Theoretical Computer Science 276 (2002) 449–459

Theoretical Computer Science

www.elsevier.com/locate/tcs

Note

A comparison of two approaches to pseudorandomness

Yongge Wang

Certicom Research, Certicom Corporation, 5520 Explorer Drive, 4th floor, Mississauga, Ont., Canada L4W 5L1

Received February 2000; received in revised form February 2001; accepted July 2001
Communicated by M. Ito

Abstract

The concept of pseudorandomness plays an important role in cryptography. In this note, we contrast the notions of complexity-theoretic pseudorandom strings (from algorithmic information theory) and pseudorandom strings (from cryptography). For example, we show that we can easily distinguish a complexity-theoretic pseudorandom ensemble from the uniform ensemble. Both notions of pseudorandom strings are uniformly unpredictable; in contrast with pseudorandom strings, complexity-theoretic pseudorandom strings are not polynomial-time unpredictable.

Keywords: Pseudorandom generators; Kolmogorov complexity; Unpredictability

1. Introduction

There are two possible approaches to define the concept of randomness. The “ontological” approach looks at the “simplest description” of a string and declares random a string which has roughly the same length as its simplest description. Algorithmic information theory—initiated by Solomonoff [16], Kolmogorov [12], and Chaitin [5]—defines the simplest description of a string x by the minimal input necessary to a universal algorithm to produce x. Depending upon the choice of the universal algorithm, two theories have emerged: Kolmogorov–Chaitin theory in which one uses a universal Turing machine and Chaitin theory relying on a self-delimiting universal Turing machine (see, e.g., [6]). Only the second theory is compatible with a theory of random infinite sequences. The first theory has been relativized (in time or space); it led to some complexity-theoretic definitions of pseudorandom strings. These notions have been very useful in many places (see [11] for a recent survey), but as Goldreich [9] observed, not in designing pseudorandom generators.

E-mail address: ywang@certicom.com (Y. Wang).

PII: S0304-3975(01)00311-5

Cryptography suggests an alternative "behaviouristic" approach to pseudorandomness. Instead of considering the "explanation" of a phenomenon, it takes into account the phenomenon's effect on the environment. A string is said to be pseudorandom if no efficient observer can distinguish it from a uniformly chosen string of the same length. The underlying postulate is that objects that cannot be told apart by efficient procedures are considered equivalent. This approach naturally leads to the concept of pseudorandom generator, which is fundamental for cryptography.

Our aim is to contrast these two definitions of pseudorandom strings. For example, we show that we can easily distinguish a complexity-theoretic pseudorandom ensemble from the uniform ensemble. Both notions of pseudorandom strings are uniformly unpredictable; in contrast with pseudorandom strings, complexity-theoretic pseudorandom strings are not polynomial-time unpredictable.

We close this section by introducing some notation we will use. The set of non-negative integers is denoted by $\mathcal{N}$. By $\{0,1\}^*$ we denote the set of (finite) binary strings; $\{0,1\}^n$ is the set of binary strings of length n. The length of a string x is denoted by $|x|$. For a string $x \in \{0,1\}^*$ and an integer number $n \geqslant 1$, $x[1..n]$ denotes the initial segment of length n of x ($x[1..n] = x$ if $|x| \leqslant n$) and $x[i]$ denotes the ith bit of x, i.e., $x[1..n] = x[1] \ldots x[n]$.

2. Computational indistinguishability

Computational indistinguishability is a fundamental concept in cryptography. The following paragraph is quoted from [9, p. 87]:

> The concept of efficient computation leads naturally to a new kind of equivalence between objects. Objects are considered to be computationally equivalent if they cannot be told apart by any efficient procedure. Considering indistinguishable objects as equivalent is one of the basic paradigms of both science and real-life situations. Hence, we believe that the notion of computational indistinguishability is fundamental.

Two distributions are called computationally indistinguishable if no efficient algorithm can tell them apart. Given an efficient algorithm D, we consider the probability that D accepts (e.g., outputs 1 on input) a string taken from the first distribution. Likewise, we consider the probability that D accepts a string taken from the second distribution. If these two probabilities are close, we say that D does not distinguish the two distributions.

Typically, an *ensemble* of the form $X = \{X_n\}_{n \in \mathcal{N}}$ has each X_n ranging over strings of length n. We will use $U = \{U_n\}_{n \in \mathcal{N}}$ to denote the uniform ensemble, that is, U_n denotes a random variable uniformly distributed over $\{0,1\}^n$.

Definition 2.1 (*Goldreich [9]*). Two ensembles, $X = \{X_n\}_{n \in \mathcal{N}}$ and $Y = \{Y_n\}_{n \in \mathcal{N}}$ are *indistinguishable in polynomial-time* if for every probabilistic polynomial-time algorithm D, every polynomial $p(\cdot)$, and all sufficiently large n such that the following two conditions are satisfied

$$\sum_{x\in\{0,1\}^n} Prob(X_n = x) \neq 0 \quad \text{and} \quad \sum_{x\in\{0,1\}^n} Prob(Y_n = x) \neq 0,$$

the following inequality holds:

$$|Prob(D(X_n) = 1) - Prob(D(Y_n) = 1)| < \frac{1}{p(n)}.$$

The probabilities in the above definition are taken over the corresponding random variables X_i (or Y_i) and the internal coin tosses of the algorithm D.

Definition 2.2 (*Goldreich [9]*). Let $U = \{U_n\}_{n\in\mathcal{N}}$ be the uniformly distributed ensemble, and $X = \{X_n\}_{n\in\mathcal{N}}$ be an ensemble. The ensemble X is called *pseudorandom* if X and U are indistinguishable in polynomial-time.

Definition 2.3 (*Goldreich [9]*). A *pseudorandom generator* is a deterministic polynomial-time algorithm G from strings to strings satisfying the following two conditions:

1. There exists a function $l:\mathcal{N}\to\mathcal{N}$ such that $l(n)>n$ for all $n\in\mathcal{N}$, and $|G(x)| = l(|x|)$, for all $x\in\{0,1\}^*$.
2. The ensemble $\{G(U_n)\}_{n\in\mathcal{N}}$ is pseudorandom.

For example, Blum et al. [2] proposed the following BBS [2] pseudorandom generator.

Example 1. Let both p and q be distinct primes congruent to $3 \bmod 4$, $N = pq$, and $l(n)>n$ be a polynomial. For each number $x<N$ and $i\leqslant l(\log N)$, let $x_{-1}=x$, $x_{i+1} = x_i^2 \bmod N$ and $b_i = parity(x_i)$ where $parity(y)$ denotes the least significant bit of y. Then, the BBS [2] pseudorandom generator is defined as $G(x) = b_0 \ldots b_{l(\log N)}$.

3. No complexity-theoretic pseudorandom ensemble is pseudorandom

Let $\mathrm{RAND}_c = \bigcup_{n\in\mathcal{N}} \mathrm{RAND}_{c,n}$ and $\mathrm{RAND}_c^t = \bigcup_{n\in\mathcal{N}} \mathrm{RAND}_{c,n}^t$ be the sets of Kolmogorov c-random and Kolmogorov t-time bounded c-random strings, respectively, where $c\geqslant 1$ and $t:\mathcal{N}\to\mathcal{N}$ is some time-constructible function such that $t(n)\geqslant n^2$ for all $n\in\mathcal{N}$. That is, for a universal Turing machine M, let

$$\mathrm{RAND}_{c,n} = \{x \in \{0,1\}^n\colon \text{ if } M(y) = x \text{ then } |y| \geqslant |x| - c\}$$

and

$$\mathrm{RAND}_{c,n}^t = \left\{x \in \{0,1\}^n\colon \begin{array}{l}\text{if } M(y) = x \text{ and } M(y) \text{ halts in less}\\ \text{than } t(|x|) \text{ steps, then } |y| \geqslant |x| - c\end{array}\right\}.$$

The strings in RAND_c (respectively RAND_c^t) are called c-random (respectively c-pseudorandom). Let $R_c = \{R_{c,n}\}_{n\in\mathcal{N}}$ and $R_c^t = \{R_{c,n}^t\}_{n\in\mathcal{N}}$ be two ensembles such that $R_{c,n}$ and $R_{c,n}^t$ are uniformly distributed over $\mathrm{RAND}_{c,n}$ and $\mathrm{RAND}_{c,n}^t$, respectively. Our first results show that these two ensembles are not pseudorandom.

Theorem 3.1. *The ensemble* $R_c^t = \{R_{c,n}^t\}_{n\in\mathcal{N}}$ *is not pseudorandom.*

Proof. Define a polynomial-time algorithm D by letting

$$D(x) = \begin{cases} 0 & \text{if } x = 0^{\lfloor \log |x| \rfloor} y \text{ for some } y \in \{0,1\}^*, \\ 1 & \text{otherwise.} \end{cases}$$

It is straightforward to show that

$$\mathrm{RAND}_{c,n}^t \cap \{x \in \{0,1\}^*\colon\ x = 0^{\lfloor \log |x| \rfloor} y \text{ for some } y \in \{0,1\}^*\} = \emptyset$$

for sufficiently large n. Hence, $Prob(D(R_{c,n}^t) = 1) = 1$ and $Prob(D(U_n) = 1) = 1 - 2^{-\lfloor \log n \rfloor}$, for sufficiently large n. That is,

$$|Prob(D(R_{c,n}^t) = 1) - Prob(D(U_n) = 1)| = 2^{-\lfloor \log n \rfloor} \geqslant \frac{1}{n}.$$

This shows that the ensembles $\{R_{c,n}^t\}_{n\in\mathcal{N}}$ and $\{U_n\}_{n\in\mathcal{N}}$ are distinguishable in polynomial-time, hence $R_c^t = \{R_{c,n}^t\}_{n\in\mathcal{N}}$ is not pseudorandom. □

Theorem 3.2. *The ensemble* $R_c = \{R_{c,n}\}_{n\in\mathcal{N}}$ *is not pseudorandom.*

Proof. The proof is similar to the proof of Theorem 3.1. □

4. Unpredictability

In this section, we will show that c-random strings, c-pseudorandom strings, and pseudorandom strings are uniformly unpredictable. In contrast with pseudorandom strings, complexity-theoretic pseudorandom strings are not polynomial-time unpredictable.

4.1. Uniform unpredictability

One of the fundamental properties of random strings is the unpredictability of the ith bit from the first $i-1$ bits of the sequence (see [17]). A weaker property has been discussed in [4]: strings in RAND_c are normal.

Definition 4.1. Let $p(\cdot)$ be a given polynomial. An ensemble $X = \{X_n\}_{n\in\mathcal{N}}$ is called *uniformly unpredictable in polynomial-time* if for every polynomial-time algorithm

$D:\{0,1\}^*\to\{0,1\}$, there is a constant n_0 such that for all $n\geqslant n_0$, a string $x\in X_n$ satisfies the following condition (1) with a probability of at least $1-(1/p(n))$:

$$\left|\frac{\|\{i<n\colon D(x[1..i-1])=x[i]\}\|}{n}-\frac{1}{2}\right|<\sqrt{\frac{\log n\log\log n}{n}}. \tag{1}$$

Note that, due to the law of the iterated logarithm, in (1) the bound $\sqrt{\log n\log\log n/n}$ cannot be strengthened to $1/p(n)$, for some polynomial $p(\cdot)$. In [17] it is shown that the law of the iterated logarithm holds for infinite pseudorandom sequences (note that our results in this paper do not apply to infinite pseudorandom sequences).

Now we show that both types of pseudorandom ensembles are uniformly unpredictable in polynomial-time. For the proof we need Chernoff's Bound.

Chernoff's bound (see, e.g., Feller [7]). Let $X_1,X_2,\ldots,X_n$ be independent 0–1 random variables so that $Prob(X_i=1)=\frac{1}{2}$, for each i. Then, for all $0<\delta<\frac{1}{4}$, the following condition holds:

$$Prob\left(\left|\frac{\sum_{i=1}^{n}X_i}{n}-\frac{1}{2}\right|\geqslant\delta\right)<2\mathrm{e}^{-2n\delta^2}. \tag{2}$$

Corollary 4.2. *For each n and* $0<\delta<\frac{1}{4}$, *we have*

$$\left\|\left\{x\in\{0,1\}^n\colon\left|\frac{\sum_{i=1}^{n}x[i]}{n}-\frac{1}{2}\right|\geqslant\delta\right\}\right\|<2^{n+1}\mathrm{e}^{-2n\delta^2}.$$

Proof. It follows from Chernoff's bound (2). □

Lemma 4.3. *Let* $U=\{U_n\}_{n\in\mathcal{N}}$ *be the uniform ensemble*, $D:\{0,1\}^*\to\{0,1\}$ *be a polynomial-time algorithm, and* $\{A_n^D\}_{n\in\mathcal{N}}$ *be a sequence of sets of strings defined as follows*:

$$A_n^D=\{x\in\{0,1\}^n\colon (1)\ \textit{does not hold for } x\}. \tag{3}$$

Then $A^D=\bigcup_{n=1}^{\infty}A_n^D$ *is a polynomial-time computable set and*

$$\|A_n^D\|\leqslant 2^{n+1-2\log e\log n\log\log n}$$

for sufficiently large n.

Proof. It is straightforward to check that A^D is polynomial-time computable. Define an injective function F from strings to strings by

$$F(x)=(D(\lambda)\oplus x[1])(D(x[1..1])\oplus x[2])\ldots(D(x[1..n-1])\oplus x[n])$$

for each $x\in\{0,1\}^n$, where λ is the empty string. Let $B_n=\{F(x)\colon x\in A_n^D\}$. Then it is straightforward that for each $x\in B_n$, we have

$$\left|\frac{\sum_{i=1}^{n}x[i]}{n}-\frac{1}{2}\right|\geqslant\sqrt{\frac{\log n\log\log n}{n}}.$$

Now let $\delta = \sqrt{\log n \log\log n/n}$. Then, by Corollary 4.2, we derive the following bound for the cardinality of A_n^D:

$$\begin{aligned}\|A_n^D\| = \|B_n\| &\leqslant 2^{n+1}\mathrm{e}^{-2n\log n\log\log n/n}\\ &= 2^{n+1}\mathrm{e}^{-2\log n\log\log n}\\ &= 2^{n+1-2\log e\log n\log\log n}. \quad \square\end{aligned}$$

Theorem 4.4. *The ensemble $R_c^t = \{R_{c,n}^t\}_{n\in\mathcal{N}}$ is uniformly unpredictable in polynomial-time, where $t(n) \geqslant 2^{2n}$ is some time-constructible function.*

Proof. Let $D : \{0,1\}^* \to \{0,1\}$ be a polynomial-time algorithm, and $\{A_n^D\}_{n\in\mathcal{N}}$ as in Lemma 4.3. Since any member x of the set A_n^D can be calculated uniquely in time 2^{2n} if we are given the polynomial-time algorithm D and the position of x in A_n^D expressed as an $n - [2\log e \log n \log\log n]$ bit string, it follows that $A_n^D \cap \mathrm{RAND}_{c,n}^t = \emptyset$, for sufficiently large n. This means that the ensemble $R_c^t = \{R_{c,n}^t\}_{n\in\mathcal{N}}$ is uniformly unpredictable in polynomial-time. □

Theorem 4.5. *The ensemble $R_c = \{R_{c,n}\}_{n\in\mathcal{N}}$ is uniformly unpredictable in polynomial-time.*

Proof. The proof is similar to the proof of Theorem 4.4. □

Theorem 4.6. *Every pseudorandom ensemble $X = \{X_n\}_{n\in\mathcal{N}}$ is uniformly unpredictable in polynomial-time.*

Proof. For the sake of contradiction, we assume that X is not uniformly unpredictable in polynomial-time. That is, there is a polynomial-time algorithm $D : \{0,1\}^* \to \{0,1\}$ and a polynomial $p_0(\cdot)$ such that the following condition holds for infinitely many n:

$$\sum_{x\in A_n^D} Prob(X_n = x) > \frac{1}{p_0(n)}, \tag{4}$$

where $\{A_n^D\}_{n\in\mathcal{N}}$ is defined in Lemma 4.3. Now we define a polynomial-time computable function D' by letting

$$D'(x) = \begin{cases} 1 & x \in A_n^D \text{ for some } n \in \mathcal{N}, \\ 0 & \text{otherwise.}\end{cases}$$

By virtue of the definition of D', we have the following equality:

$$\begin{aligned}&Prob(D'(X_n) = 1) - Prob(D'(U_n) = 1)\\ &\quad = \sum_{x\in A_n^D} Prob(X_n = x) - \sum_{x\in A_n^D} Prob(U_n = x).\end{aligned}$$

Hence, by Lemma 4.3 and (4), the following inequality holds for sufficiently large n:

$$\begin{aligned}|Prob(D'(X_n)=1)-Prob(D'(U_n)=1)| &\geqslant \frac{1}{p_0(n)}-\mathrm{e}^{-2\log n\log\log n}\\ &=\frac{1}{p_0(n)}-\frac{1}{n^{2\log e\log\log n}}\\ &\geqslant \frac{1}{2p_0(n)}.\end{aligned}$$

This contradicts the fact that X and U are indistinguishable in polynomial-time. □

Corollary 4.7. *The uniform ensemble* $U=\{U_n\}_{n\in\mathcal{N}}$ *is uniformly unpredictable in polynomial-time.*

Proof. This follows from Theorem 4.6. □

Since the ensemble $R_c=\{R_{c,n}\}_{n\in\mathcal{N}}$ is uniformly unpredictable in polynomial-time (cf. Theorem 4.5) but not pseudorandom (cf. Theorem 3.2), the converse of Theorem 4.6 is not true.

Corollary 4.8. *Let G be a pseudorandom generator. Then the ensemble* $\{G(U_n)\}_{n\in\mathcal{N}}$ *is uniformly unpredictable in polynomial-time.*

Proof. This follows from Theorem 4.6. □

Theorem 4.8 shows that given a pseudorandom generator G, and a truly random input x, the output $G(x)$ is unpredictable in polynomial-time with high probability, though $G(x)$ is not c-pseudorandom.

4.2. Cryptographic unpredictability

Definition 4.9. (Yao [18]). An ensemble $X=\{X_n\}_{n\in\mathcal{N}}$ is called *unpredictable in polynomial-time* if for every probabilistic polynomial-time algorithm D, every polynomial $p(\cdot)$, and all sufficiently large n, the following condition is satisfied:

$$Prob(D(X_n)=next_D(X_n))\leqslant\frac{1}{2}+\frac{1}{p(n)},$$

where $next_D(x)$ returns the $(i+1)$th bit of x if D on input x reads only $i<|x|$ bits of x, and returns a uniformly chosen bit otherwise (i.e., in case D reads the entire string x).

Theorem 4.10 (Yao [18] and Blum and Micali [3]). *An ensemble* $X=\{X_n\}_{n\in\mathcal{N}}$ *is pseudorandom if and only if it is unpredictable in polynomial-time.*

Corollary 4.11. *Neither the ensemble* $R_c=\{R_{c,n}\}_{n\in\mathcal{N}}$ *nor the ensemble* $R_c^t=\{R_{c,n}^t\}_{n\in\mathcal{N}}$ *is unpredictable in polynomial-time.*

Proof. This follows from Theorems 3.1, 3.2, and 4.10. □

5. Strong unpredictability

In view of Definition 4.1, an ensemble $X=\{X_n\}_{n\in\mathcal{N}}$ is uniformly unpredictable in polynomial-time if for every polynomial-time algorithm $D:\{0,1\}^*\rightarrow\{0,1\}$ and sufficiently large n, a string $x\in X_n$ satisfies (1) with a probability of at least $1-(1/p(n))$. If we replace the probability $1-(1/p(n))$ with 1, then we obtain a stronger definition.

Definition 5.1. An ensemble $X=\{X_n\}_{n\in\mathcal{N}}$ is called *strongly unpredictable in polynomial-time* if for every polynomial-time algorithm $D:\{0,1\}^*\rightarrow\{0,1\}$, there is a constant n_0 such that for all $n\geqslant n_0$ and all strings x such that $Prob(X_n=x)>0$, condition (1) holds.

The proof of Theorem 4.4 shows that the ensemble $R_c^t=\{R_{c,n}^t\}_{n\in\mathcal{N}}$ is strongly unpredictable in polynomial-time. However, pseudorandom ensembles are not necessarily strongly unpredictable in polynomial-time. For example, the uniform ensemble $U=\{U_n\}_{n\in\mathcal{N}}$ is not strongly unpredictable in polynomial-time. As another example, we show that the ensemble $\{G(U_n)\}_{n\in\mathcal{N}}$ is not strongly unpredictable in polynomial-time where G is the BBS pseudorandom generator in Example 1: it is clear that $G(0)=0\ldots0$. Thus, $\{G(U_n)\}_{n\in\mathcal{N}}$ is not strongly unpredictable in polynomial-time.

After the above discussion, one may wonder whether there exists an ensemble which is both pseudorandom and strongly unpredictable in polynomial-time. The following theorem gives an affirmative answer.

Theorem 5.2. *Let $D_1,D_2,\ldots$ be a uniform enumeration* (*that is, $D_i(x)$ is computable in time $2^{|x|+i}$*) *of all polynomial-time algorithms, and $A_n^{D_i}$ be defined in Lemma* 4.3. *Then, the ensemble $X=\{X_n\}_{n\in\mathcal{N}}$ is both pseudorandom and strongly unpredictable in polynomial-time, where X_n is a random variable uniformly distributed over $\{0,1\}^n\setminus(\bigcup_{i=1}^{\lfloor\log\log n\rfloor}A_n^{D_i})$.*

Proof. By the definitions of X and $A_n^{D_i}$, it is straightforward that $X=\{X_n\}_{n\in\mathcal{N}}$ is strongly unpredictable in polynomial-time. By Lemma 4.3,

$$\begin{aligned}\sum_{i=1}^{\lceil\log\log n\rceil}\|A_n^{D_i}\| &\leqslant \sum_{i=1}^{\lceil\log\log n\rceil}2^{n+1-2\log e\log n\log\log n}\\ &\leqslant 2^{n+1-2\log e\log n\log\log n}\cdot(\log\log n)\\ &\leqslant 2^{n+1}\frac{\log\log n}{2^{2\log e\log n\log\log n}}.\end{aligned}$$

Since $\log\log n/2^{2\log e\log n\log\log n}$ is negligible, the theorem is proved. □

Table 1
A comparison of two approaches to pseudorandomness

	Complexity approach (R_c and R_c^t)	Cryptographic approach ($U, \{G(U_n)\}_{n\in\mathcal{N}}$, etc.)
Indistinguishable from $\{U_n\}_{n\in\mathcal{N}}$?	No	Yes
Uniformly unpredictable?	Yes	Yes
Cryptographically unpredictable	No	Yes
Strongly unpredictable?	Yes	Yes or No

However, the following question remains open.

Question 1: For a pseudorandom generator G, is the ensemble $\{G(R_{c,n})\}_{n\in\mathcal{N}}$ strongly unpredictable in polynomial-time?

If the answer to the above question is positive, then we get a characterization of pseudorandom generators. That is, for a pseudorandom generator G and a truly random input $x \in \mathrm{RAND}_c$, the output $G(x)$ satisfies condition (1). This coincides with our intuition that the ith bit of a pseudorandom string should not be predictable from its first $i-1$ bits. However, the answer to Question 1 may be negative; in this case, we suggest the following alternative definitions for pseudorandom generators.

Definition 5.3 (*Suggested new Definition* 1). A *pseudorandom generator* is a deterministic polynomial-time algorithm G satisfying the following three conditions:

1. There exists a function $l : \mathcal{N} \to \mathcal{N}$ so that $l(n) > n$ for all $n \in \mathcal{N}$, and $|G(s)| = l(|s|)$ for all $s \in \{0,1\}^*$.
2. The ensemble $\{G(U_n)\}_{n\in\mathcal{N}}$ is pseudorandom.
3. The ensemble $\{G(R_{c,n})\}_{n\in\mathcal{N}}$ is strongly unpredictable in polynomial-time.

Definition 5.4 (*Suggested new Definition* 2). A *pseudorandom generator* is a deterministic polynomial-time algorithm G satisfying the following two conditions:

1. There exists a function $l : \mathcal{N} \to \mathcal{N}$ so that $l(n) > n$ for all $n \in \mathcal{N}$, and $|G(s)| = l(|s|)$ for all $s \in \{0,1\}^*$.
2. The ensemble $\{G(R_{c,n})\}_{n\in\mathcal{N}}$ is strongly unpredictable in polynomial-time.

In summary, we list in Table 1 a comparison of the complexity-theoretic pseudorandom ensembles and cryptographic pseodorandom ensembles.

6. Pseudorandomness in practice

In the previous section, we recommended new definitions of pseudorandomness in terms of strong unpredictability and Kolmogorov complexity. In practice, we may not need such a stronger definition. For example, instead of considering the ensemble R_c^t, it is practically sufficient to consider the sequences which withstand the following five basic tests (see, e.g., [15]).

1. Frequency test (mono-bit test). The purpose of this test is to determine whether the numbers of 0's and 1's in a sequence are approximately the same, as would be expected for a random sequence.
2. Serial test (two-bit test). The purpose of this test is to determine whether the numbers of occurrences of each two-bit sequence are approximately the same.
3. Poker test. This is a generalization of the frequency test (see [15] for details).
4. Runs test. The purpose of the runs test is to determine whether the numbers of runs (of either zeros or ones) of various lengths in the sequence are as expected for a random sequence.
5. Autocorrelation test. The purpose of this test is to check for correlations between the sequence and (non-cyclic) shifted versions of it.

Following the suggestion in FIPS 140-1 (see [8]), we may call a specific sequence x *FIPS-pseudorandom* if it withstands the following four tests: monobit test, poker test, runs test, and long run test (see [8] for details). Let $FIPS_n$ be the set of all n-length FIPS-pseudorandom sequences. Then, in practice, we may consider the following definition for pseudorandom generators.

Definition 6.1. A *pseudorandom generator* is a deterministic polynomial-time algorithm G satisfying the following three conditions:

1. There exists a function $l:\mathcal{N}\to\mathcal{N}$ so that $l(n)>n$ for all $n\in\mathcal{N}$, and $|G(s)|=l(|s|)$ for all $s\in\{0,1\}^*$.
2. The ensemble $\{G(U_n)\}_{n\in\mathcal{N}}$ is pseudorandom.
3. The ensemble $\{G(FIPS_n)\}_{n\in\mathcal{N}}$ is strongly unpredictable in polynomial-time.

Alternatively, we may also consider the following weaker definition.

Definition 6.2. A *pseudorandom generator* is a deterministic polynomial-time algorithm G satisfying the following conditions:

1. There exists a function $l:\mathcal{N}\to\mathcal{N}$ so that $l(n)>n$ for all $n\in\mathcal{N}$, and $G(FIPS_n)\subseteq FIPS_{l(n)}$ for all $n\in\mathcal{N}$.
2. The ensemble $\{G(U_n)\}_{n\in\mathcal{N}}$ is pseudorandom.

References

[2] L. Blum, M. Blum, M. Shub, A simple unpredictable pseudo-random number generator, SIAM J. Comput. 15 (2) (1986) 364–383.

[3] M. Blum, S. Micali, How to generate cryptographically strong sequences of pseudorandom bits, SIAM J. Comput. 13 (1984) 850–864.

[4] C. Calude, Information and Randomness, An Algorithmic Perspective, Springer, Berlin, 1994.

[5] G.J. Chaitin, On the length of programs for computing finite binary sequences, J. Assoc. Comput. Mach. 13 (1966) 547–569.

[6] G.J. Chaitin, The Limits of Mathematics, Springer, Singapore, 1997.

[7] W. Feller, Introduction to Probability Theory and Its Applications, vol. I, Wiley, Berlin, 1968.

[8] FIPS 140-1, Security requirements for cryptographic modules, Federal Information Processing Standards Publication 140-1, US Department of Commerce/N.I.S.T., National Technical Information Service, Springfield, Virginia, 1994.

[9] O. Goldreich, Foundations of Cryptography (Fragments of a Book), ECCC Monographs, http://www.eccc.uni-trier.de/eccc.

[11] L.A. Hemaspandra, A.L. Selman (Eds.), Complexity Theory Retrospective II, Springer, New York, 1997.

[12] A.N. Kolmogorov, Three approaches to the definition of the concept "quantity of information", Probl. Inform. Transmission 1 (1965) 3–7.

[15] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, New York, 1996.

[16] R.J. Solomonoff, A formal theory of inductive inference, Parts 1 and 2, Inform. Control 7 (1964) 1–22.

[17] Y. Wang, Resource bounded randomness and computational complexity, Theoret. Comput. Sci. 237 (1–2) (2000) 33–55.

[18] A. Yao, Theory and applications of trapdoor functions, Proc. 23rd IEEE Symp. on Foundation of Computer Science, 1982, pp. 80–91.

[19] M. Zimand, Large sets in AC^0 have many strings with low Kolmogorov complexity, Inform. Process. Lett. 62 (1997) 165–170.