Sparse Pseudorandom Distributions

(Extended Abstract)

Oded Goldreich
Hugo Krawczyk

Computer Science Dept.
Technion
Haifa, Israel

Abstract. Pseudorandom distributions on n-bit strings are ones which cannot be efficiently distinguished from the uniform distribution on strings of the same length. Namely, the expected behavior of any polynomial-time algorithm on a pseudorandom input is (almost) the same as on a random (i.e. uniformly chosen) input. Clearly, the uniform distribution is a pseudorandom one. But do such trivial cases exhaust the notion of pseudorandomness? Under certain intractability assumptions the existence of pseudorandom generators was proven, which in turn implies the existence of non-trivial pseudorandom distributions. In this paper we investigate the existence of pseudorandom distributions, using no unproven assumptions.

We show that *sparse* pseudorandom distributions do exist. A probability distribution is called *sparse* if it is concentrated on a negligible fraction of the set of all strings (of the same length). It is shown that sparse pseudorandom distributions can be generated by probabilistic (non-polynomial time) algorithms, and some of them are not statistically close to any distribution induced by probabilistic polynomial-time algorithms.

Finally, we show the existence of probabilistic algorithms which induce pseudorandom distributions with *polynomial-time evasive* support. Any polynomial-time algorithm trying to find a string in their support will succeed with negligible probability. A consequence of this result is a proof that the original definition of zero-knowledge is not robust under sequential composition. (This was claimed before, leading to the introduction of more robust formulations of zero-knowledge.)

First author was supported by grant No. 86-00301 from the United States - Israel Binational Science Foundation (BSF), Jerusalem, Israel.

G. Brassard (Ed.): Advances in Cryptology - CRYPTO '89, LNCS 435, pp. 113-127, 1990.

1. INTRODUCTION

In recent years, randomness has became a central notion in diverse fields of computer science. Randomness is used in the design of algorithms in fields as computational number theory, computational geometry, parallel and distributed computing, and is of course crucial to cryptography. Since in most cases the interest is in the behavior of efficient algorithms (modeled by polynomial-time computations), the fundamental notion of pseudorandomness arises. Pseudorandom distributions are those distributions which cannot be efficiently distinguished from the uniform distribution on strings of the same length.

The importance of pseudorandomness is in the fact that any efficient probabilistic algorithm performs essentially as well when substituting its source of unbiased coins by a pseudorandom sequence. Algorithms can therefore be analyzed assuming they use unbiased coin tosses, and later implemented using pseudorandom sequences. Such approach is practically beneficial if pseudorandom sequences can be generated more easily than "truly random" ones. This gave rise to the notion of a pseudorandom generator - an efficient deterministic algorithm which expands random seeds into longer pseudorandom sequences.

Most of the previous work on pseudorandomness has in fact focused on pseudorandom generators. Blum and Micali [BM] and Yao [Y] suggested the basic definitions and showed that pseudorandom generators can be constructed under certain intractability assumptions [1]. Several works [GGM, LR, L1, L2, GKL, ILL] further developed this direction. An important aspect of pseudorandom generation, namely its utility for deterministic simulation of randomized complexity classes, is further studied in [NW].

In our paper we investigate the notion of pseudorandomness when decoupled from the notion of efficient generation. The investigation will be carried out using no unproven assumptions. The first question we address is the existence of non-trivial pseudorandom distributions. That is, pseudorandom distributions that are neither the uniform distribution nor statistically close to it [2]. Yao [Y] presents a particular example of such a distribution. Further properties of such distributions are developed here.

We prove the existence of *sparse* pseudorandom distributions. A distribution is called sparse if it is concentrated on a negligible part of the set of all strings of the same length. For example, given a positive constant $\delta < 1$ we construct a probability distribution concentrated on $2^{\delta k}$ of the strings of length k which cannot be distinguished from the uniform distribution on the set of all k-bit strings (and hence is pseudorandom).

[1] Intractability assumptions in this approach are inavoidable as long as we cannot prove the existence of one-way functions and, in particular, that $P \neq NP$.

[2] The statistical distance between two probability distributions is defined as the sum (over all strings) of the absolute difference between the probabilities they assign to each string.

Sparse pseudorandom distributions can even be uniformly generated by probabilistic algorithms (that run in non-polynomial time). These generating algorithms use less random coins than the number of pseudorandom bits they produce. Viewing these algorithms as generators which expand randomly selected short strings into much longer pseudorandom sequences, we can exhibit generators achieving subexponential expansion rate. This expansion is optimal since no generator expanding strings into exponential longer ones can induce a pseudorandom distribution (which passes non-uniform tests). On the other hand, one can use the subexponential expansion property in order to construct non-uniform generators of size slightly super-polynomial. (We stress that the existence of non-uniform polynomial-size generators would separate non-uniform-P from non-uniform-NP, which would be a major breakthrough in Complexity Theory).

We also show the existence of sparse pseudorandom distributions that cannot be generated or even approximated by efficient algorithms. Namely, there exist pseudorandom distributions that are statistically far from any distribution which is induced by any probabilistic polynomial-time algorithm. In other words, even if efficient pseudorandom generators exist, they do not exhaust (nor even in an approximative sense) all the pseudorandom distributions.

A stronger notion is that of evasive probability distributions. These probability distributions have the property that any efficient algorithm will fail to find strings in their support [3] (except with a negligible probability). Certainly, evasive probability distributions are sparse, and even cannot be efficiently approximated by probabilistic algorithms. We show the existence of evasive pseudorandom distributions.

Finally, we present an interesting application of these results to the field of zero-knowledge interactive proofs. It has been claimed [F] that the original definition of zero-knowledge (which appeared in [GMR1]) is not robust under sequential composition (and thus more robust variants were introduced [O,GMR2,TW,F]). However, no rigorous proof of this claim has been given to date. Using evasive pseudorandom distributions we construct a zero-knowledge protocol which reveals significant information when executed twice in a sequence.

2. DEFINITIONS

The formal definition of pseudorandomness (given bellow) is stated in asymptotical terms, so we shall not discuss single distributions but collections of probability distributions, called probability ensembles.

[3] The support of a probability distribution is the set of elements that it assigns non-zero probability.

Definition: A *probability ensemble* Π is a collection of probability distributions $\{\pi_k\}_{k \in K}$, such that K is an infinite set of indices (nonnegative integers) and for every $k \in K$, π_k is a probability distribution on the set of (binary) strings of length k.

In particular, an ensemble $\{\pi_k\}_{k \in K}$ in which π_k is a uniform distribution on $\{0,1\}^k$ is called a *uniform ensemble*.

Next, we give a formal definition of a pseudorandom ensemble. This is done in terms of polynomial indistinguishability between ensembles.

Definition: Let $\Pi = \{\pi_k\}$ and $\Pi' = \{\pi_k'\}$ be two probability ensembles. Let T be a probabilistic polynomial time algorithm outputting 0 or 1 (T is called a *statistical test*). Denote by $p_T(k)$ the probability that T outputs 1 when fed with an input selected according to the distribution π_k. Similarly, $p_T'(k)$ is defined with respect to π_k'. The test T *distinguishes* between Π and Π' if and only if there exists a constant c >0 and infinitely many k's such that $|p_T(k) - p_T'(k)| > k^{-c}$. The ensembles Π and Π' are called *polynomially indistinguishable* if there exists no polynomial-time statistical test that distinguish between them.

Definition: A probabilistic ensemble is called *pseudorandom* if it is polynomially indistinguishable from a uniform ensemble.

Remark: Some authors define pseudorandomness by requiring that pseudorandom ensembles be indistinguishable from uniform distributions even by *non-uniform* (polynomial) tests. We stress that the results (and proofs) in this paper also hold for these stronger definitions.

In this work we are interested in the question of whether non-trivial pseudorandom ensembles can be effectively sampled by means of probabilistic algorithms. The following definition capture the notion of 'samplability'.

Definition: A *sampling algorithm* is a probabilistic algorithm A that on input a string of the form 1^n, outputs a string of length n. The *probabilistic ensemble* Π^A *induced by a sampling algorithm* A is defined as $\{\pi_n^A\}_n$, where π_n^A is the probabilistic distribution such that for any $y \in \{0,1\}^n$, $\pi_n^A(y) = Prob(A(1^n) = y)$, where the probability is taken over the coin tosses of algorithm A. A *samplable* ensemble is a probabilistic ensemble induced by a sampling algorithm. If the sampling algorithm uses, on input 1^n, less than n random bits then we call the ensemble *strongly-samplable*.

Traditionally, pseudorandom generators are defined as *deterministic* algorithms expanding short seeds into longer bit strings. With the above definitions one can define them as strong-sampling algorithms (the seed is viewed as the random coins for the sampling algorithm).

We consider as trivial, pseudorandom ensembles that are close to a uniform ensemble. The meaning of "close" is formalized in the next definition.

Definition: Two probabilistic ensembles Π and Π' are *statistically close* if for any positive c and any sufficiently large n, $\sum_{x \in \{0,1\}^n} |\pi_n(x) - \pi_n'(x)| < n^{-c}$.

A special case of non-trivial pseudorandom ensembles are those ensembles we call "sparse".

Definition: A probabilistic ensemble is called *sparse* if (for sufficiently large n's) the support of π_n is a set of *negligible* size relative to the set $\{0,1\}^n$ (i.e for every $c>0$ and sufficiently large n, $|support(\pi_n)| < n^{-c}\, 2^n$).

Clearly, a sparse pseudorandom ensemble cannot be statistically close to a uniform ensemble.

Notation: I_k will denote the set $\{0,1\}^k$.

3. THE EXISTENCE OF SPARSE PSEUDORANDOM ENSEMBLES

The main result in this section is the following Theorem.

Theorem 1: There exist strongly-samplable sparse pseudorandom ensembles.

In order to prove this theorem we present an ensemble of sparse distributions which are pseudorandom even against non-uniform distinguishers. These distributions assign equal probability to the elements in their support. We use the following definition.

Definition: A set $S \subseteq I_k$ is called $(\tau(k), \varepsilon(k))$*-pseudorandom* if for any (probabilistic) circuit C of size $\tau(k)$ with k inputs and a single output

$$| p_C(S) - p_C(I_k) | \leq \varepsilon(k)$$

where $p_C(S)$ (resp. $p_C(I_k)$) denotes the probability that C outputs 1 when given elements of S (resp. I_k), chosen with uniform probability.

If for a circuit C and a set $S \subseteq I_k$ the above inequality does not hold then we say that the the set S is $\varepsilon(k)$*–distinguished* by the circuit C.

Note that a collection of uniform distributions on a sequence of sets $S_1, S_2, \ldots$ where each S_k is a $(\tau(k), \varepsilon(k))$-pseudorandom set, constitutes a pseudorandom ensemble, provided that both functions $\tau(k)$ and $\varepsilon^{-1}(k)$ are super-polynomial, i.e. grow faster than any polynomial. Our goal is to prove the existence of such a collection in which the ratio $|S_k|/2^k$ is negligibly small.

Remark: In the following we consider only deterministic circuits (tests). The ability to toss coins does not add power to non-uniform tests. Using a standard averaging argument one can show that whatever a probabilistic non-uniform distinguisher C can do, may be achieved by a deterministic circuit in which the "best coins" of C are incorporated.

The next Lemma measures the number of sets which are $\varepsilon(k)$-distinguished by a given circuit. Notice that this result does not depend on the circuit size.

Lemma 2: For any k-input Boolean circuit C, the probability that a random set $S \subseteq I_k$ of size N is $\varepsilon(k)$-distinguished by C is at most $2\exp\left[-2N\varepsilon^2(k)\right]$. (The function exp(·)

denotes exponentiation to natural base).

Proof: Let $L_C(k)$ be the set $\{x \in I_k : C(x)=1\}$. Thus, $p_C(I_k)=\frac{|L_C(k)|}{2^k}$ and $p_C(S)=\frac{|S \cap L_C(k)|}{|S|}$.

Consider the set of strings of length k as a urn containing 2^k balls. Let those balls in $L_C(k)$ be painted white and the others black. The proportion of white balls in the urn is clearly $p_C(I_k)$, and the proportion of white balls in a sample S of N balls from the urn is $p_C(S)$. (We consider here a sample *without* replacement, i.e. sampled balls are not replaced in the urn).

Lemma 2 follows by using the Chernoff-type inequality due to W. Hoeffding [H] (see Appendix)

$$Prob\left[\ |p_C(S)-p_C(I_k)| \geq \varepsilon(k)\right] \ < 2\exp\left[-2N\varepsilon^2(k)\right]$$

where the probability is taken over all the subsets $S \subseteq I_k$ of size N, with uniform probability. ■

The following Lemma states the existence of pseudorandom ensembles composed of uniform distributions with very sparse support.

Lemma 3: Let $k(n)$ be any subexponential function of n (i.e. $k(n)=\exp(o(n))$)[4]. There exist super-polynomial functions $\tau(\cdot)$ and $\varepsilon^{-1}(\cdot)$, and a sequence of sets $S_1, S_2, \ldots$, such that S_n is a $(\tau(k(n)), \varepsilon(k(n)))$-pseudorandom subset of $I_{k(n)}$ and $|S_n|=2^n$.

Proof: Fix n and let $k=k(n)$. We show the existence of a set $S \subseteq I_k$ of size 2^n which is $(\tau(k), \varepsilon(k))$-pseudorandom, where $\tau(\cdot)$ and $\varepsilon^{-1}(\cdot)$ are suitable chosen super-polynomial functions.

The number of Boolean circuits of size $\tau(k)$ is at most $2^{\tau^2(k)}$. Thus, to show the existence of a set S that is not $\varepsilon(k)$-distinguished by any of these circuits it is sufficient to show that each circuit $\varepsilon(k)$-distinguishes at most $2^{-\tau^2(k)}$ of the sets of size 2^n. Using Lemma 2, this holds provided that

$$2^n \varepsilon^2(k) > \tau^2(k) \qquad (1)$$

It is easy to see that for any subexponential function $k(n)$ we can find super-polynomial functions $\varepsilon^{-1}(\cdot)$ and $\tau(\cdot)$ such that inequality (1) holds for each value of n. ■

The following Lemma states that the sparse pseudorandom ensembles presented above are strongly-samplable. This proves Theorem 1.

Lemma 4: Let $k(n)$ be any subexponential function of n. There exist (non-polynomial) generators which expand random strings of length n into pseudorandom strings of length

[4] $o(n)$ denotes any function $f(n)$ such that $\lim_{n\to\infty} f(n)/n=0$

$k(n)$.

Proof: Let $\tau(\cdot)$ and $\varepsilon(\cdot)$ be as in Lemma 3. We construct a generator which on input a seed of length n finds the $(\tau(k(n)), \varepsilon(k(n)))$-pseudorandom set $S_n \subseteq I_{k(n)}$ whose existence is guaranteed by Lemma 3, and uses the n input bits in order to choose a random element from S_n. Clearly, the output of the generator is pseudorandom.

To see that the set S_n can be effectively found, note that it is effectively testable whether a given set S of size 2^n is $(\tau(k), \varepsilon(k))$-pseudorandom. This can be done by enumerating all the circuits of size $\tau(k)$ and computing for each circuit C the quantities $p_C(S)$ and $p_C(I_k)$. Thus, our generator will test all the possible sets $S \subseteq I_k$ of size 2^n until S_n is found. ■

Remark 1: Inequality (1) defines a trade-off between the expansion function $k(n)$ and the size of the tests (circuits) resisted by the generated ensemble. The pseudorandom ensembles we construct may be "very" sparse, in the sense that the expansion function $k(n)$ can be chosen to be very large (e.g. $2^{\sqrt{n}}$). On the other hand if we consider "moderate" expansion functions such as $k(n) = 2n$, we can resist rather powerful tests, e.g. circuits of size $2^{n/4}$.

Remark 2: The subexponential expansion, as allowed by our construction, is optimal since no generator exists which expands strings of length n into strings of length $k(n) = \exp(O(n))$. To see this, consider a circuit of size $k(n)^{O(1)}$ which incorporates the (at most) 2^n output strings of the generator. Clearly, this circuit constitutes a (non-uniform) test distinguishing the output of this generator from the uniform distribution on $I_{k(n)}$.

Remark 3: The subexponential expansion implies that the supports of the resultant pseudorandom distributions are very sparse. More precisely, our construction implies the existence of generators which induce on strings of length k a support of size *slightly* super-polynomial (i.e. of size $k^{u(k)}$ for an arbitrary non-decreasing unbounded function $u(k)$). Thus, by wiring this support into a Boolean circuit, we are able to construct *non-uniform* generators of size slightly super-polynomial. (On input a seed s the circuit (generator) outputs the s-th element in this "pseudorandom" support). Let us point out that an improvement of this result, i.e. a proof of the existence of non-uniform pseudorandom generators of polynomial size, will imply that non-uniform-P ≠ non-uniform-NP !. This follows by considering the language $\{x \in I_k : x \textit{ is in the image of } G\}$, where G is a pseudorandom generator in non-uniform-P. Clearly, this language is in non-uniform-NP, but not in non-uniform-P, otherwise a deciding procedure for it can be transformed into a test distinguishing the output of G from the uniform distribution on I_k.

Remark 4: The (uniform) complexity of the generators constructed in Lemma 4 is slightly super-exponential, i.e. $2^{k^{u(k)}}$, for unbounded $u(\cdot)$. (The complexity is, up to a polynomial factor, $2^{\tau^2(k)} \cdot (2^n + 2^k) \cdot \binom{2^k}{2^n}$, and 2^n is, as in Remark 3, slightly super-polynomial in k). We stress that the existence of pseudorandom generators running in exponential time, and with arbitrary polynomial expansion function, would have interesting consequences in Complexity Theory as BPP $\subseteq \bigcap_{\varepsilon>0} \mathrm{DTIME}(2^{n^\varepsilon})$ [Y, NW].

4. THE COMPLEXITY OF APPROXIMATING PSEUDORANDOM ENSEMBLES

In the previous section we have shown sparse pseudorandom ensembles which can be sampled by probabilistic algorithms running super-exponential time. Whether is it possible to sample pseudorandom ensembles by polynomial-time algorithms or even exponential ones, cannot be proven today without using complexity assumptions. On the other hand, do such assumptions guarantee that each samplable pseudorandom ensemble can be sampled by polynomial, or even exponential means? We give here a negative answer to this question, proving that for any complexity function $\phi(\cdot)$ there exists a samplable pseudorandom ensemble which cannot be sampled nor even "approximated" by algorithms in RTIME(ϕ). The notion of approximation is defined next.

Definition: A probabilistic ensemble Π is *approximated* by a sampling algorithm A if the ensemble Π^A induced by A is statistically close to Π.

The main result of this section is stated in the following Theorem.

Theorem 5: For any complexity (constructive) function $\phi(\cdot)$, there is a strongly samplable pseudorandom ensemble that cannot be approximated by any algorithm whose running time is bounded by ϕ.

Proof: We say that two probability distributions π and π' on a set X are *½–close* if

$$\sum_{x \in X} |\pi(x) - \pi'(x)| < ½ .$$

We say that a sampling algorithm M ½-approximates a set $S \subseteq I_k$ if the probability distribution π_k^M induced by M on I_k and the uniform distribution U_S on S are ½-close.

We show that for any sampling algorithm M most subsets of I_k of size 2^n are not ½-approximated by M (for k sufficienly large with respect to n). This follows from the next Lemma.

Lemma 6: Let π be a probability distribution on I_k. The probability that π and U_S are ½-close, for S randomly chosen over the subsets of I_k of size 2^n, is less than $(1/2)^{k-n-1}$.

Proof: Notice that if two different sets S and T are ½-close to π, then the two sets are close themselves. More precisely, we have that $\sum_{x \in I_k} |U_S(x) - \pi(x)| < \frac{1}{2}$ and $\sum_{x \in I_k} |U_T(x) - \pi(x)| < \frac{1}{2}$. Using the triangle inequality we conclude that $\sum_{x \in I_k} |U_S(x) - U_T(x)| < 1$. Denoting the last sum by σ and the symmetric difference of S and T by D, we have that $|D| \cdot \frac{1}{2^n} < \sigma < 1$ (this follows from the fact that U_S and U_T

assign uniform probability to the 2^n elements of S and T, respectively). But this implies that $|D| < 2^n$, and then (using $|S| + |T| = |D| + 2 \cdot |S \cap T|$) we get $|S \cap T| > 2^n/2$.

Let T be a particular subset of I_k of size 2^n which is ½-close to π. From the above argument it follows that the collection of subsets of size 2^n which are ½-close to π is included in the collection $\{S \subseteq I_k : |S| = 2^n, |S \cap T| > 2^n/2\}$. Thus, we are able to bound the probability that π is ½-close to a random set S of size 2^n, by the probability of the following experiment. Fix a set $T \subseteq I_k$ of size 2^n, and take at random a set S of 2^n elements among all the strings in I_k. We are interested in the probability that $|S \cap T| > 2^n/2$. Clearly, the expectation of $|S \cap T|$ is $\frac{|S| \cdot |T|}{2^k}$. Using Markov inequality for nonnegative random variables we have

$$Prob\left[|S \cap T| > \frac{2^n}{2}\right] \cdot \frac{2^n}{2} < \frac{|S| \cdot |T|}{2^k}$$

and then

$$Prob\ (|S \cap T| > 2^n/2) < 2/2^{k-n} \tag{2}$$

The lemma follows. □

We now extend the pseudorandom generator constructed in Lemma 4, in order to obtain a generator for a pseudorandom ensemble which is not approximated by any ϕ-time sampling algorithm. On input a string of length n, the generator proceeds as in Lemma 4. Once a $(\tau(k(n)), \varepsilon(k(n)))$-pseudorandom subset S_n is found, the generator checks whether S_n is ½-approximated by some of the first n Turing machines, in some canonical enumeration, by running each of them as a sampling algorithm for $\phi(k(n))$ steps. Clearly, it is effectively testable whether a given machine M ½-approximates a given set S. If the set S_n is ½-approximated by some of these machines, it is discarded and the next $S \subseteq I_k$, $|S| = 2^n$ is checked (first for pseudorandomness and then for approximation).

In section 3 we have actually shown that the probability that a set S is $(\tau(k(n)), \varepsilon(k(n)))$-pseudorandom is almost 1. On the other hand, the probability that a set S is ½-approximated by n sampling machines is, using Lemma 6, less than $n/2^{k(n)-n-1}$. For suitable $k(\cdot)$, e.g. $k(n) \geq 2n$, this probability is negligible. Thus, we are guaranteed to find a set S_n which is $(\tau(k(n)), \varepsilon(k(n)))$-pseudorandom as well as not ½-approximated by the first n sampling algorithms running ϕ-time. The resultant ensemble is as stated in the theorem. ■

Remark: The result in Theorem 5 clearly relies on the fact that the sampling algorithms we have run are uniform ones. Nevertheless, if we use Hoeffding inequality to bound the left side in (2), we get a much better bound, which implies that for any constant $\alpha < 1$, there existe strongly-samplable pseudorandom ensembles that cannot be approximated by Boolean circuits of size $2^{\alpha n}$.

5. POLYNOMIAL-TIME EVASIVE PSEUDORANDOM ENSEMBLES

In this section we prove the existence of pseudorandom ensembles which have the property that no polynomial-time sampling algorithm will output an element in their support, except for a negligible probability.

Definition: A probability ensemble $\Pi = \{\pi_k\}_{k \in K}$ is called *polynomial-time evasive* if for any polynomial-time sampling algorithm A, any constant c and sufficiently large k,

$$Prob\left[A(1^k) \in support(\pi_k) \right] < k^{-c}$$

($support(\pi_k)$ denotes the set $\{x \in I_k : \pi_k(x) > 0\}$).

Notice that evasiveness does not imply pseudorandomness. For example, any evasive ensemble remains evasive if we add to each string in the support a leading '0', while the resultant distributions are obviously not pseudorandom. On the other hand, an evasive pseudorandom ensemble is clearly sparse.

Following is the main result of this section. An interesting application of this result appears in section 6.

Theorem 7: There exist (strongly-samplable) polynomial-time evasive pseudorandom ensembles.

Proof: The proof outline is similar to the proof of Theorem 5. We again extend the generator of Lemma 4 by testing whether the $(\tau(k(n)), \varepsilon(k(n)))$-pseudorandom set S_n, found by that generator on input of length n, evades the first n Turing machines (run as polynomial-time sampling algorithms). We have to show that for each sampling algorithm M there is a small number of sets $S \subseteq I_k$ of size 2^n for which machine M outputs an element of S with significant probability. Throughout this proof we shall consider as "significant" a probability that is greater than $2^{3n}/2^k$. (This choice is motivated by a later application of this Theorem. Any negligible portion suffices here. Thus, we are assuming $k \geq 4n$). We need the following technical Lemma.

Lemma 8: Let π be a fixed probability distribution on a set U of size K. For any $S \subseteq U$ denote $\pi(S) = \sum_{s \in S} \pi(s)$. Then

$$Prob\left[\pi(S) > \varepsilon \right] < \frac{N}{\varepsilon K}$$

where the probability is taken over all the sets $S \subseteq U$ of size N with uniform probability.

Proof: Consider a random sample of N *distinct* elements from the set U. Let X_i ,$1 \leq i \leq N$, be random variables so that X_i assumes the value $\pi(u)$ if the i-th element chosen in the sample is u. Define the random variable X to be the sum of the X_i's (i.e. $X = \sum_{i=1}^{N} X_i$).

Clearly, each X_i has expectation $1/K$ and then the expectation of X is N/K. Using Markov inequality for nonnegative random variables we get

$$Prob(X > \varepsilon) < \frac{E(X)}{\varepsilon} = \frac{N}{\varepsilon K}$$

proving the Lemma. □

Let π_k^M be the probability distribution induced by the sampling algorithm M on I_k. Consider a randomly chosen $S \subseteq I_k$ of size 2^n. Lemma 8 states that

$$Prob\left[\pi_k^M(S) > \frac{2^{3n}}{2^k}\right] < \frac{1}{2^{2n}} \tag{3}$$

Thus, we get that only $1/2^{2n}$ of the subsets S fail the evasivity test for a single machine. Running n such tests the portion of failing sets is at most $n/2^{2n}$. Therefore, there exists a set passing all the distinguishing and evasivity tests. (Actually, most of the sets of size 2^n pass these tests). This completes the proof of the Theorem. ■

Remark 1: Actually, we have proven that for any uniform time-complexity class **C**, there exist pseudorandom ensembles which evades any sampling algorithm of the class **C**. Notice that no restriction on the running time of the sampling machines is required. It is interesting to note that we cannot find ensembles evading the output of non-uniform circuits of polynomial-size, since for each set S there exists a circuit which outputs an element of S with probability 1. Thus, the results in this sections imply the results of section 4 on unapproximability by uniform algorithms, but not the unapproximability by non-uniform circuits (see remark after the proof of Theorem 5).

Remark 2: For the results in section 6, we need a slightly stronger result than the one stated in Theorem 3. This application requires a pseudorandom ensemble that evades not only sampling algorithm receiving 1^k as the only input, but also algorithms having an additional input of length n (the parameters k and n are as defined above). The proof of Theorem 3 remains valid also in this case. This follows by observing that each such algorithm defines 2^n distributions, one for each possible input of length n. Thus, the n algorithms we run in the above proof contribute $n \cdot 2^n$ distributions. Using the above bound (3) we can guarantee the existence of sets S that evade any of these distributions.

6. ON THE SEQUENTIAL COMPOSITION OF ZERO-KNOWLEDGE PROTOCOLS

In this section we apply the results of section 5 in order to demonstrate a weakness in the *original* definition of zero-knowledge interactive proofs. Before presenting this result we shall give an informal outline of the notions of interactive-proofs and zero-knowledge. For formal and complete definitions, as well as the basic results concerning these concepts, the reader is referred to [GMR1, GMW].

An *interactive proof* for a language L is a two-sided protocol in which a computationally powerful *Prover* convinces a probabilistic polynomial-time *Verifier* that their common input x belongs to the language L. If the assertion is true, i.e. $x \in L$, then the verifier will be convinced of its validity with very high probability. If the assertion is false then the probability to convince the verifier of the contrary is negligibly small, no matter how the prover behaves during the execution of the protocol.

An interactive proof is called *zero-knowledge* if no polynomial-time verifier (even one that arbitrarily deviates from the predetermined program) gains no information from the execution of the protocol except the knowledge whether x belongs to L. That is, any polynomial-time computation based on the conversations with the prover, on input $x \in L$, can be simulated by a probabilistic polynomial-time machine ("the simulator") that gets x as its only input. More precisely, let $[P, V^*](x)$ denote the probability distribution generated by the interactive machine (verifier) V^* which interacts with the prover P on input $x \in L$. We say that an interactive proof is *zero-knowledge* if for all probabilistic polynomial-time machines V^*, there exists a probabilistic polynomial-time algorithm M_{V^*} (called a *simulator*) that on input $x \in L$ produces a probability distribution $M_{V^*}(x)$ that is polynomially indistinguishable from the distribution $[P, V^*](x)$.
(This notion of zero-knowledge is also called *computational zero-knowledge*. The results in this section concern only this notion [5]).

A natural requirement from the notion of zero-knowledge proofs is that the information obtained by the verifier during the execution of a zero-knowledge protocol will not enable him to extract any additional knowledge from subsequent executions of the same protocol. That is, it would be desirable that the *sequential composition* of zero-knowledge protocols would yield a protocol which is itself zero-knowledge. Such a property is crucial for applications of zero-knowledge protocols in cryptography. See [O] for a formal definition of "sequential composition", and further motivation of its need.

In this section we prove that the original definition of (computational) zero-knowledge introduced by Goldwasser, Micali and Rackoff in [GMR1] (as we have sketched above) *is not closed* under sequential composition. Several authors have previously observed that this definition *probably* does not guarantee its robustness under sequential composition, and hence have introduced more robust formulations of zero-knowledge [GMR2, O, TW, F].

Feige [F] proposed a protocol that appears to be zero-knowledge when executed once but reveals significant information during a second execution. Using the underlying idea of this protocol and the results of the previous section we prove the following

Theorem 9: Computational Zero-Knowledge ([GMR1] formulation) is not closed under sequential composition.

[5] Other definitions were proposed in which it is required that the distribution generated by the simulator is *identical* to the distribution of conversations between the verifier and the prover (*perfect* zero-knowledge), or at least statistically close (*statistical* zero-knowledge). See [Fo,GMR2] for further details.

Proof: Let G be a generator as constructed in Theorem 7, i.e. its output induces a pseudorandom and polynomial-time evasive ensemble. Let G expand strings of length n into strings of length $k=4n$, and let $S_n \subseteq I_{4n}$ be the set of images of G on strings of length n. Also, let K be a hard Boolean function, in the sense that the language $L_K = \{x : K(x)=1\}$ is not in BPP.

We define the following interactive-proof protocol $<P,V>$ for the language $L=\{0,1\}^*$. (Obviously, this language has a trivial zero-knowledge proof in which the verifier accepts every input, without carrying out any interaction. We intentionally modify this protocol in order to demonstrate a zero-knowledge protocol which fails sequential composition).

Let x be the common input for P and V, and let n denote the length of x. The verifier V begins by sending to the prover a randomly chosen string s of length $4n$. The prover P checks whether $s \in S_n$ If this is the case then P sends to V the value of $K(x)$. Otherwise ($s \notin S_n$), P sends to V a string s_0 randomly selected from S_n. In any case the verifier accepts the input x (as belonging to L).

We stress that the same generator G is used in all the executions of the protocol. Thus, the sets S_n do not depend on the specific input to the protocol, but only on its length. Therefore, the string s_0, obtained by the 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 $K(x')$, for any x' of length n. Indeed, consider a second execution of the protocol, this time on input x'. A "cheating" verifier which sends the string $s=s_0$ instead of chosing it at random, will get the value of $K(x')$ from the prover. Observe that this cheating verifier obtain information that cannot be computed by itself. There is no way to simulate in probabilistic polynomial-time the interaction in which the prover sends the value of $K(x')$. Otherwise the language L_K is in BPP.

Thus, it is clear that the protocol is not zero-knowledge when composed twice. On the other hand, the protocol is zero-knowledge (when executed the first time). To show that, 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 message sent by the prover during the protocol. It sends the value of $K(x)$, in case that the string s sent by the verifier belongs to the set S_n, and a randomly selected element of S_n, otherwise. By the evasivity condition of the set S_n, there is only a negligible probability that the first case holds. Indeed, no probabilistic polynomial-time machine (in our case, the verifier) can find such a string $s \in S_n$, except with insignificant probability (no matter the input x to the protocol is; see Remark 2 following the proof of Theorem 7). Thus, the simulator can succeed by always simulating the second possibility, i.e. the sending of a random element s_0 from S_n. This step is simulated by randomly choosing s_0 from I_{4n} rather than from S_n. The indistinguishability of this choice from the original one follows from the fact that each S_n is a pseudorandom subset of I_{4n}, and that s_0 is chosen at random from S_n. ■

Remark: For any language L having a zero-knowledge interactive proof, one can present a zero-knowledge protocol which fails sequential composition. Simply, modify the original protocol for L as done in the above proof. (There, we have arbitrarily chosen $L = \{0,1\}^*$).

ACKNOWLEDGEMENTS

We would like to thank Micha Hofri for referring us to Hoeffding inequality, and to Benny Chor and Eyal Kushilevitz for helpful comments.

REFERENCES

[BM] Blum, M., and Micali, S., "How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits", *SIAM Jour. on Computing*, Vol. 13, 1984, pp. 850-864.

[C] Chernoff, H., "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations", *Annals of Mathematical Statistics*, Vol. 23, 1952, pp. 493-507.

[F] Feige, U., M.Sc. Thesis, Weizmann Institute, 1987.

[Fo] Fortnow, L., "The Complexity of Perfect Zero-Knowledge", *Proc. of 19th STOC*, 1987, pp. 204-209.

[GGM] Goldreich, O., S. Goldwasser, and S. Micali, "How to Construct Random Functions", *Jour. of ACM*, Vol. 33, No. 4, 1986, pp. 792-807.

[GKL] Goldreich, O., Krawczyk, H. and Luby, M., "On the Existence of Pseudorandom Generators", *Proc. of the 29th IEEE Symp. on Foundation of Computer Science*, 1988, pp. 12-24.

[GMW] Goldreich, O., S. Micali, and A. Wigderson, "Proofs that Yield Nothing But their Validity and a Methodology of Cryptographic Protocol Design", *Proc. 27th FOCS*, 1986, pp. 174-187.

[GMR1] Goldwasser, S., S. Micali, and C. Rackoff, "Knowledge Complexity of Interactive Proofs", *Proc. 17th STOC*, 1985, pp. 291-304.

[GMR2] Goldwasser, S., S. Micali, and C. Rackoff, "Knowledge Complexity of Interactive Proofs", *SIAM Jour. on Computing*, Vol. 18, 1989, pp. 186-208.

[H] Hoeffding W., "Probability Inequalities for Sums of Bounded Random Variables", *Journal of the American Statistical Association*, Vol. 58, 1963, pp. 13-30.

[ILL] Impagliazzo, R., L.A., Levin and M.G. Luby, "Pseudo-Random Generation from One-Way Functions", *Proc. 21st STOC*, 1989, pp. 12-24.

[L1] L.A. Levin, "One-Way Function and Pseudorandom Generators", *Combinatorica*, Vol. 7, No. 4, 1987, pp. 357-363.

[L2] L.A. Levin, "Homogeneous Measures and Polynomial Time Invariants", *Proc. of the 29th IEEE Symp. on Foundation of Computer Science*, 1988, pp. 36-41.

[LR] M. Luby and C. Rackoff, "How to Construct Pseudorandom Permutations From Pseudorandom Functions", *SIAM Jour. on Computing*, Vol. 17, 1988, pp. 373-386.

[NW] Nissan, N. and Wigderson, A., "Hardness vs. Randomness", *Proc. of the 29th IEEE Symp. on Foundation of Computer Science*, 1988, pp. 2-11.

[O] Oren, Y., ''On the Cunning Power of Cheating Verifiers: Some Observations About Zero-Knowledge Proofs'', *Proc. of the 28th IEEE Symp. on Foundation of Computer Science*, 1987, pp. 462-471.

[TW] Tompa, M., and H. Woll, ''Random Self-Reducibility and Zero-Knowledge Interactive Proofs of Possession of Information'', *Proc. of the 28th IEEE Symp. on Foundation of Computer Science*, 1987, pp. 472-482.

[Y] Yao, A.C., "Theory and Applications of Trapdoor Functions", *Proc. of the 23rd IEEE Symp. on Foundation of Computer Science*, 1982, pp. 80-91.

APPENDIX: HOEFFDING INEQUALITY [H]

Suppose a urn contains u balls of which w are white and $u-w$ are black. Consider a random sample of s balls from the urn (without replacing any balls in the urn at any stage).

Hoeffding inequality states that the proportion of white balls in the sample is close, with high probability, to its expected value, i.e. to the proportion of white balls in the urn. More precisely, let x be a random variable assuming the number of white balls in a random sample of size s. Then, for any ε, $0 \le \varepsilon \le 1$

$$Prob\left[\ \left| \frac{x}{s} - \frac{w}{u} \right| \ge \varepsilon \right] < 2\, e^{-2s\varepsilon^2}$$

This bound is oftenly used for the case of binomial distributions (i.e when drawn balls are replaced in the urn). The inequality for that case is due to H. Chernoff [C]. More general inequalities appear in Hoeffding's paper [H], as well as a proof that these bounds apply also for the case of samples without replacement.