Final
Your name here

Due on 11:59pm Dec. 19th.
Solution must be typed, preferably using LaTeX.
It should be submitted via email to rachel.lin@cs.ucsb.edu.

You must complete this final on your own. In particular, you cannot collaborate with anyone else, nor reference public resources. The only materials you can use during solving the final are your own notes and homeworks, homework solutions, slides, and scribe notes distributed and created during this class, and lecture notes and textbooks linked to in the course webpage.

Part 1 — Basics

a) [6 pts] Consider the group \( \mathbb{Z}^*_17 \) with multiplication modulo 17. Compute the following:

(i) \( 9^{-1} \mod 17 \),
(ii) \( 7^{-1} \mod 17 \),
(iii) \( 7^{16} \mod 17 \)

Justify your answers!

b) [9 pts] Which ones of the following functions are negligible? Justify your answers!

(i) \( f_1(n) = 2^{-200\log(n)} \),
(ii) \( f_2(n) = 2^{-\sqrt{n}} \),
(iii) \( f_3(n) = 1/\log(n) \).

In particular, to show that a function \( f(n) \) is negligible, prove that for all constants \( c \geq 1 \) there exists some value \( n_0 = n_0(c) \) such that \( f(n) \leq \frac{1}{n^c} \) for all \( n \geq n_0 \). To show that a function is not negligible, give a \( c \geq 1 \) such that \( f(n) \geq \frac{1}{n^c} \) for infinitely many values of \( n \).

c) [4 pts] Let \( \varepsilon_1, \varepsilon_2, \varepsilon_3, \ldots \) be an infinitely long sequence of negligible functions. Define the following function \( \mu \):

\[ \mu(n) = \sum_{i \in \mathbb{N}} \varepsilon_i(n) \]

Is \( \mu \) necessarily a negligible function? (i.e., Is \( \mu \) negligible no matter which negligible functions \( \varepsilon_1, \varepsilon_2, \ldots \) are?) Justify your answer!

Part 2 — One-Way Function

a) [4 pts] Let \( f : \{0, 1\}^k \to \{0, 1\}^k \) be an efficiently computable permutation, i.e., \( f \) is one-to-one.

Now, recall that a hardcore predicate \( P : \{0, 1\}^k \to \{0, 1\} \) is an efficiently computable function such that for every PPT adversary \( A \) there exists a negligible function \( \varepsilon = \varepsilon(k) \) such that \( A \) guesses \( P(x) \) given \( f(x) \) with probability at most \( \frac{1+\varepsilon(k)}{2} \).

Prove that if \( f \) has a hardcore predicate \( P \), then \( f \) is one-way.
**Hint:** Given an inverter for \( f \) with non-negligible success probability, recall that you need to build an adversary predicting \( P(x) \) from \( f(x) \) with probability non-negligibly higher than \( 1/2 \). Make sure your strategy always performs better than with probability \( 1/2 \).

b) **[2 pts]** Does the above statement hold true if \( f \) is not necessarily a permutation, but can be an arbitrary function?

---

**Part 3 — Pseudo-Random Functions and Generators** (19 points)

Let \( G : \{0,1\}^k \rightarrow \{0,1\}^{2k} \) be a length-doubling PRG, and let us explicitly write \( G(x) = G_0(x)\|G_1(x) \) for two maps \( G_0, G_1 : \{0,1\}^k \rightarrow \{0,1\}^k \) and every input seed \( x \) (i.e., \( G_0 \) outputs the first \( k \) bits of the output of \( G \) and \( G_1 \) outputs the second \( k \) bits). In class, we presented a construction of a PRF from a length-doubling PRG using a tree—this is called the GGM construction.

In this exercise, we consider the following construction of a keyed function \( F : \{0,1\}^k \times \{0,1\}^\ast \rightarrow \{0,1\}^k \) which is a variant of the GGM construction, adapted to the case of messages of arbitrary length. For key \( K \in \{0,1\}^k \) and input \( M \in \{0,1\}^\ast \), the construction parses \( M \) as \( M = m_1, m_2, \ldots, m_{|M|} \), where \( m_i \in \{0,1\} \) for all \( i = 1, \ldots, |M| \). (Recall that \( |M| \) denotes the bit length of the message \( M \).) Then, it sets \( y_0 \leftarrow K \), and for all \( i = 1, \ldots, |M| \) computes \( y_i \leftarrow G_{m_i}(y_{i-1}) \). Finally, it outputs \( y_{|M|} = F(K, M) \).

The security of a PRF for arbitrary-length inputs, is defined as in the case of fixed-length inputs (e.g., \( n \) bits for some given \( n \)), except that we compare the PRF function \( F_k \) with a random function \( RF_{*,k} \) that takes an input of arbitrary length and maps it to a random \( k \) bit string. The PRF function is secure if no non-uniform PPT distinguishers can distinguish an interaction with \( F_k \) with \( k \) sampled at random, from an interaction with \( RF_{*,k} \).

a) **[4 pts]** Prove that \( F \) described above is not a PRF. In particular, analyze your attack and show that it achieves non-negligible advantage in distinguishing \( F_k \) with a randomly sampled \( k \) from \( RF_{*,k} \).

**Hint:** Find a non-trivial yet easily verifiable relation between outputs \( Y = F_K(M) \) and \( Y' = F_K(M') \) for two carefully chosen messages \( M \) and \( M' \) which is not satisfied if \( Y, Y' \) are random and independent, except with negligible probability.

b) **[3 pts]** In class, we showed that a MAC scheme can be constructed from a keyed function. Show that the MAC scheme instantiated using the above keyed function is not unforgeable. In particular, analyze your attack and show that it breaks unforgeability.

c) **[5 pts]** Present a fix to the above construction that prevents the attacks shown in a). Argue (informally!) that the construction is now a secure PRF.

**Hint:** Encode the input \( M \) so that the above attack is not possible.

d) **[3 pts]** How can we make the above construction more efficient if we are given a PRG \( G : \{0,1\}^k \rightarrow \{0,1\}^{2^r \cdot k}? \)
Hint: The construction will process now $r$ bits for every PRG invocation.

e) [4 pts] Assume that we are given now a block cipher $E : \{0, 1\}^k \times \{0, 1\}^k \rightarrow \{0, 1\}^k$ which is assumed to be a secure PRF, how can we modify the construction idea given in b-c) to obtain a new construction using $E$ instead of $G$ and processing $k$ bits per invocation of $E$?

Hint: Think of using $E$ to build a PRG which expands from $k$ bits to $2^k \cdot k$ bits. (Of course, such a “PRG” would not be efficient, but we will not need to compute the whole output when evaluating the construction and it helps to conceptually think this way to obtain the right approach.)

Part 4 — Public Key Encryption (PKE) (6 points)

In class we saw the RSA assumption. This exercise considers a different assumption:

**Assumption 1.** There is a generation process $Gen'$ that samples a multiplicative group $G$ with prime order $q$ and a generator $g$ of $G$, such that, the following ensembles are indistinguishable.

$$\{(G, q, g) \leftarrow Gen'(1^n), x \leftarrow \mathbb{Z}_q, y \leftarrow \mathbb{Z}_q : G, q, g, g^x, g^y, g^{xy}\} \sim \{(G, q, g) \leftarrow Gen'(1^n), x \leftarrow \mathbb{Z}_q, y \leftarrow \mathbb{Z}_q, z \leftarrow \mathbb{Z}_q : G, q, g, g^x, g^y, g^z\} \quad \forall n \in \mathbb{N}$$

**Note:** A group $G$ with prime order $q$ is a group with $q$ elements, and a generator $g$ of $G$ is simply an element in $G$ with the property that $\{g^0, g^1, \ldots, g^{q-1}\} = G$. In other words, powers of $g$ “generates” the whole group. The concrete choice of $G$, $q$, and $g$ is important for the assumption to hold (as captured inside $Gen'$), but not particularly important for constructing a bit public key encryption from this assumption, which is the focus of this exercise.

Consider the following construction of a bit public key encryption $\Pi = (Gen, Enc, Dec)$ scheme based on the above assumption.

- $Gen(1^n)$: Sample $(G, q, g) \leftarrow Gen'(1^n)$ and $x \leftarrow \mathbb{Z}_q$; compute $h = g^x$; and set $pk = (G, q, g, h)$ and $sk = x$.

- $Enc(pk, b)$: Parse $pk = (G, q, g, h)$; sample random $y \leftarrow \mathbb{Z}_q$; compute $\alpha = g^y$ and $\beta = h^y$; output ciphertext $c = (\alpha, \beta \cdot g^b)$.

- $Dec(sk, b)$: Parse $sk = x$ and $c = (\alpha, \gamma)$; compute $\beta = \alpha^x$ and $g^b = \gamma \cdot \beta^{-1}$; output $0$ if $g^b = 1$ and $1$ if $g^b = g$.

Prove that $\Pi$ is multi-message secure under Assumption 1. (You can directly use theorems proven in class.)
Part 5 — Digital Signatures

Task (a) — Strong one-time signature scheme

A strong one-time signature scheme \((\text{Gen}, \text{Sign}, \text{Ver})\) satisfies the following (informally): A pair of keys \((pk, sk) \overset{\$}{\leftarrow} \text{Gen}(1^n)\) is sampled; an adversary \(A\) receiving \(pk\) requests to see a signature \(\sigma\) on a message \(m\) of its choice; it is infeasible for \(A\) to then output \((m', \sigma')\) satisfying that \(\sigma'\) is a valid signature for \(m'\) and \((m, \sigma) \neq (m', \sigma')\) (except with negligible probability).

Note that different from one-time security introduced in class, for strong one-time security, \(A\) is allowed to output \(m' = m\), as long as \(\sigma' \neq \sigma\) (there could be multiple signatures for the same message).

a) [3 pts] Assuming the existence of one-way functions, show a one-way function \(f\) for which Lamports scheme is not a strong one-time signature scheme.

b) [3 pts] Construct a strong one-time signature scheme using any assumption we have seen in class. (Hint: Use a particular one-way function in Lamports scheme.)

Task (b) — Relation between signature scheme and one-way function

Prove that the existence of a one-time signature scheme for 1-bit messages implies the existence of one-way functions.