

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 $1.5 \times 170 = 255$ minutes or $2 \times 170 = 340$ 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 $\mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$. 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.
 - $A(n) = f(n)^2 + g(n)$
 - $B(n) = |g(n) - f(n)|$
 - $C(n) = \frac{1}{n} \cdot g(n)$
 - $D(n) = g(n) \cdot f(n)$
 - $E(n) = g(n) \cdot g(n)$
 - $F(n) = g(n) \cdot g(n + 1)$
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.
3. Let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ 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?
 - A. Decisional Diffie Hellman in \mathbb{G} .

Name:

-
- B. Computational Diffie Hellman in \mathbb{G} .
 - C. Discrete Log in \mathbb{G} .
 - D. Discrete Log in \mathbb{G}_T .
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
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

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

1. $\text{Gen}(1^n)$: Sample $k \leftarrow \{0, 1\}^n$ and output k .
2. $\text{Enc}(k, m)$: Sample $r \leftarrow \{0, 1\}^n$. Compute and output

$$c = F_k(m \parallel r)$$

3. $\text{Dec}(k, c)$: 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.

Name:

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





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

- Yes No

Name:

2.2 Concatenating The Base Scheme (15 Points)

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

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

1. $\text{Gen}'(1^n) = \text{Gen}(1^n)$

2. $\text{Enc}'(k, m)$: Output

$$c = (c_1 \parallel \dots \parallel c_t) = (\text{Enc}(k, m_1) \parallel \dots \parallel \text{Enc}(k, m_t))$$

3. $\text{Dec}'(\text{sk}, c) = \text{Dec}(k, c_1) \parallel \dots \parallel \text{Dec}(k, c_t)$

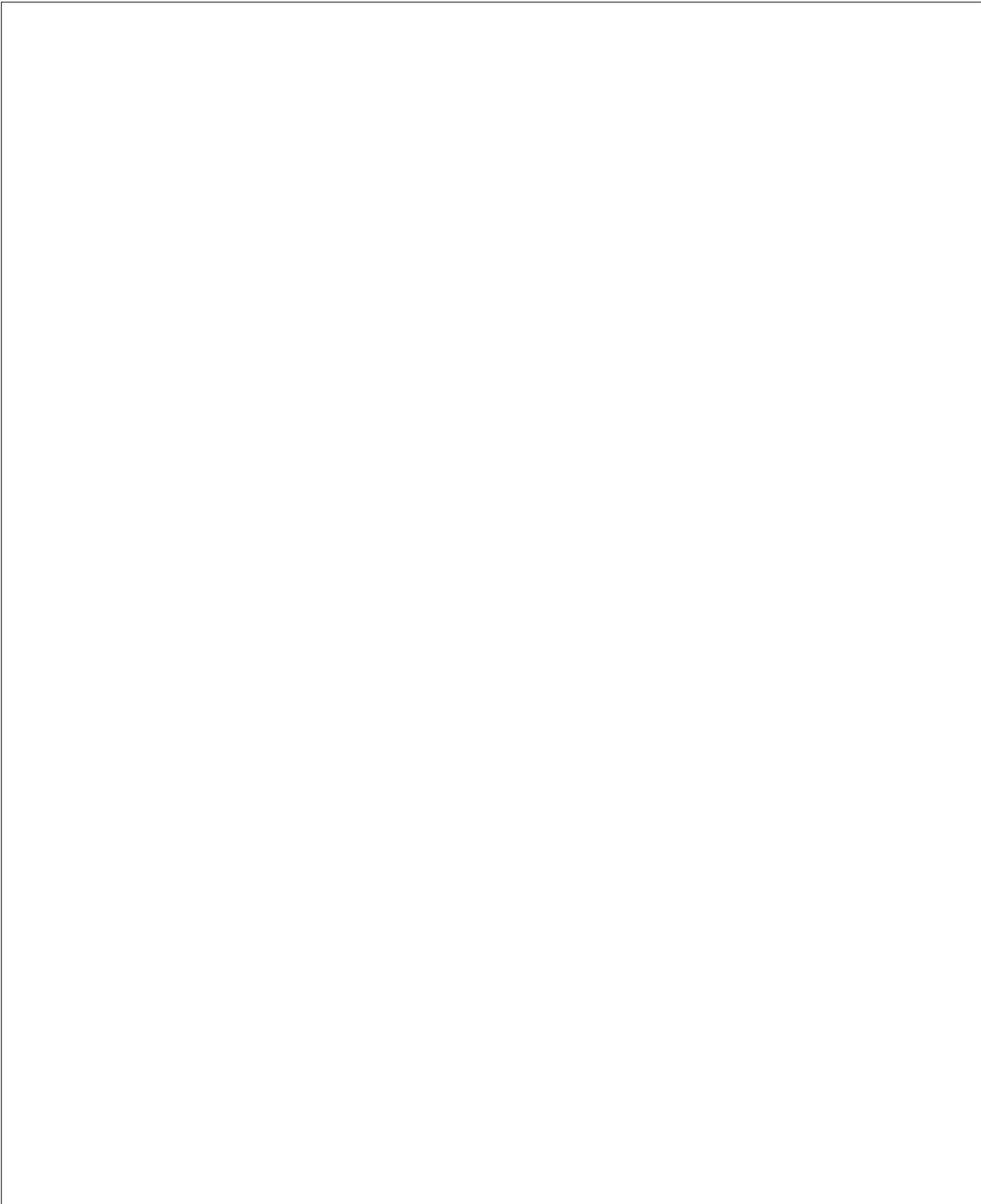
Question 4: Is Π' necessarily CPA-secure? No proof is needed.

Yes No

Question 5: Is Π' necessarily CCA2-secure?

Yes No

Prove your answer.



Name:

3 One-Way Functions (25 Points)

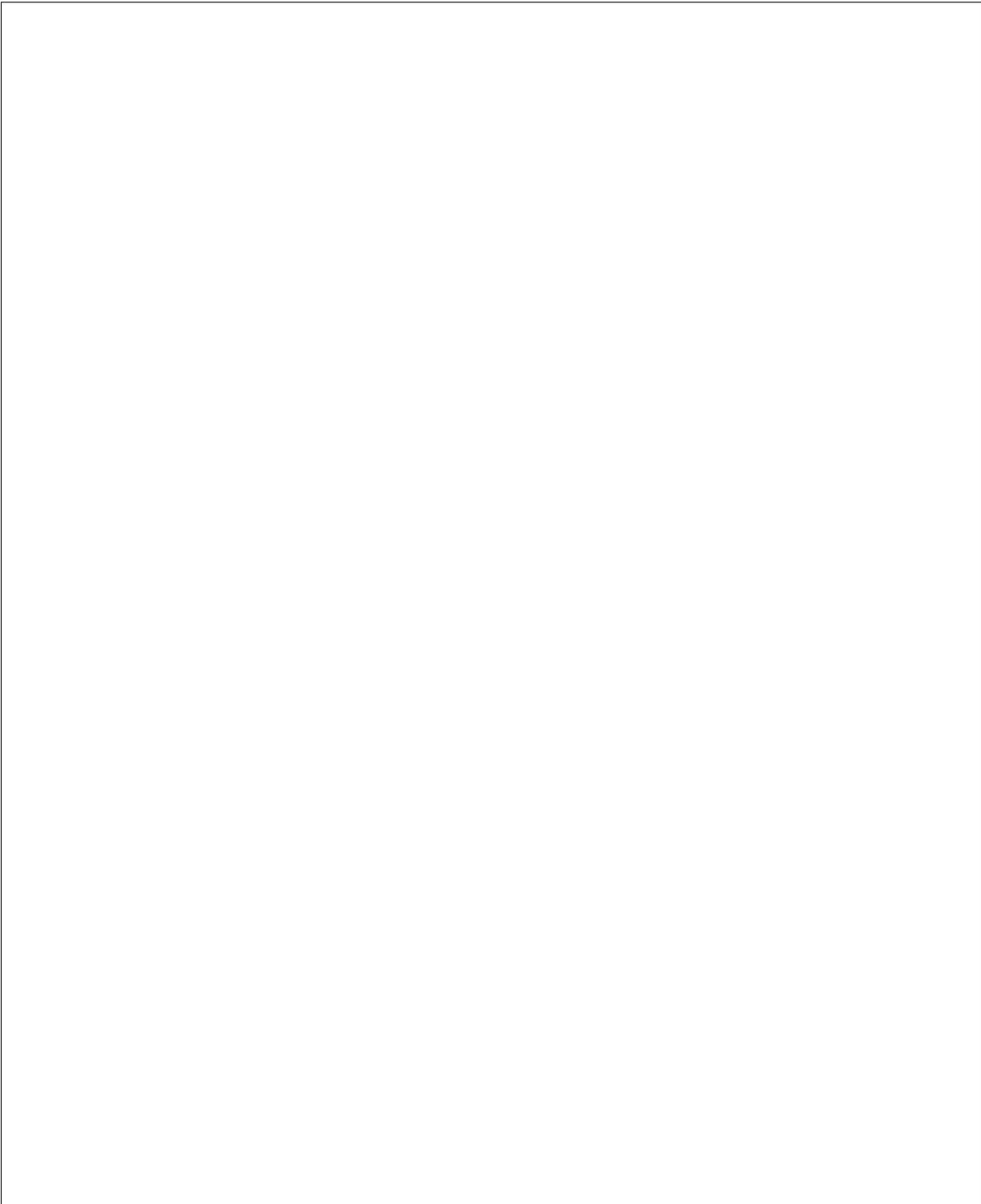
Let $f : \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a one-way function. Let $x = (x_L, x_R) \in \{0, 1\}^n \times \{0, 1\}^n$ be a generic input. Now consider the following functions constructed from f :

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. $g_4(x) = f(x_L) \oplus f(x_R)$

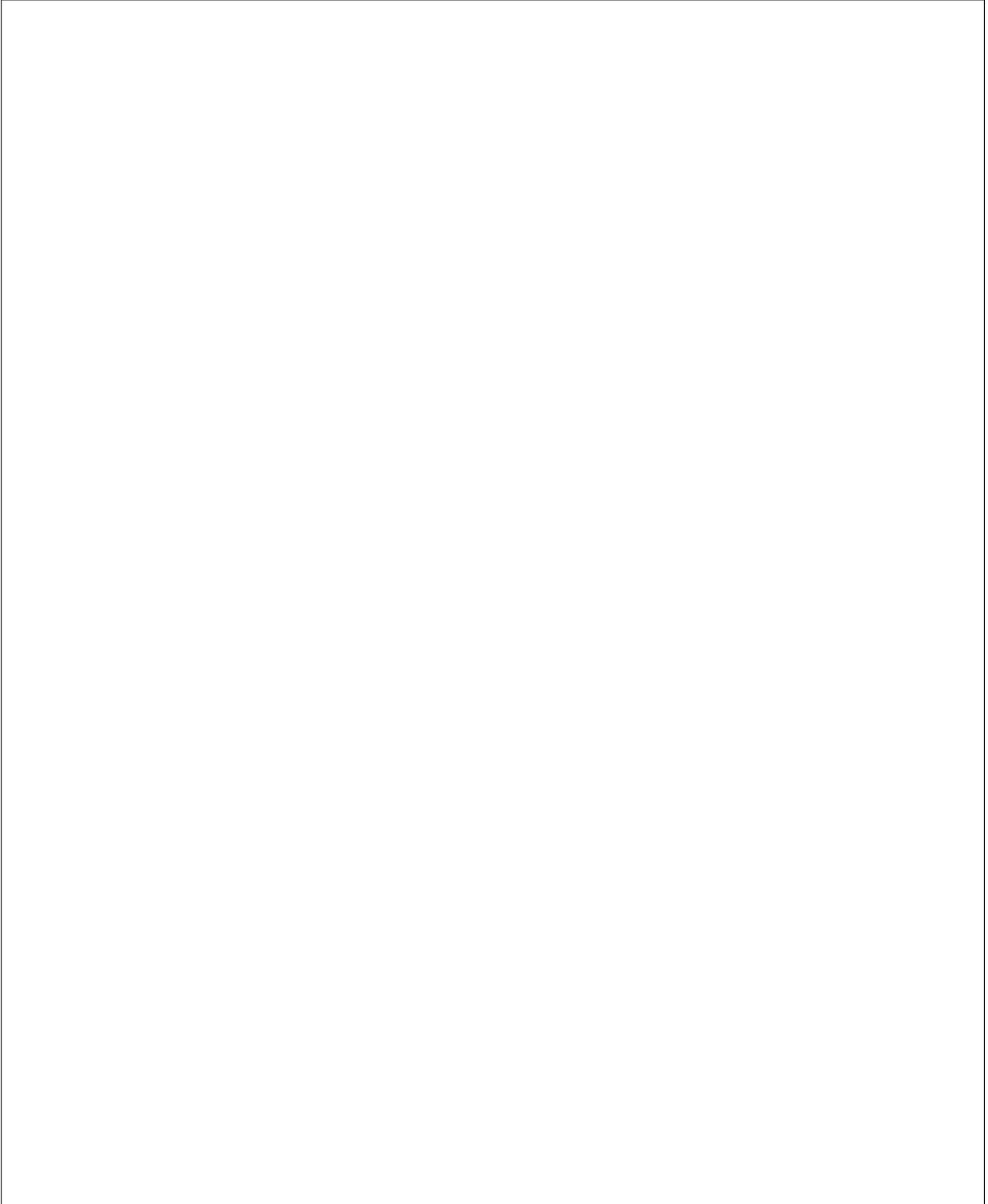
Question: For each function (g_1, g_2, g_3, g_4) , indicate whether it is necessarily a one-way function, and prove your answer.

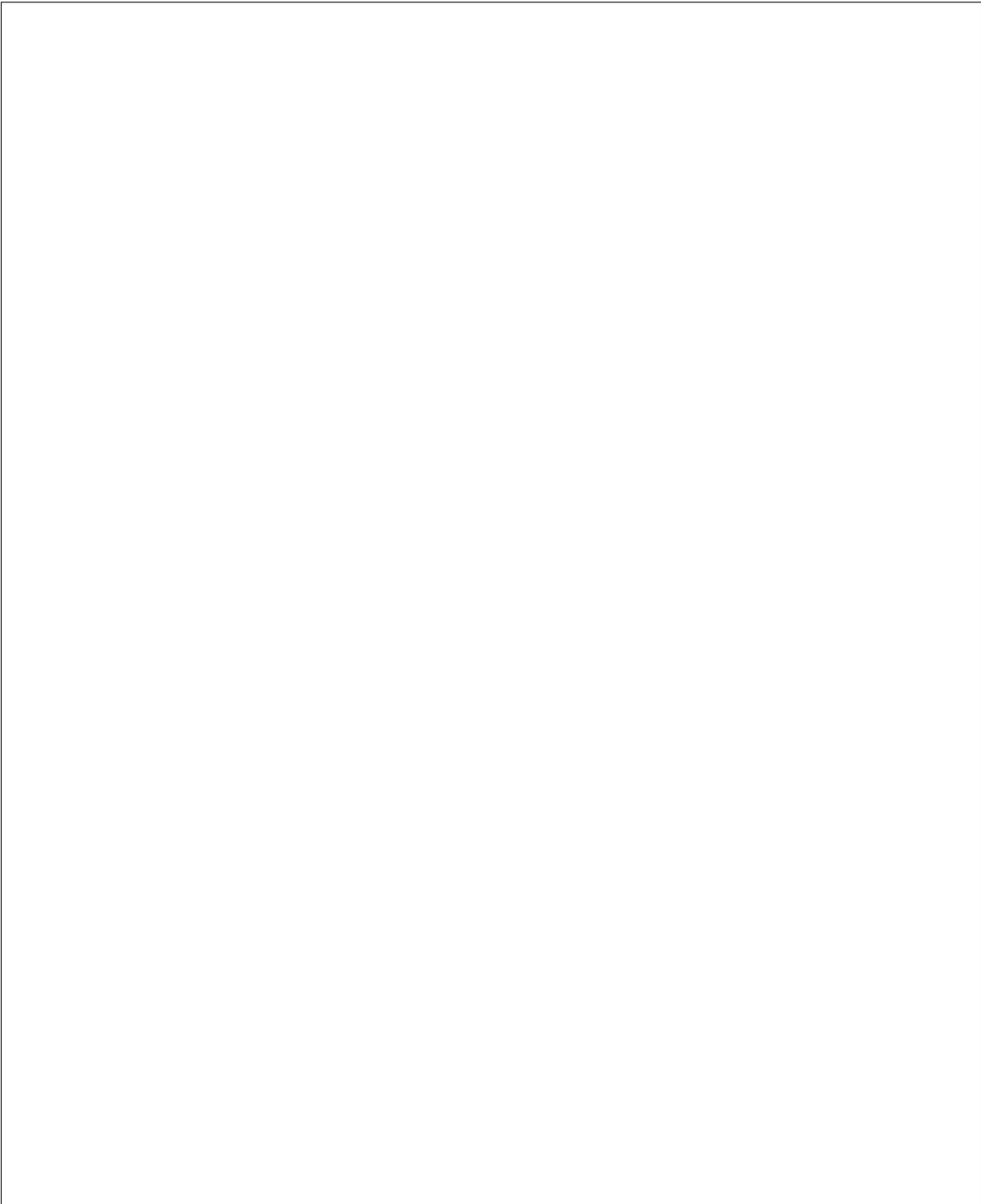
As a guideline, your answer for each g_i should do one of the following:

- 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 .



Name:





Name:

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 $\mathcal{S} = (\text{Gen}, \text{Sign}, \text{Verify})$ 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 $\text{Sign}(\text{sk}, m; r)$ to make the random input explicit.

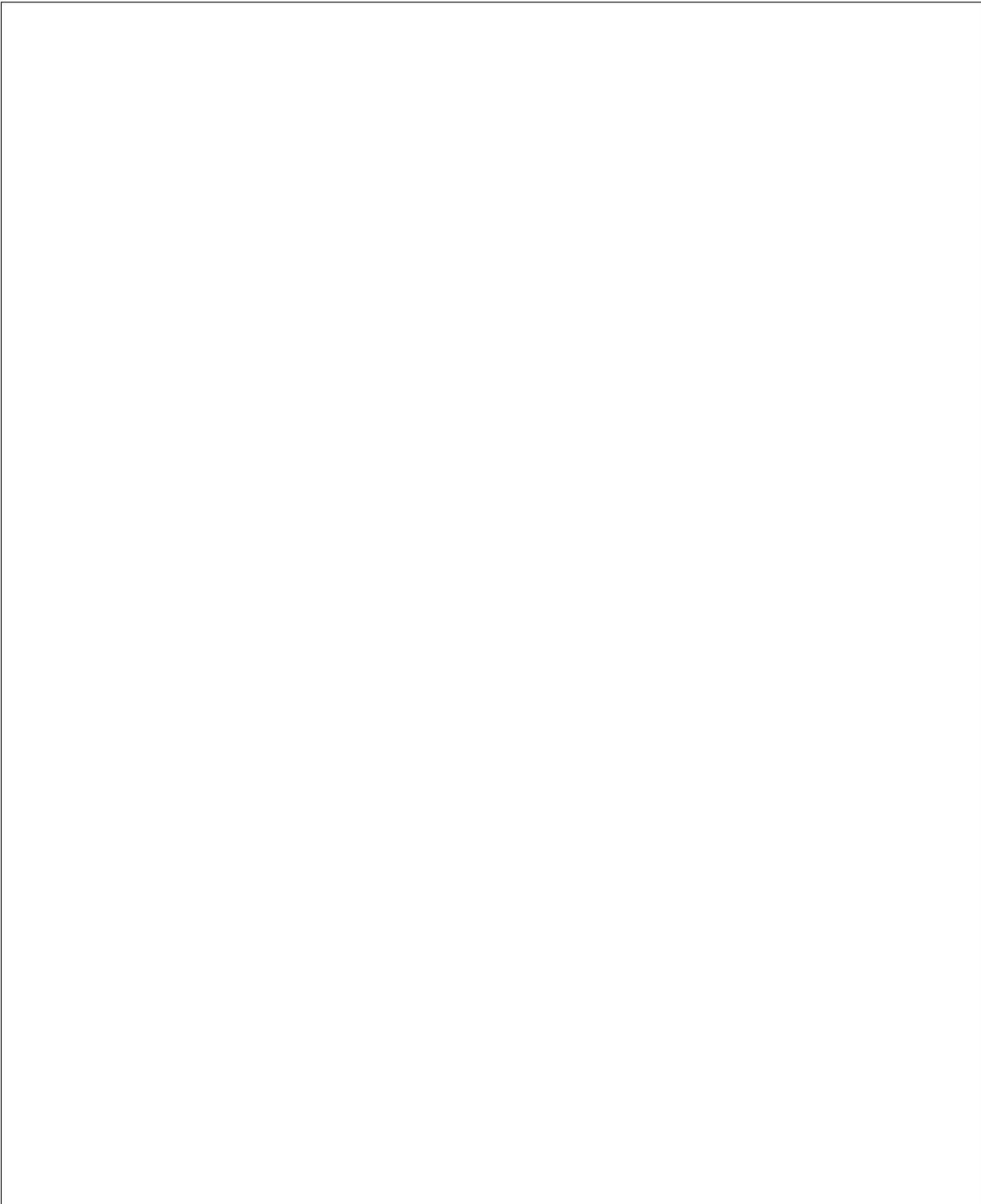
Let $F : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a secure PRF.

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

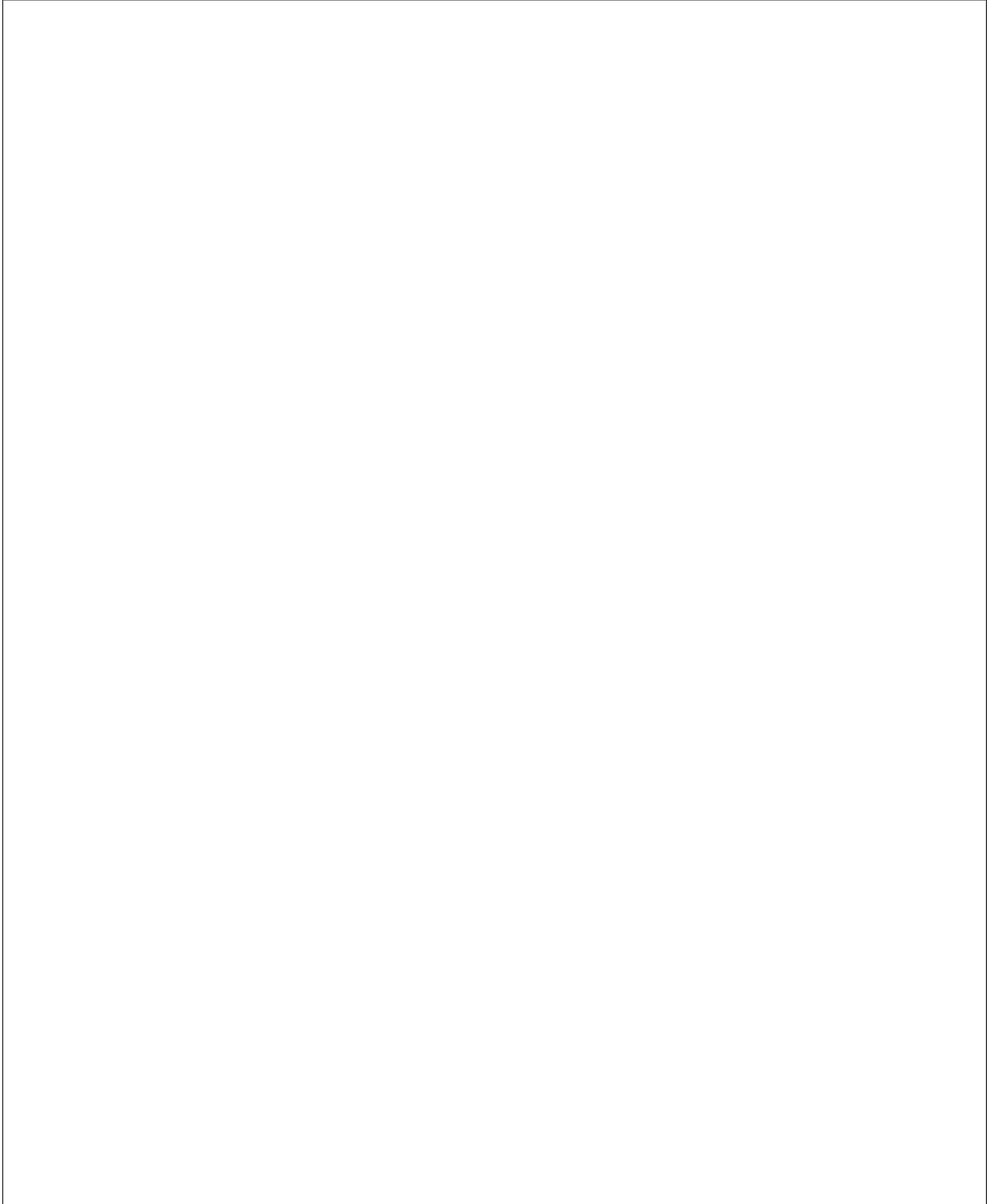
1. $\text{Gen}'(1^n)$:
 - (a) Sample $(\text{pk}, \text{sk}) \leftarrow \text{Gen}(1^n)$.
 - (b) Sample $k \leftarrow \{0, 1\}^n$.
 - (c) Output $\text{pk}' = \text{pk}$ and $\text{sk}' = (\text{sk}, k)$.
2. $\text{Sign}'(\text{sk}, m)$: Output $\sigma = \text{Sign}(\text{sk}, m; F(k, m))$.
3. $\text{Verify}'(\text{pk},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

Note that Sign' is deterministic.

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



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. $\text{Gen}(1^n)$:

- (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 $\text{pk} = (\mathbb{G}, q, g, h)$ and $\text{sk} = (\text{pk}, x)$.

2. $\text{Enc}(\text{pk}, m)$:

- 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

$$c = (c_1, c_2) = (g^y, g^z)$$

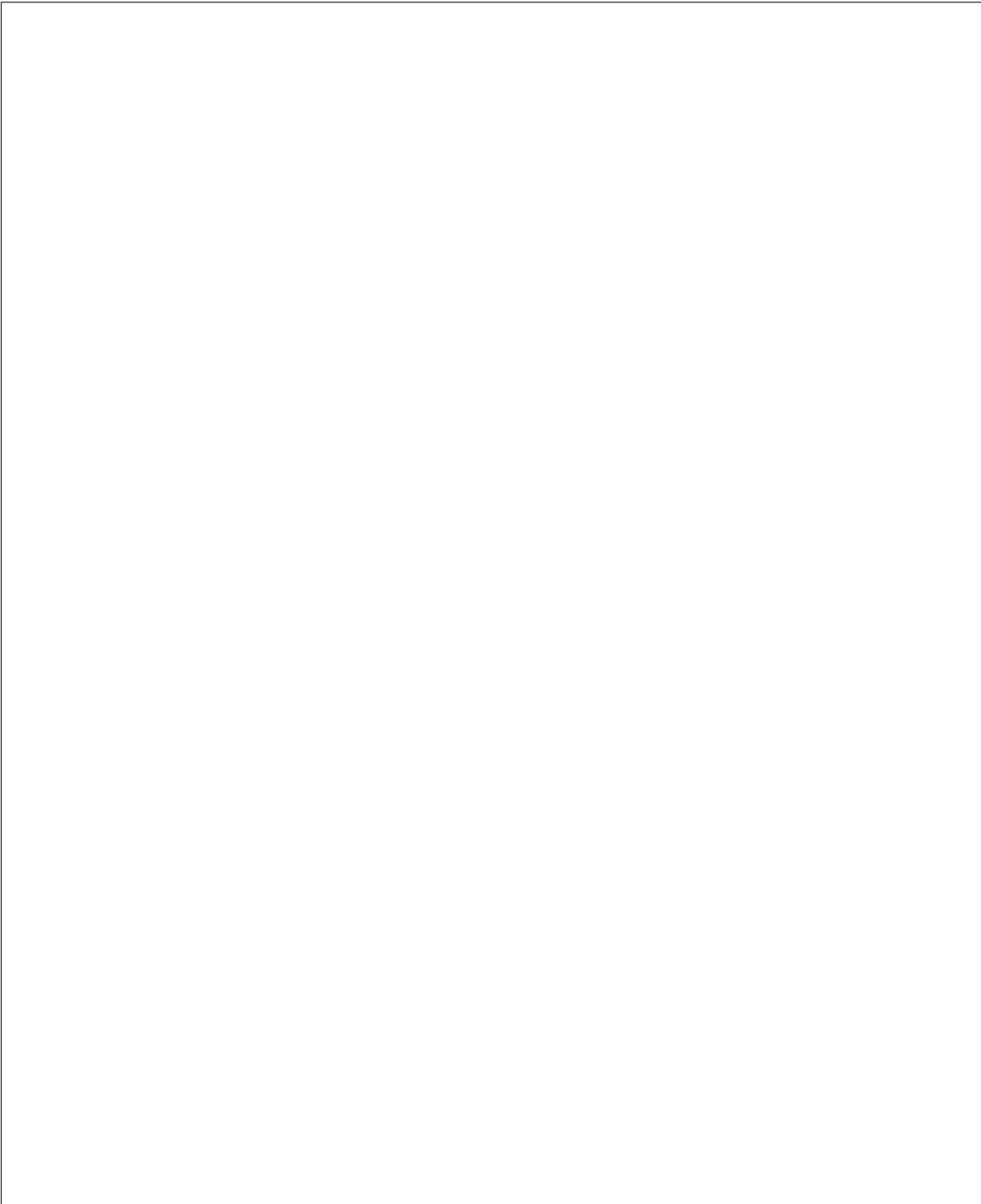
3. $\text{Dec}(\text{sk}, c)$:

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

Name:

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

Question 3: Prove that $(\text{Gen}, \text{Enc}, \text{Dec})$ 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. $\text{Gen}(1^n)$:

- (a) Sample $(\mathbb{G}, q, g) \leftarrow \mathcal{G}(1^n)$.
- (b) Sample $n + 1$ group elements $g_1, \dots, g_n, h \leftarrow \mathbb{G}$ independently and uniformly at random. Let $\mathbf{g} = (g_1, \dots, g_n)$.
- (c) Output $\text{params} = (\mathbb{G}, q, g, \mathbf{g}, h)$

2. $\text{Commit}(\text{params}, m; r)$:

- (a) Let $m = (m_1, \dots, m_n) \in \mathbb{Z}_q^n$. Let $r \leftarrow \mathbb{Z}_q$ be sampled uniformly at random.
- (b) Compute and output:

$$\text{com} = h^r \cdot \prod_{i=1}^n g_i^{m_i}$$

3. Open :

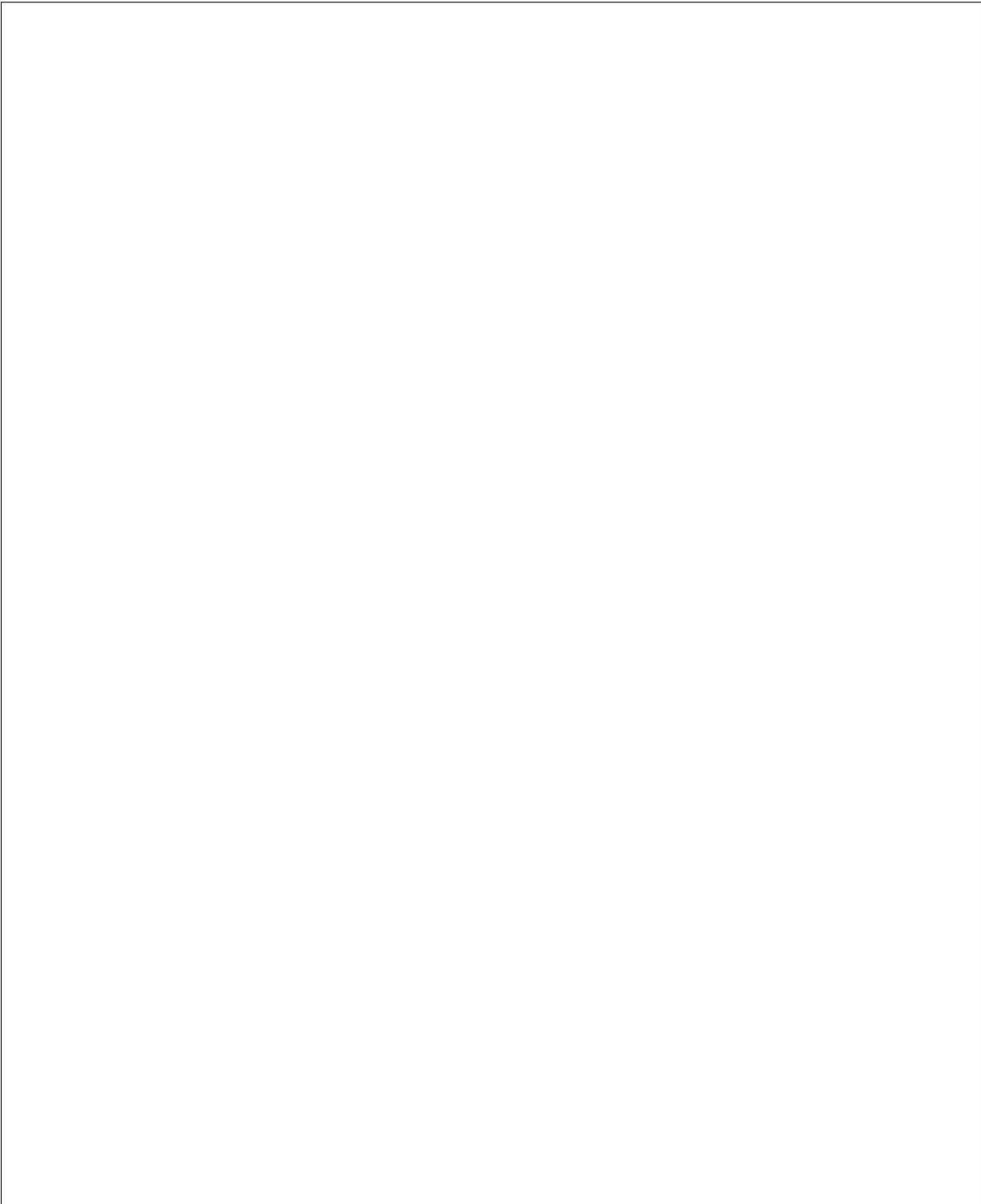
- (a) The committer outputs (m, r) .
- (b) The verifier checks whether $\text{com} = \text{Commit}(\text{params}, m; r)$. If so, the verifier accepts, and if not, the verifier rejects.

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.

Question 1: Prove that the commitment scheme is hiding.

Name:

Question 2: Prove that the commitment scheme is binding.



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

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

Consider the following proof system for the above relation.

1. The prover samples $a, a_1, \dots, a_{n-1} \leftarrow \mathbb{Z}_q$ 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, \dots, c_{n-1}) :

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

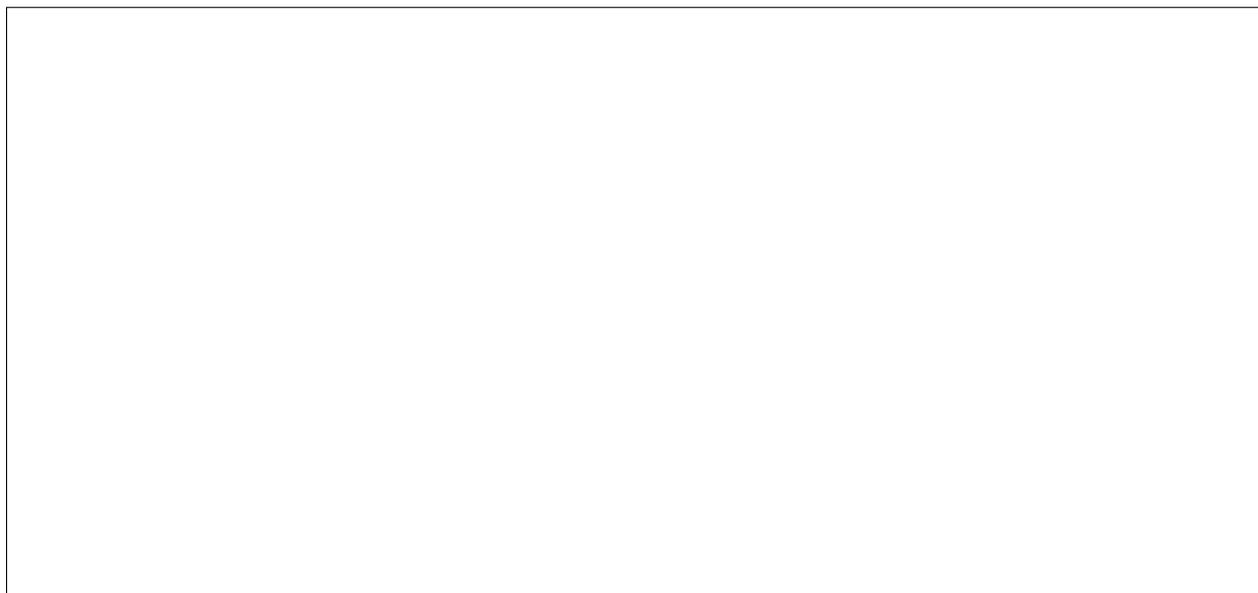
4. The verifier outputs 1 if

$$A \cdot (\text{com})^b = \boxed{\phantom{\text{com}}}$$

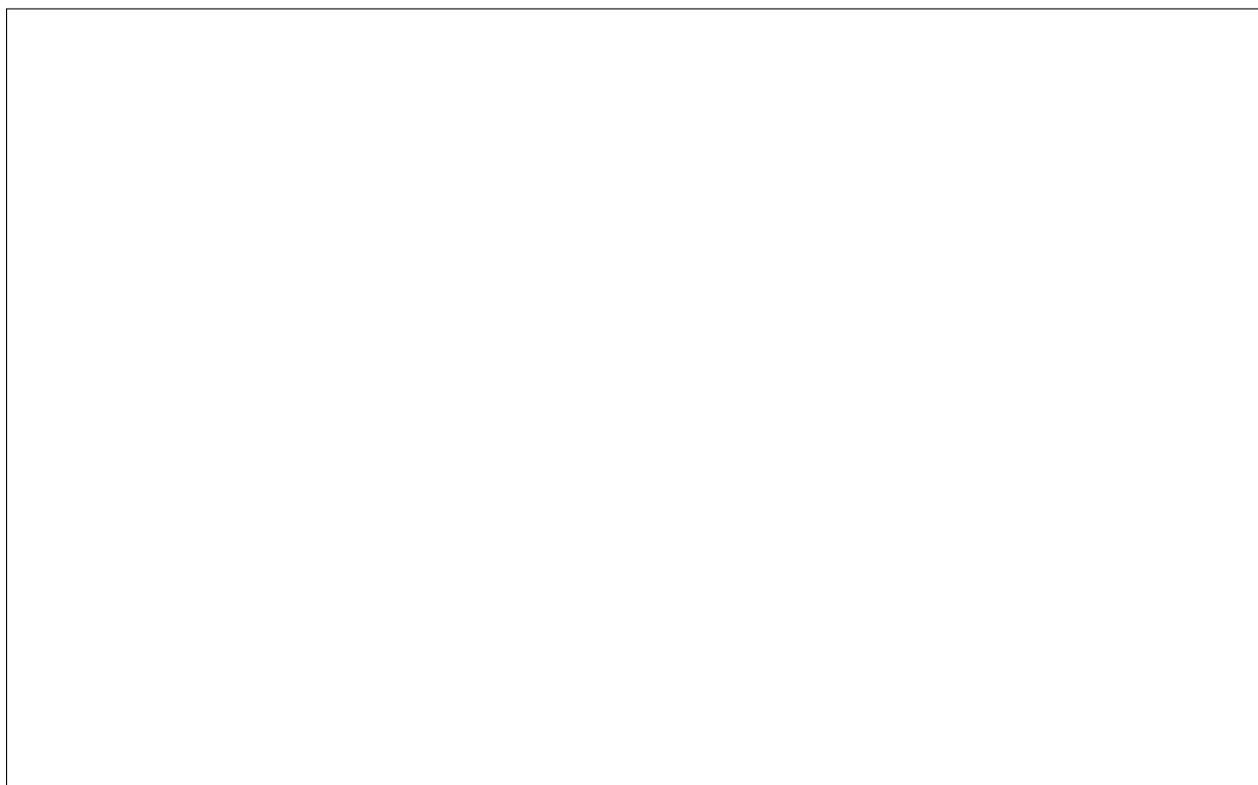
and outputs 0 otherwise.

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

Question 4: Prove that the protocol satisfies completeness.



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



Name:

