CSE 5852: Problem Set 4

Due: October 17, 2016

1 Discrete Logarithm Problem

Recall the discrete logarithm assumption from class.

Assumption 1. *For any PPT $\mathcal{A}$, there exists a negligible ϵ such that for a random n-bit p and its generator and select a random $x \in \mathbb{Z}_p^*$,*

$$\Pr[\mathcal{A}(1^n, p, g, g^x \mod p) = x] \leq \epsilon(n).$$

We will consider the following variations of the discrete logarithm problem.

a) **20 pts** Show the following theorem: Consider the following problem, let c be a n-bit constant, pick a random n-bit p and its generator g and selected a random $x \in \mathbb{Z}_p^*$.

Theorem 1 (c-Discrete Logarithm Problem)**.** *Let c be a n-bit constant. If the discrete logarithm problem is hard then for any PPT $\mathcal{A}$, there exists a negligible ϵ such that for a random n-bit p and its generator g and selected a random $x \in \mathbb{Z}_p^*$.*

$$\Pr[\mathcal{A}(1^n, p, g, g^{x+c} \mod p) = x] \leq \epsilon(n).$$

Hint: You are being asked to show that this problem is hard if the discrete logarithm problem is hard. The easiest way to do this is to show that any attacker that can break the *c-Discrete Logarithm* problem can be used to build an efficient attacker that breaks the standard *Discrete Logarithm* problem. Be precise about what each problem means and what each attacker expects as input. Part of the question is to show that you provide the right type of inputs when you call the *c-Discrete Logarithm* attacker.

b) **15 pts** In class we considered the discrete logarithm problem with modular exponentiation. In this problem, we consider the group $\mathbb{Z}_p$ with addition.

Consider the following problem (it is possible to find a generator of $\mathbb{Z}_p$):

(a) Choose a random n-bit p and its generator
(b) Select a random $x \in \mathbb{Z}_p$,
(c) Given $1^n, p, g, xg \mod p$ find x.

Show a polynomial time algorithm for solving this problem. As a reminder xg is the generator being added to itself x times. This happens to correspond to multiplication $\mod p$.

Hint: Solving this problem is connected to the gcd problem.

2 Pseudorandom Number Generators

Let $G_1, G_2 : \{0,1\}^n \to \{0,1\}^{2n}$ be pseudorandom generators. Answer the following questions. For proofs you must show a reduction. That is, you must show transform a break on the new construction to a break on the original construction. In your proofs you may show the PRG is secure with regards to the *next bit unpredictability* definition or the *statistical tests* definition, whatever suits your needs.

a) **15 pts** Show that for any pseudorandom G_1 the function $G_3(s) = G_1(s) \oplus c$ is a pseudorandom generator for any fixed c.

b) **20 pts** Show that for any G_1 the function $\overline{G_1}$ is also a pseudorandom generator. Here $\overline{G_1}$ is a function that runs G_1 and flips each bit. That is for $G_1(s) = y_1...y_{2n}$, $\overline{G_1}(s) = \overline{y_1}....\overline{y_{2n}}$.

c) **20 pts** Show that for any G_1 the function $reverse(G_1)$ is also a pseudorandom generator. Here $reverse(G_1)$ is a function that runs G_1 and outputs the bits in opposite order. That is for $G_1(s) = y_1...y_{2n}$, $reverse(G_1(s)) = y_{2n}....y_1$.

d) **10 pts** Show that $G_3(s) = G_1(s)||G_2(s)$ is not necessarily a pseudorandom generator. Here $||$ represents concatenation. That is, present a pair of pseudorandom generators G_1, G_2 whose concatenation is not pseudorandom.