0368.4162 Foundations of Cryptography November 2, 2017

Problem Set 1 - Solution

,Nathan Geier *Due: November 16*

1. (25 pts) In class, we saw that for any encryption scheme (E, D) for messages of length ℓ, with keys of length $n \leq \ell - 10$, if E is deterministic, there exist two messages m_0, m_1 and an inefficient A such that

$$\Pr\left[A(ct) = m_b \;\middle|\; \begin{array}{l} sk \leftarrow \{0,1\}^n \\ b \leftarrow \{0,1\} \\ ct \leftarrow E_{sk}(m_b) \end{array}\right] > 0.99 \ .$$

Show that the same holds even if E also tosses, say n, coins (on top of the key). That is, an encryption of any message $m \in \{0,1\}^\ell$ is drawn from a distribution $\{ct \mid sk \leftarrow \{0,1\}^n, r \leftarrow \{0,1\}^n, ct = E_{sk}(m;r)\}$.

Proof. Fix a message m_0 and let $m_1 \leftarrow \{0,1\}^\ell$.
Consider the adversary A that given ct outputs m_1 if

$$m_1 \in \{D_s(ct) \mid s \in \{0,1\}^n\}$$

and otherwise outputs m_0.
We have that

$$\Pr_{m_1,sk,r}\left[m_1 \in \{D_s(E_{sk}(m_0;r)) \mid s \in \{0,1\}^n\}\right] \leq \frac{2^n}{2^\ell}$$

because after fixing sk and r the size of $\{D_s(E_{sk}(m_0;r)) \mid s \in \{0,1\}^n\}$ is at most 2^n and m_1 was drawn uniformly over $\{0,1\}^\ell$ of size 2^ℓ.
Therefore, we have that

$$\mathbb{E}_{m_1}\Pr_{sk,r}\left[m_1 \in \{D_s(E_{sk}(m_0;r)) \mid s \in \{0,1\}^n\}\right] = \Pr_{m_1,sk,r}\left[m_1 \in \{D_s(E_{sk}(m_0;r)) \mid s \in \{0,1\}^n\}\right] \leq \frac{2^n}{2^\ell}$$

so there must exist a specific m_1 such that

$$\Pr_{sk,r}\left[m_1 \in \{D_s(E_{sk}(m_0;r)) \mid s \in \{0,1\}^n\}\right] \leq \frac{2^n}{2^\ell} \ .$$

Let us fix such specific m_1.
This means that

$$\Pr\left[A(ct) = m_1 \mid b = 0\right] \leq \frac{2^n}{2^\ell} \leq 2^{-10} \ .$$

Obviously,

$$\Pr_{sk,r}\left[m_1 \in \{D_s(E_{sk}(m_1;r)) \mid s \in \{0,1\}^n\}\right] = 1$$

becuase

$$\Pr_{sk,r}\left[m_1 = D_{sk}(E_{sk}(m_1;r))\right] = 1$$

so we have that

$$\Pr\left[A(ct) = m_1 \mid b = 1\right] = 1 \ .$$

Overall, it holds that

$$\begin{aligned}\Pr[A(ct)=m_b] &= \Pr[b=0]\Pr[A(ct)=m_0 \mid b=0] + \Pr[b=1]\Pr[A(ct)=m_1 \mid b=1] \\ &= \frac{1}{2}\left(\Pr[A(ct)=m_0 \mid b=0] + 1 - \Pr[A(ct)=m_0 \mid b=1]\right) \\ &\geq \frac{1}{2}\left(1+1-2^{-10}\right) > 0.99 \ .\end{aligned}$$

□

2. In this question, let X_0, X_1 be two distributions with the same finite supoort S. Recall that:

$$\Delta_A(X_0, X_1) := \left|\Pr[A(X_0)=1] - \Pr[A(X_1)=1]\right| \ .$$

(a) (15 pts) Show that for any function A (efficient or not):

$$\left|\Pr\left[A(x)=b \;\middle|\; \begin{array}{l} b \leftarrow \{0,1\} \\ x \leftarrow X_b \end{array}\right] - \frac{1}{2}\right| = \frac{\Delta_A(X_0,X_1)}{2} \ .$$

Proof.

$$\begin{aligned}&\left|\Pr\left[A(x)=b \;\middle|\; \begin{array}{l} b \leftarrow \{0,1\} \\ x \leftarrow X_b \end{array}\right] - \frac{1}{2}\right| = \\ &\left|\Pr[b=0]\Pr[A(X_0)=0] + \Pr[b=1]\Pr[A(X_1)=1] - \frac{1}{2}\right| = \\ &\frac{1}{2}\left|1 - \Pr[A(X_0)=1] + \Pr[A(X_1)=1] - 1\right| = \\ &\frac{1}{2}\left|\Pr[A(X_1)=1] - \Pr[A(X_0)=1]\right| = \frac{\Delta_A(X_0,X_1)}{2}\end{aligned}$$

□

(b) (15 pts) Show that

$$\max_A \Delta_A(X_0,X_1) = \frac{1}{2}\sum_{x\in S}\left|\Pr[X_0=x] - \Pr[X_1=x]\right| = \max_{T\subseteq S}\left|\Pr[X_0\in T] - \Pr[X_1\in T]\right| \ .$$

Proof. It is easy to see that for deterministic A's we have that

$$\max_A \Delta_A(X_0,X_1) = \max_A \left|\Pr[A(X_0)=1] - \Pr[A(X_1)=1]\right| = \max_{T\subseteq S}\left|\Pr[X_0\in T] - \Pr[X_1\in T]\right|$$

because we can always define $T = \{x \in S \mid A(x)=1\}$ or $A(x) = \begin{cases} 1 & x \in T \\ 0 & otherwise \end{cases}$ and then $A(x)=1 \iff x\in T$, and $\Pr[A(X_i)=1] = \Pr[X_i \in T]$. (recall that A is computationally unbounded so it can always decide if $x \in T$)

Let $T := arg\max_{T\subseteq S} |\Pr[X_0 \in T] - \Pr[X_1 \in T]|$

$$\left|\sum_{x\in T} \Pr[X_0 = x] - \Pr[X_1 = x]\right| = \left|\sum_{x\in T} \Pr[X_0 = x] - \sum_{x\in T} \Pr[X_1 = x]\right| = |\Pr[X_0 \in T] - \Pr[X_1 \in T]| =$$

$$|1 - \Pr[X_0 \in S \setminus T] - 1 + \Pr[X_1 \in S \setminus T]| = |\Pr[X_1 \in S \setminus T] - \Pr[X_0 \in S \setminus T]| =$$

$$\left|\sum_{x\in S\setminus T} \Pr[X_1 = x] - \Pr[X_0 = x]\right|$$

$$2|\Pr[X_0 \in T] - \Pr[X_1 \in T]| = \left|\sum_{x\in T} \Pr[X_0 = x] - \Pr[X_1 = x]\right| + \left|\sum_{x\in S\setminus T} \Pr[X_1 = x] - \Pr[X_0 = x]\right| \leq$$

$$\sum_{x\in S} |\Pr[X_0 = x] - \Pr[X_1 = x]|$$

$$|\Pr[X_0 \in T] - \Pr[X_1 \in T]| \leq \frac{1}{2}\sum_{x\in S} |\Pr[X_0 = x] - \Pr[X_1 = x]|$$

For the other direction, let $T := \{x \mid \Pr[X_0 = x] \geq \Pr[X_1 = x]\}$, then:

$$\frac{1}{2}\sum_{x\in S} |\Pr[X_0 = x] - \Pr[X_1 = x]| =$$

$$\frac{1}{2}\left(\sum_{x\in T} \Pr[X_0 = x] - \Pr[X_1 = x] + \sum_{x\in S\setminus T} \Pr[X_1 = x] - \Pr[X_0 = x]\right) =$$

$$\frac{1}{2}(\Pr[X_0 \in T] - \Pr[X_1 \in T] + \Pr[X_1 \in S \setminus T] - \Pr[X_0 \in S \setminus T]) =$$

$$\frac{1}{2}(\Pr[X_0 \in T] - \Pr[X_1 \in T] + 1 - \Pr[X_1 \in T] - 1 + \Pr[X_0 \in T]) =$$

$$\frac{1}{2}(2\Pr[X_0 \in T] - 2\Pr[X_1 \in T]) = \Pr[X_0 \in T] - \Pr[X_1 \in T] = |\Pr[X_0 \in T] - \Pr[X_1 \in T]|$$

□

(c) (15 pts) let A be a randomized distinguisher that uses n random coins. That is, $A(x; r)$ is given a sample x and a random string $r \leftarrow \{0,1\}^n$ (and as usually outputs a bit). Prove that there exists a fixed string $r \in \{0,1\}^n$ such that

$$\Delta_A((X_0; r), (X_1; r)) \geq \Delta_A((X_0; U_n), (X_1; U_n)) \ ,$$

where U_n is the uniform distribution on $\{0,1\}^n$.
(This means that for non-uniform algorithms, we can often assume w.l.o.g that they're deterministic.)

Proof. Assume w.l.o.g that $\Pr[A(X_0; U_n) = 1] \geq \Pr[A(X_1; U_n) = 1]$.

$$\mathbb{E}_r[\Pr[A(X_0; r) = 1] - \Pr[A(X_1; r) = 1]] = \mathbb{E}_r[\Pr[A(X_0; r) = 1]] - \mathbb{E}_r[\Pr[A(X_1; r) = 1]] =$$

$$\sum_{r\in\{0,1\}^n} \frac{\Pr[A(X_0; r) = 1]}{2^n} - \sum_{r\in\{0,1\}^n} \frac{\Pr[A(X_1; r) = 1]}{2^n} =$$

$$\Pr[A(X_0; U_n) = 1] - \Pr[A(X_1; U_n) = 1] = \Delta_A((X_0; U_n), (X_1; U_n))$$

Therefore, there must be some specific r such that

$$\Delta_A((X_0;r),(X_1;r)) = \Pr[A(X_0;r)=1] - \Pr[A(X_1;r)=1] \geq \Delta_A((X_0;U_n),(X_1;U_n))$$

otherwise the expection would be less than $\Delta_A((X_0;U_n),(X_1;U_n))$.

Remark: if $\Pr[A(X_0;U_n)=1] < \Pr[A(X_1;U_n)=1]$ we simply change the expectation to be of $\Pr[A(X_1;r)=1] - \Pr[A(X_0;r)=1]$. □

3. Let S_0 and S_1 be two non-uniform PPT algorithms, and let $X = \{X_n\}_{n\in\mathbb{N}}$ be a distribution ensemble. Assume that

$$X, S_0(X) \approx_c X, S_1(X) \ ,$$

Here $(X, S_b(X))$ denotes the distribution ensemble $\{X_n, S_b(X_n)\}_{n\in\mathbb{N}}$, where a sample (x,y) is given by first sampling $x \leftarrow X_n$ and then sampling $y \leftarrow S_b(x)$ (note that S_b is randomized and may toss additional coins of its own).

Let p be any polynomial. For $b \in \{0,1\}$, consider a new ensemble $Y_b = \{Y_{b,n}\}_{n\in\mathbb{N}}$, given by

$$Y_{b,n} = (X_n, \overbrace{S_b(X_n), \ldots, S_b(X_n)}^{p(n) \text{ times}}) \ ,$$

where a sample $(x, y_1, \ldots, y_{p(n)})$ is given by sampling $x \leftarrow X_n$ and then independently sampling each $y_i \leftarrow S_b(x)$.

(a) (30 pts) Show that $Y_0 \approx_c Y_1$.

Proof. First, we will prove a useful lemma.

Lemma 0.1. $\Delta_A(X,Z) \leq \Delta_A(X,Y) + \Delta_A(Y,Z)$

Proof.

$$\begin{aligned}
&\Delta_A(X,Z) = |\Pr[A(X)=1] - \Pr[A(Z)=1]| = \\
&|\Pr[A(X)=1] - \Pr[A(Y)=1] + \Pr[A(Y)=1] - \Pr[A(Z)=1]| \leq \\
&|\Pr[A(X)=1] - \Pr[A(Y)=1]| + |\Pr[A(Y)=1] - \Pr[A(Z)=1]| = \\
&\Delta_A(X,Y) + \Delta_A(Y,Z)
\end{aligned}$$

□

Define $Z_{i,n} = (X_n, \overbrace{S_1(X_n), \ldots, S_1(X_n)}^{i \text{ times}}, \overbrace{S_0(X_n), \ldots, S_0(X_n)}^{p(n)-i \text{ times}})$ and $Z_i = \{Z_{i,n}\}_{n\in\mathbb{N}}$ so $Y_0 = Z_0$ and $Y_1 = Z_{p(n)}$.
Let $A = \{A_n\}_{n\in\mathbb{N}}$ be a non-uniform PPT.
We have that $\forall n \in \mathbb{N} \quad \Delta_{A_n}(Z_{i,n}, Z_{i+1,n}) \leq \Delta_{A'_n}((X_n, S_0(X_n)), (X_n, S_1(X_n)))$ where A'_n is the algorithm that given a sample (x, y_b) of $(X_n, S_b(X_n))$ constructs $(x, \overbrace{S_1(x), \ldots, S_1(x)}^{i \text{ times}}, y_b, \overbrace{S_0(x), \ldots, S_0(x)}^{p(n)-i-1 \text{ times}})$ in poly-time, which is a sample of $Z_{i,n}$ if $b=0$ and of $Z_{i+1,n}$ if $b=1$, then it calls A_n to differentiate between $Z_{i,n}$ and $Z_{i+1,n}$ with some probability, and so it determines the value of b with the same probaility.
$A' = \left\{A'_n\right\}_{n\in\mathbb{N}}$ is PPT because we used S_0 and S_1 at most a poly number of times, and each of them runs in poly time, and A also runs in poly time.

Denote by $\mu(n)$ the negligible function such that $\Delta_{A'_n}((X_n, S_0(X_n)), (X_n, S_1(X_n))) \leq \mu(n)$, then we have for all n:

$$\Delta_{A_n}(Y_{0,n}, Y_{1,n}) = \Delta_{A_n}(Z_{0,n}, Z_{p(n),n}) \overset{(0.1)}{\leq} \sum_{i=0}^{p(n)-1} \Delta_{A_n}(Z_{i,n}, Z_{i+1,n}) \leq$$
$$p(n) \cdot \Delta_{A'_n}((X_n, S_0(X_n)), (X_n, S_1(X_n))) \leq p(n)\mu(n)$$

And since a poly multiplied by a negligible function is also a negligible function, we have that $Y_0 \approx_c Y_1$. □

(b) (**Bonus**: 10 pts) show that if S_0, S_1 may be **inefficient**, and there exists computationally-secure (secret-key) encryption, the previous claim is not true.

Proof. Let (E,D) be a computationally-secure (secret-key) encryption with keys of length n and messages of length $n+10$. (We proved in class that if computationally-secure encryption for keys of length n and larger messages exists, one also exists for messages of length up to $n + p(n)$ for any poly p)
Let m_0, m_1, A be the two messages and unbounded adversary such that

$$\Pr\left[A(ct) = m_b \;\middle|\; \begin{array}{l} sk \leftarrow \{0,1\}^n \\ b \leftarrow \{0,1\} \\ ct \leftarrow E_{sk}(m_b) \end{array} \right] > 0.99 \ .$$

Define:
X_n as the distribution of $E_{sk}(m_b)$ for random sk, b.
$S_i(x_n)$ computes $A(x_n)$, and builds with it the 2-bit string

$$st := (A(x_n) == m_i, (A(x_n) == m_i) \oplus i)$$

Then it draws $j \leftarrow \{1,2\}$ and returns $(j, st(j))$ where $st(j)$ is the j'th bit of st.

	S_0	S_1
$A(x_n) = m_0$	11	01
$A(x_n) = m_1$	00	10

So if we get enough samples of $S_i(x_n)$ we will know both bits with high probability, then we can XOR them and know what i is. Therefore, $Y_0 \not\approx_c Y_1$.
But what if we only have one sample? We would like to prove that we cannot learn anything from it.
Say the sample is $(1,1)$, then deciding that it came from $S_i(x_n)$ would be the same as deciding that $A(x_n) = m_i$. Since A "beats" (E, D) for m_0, m_1, obviously we can't do that with non-negligible probability in polynomial time, otherwise (E, D) would not be secure.
Symmetrical argument goes for $(1,0)$.
If the sample is $(2,0)$ then we know for sure that $A(x_n) = m_1$, but even an unbounded adversary wouldn't be able to know if the sample came from S_0 or S_1, because both of them return the same thing given that $j = 2$.
Same argument goes for $(2,1)$.
Since no matter what we get we cannot distinguish with non-negligible probability in poly time, we have that $X, S_0(X) \approx_c X, S_1(X)$. □