CS276: Cryptography Due date: September 19, 2017

Problem Set 1

Instructor: Alessandro Chiesa *GSI: Benjamin Caulfield*

Problem 1

Assume that f is a length preserving one-way function, i.e., for every $x \in \{0,1\}^*$ it holds that $|f(x)| = |x|$. For each of the following functions g, prove that g is a one-way function, or provide a counterexample to demonstrate that it is not.

$$\begin{aligned}
&\textbf{A:} \quad g(x) = f(f(x)) \\
&\textbf{B:} \quad g(x) = f(\bar{x}) \\
&\textbf{C:} \quad g(x) = f(x) \oplus x \\
&\textbf{D:} \quad g(x, y) = f(x \oplus y) \\
&\textbf{E:} \quad g(x) = f(x) \| f(\bar{x})
\end{aligned}$$

(Above $\bar{x}$ denotes the bitwise complement of x and $\|$ denotes concatenation, e.g., $1011\|\overline{1011} = 10110100$.)

Hint: To show g is not a one-way function, construct an f from an arbitrary one-way function h. Then prove that f is one-way but g is not. To show that g is one-way, assume that a PPT-inverter A exists for g and use it to construct a PPT-inverter A' for f.

Problem 2

Prove that if one-way functions exist then $P \neq NP$.

Problem 3

Let p be a prime and let g and h be (not necessarily distinct) generators of $\mathbb{Z}_p^*$. Prove or disprove the following statements:

$$\begin{aligned}
&\textbf{A:} \quad \{x \leftarrow \mathbb{Z}_p^* \,:\, g^x \mod p\} = \{x \leftarrow \mathbb{Z}_p^* \,;\, y \leftarrow \mathbb{Z}_p^* \,:\, g^{xy} \mod p\} \\
&\textbf{B:} \quad \{x \leftarrow \mathbb{Z}_p^* \,:\, g^x \mod p\} = \{x \leftarrow \mathbb{Z}_p^* \,:\, h^x \mod p\} \\
&\textbf{C:} \quad \{x \leftarrow \mathbb{Z}_p^* \,:\, g^x \mod p\} = \{x \leftarrow \mathbb{Z}_p^* \,:\, x^g \mod p\} \\
&\textbf{D:} \quad \{x \leftarrow \mathbb{Z}_p^* \,:\, x^g \mod p\} = \{x \leftarrow \mathbb{Z}_p^* \,:\, x^{gh} \mod p\}
\end{aligned}$$

(Recall that $\{x \leftarrow \mathbb{Z}_p^* : g^x \mod p\}$ is a probability distribution. You are being asked to prove or disprove the statement that two probability distributions are *identical*.)

Problem 4

Suppose that you have a polynomial-time algorithm A that solves the Discrete Logarithm Problem *in a special case*. Namely on inputs p, g, and $g^x \mod p$, the algorithm A outputs x if p is a prime,

g is a generator of $\mathbb{Z}_p^*$ and $g^x \mod p$ is prime.

Show that there exists a *probabilistic* polynomial-time algorithm B that solves any instance of the Discrete Logarithm Problem.

Note: The general instance of the Discrete Logarithm Problem still assumes that g is a generator and p is prime (but not that g^x is prime).

Keep in mind that you are trying to find a PPT solver for the problem, so you only need to solve the problem with *non-negligible* probability. But you do need to succeed with non-negligible probability on *all* inputs. It is not enough to provide an algorithm that will solve the problem on non-negligibly many inputs, assuming the inputs are chosen uniformly at random.

Problem 5

In this problem, we study how to efficiently sample generators modulo a prime.

Let p be a prime. The group $\mathbb{Z}_p^*$ can be shown to be cyclic of order $p-1$; in fact, while proving this, one also obtains the fact that the number of elements of order $p-1$ in $\mathbb{Z}_p^*$ (i.e., the number of generators in $\mathbb{Z}_p^*$) is equal to $\phi(p-1)$. Since $\phi(n) = \Theta(n/\log\log n)$, the quantity $\phi(p-1)/p-1$ is non-negligible. In particular, by choosing an element g of $\mathbb{Z}_p^*$ at random, the probability that g is a generator of $\mathbb{Z}_p^*$ is non-negligible. However, given an element g in $\mathbb{Z}_p^*$, how can we decide if it is a generator or not?

Describe a polynomial-time algorithm that, on input an element $g \in \mathbb{Z}_p^*$, an odd prime p, and the factorization of $p-1$, decides whether g is a generator of $\mathbb{Z}_p^*$.

(Note: Efficiently sampling generators modulo a prime is sometimes needed in practice, such as in Elgamal's public-key cryptosystem. But, how does one obtain the factorization of $p-1$? Usually, one generates the prime p *along with* the factorization of $p-1$. For example, in Elgamal's public-key cryptosystem a prime p is chosen to have the form $p = 2q+1$ for some prime q, so that $p-1 = 2q$; a prime of this form is called a *safe prime*.)

Problem 6

Give a strategy to distinguish between $(g^x, g^y, g^{xy}) \mod p$ and $(g^x, g^y, g^r) \mod p$ with non-negligible advantage, where x, y, r are chosen at random such that $1 \le x, y, r \le p-1$, and g is a generator of $\mathbb{Z}_p^*$.