers).*

While this result demonstrates that zero knowledge is not closed under parallel composition in general, the proof (like that of Theorem 3.1) inherently relies on the unbounded computational power of the provers. Without the additional computational resources necessary to generate a string and test membership in an evasive pseudorandom ensemble, the prover would be unable to execute the defined protocol.

The second such result constructs an *efficient-prover* non-parallelizable zero-knowledge proof based on a zero-knowledge proof of knowledge of the discrete-logarithm relation.

Theorem 4.2 (Feige and Shamir [9]). *If the discrete logarithm assumption holds then there exists an efficient-prover auxiliary-input zero-knowledge proof whose 2-fold parallel composition is not auxiliary-input zero knowledge (or even plain zero knowledge with respect to V-uniform distinguishers).*

This proof relies on the very specific assumption that the discrete logarithm problem is intractable. However as Feige and Shamir observed [9], the only properties of this problem which are actually necessary are the fact that discrete logarithms are unique and that they have a zero-knowledge proof of knowledge. It is therefore natural to consider generalizing the result to proofs of language membership for any language $L \in \mathcal{NP}$ with exactly one witness for each element $x \in L$. The class of such languages is known as $\mathcal{UP}$. Moreover, if one-way functions exist, then every problem in $\mathcal{NP}$ (and hence in $\mathcal{UP}$) has a zero-knowledge proof of knowledge [18]. Thus:

Theorem 4.3 (Feige and Shamir [9])**.** *If $\mathcal{UP} \nsubseteq \mathcal{BPP}$ and one-way functions exist then there exists an efficient-prover auxiliary-input zero-knowledge proof whose 2-fold parallel composition is not auxiliary-input zero knowledge (or even plain zero knowledge with respect to V-uniform distinguishers).*

4.2 New Results

In this work, we demonstrate the existence of *efficient-prover* non-parallelizable zero-knowledge proofs under more general complexity assumptions. Specifically, we recall the notion of a feature function, first introduced by Haitner, Rosen, and Shaltiel [22], and show that such proofs can be constructed for any language with a uniquely determined secret feature function. We then show that this general result implies both the previous work by Feige and Shamir [9] and a new, incomparable construction based on key agreement protocols.

Definition 4.4 (Feature Function)**.** Let $L \in \mathcal{NP}$ and let $m(n)$ denote the length of witnesses $w \in R_L(x)$ for $x \in L \cap \{0,1\}^n$. Let $\ell(n)$ be an integer function. A *feature function* $g : \{0,1\}^n \times \{0,1\}^m(n) \to \{0,1\}^{\ell(n)}$ is a polynomial-time computable function. We say that g is *uniquely determined* if for all $x \in L$ and for all $w_1, w_2 \in R_L(x)$, $g(x, w_1) = g(x, w_2)$. We say that such a function is *secret* if for every sufficiently large n and every nonuniform polynomial-time adversary A there exists some negligible function ϵ such that,

$$|\Pr_{(x,w)\in R_L}[A(x,w) = g(x,w)] - 2^{\ell(|x|)}| \leq \epsilon(|x|).$$

Using this language, our main result on non-parallelizable zero knowledge proofs follows:

Theorem 4.5. *If there exist languages $L \in \mathcal{NP} \setminus \mathcal{BPP}$ with uniquely determined, secret feature functions and one-way functions exist, then there exists an efficient-prover auxiliary-input zero-knowledge proof whose 2-fold parallel composition is not auxiliary-input zero knowledge (or even plain zero knowledge with respect to V-uniform distinguishers).*

Proof. Let $L \in \mathcal{NP} \setminus \mathcal{BPP}$, and let $R_L = \{(x, w) : w \text{ is a witness for } x \in L\}$ be an $\mathcal{NP}$-relation with a uniquely determined feature function. Since $L \in \mathcal{NP}$, there exists an efficient-prover zero-knowledge proof of knowledge (ZKPOK) of a witness w with error $s(n) \leq 2^{-m}$ where m is again the maximum length of a witness w[18]. If necessary, the required error can be achieved by sequential composition of any initial ZKPOK.

We can use this proof as a subprotocol for constructing the following interactive proof for the language L. V begins by sending the message $c = 0$ to P. If $c = 0$, then P uses the ZKPOK to demonstrate that he knows a witness $w \in R_L(x)$. If $c \neq 0$, V demonstrates knowledge of $w' \in R_L(x)$ using the same ZKPOK. If the proof is successful and the transcript is valid (which can be checked by P by our assumption of verifiable transcripts), then P shows in zero knowledge that he too knows a witness $w \in R_L(x)$ and then sends the value of the feature function $g(x, w)$ to V.

The protocol is summarized below.

Efficient-Prover Interactive Proof: The fact that this protocol is an interactive proof follows directly from the fact that the subprotocol is (by assumption) a proof of knowledge. Completeness

Step	$P(x,w)$		$V(x,y)$
1		$\leftarrow$	$c=0$
2	if $c=0$: ZKPOK of $w \in R_L(x)$	$\rightarrow$	
		$\leftarrow$	if $c \neq 0$: ZKPOK of $w' \in R_L(x)$
3	if $c \neq 0$: ZKPOK of $w \in R_L(x)$	$\rightarrow$	
4	if $c \neq 0$, V's ZKPOK is successful: $g(x,w)$	$\rightarrow$	

Figure 2: A efficient-prover non-parallelizable zero-knowledge proof for L.

and soundness follow from completeness and extraction properties of the ZKPOK that P conducts in Step 2 or Step 3 respectively. Prover and verifier efficiency likewise follow from the respective properties of the ZKPOK subprotocol.

Zero Knowledge: Given any verifier strategy V^* we can construct a simulator M_{V^*}. M_{V^*} begins by randomly choosing and fixing the coin tosses of the verifier V^*, and then runs the verifier V^* in order to obtain its first message c. If $c = 0$, M_{V^*} then emulates the simulator for the ZKPOK to simulate Step 2. It then does nothing for Step 3. If $c \neq 0$, then M_{V^*} simulates the ZKPOK in Step 2 by following the correct "verifier" protocol and running V^* in order to simulate the "prover" half of the protocol. M_{V^*} then simulates Step 3 using the simulator for the subprotocol. The expected time of all of these steps is polynomial; this follows directly from the running time of the simulators provided by the various subprotocols.

Finally, the simulator proceeds to Step 4. If $c = 0$ then there is no message sent in Step 4. If $c \neq 0$ and the ZKPOK in Step 2 was unsuccessful, then there is again no message sent in Step 4. If $c \neq 0$ and the proof in Step 2 was successful, then M_{V^*} runs the following two extraction techniques in parallel, halting when one succeeds: First, it attempts to extract some $w' \in R_L(x)$ by employing the extractor K using V^*'s strategy from Step 2 as an "oracle." Second it attempts to learn some witness $w' \in R_L(x)$ by trying each of the 2^m possible witnesses in sequence. If M_{V^*} has successfully found a witness, it evaluates $g(x, w')$ and uses this to simulate Step 4.

The indistinguishability and expected polynomial running time of the simulation follow from those of the ZKPOK simulator, except for the simulation of Step 4 in the case $c \neq 0$. To analyze this, let p be the probability that V^* succeeds in the ZKPOK in Step 2. If $p > 2 \cdot 2^{-m}$, then there exists such an extractor K that extracts a witness (j, r_j) in expected time $q(|x|)/(p - s(|x|)$. Since this occurs with probability p, the expected time for this case is bounded by $(p \cdot q(|x|))/(p - s(|x|)) \leq (p \cdot q(|x|))/(p - 2^{-m}) \leq (p \cdot q(|x|))/(p/2) \leq 2q(|x|) = \text{poly}(|x|)$. If $p \leq 2 \cdot 2^m$ then the brute force technique will find a witness in expected time $p \cdot 2^m \leq 2 = \text{poly}(|x|)$. Checking t's validity takes polynomial time by assumption, and determining k takes time $\Theta(|x|)$, therefore the entire simulation runs in expected polynomial time.

The indistinguishability of the final step of this simulation relies on the fact that the feature function g is uniquely determined, therefore $g(x, w) = g(x, w')$, so the simulation is polynomially indistinguishable from V^*'s view of the interactive protocol.

Parallel Execution: Consider now two executions, $(\widetilde{P}_1, \widetilde{V})$ and $(\widetilde{P}_2, \widetilde{V})$ in parallel. A cheating verifier V^* can always extract some witness $w \in \{(1, r_1), (2, r_2)\}$ from $\widetilde{P}_1$ and $\widetilde{P}_2$ using the following strategy: in Step 1, V^* sends $c = 0$ to $\widetilde{P}_1$ and $c = 1$ to $\widetilde{P}_2$. Now V^* has to execute the protocol (P, V) twice: once as a verifier talking to the prover $\widetilde{P}_1$, and once as a prover talking to the verifier $\widetilde{P}_2$. This he does by serving as an intermediary between $\widetilde{P}_1$ and $\widetilde{P}_2$, sending $\widetilde{P}_1$'s messages to $\widetilde{P}_2$, and $\widetilde{P}_2$'s messages to $\widetilde{P}_1$. Now $\widetilde{P}_2$ willfully sends $g(x, w)$ to $\widetilde{V}$ (which, by the secrecy property of the feature function, $\widetilde{V}$ is incapable of computing on his own). $\square$

This proof generalizes the previous work towards demonstrating the existence of non-parallelizable efficient-prover zero-knowledge proofs.

Corollary 4.6 (Feige and Shamir [9])**.** *If $\mathcal{UP} \nsubseteq \mathcal{BPP}$ and one-way functions exist then there exists an efficient-prover auxiliary-input zero-knowledge proof whose 2-fold parallel composition is not auxiliary-input zero knowledge (or even plain zero knowledge with respect to V-uniform distinguishers).*

Proof. Let $L \in UP \subsetneq BPP$, and let $g(x, w) = w$. Since for all $x \in L$ there exists exactly one witness $w \in R_L(x)$, g is uniquely determined. Furthermore it is immediately clear that the secrecy condition holds, therefore the result follows from Theorem 4.5. □

Morevoer, our main parallel result is a non-trivial generalization of that previous work, that is it implies the existence of such non-parallelizable protocols under an incomparable assumption: the existence of verifiable key agreement protocols. Following the standard notion of key agreement, we introduce the following definition.

Definition 4.7. A *key agreement protocol* is an efficient protocol between two parties P_1, P_2 with the following four properties:

- Input: Both parties have common input 1^ℓ which is a security parameter written in unary.
- Output: The outputs of both parties are k-bit strings (for some $k = \text{poly}(\ell)$).
- Correctness: The parties have the same output with probability 1 (when they follow the protocol). This common output is called the *key.*
- Secrecy: No probabilistic polynomial time Turing machine E given 1^ℓ and the transcript of the protocol (messages between P_1, P_2) can distinguish with non-negligible advantage the key from a uniformly distributed k-bit string. That is, $\{(1^\ell, \text{transcript}(P_1, P_2), \text{output}(P_1, P_2))\}_{1^\ell : \ell \in \mathbb{N}}$ is nonuniformly indistinguishable from $\{(1^\ell, \text{transcript}(P_1, P_2), U_k)\}_{1^\ell : \ell \in \mathbb{N}}$.

For technical reasons, we impose an additional technical condition.

Definition 4.8. Let (P_1, P_2) be a key agreement protocol. We say that a pair $(i, r) \in \{1, 2\} \times \{0, 1\}^*$ is *consistent* with a transcript t of messages if the messages from P_i in t are what P_i would have sent had its coin tosses been r and had it received the prior messages specified by t. We say that t is *valid* if there exist r_1, r_2 such that t is consistent with both $(1, r_1)$ and $(2, r_2)$; that is, t occurs with nonzero probability when the honest parties P_1 and P_2 interact. We say that (P_1, P_2) has *verifiable transcripts* if there is a polynomial-time algorithm that can decide whether a transcript t is valid given t and any pair (i, r) consistent with t.

We note that many existing key agreement protocols have verifiable transcripts, including the Diffie-Hellman key exchange and the protocols constructed from any public-key encryption scheme with verifiable public keys. The existence of secure key agreement protocols with verifiable transcripts seems incomparable to the assumption that $\mathcal{UP} \nsubseteq \mathcal{BPP}$ which was used in Theorem 4.3.

Corollary 4.9. *If key agreement protocols with verifiable transcripts exist then there exists an efficient-prover auxiliary-input zero-knowledge proof whose 2-fold parallel composition is not zero knowledge.*

Proof. If such protocols exit, then one-way functions exist. It therefore suffices to demonstrate a non-trivial $\mathcal{NP}$ language with a uniquely determined, secret feature function. Define a language

$$L = \{t : t \text{ is a valid key-agreement transcript and } \exists(i, r_i) \text{ consistent with } t\}.$$

We claim that the function g that maps $(t, (i, r_i)) \in R_L$ to the key k generated by that transcript is a uniquely determined, secret feature function. g can be efficiently computed by using the coin tosses r_i and the messages of theothe player as recorded in the transcript t to follow the protocol and generate a key k. Since t must be valid, such a key k is unique (the feature function is uniquely determined) and the secrecy property follows from the secrecy property of any valid transcript for the key agreement protocol. Therefore the result follows from Theorem 4.5. □

5 Conclusions and Open Problems

We view our results as pointing out the significance of prover efficiency, as well as the power of the distinguishers, in the composability of zero-knowledge proofs. Indeed, we have shown that with prover efficiency, the original GMR definition enjoys a greater level of composability than without. Nevertheless, the now-standard notion of auxiliary input zero knowledge still seems to be the appropriate one for most purposes. In particular, we still do not know whether plain zero knowledge is closed under a super-constant number of compositions. We also have not considered the case that different statements are being proven in each of the copies, much less (sequential) composition with arbitrary protocols. For these, it seems likely that auxiliary input zero knowledge, or something similar, is necessary.

One way in which our negative result on sequential composition (of plain zero knowledge with respect to V-uniform distinguishers, Theorem 3.4) can be improved is to provide an example where the prover's auxiliary inputs are defined by a relation that can be decided in polynomial time (in contrast to our construction, where the prover's auxiliary input contains the advice string π_{4n}, which may be hard to recognize).

For the parallel composition of auxiliary-input zero knowledge with efficient provers, it remains open to determine whether a negative result can be proven under a more general assumption such as the existence of one-way functions. Our method provide an intermediate generalization that replaces the assumption $\mathcal{UP} \not\subseteq \mathcal{BPP}$ with the assumption that there is a a problem in $\mathcal{NP}$ for which the witnesses have a "uniquely determined feature" [22] that is hard to compute. Our construction complements that of Haitner, Rosen, and Shaltiel [22] — they consider the parallel repetition of natural zero-knowledge proofs (such as 3-Coloring [18] or Hamiltonicity [7]), and argue that "certain black-box techniques" cannot prove that a feature $g(x)$ will remain hard to compute by the verifier (on average). In contrast, we consider the parallel repetition of a contrived zero-knowledge proof and show that a cheating verifier can always learn a certain hard-to-compute feature $g(x)$.

Acknowledgments

We thank the TCC 2010 reviewers for helpful comments.

References

[1] B. Barak. How to go beyond the Black-Box Simulation Barrier. In *42nd IEEE Symposium on Foundations of Computer Science*, pages 106-115, 2001.

[2] B. Barak, Y. Lindell, S. Vadhan. Lower Bounds for Non-Black-Box Zero Knowledge. *Proc. of the 44th IEEE Symposium on the Foundation of Computer Science*, pages 384393, 2003.

[3] M. Bellare and O. Goldreich. On defining proofs of knowledge. In *Advances in Crpytology – CRYPTO '92*, volume 740 of *Lecture Notes in Computer Science*, pages 390-420. Springer-Verlag, 1993, 16-20 August 1992.

[4] M. Ben-Or, O. Goldreich, S. Goldwasser, J. Hastad, J. Kilian, S. Micali, and P. Rogaway. Everything provable is provable in zero-knowledge. In Proceedings of Advances in Cryptology Crypto88. Lecture Notes in Computer Science, Vol. 403. Springer-Verlag, New York, 1990, pp. 37-56.

[5] E. Birrell. Composition of Zero-Knowledge Proofs. Undergraduate Thesis. Harvard University, 2009.

[6] E. Birrell and S. Vadhan. Composition of Zero-Knowledge Proofs with Efficient Provers. To appear in *Proceedings of the 7th IACR Theory of Cryptography Conference (TCC '10).* Springer-Verlag Lecture Notes in Computer Science. 9–11 February, 2010.

[7] M. Blum. How to prove a theorem so no one else can claim it. In *Proceedings of the International Congress of Mathematicians*, pages 1444-1451, 1987.

[8] W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Trans. on Info. Theory*, IT-22 Nov. 1976, pages 644-654.

[9] U. Feige and A. Shamir. Witness Indistinguishability and Witness Hiding Protocols. In *22nd ACM Symposium on the Theory of Computing*, pages 416-426, 1990.

[10] U. Feige and A. Shamir. Zero-Knowledge Proofs of Knowledge in Two Rounds. In *Crypto '89*, Springer-Verlag LNCS Vol. 435, pages 526-544, 1990.

[11] O. Goldreich. *Foundations of Cryptography - Basic Tools.* Cambridge University Press, 2001.

[12] O. Goldreich. A Uniform Complexity Treatment of Encryption and Zero Knowledge. *Journal of Cryptology*, Vol. 6, No. 1, pages 21-53, 1993.

[13] O. Goldreich. Zero-Knowledge twenty years after its invention. Cryptology ePrint Archive, Report 2002/186, 2002. http://eprint.iacr.org/

[14] O. Goldreich, S. Goldwasser, and S. Micali. How to Construct Random Functions. Journal of the Association for Computing Machinery, Vol. 33, No. 4, pages 792-807, 1986.

[15] O. Goldreich and A. Kahan. How to Construct Constant-Round Zero-Knowledge Proof Systems for NP. *Journal of Cryptology.* Vol. 9, No. 2, pages 167-189, 1996

[16] O. Goldreich and H. Krawczyk. On the Composition of Zero-Knowledge Proof Systems. *SIAM Journal on Computing*, Vol. 25, No. 1, February 1996, pages 169-192. Preliminary version in *ICALP '90.*

[17] O. Goldreich and H. Krawczyk. Sparse Pseudorandom Distributions. *Random Structures & Algorithms*, Vol. 3, No. 2, pages 163-174, 1992.

[18] O. Goldreich, S. Micali, and A. Wigderson. Proofs that Yield Nothing but their Validity or All Languages in NP have Zero-Knowledge Proof Systems. *Journal of the ACM*, Vol. 38, No. 1, pages 691-729, 1991.

[19] O. Goldreich and Y. Oren. Definitions and Properties of Zero-Knowledge Proof Systems. *Journal of Cryptology,* Vol. 7. No. 1, pages 1-32, 1994.

[20] S. Goldwasser, S. Micali, and C. Rackoff. Knowledge Complexity of Interactive Proofs. *Proc. 17th STOC*, pages 291-304. 1985.

[21] S. Goldwasser, S. Micali, and C. Rackoff. The Knowledge Complexity of Interactive Proof Systems. *SIAM Journal on Computing*, Vol. 18, pages 186-208, 1989.

[22] I. Haitner, A. Rosen, and R. Shaltiel. On the (Im)possibility of Arthur-Merlin Witness Hiding Protocols. In *Theory of Cryptography Conference*, 2009.

[23] S. Vadhan. *Pseudorandomness.* To appear in *Foundations and Trends in Theoretical Computer Science*, 2010.

A Construction of Pseudorandom Evasive Ensembles

Theorem A.1. *There exists an efficient $n/4$-nonuniform evasive pseudorandom ensemble.*

Definition A.2 (Pairwise Independent Hash Functions)**.** A family (i.e. multiset) of functions $\mathcal{H} = \{h : [N] \to [M]\}$ is *pairwise independent* if for all $x_1 \neq x_2 \in [N]$, when $h \in \mathcal{H}$ is a function chosen uniformly at random from $\mathcal{H}$, the random variables $h(x_1), h(x_2)$ are independent and uniformly distributed in $[M]$.

A standard construction of such families is the following:

Example A.3. Let $n, m \in \mathbb{N}$ such that $n > m$. Consider the hash family

$$\mathcal{H}_{n,m} = \{h_{a,b}|_m : \{0,1\}^n \to \{0,1\}^m\}_{a,b}$$

where $h_{a,b}(x)|_m$ is defined to be the first m bits of $a \cdot x + b$ where the arithmetic is done in the field of 2^n elements. Then $\mathcal{H}_{n,m}$ is a pairwise independent hash function.

Using such a family of hash functions it is possible to construct an efficient bounded-nonuniform evasive pseudorandom ensemble. The proof uses the following two standard inequalities.

Lemma A.4 (Pairwise Independent Tail Inequality)**.** *Let $X_1 \dots X_k$ be pairwise independent random variables taking values in the interval $[0,1]$, let $X = (\sum_i X_i)/k$, and $\mu = E[X]$. Then*

$$\Pr[|X - \mu| \geq \epsilon] \leq \frac{\mu}{k\epsilon^2}$$

Lemma A.5 (Markov Inequality)**.** *Let X be any random variable taking only non-negative values, and let $\epsilon > 0$. Then*

$$\Pr[X \geq \epsilon] \leq \frac{E[X]}{\epsilon}$$

The proof of Theorem A.1 proceeds as follows. We begin by fixing a collection of pairwise independent hash families $\mathcal{H}_{n.m}$, where $m = 5n/16$ for each $n \in \mathbb{N}$. We then establish three lemma, each corresponding to one of the three properties required by efficient bounded-nonuniform evasive pseudorandom ensembles. In Lemma A.6 we show that for sufficiently large n, a randomly chosen function $h_n \in \mathcal{H}_{n,m}$ has the property that with probability at least 3/4, no polynomial-time machine with $k \leq n/4$ bits of nonuniformity can non-negligibly distinguish between a random element of $S_n = \{x : h_n(x) = 0^m\}$ and the uniform distribution on $\{0,1\}^n$. In Lemma A.7 we prove a similar result for the second property. Namely we show that for sufficiently large n and for a randomly chosen function $h_n \in \mathcal{H}_{n,m}$ with probability at least 3/4 no polynomial machine B can generate an element of $S_n = \{x : h_n(x) = 0^m\}$ with non-negligible probability.

Taken together, these two lemmas demonstrate that for sufficiently large n, there must exist some $h_n \in \mathcal{H}_{n,m}$ such that the first two properties hold for the corresponding set S_n. Lemma A.8 then establishes that the third property also holds: a polynomial machine with *poly*(n) bits of nonuniformity can both recognize and generate elements of S_n. This intuition is formalized in the proof of Theorem A.1.

We now proceed to establish each of these three lemmas.

Lemma A.6. *For all n, consider $\mathcal{H}_{n,m}$ to be the pairwise independent hash family described above where $m = 5n/16$. Choose an element $h_n \in \mathcal{H}_{n,m}$ uniformly at random and define $S_n = \{x \in \{0,1\}^n : h_n(x) = 0^m\}$. For sufficiently large n, with probability at least 3/4, S_n exhibits the property that for all probabilistic polynomial-time machines A with $k \leq n/4$ bits of nonuniformity, S_n is indistinguishable from the uniform distribution on strings of length n. That is, there exists a negligible function ϵ such that for all sufficiently large n,*

$$\left| \Pr_{x \in S_n}[A(x) = 1] - \Pr_{x \in U_n}[A(x) = 1] \right| \leq \epsilon(n).$$

Proof. Fix a probabilistic polynomial-time machine $A : \{0,1\}^n \to \{0,1\}$ with $k \leq n/4$ bits of nonuniformity. Let $Id(\phi)$ denote the indicator function for the predicate ϕ, that is the function with value 1 when ϕ is true and value 0 else. The probability that A outputs 1 on input x (chosen uniformly from S_n) and random coin tosses r is given by:

$$\begin{aligned}
\Pr_{x \in S_n, r}[A(x) = 1] &= \frac{\sum_{x \in S_n} \Pr_r[A(x) = 1]}{|S_n|} \\
&= \frac{\sum_{x \in \{0,1\}^n} \mathrm{Id}(x \in S_n) \cdot \Pr_r[A(x) = 1]}{\sum_{x \in \{0,1\}^n} \mathrm{Id}(h_n(x) = 0^m)} \\
&= \frac{\sum_{x \in \{0,1\}^n} \mathrm{Id}(h(x) = 0^m) \cdot \Pr_r[A(x) = 1]}{\sum_{x \in \{0,1\}^n} \mathrm{Id}(h_n(x) = 0^m)} \\
&= \frac{X_1}{X_2}
\end{aligned}$$

where $X_1 = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \mathrm{Id}(h(x) = 0^m) \cdot \Pr_r[A(x) = 1]$ and $X_2 = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \mathrm{Id}(h_n(x) = 0^m)$.

Observe that each X_i is the average of 2^n pairwise-independent random variables (over the choice of h_n), so with high probability its value is close to its expectation, where the expectations are given by:

$$\begin{aligned}
\mu(X_1) &= \frac{1}{2^m} \Pr_{x \in U_n, r}[A(x) = 1] \\
\mu(X_2) &= \frac{1}{2^m}
\end{aligned}$$

More specifically, by Lemma A.4, with probability at least $1 - 1/2^n\epsilon^2$,

$$|X_i - \mu(X_i)| < \epsilon$$

.

We can combine these two inequalities to acquire upper and lower bounds for the probability that A outputs 1. Namely with probability at least $(1 - 1/2^n\epsilon^2)^2$

$$\frac{\mu(X_1) - \epsilon}{\mu(X_2) + \epsilon} \leq \frac{X_1}{X_2} = \Pr_{x \in S_n}[A(x) = 1] \leq \frac{\mu(X_1) + \epsilon}{\mu(X_2) - \epsilon}$$

By applying a Taylor series expansion, we observe that if $|\epsilon| < 1$, then we can bound values of this form using the following inequalities:

$$\frac{\mu(X_1) - \epsilon}{\mu(X_2) + \epsilon} \geq \frac{\mu(X_1) - \epsilon}{\mu(X_2)} - \frac{2(\mu(X_1) - \epsilon)}{\mu(X_2)^2} \cdot \epsilon \geq \frac{\mu(X_1)}{\mu(X_2)} - \frac{2(\mu(X_1) + \epsilon) + \mu(X_2)}{\mu(X_2)^2} \cdot \epsilon$$
$$\frac{\mu(X_1) + \epsilon}{\mu(X_2) - \epsilon} \leq \frac{\mu(X_1) + \epsilon}{\mu(X_2)} + \frac{2(\mu(X_1) + \epsilon)}{\mu(X_2)^2} \cdot \epsilon \leq \frac{\mu(X_1)}{\mu(X_2)} + \frac{2(\mu(X_1) + \epsilon) + \mu(X_2)}{\mu(X_2)^2} \cdot \epsilon$$

Therefore if $0 < \epsilon < 1$,then with probability at least $1 - 2/2^n\epsilon^2$

$$\left| \Pr_{x \in S_n}[A(x) = 1] - \frac{\mu(X_1)}{\mu(X_2)} \right| \leq \frac{2(\mu(X_1) + \epsilon) + \mu(X_2)}{\mu(X_2)^2} \cdot \epsilon$$

Observe that the probability that A outputs 1 on input x chosen uniformly from $\{0,1\}^n$ is simply $\mu(X_1)/\mu(X_2)$, therefore with probability at least $1 - 2/2^n\epsilon^2$

$$\left| \Pr_{x \in S_n}[A(x) = 1] - \Pr_{x \in \{0,1\}^n}[A(x) = 1] \right| \leq \frac{2(\mu(X_1) + \epsilon) + \mu(X_2)}{\mu(X_2)^2} \cdot \epsilon \leq 3\epsilon 2^m = 2^{-\Omega(n)}$$

We observe that the number of functions A that can be expressed by a Turing machine of description length $k_1 \leq \log(n/4)$ with nonuniform advice of length $k_2 \leq n/4$ is at most $2^{\log(n/4)+n/4} = (n/4)2^{n/4}$. We can therefore take a union bound over the possible choices of A. After applying the union bound, we observe that (for our chosen h_n), the probability that *any* A can distinguish between S_n and U_n with advantage greater than $2^{-\Omega(n)}$ is at most

$$\frac{(n/4)2^{n/4} \cdot 2}{2^n\epsilon^2} \leq \frac{1}{2}.$$

For $\epsilon = 2^{-11n/32}$, this implies that for every probabilistic polynomial-time machine A with $k \leq n/4$ bits of nonuniform advice, A distinguishes between S_n and U_n with negligible advantage (for sufficiently large n). □

Lemma A.7. *For all n, consider $\mathcal{H}_{n,m}$ to be the pairwise independent hash family described above where $m = 5n/16$ as in Lemma A.6. Choose an element $h_n \in \mathcal{H}_{n,m}$ uniformly at random and define $S_n = \{x \in \{0,1\}^n : h_n(x) = 0^m\}$. For sufficiently large n, with probability at least $3/4$, S_n exhibits the property that for all probabilistic polynomial-time machines B with $k \leq n/4$ bits of nonuniformity, it is infeasible for B to generate any element of S_n except with negligible probability. That is, there exists a negligible function ϵ such that for all sufficiently large n,*

$$\Pr_{r \in \{0,1\}^{q(n)}}[B(x, r) \in S_n] \leq \epsilon(n).$$

.

Proof. Fix a probabilistic polynomial-time machine B with $k \leq n/4$ bits of nonuniformity.

$$\mathrm{E}_{S_n}[\Pr_r[B(r) \in S_n]] = \mathrm{E}_r[\Pr_{S_n}[B(r) \in S_n]] = \mathrm{E}_r[\Pr_{h_n}[h_n(B(r)) = 0^m]] = \frac{1}{2^m}$$

By Markov's Inequality:

$$\Pr\left[\Pr_r[B(x,r) \in S_n] \geq \epsilon\right] \leq \frac{\mathrm{E}_{S_n}[\Pr_r[B(r) \in S_n]]}{\epsilon} \leq \frac{1}{2^m \epsilon}$$

As in the proof of Lemma A.6, we need to consider at most $(n/4)2^{n/4}$ possible machines B, therefore by a union bound over the choice of B we get that:

$$Pr\left[\exists B \text{ s.t. } Pr_r[B(x,r) \in S_n] \geq \epsilon\right] \leq \frac{n2^{n/4}}{2^{5n/16}\epsilon} = O(n2^{-n/16}\epsilon^{-1})$$

If we let $\epsilon = 2^{-n/32}$ then we observe that for sufficiently large n, the probability (over the choice of h_n) that no probabilistic polynomial-time machine B with $k \leq n/4$ bits of nonuniformity generates an element of S_n with probability greater than ϵ is at least $3/4$. □

Lemma A.8. *Let $\mathcal{H}_{n,m}$ be the pairwise independent hash family described above (with $m = 5n/16$). Choose an element $h_n \in \mathcal{H}_{n,m}$ uniformly at random and define the set $S_n = \{x \in \{0,1\}^n : h_n(x) = 0^m\}$. For all n there exists a string π_n of length $q(n)$ such that:*

(a) There exists a probabilistic polynomial-time machine D such that $D(\pi_n, x) = 1$ if $x \in S_n$ and $D(\pi_n, x) = 0$ else.

(b) There exists a probabilistic expected polynomial-time machine E such that on input π_n, E returns a uniformly random element of S_n.

Proof. $h_n \in \mathcal{H}_{n,m}$ therefore $h_n(x) = a \cdot x + b$. Define $\pi_n = a\#b$ (i.e. π_n is an encoding of the coefficients of h_n). This advice string has length $q(n) = (2n+1)$.

- Part (a): We define a probabilistic Turing machine D that given inputs $\pi_n = a\#b$ and x does the following. It calculates $h_{a,b}(x) = a \cdot x + b$. If the first m bits are all zero, the D outputs 1, else it outputs 0. D clearly runs in polynomial time as desired.

- Part (b): We define a probabilistic Turing machine E that given input $\pi_n = a\#b$ does the following.

  If $a = 0^n$, E begins by determining whether or not $b = 0^m$. If so, it returns a randomly chosen element of $\{0,1\}^n = S_n$. If $a = 0^n$ and $b \neq 0^m$ then $S_n = \emptyset$ so E simply halts.

  If $a \neq 0^n$, E chooses a string r of length $n - m$ uniformly at random, determines the set P of solutions to the equation $h_{a,b}(z) = 0^m \circ r$, and returns an element chosen uniformly at random from the set P.

  For any $z \in S_n$, recall that by our definition of S_n, $h_n(z) = 0^m \circ s$ for some $s \in \{0,1\}^{n-m}$. The probability that E returns a fixed z is therefore given by: $\Pr[r = s] \cdot \frac{1}{|P|} = \frac{1}{2^{n-m} \cdot |P|}$. Since $a \neq 0^n$ then $P = \{(0^m \circ r - b) \cdot a^{-1}\}$. Since $|P|$ does not depend on the choice of z, this probability is independent of the choice of z, therefore E returns a uniformly random element of S_n.

□

Having established these three intermediate results, the existence of $\frac{n}{4}$-nonuniform efficiently-evasive pseudorandom ensembles follows.

Proof of Theorem A.1 Fix a polynomial $p(n)$ and for all n, consider $\mathcal{H}_{n,m}$ to be the pairwise independent hash family described above (with $m = 5n/16$).

Choose an element $h_n \in \mathcal{H}_{n,m}$ uniformly at random and define the corresponding set $S_n = \{x \in \{0,1\}^n : h_n(x) = 0^m\}$. By Lemma A.6 with probability at least 3/4, S_n is indistinguishable from U_n from the point of view of any probabilistic polynomial-time machine A. Similarly by Lemma A.7 with probability at least 3/4, no probabilistic polynomial time machine B can generate an element of S_n. It is therefore clear that for all (sufficiently large) values of n there must exist some $h_n \in \mathcal{H}_{n,m}$ and a corresponding S_n such that both of the above properties hold.

For this S_n, by Lemma A.8 there exist probabilistic polynomial-time machines D and E as required. The result follows immediately. $\square$