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) $g_3(n) = 2^{-\sqrt{\log n}}$

2. Two Versions of CPA security (10 points)

There are two common definitions of CPA security, which are given in definitions 0.1 and 0.2 below[1]. Prove that definitions 0.1 and 0.2 are equivalent, i.e. if a scheme is secure under one definition, then it is secure under the other definition.

Definition 0.1 *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:*

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

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 0.2 below, any changes from definition 0.1 are shown in red.

Definition 0.2 *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:*

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

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)

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 0.3 below[3].

Definition 0.3 (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*.

[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.