Introduction to Cryptography
Courant, Fall 2016

Homework 2

Instructor: Oded Regev
Student: YOUR NAME

Homework is due by **7am of Sep 26**. Send by email to both "regev" (under the cs.nyu.edu domain) and "avt237@nyu.edu" with subject line "CSCI-GA 3210 Homework 2" and name the attachment "YOUR NAME HW2.tex/pdf", and please also bring a printed copy to class. Start early!

Instructions. Solutions must be typeset in LaTeX (a template for this homework is available on the course web page). Your work will be graded on *correctness*, *clarity*, and *conciseness*. You should only submit work that you believe to be correct; if you cannot solve a problem completely, you will get significantly more partial credit if you clearly identify the gap(s) in your solution. It is good practice to start any long solution with an informal (but accurate) "proof summary" that describes the main idea.

You are expected to read all the hints either before or after submission, but before the next class.

You may collaborate with others on this problem set and consult external sources. However, you must ***write your own solutions***. You must also ***list your collaborators/sources*** for each problem.

1. (3 points) *(Weak vs strong one-way functions.*$^\clubsuit$*)* Recall that we say that $f : \{0,1\}^n \rightarrow \{0,1\}$ 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 if the output of f happens to be 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?

Solution: Note that for any $c > 0$, we have that $1 - \frac{1}{n^{c-1}} \leq 1 - \frac{1}{n^c}, \forall n \geq 1$. This means that given our choice of c, if we can find a function that succeeds with probability $1 - \frac{1}{n^{c-1}}$, then it is a one-way function. Choose $c = 1$, so we are looking for a function whose inversion success probability is $\frac{1}{n}$. In this case there happens to be a function with this success probability: a function that maps all non-prime numbers to themselves, and all prime numbers to the next greatest prime. Since it there is no generally efficient way to compute primality, and even more difficult to find the next largest prime, the best guess would be to always guess the same number that you are given. This choice will never be correct if the number is prime, and always correct if it is non-prime. Since there are $O(\frac{1}{log(N)}) = O(\frac{1}{n})$ fraction of primes less than N, this is a weakly one-way function. However, it is not strongly one-way, as this success probability is clearly not negligible (the limit of success for n large isn't even 0). In order to construct a strong one-way funciton from a weak one, we might consider requiring that the adversary determine if a list of n-bit numbers are prime, with the same mapping on each component as before. If we allow the list to be of length greater than $cnlog(n)$ (where $c > 0$ is from the x^{-c} in the definition of negligible functions). The probability of getting all of them right is then less than $(1 - \frac{1}{n})^{cnlog(n)} \approx n^c, \forall n > 0$. Thus the probability is negligible, and the new function is strong one-way.

$^\clubsuit$Again, this is a question meant to encourage you to think; you are not required to solve it fully, but you are required to demonstrate that you thought about it seriously.

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.

Solution: Let $P(x,n) = \Pr_{x\in\{0,1\}^n}[I(1^n, f(x)) \in f^{-1}(f(x))] \in \mathrm{negl}(n)$, and let l be the length of the list. Then the probability of guessing correctly at least once out of l independent trials (i.e., assuming there is no improvment or deterioration on the guessing for successive guesses in the list) is $1-(1-P(x,n))^l)$. If this function is negligible, this tells us that $\lim_{n\to\infty} 1-(1-P(x,n))^l)n^c = 0, \forall c > 0$. Rearranging the equation and then taking the l-th root (which is preserved under taking the limit, since it is a convex function), we get $\lim_{n\to\infty} P(x,n)c^{\frac{c}{l}} = 0, \forall \frac{c}{l} > 0$. In other words, since all of the steps above were double implications, we can see that the two definitions of one-way functions are completely equivalent.

(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 $\mathbb{Z}_N$)

Solution: No, this is not a one-way function. The inverse function, which is the square root, can be found using a binary search/bisection algorithm . We can always evaluate whether some number n is greater than, less than, or equal to $\sqrt{f(x)}$, so we have all the tools we need to perform a binary search in $O(n)$ time. This is not a negligible function of n, so the function $f(x)$ is not one-way.

(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.

Solution: We assume that $\exists c > 0 | \forall x \in \{0,1\}^* : |f(x)| \le clog(|x|)$. Let $P(x)$ be the probability of guessing an inverse that is in the preimage. Suppose we are given some known f(x) (with x unknown), and then guess an $x^* : clog|x^*| > |f(x^*)|$, but is as small as possible, i.e. $x^* \le x$. Then $|f(x^*)| \le |f(x)|$. Since there are no more than $2^{|f(x)|}$ possible images of x^*, the probability $P[f(x) = f(x^*)] \ge 2^{-|f(x)|}$ Since this condition is the same as saying that your guess was in the preimage, we have that $\exists c > 0 : \forall x, P(x) \ge 2^{-|f(x)|} \ge 2^{-clog|x|} = |x|^{-c}$, where we have again used the fact that $|f(x)| \le clog(|x|)$. Since negligible functions require that $\forall c > 0, \exists x : P(x) < |x|^{-c}$ (in fact the definition requires that this be true $\forall x$ larger than some N_c), our guessing probability is not negligible, and therefore the function is not one-way.

(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?

[2] A question from Dodis's class

[3] A question from Peikert's class

Solution: Yes. The result follows directly from the fact that x_1 and $x_1 \oplus x_2$ are pairwise independent We are trying to show that $P[x_1 \in g^{-1}(g(x_1)) \wedge (x_1 \oplus x_2) \in g^{-1}(g(x_1 \oplus x_2))] \in negl(x)$. Note that because x_1 and $x_1 \oplus x_2$ are pairwise independent, this joint probability is just the product $P[x_1 \in g^{-1}(g(x_1))] * P[(x_1 \oplus x_2) \in g^{-1}(g(x_1 \oplus x_2))]$. Since the product of two negligible functions is also negligible, the function $f(x_1, x_2)$ is one-way.

(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)

Solution: Yes. Choose some a secret key $k \in \{0,1\}^n$, and define the function $f : \{0,1\}^2 n \to \{0,1\}^2 n, f(x,k) = (x \oplus k, 0)$. Since XORs are all pairwise independent, having $x \oplus k$ does not tell us anything about either k or x, and thus the function is one-way. However, $x \oplus f(x,k) = (k,x)$. This tells us the full information about both x and k, and therefore allows us to choose an element in the preimage every time. So the function is not one-way.

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:

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

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

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

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

Solution: Now suppose we have some $c > 0$ as in the definition of negligible functions, and we wish to find an algorithm such that that the failure probability is negligible, i.e. that $\lim_{n \to \infty} \Pr[\mathcal{B}^{\mathcal{A}}(N,y) \in \sqrt{y} \bmod N]x^c = 0, \forall c > 0$. In this case, consider the following strategy: Repeat the algorithm $\mathcal{A}$ a total of $(c+1) * poly(n) * log(n)$ times. After each iteration we can efficiently compute whether our square root was actually valid. So the algorithm is successful if any of the computed roots are correct. The probability that all of the guesses are wrong is $(1 - \frac{1}{poly(n)})^{(c+1)*poly(n)*log(n))} = e^{-(c+1)log(n)} = n^{-(c+1)}$ As we can see, $\lim_{n \to \infty} n^{-(c+1)} n^c = 0$, and therefore the failure probability is negligible. Note that this algorithm is still efficien, because it is repeated a polynomial number of times. The reason why these reduction might be called worst case to average case reductions is that the algorithm went from being correct only on very rare occasions (worst case) to being correct nearly all of the time.

[4] By Bao Feng, as appears in Goldreich's book

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 $\{0,1\}^n$. There is no need to write down your solution: just think about it in preparation for Monday's class. Such efficiently computable functions are known as *pseudorandom generators*.