Instructions

- Please check that your answer script has 24 pages, and use the space provided for your answers (you can ask for more rough sheets if needed).
- The first question (MCQs/Short Answer Questions) has **thirteen** parts, total worth 37 marks.

  The second question (Signature Combiners) has **three** parts, total worth 10 marks.

  The third question (Broadcast Encryption) has **three** parts, total worth 13 marks.
- MCQs/Short Answer Questions: Page 2 to Page 10.
  - $\mathbb{Z}_p^*$, DDH: Page 2
  - RSA: Page 3
  - Random Oracle Model: Page 4
  - MAC: Page 6
  - CCA: Page 7
  - UHFs, CRHFs: Page 9
- Signatures (Long Answer Question): Page 13 to Page 16.
- Broadcast Encryption (Long Answer Question): Page 17 to Page 21.
- Rough work: Page 22 to Page 24.

Notations

- For a positive integer a, $[a]$ denotes the set $\{1, 2, \ldots, a\}$. For integers a and $b > a$, $[a, b]$ denotes the set $\{a, a + 1, \ldots, b\}$.
- $\mathbb{Z}_N = \{0, 1, \ldots, N - 1\}$. $\mathbb{Z}_N^* = \{x \in \mathbb{Z}_N : \mathsf{gcd}(x, N) = 1\}$.
- $x \,||\, y$ denotes the concatenation of x and y.
- $\{0,1\}^{\leq \ell} = \bigcup_{i=1}^{\ell} \{0,1\}^i$ (the set of all bit strings with at most ℓ bits).

1 MCQs/Short Answers (37 marks)

For each of the following questions, provide a short answer **in the space provided**.

$\mathbb{Z}_p^*$, Group theory, DDH

1. (2 marks) Let $p = 2q + 1$ be a prime, where q is also prime. Which of the following statements are true about the set $\mathbb{Z}_p^*$ (multiple statements can be true; write 'none-of-the-above' if all are false):

   (A) All elements of $\mathbb{Z}_p^*$ are generators of $\mathbb{Z}_p^*$, except 1.

   (B) For any number $a \in \mathbb{Z}_p^*$, there exists a number $b \in \mathbb{Z}_p^*$ such that $a \cdot b \bmod p = 1$.

   (C) For any number $a \in \mathbb{Z}_p^*$, there exists a number $b \in \mathbb{Z}_p^*$ such that $a + b \bmod p = 1$.

2. (3 marks) Recall the Elgamal public key encryption scheme. The message space is a prime-order group $\mathbb{G}$ of size q. Let $\mathsf{pk} = (g, h) \in \mathbb{G}^2$ be an Elgamal public key. For a message $m \in \mathbb{G}$, let $\mathcal{S}_{\mathsf{pk},m}$ denote the set of all Elgamal ciphertexts that are encryptions of m using pk.

   You are given a **uniformly random sample** $((\mathsf{ct}_{1,1}, \mathsf{ct}_{1,2}), (\mathsf{ct}_{2,1}, \mathsf{ct}_{2,2}))$ from $\mathcal{S}_{\mathsf{pk},m_1} \times \mathcal{S}_{\mathsf{pk},m_2}$. Describe how to generate a uniformly random sample from $\mathcal{S}_{\mathsf{pk},m_1} \times \mathcal{S}_{\mathsf{pk},m_2} \times \mathcal{S}_{\mathsf{pk},m_1 \cdot m_2}$ **without knowing** m_1, m_2. You should use $(\mathsf{ct}_{1,1}, \mathsf{ct}_{1,2}, \mathsf{ct}_{2,1}, \mathsf{ct}_{2,2})$ and (g, h) for generating this sample.

RSA

3. (2 marks) Recall the 'textbook RSA' signature scheme. Complete the following attack on the signature scheme (fill in the blank space provided).

   1. Adversary receives $\mathsf{vk} = (N, e)$ from the challenger.

   2. Adversary picks random $m \leftarrow \mathbb{Z}_N$, queries for a signature on m, receives signature σ.

   3. Adversary sends ($m^* =$ ____________ , $\sigma^* = 2\sigma \bmod N$) as a forgery.

4. (3 marks) A *twin prime* is a pair of numbers $(p, p+2)$ such that both p and $p+2$ are primes. It is conjectured that there are infinitely many twin primes, and they are also easy to sample. Consider the following public key encryption scheme (defined using a publicly computable hash function $H : \mathbb{Z}_N \rightarrow \{0,1\}^n$). The message space is $\{0,1\}^n$, and the algorithms are defined as follows:

   - KeyGen : Choose a twin prime pair $(p, p+2)$. Set $N = p \cdot (p+2)$. Choose e co-prime to $\phi(N)$, and an integer d such that $e \cdot d \bmod \phi(N) = 1$. Set $\mathsf{pk} = (N, e)$, $\mathsf{sk} = (N, d)$.
   - $\mathsf{Enc}(m, \mathsf{pk})$: Choose $x \leftarrow \mathbb{Z}_N^*$, output $\mathsf{ct}_1 = x^e \bmod N$, $\mathsf{ct}_2 = H(x) \oplus m$.
   - $\mathsf{Dec}(\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2), \mathsf{sk})$: Compute $y_1 = \mathsf{ct}_1^d \bmod N$. Output $\mathsf{ct}_2 \oplus H(y_1)$.

   Show that this scheme is not semantically secure. More formally, show a polynomial time algorithm that, given pk and $\mathsf{Enc}(m, \mathsf{pk})$ for any $m \in \{0,1\}^n$, can fully recover m. [1]

[1] Trivia: Unfortunately, such errors often arise in implementations of RSA-based encryption schemes. This is completely insecure.

Random Oracle Model

5. (2 marks) Is it possible to have a **semantically secure** private-key encryption scheme with **deterministic** encryption in the **random oracle model**? If yes, then provide a candidate construction (no security proof needed for the candidate). If no, then briefly state why it is not possible.

6. (6 marks) Let $H : \{0,1\}^* \rightarrow \{0,1\}^n$ be a deterministic function. Construct a private-key encryption scheme $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$ with key space $\{0,1\}^n$, message space $\{0,1\}^*$. The encryption and decryption algorithms should use the function H, and the scheme should be semantically secure in the **random oracle model**. The proof of security in random oracle model should not use any other computational assumptions. Provide a short justification why it is semantically secure (formal proof not needed).

$\mathsf{Enc}(m, k)$:

$\mathsf{Dec}(\mathsf{ct}, k)$:

Informal security proof for your construction:

Unconditionally Secure MAC

7. (5 marks) Construct a MAC scheme with message space $\{0,1\}^n$ that is **unconditionally unforgeable** against a **single** query. More formally, for any adversary (even computationally unbounded ones), the adversary's winning probability in the following game is negligible:

   - Challenger picks a MAC key $k \leftarrow \mathcal{K}$.
   - Adversary sends a signing query for message m, and receives $\mathsf{Sign}(m, k)$.
   - Adversary must output m' and signature σ'. It wins if $m' \neq m$ and $\mathsf{Verify}(m', \sigma', k) = 1$.

   No security proof needed for this question.

   Choose an appropriate key space $\mathcal{K}$:

   Describe the signing algorithm $\mathsf{Sign}(m, k)$:

   Describe the verification algorithm $\mathsf{Verify}(m, \sigma, k)$:

CCA security

8. (2 marks) Which of the following statements are true about CCA security for **public-key encryption schemes** (multiple statements can be true; write 'none-of-the-above' if all are false):

   (A) In the private-key setting, the 'Encrypt-then-MAC' approach results in a CCA-secure private-key encryption scheme. Similarly, in the public-key setting, 'Encrypt-then-Sign' approach results in a CCA secure encryption scheme.

   (B) CCA security in the public key setting implies ciphertext integrity.

   (C) Let $\mathsf{CCA\text{-}no\text{-}pre}$ denote the CCA security game where no pre-challenge decryption queries are made by the adversary. This game is equivalent to the CCA security game.

9. (3 marks) Let $\mathcal{E} = (\mathsf{Enc}, \mathsf{Dec})$ be a **private-key encryption scheme** with key space $\mathcal{K}$, message space $\{0,1\}^n$, satisfying security against **chosen ciphertext attacks**. Consider the following private-key encryption scheme $\mathcal{E}' = (\mathsf{Enc}', \mathsf{Dec}')$ with key space $\mathcal{K} \times \mathcal{K}$, message space $\{0,1\}^n$:

   - $\mathsf{Enc}'(m, (k_1, k_2))$: Compute $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(m, k_1)$, $\mathsf{ct}_2 \leftarrow \mathsf{Enc}(m, k_2)$ and output $(\mathsf{ct}_1, \mathsf{ct}_2)$.
   - $\mathsf{Dec}'((\mathsf{ct}_1, \mathsf{ct}_2), (k_1, k_2))$: Compute $y_1 = \mathsf{Dec}(\mathsf{ct}_1, k_1)$ and $y_2 = \mathsf{Dec}(\mathsf{ct}_2, k_2)$. If either y_1 or y_2 is $\perp$, then output $\perp$. If $y_1 \neq y_2$, output $\perp$. Else output y_1.

   Show that $\mathcal{E}'$ is **not** secure against chosen ciphertext attacks. Describe the attack formally, clearly stating the pre-challenge encryption/decryption queries, the challenge query, followed by the post-challenge encryption/decryption queries.

UHFs, CRHFs

10. (2 marks) Let $\{U_k : \{0,1\}^{2n} \to \{0,1\}^n\}_{k\in\mathcal{K}}$ be a family of secure universal hash functions. Consider $U'_k : \{0,1\}^{2n\ell} \to \{0,1\}^n$, where

$$U'_k(m_1 \mid\mid \ \ldots\ \mid\mid m_\ell) = U_k(m_1) \oplus U_k(m_2) \oplus \cdots \oplus U_k(m_\ell)$$

where each $m_i \in \{0,1\}^{2n}$. Is $\{U'_k\}_{k\in\mathcal{K}}$ a universal hash function family? If it is, provide a two-line justification, else provide an attack.

11. (2 marks) Let $\{H_k : \{0,1\}^{2n} \to \{0,1\}^n\}_{k\in\mathcal{K}}$ be a secure collision-resistant family of hash functions. Consider $H'_k : \{0,1\}^{\leq 2n} \to \{0,1\}^n$, where

$$H'_k(x) = \begin{cases} H_k(x) & \text{if } x \in \{0,1\}^{2n} \\ x \mid\mid 0^i & \text{if } x \in \{0,1\}^{n-i}, i \in [0, n-1] \\ H_k(x \mid\mid 0^i) & \text{if } x \in \{0,1\}^{2n-i}, i \in [1, n-1] \end{cases}$$

Is $\{H'_k\}_{k\in\mathcal{K}}$ a collision-resistant hash function family? If it is, provide a two-line justification, else provide an attack.

12. (3 marks)

Let (F, F^{-1}) be a secure PRP with key space, input space and output space $\{0,1\}^n$. Consider the following hash function with key space $\{0,1\}^n$, input space $\{0,1\}^n \times \{0,1\}^n$ and output space $\{0,1\}^n$.

$$H_k(a,b) = F(a \oplus b \oplus k, a) \oplus a.$$

That is, the hash function uses the first input a as the PRP key. The PRP evaluation is on the n-bit string $a \oplus b \oplus k$, and this PRP evaluation is XORd with the string a.

Show that this is not a secure CRHF by providing an explicit attack. [2]

13. (2 marks) Let $\mathbb{G}$ be a group of size q, where q is prime. Consider the following candidate hash function. The domain is $\mathbb{Z}_q \times \mathbb{Z}_q \times \mathbb{Z}_q$, the range is $\mathbb{Z}_q \times \mathbb{G}$. The hash key consists of two group generators g, h and an integer $x \in \mathbb{Z}_q$. The hash function is defined as follows:

$$H_{(g,h,x)}(a,b,c) = \left(a\ ,\ g^b \cdot h^{a \cdot c} \cdot h^{-x \cdot c}\right)$$

Show that this is not collision resistant (that is, show a collision on the above hash function, given the hash key (g,h,x)).

[2] Trivia: a few variants of this scheme are provably secure in certain idealized models, and are used in practice.

(This page is intentionally left blank for extra workspace)

(This page is intentionally left blank for extra workspace)

2 Signature Combiner (10 marks)

You are given two signature schemes $\mathcal{S}_1 = (\mathsf{KeyGen}_1, \mathsf{Sign}_1, \mathsf{Verify}_1)$ and $\mathcal{S}_2 = (\mathsf{KeyGen}_2, \mathsf{Sign}_2, \mathsf{Verify}_2)$. Both schemes have message space $\{0,1\}^*$, signature space $\{0,1\}^n$, and are perfectly correct. However, only one of these schemes is **weakly unforgeable**. We don't know which one it is, and we have no security guarantees for the other signature scheme. Construct a new signature scheme $\mathcal{S} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ by combining these two schemes, such that $\mathcal{S}$ is also perfectly correct, and is **weakly unforgeable**, assuming at least one of $\mathcal{S}_1$ or $\mathcal{S}_2$ is weakly unforgeable.

1. (3 marks) First, describe the signature scheme $\mathcal{S}$. You must describe all three algorithms.

2. (4 marks) Show that if there exists a p.p.t. adversary that breaks the weak unforgeability of $\mathcal{S}$, then there exists a p.p.t. reduction $\mathcal{B}_1$ that breaks the weak unforgeability of $\mathcal{S}_1$.

   Similarly show that if there exists a p.p.t. adversary that breaks the weak unforgeability of $\mathcal{S}$, then there exists a p.p.t. reduction $\mathcal{B}_2$ that breaks the weak unforgeability of $\mathcal{S}_2$.

3. (3 marks) Suppose you are given that one of the schemes is **strongly unforgeable** (but again, you don't know which one). Will your signature scheme $\mathcal{S}$ (described in part 1 above) also be strongly unforgeable? You should not assume that the signing algorithms Sign_1 or Sign_2 are deterministic.

(This page is intentionally left blank for extra workspace)

3 Encryption for broadcast channels, with piracy detection (10 marks)

[Since the problem statement is lengthy, feel free to discuss with instructor for problem overview.]

You are starting a new digital content delivery platform, based on subscription model. Suppose you wish to support at most t subscribers, here's the rough plan:

- Initially, you will choose a public key pk together with t secret keys $\mathsf{sk}_1, \mathsf{sk}_2, \ldots, \mathsf{sk}_t$.
- Whenever a new (say i^{th}) subscriber joins, he/she makes a payment, and you give him/her the secret key sk_i. Using this secret key, the subscriber can access all your old/new content.
- Whenever you wish to release new content, say a message m, you encrypt this message using pk (and place it on some public server). Any of the subscribers must be able to decrypt the ciphertext using **their own secret key**. However, if someone is not a subscriber (that is, he/she does not have any of the secret keys) then he/she should not learn anything about the message.

Let us call this a 'broadcast encryption scheme'. Formally, it consists of the following algorithms:

- $\mathsf{BKeygen}(1^n, 1^t)$: The key generation algorithm takes as input the security parameter n, the number of subscribers t. It outputs a public key pk, together with t secret keys $\mathsf{sk}_1, \ldots, \mathsf{sk}_t$.
- $\mathsf{BEnc}(m, \mathsf{pk})$: The encryption algorithm is randomized; it takes as input a message m, a public key pk, and outputs a ciphertext ct.
- $\mathsf{BDec}(\mathsf{ct}, \mathsf{sk})$: The decryption algorithm takes as input a ciphertext ct, a secret key sk, and outputs a message m.

For correctness, we require the following guarantee for any message m: if $(\mathsf{pk}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_t)) \leftarrow \mathsf{KeyGen}(1^n, 1^t)$, and $\mathsf{ct} \leftarrow \mathsf{Enc}(m, \mathsf{pk})$, then for all $i \in [t]$, $\mathsf{Dec}(\mathsf{ct}, \mathsf{sk}_i) = m$.

Semantic Security

For semantic security, we require the following guarantee - no p.p.t. adversary should win in the following game with non-negligible advantage:

Broadcast Encryption - Semantic Security

1. Challenger chooses $(\mathsf{pk}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_t)) \leftarrow \mathsf{BKeygen}(1^n, 1^t)$. It sends pk to the adversary.
2. Adversary chooses two messages m_0, m_1 and sends them to the challenger. Challenger picks a uniformly random bit $b \leftarrow \{0, 1\}$, sends $\mathsf{ct} \leftarrow \mathsf{BEnc}(m_b, \mathsf{pk})$ to the adversary.
3. Adversary sends its guess b', and wins if $b = b'$.

Figure 1: Semantic Security Game for Broadcast Encryption

1. (3 marks) Let $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be a **public key encryption scheme** with message space $\mathcal{M}$. Construct a broadcast encryption scheme for t users, with message space $\mathcal{M}$. You must define the three algorithms $\mathsf{BKeygen}, \mathsf{BEnc}$ and BDec.

2. (5 marks) Show that the scheme satisfies semantic security (as described in Figure 1). Carefully define the hybrid experiments, and show that the consecutive hybrids are computationally indistinguishable, assuming $\mathcal{E}$ is a semantically secure public key encryption scheme.

(This page is intentionally left blank for extra workspace)

Piracy detection

In addition to semantic security, you would also want to prevent piracy, especially in subscription-based model. You do not want a subscriber to create a pirate website/decrypting service that can decrypt your ciphertexts. Given access to this pirate website/decrypting service, you would like to identify the 'pirate'.

More formally, we say that a website/decrypting service $\mathcal{D}$ is a '*good pirate decryptor*' if it takes as input a ciphertext, and has the following guarantee:

$$\text{for all messages } m, \Pr[\mathcal{D}(\mathsf{ct}) = m \ : \ \mathsf{ct} \leftarrow \mathsf{Enc}(m, \mathsf{pk})] = 1$$

where the probability is over the randomness used during encryption. Assume $\mathcal{D}$ is deterministic and stateless.

A broadcast encryption scheme with piracy detection has an additional algorithm called Trace. This algorithm uses the public key pk, has oracle access to a 'good' pirate decryptor $\mathcal{D}$ (that is, it can send ciphertexts to $\mathcal{D}$ and observe the response). Intuitively, we want that if $\mathcal{D}$ is a good pirate decoder, then we should be able to use Trace to **recover the pirate subscriber**. This is formally captured by the following security game.

The piracy detection security game

- Challenger chooses $(\mathsf{pk}, (\mathsf{sk}_1, \ldots, \mathsf{sk}_t)) \leftarrow \mathsf{KeyGen}(1^n, 1^t)$. It sends pk to the adversary.
- Next, the adversary sends an index $j \in [t]$. It receives sk_j from the challenger.
- The adversary sends the pirate decrypting service $\mathcal{D}$. The challenger runs $j' \leftarrow \mathsf{Trace}^{\mathcal{D}}(\mathsf{pk})$. The adversary wins if $\mathcal{D}$ is a 'good pirate decryptor', but $j \neq j'$.

Figure 2: Piracy detection security game

3. (5 marks) Augment your construction in part 1 with a Trace algorithm. This algorithm is a randomized algorithm that has the public key pk and must identify the 'pirate' by making queries to the pirate decryptor $\mathcal{D}$. Note that the only information you have about $\mathcal{D}$ is the following:

   - it is a stateless, deterministic, polynomial time algorithm.
   - if $\mathsf{ct} \leftarrow \mathsf{Enc}(m, \mathsf{pk})$, then $\mathcal{D}(\mathsf{ct}) = m$.

   The tracing algorithm is allowed to make polynomially many queries to $\mathcal{D}$, and must use this to identify the 'pirate'.

   No proof needed for this part, just describe how Trace will work.

   (Hint: Trace can send malformed ciphertexts to $\mathcal{D}$. Of course, if $\mathcal{D}$ can figure out that it is malformed, then it may not send a correct response.)

(This page is intentionally left blank for extra workspace)

(This page is intentionally left blank for extra workspace)

(This page is intentionally left blank for extra workspace)