Fundamentals of Cryptography: Problem Set 3

Due Wed Oct 9 3PM

Collaboration is permitted (and encouraged); however, you must write up your own solutions and acknowledge your collaborators.

If a problem has **0pt**, it will not be graded.

Problem 0 Read Section 7.1, 7.2, 7.3 of "Introduction to Modern Cryptography (2nd ed)" by Katz & Lindell.

If you are curious about how to construct PRG from OWF, you may read "Pseudorandom Generators from One-Way Functions: A Simple Construction for Any Hardness" by Thomas Holenstein.

Problem 1 (0pt): Concentration Inequalities This problem recaps a few useful probability bounds. They show how random variables "concentrate" around their means. Section A of "Introduction to Modern Cryptography (2nd ed)" may help you answer this question.

Part A (Markov's Inequality) Let X be a random variable over non-negative real numbers. Prove that, for any $a > 0$,

$$\Pr[X \geq a] \leq \frac{\mathbb{E}[X]}{a}.$$

Part B (Chernoff Bound) Let $p \in [0,1]$ be a constant. Let $X_1, \ldots, X_n$ be random variables that are sampled independently from $\mathrm{Bern}(p)$. That is, for each $i \in \{1, \ldots, n\}$, we have $X_i \in \{0,1\}$ and $\Pr[X_i = 1] = p$.

(1) Compute $\mathbb{E}[e^{t\sum_i X_i}]$ for any $t \in \mathbb{R}$.

(2) Prove that,

$$\Pr\Big[\frac{1}{n}\sum_i X_i \geq p + \varepsilon\Big] \leq \frac{\mathbb{E}[e^{t\sum_i X_i}]}{e^{tn(p+\varepsilon)}},$$

for any $t > 0$.

(3) Optimize the above bound by choosing t wisely.

The optimized bound is call *Chernoff bound*, it should looks like

$$\Pr\Big[\frac{1}{n}\sum_i X_i \geq p + \varepsilon\Big] \leq e^{-D(p+\varepsilon\|p)\cdot n},$$

where $D(p+\varepsilon\|p)$ is the notatino of KL divergence, and is defined as $D(p+\varepsilon\|p) := (p+\varepsilon)\log(\frac{p+\varepsilon}{p}) + (1-p-\varepsilon)\log(\frac{1-p-\varepsilon}{1-p})$. Since $D(p+\varepsilon\|p) \geq 2\varepsilon^2$, *Chernoff bound* can be relaxed to

$$\Pr\Big[\frac{1}{n}\sum_i X_i \geq p + \varepsilon\Big] \leq e^{-2\varepsilon^2 n}.$$

Part C (Chebyshev's Inequality) Let X be a random variable. Prove that

$$\Pr\Big[|X - \mathbb{E}[X]| \geq a\Big] \leq \frac{\mathrm{Var}[X]}{a^2}$$

for any $a > 0$. Here $\mathrm{Var}[X] := \mathbb{E}[(X - \mathbb{E}[X])^2]$ is the variance of X.

Let $X_1, \ldots, X_n$ be random variables such that $\mathbb{E}[X_i] = p$ and $\mathrm{Var}[X_i] = \sigma^2$ for all i. We also assume that $X_1, \ldots, X_n$ are *pair-wise independent*. Prove that

$$\Pr\Big[\Big|\frac{1}{n}\sum_i X_i - p\Big| \geq a\Big] \leq \frac{\sigma^2}{na^2}$$

for any $a > 0$.

Problem 2 (14pt) Assume f is a length-preserving OWF (i.e., $|f(x)| = |x|$). In each of the following cases, prove f' is a OWF, or show a counterexample.

Part A $f'(x) := f(x)\|f(f(x))$.

Part B $f'(x) := x \oplus f(x)$.

Part C $f'(x) := f(x)\|f(\bar{x})$, where $\bar{x}$ denote the bit-wise NOT operation.

Part D $f'(x) := f(G(x))$, where G is a PRG that $|G(s)| = |s| + 1$.

Part E $f'(x) := G(f(x))$, where G is a PRG that $|G(s)| = |s| + 1$.

Part F $f'(x) := f(x\|\underbrace{0\ldots0}_{\log n \text{ many}})$, where $n = |x|$.

Part G $f'(x) := (f(x))_{1:(n-\log n)}$, where $n = |x|$. That is, $f'(x)$ outputs the first $n - \log(n)$ bits of $f(x)$.

Problem 3 (6pt) Hardness Amplification of Weak OWFs For simplicity, we consider length-preserving weak OWF. $f : \{0,1\}^* \to \{0,1\}^*$ is a length-preserving weak OWF, if $|f(x)| = |x|$ for any $x \in \{0,1\}^*$, and there exists a polynomial q, such that for any PPT $\mathcal{A}$, for any sufficiently large n,

$$\Pr_{\substack{x \xleftarrow{\$} \{0,1\}^n \\ \hat{x} \leftarrow \mathcal{A}(f(x))}}\Big[f(\hat{x}) = f(x)\Big] \leq 1 - \frac{1}{q(n)}.$$

(Note the order of the quantifiers!)

Assume f is such a weak OWF. Define f' such that for $x_1, \ldots, x_m \in \{0,1\}^n$,

$$f'(x_1\|\ldots\|x_m) = f(x_1)\|\ldots\|f(x_m)$$

where $m = m(n)$ is a polynomial on n. ($m(n)$ will be fixed later.)

We prove f' is a OWF by contradiction. Assume f' is not a OWF, then there exists PPT $\mathcal{A}'$, and polynomial p such that

$$\Pr_{\substack{x_1,\ldots,x_m \xleftarrow{\$} \{0,1\}^n \\ \hat{x}_1,\ldots,\hat{x}_m \leftarrow \mathcal{A}'(f(x_1)\|\ldots\|f(x_m))}}\Big[f(\hat{x}_1) = f(x_1), \ldots, f(\hat{x}_m) = f(x_m)\Big] > \frac{1}{p(n)}$$

for infinitely many integer n.

Define $\mathcal{A}$ as

$$\begin{array}{|ll|}\hline \mathcal{A}(y) & \text{let } n = |y|, \text{ let } m = m(n) \\ & \text{sample } i \xleftarrow{\$} \{1, \ldots, m\} \\ & \text{for all } j \neq i, \text{ sample } x_j \xleftarrow{\$} \{0,1\}^n \text{ and let } y_j = f(x_j) \\ & \text{let } y_i = y \\ & \text{call } \hat{x}_1 \| \ldots \| \hat{x}_m \leftarrow \mathcal{A}'(y_1 \| \ldots \| y_m) \\ & \text{if } f'(\hat{x}_1 \| \ldots \| \hat{x}_m) = y_1 \| \ldots \| y_m, \\ & \qquad \text{output } \hat{x}_i \\ \hline \end{array}$$

We say $x \in \{0,1\}^n$ is "good" if $\mathcal{A}$ inverts $f(x)$ with a good probability. Concretely, we define $x \in \{0,1\}^n$ is "good" if and only if

$$\Pr_{\hat{x} \leftarrow \mathcal{A}(f(x))} \Big[f(\hat{x}) = f(x) \Big] \geq \frac{1}{r(n)}$$

for a polynomial $r(n)$. ($r(n)$ will be fixed later.) If x is not "good", we say x is "bad".

Part A Prove that

$$\Pr_{\substack{x_1, \ldots, x_m \xleftarrow{\$} \{0,1\}^n \\ \hat{x}_1, \ldots, \hat{x}_m \leftarrow \mathcal{A}'(f(x_1) \| \ldots \| f(x_m))}} \Big[f(\hat{x}_1) = f(x_1), \ldots, f(\hat{x}_m) = f(x_m) \Big] \leq \frac{m^2}{r(n)} + \Big(\Pr_{x \leftarrow \{0,1\}^n} [x \text{ is "good"}] \Big)^m,$$

for any sufficiently large n.

Part B Choose polynomials $m(n)$, $r(n)$ properly, so that

$$\Pr_{x \leftarrow \{0,1\}^n} [x \text{ is "bad"}] \leq \frac{1}{2q(n)}$$

for infinitely many n. (Note that, you can let $r(n)$ depend on both $p(n)$ and $q(n)$; while $m(n)$ can depend on $q(n)$ and cannot depend on $p(n)$.)

Part C Define $\mathcal{A}_{\text{repeat}}$ as

$$\begin{array}{|ll|}\hline \mathcal{A}_{\text{repeat}}(y) & \text{let } n = |y| \\ & \text{repeat the following for } n \cdot r(n) \text{ times} \\ & \qquad \text{call } \hat{x} \leftarrow \mathcal{A}(y) \\ & \qquad \text{if } f(\hat{x}) = y, \\ & \qquad\qquad \text{output } \hat{x} \\ \hline \end{array}$$

Show that $\mathcal{A}_{\text{repeat}}$ violates our assumptions on f.

The contradiction rules out our assumption. So f' must be a OWF.

Problem 4 (bonus 3pt) How to invert OWP, with Preproccessing To invert an OWP, the naive algorithm is to enumerate all possible inputs, which takes 2^n times for n-bit inputs/outputs. There is no obvious smarter algorithm for a general OWP. In this problem, we show how to reduce the time complexity of inverting a OWP, if preproccessing is allowed.

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a permutation that can be efficiently computed. For this problem, we are not considering the asymptotic complexity, you can assume that f can be computed in unit time.

The attack is done in a different setting that allows preproccessing. The adversary is split into $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$.

- $\mathcal{A}_0$ is the preproccessing algorithm and is unbounded. For example, $\mathcal{A}_0$ can query $f(x)$ for all $x \in \{0,1\}^n$. In the end, $\mathcal{A}_0$ should ouput a bounded length advice string L, which will be passed to $\mathcal{A}_1$.

- $\mathcal{A}_1$ is the online algorithm (in the RAM model). It takes the advice string L and a value $f(x)$ as inputs. Its task is to find x in bounded time.

For any integer $t < 2^n$, construct adversary $\mathcal{A} = (\mathcal{A}_0, \mathcal{A}_1)$ such that

- the advice string L contains up to $\frac{2^n}{t}\,\mathrm{poly}(n)$ bits,

- for any $x \in \{0,1\}^n$, the online algorithm $\mathcal{A}_1(L, f(x))$ always outputs x in at most $t\,\mathrm{poly}(n)$ times.