Exercise 2 – Foundations of Cryptography 89-856 Solutions

July 6, 2008

Exercise 1: Prove that if an efficiently-computable 1–1 function f has a hard-core predicate, then it is one-way. Why is the 1–1 requirement necessary?

Solution 1: Let f be an efficiently-computable 1–1 function and let b be a hard-core predicate of f. Assume by contradiction that there exists a probabilistic polynomial-time adversary $\mathcal{A}$ and a polynomial $p(\cdot)$ such that for infinitely many n's

$$\Pr[\mathcal{A}(f(U_n), 1^n) \in f^{-1}(f(U_n), 1^n)] \geq \frac{1}{p(n)}$$

We now construct an adversary $\mathcal{B}$ that contradicts the assumption that b is a hard-core predicate of f. The adversary $\mathcal{B}$ receives for input some $(y = f(x), 1^n)$, with $x \in_R \{0,1\}^n$, and attempts to guess $b(x)$.[1] In order to do this, $\mathcal{B}$ first invokes $\mathcal{A}$ upon input y; let x' be the output of $\mathcal{A}$. Next, $\mathcal{B}$ checks if $f(x') = y$ (notice that f is efficiently computable, so $\mathcal{B}$ can do this). If yes, then $\mathcal{B}$ outputs $\sigma = b(x')$ and halts (in this case, $\mathcal{B}$ is correct with probability 1). Otherwise, $\mathcal{B}$ outputs a uniformly chosen bit $\sigma \in_R \{0,1\}$. It remains to analyze $\mathcal{B}$'s success:

$$\begin{aligned}
&\Pr[\mathcal{B}(f(U_n)) = b(U_n)] \\
&\quad= \Pr[\mathcal{B}(f(U_n)) = b(U_n) \mid \mathcal{A}(f(U_n)) \in f^{-1}(f(U_n))] \cdot \Pr[\mathcal{A}(f(U_n)) \in f^{-1}(f(U_n))] \\
&\quad + \Pr[\mathcal{B}(f(U_n)) = b(U_n) \mid \mathcal{A}(f(U_n)) \notin f^{-1}(f(U_n))] \cdot \Pr[\mathcal{A}(f(U_n)) \notin f^{-1}(f(U_n))] \\
&\quad= 1 \cdot \Pr[\mathcal{A}(f(U_n)) \in f^{-1}(f(U_n))] + \frac{1}{2} \cdot \Pr[\mathcal{A}(f(U_n)) \notin f^{-1}(f(U_n))] \\
&\quad= 1 \cdot \Pr[\mathcal{A}(f(U_n)) \in f^{-1}(f(U_n))] + \frac{1}{2} \cdot \Big(1 - \Pr[\mathcal{A}(f(U_n)) \in f^{-1}(f(U_n))]\Big) \\
&\quad= \frac{1}{2} + \frac{1}{2} \cdot \Pr[\mathcal{A}(f(U_n)) \in f^{-1}(f(U_n))] \\
&\quad\geq \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{p(n)}
\end{aligned}$$

We therefore have that for infinitely many n's, adversary $\mathcal{B}$ guesses the value of $b(x)$ from $f(x)$, with probability at least $1/2 + 1/2p(n)$, in contradiction to the assumption that b is a hard-core predicate of f.

Note that the 1–1 condition is necessary to ensure that whenever $\mathcal{A}$ succeeds in inverting $f(x)$, it follows that $\mathcal{B}$ succeeds in guessing $b(x)$. For example, consider the function

[1] For simplicity of notation, we omit the input 1^n in the rest of this solution.

$f(x) = 0^{|x|}$ and $b(x) = x_1$ where $x = x_1 \cdots x_n$. Then, clearly b is a hard-core predicate of f. However, f is *not* one-way.

Exercise 2: Let $X = \{X_n\}_{n\in\mathsf{N}}$ and $Y = \{Y_n\}_{n\in\mathsf{N}}$ be computationally indistinguishable probability ensembles.

1. Prove that for any probabilistic polynomial-time algorithm A it holds that $\{A(X_n)\}_{n\in\mathsf{N}}$ and $\{A(Y_n)\}_{n\in\mathsf{N}}$ are computationally indistinguishable.
2. Prove that the above does not hold if A does not run in polynomial-time.

Solution 2:

1. Assume by contradiction that there exists a PPT A, a distinguisher D and a polynomial $p(\cdot)$ such that for infinitely many n's it holds that

$$|\Pr[D(A(X_n), 1^n) = 1] - \Pr[D(A(Y_n), 1^n) = 1]| \geq \frac{1}{p(n)}$$

We construct a distinguisher D' that receives a sample z (from X_n or Y_n), runs A on the input (i.e. z, but not the 1^n part) and then runs D on the result from A. Finally, D' outputs whatever D does. Clearly, for every z, $D'(z, 1^n) = 1$ if and only if $D(A(z), 1^n) = 1$. Therefore,

$$|\Pr[D(X_n, 1^n) = 1] - \Pr[D(Y_n, 1^n) = 1]| \geq \frac{1}{p(n)}$$

in contradiction.

2. For any A and $Z = \{Z_n\}_{n\in\mathsf{N}}$ denote $A(Z) = \{A(Z_n)\}_{n\in\mathsf{N}}$. Now, if X and Y are identically distributed (or even statistically close), then $A(X)$ and $A(Y)$ will remain identically distributed (or statistically close). Therefore, we will show that for any X and Y that are *not* statistically close (but may be computationally indistinguishable), there exists an A such that $A(X)$ is *not* computationally indistinguishable from $A(Y)$. Such distribution ensembles X and Y exist if one-way functions exist. However, they can also be shown to exist unconditionally. For simplicity, we assume that X_n and Y_n obtain values in $\{0,1\}^n$. Now, since X and Y are not statistically close, it holds that for some set S some polynomial p and infinitely many n's

$$|\Pr[X_n \in S] - \Pr[Y_n \in S]| \geq \frac{1}{p(n)}$$

Without loss of generality, we can assume that

$$\Pr[X_n \in S] - \Pr[Y_n \in S] \geq \frac{1}{p(n)}$$

This yields an algorithm A as follows: given an input α, algorithm A outputs 1 if and only if $\alpha \in S$ (this check is of course not efficient). It therefore follows that

$$\Pr[A(X_n) = 1] - \Pr[A(Y_n) = 1] = \Pr[X_n \in S] - \Pr[Y_n \in S] \geq \frac{1}{p(n)}$$

Thus, clearly $A(X)$ is not computationally indistinguishable from $A(Y)$. Since we did not learn about statistical closeness, a simpler and yet still acceptable answer is as follows. Assume the existence of one-way functions and construct a pseudorandom generator with expansion factor $2n$. Then, construct A that outputs 1 if and only if its input string is in the range of the generator. It holds that

$$\Pr[A(U_{2n}) = 1] \leq \frac{2^n}{2^{2n}} = \frac{1}{2^n}$$

and

$$\Pr[A(G(U_n)) = 1] = 1$$

Thus, $A(U_{2n})$ and $A(G(U_n))$ are not computationally indistinguishable.

Exercise 3: Let f be a length-preserving one-way function, and let b be a hard-core predicate of f. Prove or refute: $G(x) = (f(U_n), b(U_n))$ is a pseudorandom generator.

Solution 3: It can be shown that if there exist one-way functions then there exist length-preserving one-way functions. Now, given a length-preserving one-way function f, the function g that is derived by computing f and then setting the first bit to 0 is still a one-way function (prove!). However, in this case, the first bit of the output of the generator is always 0. This yields a distinguisher that succeeds in distinguishing with probability 1/4 (prove!).

Exercise 4:

1. Prove that if there exist pseudorandom generators, then there exist pseudorandom generators that are not 1–1.
2. Prove that if there exist one-way permutations, then there exist pseudorandom generators (with any expansion factor) that are 1–1.

Solution 4: Let G be a pseudorandom generator with stretch factor $4n$ (such a generator exists if pseudorandom generators exist). Then, define $G' : \{0,1\}^{2n} \to \{0,1\}^{4n}$ such that $G'(x_1, x_2) = G(x_1) \oplus G(x_2)$. Note that $G'(x_1, x_2) = G'(x_2, x_1)$ and so G' is not 1–1. Furthermore, it is not difficult to prove that G' is pseudorandom (you must prove this).

Regarding the second part of the exercise, note that the construction that we saw in class is 1–1 and thus fulfills the requirements. The reason that it is 1–1 is that it contains $f(x)$ where f is a permutation. Thus, for all $x \neq x'$ it holds that $G(x) \neq G(x')$ as required.

Exercise 5: Prove that the existence of pseudorandom generators with expansion factor $l(n) = 2n$ implies the existence of one-way functions.[2] You may *not* copy the answer from a text (or the Internet), but must prove the theorem by yourselves.

Hint: Define $f(x, y) = G(x)$, where $|x| = |y|$.

[2] We will see in class that the assumption is equivalent to the existence of any pseudorandom generator.

Solution 5: Let G be a pseudorandom generator with expansion factor $l(n)$. Assume by contradiction that $f(x,y) = G(x)$, where $|x| = |y|$ is not a one-way function. This implies that there exists a PPT algorithm A and a polynomial p such that for infinitely many n's

$$\Pr[A(G(U_n), 1^{2n}) \in G^{-1}(G(U_n))] \geq \frac{1}{p(n)}$$

We construct a distinguisher D for G as follows. D runs A on its input $z \in \{0,1\}^{2n}$ and outputs 1 if and only if A outputs (x,y) such that $G(x) = z$. We claim that D is a good distinguisher for G. First note that

$$\Pr[D(U_{2n} = 1] < \frac{1}{2^n}$$

This holds because there are at most 2^n values z in the range of G. Since there are 2^{2n} different values in $\{0,1\}^{2n}$ it follows that a uniformly distributed string of length $2n$ is in the range of G (with input of length n) with probability at most $2^n/2^{2n} = 2^{-n}$. Next, we claim that for infinitely many n's

$$\Pr[D(G(U_n)) = 1] \geq \frac{1}{p(n)}$$

This follows immediately from the way we constructed D and from the assumed success of A in inverting the function $f(x,y) = G(x)$. Since for sufficiently large n's it holds that $2^{-n} < 1/2p(n)$ we have that for infinitely many n's

$$|\Pr[D(U_{2n} = 1] - \Pr[D(G(U_n)) = 1]| \geq \frac{1}{2p(n)}$$

in contradiction to the assumed pseudorandomness of G. Thus, we conclude that f is a one-way function and so the existence of pseudorandom generators implies the existence of one-way functions.

Exercise 5: Since the solution of this exercise is really just to rewrite parts of the Goldreich-Levin proof (in simplified cases), I will not provide a solution. (A solution is implicit in the lecture notes.)