ICS 180: Introduction to Cryptography 5/27/2004

Homework 5

Due **Thursday**, *6/03/2004*

1 Constructing a PRG from a PRF

This question is designed so that you see a relation between a PRF and a PRG. You have seen in class that with some work one can build a PRF out of any PRG. But PRF does seem like a more powerful construct, so the other direction, construction of a PRG from a PRF should be easy. But how shall this be done exactly?

Let$\{f_s \mid s \in \{0,1\}^\tau\}_{\tau=1,2,...}$ be a PRF family, where for each τ and each $s \in \{0,1\}^\tau$, function f_s maps domain $\{0,1\}^\tau$ onto the same range $\{0,1\}^\tau$. (Using the notation from the lecture and the notes, we'd say that $l(\tau) = L(\tau) = \tau$.)

Consider the following attempts to construct a PRG from this PRF family. For each of the attempts, either prove that the PRG is secure or prove that it is not, by showing an efficient algorithm that distinguishes its outputs from random strings:

1. $G_1(x) = [f_x(0^\tau) \mid f_x(1^\tau)]$ for $x \in \{0,1\}^\tau$
2. $G_2(x) = [f_{0^\tau}(x) \mid f_{1^\tau}(x)]$ for $x \in \{0,1\}^\tau$

Note that both constructions, on purpose, are done in a way so that the G_i's are trivially stretching: $|G_i(x)| = 2|x|$ for both $i = 1, 2$.

Hint: First, recall what a (secure) PRG is and what a (secure) PRF is. If you want to prove that a PRG construction is *secure*, use one of the two security arguments we have had. Namely, either prove that some two required probability distribution are indistinguishable directly by a series of transformations (for example as in the solutions to problem (1.1) in homework 4). Or, prove it by contradiction, i.e. assume that there exists a PPT adversary A that breaks the PRG security property for the construction G_1 or G_2, and use that adversary to create a PPT attack A' that breaks the PRF security property for the function family $\{f_s\}$.

If you want to show that the PRG construction is *insecure*, you can do so similarly as in the problem (1.2) in homework 4, i.e. by showing that for *some* PRF family $\{f_s\}$, the family itself is a secure PRF family, but the G_i construction (for i either 1 or 2) fails to produce a pseudorandom number generator. How can you do this? Recall the method we used in problem (1.2) of homework 4 and apply it in this case. Namely, try to *create* function family $\{f_s\}$ from any PRF family $\{\bar{f}_s\}$ s.t. $\{f_s\}$ remains a PRF family, but it makes the G_i construction fail as a PRG.

2 Extending the range of a PRF

Let $\{f_s\}$ be a PRF family as above. Below there are several attempts to make another PRF family $\{f'_s\}$, using the existing PRF family $\{f_s\}$, s.t. for each s, $f'_s : \{0,1\}^\tau \to \{0,1\}^{2\tau}$.

In other words, the outputs of f' are twice longer than the outputs of the existing PRF family.[1]

In each case either prove that the result is also a secure PRF or show a PPT algorithm which breaks it, i.e. which distinguishes between conversations with f' and conversations with a truly random function.

1. $f'_s(x) = [f_s(0^\tau) \mid f_s(x)]$
2. $f'_s(x) = [f_{0^\tau}(x) \mid f_s(x)]$
3. $f'_s(x) = [f_s(x) \mid f_s(\bar{x})]$ ($\bar{x}$ is a bitwise negation of x)
4. $f'_s(x) = [f_s(0x) \mid f_s(1x)]$

Hint: The methodology you can use here differs slightly from the one used in question (1): Namely, if a construction fails then you can actually show an attack algorithm A which distinguishes conversations with f'_s from conversations with a truly random function $R : \{0,1\}^\tau \to \{0,1\}^{2\tau}$, such that this attack will work regardless of what the underlying PRF family $\{f_s\}$ is. In other words, I don't think you'll need to show a special PRF family $\{f_s\}$ which fails the f'_s construction: The wrong construction can be broken for *every* PRF family $msetf_s$. However, if you think that you one of the constructions is secure, you should use one fo the standard techniques to show that, as in question (1).

[1]This type of question is a common crypto issue which we have seen with regards to pseudorandom generators [PRG] and to encryption schemes [EΣ]. The question is: Given an algorithm which provides a EΣ/PRG/PRF functionality to some degree, e.g. a PRG which stretches its inputs by just one bit, or an EΣ which encrypts only one-bit long messages, or as we have here a PRF which maps $\{0,1\}^\tau$ onto $\{0,1\}^\tau$), can you use this algorithm as a black box to provide the PRG/EΣ/PRF functionality with better parameters, i.e. a PRG which stretches its inputs by any polynomial number of bits, or an EΣ which encrypts messages of any (polynomial) length, or as here, a PRF which produces twice longer outputs.