236508 - Cryptography & Complexity

Home exercise #1 (One-Way Functions)

Instructor: Dr. Erez Petrank, T.A.: Benny Applebaum,

To be delivered by at 14:00 on Monday 11/4/04 in Erez's mailbox.

(Exercise 1 on the web: http://www.cs.technion.ac.il/~erez/courses/cc/ex1.pdf)

Question 1:

During the lecture, we constructed a strong one-way function g from a weak one-way function f. The goal of this question is to make sure that you understand the proof. In the proof, we used a procedure C in which the input y was used in each possible entry of g among entries $1,2,...,t(n)$.

1. Where in the proof did we use the fact that we try all indices?
2. We would like to show that the proof is not valid if we only try to substitute y in the first entry only and not in the rest. An implicit lemma in the proof asserts that if Algorithm B inverts g with probability $1/q(n^2p(n))$ on inputs of length $n^2p(n)$ then Algorithm A inverts f with probability greater than $1-1/p(n)$. Show that for Algorithm A which uses a modified procedure C' that only tries to substitute y in the first entry, this lemma does not hold .
   Hint: build a function f (not necessarily one way) and an algorithm B that inverts g with probability at least $1/q(n^2p(n))$ whereas Algorithm A, using C', inverts f with probability less than $1-1/p(n)$.
   (The polynomial $p(n)$ will probably be meaningless in the construction. Set the parameters so that for any non-constant $p(n)$ and sufficiently large n, the statement holds.)

Question 2:

Let f be a strong one-way function. Show that for any probabilistic polynomial time algorithm A and for any polynomial $p(n)$, the set:
$S(A,p,n) = \{ x \in \{0,1\}^n : \text{Prob}\,[\, A(f(x)) \in f^{-1}(f(x))\,] \geq 1/p(|x|)\,\}$
has negligible density in $\{0,1\}^n$. Namely, for any polynomial q(n) and for sufficiently large n it holds that $|S(A,p,n)| < 2^n/q(n)$.

Question 3:

Suppose f and g are one-way functions. Are the following functions necessarily one-way? Prove or give a counter example.

- $H(x) = g(f(x))$
- $H(x) = f(x) \bullet g(x)$, where "•" means concatenation.
- $H(x) = f(x) \oplus g(x)$, where $\oplus$ means bitwise exclusive or.
- $H(x) = f(x_1) \oplus g(x_2)$, where $\oplus$ means bitwise exclusive or, x_1 is the left half of the bits in x, and x_2 is the right half.
- $H(x) = f(x_1) \bullet g(x_2)$, where x_1 is the left half of the bits in x, and x_2 is the right half.
- $H(x) = f(x_1 \oplus x_2) \oplus g(x_2)$, where x_1 is the left half of the bits in x, and x_2 is the right half.