CS 171: Problem Set 4

Due Date: February 29th, 2024 at 8:59pm via Gradescope

1. Negligible and Non-Negligible Functions (10 points)

Define functions $f, g : \mathbb{N} \to \mathbb{R}_{\geq 0}$, and let $g(n) = 2^{-f(n)}$.

1. Prove that if $f(n) = \omega(\log n)$, then $g(n)$ is negligible. Give a fully rigorous proof.
2. Prove that if $f(n) = O(\log n)$, then $g(n)$ is non-negligible. Give a fully rigorous proof.
3. Identify which of the following functions are negligible. There may be multiple negligible functions. No explanation is necessary for this part:
   (a) $g_1(n) = 2^{-\sqrt{n}}$
   (b) $g_2(n) = 2^{-(\log n)^2}$
   (c) $g_3(n) = 2^{-\sqrt{\log n}}$

Solution

1. **Claim 0.1** *If $f(n) = \omega(\log n)$, then $g(n)$ is negligible.*

   Proof If $f(n) = \omega(\log n)$, then for all $c > 0$, there exists an $N \in \mathbb{N}$ such that for all $n > N$,
   $$f(n) > c \log n$$
   Equivalently, that means for all $c > 0$, there exists an $N \in \mathbb{N}$ such that for all $n > N$,
   $$g(n) < 2^{-c \log n} = n^{-c}$$
   Therefore, g is negligible. ■

2. **Claim 0.2** *If $f(n) = O(\log n)$, then $g(n)$ is non-negligible.*

   Proof If $f(n) = O(\log n)$, then there exists a $c > 0$ such that there exists an $N \in \mathbb{N}$ such that for all $n > N$,
   $$f(n) \leq c \log n$$
   Equivalently, that means there exists a $c > 0$ such that there exists an $N \in \mathbb{N}$ such that for all $n > N$,
   $$g(n) \geq 2^{-c \log n} = n^{-c}$$
   Therefore, g is non-negligible. ■

3. g_1 and g_2 are negligible because $\sqrt{n} = \omega(\log n)$ and $(\log n)^2 = \omega(\log n)$. g_3 is non-negligible because $\sqrt{\log n} = O(\log n)$.

■

2. Two Versions of CPA security (10 points)

There are two common definitions of CPA security, which are given in definitions 0.3 and 0.4 below[1]. Prove that definitions 0.3 and 0.4 are equivalent, i.e. if a scheme is secure under one definition, then it is secure under the other definition.

Definition 0.3 *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme and let $\mathcal{A}$ be an adversary for the CPA security game. Define the CPA security game as follows:*

$G_{\mathcal{A},\Pi}(n)$*:*

1. *The challenger samples a key $k \leftarrow \mathsf{Gen}(1^n)$.*
2. *The adversary $\mathcal{A}$ is given input 1^n and oracle access to $\mathsf{Enc}(k,\cdot)$, and outputs a pair of messages (m_0, m_1) with $|m_0| = |m_1|$.*
3. *The challenger samples a bit $b \leftarrow \{0,1\}$, and computes the ciphertext $c \leftarrow \mathsf{Enc}(k, m_b)$. Then they give c to $\mathcal{A}$.*
4. *$\mathcal{A}$ continues to have oracle access to $\mathsf{Enc}(k,\cdot)$ and outputs a bit b'.*
5. *The output of the game is 1 if $b' = b$, and 0 otherwise.*

We say that the encryption scheme Π is CPA-secure if for all probabilistic polynomial-time (PPT) adversaries $\mathcal{A}$, there is a negligible function negl such that

$$\Pr\left[G_{\mathcal{A},\Pi}(n) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(n)$$

In definition 0.4 below, any changes from definition 0.3 are shown in red.

Definition 0.4 *Let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme and let $\mathcal{A}$ be an adversary for the CPA security game. Define the CPA security game as follows:*

$H_{\mathcal{A},\Pi}(n, b)$*:*

1. *The challenger samples a key $k \leftarrow \mathsf{Gen}(1^n)$.*
2. *The adversary $\mathcal{A}$ is given input 1^n and oracle access to $\mathsf{Enc}(k,\cdot)$, and outputs a pair of messages (m_0, m_1) with $|m_0| = |m_1|$.*
3. *The challenger computes the ciphertext $c \leftarrow \mathsf{Enc}(k, m_b)$. Then they give c to $\mathcal{A}$.*
4. *$\mathcal{A}$ continues to have oracle access to $\mathsf{Enc}(k,\cdot)$ and outputs a bit b'.*
5. *The output of the game is b'.*

We say that the encryption scheme Π is CPA-secure if for all probabilistic polynomial-time (PPT) adversaries $\mathcal{A}$, there is a negligible function negl such that

$$\left|\Pr\left[H_{\mathcal{A},\Pi}(n,0) = 1\right] - \Pr\left[H_{\mathcal{A},\Pi}(n,1) = 1\right]\right| \leq \mathsf{negl}(n)$$

[1] These are analogous to the two definitions of security for EAV security (lecture 3, slides 19-20) and PRGs (lecture 4, slides 8-9)

Solution

1. First, note that

$$\begin{aligned}\Pr\left[G_{\mathcal{A},\Pi}(n)=1\right] &= \Pr[b=0]\cdot\Pr\left[H_{\mathcal{A},\Pi}(n,0)=0\right]+\Pr[b=1]\cdot\Pr\left[H_{\mathcal{A},\Pi}(n,1)=1\right]\\ &= \frac{1}{2}\cdot\Big(1-\Pr\left[H_{\mathcal{A},\Pi}(n,0)=1\right]\Big)+\frac{1}{2}\cdot\Pr\left[H_{\mathcal{A},\Pi}(n,1)=1\right]\\ &= \frac{1}{2}+\frac{1}{2}\cdot\Big(\Pr\left[H_{\mathcal{A},\Pi}(n,1)=1\right]-\Pr\left[H_{\mathcal{A},\Pi}(n,0)=1\right]\Big)\end{aligned}$$

2. **Claim 0.5** *Definition 0.4 implies definition 0.3.*

   Proof If for all PPT adversaries $\mathcal{A}$, there exists a negligible function negl such that

$$\left|\Pr\left[H_{\mathcal{A},\Pi}(n,0)=1\right]-\Pr\left[H_{\mathcal{A},\Pi}(n,1)=1\right]\right|\leq\mathsf{negl}(n)$$

   then

$$\Pr\left[G_{\mathcal{A},\Pi}(n)=1\right]\leq\frac{1}{2}+\frac{\mathsf{negl}(n)}{2}$$

   Note that $\frac{\mathsf{negl}(n)}{2}$ is still a negligible function. Then for any PPT adversary $\mathcal{A}$, there exists a negligible function $\mathsf{negl}_1(n):=\frac{\mathsf{negl}(n)}{2}$ such that $\Pr\left[G_{\mathcal{A},\Pi}(n)=1\right]\leq\frac{1}{2}+\mathsf{negl}_1(n)$. ■

3. **Claim 0.6** *Definition 0.3 implies definition 0.4.*

   Proof

   (a) If for all PPT adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}_{\mathcal{A}}$ such that

$$\Pr\left[G_{\mathcal{A},\Pi}(n)=1\right]\leq\frac{1}{2}+\mathsf{negl}_{\mathcal{A}}(n)$$

   then

$$\Pr\left[H_{\mathcal{A},\Pi}(n,1)=1\right]-\Pr\left[H_{\mathcal{A},\Pi}(n,0)=1\right]\leq 2\cdot\mathsf{negl}_{\mathcal{A}}(n)$$

   Note that $2\cdot\mathsf{negl}_{\mathcal{A}}(n)$ is still a negligible function.

   (b) Next, we'll show that

$$\Pr\left[H_{\mathcal{A},\Pi}(n,0)=1\right]-\Pr\left[H_{\mathcal{A},\Pi}(n,1)=1\right]\leq 2\cdot\mathsf{negl}_{\mathcal{B}}(n)$$

   for some negligible function $\mathsf{negl}_{\mathcal{B}}$. Let us define a new PPT adversary $\mathcal{B}$ that runs the same algorithm as $\mathcal{A}$, except that when $\mathcal{A}$ outputs b', $\mathcal{B}$ outputs $b'\oplus 1$. Since $\mathcal{B}$ is a PPT adversary, we know that there exists a negligible function $\mathsf{negl}_{\mathcal{B}}$ such that

$$\Pr\left[H_{\mathcal{B},\Pi}(n,1)=1\right]-\Pr\left[H_{\mathcal{B},\Pi}(n,0)=1\right]\leq 2\cdot\mathsf{negl}_{\mathcal{B}}(n)$$

   Therefore

$$\begin{aligned}\Pr\left[H_{\mathcal{A},\Pi}(n,0)=1\right]-\Pr\left[H_{\mathcal{A},\Pi}(n,1)=1\right] &= \Pr\left[H_{\mathcal{B},\Pi}(n,1)=1\right]-\Pr\left[H_{\mathcal{B},\Pi}(n,0)=1\right]\\ &\leq 2\cdot\mathsf{negl}_{\mathcal{B}}(n)\end{aligned}$$

(c) Finally, let $\mathsf{negl}_1(n) = 2 \cdot \mathsf{negl}_{\mathcal{A}}(n) + 2 \cdot \mathsf{negl}_{\mathcal{B}}(n)$. Note that negl_1 is a negligible function, and $2 \cdot \mathsf{negl}_{\mathcal{A}}(n) \leq \mathsf{negl}_1(n)$, and $2 \cdot \mathsf{negl}_{\mathcal{B}}(n) \leq \mathsf{negl}_1(n)$.
We've shown that for any PPT adversary $\mathcal{A}$, there exists a negligble function negl_1 such that

$$\left| \Pr\left[H_{\mathcal{A},\Pi}(n,0) = 1\right] - \Pr\left[H_{\mathcal{A},\Pi}(n,1) = 1\right] \right| \leq \mathsf{negl}_1(n)$$

■

■

3. Feistel Network (10 points)

A Feistel network is used to construct a pseudorandom permutation F given a pseudorandom function f that is not necessarily a permutation[2]. However, if f is not pseudorandom, then F is potentially not pseudorandom either.

Consider the following three-round Feistel network given in definition 0.7 below[3].

Definition 0.7 (Three-Round Feistel Network F)

1. *Let* $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$.
2. ***Inputs:*** *Let* F *take as input a key* $k \in \{0,1\}^{3n}$ *and an input* $x \in \{0,1\}^{2n}$*, which are parsed as:*

$$k = (k^1, k^2, k^3) \in \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n$$
$$x = (L_0, R_0) \in \{0,1\}^n \times \{0,1\}^n$$

3. ***Computation:***
   (a) F *computes* $L_1 := R_0$ *and* $R_1 := L_0 \oplus f(k^1, R_0)$.
   (b) F *computes* $L_2 := R_1$ *and* $R_2 := L_1 \oplus f(k^2, R_1)$.
   (c) F *computes* $L_3 := R_2$ *and* $R_3 := L_2 \oplus f(k^3, R_2)$.
   (d) F *outputs* (L_3, R_3).

Suppose that there was a flaw in the design of f so that for all keys k and all inputs x, the first bit of $f(k, x)$ equals the first bit of x. Show that there exists some efficient adversary $\mathcal{A}$ that can break the pseudorandom permutation security of F *by making only a single query to* F.

Solution

1. Let a_0 be the first bit of L_0, and let b_0 be the first bit of R_0. Let $a_1, a_2, a_3, b_1, b_2, b_3$ be defined analogously. Then these bits will have the following values:

$$(a_1, b_1) = (b_0, (a_0 \oplus b_0))$$
$$(a_2, b_2) = ((a_0 \oplus b_0), a_0)$$
$$(a_3, b_3) = (a_0, b_0)$$

2. In other words, two specific output bits of F, (a_3, b_3), are equal to two specific input bits of F, (a_0, b_0). On any given input, the probability that a uniformly random permutation will produce an output pair with this property is approximately $1/4$.

3. Now we can construct our distinguisher $\mathcal{A}$ that breaks the pseudorandom permutation security of F. A will query F on an arbitrary input and check whether $(a_3, b_3) = (a_0, b_0)$. If so, $\mathcal{A}$ ouputs 1; if not, $\mathcal{A}$ outputs 0.

[2]For more details, see Katz & Lindell, 3rd edition, sections 7.2.2 and 8.6.
[3]This definition is adapted from Katz & Lindell, 3rd edition, construction 8.23.

4. The distinguishing advantage of $\mathcal{A}$ is approximately $1 - 1/4 = .75$, which is non-negligible. Therefore, $\mathcal{A}$ breaks the PRP security of F.

■