CS355: Topics in cryptography **Spring 2000**

Assignment #1

Due: Wednesday, May 10th, 2000.

Problem 1: **a.** Let $f : \{0,1\}^n \to \{0,1\}^m$ be an efficiently computable one-to-one function. Show that if f has a (t, ϵ) hard core bit then f is $(t, 2\epsilon)$ one-way.

b. Show that if $G : \{0,1\}^n \to \{0,1\}^{2n}$ is a (t, ϵ) PRNG then G is also (t', ϵ') one-way for some (t', ϵ') close to (t, ϵ). Give the best bounds you can.

c. Show that if $F : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ is a (t, ϵ, q) PRF then

$$G(s) = F(1, s) \| F(2, s) \| \cdots \| F(q, s)$$

is a $(t - q, \epsilon)$ PRNG. We are assuming that evaluating F takes unit time.

Problem 2: Let p be a prime and let $g \in \mathbb{Z}_p^*$ generate a subgroup of order q for some $q = 3 \bmod 4$, where q is public. Define $\mathrm{lsb}_2(x) = 0$ if $x \bmod 4$ is 0 or 1 and $\mathrm{lsb}_2(x) = 1$ otherwise. Show that if $\mathrm{lsb}(x)$ is a (t, ϵ) hard core bit of $f(x) = g^x \bmod p$ then so is $\mathrm{lsb}_2(x)$. Note that q may not be prime.

Problem 3: In this problem we develop a simple version of the Goldreich-Levin algorithm. Suppose $\alpha \in \{0,1\}^n$ and $f_\alpha : \{0,1\}^n \to \{0,1\}$ is an oracle satisfying

$$\Pr_x[f_\alpha(x) = \langle x, \alpha \rangle] > \frac{3}{4} + \epsilon$$

Show that α can be recovered from the oracle f with probability 1/2 by making $\tilde{O}(n/\epsilon)$ oracle queries.

Hint: Show that the first bit of α can be found by querying f_α at many pairs of points $(r_1 r_2 \ldots r_n,\ \bar{r}_1 r_2 \ldots r_n)$. Generalize to show that all bits of α can be found. Use the Chernoff bound to bound the success probability of your algorithm.

Remark: This approach can be extended to reduce the $\frac{3}{4} + \epsilon$ bound to $\frac{1}{2} + \epsilon$. The extension is based on making the query points pair wise independent rather than completely independent.

Problem 4: In this problem we study an alternate definition for PRF's that is easier to work with and was briefly discussed in class. Our goal is to show that this alternate definition is equivalent to the original one. Let $f : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^m$. Intuitively, the new definition says that f is a PRF if no t-time algorithm can distinguish the pair $(x, f_k(x))$ from the pair (x, R) after querying $f_k()$ at inputs of its choice. Here x is chosen by the algorithm and R is a random m-bit string.

More precisely, we say that f is a (t, ϵ, q) PRF according to the new definition if there is no pair of t-time algorithms (A, B) that satisfy the following criteria:

a. Algorithm A is an oracle algorithm. It queries an oracle $f_k(x) = f(x, k)$ in at most q points and outputs a pair (x, state). Here x is an n-bit string. state is a string in $\{0,1\}^*$ and is used for communicating with algorithm B.

b. A challenger picks a random bit $b \in \{0,1\}$ and does the following: if $b = 0$ it sets $y = f_k(x)$. If $b = 1$ it sets y to be a random m bit string R. Using C notation we write $y = b\,?R : f_k(x)$.

c. Algorithm B takes as input (x, y, state) and outputs a bit $b' \in \{0,1\}$. The function f is not a (t, ϵ, q) PRF if $Pr[b = b'] > \frac{1}{2} + \epsilon$. In other words, algorithm B is able to tell whether it received $f_k(x)$ or a random string.

Formally, one says that f is a (t, ϵ, q) PRF if no pair of t-time algorithms (A, B) satisfies:

$$Pr\left[b = b' \;\middle|\; \begin{array}{l} k \leftarrow \{0,1\}^k, \;\; (x, s) \leftarrow A^{f_k}(), \;\; b \leftarrow \{0,1\}, \\ R \leftarrow \{0,1\}^m, \;\; y = b\,?R : f_k(x), \;\; b' = B(x, y, s) \end{array} \right] > \frac{1}{2} + \epsilon$$

Show that if f is a $(t, \epsilon/2q, q)$-PRF according to the new definition then it is a (t, ϵ, q)-PRF according to the original definition.

Problem 5: Let $\pi : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ be a (t, ϵ, q) PRP. Given k, both $\pi_k(x)$ and $\pi_k^{-1}(x)$ can be efficiently computed. Show how to construct an SPRP out of π. Prove that your construction is a (t', ϵ', q) SPRP. Give the best values of t', ϵ' you can. Your solution suggests a way of converting any block cipher that is resistant to chosen PT attacks into a block cipher that resists both chosen PT and chosen CT attacks.