1 Counter-mode MAC, long messages and small signatures (20 marks)

Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a secure pseudorandom function with input space, key space and output space all equal to $\{0,1\}^n$. Consider the following MAC scheme $\mathsf{MAC} = (\mathsf{Sign}, \mathsf{Verify})$ with message space $(\{0,1\}^n)^*$ (that is, any message is of $n \cdot k$ bits for some positive integer k) and key space $\{0,1\}^n$.

- $\mathsf{Sign}(m,k)$: Let $m = (m_1, m_2, \ldots, m_\ell)$. The signing algorithm chooses a uniformly random string $r \leftarrow \{0,1\}^{n/4}$. Next, it sets $x_i = [\mathsf{bin}(\ell)]_{n/4} \mid\mid [\mathsf{bin}(i)]_{n/4} \mid\mid r \mid\mid m_i$, computes $y_i = F(x_i, k)$ and outputs $\sigma = (r, \oplus_i\ y_i)$.
- $\mathsf{Verify}(m, \sigma, k)$: Let $m = (m_1, m_2, \ldots, m_\ell)$ and $\sigma = (r, z)$. The verification algorithm sets $x_i = [\mathsf{bin}(\ell)]_{n/4} \mid\mid [\mathsf{bin}(i)]_{n/4} \mid\mid r \mid\mid m_i$, computes $y_i = F(x_i, k)$ and checks if $z = \oplus_i\ y_i$.

1.1 The above MAC is strongly unforgeable (10 marks)

Show that the above MAC scheme is strongly unforgeable, assuming that F is a secure pseudorandom function. For the security analysis, you must define appropriate hybrid games, and formally state why the consecutive hybrids are indistinguishable. Finally, you must argue why the adversary's probability of success in the final game is negligible.

Theorem 1.1. Assuming F is a secure PRF, the above MAC scheme is strongly unforgeable.

Proof. We will prove security via a sequence of hybrid games.

Game 0 : This is the original security game.

- The challenger samples a PRF key k.
- The adversary makes polynomially many signature queries. For each query $m_i = (m_{i,1}, \ldots, m_{i,\ell})$, the challenger picks a uniformly random string r_i.

  For each $j \in [\ell]$, it sets $x_{i,j} = [\mathsf{bin}(\ell)]_{n/4} \mid\mid [\mathsf{bin}(j)]_{n/4} \mid\mid r_i \mid\mid m_{i,j}$, computes $y_{i,j} = F(x_{i,j}, k)$.

  Finally, it outputs $\sigma_i = (r_i, \oplus_j\ y_{i,j})$.
- After the signature queries, the adversary outputs a forgery (m^*, σ^*). Let $m^* = (m^*_1, \ldots, m^*_\ell)$ and $\sigma^* = (r^*, y^*)$. It wins if all the following conditions hold:
  - $(m^*, \sigma^*) \neq (m_i, \sigma_i)$ for all $i \in [q]$.
  - For each $j \in [\ell]$, let $x^*_j = [\mathsf{bin}(\ell)]_{n/4} \mid\mid [\mathsf{bin}(j)]_{n/4} \mid\mid r^* \mid\mid m^*_j$, computes $y^*_j = F(x^*_j, k)$. Finally, y^* must be equal to $\oplus_j\ y^*_j$.

Game 1 : Here, the challenger uses the random function $f : \{0,1\}^n \to \{0,1\}^n$ instead of a PRF.

- The adversary makes polynomially many signature queries. For each query $m_i = (m_{i,1}, \ldots, m_{i,\ell})$, the challenger samples a uniformly random string r_i with replacement.

  For each $j \in [\ell]$, it sets $x_{i,j} = [\mathsf{bin}(\ell)]_{n/4} \mid\mid [\mathsf{bin}(j)]_{n/4} \mid\mid r_i \mid\mid m_{i,j}$, computes $y_{i,j} = f(x_{i,j})$.

  Finally, it outputs $\sigma_i = (r_i, \oplus_j\ y_{i,j})$.
- After the signature queries, the adversary outputs a forgery (m^*, σ^*). Let $m^* = (m^*_1, \ldots, m^*_\ell)$ and $\sigma^* = (r^*, y^*)$. It wins if all the following conditions hold:
  - $(m^*, \sigma^*) \neq (m_i, \sigma_i)$ for all $i \in [q]$.
  - For each $j \in [\ell]$, let $x^*_j = [\mathsf{bin}(\ell)]_{n/4} \mid\mid [\mathsf{bin}(j)]_{n/4} \mid\mid r^* \mid\mid m^*_j$, computes $y^*_j = f(x^*_j)$. Finally, y^* must be equal to $\oplus_j\ y^*_j$.

Game 2 : Here, the challenger uses the random function $f : \{0,1\}^n \to \{0,1\}^n$ instead of a PRF.

- The adversary makes polynomially many signature queries. For each query $m_i = (m_{i,1}, \ldots, m_{i,\ell})$, the challenger samples a uniformly random string r_i without replacement.

  For each $j \in [\ell]$, it sets $x_{i,j} = [\mathsf{bin}(\ell)]_{n/4} \ || \ [\mathsf{bin}(j)]_{n/4} \ || \ r_i \ || \ m_{i,j}$, computes $y_{i,j} = f(x_{i,j})$.

  Finally, it outputs $\sigma_i = (r_i, \oplus_j \ y_{i,j})$.
- After the signature queries, the adversary outputs a forgery (m^*, σ^*). Let $m^* = (m_1^*, \ldots, m_\ell^*)$ and $\sigma^* = (r^*, y^*)$. It wins if all the following conditions hold:
  - $(m^*, \sigma^*) \neq (m_i, \sigma_i)$ for all $i \in [q]$.
  - For each $j \in [\ell]$, let $x_j^* = [\mathsf{bin}(\ell)]_{n/4} \ || \ [\mathsf{bin}(j)]_{n/4} \ || \ r^* \ || \ m_j^*$, computes $y_j^* = f(x_j^*)$. Finally, y^* must be equal to $\oplus_j \ y_j^*$.

[**TODO**: Claims to relate the success probabilities across hybrids. Finally, you also need to show that the adversary has negligible success probability in the last game. (6 marks)]

Analysis

- Indistinguishability between Game 0 and Game 1 follows from the property of indistinguishability between a PRF and random function. (Using Claim 18.02 of Lec-19)
- Indistinguishability between Game 1 and Game 2 follows from the birthday bound (probability of sampling r_i more than once in polynomial number of queries is negligible). (Theorem 20.01 of Lec-20)
- For Game 2, we prove that the probability that adversary wins is negligible.

  Proof. First, we show that the output signature of a query is uniformly random. Let the signature of i^{th} query be $\sigma_i = (r_i, \oplus_j \ y_{i,j})$. Notice, that r_i is random, now note that each $x_{i,j}$ is distinct for a message m_i $\forall j \in [l]$ (because of bin(i)). This means that $\forall j \in [l]$, y_j must be uniformly random and this implies, for $j \neq j'$, $y_{i,j} \oplus y_{i,j'}$ must be random (as respective inputs $x_{i,j}$ and $x_{i,j'}$ are different and f outputs uniformly random output on any input). This implies $\oplus_j \ y_{i,j}$ must be uniformly random in i^{th} query. Now,this means that the adversary has a probability of $\frac{1}{2^n}$ of correctly predicting the signature σ^* for a message m^*. This proves that the winning probability of adversary in this case is negligible. □

□

1.2 Concrete security (5 marks)

Suppose we are using AES-128 for the PRF. The input space, key space and output space are all $\{0,1\}^{128}$. Additionally, you are given that any algorithm that sees at most 2^{64} AES evaluations (on inputs of its choice, using the same randomly chosen key) has at most $1/2^{64}$ advantage in the PRP security game. Propose the best possible attack on the above MAC scheme.

[**TODO**: Describe the best possible attack in this scenario. Analyse the winning probability and the running time of your attack.]

The adversary in the attack maintains a table T in which it stores the random string r(in the signature) and the corresponding signature($\oplus_j \ y_j$) for each type of query done. We provide the following attack:

- **Type 1 query:** The adversary queries on $m_0||m_1$ for 2^{32} times
- **Type 2 query:** Now, the adversary queries on $m_0'||m_1'$ for 2^{32} times.

- **Type 3 query:** Now, the adversary queries on $m_0||m_1'$ for 2^{32} times.
- Finally, if the adversary gets a signature σ_1, σ_2 and σ_3 in each of the above three type of queries respectively such that a common random string r is used to sign the messages in all three of them. Then, it simply outputs $m^* = (m_0'||m_1), \sigma^* = (r, \sigma_1 \oplus \sigma_2 \oplus \sigma_3)$. Note that σ^* is a valid signature for message m^*

Analysis Let $n = 2^{32}$. We show that the probability of sampling atleast one common random string r in the three types of queries is close to 1. Note, that the probability of sampling an element a when sampling randomly n-times from a set S of size n is $1-(1-\frac{1}{n})^n$. For $n \to \infty$, this probability goes to $(1-\frac{1}{e})$. Now, probability of sampling r in all three types of queries is $(1-\frac{1}{e})^3$. This means probability p of not sampling any common random string $r = (1-(1-\frac{1}{e})^3)^n$ (Using independence over all possible random string r. There are n possible random strings). This means sampling atleast one common random string $r = 1-p$. Note that as $n \to \infty$, p approaches to 0, this means $1-p$ goes to 1 as n goes to ∞. Here, we have approximated $n = 2^{32}$ as $n \to \infty$. This proves that the adversary will most likely get atleast one common random string r that will lead to the attack.

1.3 Signing bit strings (5 marks)

Suppose we wish to support arbitrary bit-strings, instead of bit strings whose length is a multiple of n. Propose a modification of the above scheme that can support message space $\mathcal{M} = \{0,1\}^*$. Argue informally why you think the modification is a secure MAC scheme (no formal proof of security needed here).

Consider the following MAC-scheme:

- $\mathsf{Sign}(m,k)$: Let $m = (m_1, m_2, ..., m_l)$. Here, $|m_i| = n$ $\forall i \in [l-1]$ and $|m_l| \leq n$. Pad m_l with a random string x of length $t = n - |m_l|$ on the right. Now, the signing algorithm chooses a uniformly random string $r \leftarrow \{0,1\}^{n/4}$. Next, it sets $x_i = [\mathsf{bin}(\ell)]_{n/4} \;||\; [\mathsf{bin}(i)]_{n/4} \;||\; r \;||\; m_i$, computes $y_i = F(x_i, k)$ and outputs $\sigma = (x, r, \oplus_i\, y_i)$.
- $\mathsf{Verify}(m,\sigma,k)$: Let $m = (m_1, m_2, \ldots, m_\ell)$ and $\sigma = (x, r, z)$. Here, $|m_i| = n$ $\forall i \in [l-1]$ and $|m_l| \leq n$. Pad m_l with string x on the right. The verification algorithm sets $x_i = [\mathsf{bin}(\ell)]_{n/4} \;||\; [\mathsf{bin}(i)]_{n/4} \;||\; r \;||\; m_i$, computes $y_i = F(x_i, k)$ and checks if $z = \oplus_i\, y_i$.
- Informal Proof of security: The adversary cannot output a signature on a new message(message not queried before) because it does n't know the key k. Also, it cannot output a signature on some chunk of previous messages because the Sign algorithm uses the length(l) of the message.

  Now, note that the adversary cannot get a new signature for any of the queried message using the previously obtained signatures because for that it will require that the adversary uses the same (x, r) pair (since it does n't know the key k, it can't apply the PRF on any (x, r) of its choice and get a signature). And now using the same (x, r) pair for a message would produce the same signature as received by the adversary earlier, which is not a correct forgery. This proves that the defined MAC-scheme is secure.

2 CBC-MAC and its variants (10 marks)

Recall the CBC-based MAC scheme discussed in class. This construction, for fixed block-length messages, uses a PRF $F : \mathcal{X} \times \mathcal{K} \to \mathcal{X}$. Let $\mathcal{M} = \mathcal{X}^\ell$ be the message space of our MAC scheme $\mathsf{MAC}_\ell = (\mathsf{Sign}_\ell, \mathsf{Verify}_\ell)$, where Sign_ℓ and Verify_ℓ are defined below.

- $\mathsf{Sign}_\ell(m = (m_1, \dots, m_\ell) \in \mathcal{X}^\ell, k \in \mathcal{K})$: Let $t_1 = F(m_1, k)$. For all $i \in [2, \ell]$, compute $t_i = F(m_i \oplus t_{i-1}, k)$. Output t_ℓ as the final signature.
- $\mathsf{Verify}_\ell(m = (m_1, \dots, m_\ell), \sigma, k)$: Let $t_1 = F(m_1, k)$. For all $i \in [2, \ell]$, compute $t_i = F(m_i \oplus t_{i-1}, k)$. Output 1 iff $t_\ell = \sigma$.

We discussed that the above scheme is secure for fixed block-length messages.

Theorem A3.01. *Assuming F is a secure PRF scheme, for every fixed ℓ the above MAC scheme MAC_ℓ is a strongly unforgeable MAC scheme for message space $\mathcal{X}^\ell$.*

2.1 A randomized variant of the above scheme (5 marks)

Suppose we alter the scheme above, and make the signing algorithm randomized.

- $\mathsf{Sign}'_\ell(m = (m_1, \dots, m_\ell) \in \mathcal{X}^\ell, k \in \mathcal{K})$: Choose a random string $x \leftarrow \mathcal{X}$. Let $t_1 = F(m_1 \oplus x, k)$. For all $i \in [2, \ell]$, compute $t_i = F(m_i \oplus t_{i-1}, k)$. Output (x, t_ℓ) as the final signature.

The verification algorithm can be appropriately defined.

[**TODO**: describe a forgery that works **even for fixed length messages**.]

Let $m = m_1, m_2, \cdots, m_l$. Send 1 signing query for message m.
We get $\sigma = \mathsf{Sign}'_\ell(m, k) = (x, t_l)$

Forgery(m^*, σ^*): $m^* = m'_1, m'_2, \cdots, m'_l$, where $m'_1 = x$ and $\forall i \in [2, l], m'_i = m_i$
$\sigma^* = (x', t'_l)$ where Randomness $x' = m_1$ and $t'_l = t_l$
Length of message m^* = length of message m
Clearly, $(m^*, \sigma^*) \notin \{(m, \sigma)\}$ as $m \neq m^*$
[If $m_1 = x$, then send the same query m again and again till you get $m_1 \neq x$]

For message m^* and key k, random string $x' = m_1$,
$t'_1 = F(m'_1 \oplus x', k) = F(x \oplus m_1, k)$
Since $m_1 \oplus x = x \oplus m_1$, therefore $t'_1 = t_1$.
Since $\forall i \in [2, l], m'_i = m_i$ and $t'_1 = t_1$, therefore $\forall i \in [1, l] t'_i = t_i$. In particular $t'_l = t_l$.
Thus $\mathsf{Verify}'_\ell(m^*, \sigma^*) = 1$ Hence, (m^*, σ^*) is a valid forgery. ■

2.2 Handling unbounded length messages (5 marks)

There are a few easy modifications for handling unbounded block-length messages. One of them is described below. It gives us a MAC scheme with message space $\mathcal{X}^*$.

- $\mathsf{Sign}^*(m = (m_1, \dots, m_r), k)$: Let $[r]_\mathcal{X}$ denote some canonical representation of the length r as an element in $\mathcal{X}$. For instance, if $\mathcal{X} = \{0, 1\}^n$, then this would simply be the binary representation of r. Let $m_0 = [r]_\mathcal{X}$, and $m^* = (m_0, m_1, \dots, m_r)$. Output $\sigma \leftarrow \mathsf{Sign}_{r+1}(m^*, k)$ as the signature.

Verification can be defined appropriately, and this gives us a secure MAC scheme for message space $\mathcal{X}^*$. It is crucial that the message block-length is **prepended** before signing. Consider the following variant where we **append** the block-length:

- $\mathsf{Sign}'(m = (m_1, \ldots, m_r), k)$: Let $[r]_\mathcal{X}$ denote some canonical representation of the length r as an element in $\mathcal{X}$. Let $m_{\ell+1} = [r]_\mathcal{X}$, and $m' = (m_1, \ldots, m_\ell, m_{r+1})$. Output $\sigma' \leftarrow \mathsf{Sign}_{r+1}(m', k)$ as the signature.

[**TODO**: Show a forgery for the above MAC scheme.]

We can get a forgery using 3 queries.

Let us assume $\mathcal{X}$ has atleast 2 strings. If $|\mathcal{X}| = 0$, then Message Space is empty. If $|\mathcal{X}| = 1$, then PRF $F : \mathcal{X} \times \mathcal{K} \to \mathcal{X}$ always maps the single element in $\mathcal{X}$ to itself for any key. Then (x, x) is a valid forgery without any queries where $x \in \mathcal{X}$. Thus, we assume $\mathcal{X}$ has atleast 2 strings. Another observation is that signature under Sign_ℓ of any message is also a string in $\mathcal{X}$

Query 1: Let $p \in \mathcal{X}$ be some message. Send $m = p$ as a message of 1 block length
We get $\sigma_1 = \mathsf{Sign}^*(m, k) = F(F(p, k) \oplus [1]_\mathcal{X}, k)$

Query 2: Let $q \in \mathcal{X}, q \neq p$ be some message. Since $|\mathcal{X}| \geq 2$, therefore such p, q exist. Send $m' = q$ as a message of 1 block length
We get $\sigma_2 = \mathsf{Sign}^*(m', k) = F(F(q, k) \oplus [1]_\mathcal{X}, k)$

Query 3: σ_1, σ_2 are also of 1 block length each.
Let $m'' = (m''_1, m''_2, m''_3)$ where $m''_1 = p,\ m''_2 = [1]_\mathcal{X}$ and $m''_3 = \sigma_1$.
Send m'' as a message of 3 block length.
Let $\lambda_1 = F(p, k)$,
$\lambda_2 = F(\lambda_1 \oplus [1]_\mathcal{X}, k)$,
$\lambda_3 = F(\lambda_2 \oplus \sigma_1, k)$,
$\lambda_4 = F(\lambda_3 \oplus [3]_\mathcal{X}, k)$
$\mathsf{Sign}^*(m'', k) = \lambda_4$ as per defintion of Sign^*
Now, $\lambda_2 = F(F(p, k) \oplus [1]_\mathcal{X}, k) \ = \ \sigma_1$
This implies $\lambda_3 = F(\lambda_2 \oplus \sigma_1, k) = F(\sigma_1 \oplus \sigma_1, k) = F([0]_\mathcal{X}, k)$
$\Rightarrow \lambda_4 = F(F([0]_\mathcal{X}, k) \oplus [3]_\mathcal{X}, k)$
$\Rightarrow \sigma_3 = \mathsf{Sign}^*(m'', k) = \lambda_4 = F(F([0]_\mathcal{X}, k) \oplus [3]_\mathcal{X}, k)$

Forgery(m^*, σ^*): $m^* = (m^*_1, m^*_2, m^*_3)$ where $m^*_1 = q,\ m^*_2 = [1]_\mathcal{X}$ and $m^*_3 = \sigma_2$
$\sigma^* = \sigma_3$
Clearly $(m^*, \sigma^*) \notin \{(m, \sigma_1), (m', \sigma_2), (m'', \sigma_3)\}$

Let $\lambda'_1 = F(q, k)$,
$\lambda'_2 = F(\lambda'_1 \oplus [1]_\mathcal{X}, k)$,
$\lambda'_3 = F(\lambda'_2 \oplus \sigma_2, k)$,
$\lambda'_4 = F(\lambda'_3 \oplus [3]_\mathcal{X}, k)$
$\mathsf{Sign}^*(m^*, k) = \lambda'_4$ as per defintion of Sign^*
Now, $\lambda'_2 = F(F(q, k) \oplus [1]_\mathcal{X}, k) \ = \ \sigma_2$
This implies $\lambda'_3 = F(\lambda'_2 \oplus \sigma_2, k) = F(\sigma_2 \oplus \sigma_2, k) = F([0]_\mathcal{X}, k)$
$\Rightarrow \lambda'_3 = \lambda_3$
$\Rightarrow \lambda'_4 = F(F([0]_\mathcal{X}, k) \oplus [3]_\mathcal{X}, k) = \lambda_4$
$\Rightarrow \sigma_4 = \mathsf{Sign}^*(m^*, k) = \lambda'_4 = F(F([0]_\mathcal{X}, k) \oplus [3]_\mathcal{X}, k) = \sigma_3 = \sigma^*$

Therefore, $\mathsf{Verify}^*(m^*, \sigma^*) = 1$. Hence, (m^*, σ^*) is a valid forgery. ■

3 Semantic Security: Equivalent Definitions (20 marks)

3.1 Equivalence of query-based semantic security and semantic security (10 marks)

In the quiz, you had shown that semantic security implies pre-challenge query-based semantic security. A similar reduction can be used to show that semantic security also implies query-based semantic security.

In this exercise, we will show that query-based semantic security is equivalent to semantic security. In particular, show that if an encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ satisfies **query-based semantic security**, then it also satisfies **semantic security** (Definition 15.01 in Lecture 15).

For simplicity, you can assume the adversary makes at most q queries in the semantic security game.

Theorem 3.1. If an encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ satisfies **query-based semantic security**, then it also satisfies **semantic security** (Definition 15.01 in Lecture 15).

Proof. Consider any p.p.t. adversary $\mathcal{A}$ that makes q queries to the semantic security challenger. Each query consists of a pair of messages $(m_{i,0}, m_{i,1})$. In World-0, the challenger encrypts $m_{i,0}$ for all i, while in World-1, the challenger encrypts $m_{i,1}$ for all i. We will define intermediate hybrids, and use the query-based semantic security to show that the intermediate hybrids are indistinguishable.

Hybrids required for proving equivalence of definitions

World-0: Let p_0 denote the probability of $\mathcal{A}$ outputting 0 in this world.

- **Setup phase:** Challenger chooses a key k.
- **Query phase:** Adversary $\mathcal{A}$ sends q queries. For the i^{th} query, it sends $(m_{i,0}, m_{i,1})$ and receives $\mathsf{Enc}(m_{i,0}, k)$.
- **Guess:** Finally, $\mathcal{A}$ outputs its guess b'.

[**TODO**: Define intermediate hybrids such that Hybrid-0 corresponds to World-0, and the final hybrid corresponds to World-1. Let $p_{\text{hyb},i}$ denote the probability of $\mathcal{A}$ outputting 0 in the i^{th} hybrid. (5 marks)]

We define $q+1$ hybrids $\mathcal{H}_0, \mathcal{H}_1, \cdots, \mathcal{H}_q$, where $\forall i \in [0, q]$, Hybrid $\mathcal{H}_i$ is defined as follows:

Hybrid $\mathcal{H}_i$: Let $p_{hyb,i}$ denote the probability of $\mathcal{A}$ outputting 0 in this world.

- **Setup phase:** Challenger chooses a key k.
- **Query phase:** Adversary $\mathcal{A}$ sends q queries. For the first $q - i$ queries, it sends $(m_{j,0}, m_{j,1})$ and receives $\mathsf{Enc}(m_{j,0}, k)$. For the last i queries, it sends $(m_{j,0}, m_{j,1})$ and receives $\mathsf{Enc}(m_{j,1}, k)$.
- **Guess:** Finally, $\mathcal{A}$ outputs its guess b'.

World-1: Let p_1 denote the probability of $\mathcal{A}$ outputting 0 in this world.

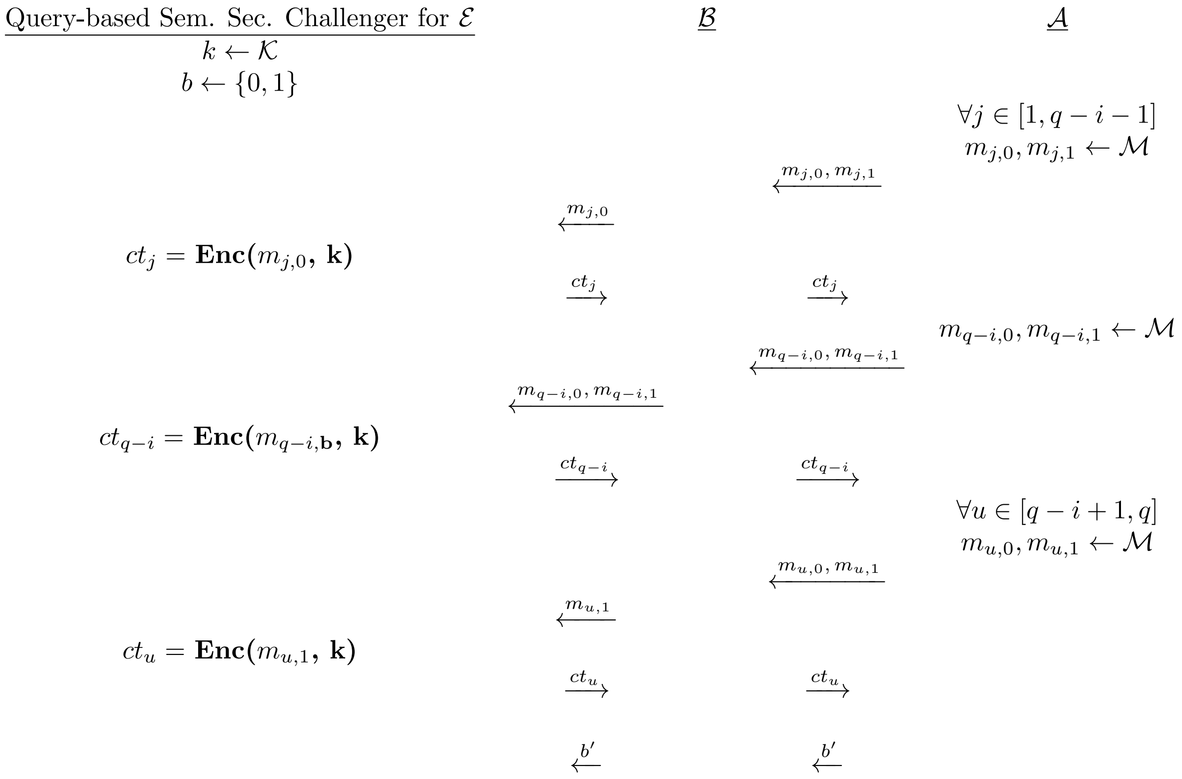
- **Setup phase:** Challenger chooses a key k.
- **Query phase:** Adversary $\mathcal{A}$ sends q queries. For the i^{th} query, it sends $(m_{i,0}, m_{i,1})$ and receives $\mathsf{Enc}(m_{i,1}, k)$.
- **Guess:** Finally, $\mathcal{A}$ outputs its guess b'.

Analysis Let $p_{\text{hyb},i}$ denote the probability of $\mathcal{A}$ outputting 0 in the i^{th} hybrid. Note that $p_{\text{hyb},0} = p_0$.

Claim 3.2. [**TODO**: complete the claim]. Assuming $\mathcal{E}$ satisfies query-based semantic security, if $p_{hyb,i} - p_{hyb,i+1}$ is non-negligible for some $i \in [1, q]$, then there exists a p.p.t. adversary $\mathcal{B}$ that wins the **query-based semantic security** game corresponding to $\mathcal{E}$ with non-negligible probability.

Proof. [**TODO**: prove the above claim via appropriate reduction. You can skip the reduction's success probability analysis. (5 marks)]

Let us assume $p_{hyb,i} - p_{hyb,i+1} = \epsilon$ for some $i \in [1, q]$.
Adversary $\mathcal{A}$ can distinguish between hybrids $\mathcal{H}_{i+11}$ and $\mathcal{H}_i$ with probability ϵ
Consider the following p.p.t. adversary $\mathcal{B}$.


Query-based Sem. Sec. Challenger for $\mathcal{E}$
$\mathcal{B}$
$\mathcal{A}$
$k \leftarrow \mathcal{K}$
$b \leftarrow \{0,1\}$
$\forall j \in [1, q-i-1]$
$m_{j,0}, m_{j,1} \leftarrow \mathcal{M}$
$m_{j,0}, m_{j,1}$
$m_{j,0}$
$ct_j = \mathbf{Enc}(m_{j,0}, \mathbf{k})$
ct_j
ct_j
$m_{q-i,0}, m_{q-i,1} \leftarrow \mathcal{M}$
$m_{q-i,0}, m_{q-i,1}$
$m_{q-i,0}, m_{q-i,1}$
$ct_{q-i} = \mathbf{Enc}(m_{q-i,\mathbf{b}}, \mathbf{k})$
ct_{q-i}
ct_{q-i}
$\forall u \in [q-i+1, q]$
$m_{u,0}, m_{u,1} \leftarrow \mathcal{M}$
$m_{u,0}, m_{u,1}$
$m_{u,1}$
$ct_u = \mathbf{Enc}(m_{u,1}, \mathbf{k})$
ct_u
ct_u
b'
b'

- Step 1: Challenger chooses key $k \leftarrow \mathcal{K}$ and random bit $b \leftarrow \{0, 1\}$.
- Step 2: Adversary $\mathcal{A}$ sends first $q - i - 1$ queries.
- Step 3: For each of these first $q - i - 1$ queries, adversary $\mathcal{B}$ sends $m_{j,0}$ to the Challenger
- Step 4: Challenger sends the encryption of $m_{j,0}$ under key k for each of these $q - i - 1$ queries to $\mathcal{B}$
- Step 5: $\mathcal{B}$ forwards the ciphertext received from Challenger as reply to $\mathcal{A}$'s queries.
- **Steps 2-5** correspond to polynomially many **pre-challenge** queries in the query-based semantic security game. To adversary $\mathcal{A}$ these first $q - i - 1$ queries seem like interaction in semantic security game for hybrids $\mathcal{H}_{i+1}, \mathcal{H}_i$ where he gets back encryption of $m_{j,0}$ for each of these queries.
- Step 6: Adversary $\mathcal{A}$ sends query consisting $m_{q-i,0}, m_{q-i,1}$ to $\mathcal{B}$
- Step 7: $\mathcal{B}$ forwards both the messages $m_{q-i,0}, m_{q-i,1}$ to the Challenger.

- Step 8: Challenger sends the encryption of $m_{q-i,b}$ under key k to $\mathcal{B}$. Here, Challenger uses the random bit b.
- Step 9: $\mathcal{B}$ forwards the ciphertext received from Challenger as reply to $\mathcal{A}$'s query.
- **Steps 6-9** correspond to the **Challenge phase** in the query-based semantic security game. To adversary $\mathcal{A}$ this seems like interaction in semantic security game for hybrids $\mathcal{H}_{i+1}, \mathcal{H}_i$ where he gets back encryption of m_0 in one case and encryption of m_1 in other.
- Step 10: Adversary $\mathcal{A}$ sends the last $i-1$ queries.
- Step 11: For each of these last $i-1$ queries, adversary $\mathcal{B}$ sends $m_{u,1}$ to the Challenger
- Step 12: Challenger sends the encryption of $m_{u,1}$ under key k for each of these $i-1$ queries to $\mathcal{B}$
- Step 13: $\mathcal{B}$ forwards the ciphertext received from Challenger as reply to $\mathcal{A}$'s queries.
- **Steps 10-13** correspond to polynomially many **post-challenge** queries in the query-based semantic security game. To adversary $\mathcal{A}$ these last $i-1$ queries seem like interaction in semantic security game for hybrids $\mathcal{H}_{i+1}, \mathcal{H}_i$ where he gets back encryption of $m_{u,1}$ for each of these queries.
- Step 14: Finally $\mathcal{A}$ sends his guess b' to $\mathcal{B}$.
- Step 15: $\mathcal{B}$ forwards the bit b' received from $\mathcal{A}$ as his guess to challenger.

Since, $\mathcal{B}$ just transmits messages between challenger and $\mathcal{A}$, and $\mathcal{A}$ makes polynomially many queries, therefore $\mathcal{B}$ is a p.p.t. adversary.
With probability $1/q$, the hybrids $\mathcal{H}_{i+1}, \mathcal{H}_i$ are distinguishable.
If b=0, then adversary $\mathcal{A}$ is in hybrid $\mathcal{H}_i$, and if b =1, then adversary $\mathcal{A}$ is in hybrid $\mathcal{H}_{i+1}$.
Pr[b'=0 | b= 0]= $p_{hyb,i}/q$
Pr[b'=0 | b = 1] = $p_{hyb,i+1}/q$
Since, we have assumed $\mathcal{E}$ to satisfy query-based semantic security, therefore $p_{hyb,i} - p_{hyb,i+1}$ is negligible for all $i \in [1, q]$

□

This implies if $\mathcal{E}$ satisfies query-based semantic security, then consecutive hybrids are indistinguishable, and consequently World-0 and World-1 are indistinguishable.

Thus, **query-based semantic security** implies **semantic security** □

3.2 Equivalence of pre-challenge query-based semantic security and semantic security (10 marks)

Somewhat surprisingly, we can show that the post-challenge queries are not very useful. Pre-challenge query-based semantic security, query-based semantic security and semantic security are all equivalent security definitions! Show that if there exists a p.p.t. adversary $\mathcal{A}$ that breaks query-based semantic security, then there exists a p.p.t reduction algorithm $\mathcal{B}$ that breaks pre-challenge query-based security.

Note tha the reduction algorithm is not allowed to make any queries to the challenger after it receives the challenge ciphertext. However, it must somehow respond to the adversary's post-challenge queries. For simplicity, you can make the following assumptions:

- The message space is $\{0,1\}^n$. However, you must not assume that the encryption scheme encrypts the message bit-by-bit.
- The adversary makes at most q post-challenge queries.

Theorem 3.3. Suppose there exists a p.p.t. adversary $\mathcal{A}$ that makes q_{pre} pre-challenge queries, q post-challenge queries, and wins the query-based semantic security game with probability $1/2 + \epsilon$, where ϵ is non-negligible. Then there exists a p.p.t. algorithm $\mathcal{B}$ that makes $q_{\text{pre}} + q$ pre-challenge queries (no post-challenge queries), and wins the pre-challenge query-based semantic security game with probability $1/2 + \ldots$

Proof. We will first construct a sequence of hybrid experiments between world-0 and world-1, and then show that if any two hybrids are distinguishable, then there exists a reduction algorithm $\mathcal{B}$ that breaks the pre-challenge query-based semantic security. The unified reduction algorithm can be obtained by guessing the hybrids which are 'most-distinguishable'.[1]

Hybrid experiments:

World-0 $\equiv$ Hybrid$_{0,0}$: In this experiment, the challenger responds to pre and post-challenge encryption queries made by the adversary (it uses the same key for all queries). For the challenge messages (m_0^*, m_1^*), it encrypts m_0^*.

- **Setup:** Challenger chooses key k.
- **Pre-challenge query phase:** The adversary makes q_{pre} encryption queries in this phase. For each encryption query m_i, the challenger sends $\mathsf{ct}_i \leftarrow \mathsf{Enc}(m_i, k)$.
- **Challenge phase:** The adversary sends two challenge messages (m_0^*, m_1^*). The challenger sends $\mathsf{ct}^* \leftarrow \mathsf{Enc}(m_0^*, k)$.
- **Post-challenge query phase:** The adversary makes q encryption queries in this phase. For each encryption query m_i', the challenger sends $\mathsf{ct}_i' \leftarrow \mathsf{Enc}(m_i', k)$.
- **Guess:** Finally, the adversary sends its guess b'.

[**TODO**: define intermediate hybrids] We define the following hybrids:
Hybrid$_{i,0}$ for $i \in [1, q]$ (Here q is the number of post-challenge queries)

- **Setup:** Challenger chooses key k.
- **Pre-challenge query phase:** The adversary makes q_{pre} encryption queries in this phase. For each encryption query m_i, the challenger sends $\mathsf{ct}_i \leftarrow \mathsf{Enc}(m_i, k)$.
- **Challenge phase:** The adversary sends two challenge messages (m_0^*, m_1^*). The challenger sends $\mathsf{ct}^* \leftarrow \mathsf{Enc}(m_0^*, k)$.
- **Post-challenge query phase:** The adversary makes q encryption queries in this phase. In each query, adversary $\mathcal{A}$ sends m_i' and receives $\mathsf{Enc}(m_i', k)$ as the ciphertext for the first $(q - i)$ queries and for the last i queries, the challenger chooses a random $m \leftarrow \mathcal{M}$ and sends the ciphertext $\mathsf{Enc}(m, k)$.
- **Guess:** Finally, the adversary sends its guess b'.

Hybrid$_{i,1}$ for $i \in [1, q]$ (Here q is the number of post-challenge queries)

- **Setup:** Challenger chooses key k.
- **Pre-challenge query phase:** The adversary makes q_{pre} encryption queries in this phase. For each encryption query m_i, the challenger sends $\mathsf{ct}_i \leftarrow \mathsf{Enc}(m_i, k)$.
- **Challenge phase:** The adversary sends two challenge messages (m_0^*, m_1^*). The challenger sends $\mathsf{ct}^* \leftarrow \mathsf{Enc}(m_1^*, k)$.
- **Post-challenge query phase:** The adversary makes q encryption queries in this phase. In each query, adversary $\mathcal{A}$ sends m_i' and receives $\mathsf{Enc}(m_i', k)$ as the ciphertext for the first $(q - i)$ queries and for the last i queries, the challenger chooses a random $m \leftarrow \mathcal{M}$ and sends the ciphertext $\mathsf{Enc}(m, k)$.
- **Guess:** Finally, the adversary sends its guess b'.

[1] The unified reduction is not needed for the assignment.

World-1 $\equiv$ Hybrid$_{0,1}$: In this experiment, the challenger responds to pre and post-challenge encryption queries made by the adversary (it uses the same key for all queries). For the challenge messages (m_0^*, m_1^*), it encrypts m_1^*.

- **Setup:** Challenger chooses key k.
- **Pre-challenge query phase:** The adversary makes q_{pre} encryption queries in this phase. For each encryption query m_i, the challenger sends $\mathsf{ct}_i \leftarrow \mathsf{Enc}(m_i, k)$.
- **Challenge phase:** The adversary sends two challenge messages (m_0^*, m_1^*). The challenger sends $\mathsf{ct}^* \leftarrow \mathsf{Enc}(m_1^*, k)$.
- **Post-challenge query phase:** The adversary makes q encryption queries in this phase. For each encryption query m_i', the challenger sends $\mathsf{ct}_i' \leftarrow \mathsf{Enc}(m_i', k)$.
- **Guess:** Finally, the adversary sends its guess b'.

Analysis: □

[**TODO**: Show indistinguishability of hybrids. You should have a sequence of formal claims, followed by a **precise** description of the reduction algorithm for each one. You can skip the reduction's success probability analysis.]

Indistinguishability between Hybrid$_{i,0}$ and Hybrid$_{i+1,0}$

We show that if an adversary $\mathcal{A}$ can distinguish between Hybrid$_{i,0}$ and Hybrid$_{i+1,0}$, then $\exists$ a reduction $\mathcal{B}$ that can break the pre-challenge semantic security of encryption scheme $\mathcal{E}$.

What $\mathcal{B}$ and $\mathcal{A}$ does

- First, $\mathcal{B}$ makes $(i+1)$ queries on some randomly sampled $m' \leftarrow \mathcal{M}$ to the challenger, receives ciphertext ct and stores the (m', ct) pair in table $\mathcal{T}$. In each query, m' is being queried.

- Next, $\mathcal{A}$ makes q_{pre} pre-challenge queries of the form m_j and $\mathcal{B}$ sends them to the Challenger of pre-challenge semantic game. $\mathcal{B}$ forwards the received ciphertext ct_j to $\mathcal{A}$.

- Now, the adversary $\mathcal{A}$ sends the challenge query (m_0^*, m_1^*) and $\mathcal{B}$ forwards only m_0^* to the challenger. $\mathcal{B}$ forwards the received ciphertext ct_0 to $\mathcal{A}$.

- Now, the adversary $\mathcal{A}$ starts sending the post-challenge queries m_j' and $\mathcal{B}$ simply forwards the first $(q-i-1)$ queries to challenger as a pre-challenge query. $\mathcal{B}$ forwards the received ciphertext ct_j' to $\mathcal{A}$.

- At the $(q-i)^{th}$ query, the adversary $\mathcal{A}$ sends m_j' and $\mathcal{B}$ forwards (m_j', m') to the challenger as a challenge query. Challenger chooses a random bit $b \leftarrow \{0,1\}$ and sends the ciphertext ct_b to $\mathcal{B}$. The reduction $\mathcal{B}$ forwards the received ciphertext ct_b to $\mathcal{A}$.

- For the remaining i queries, the adversary$\mathcal{A}$ sends m_j' and $\mathcal{B}$ sends the stored encryption of message m' in table $\mathcal{T}$ to the adversary $\mathcal{A}$.

Claim Hybrid$_{i,0}$ and Hybrid$_{i+1,0}$ are indistinguishable.

Proof. In the above game, when $b = 0$, the adversary is in Hybrid$_{i,0}$ and when $b = 1$, the adversary is in Hybrid$_{i+1,0}$. Therefore, $\Pr(b' = 0|b = 0)$ corresponds to $p_0 = \Pr(b' = 0)$ in Hybrid$_{i,0}$ and $\Pr(b' = 0|b = 1)$ corresponds to $p_1 = \Pr(b' = 0)$ in Hybrid$_{i+1,0}$. If $p_0 - p_1$ is non-negl., then $\Pr(b' = 0|b = 0)$ - $\Pr(b' = 0|b = 1)$ is also non-negl. and $\mathcal{B}$ can break the semantic security of $\mathcal{E}$. □

This proves that Hybrid$_{i,0}$ and Hybrid$_{i+1,0}$ must be indistinguishable for $\mathcal{E}$ to remain pre-challenge semantically secure.

Indistinguishability between Hybrid$_{i,1}$ and Hybrid$_{i+1,1}$ follows similiarly.

Indistinguishability between Hybrid$_{q,0}$ and Hybrid$_{q,1}$

What $\mathcal{B}$ and $\mathcal{A}$ does

- First, $\mathcal{B}$ makes q queries on some randomly sampled $m' \leftarrow \mathcal{M}$ to the challenger, receives ciphertext ct and stores the (m', ct) pair in table $\mathcal{T}$. In each query, m' is being queried.
- Next, $\mathcal{A}$ makes q_{pre} pre-challenge queries of the form m_j and $\mathcal{B}$ sends them to the Challenger of pre-challenge semantic game. $\mathcal{B}$ forwards the received ciphertext ct_j to $\mathcal{A}$.
- Now, the adversary $\mathcal{A}$ sends the challenge query (m_0^*, m_1^*) and $\mathcal{B}$ forwards (m_0^*, m_1^*) to the challenger. $\mathcal{B}$ forwards as a challenge query. The challenger sends ct_b to $\mathcal{B}$ and $\mathcal{B}$ forwards ct_b to the adversary $\mathcal{A}$.
- Now, the adversary $\mathcal{A}$ starts sending the post-challenge queries m_j' and $\mathcal{B}$ sends the stored encryption of message m' in table $\mathcal{T}$ to the adversary $\mathcal{A}$.

Proof. In the above game, when $b = 0$, the adversary is in Hybrid$_{q,0}$ and when $b = 1$, the adversary is in Hybrid$_{q,1}$. Therefore, $\Pr(b' = 0|b = 0)$ corresponds to $p_0 = \Pr(b' = 0)$ in Hybrid$_{q,0}$ and $\Pr(b' = 0|b = 1)$ corresponds to $p_1 = \Pr(b' = 0)$ in Hybrid$_{q,1}$. If $p_0 - p_1$ is non-negl., then $\Pr(b' = 0|b = 0)$ - $\Pr(b' = 0|b = 1)$ is also non-negl. and $\mathcal{B}$ can break the pre-challenge semantic security of $\mathcal{E}$. □

Hence, proved.