ICS 180: Introduction to Cryptography 5/13/2004

Homework 4

Due **Tuesday**, *5/20/2004*

1 "One-bit-streching" PRG implies "polynomially-streching" PRG

Assume that G is a PRG which stretches input by only one bit, i.e. for all inputs x, the length $|G(x)|$, of the output of G on x is equal to $|x|+1$.

1.1

For *any* polynomial $p(\cdot)$, use the 1-bit stretching PRG G to construct a PRG G' which stretches the (random) k-bit input into a (pseudorandom) output of length $p(k)$. Prove that your construction G' is indeed a PRG if G is a PRG.

Hint(s): First try to construct a two-bit stretching G', i.e. do it for $p(k) = k+2$. (Note that in the subsection below you have some *wrong* ways of making the 2-bit stretching PRG. I think that all ways where you try to use G just once will fail, and to get $(2+k)$-bit output you need to use G twice.) If you do get it for 2-bit stretching PRG, chances are that your construction generalizes to any polynomial number of extra bits, and that you can prove this generalized construction using the proof you did for the 2-bit case and induction.

And how can you prove that your construction for G' is secure? You can try to prove this by contradiction, i.e. assume that G' is not a PRG, i.e. that there exists a PPT adversary which distinguishes outputs of G' from random strings, and try to use that adversary to attack the PRG G itself, which is supposed to be secure.

You might also try a direct proof (this could in fact be easier!) to argue why the distribution $\{G'(x)\}_{x\leftarrow\{0,1\}^k}$ is computationally indistinguishable from distribution $\{r\}_{r\leftarrow\{0,1\}^{k+2}}$. Recall that the fact that G is a good (1-bit stretching) PRG can be phrased as

$$\{G(x)\}_{x\leftarrow\{0,1\}^k} \approx \{r\}_{r\leftarrow\{0,1\}^{k+1}}$$

(where "$\approx$" stands for "computationally indistinguishable").

In coming up with the direct proof, you can use the following two lemmas, which we used recently in lectures:[1]

Lemma 1 *If X, Z are two computationally indstinguishable distributions, i.e. $\{s\}_{s\leftarrow X} \approx \{s\}_{s\leftarrow Z}$, and $f(\cdot)$ is a PPT algorithm, then $\{f(s)\}_{s\leftarrow X} \approx \{f(s)\}_{s\leftarrow Z}$.*

Using a simplified notation: If $\{X\} \approx \{Y\}$ and f is PPT then $\{f(X)\} \approx \{f(Y)\}$.

Lemma 2 (Hybrid Lemma) *If $X_1, ..., X_n$ are distributions s.t. $\{X_i\} \approx \{X_{i+1}\}$ for every $i = 1, ..., n-1$, and n is* polynomial in the security parameter*, then $\{X_1\} \approx \{X_n\}$.*

[1] For an example how to use them, recall the proof that a PRG G with a very long stretch implies a stream cipher.

1.2

Here are some *incorrect* constructions of a 2-bit stretching PRG G' from 1-bit stretchning PRG G. They all fail in the sense that the resulting algorithm G' could fail to be a secure PRG even if G is a secure PRG. Try to show why that's the case:

- $G'(x) = G(0x)$
- $G'(x) = [G(x)|b_{\oplus}(x)]$ where $b_{\oplus}(x)$ denotes an exclusive or of all bits of x

Hint: In each case, try to design algorithm G s.t. G is a good PRG but it is designed on purpose so that the G' construction which uses G, fails to be a PRG. In other words, design a PRG G which makes outputs of G' easily distinguishable from random strings. How can you make such a G? The easiest way is to take yet another PRG, say $\bar{G}$, and construct G from $\bar{G}$ in such a way that: (1) G is a PRG if $\bar{G}$ is, and (2) G is designed so that to make the G' construction easily breakable (i.e. makes G' distinguishable from random). To construct one PRG from another you might resort to the same two lemmas above.

2 PRGs imply OWFs

We showed in class that existance of a One-Way *Permutation* implies existance of a PRG (pseudorandom generator).

2.1

Show that existense of a PRG implies the existence of a a One-Way *Function*. In fact, simply show that a PRG G which is stretching its inputs by some polynomial $p(\tau)$ amount, for large-enough $p(\tau)$, must be a one-way function.

Hint: If you are lost, compare this to the previous homework, where we showed that any encryption implies one-way functions.

2.2

Show that the converse is not true, i.e. show there a one-way function which is not a PRG. (You should assume here that *some* one-way functions, like modular exponentiation or RSA, do exist.)

Clarification: One very simple (and good!) answer is that a one-way function does not have to stretch the inputs, i.e. if f is OWF we might have $|f(x)| \leq |x|$ for all x, and hence immediately it is not a good PRG. In fact, for all one-way *permutations* (which are also one way functions), $|f(x)| = |x|$ for all x, and this is the case for the two one-way functions we most often talk about, the modular exponentiation and the RSA permutation. However, show that f can be both one-way *and* stretching, and still not a PRG.