Final Exam

Name:

SID:

- **You may consult at most *3 double-sided sheets of handwritten notes*. Apart from that, you may not look at books, notes, etc. Calculators, phones, computers, and other electronic devices are NOT permitted for looking up content. However, you may use an electronic device such as a tablet for writing your answers.**
- **You have 170 minutes to complete the exam. For DSP students, you may have $1.5\times170 = 255$ minutes or $2\times170 = 340$ minutes, depending on your accommodation.**
- **The instructors will not be answering questions during the exam. If you feel that something is unclear, please write a note in your answer.**

1 Multiple Choice (25 Points)

In the multiple choice section, no explanations are needed for your answers. Please mark your answers clearly.

In a question with multiple correct answers, your score will be proportional to the number of correct answers selected minus the number of incorrect answers selected.

1. Let f and g be functions that map $\mathbb{N} \to \mathbb{R}_{\geq 0}$. Let f be a negligible function and let g be a non-negligible function. Which of the following functions must be non-negligible? There may be several.

   ◯ $A(n) = f(n)^2 + g(n)$

   ◯ $B(n) = |g(n) - f(n)|$

   ◯ $C(n) = \frac{1}{n} \cdot g(n)$

   ◯ $D(n) = g(n) \cdot f(n)$

   ◯ $E(n) = g(n) \cdot g(n)$

   ◯ $F(n) = g(n) \cdot g(n+1)$

   Solution: A, B, C, E

2. Suppose CDH is hard for some cryptographic group. Then, which of the following statements must be true? There may be several.

   ◯ A. PRGs exist.

   ◯ B. DBDH is hard for some cryptographic group.

   ◯ C. DDH is easy for some cryptographic group.

   ◯ D. Discrete log is hard for some cryptographic group.

   Solution: A, D

3. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map for which the decisional bilinear Diffie-Hellman (DBDH) problem is computationally hard. Which of the following problems are also computationally hard?

Name:

◯ A. Decisional Diffie Hellman in $\mathbb{G}$.

◯ B. Computational Diffie Hellman in $\mathbb{G}$.

◯ C. Discrete Log in $\mathbb{G}$.

◯ D. Discrete Log in $\mathbb{G}_T$.

Solution: B, C, D

4. Which of the following is a secure way to construct an authenticated encryption scheme:

◯ A. Encrypt and MAC

◯ B. Encrypt then MAC

◯ C. MAC then Encrypt

◯ D. MAC, then encrypt, and then MAC again

Solution: B, D

5. An Identity Based Encryption scheme can be used to construct which of the following primitives?

◯ A. One-way functions

◯ B. One-way permutations

◯ C. Digital signatures

◯ D. CCA-secure public key encryption

Solution: A, C, D

2 CCA Security

2.1 A Scheme For n-Bit Messages (20 Points)

Consider the following secret-key encryption scheme with message space $\mathcal{M} = \{0,1\}^n$.

Let $F : \{0,1\}^n \times \{0,1\}^{2n} \to \{0,1\}^{2n}$ be a strong pseudorandom permutation.

1. $\mathsf{Gen}(1^n)$: Sample $k \leftarrow \{0,1\}^n$ and output k.
2. $\mathsf{Enc}(k, m)$: Sample $r \leftarrow \{0,1\}^n$. Compute and output
$$c = F_k(m \parallel r)$$
3. $\mathsf{Dec}(k, c)$: Compute
$$m' \parallel r' = F_k^{-1}(c)$$
where $m', r' \in \{0,1\}^n$. Then output m'.

Question 1: Give the security definition for a strong PRP.

Solution: Intuitively, a strong PRP is a permutation that appears random to an adversary who gets query access to the permutation *and its inverse*.

Definition 2.1 (Strong PRP) *Let $F : \{0,1\}^n \times \{0,1\}^{2n} \to \{0,1\}^{2n}$. For any $k \leftarrow \{0,1\}^n$, let $F_k(\cdot)$ and $F_k^{-1}(\cdot)$ be efficiently computable.*

*F is a **strong pseudorandom permutation** if for all probabilistic polynomial-time distinguishers D, there exists a negligible function negl such that*

$$\left| \Pr\left[D^{F_k(\cdot), F_k^{-1}(\cdot)}(1^n) \to 1\right] - \Pr\left[D^{f(\cdot), f^{-1}(\cdot)}(1^n) \to 1\right] \right| \leq \mathsf{negl}(n)$$

The first probability is taken over the randomness of sampling $k \leftarrow \{0,1\}^n$ and the randomness of D. The second probability is taken over the randomness of sampling f uniformly at random from the set of all permutations mapping $\{0,1\}^{2n} \to \{0,1\}^{2n}$, as well as the randomness of D.

Question 2: Prove that $\Pi := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is CCA2-secure.

Solution:

Theorem 2.1 *$\Pi := (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is CCA2-secure.*

Proof:

Name:

1. Assume toward contradiction that there is an adversary $\mathcal{A}$ that breaks CCA2-security of Π. Then we will use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that breaks the strong PRP security of F. This is a contradiction because F is satisfies strong PRP security. Therefore, the initial assumption was false, and in fact Π is CCA2-secure.

2. $\mathcal{B}$ will play in the strong PRP game for F and simulate the CCA2 security game.

   In the strong PRP game, the challenger samples a PRP key $k \leftarrow \{0,1\}^n$ and samples a permutation $f : \{0,1\}^{2n} \to \{0,1\}^{2n}$ uniformly at random. Then the challenger gives $\mathcal{B}$ query access to two oracles $(\mathcal{O}, \mathcal{O}^{-1})$, which are either $\left(F_k(\cdot), F_k^{-1}(\cdot)\right)$ or $\left(f(\cdot), f^{-1}(\cdot)\right)$.

   Construction of $\mathcal{B}$:
   $\mathcal{B}$ runs $\mathcal{A}(1^n)$ and simulates the CCA2 security game.

   (a) *Encryption Queries:* When $\mathcal{A}$ outputs a message $m \in \{0,1\}^n$ to be encrypted:
      i. $\mathcal{B}$ samples $r \leftarrow \{0,1\}^n$.
      ii. $\mathcal{B}$ computes $c = \mathcal{O}(m \parallel r)$ and sends c to $\mathcal{A}$.

   (b) *Challenge query:* $\mathcal{A}$ outputs two messages $m^{(0)}, m^{(1)} \in \{0,1\}^n$.
      i. $\mathcal{B}$ samples $b \leftarrow \{0,1\}$ and $r^* \leftarrow \{0,1\}^n$.
      ii. $\mathcal{B}$ computes $c^* = \mathcal{O}(m^{(b)} \parallel r^*)$ and sends c^* to $\mathcal{A}$.

   (c) *Decryption Queries:* When $\mathcal{A}$ outputs a ciphertext $c \in \{0,1\}^{2n}$ to be decrypted:
      i. $\mathcal{B}$ checks whether $c = c^*$. If so, $\mathcal{B}$ outputs $\perp$. If not, $\mathcal{B}$ continues.
      ii. $\mathcal{B}$ computes $(m' \parallel r') = \mathcal{O}^{-1}(c)$, where $m', r' \in \{0,1\}^n$. Then $\mathcal{B}$ sends m' to $\mathcal{A}$.

   (d) In the end $\mathcal{A}$ outputs a bit b', and $\mathcal{B}$ checks whether $b = b'$. If so, $\mathcal{B}$ outputs 1. If not, $\mathcal{B}$ outputs 0.

3. *Pseudorandom Case:* If $(\mathcal{O}, \mathcal{O}^{-1}) = \left(F_k(\cdot), F_k^{-1}(\cdot)\right)$, then $\mathcal{B}$ simulates the CCA2 security game exactly. In particular,

   $$\Pr\left[D^{F_k(\cdot), F_k^{-1}(\cdot)}(1^n) \to 1\right] = \Pr[b' = b] = \Pr[\mathcal{A} \text{ wins the strong PRP security game }] \geq \frac{1}{2} + \mathsf{non\text{-}negl}(n)$$

4. *Truly Random Case:* $(\mathcal{O}, \mathcal{O}^{-1}) = \left(f(\cdot), f^{-1}(\cdot)\right)$. With $1 - \mathsf{negl}(n)$ probability over the randomness of r and f: $(m^{(b)} || r^*)$ and c^* are not queried by $\mathcal{A}$ during an encryption or decryption query. Then c^* is a uniformly random string that is independent of all the queries made by $\mathcal{A}$, so c^* reveals no information about b. Therefore:

   $$\Pr\left[D^{f(\cdot), f^{-1}(\cdot)}(1^n) \to 1\right] = \frac{1}{2} \pm \mathsf{negl}(n)$$

5. In summary:

   $$\left| \Pr\left[D^{F_k(\cdot), F_k^{-1}(\cdot)}(1^n) \to 1\right] - \Pr\left[D^{f(\cdot), f^{-1}(\cdot)}(1^n) \to 1\right] \right| \geq \mathsf{non\text{-}negl}(n) - \mathsf{negl}(n)$$
   $$\geq \mathsf{non\text{-}negl}(n)$$

   Therefore, $\mathcal{B}$ breaks the strong PRP security of F.

■

Question 3: Is $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ necessarily CPA-secure? No proof is needed.

◯ Yes ◯ No

Solution: Yes, Π is CPA-secure because CCA2-security implies CPA-security.

Name:

2.2 Concatenating The Base Scheme (15 Points)

Now we will construct a candidate encryption scheme $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ for tn-bit messages, where $t = \mathsf{poly}(n)$.

As before, let $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a CCA2-secure secret-key encryption scheme for n-bit messages. Then, for a message $m \in \{0,1\}^{tn}$, let $m = (m_1 \parallel \cdots \parallel m_t)$, where for each $i \in [t]$, $m_i \in \{0,1\}^n$. Finally, $\Pi' = (\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ is defined as follows:

1. $\mathsf{Gen}'(1^n) = \mathsf{Gen}(1^n)$
2. $\mathsf{Enc}'(k, m)$: Output
$$c = (c_1 \parallel \cdots \parallel c_t) = \big(\mathsf{Enc}(k, m_1) \parallel \cdots \parallel \mathsf{Enc}(k, m_t)\big)$$
3. $\mathsf{Dec}'(\mathsf{sk}, c) = \mathsf{Dec}(k, c_1) \parallel \cdots \parallel \mathsf{Dec}(k, c_t)$

Question 4: Is Π' neccessarily CPA-secure? No proof is needed.

◯ Yes ◯ No

Solution: Yes

Question 5: Is Π' necessarily CCA2-secure?

◯ Yes ◯ No

Prove your answer.

Solution: Π' is not CCA2-secure. We will construct an adversary $\mathcal{A}$ that breaks CCA2-security.

Construction of $\mathcal{A}$:

1. Output challenge messages $m^{(0)} = 0^{tn}$ and $m^{(1)} = 1^{tn}$ and then receive the challenge ciphertext
$$c^* = (c_1^* \parallel \cdots \parallel c_t^*) = \mathsf{Enc}'(k, m^{(b)})$$
2. Choose an arbitrary message $m^{(2)} \in \{0,1\}^{tn}$ such that $m_t^{(2)} \notin \{0^n, 1^n\}$. Then make an encryption query on $m^{(2)}$ and receive
$$c^{(2)} = (c_1^{(2)} \parallel \cdots \parallel c_t^{(2)}) = \mathsf{Enc}'(k, m^{(2)})$$
3. Make a decryption query on the following ciphertext:
$$c^{(3)} := c_1^* \parallel \cdots \parallel c_{t-1}^* \parallel c_t^{(2)}$$
and receive $m^{(3)} = \mathsf{Dec}'\left(k, c^{(3)}\right)$.

4. If $m_1^{(3)} = 0^n$, then output $b' = 0$. Otherwise output $b' = 1$.

Analysis:

1. With overwhelming probability, $c_t^* \neq c_t^{(2)}$. This is because $m_t^{(b)} \neq m_t^{(2)}$ and Π satisfies correctness.
2. If $c_t^* \neq c_t^{(2)}$, then $c^* \neq c^{(2)}$, so $c^{(2)}$ is a valid decryption query, and the challenger will decrypt it as desired.
3. $m_1^{(3)} = m_1^{(b)}$. So if $b = 0$, then $m_1^{(3)} = 0^n$, and if $b = 1$, then $m_1^{(3)} = 1^n$. Therefore, $\mathcal{A}$ correctly guesses the value of b.

Name:

3 One-Way Functions (25 Points)

Let $f : \{0,1\}^n \to \{0,1\}^n$ be a one-way function. Let $x = (x_L, x_R) \in \{0,1\}^n \times \{0,1\}^n$ be a generic input. Now consider the following functions constructed from f:

1. $g_1(x) = f(x_L) \parallel x_R$
2. $g_2(x) = f(x_L) \oplus x_R$
3. $g_3(x) = f(x_L) \parallel f(x_R)$
4. $g_4(x) = f(x_L) \oplus f(x_R)$

Question: For each function (g_1, g_2, g_3, g_4), indicate whether it is necessarily a one-way function, and prove your answer.

As a guideline, your answer for each g_i should do one of the following:

- Prove that if f is a OWF, then g_i is a OWF.
- Construct a OWF f and an adversary $\mathcal{A}$ such that when g_i is constructed using this choice of f, $\mathcal{A}$ can break the OWF security of g_i.

Solution:

Claim 3.1 *g_1 is a OWF.*

Proof: Assume for the sake of contradiction that g_1 is not a OWF. We will use an adversary that breaks the one-wayness of g_1 to break the one-wayness of f, resulting in a contradiction.

Let $\mathcal{A}$ be the adversary that inverts g_1 with non-negligible probability. We define adversary $\mathcal{B}$ attempting to invert f as follows:

On input $(1^n, y)$:

1. Sample $x_R^* \leftarrow \{0,1\}^n$.
2. Run $\mathcal{A}$ on input $(1^{2n}, y \parallel x_R^*)$.
3. Receive $x^* = x \parallel x_R^*$ from $\mathcal{A}$ and return x.

Analysis. Observe that if $\mathcal{A}$ inverts $y \parallel x_R^*$ under g_1, then it returns $x^* = x \parallel x_R^*$ such that $f(x) = y$. In this case, $\mathcal{B}$ successfully inverts f. Then,

$$\Pr[\mathcal{B} \text{ inverts } f] \geq \Pr[\mathcal{A} \text{ inverts } g_1] = \mathsf{non\text{-}negl}(n).$$

This contradictions our original assumption, so in fact, g_1 is a one-way function. ■

Claim 3.2 *g_2 is not a OWF.*

Proof: We will construct an adversary $\mathcal{A}$ that inverts g_2. $\mathcal{A}$ works for any choice of f.

$\mathcal{A}$ works as follows. On input $(1^{2n}, y)$:

1. $\mathcal{A}$ chooses $x_L = 0^n$ and $x_R = f(0^n) \oplus y$.
2. $\mathcal{A}$ outputs $x = (x_L, x_R)$.

Analysis. Observe that

$$g_2(x) = f(x_L) \oplus x_R = f(0^n) \oplus f(0^n) \oplus y = y$$

So $\mathcal{A}$ successfully inverts g_2 on any input. ■

Claim 3.3 *g_3 is a OWF.*

Proof: Assume for the sake of contradiction that g_3 is not a OWF. We will use an adversary that breaks the one-wayness of g_3 to break the one-wayness of f, resulting in a contradiction.

Let $\mathcal{A}$ be the adversary that inverts g_3 with non-negligible probability. We define adversary $\mathcal{B}$ attempting to invert f as follows:

On input $(1^n, y)$:

1. Sample $x_R^* \leftarrow \{0,1\}^n$.
2. Run $\mathcal{A}$ on input $(1^{2n}, y \parallel f(x_R^*))$.
3. Receive $x' = x'_L \parallel x'_R$ from $\mathcal{A}$ and return x'_L.

Analysis. Observe that if $\mathcal{A}$ inverts $y \parallel f(x_R^*)$ under g_3, then it returns $x' = x'_L \parallel x'_R$ such that $f(x'_L) = y$. In this case, $\mathcal{B}$ will successfully invert f. Then,

$$\Pr[\mathcal{B} \text{ inverts } f] \geq \Pr[\mathcal{A} \text{ inverts } g_3] = \mathsf{non\text{-}negl}(n).$$

This contradictions our original assumption, implying that g_3 is indeed a one-way functions. ■

Claim 3.4 *g_4 is not necessarily a OWF.*

Proof:

1. First, let us construct a OWF f that will make g_4 insecure. Let us start with a generic OWF $h : \{0,1\}^n \to \{0,1\}^{n-2}$. Next, denote the first $n/2$ bits of x by x_L and the second $n/2$ bits by x_R. Now we will construct f as follows:

$$f(x) = \begin{cases} x & x_L = 0^{n/2} \text{ or } x_R = 0^{n/2} \\ 1 \parallel h(x) \parallel 1 & \text{else} \end{cases}$$

2. We claim that f is a OWF. This is because the first case occurs with negligible probability:

$$\Pr_x[x_L = 0^{n/2} \text{ or } x_R = 0^{n/2}] \leq 2 \cdot 2^{-n/2} = \mathsf{negl}(n)$$

In the second case ($x_L \neq 0^{n/2}$ and $x_R \neq 0^{n/2}$), f outputs $h(x)$ with two extra bits, so inverting f is equivalent to inverting h. This is just a proof sketch that f is a OWF, and we will omit the full proof.

3. Now let g_4 be instantiated with this choice of f. Then given $y = g_4(x)$, it is easy to find a preimage of y. First, let y_L be the first $n/2$ bits of y and y_R be the second $n/2$ bits. Then, choose

$$x' = 0^{n/2} \parallel y_R \parallel y_L \parallel 0^{n/2}$$

x' is a preimage of y since

$$\begin{aligned} g_4(0^{n/2} \parallel y_R \parallel y_L \parallel 0^{n/2}) &= f(0^{n/2} \parallel y_R) \oplus f(y_L \parallel 0^{n/2}) \\ &= (0^{n/2} \parallel y_R) \oplus (y_L \parallel 0^{n/2}) \\ &= y_L \parallel y_R \\ &= y. \end{aligned}$$

■

4 Derandomizing Signatures (25 Points)

We will show how to convert a randomized signature scheme into a deterministic signature scheme by replacing the random input with a PRF.

Let $\mathcal{S} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ be a secure signature scheme with message space $\mathcal{M} = \{0,1\}^n$. In this scheme, Sign is randomized and takes a random string $r \leftarrow \{0,1\}^n$. We write $\mathsf{Sign}(\mathsf{sk}, m; r)$ to make the random input explicit.

Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a secure PRF.

Consider the following signature scheme $\mathcal{S}' = (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Verify}')$:

1. $\mathsf{Gen}'(1^n)$:
   (a) Sample $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)$.
   (b) Sample $k \leftarrow \{0,1\}^n$.
   (c) Output $\mathsf{pk}' = \mathsf{pk}$ and $\mathsf{sk}' = (\mathsf{sk}, k)$.
2. $\mathsf{Sign}'(\mathsf{sk}, m)$: Output $\sigma = \mathsf{Sign}\big(\mathsf{sk}, m; F(k, m)\big)$.
3. $\mathsf{Verify}'(\mathsf{pk}, m, \sigma) = \mathsf{Verify}(\mathsf{pk}, m, \sigma)$.

Note that Sign' is deterministic.

Question: Prove that $\mathcal{S}'$ is a secure signature scheme.

Solution: The proof is similar to the answer to homework 5, question 3.

Name:

5 A Variation on El Gamal Encryption (20 points)

We will examine a variation on El Gamal encryption and prove that this version is also CPA-secure.

Consider the following candidate public key encryption scheme with message space $\mathcal{M} = \{0, 1\}$. Let $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ be a cryptographic group of prime order q for which DDH is hard.

1. $\mathsf{Gen}(1^n)$:
   (a) Sample $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$.
   (b) Sample $x \leftarrow \mathbb{Z}_q$, and compute $h = g^x$.
   (c) Output $\mathsf{pk} = (\mathbb{G}, q, g, h)$ and $\mathsf{sk} = (\mathsf{pk}, x)$.
2. $\mathsf{Enc}(\mathsf{pk}, m)$:
   - If $m = 0$, then sample $y \leftarrow \mathbb{Z}_q$ and output
   $$c = (c_1, c_2) = (g^y, h^y)$$
   - If $m = 1$, then sample $y, z \leftarrow \mathbb{Z}_q$ independently. Next, output
   $$c = (c_1, c_2) = (g^y, g^z)$$
3. $\mathsf{Dec}(\mathsf{sk}, c)$: Output 0 if $c_1^x = c_2$ and output 1 otherwise.

Question 1: Fill in $\mathsf{Dec}(\mathsf{sk}, c)$ above so that it is correct (except with negligible probability in n) and it runs in probabilistic polynomial time.

Question 2: Prove that $\mathsf{Dec}(\mathsf{sk}, c)$ is correct, except with negligible probability in n.

Solution:

Claim 5.1 *The encryption scheme is correct.*

Proof: When $m = 0$,

$$c_1^x = (g^y)^x = (g^x)^y = h^y = c_2.$$

So decryption will output 0 with certainty.

When $m = 1$,

$$c_1^x = g^{yx}$$

Decryption will correctly output 1 unless $z = xy \mod q$. The probability that $z = xy \mod q$ occurs is $1/q$ which is $\mathsf{negl}(n)$. ■

Question 3: Prove that $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is CPA-secure.

Solution:

Claim 5.2 *The encryption scheme is CPA-secure.*

Proof:
Overview. Observe that when c is an encryption of 0, then $(h, c_1, c_2) = (g^x, g^y, g^{xy})$. When c is an encryption of 1, then $(h, c_1, c_2) = (g^x, g^y, g^z)$, where x, y, z are independent and uniformly random elements of $\mathbb{Z}_q$. The DDH problem asks an adversary to distinguish between these two types of tuples.

We will prove that given an adversary $\mathcal{A}$ that breaks the security of the modified El Gamal scheme, we can construct a different adversary $\mathcal{B}$ that will distinguish between DDH triples, breaking the DDH assumption. Define $\mathcal{B}$ as follows:

1. $\mathcal{B}$ gets $(\mathbb{G}, q, g, h, c_1, c_2)$ as input.
2. $\mathcal{B}$ runs $\mathcal{A}$ on $\mathsf{pk} = (\mathbb{G}, q, g, h)$ until $\mathcal{A}$ outputs two challenge messages $m_0 = 0$ and $m_1 = 1$ (This is without loss of generality because the message-space is $\{0, 1\}$).
3. $\mathcal{B}$ gives the ciphertext (c_1, c_2) to $\mathcal{A}$ and outputs whatever bit b' is outputted by $\mathcal{A}$.

Analysis. Consider the case where $\mathcal{B}$'s inputs are generated by choosing uniformly random $x, y \in \mathbb{Z}_q$ and setting $h = g^x$, $c_1 = g^y$, and $c_2 = g^{xy}$. From $\mathcal{A}$'s perspective, the experiment is distributed identically to the CPA-security game when $b = 0$. It follows that:

$$\Pr[\mathcal{B} \text{ wins the DDH game given } (g^x, g^y, g^{xy})] = \Pr[\mathcal{A} \text{ wins the CPA-security game} \mid b = 0].$$

Now consider the other case, where $\mathcal{B}$'s inputs are generated by running $\mathcal{G}(1^n)$, choosing uniformly random $x, y, z \in \mathbb{Z}_q$, and setting $h = g^x, c_1 = g^y$, and $c_2 = g^z$. From $\mathcal{A}$'s perspective, this is distributed identically to the CPA-security game conditioned on $b = 1$. Consequently,

$$\Pr[\mathcal{B} \text{ wins the DDH game given } (g^x, g^y, g^z)] = \Pr[\mathcal{A} \text{ wins the CPA-security game} \mid b = 1].$$

Putting these two equations together:

$$\begin{aligned}
\Pr[\mathcal{B} \text{ wins the DDH game}] &= \frac{1}{2} \cdot \Pr[\mathcal{B} \text{ wins the DDH game given } (g^x, g^y, g^{xy})] \\
&\quad + \frac{1}{2} \cdot \Pr[\mathcal{B} \text{ wins the DDH game given } (g^x, g^y, g^z)] \\
&= \frac{1}{2} \cdot \Pr[\mathcal{A} \text{ wins the CPA-security game} \mid b = 0] \\
&\quad + \frac{1}{2} \cdot \Pr[\mathcal{A} \text{ wins the CPA-security game} \mid b = 1] \\
&= \Pr[\mathcal{A} \text{ wins the CPA-security game}] \\
&= \frac{1}{2} + \mathsf{non\text{-}negl}(n).
\end{aligned}$$

This contradicts the hardness of the DDH problem in $\mathcal{G}$. Therefore, our initial assumption was false, and in fact, the encryption scheme is CPA-secure. ■

6 Pedersen Vector Commitments

6.1 The Commitment Scheme (20 Points)

We will examine an efficient way to commit to a long message. Let $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$ be a cryptographic group of prime order q for which discrete log is hard.

1. $\mathsf{Gen}(1^n)$:
   (a) Sample $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$.
   (b) Sample $n+1$ group elements $g_1, \ldots, g_n, h \leftarrow \mathbb{G}$ independently and uniformly at random. Let $\mathbf{g} = (g_1, \ldots, g_n)$.
   (c) Output $\mathsf{params} = (\mathbb{G}, q, g, \mathbf{g}, h)$
2. $\mathsf{Commit}(\mathsf{params}, m; r)$:
   (a) Let $m = (m_1, \ldots, m_n) \in \mathbb{Z}_q^n$. Let $r \leftarrow \mathbb{Z}_q$ be sampled uniformly at random.
   (b) Compute and output:
   $$\mathsf{com} = h^r \cdot \prod_{i=1}^{n} g_i^{m_i}$$
3. Open :
   (a) The committer outputs (m, r).
   (b) The verifier checks whether $\mathsf{com} = \mathsf{Commit}(\mathsf{params}, m; r)$. If so, the verifier accepts, and if not, the verifier rejects.

Note that the commitment to n values in $\mathbb{Z}_q$ is a single group element in $\mathbb{G}$, so the scheme is more efficient than simply committing to each value separately.

Name:

Question 1: Prove that the commitment scheme is hiding.

Solution:

Theorem 6.1 *The commitment scheme is hiding.*

Proof:

1. Key Idea: h^r is uniformly random in $\mathbb{G}$ over the randomness of r, so h^r masks the value of $\overline{\prod_{i=1}^{n} g_i^{m_i}}$.
2. For any message vector m and any parameters params, the output of $\mathsf{Commit}(\mathsf{params}, m)$ is uniformly random in $\mathbb{G}$ due to the randomness of r.
3. Then the commitment $\mathsf{com}^* = \mathsf{Commit}(\mathsf{params}, m_b)$ is actually independent of b. In this case, the adversary's probability of correctly guessing b is exactly $\frac{1}{2}$. Therefore, the scheme is hiding.

■

Question 2: Prove that the commitment scheme is binding.

Solution:

Theorem 6.2 *Since discrete log is hard in $\mathcal{G}$, the commitment scheme is binding.*

Proof:

1. Assume toward contradiction that there is an adversary $\mathcal{A}$ that breaks binding. Then we will construct a PPT adversary $\mathcal{B}$ that breaks the discrete log assumption.
2. $\mathcal{B}$ will embed the discrete log instance into one index of the vector commitment and sample the other indices of randomly.

   Construction of $\mathcal{B}$:

   (a) Receive $(\mathbb{G}, q, g, g^y)$ from the challenger.
   (b) Sample $i \leftarrow [n]$, and set $g_i = g^y$.
   (c) For each $j \in [n] \setminus \{i\}$, sample $\gamma_j \leftarrow \mathbb{Z}_q$ and set $g_j := g^{\gamma_j}$. Also let $\mathbf{g} = (g_1, \ldots, g_n)$.
   (d) Sample $\eta \leftarrow \mathbb{Z}_q$ and set $h = g^\eta$.
   (e) Run $\mathcal{A}$ on $(\mathbb{G}, q, g, \mathbf{g}, h)$, and receive two openings (m, r) and (m', r'). Check whether $m_i = m'_i$. If so, abort the computation. If not, continue.

(f) Compute and output:

$$y' = \left[\eta \cdot (r' - r) + \sum_{j \in [n] \setminus \{i\}} \gamma_j \cdot (m'_j - m_j)\right] \cdot (m_i - m'_i)^{-1} \mod q \tag{1}$$

3. We will show that whenever $\mathsf{Commit}(\mathsf{params}, m; r) = \mathsf{Commit}(\mathsf{params}, m'; r')$ and $m_i \neq m'_i$, then $\mathcal{B}$ outputs $y' = y$ and wins the discrete log game.

   If $\mathsf{Commit}(\mathsf{params}, m; r) = \mathsf{Commit}(\mathsf{params}, m'; r')$ and $m_i \neq m'_i$, then:

$$h^r \cdot \prod_{j=1}^{n} g_j^{m_j} = h^{r'} \cdot \prod_{j=1}^{n} g_j^{m'_j}$$

$$g^{\eta \cdot r} \cdot g^{y \cdot m_i} \cdot \prod_{j \in [n] \setminus \{i\}} g^{\gamma_j \cdot m_j} = g^{\eta \cdot r'} \cdot g^{y \cdot m'_i} \cdot \prod_{j \in [n] \setminus \{i\}} g^{\gamma_j \cdot m'_j}$$

$$\eta \cdot r + y \cdot m_i + \sum_{j \in [n] \setminus \{i\}} \gamma_j \cdot m_j = \eta \cdot r' + y \cdot m'_i + \sum_{j \in [n] \setminus \{i\}} \gamma_j \cdot m'_j \mod q$$

$$y = \left[\eta \cdot (r' - r) + \sum_{j \in [n] \setminus \{i\}} \gamma_j \cdot (m'_j - m_j)\right] \cdot (m_i - m'_i)^{-1} \mod q$$

$$y = y'$$

4. We will now show that with non-negligible probability, $\mathcal{A}$'s output satisfies $\mathsf{Commit}(\mathsf{params}, m; r) = \mathsf{Commit}(\mathsf{params}, m'; r')$ and $m_i \neq m'_i$.

   (a) First note that $\mathcal{B}$ correctly simulates the hiding security game. The params given to $\mathcal{A}$ by $\mathcal{B}$ have the same distribution as params in the hiding game. Therefore, with non-negligible probability, $\mathcal{A}$'s output satisfies $\mathsf{Commit}(\mathsf{params}, m; r) = \mathsf{Commit}(\mathsf{params}, m'; r')$ and $m \neq m'$.

   (b) If $m \neq m'$, then for at least one $k \in [n]$ we have $m_k \neq m'_k$.

   (c) $\mathcal{A}$ has no information about $\mathcal{B}$'s choice of i. No matter which i-value is chosen by $\mathcal{B}$, the distribution of $(g_1, \ldots, g_n)$ is the same: they are sampled independently and uniformly from $\mathbb{G}$. Then:

$$\Pr[m_i \neq m'_i | m \neq m'] \geq \frac{1}{n}$$

   Therefore, $\Pr[\mathcal{B} \text{ breaks discrete log}] \geq \frac{\Pr[\mathcal{A} \text{ breaks hiding}]}{n}$, which is non-negligible.

■

Name:

6.2 Zero-Knowledge Opening Proof (20 Points)

Next, we will examine a protocol to open the commitment to a single index of the message vector without revealing any information about the rest of the message.

As before, let $\mathsf{com} = \mathsf{Commit}(\mathsf{params}, m; r)$. The instance of the proof will be $x = (\mathsf{params}, \mathsf{com}, m_n)$, and the witness will be $w = (m_1, \ldots, m_{n-1}, r)$. A given pair (x, w) is considered valid if the following relation is satisfied:

$$\mathfrak{R}(x, w) = \begin{cases} 1 & \text{if } \mathsf{com} = \mathsf{Commit}\big(\mathsf{params}, (m_1, \ldots, m_n); r\big) \\ 0 & \text{else} \end{cases}$$

Consider the following proof system for the above relation.

1. The prover samples $a, a_1, \ldots, a_{n-1} \leftarrow \mathbb{Z}_q$ independently and uniformly at random. Then they send the verifier the following value A:

$$A = h^a \cdot \prod_{i=1}^{n-1} g_i^{a_i}$$

2. The verifier samples $b \leftarrow \mathbb{Z}_q$ and sends it to the prover.

3. The prover sends the verifier the following values $(c, c_1, \ldots, c_{n-1})$:

$$\begin{aligned} c &= b \cdot r + a \\ c_1 &= b \cdot m_1 + a_1 \\ &\vdots \\ c_{n-1} &= b \cdot m_{n-1} + a_{n-1} \end{aligned}$$

4. The verifier outputs 1 if

$$A \cdot (\mathsf{com})^b = g_n^{b \cdot m_n} \cdot h^c \cdot \prod_{i=1}^{n-1} g_i^{c_i}$$

   and outputs 0 otherwise.

Question 3: Complete the verifier's algorithm above so that the protocol satisfies completeness.

Question 4: Prove that the protocol satisfies completeness.

Solution: Let us assume that $\mathfrak{R}(x, w) = 1$, and the prover and verifier follow the protocol as-written.

Then the verifier's check (eq. (2)) is equivalent to each of the following equations:

$$A \cdot (\mathsf{com})^b = g_n^{b \cdot m_n} \cdot h^c \cdot \prod_{i=1}^{n-1} g_i^{c_i} \tag{2}$$

$$h^a \cdot \left(\prod_{i=1}^{n-1} g_i^{a_i} \right) \cdot h^{b \cdot r} \cdot \left(\prod_{i=1}^{n} g_i^{b \cdot m_i} \right) = g_n^{b \cdot m_n} \cdot h^{b \cdot r + a} \cdot \prod_{i=1}^{n-1} g_i^{b \cdot m_i + a_i} \tag{3}$$

$$g_n^{b \cdot m_n} \cdot h^{b \cdot r + a} \cdot \prod_{i=1}^{n-1} g_i^{b \cdot m_i + a_i} = g_n^{b \cdot m_n} \cdot h^{b \cdot r + a} \cdot \prod_{i=1}^{n-1} g_i^{b \cdot m_i + a_i} \tag{4}$$

Equation (4) is a tautology – it is clearly true. This means the verifier will accept the proof with probability 1, so the protocol satisfies completeness.

Question 5: Prove that the proof system satisfies honest-verifier zero-knowledge.

Solution:

Theorem 6.3 *The protocol satisfies honest-verifier zero-knowledge.*

Proof:

1. Let us assume that $\mathfrak{R}(x, w) = 1$, and the prover and verifier follow the protocol as-written.
2. The verifier's view of the protocol comprises the following variables:

$$\mathsf{view}(V; x, w) = (x, A, b, c, c_1, \ldots, c_{n-1})$$

3. Now we will construct a simulator $\mathsf{Sim}^V(x)$ that simulates $\mathsf{view}(V; x, w)$.
   Construction of $\mathsf{Sim}^V(x)$:
   (a) Sample $b, c, c_1, \ldots, c_{n-1} \leftarrow \mathbb{Z}_q$ independently and uniformly at random.
   (b) Compute

$$A = (\mathsf{com})^{-b} \cdot g_n^{b \cdot m_n} \cdot h^c \cdot \prod_{i=1}^{n-1} g_i^{c_i}$$

   (c) Output $(x, A, b, c, c_1, \ldots, c_{n-1})$.
4. Now we will argue that the simulator's output has the same distribution as $\mathsf{view}(V; x, w)$.

(a) In the real protocol, b is uniformly random in $\mathbb{Z}_q$, as it is in the simulated protocol.

(b) For any given values of $(b, r, m_1, \ldots, m_n)$ the values of $(c, c_1, \ldots, c_{n-1})$ are independent and uniformly random in $\mathbb{Z}_q$, due to the randomness of $(a, a_1, \ldots, a_{n-1})$.

(c) For any given values of $\big(b, r, (m_1, \ldots, m_n), (c, c_1, \ldots, c_{n-1})\big)$, A is the unique value that satisfies eq. (2). So

$$A = (\mathsf{com})^{-b} \cdot g_n^{b \cdot m_n} \cdot h^c \cdot \prod_{i=1}^{n-1} g_i^{c_i}$$

This shows that the distribution of $(x, A, b, c, c_1, \ldots, c_{n-1})$ in the real protocol is the same as the distribution of the simulator's output.

Therefore, the protocol satisfies honest-verifier zero-knowledge. ■