CS 171: Discussion Section 3 (Feb 5)

1. Pseudorandom Generators

Let $F, G : \{0,1\}^n \to \{0,1\}^{3n}$ be pseudorandom generators. For each of the functions below, prove or disprove that H is necessarily a pseudorandom generator.

(a) $H(s_0 s_1 \dots s_{n-1}) := G(s_{n-1} s_{n-2} \dots s_0)$.

(b) $H(s) := G(s)_{0,\dots,2n-1}$ (i.e., the first $2n$ bits of $G(s)$).

(c) $H(s) = G(s) \| F(s)$.

Solution (a) H is a PRG. Suppose for the purpose of contradiction that H is not a PRG, then there exists a PPT $\mathcal{A}$ such that $|\Pr[\mathcal{A}(U_{3n}) = 1] - \Pr[\mathcal{A}(H(U_n)) = 1]|$ is non-negligible. Construct a PPT $\mathcal{B}$ to break G as follows: on input $t \in \{0,1\}^{3n}$, output $\mathcal{A}(t)$. Note that $\Pr[\mathcal{B}(U_{3n}) = 1] = \Pr[\mathcal{A}(U_{3n}) = 1]$, $\Pr[\mathcal{B}(G(U_n)) = 1] = \Pr[\mathcal{A}(H(U_n)) = 1]$, hence $|\Pr[\mathcal{B}(U_{3n}) = 1] - \Pr[\mathcal{B}(G(U_n)) = 1]| = |\Pr[\mathcal{A}(U_{3n}) = 1] - \Pr[\mathcal{A}(H(U_n)) = 1]|$ is non-negligible, $\mathcal{B}$ breaks the pseudorandomness of G, contradiction.

(b) H is a PRG. Suppose for the purpose of contradiction that H is not a PRG, then there exists a PPT $\mathcal{A}$ such that $|\Pr[\mathcal{A}(U_{2n}) = 1] - \Pr[\mathcal{A}(H(U_n)) = 1]|$ is non-negligible. Construct a PPT $\mathcal{B}$ to break G as follows: on input $t \in \{0,1\}^{3n}$, output $\mathcal{A}(t_{0,1,\dots,2n-1})$. Note that $\Pr[\mathcal{B}(U_{3n}) = 1] = \Pr[\mathcal{A}(U_{2n}) = 1]$, $\Pr[\mathcal{B}(G(U_n)) = 1] = \Pr[\mathcal{A}(H(U_n)) = 1]$, hence $|\Pr[\mathcal{B}(U_{3n}) = 1] - \Pr[\mathcal{B}(G(U_n)) = 1]| = |\Pr[\mathcal{A}(U_{2n}) = 1] - \Pr[\mathcal{A}(H(U_n)) = 1]|$ is non-negligible, $\mathcal{B}$ breaks the pseudorandomness of G, contradiction.

(c) H is not a PRG. If $G = F$ then the first and the second half will be the same.

□

2. Equivalence of Definitions

Consider the following variant of CPA secure definition.

1. A key k is generated by running $\mathsf{Gen}(1^n)$.
2. The adversary $\mathcal{A}$ on input 1^n and oracle access to $\mathsf{Enc}_k(\cdot)$ produces a tuple of messages $(m_{0,1}, \dots, m_{0,r})$ and $(m_{1,1}, \dots, m_{1,r})$ where $m_{0,i}$ and $m_{1,i}$ have the same length.
3. A uniform bit $b \in \{0,1\}$ is chosen and for each $i \in [r]$, c_i is generated as $\mathsf{Enc}_k(m_{b,i})$ and the tuple of ciphertexts $(c_1, \dots, c_r)$ is given to the adversary.
4. The adversary $\mathcal{A}$ continues to have oracle access to $\mathsf{Enc}_k(\cdot)$ and outputs a bit b'.
5. The output of the experiment is defined to be 1 if and only if $b = b'$.

We say that an encryption scheme to be strong CPA secure if for every $\mathcal{A}$ there is a negligible function ν such that:

$$\Pr[PrivK^{S-CPA}_{\mathcal{A},\Pi}(n) = 1] \leq 1/2 + \nu(n)$$

Show that the strong CPA security is equivalent to CPA security.

Solution *Hint: A common technique in cryptography is called hybrid argument. The basic idea is the following: If $\mathcal{A}$ can distinguish two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$ with non-negligible probability, then we can construct a sequence of distributions $H_0, H_1, \ldots, H_n$ where $H_0 = \mathcal{D}_1$ and $H_n = \mathcal{D}_2$, and n is polynomial in the security parameter. By the properties of negligible functions, $\mathcal{A}$ must be able to distinguish at least two neighboring hybrids H_i and H_{i+1} with non-negligible probability.*

In order to show the equivalence, we need to show that strong CPA security implies the standard CPA security and vice versa. Let us first start with the easy direction where we first show that strong CPA security implies CPA security.

Strong CPA $\Rightarrow$ CPA security. Notice that the only difference between these two definitions is that in the strong CPA security, $\mathcal{A}$ has the option of outputting two tuples of messages instead of a pair of messages. Thus, if we set $r = 1$ in the above definition, it is equivalent to standard CPA security. This shows that strong CPA security implies the standard CPA security.

CPA Security $\Rightarrow$ Strong CPA security. The interesting direction is the other one where we show that strong CPA security is implied by the weaker version.

Consider an encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ which is secure as per the standard CPA security definition. We will show that it is secure as per the stronger CPA security definition. To show this, we will use a hybrid argument. Specifically, we will define a sequence of hybrids starting with the hybrid which corresponds to the strong CPA experiment with the bit $b = 0$ and end with a hybrid which corresponds to the strong CPA experiment with the bit $b = 1$. We will show that each of the intermediate hybrids are indistinguishable from the standard CPA security of the encryption scheme.

$\underline{\mathsf{Hyb}_0}$: This corresponds to the strong CPA experiment where the bit $b = 0$. More formally, for any adversary $\mathcal{A}$,

1. A key k is generated by running $\mathsf{Gen}(1^n)$.
2. The adversary $\mathcal{A}$ on input 1^n and oracle access to $\mathsf{Enc}_k(\cdot)$ produces a tuple of messages $(m_{0,1}, \ldots, m_{0,r})$ and $(m_{1,1}, \ldots, m_{1,r})$ where $m_{0,i}$ and $m_{1,i}$ have the same length.
3. For each $i \in [r]$, c_i is generated as $\mathsf{Enc}_k(m_{0,i})$ and the tuple of ciphertexts $(c_1, \ldots, c_r)$ is given to the adversary.
4. The adversary $\mathcal{A}$ continues to have oracle access to $\mathsf{Enc}_k(\cdot)$ and outputs a bit b'.
5. The output of the experiment is defined to be b'.

$\underline{\mathsf{Hyb}_1}$: This hybrid is exactly same as the previous hybrid except that for $i = 1$, we change c_1 from encrypting $m_{0,1}$ to encrypting $m_{1,1}$. More formally,

1. A key k is generated by running $\mathsf{Gen}(1^n)$.
2. The adversary $\mathcal{A}$ on input 1^n and oracle access to $\mathsf{Enc}_k(\cdot)$ produces a tuple of messages $(m_{0,1}, \ldots, m_{0,r})$ and $(m_{1,1}, \ldots, m_{1,r})$ where $m_{0,i}$ and $m_{1,i}$ have the same length.

3. Generate c_1 as $\mathsf{Enc}_k(m_{1,1})$. For each $i \in [r] \setminus \{1\}$, c_i is generated as $\mathsf{Enc}_k(m_{0,i})$ and the tuple of ciphertexts $(c_1, \ldots, c_r)$ is given to the adversary.

4. The adversary $\mathcal{A}$ continues to have oracle access to $\mathsf{Enc}_k(\cdot)$ and outputs a bit b'.

5. The output of the experiment is defined to be b'.

More generally, we define Hyb_j for any $j \in [r]$ is defined as follows.

Hyb_j :

1. A key k is generated by running $\mathsf{Gen}(1^n)$.

2. The adversary $\mathcal{A}$ on input 1^n and oracle access to $\mathsf{Enc}_k(\cdot)$ produces a tuple of messages $(m_{0,1}, \ldots, m_{0,r})$ and $(m_{1,1}, \ldots, m_{1,r})$ where $m_{0,i}$ and $m_{1,i}$ have the same length.

3. For each $i \leq j$, generate c_i as $\mathsf{Enc}_k(m_{1,i})$. For each $i \in [r] \setminus [j]$, c_i is generated as $\mathsf{Enc}_k(m_{0,i})$ and the tuple of ciphertexts $(c_1, \ldots, c_r)$ is given to the adversary.

4. The adversary $\mathcal{A}$ continues to have oracle access to $\mathsf{Enc}_k(\cdot)$ and outputs a bit b'.

5. The output of the experiment is defined to be b'.

We now show that for any $j \in [r]$, Hyb_j is computationally indistinguishable to Hyb_{j-1}.

Claim 0.1. *Assume that* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *satisfies the standard CPA security definition. Then, for any adversary* $\mathcal{A}$ *and* $j \in [r]$, *there exists a negligible function* $\nu(\cdot)$

$$|\Pr[\mathsf{Hyb}_{j-1} \textit{ outputs 1}] - \Pr[\mathsf{Hyb}_j \textit{ outputs 1}]| \leq \nu(n)$$

Proof. Assume for the sake of contradiction that there exists an adversary $\mathcal{A}$ and $j \in [r]$ such for every negligible function $\nu(\cdot)$,

$$|\Pr[\mathsf{Hyb}_{j-1} \text{ outputs 1}] - \Pr[\mathsf{Hyb}_j \text{ outputs 1}]| \geq \nu(n)$$

We will now use such an adversary $\mathcal{A}$ and the corresponding j, to construct an adversary $\mathcal{B}$ against the standard CPA security definition of $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. We now give the description of $\mathcal{B}$.

Description of $\mathcal{B}$.

1. $\mathcal{B}$ on input 1^n, starts running $\mathcal{A}$ on input 1^n.

2. **Phase-1 oracle queries.** For every query that $\mathcal{A}$ makes to the the encryption oracle in phase-1, $\mathcal{B}$ answers them using its own encryption oracle. Specifically, for every message m that $\mathcal{A}$ queries to $\mathsf{Enc}_k(\cdot)$ oracle, $\mathcal{B}$ submits m as the message to its $\mathsf{Enc}_k(\cdot)$ oracle and obtains the response. It forwards this response to $\mathcal{A}$.

3. **Challenge Messages.** $\mathcal{A}$ now submits a tuple of r messages, $(m_{0,1}, \ldots, m_{0,r})$ and $(m_{1,1}, \ldots, m_{1,r})$. To generate the challenge ciphertexts, $(c_1, \ldots, c_r)$, $\mathcal{B}$ does the following. For each $i < j$, $\mathcal{B}$ submits $m_{1,i}$ to its $\mathsf{Enc}_k(\cdot)$ oracle and obtains the ciphertexts $(c_1, \ldots, c_{j-1})$. For each $i \geq j + 1$, it submits $m_{0,i}$ to its $\mathsf{Enc}_k(\cdot)$ oracle and obtains the ciphertexts $(c_{j+1}, \ldots, c_r)$. Finally, it submits $m_{0,j}$ and $m_{1,j}$ as its challenge messages and obtains a ciphertext c^*. It gives the tuple $(c_1, \ldots, c_{j-1}, c^*, c_{j+1}, \ldots, c_r)$ as the challenge ciphertext.

4. **Phase-2 oracle queries.** For every query $\mathcal{A}$ makes to the encryption oracle, $\mathcal{B}$ answers them exactly as in phase-1.

5. $\mathcal{A}$ finally outputs a bit b' and $\mathcal{B}$ outputs this bit.

Now, note that if c^* is an encryption of the message $m_{0,j}$, then distribution of the challenge ciphertexts given to $\mathcal{A}$ is identically distributed to Hyb_{j-1}. On the other hand, if c^* was an encryption of the message $m_{1,j}$, then the distribution of the challenge ciphertexts given to $\mathcal{A}$ is identically distributed to Hyb_j. Thus, if for every negligible function,

$$|\Pr[\mathsf{Hyb}_{j-1} \text{ outputs } 1] - \Pr[\mathsf{Hyb}_j \text{ outputs } 1]| \geq \nu(n)$$

then, for every negligible function $\nu(\cdot)$

$$\Pr[PrivK^{cpa}_{\mathcal{B},\Pi} = 1] \geq 1/2 + \nu(n)$$

and this contradicts the standard CPA security of $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$.

□ Now,

$$\begin{aligned} |\Pr[\mathsf{Hyb}_0 \text{ outputs } 1] - \Pr[\mathsf{Hyb}_r \text{ outputs } 1]| &\leq \sum_{j \in [r]} |\Pr[\mathsf{Hyb}_{j-1} \text{ outputs } 1] - \Pr[\mathsf{Hyb}_j \text{ outputs } 1]| \\ &\leq r \cdot \nu(n) \text{ (from Claim 0.1)} \\ &= \nu'(n) \end{aligned}$$

Observe that Hyb_r is identically distributed to the strong CPA security experiment where the challenge bit is set to 1. Thus, we showed that $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ satisfies the stronger CPA security definition. □