ECS 227 — Modern Cryptography — Fall 96
Problem Set 2

Phillip Rogaway

Out: 22 October 1996. Due: 5 November 1996.

4. Bogus bit commitment. Prove the statement on page 131 of Naor's paper "Bit commitment using pseudo-randomness": *Furthermore, given any pseudo-random generator G, one can construct another pseudo-random generator G' that has such pairs.*

5. Extending the output length of PRGs. Suppose you are given a pseudorandom generator G which stretches a k bit seed to a $2k$ bit pseudorandom sequence. We would like to stretch even more. We want to construct a pseudorandom generator G' which stretches a k bit seed to a $3k$ bit pseudorandom sequence.

Let $G_1(s)$ denote the first k bits of the string $G(s)$ and let $G_2(s)$ denote the last k bits of the string $G(s)$ for any k-bit string s. (That is, $G(s) = G_1(s)\,,\, G_2(s)$.) We consider two constructions:

(a) $G'(s) = G_1(s)\,,\, G(G_1(s))$ for any k-bit string s.

(b) $G'(s) = G_1(s)\,,\, G(G_2(s))$ for any k-bit string s.

For each construction, say whether it works of not, and justify your answer. If the answer is no, provide a simple statistical test that distinguishes the output of G' from a random $3k$ bit string. If the answer is yes, prove it, and compute the security of G' as a function of that of G. Namely, assuming G was (t, ϵ)-secure, provide t', ϵ', functions of t, ϵ, k, so that G' is (t', ϵ')-secure.

6. Universality of the next-bit test. Consider the following simple type of statistical test that an adversary may perform on a pseudorandom generator G: the adversary inspects some number ℓ of bits output by G; then the adversary predicts the $(\ell + 1)$-st bit; and then the adversary looks at the $(\ell + 1)$-st bit and checks whether or not she had guessed right.

(a) Formalize the above *next-bit test* by defining what it means for an adversary A to (t, ℓ, ϵ)-win a next-bit test. Say what it means for a generator G to be (t, ℓ, ϵ)-secure with respect to next-bit tests.

(b) Carefully state and prove a theorem which says, in effect, that if a generator passes all next-bit tests, then the generator is pseudorandom (ie., its output is indistinguishable from random strings).

Appendix. A pseudorandom generator (PRG) is a function G which takes a k-bit *seed* s and expands it to an $n > k$ bit output $R = G(s)$. We call k the seed length and n the output length. The property it has is that R should be "indistinguishable" from a random string of the same length.

A statistical test for G is an algorithm T taking an n bit input and outputting a bit. We define

$$
\begin{aligned}
P_T^0 &= \Pr[R \stackrel{R}{\leftarrow} \{0,1\}^n : T(R) = 1] \\
P_T^1 &= \Pr[s \stackrel{R}{\leftarrow} \{0,1\}^k;\ R \leftarrow G(s) : T(R) = 1]
\end{aligned}
$$

The advantage of T is $\mathrm{Adv}_T = |P_T^0 - P_T^1|$. We say that T (t, ϵ) breaks G if T runs in time at most t and achieves advantage at least ϵ. We say that G is (t, ϵ)-secure if $\mathrm{Adv}_T \leq \epsilon$ for any statistical test T that runs in time at most t.

Useful hint. You'll want a "hybrid argument" for both (5) and (6).