

Cryptography & Complexity

Spring 2005 - Home exercise #2

Instructor: [Dr. Erez Petrank](#), T.A.: Benny Appelbaum

To be delivered on Sunday 8/5/2005 in Erez's mailbox.

Exercise 2 on the web: <http://www.cs.technion.ac.il/~erez/courses/cc/ex2.pdf>

Question 1:

An interesting tester for pseudo-randomness of a distribution D of strings is the *next-bit predictor*. The predictor reads bit after bit from a string drawn from D and after reading i bits from the string it halts and outputs a prediction to the $i+1$ st bit. In this question you are asked to prove that there are no successful efficient predictors to an ensemble of distributions if and only if it is pseudo-random. Formally,

Definition: An ensemble $\{X_n\}$ passes the *next bit test* if for all probabilistic polynomial time algorithms, any polynomial $p()$, and all sufficiently large n 's, $\text{Prob}[A(w, 1^n) = \sigma] < 0.5 + 1/p(n)$, where the probability is taken over a uniform choice of x according to X_n , a uniform choice of a prefix $w\sigma$ of x , and the coin tosses of algorithm A . The variable w represents a string and the variable σ represents a single bit.

Part a: Prove that all pseudo-random ensembles (i.e., ensembles that are polynomial-time indistinguishable from the uniform ensemble) pass the next bit test.

Part b: Prove that if an ensemble $\{X_n\}$ passes the next bit test then it is pseudo-random. (Hint: use hybrids.)

Question 2:

- Show that for any function $e()$, $0 \leq e(n) \leq 1$, there exist two sequences of distributions $\{X_n\}$ and $\{Y_n\}$ such that their statistical difference is exactly $e(n)$.
- Show that a pseudo random generator with an expansion of one bit is a one way function.
- Show that if there exists a one-way permutation, then there exist two sequences of distributions $\{X_n\}$ and $\{Y_n\}$ such that their statistical difference is non-negligible, but are computationally indistinguishable.
- Prove that the two definitions given in class for the statistical difference between distributions are equivalent.

Question 3:

Suppose we have a PRG G which is defined only on inputs whose length is a power of 2. Convert it to a generator G' that is defined on all input lengths and is still a PRG.

Question 4:

In class we saw that given a pseudo random generator that expands one bit, it is possible to build a pseudo random generator that expands any polynomial number of bits. The construction was as follows. Given a seed s_0 , we construct s_1, \dots, s_m by $s_i = n$ leftmost bits of $G(s_{i-1})$, and the rightmost bit is written at the output, yielding m bits in the output.

What happens to the construction if we also output s_m in the end of the output? Is the modified algorithm still a pseudo-random generator? Or does it lose its pseudo-randomness? Prove your answer.