CS355: Topics in Cryptography **Fall 1998**

Assignment #2

Due: Wednesday, November 25th, 1998.

1. **Alternate proof of Goldreich-Levin:** In this problem we study the first step in an alternate proof to the Goldreich-Levin theorem. 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 easily recovered from the oracle f.

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.

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.

2. **Bad construction for a PRF:** Consider the following construction for a Pseudo Random Function $F_G : \{0,1\}^n \times \{0,1\}^s \to \{0,1\}^n$ based on a Pseudo Random Number Generator $G : \{0,1\}^n \to \{0,1\}^{2n}$:

$$F_G(x, k_1 \ldots k_s) = G_{k_s}(G_{k_{s-1}}(\cdots(G_{k_1}(x))\cdots))$$

where $G(x) = G_0(x) \parallel G_1(x)$ and $G_0(x)$ is the n least significant bits of $G(x)$ and $G_1(x)$ is the n most significant bits of $G(x)$.

Show that $F(x, s)$ may not be pseudo random function no matter how secure the PRNG is.

Hint: First show that a (t, ϵ) PRNG G can be converted into a generator G' for which $G'(0^n) = 0^{2n}$.

3. **Insecure SPRP:** Show that the Luby-Rackoff construction for strong pseudo random permutations is insecure if one uses only three PRF's rather than four.

4. **SSPA of ElGamal encryption:** Let G be a group of prime order q. Recall that an ElGamal public key consists of (g, h) where $g, h \in G$ and $h = g^x$. The private key is x. To encrypt a message $m \in G$ one picks a random $r \in \mathbb{Z}_q$ and computes $C = (g^r, m \cdot h^r)$.

Prove that ElGamal encryption is (t, ϵ) secure against a passive adversary assuming (t, ϵ)–DDH. To do so you must show how an algorithm A able to mount a passive attack on ElGamal encryption can be used to solve the DDH problem.

Extra credit:

1. Show how to construct a PRF out of a PRNG such that the PRF can be evaluated in a logarithmic number of steps on a parallel computer. (Currently the only constructions are based on synthesizers rather than PRNG's).

2. Prove that the DDH assumption is equivalent to some other standard complexity assumption or show that the DDH assumption is false.