CS 171: Problem Set 3

Due Date: February 15th, 2024 at 8:59pm via Gradescope

1. Pseudorandom Functions

Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a pseudorandom function (PRF). For the functions f' below, either prove that f' is a PRF (for all choices of f), or prove that f' is not a PRF.

(a) $f'_k(x) := f_k(0||x)||f_k(1||x)$.

(b) $f'_k(x) := f_k(0||x)||f_k(x||1)$.

Solution

(a) Yes, f' is a PRF. Suppose for the purpose of contradiction that f' is not a PRF. Then, there exists a PPT $\mathcal{A}$ that breaks the PRF security of f'. Construct PPT $\mathcal{B}$ using $\mathcal{A}$ to break the PRF security of f as follows: $\mathcal{B}$ runs $\mathcal{A}$ internally. To answer $\mathcal{A}$'s queries for x, $\mathcal{B}$ queries the oracle (or challenger) with input $0||x$ and $1||x$ to get back y_0 and y_1. $\mathcal{B}$ then responds $y_0||y_1$ to $\mathcal{A}$. Finally, $\mathcal{B}$ outputs whatever $\mathcal{A}$ outputs.

By definition, $\mathcal{B}$ querying $f_k(\cdot)$ gives $\mathcal{A}$ access to $f'_k(\cdot)$. If $\mathcal{B}$ is querying a random function $F : \{0,1\}^n \to \{0,1\}^n$, this gives $\mathcal{A}$ access to a random function $F' : \{0,1\}^{n-1} \to \{0,1\}^{2n}$, where F' is defined as $F'(x) = F(0||x)||F(1||x)$ (this defines a one-to-one mapping from random F to random F'). Therefore,

$$\begin{aligned}\left|\Pr[\mathcal{B}^{f_k(\cdot)}(1^n) = 1] - \Pr[\mathcal{B}^{F(\cdot)}(1^n) = 1]\right| &= \left|\Pr[\mathcal{A}^{f'_k(\cdot)}(1^{n-1}) = 1] - \Pr[\mathcal{A}^{F'(\cdot)}(1^{n-1}) = 1]\right| \\ &\geq \text{nonnegl}(n)\end{aligned}$$

Hence $\mathcal{B}$ breaks the PRF security of f, contradiction.

(b) No. Construct $\mathcal{A}$ to break f': it queries for $x = 0\ldots0$ and $x = 0\ldots01$.

∎

2. Weak CPA Security

Consider a weaker definition of CPA security where in the indistinguishability experiment the adversary $\mathcal{A}$ is not given oracle access to $\mathsf{Enc}_k(\cdot)$ after choosing m_0, m_1. That is, $\mathcal{A}$ can only query $\mathsf{Enc}_k(\cdot)$ in phase 1, but not in phase 2. We call this definition weak-CPA-security. Prove that weak-CPA-security is equivalent to CPA-security (i.e., Definition 3.22 in the textbook).

Hint: Begin by showing via a hybrid argument that any $\mathcal{A}$ interacting in the usual CPA game cannot distinguish whether its phase 2 queries are answered honestly (that is, if the response to the query m is $\mathsf{Enc}_k(m)$ or an encryption of 0; $\mathsf{Enc}_k(0)$).

Solution One of the directions is easy to see. We will show that weak-CPA-security implies CPA security.

Consider an encryption scheme $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ for message space $\mathcal{M}$ that is weak-CPA secure. We will now show that it is CPA secure via a hybrid argument. Specifically, we will define a sequence of hybrids starting with the hybrid which corresponds to the CPA experiment with the bit $b = 0$ and end with a hybrid which corresponds to the CPA experiment with the bit $b = 1$. We will show that each of the intermediate hybrids are indistinguishable from the weak CPA security of the encryption scheme.

$\underline{\mathsf{Hyb}_0}$: This corresponds to the standard 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 pair of messages m_0, m_1.
3. $\underline{c^* \text{ is generated as } \mathsf{Enc}_k(m_0)}$.
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 give the next hybrid.

$\underline{\mathsf{Hyb}_1}$: This is identical to the previous hybrid except that the last query to the encryption oracle (say on a message m) in Phase-2 is answered as $\mathsf{Enc}_k(m^*)$ where m^* is an arbitrary message in $\mathcal{M}$. 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 pair of messages m_0, m_1.
3. c^* is generated as $\mathsf{Enc}_k(m_0)$.
4. The adversary $\mathcal{A}$ continues to have oracle access to $\mathsf{Enc}_k(\cdot)$ except that for the last query on a message $m \in \mathcal{M}$, $\underline{\text{we answer it as } \mathsf{Enc}_k(m^*)}$ for some arbitrary $m^* \in \mathcal{M}$. The adversary outputs b'
5. The output of the experiment is defined to be b'.

More generally, we define Hyb_j as follows:

$\underline{\mathsf{Hyb}_j}$:

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 pair of messages $m_0, m_1 \in \mathcal{M}$.
3. c^* is generated as $\mathsf{Enc}_k(m_0)$.

4. The adversary $\mathcal{A}$ continues to have oracle access to $\mathsf{Enc}_k(\cdot)$ except that for the last j queries to the encryption oracle, we answer them as independent encryptions of m^*. The adversary outputs b'

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

We now show that for any $j \in [q]$ where q is the number of queries that adversary makes in phase-2, Hyb_j is computationally indistinguishable to Hyb_{j-1}.

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

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

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

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

We will now use such an adversary $\mathcal{A}$ and the corresponding j, to construct an adversary $\mathcal{B}$ against the weak 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 two messages m_0, m_1. $\mathcal{B}$ queries its encryption oracle on m_0 and obtains the response and gives it to $\mathcal{A}$.

4. **Phase-2 oracle queries.** For every query except that last j queries that $\mathcal{A}$ makes to the encryption oracle, $\mathcal{B}$ answers them exactly as in phase-1. When the $\mathcal{A}$ asks its $(q-j+1)$-th query on a message m, $\mathcal{B}$ does the following. It makes $(j-1)$ queries to its encryption oracle on m^* and obtains the corresponding ciphertexts. It then produces (m, m^*) as the challenge messages to the weak CPA security challenger and obtains c^* as the challenge ciphertext. It returns c^* as the response to the $(q-j+1)$-th query. For the last $(j-1)$ queries, it uses the encryptions obtained on m^* to answer them.

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, then the view of $\mathcal{A}$ is identically distributed to Hyb_{j-1}. On the other hand, if c^* was an encryption of the message m^*, then the view of $\mathcal{A}$ is identically distributed to Hyb_j. Thus, if for every negligible function,

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

then, for every negligible function $\mathsf{negl}(\cdot)$

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

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

■

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

Now, notice that in Hyb_q, all the phase two queries of $\mathcal{A}$ are answered with encryptions of an arbitrary message m^*. Thus, via an identical argument as in Claim 0.1, we can show that Hyb_q is computationally indistinguishable to Hyb^* where the challenge ciphertext that was given to $\mathcal{A}$ is an encryption of m_1. Now, again via a same argument as before, we can show that Hyb^* is computationally indistinguishable to the standard CPA security game where $b = 1$. Thus, $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is standard CPA secure.

■

3. Modes of operations are not CCA-Secure

Show that the CBC and CTR modes of encryption are not CCA-secure.

Solution

1. **CBC:** Define an adversary $\mathcal{A}$ that outputs the messages $m_0 = 0^n$ and $m_1 = 1^n$ to the challenger, and receives a challenge ciphertext (IV, c). Note that for CBC mode, we have $c = F_k(IV \oplus m_b)$. The adversary then issues a decryption query for the ciphertext $(0^n, c)$. This is a valid query since $IV \neq 0^n$ with overwhelming probability.
   Now, the result for this query is $m' = F_k^{-1}(c) \oplus 0^n$ which turns out to just be IV. The adversary then computes $m' \oplus IV$ – this is either m_0 or m_1, which allows the adversary to guess the correct bit.

2. **CTR:** Define an adversary $\mathcal{A}$ that outputs the messages $m_0 = 0^n$ and $m_1 = 1^n$ to the challenger, and receives a challenge ciphertext (IV, c). Note that for CTR mode, we have $c = F_k(IV + 1) \oplus m_b$. The adversary then issues a decryption query for the ciphertext $(IV, 0^n)$. This is a valid query since $c \neq 0^n$ with overwhelming probability.
   Now, the result for this query is $m' = F_k(IV + 1) \oplus 0^n$, which turns out to just be $F_k(IV + 1)$. The adversary then computes $m' \oplus c$ – this is either m_0 or m_1, which allows the adversary to guess the correct bit.

■