Homework 2

Homework is due by **11pm of Sep 23**. Send by email to both "regev" (under the cs.nyu.edu domain) and "des480" (under the nyu.edu domain) with subject line "CSCI-GA 3210 Homework 2" and name the attachment "YOUR NAME HW2.tex/pdf". There is no need to print it. Start early!

1. (3 points) *(Weak vs strong one-way functions.*[♣]*)* Recall that we say that $f : \{0,1\}^* \to \{0,1\}^*$ is a one-way function if there is an efficient algorithm for computing it, and moreover, for any PPT algorithm I,

$$\Pr_{x\in\{0,1\}^n}[I(1^n, f(x)) \in f^{-1}(f(x))] \in \text{negl}(n) \;, \tag{0.1}$$

where the 1^n is simply a convenient hack for allowing I to run in time $\text{poly}(n)$ (which would not be the case otherwise when the output of f is short). One can also consider a variant of this definition, known as a *weak* one-way function, saying that there exists a constant $c > 0$ such that for any PPT I, Equation (0.1) holds with $< 1 - n^{-c}$ instead of $\in \text{negl}(n)$. As their names suggest, any (strong) one-way function is also a weak one-way function (make sure you see why). Can you construct an example of a weak one-way function that is not a strong one-way function? (You can assume that strong one-way functions exist) Can you think of a way to create a strong one-way function from a weak one-way function?

2. *(Fun with one-way functions.)*

   (a) (2 points) Assume we modify the definition of a one-way function by allowing the adversary to output a *list* of supposed preimages, and he wins if at least one of them is a valid preimage (and as before the winning probability of any efficient adversary should be negligible). How does this modified definition compare with the original one? Formally prove your answer.

   (b) (2 points) [2] For a security parameter n, define $f : \{2^{n-1}, \dots, 2^n\} \to \{1, \dots, 2^{2n}\}$ by $f(x) = x^2$ (over the integers). Is it a one-way function? (Rabin's function is similar, except it's done in $\mathbb{Z}_N$)

   (c) (4 points) [3] Suppose that $f : \{0,1\}^* \to \{0,1\}^*$ is such that $|f(x)| \le c \log|x|$ for every $x \in \{0,1\}^*$, where $c > 0$ is some fixed constant. (Here $|\cdot|$ denotes the length of a string.) Prove that f is *not* a one-way function.

   (d) (5 points) [2] Assume $g : \{0,1\}^n \to \{0,1\}^n$ is a one-way function. Is the function $f : \{0,1\}^{2n} \to \{0,1\}^{2n}$ defined by $f(x_1, x_2) = (g(x_1), g(x_1 \oplus x_2))$ necessarily also a one-way function?

   (e) (3 points) (bonus[4]) Show that there exists a one-way function $f : \{0,1\}^n \to \{0,1\}^n$ for which the function $f'(x) := f(x) \oplus x$ is *not* one-way. You can assume the existence of a one-way function $g : \{0,1\}^n \to \{0,1\}^n$ for all n. I need a hint for 1/2 points! (ID 82778)

3. (6 points) *(Worst-case to average-case reduction.*[3]*)* Let N be the product of two distinct n-bit primes, and suppose there is an efficient algorithm $\mathcal{A}$ that computes square roots on a noticeable fraction of quadratic residues mod N:

$$\Pr_{y\leftarrow\mathbb{QR}_N^*}[\mathcal{A}(N, y) \in \sqrt{y} \bmod N] = \delta \ge 1/\text{poly}(n).$$

[♣] See the instructions regarding club questions in Homework 1.
[2] A question from Dodis's class
[3] A question from Peikert's class
[4] By Bao Feng, as appears in Goldreich's book

Construct an efficient algorithm $\mathcal{B}$ that, using $\mathcal{A}$ as an oracle, computes the square root of *any* $y \in \mathbb{QR}_N^*$ with *overwhelming* probability (solely over the random coins of $\mathcal{A}$ and $\mathcal{B}$). That is, for every $y \in \mathbb{QR}_N^*$, it should be the case that

$$\Pr[\mathcal{B}^{\mathcal{A}}(N, y) \in \sqrt{y} \bmod N] = 1 - \mathrm{negl}(n).$$

Explain in your own words why such reductions are known as worst-case to average-case reductions.

4. *(PRG)*♣ Try to think how to precisely define the property that a function $f : \{0,1\}^n \to \{0,1\}^{n+1}$ satisfies that $f(U)$ "looks" like a uniform string in $\{0,1\}^{n+1}$ where U is sampled uniformly from $\{0,1\}^n$. Such efficiently computable functions are known as *pseudorandom generators*.