Advanced Cryptography

Spring Semester 2021

Homework 2

- This homework contains two questions: (1) an introduction to the Random Oracle Model (ROM) and (2) relations between several cryptographic primitives.
- You will submit a **report** that will contain all your answers and explanations. The report should be a PDF document. You can use any editor to prepare the report, but Latex is usually the best choice for typesetting math and pseudocode.
- We ask you to **work alone or in groups of 2**. No collaborations are allowed outside of the group you registered for HW1. Please contact the T.A. if you have a good reason to change group. Feel free to ask questions to the T.A.
- We might announce some typos for this homework on Moodle in the news forum. Everybody is subscribed to it and does receive an email as well. If you decided to ignore Moodle emails we recommend that you check the forum regularly.
- The homework is due on Moodle on Monday, 26^{th} of April at 23h59. Please submit 1 report per group.

1 Introduction to Random Oracles

In this exercise, you will get familiar with the concept of the random oracle model (ROM). In short, in the ROM, every call to a hash function $\{0,1\}^{\ell_1} \mapsto \{0,1\}^{\ell_2}$ is replaced by a call to a perfectly random function over the same domains, accessed as an oracle. In the context of a security game, this can be illustrated by the following example, where Ψ_{ℓ_1,ℓ_2} is the set of functions from $\{0,1\}^{\ell_1}$ to $\{0,1\}^{\ell_2}$.

$\Gamma(\mathcal{A})$	Oracle $H(x)$
some steps where $H(\cdot)$ might be called	// before first query:
$out \leftarrow\!\!\$\ \mathcal{A}^H(input)$	sample $\mathcal{H}$ uniformly from Ψ_{ℓ_1,ℓ_2}
final steps of game where $H(\cdot)$ might be called	// on a query:
	return $\mathcal{H}(x)$

That is, a random function is sampled at the beginning of the game, then all calls to $\mathcal{H}$ are made through the oracle H, which we call the *random oracle*. Note that the adversary is also given access to the oracle.

Question 1. Sampling a random function from Ψ_{ℓ_1,ℓ_2} cannot usually be done in polynomial time, which leads to a problem when one wants to simulate the random oracle in a ppt reduction. Show that you can modify the random oracle H s.t. its distribution stays the same but it runs in polynomial time (in the number of queries).
Hint: Could you generate the values $H(x)$ on the fly?

(2pts) We simply sample uniformly at random the value $H(x)$ when x is queried for the first time. All values are independently and uniformly sampled at random as in the real random oracle H.

Oracle $H(x)$

// before first query:
$\mathcal{L} \leftarrow []$
// on a query:
if $x \in \mathcal{L}$:
  return $\mathcal{L}[x]$
$y \leftarrow\!\!\$\ \{0,1\}^{\ell_2}$
$\mathcal{L}[x] \leftarrow y$
return y

Question 2. We consider the following two games, where H is a random oracle.

$\Gamma^0(\mathcal{A})$	$\Gamma^1(\mathcal{A})$
$x \leftarrow\!\!\$\, \{0,1\}^{\ell_1}$	$x \leftarrow\!\!\$\, \{0,1\}^{\ell_1}$
$y \leftarrow H(x)$	$y \leftarrow\!\!\$\, \{0,1\}^{\ell_2}$
$b' \leftarrow\!\!\$\, \mathcal{A}^H(x,y)$	$b' \leftarrow\!\!\$\, \mathcal{A}^H(x,y)$
return b'	**return** b'

Show that

$$\left|\Pr[\Gamma^0(\mathcal{A}) \Rightarrow 1] - \Pr[\Gamma^1(\mathcal{A}) \Rightarrow 1]\right| \leq \Pr[\mathsf{query}] \ ,$$

where query is the event *"$\mathcal{A}$ queries x to H"*.

(3pts) Let the game Γ' be the same as Γ^1 except we abort (or return 0 or 1 with prob. $\frac{1}{2}$) whenever query happens. Then, by a similar argument to the difference lemma we have

$$|\Pr[\Gamma'(\mathcal{A}) \Rightarrow 1] - \Pr[\Gamma^1(\mathcal{A}) \Rightarrow 1]| \leq \frac{1}{2}\Pr[\mathsf{query}] \ .$$

Now, we modify Γ' s.t. we replace the value $H(x)$ with y and we call this game Γ''. Clearly, this game is the same as Γ^0, except we abort when query occurs. Since $\mathcal{A}$ cannot query $H(x)$, it cannot tell whether $H(x) = y$ or not. Thus, $\Pr[\Gamma'(\mathcal{A}) \Rightarrow 1] = \Pr[\Gamma''(\mathcal{A}) \Rightarrow 1]$. Applying again the difference lemma trick as above we have

$$|\Pr[\Gamma^0(\mathcal{A}) \Rightarrow 1] - \Pr[\Gamma''(\mathcal{A}) \Rightarrow 1]| \leq \frac{1}{2}\Pr[\mathsf{query}] \ .$$

Collecting the probabilities holds the result. We note that the probability of the event query in game Γ^0 is the same as the prob. that query happens in game Γ^1. This follows from the fact that as long query does not occur, the view of the adversary $\mathcal{A}$: $(x, y, H(x_1), \ldots, H(x_n))$ with $x_i \neq x, i \in \{1, \ldots, n\}$ is identically distributed in both games.

We now define the notion of *Key Encapsulation Mechanism*, which is somewhat similar to the notion of PKC.

Definition 1. A KEM is a tuple of three algorithms $(\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$.

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\, \mathsf{Gen}(1^\lambda)$: The generation algorithm outputs a pair of public and secret keys.
- $\mathsf{ct}, K \leftarrow\!\!\$\, \mathsf{Encaps}(\mathsf{pk})$: The encapsulation algorithm takes the public key as input and outputs a ciphertext and a key.
- $K' \leftarrow\!\!\$\, \mathsf{Decaps}(\mathsf{sk}, \mathsf{ct})$: The decapsulation algorithm takes the secret key and a ciphertext as inputs and outputs a key.

Intuitively, the KEM is correct if we run these three algorithms in order and $K' = K$. More formally, a KEM is correct if $\Pr[\mathrm{CORR}_{\mathsf{KEM}} \Rightarrow \mathbf{true}] = 1$, where CORR is the game defined in Fig. 1.

The corresponding IND-CPA definition is defined as follows.

$\underline{\text{CORR}_{\mathsf{KEM}}}$

$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$
$\mathsf{ct}, K \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk})$
$K' \leftarrow \mathsf{Decaps}(\mathsf{sk}, \mathsf{ct})$
return $1_{K=K'}$

Figure 1: KEM correctness.

$\underline{\text{IND-CPA}^0_{\mathsf{KEM}}(\mathcal{A})}$	$\underline{\text{IND-CPA}^1_{\mathsf{KEM}}(\mathcal{A})}$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$	$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{Gen}(1^\lambda)$
$\mathsf{ct}^*, K^* \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk})$	$\mathsf{ct}^*, K \leftarrow\!\!\$\ \mathsf{Encaps}(\mathsf{pk})$
$b' \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}, \mathsf{ct}^*, K^*)$	$K^* \leftarrow\!\!\$\ \{0,1\}^{\ell_2}$
return b'	$b' \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}, \mathsf{ct}^*, K^*)$
	return b'

Figure 2: KEM IND-CPA games.

Definition 2 (KEM IND-CPA)**.** We say a KEM KEM is IND-CPA if for all ppt adversary $\mathcal{A}$ the advantage

$$\mathsf{Adv}^{\text{ind-cpa}}_{\mathcal{A},\mathsf{KEM}} := \left|\Pr[\text{IND-CPA}^1{}_{\mathsf{KEM}}(\mathcal{A}) \Rightarrow 1] - \Pr[\text{IND-CPA}^0{}_{\mathsf{KEM}}(\mathcal{A}) \Rightarrow 1]\right|$$

is negligible in λ and the games IND-CPA^b are defined in Figure 2. Note that in the ROM, the adversary can query the random oracle (H in our previous example) in addition.

Question 3. We define a KEM $\mathsf{KEM} = (\mathsf{Gen}, \mathsf{Encaps}, \mathsf{Decaps})$ from a PKC $\mathsf{PKC} = (\mathsf{gen}, \mathsf{enc}, \mathsf{dec})$ and a random oracle H as follows.

- $\mathsf{Gen} := \mathsf{gen}$.
- $\mathsf{Encaps}(\mathsf{pk})$:
  1. Sample $\mathsf{pt} \leftarrow\!\!\$\ \{0,1\}^{\ell_1}$
  2. Compute $\mathsf{ct} \leftarrow \mathsf{enc}(\mathsf{pk}, \mathsf{pt})$.
  3. Output $\mathsf{ct}, H(\mathsf{pt})$
- $\mathsf{Decaps}(\mathsf{sk}, \mathsf{ct})$:
  1. Compute $\mathsf{pt}' \leftarrow \mathsf{dec}(\mathsf{sk}, \mathsf{ct})$.
  2. Output $H(\mathsf{pt}')$.

a) Rewrite explicitly the KEM IND-CPA game (see Fig. 2) for the KEM defined above.
b) Prove that if the underlying PKC PKC used in the KEM is one-way (i.e. CPA security against decryptions or OW-CPA, see last homework), then KEM is IND-CPA.

Hint: As our KEM is defined in the ROM, a KEM IND-CPA adversary $\mathcal{A}$ can make queries to H. Then, in a OW-CPA reduction $\mathcal{B}$ playing with $\mathcal{A}$, $\mathcal{B}$ needs to simulate the oracle H for $\mathcal{A}$. In particular, $\mathcal{B}$ can *observe* $\mathcal{A}$'s queries to H.

(4pts) a) We just unfold Encaps in the IND-CPA games:

$\text{IND-CPA}^0_{\mathsf{KEM}}(\mathcal{A})$	$\text{IND-CPA}^1_{\mathsf{KEM}}(\mathcal{A})$
$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{gen}(1^\lambda)$	$(\mathsf{pk}, \mathsf{sk}) \leftarrow\!\!\$\ \mathsf{gen}(1^\lambda)$
$\mathsf{pt}^* \leftarrow\!\!\$\ \{0,1\}^{\ell_1}; \mathsf{ct}^* \leftarrow \mathsf{enc}(\mathsf{pk}, \mathsf{pt}^*)$	$\mathsf{pt}^* \leftarrow\!\!\$\ \{0,1\}^{\ell_1}; \mathsf{ct}^* \leftarrow \mathsf{enc}(\mathsf{pk}, \mathsf{pt}^*)$
$K^* \leftarrow H(\mathsf{pt}^*)$	$K^* \leftarrow\!\!\$\ \{0,1\}^{\ell_2}$
$b' \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}, \mathsf{ct}^*, K^*)$; **return** b'	$b' \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}, \mathsf{ct}^*, K^*)$; **return** b'

b) The difference between both games is simply that $\mathcal{A}$ receives a random K^* in one case and the real $K^* = H(\mathsf{pt}^*)$ in the other. Thus, using Question 2, we can deduce that

$$\left|\Pr[\text{IND-CPA}^1{}_{\mathsf{KEM}}(\mathcal{A}) \Rightarrow 1] - \Pr[\text{IND-CPA}^0{}_{\mathsf{KEM}}(\mathcal{A}) \Rightarrow 1]\right| \leq \Pr[\mathsf{query}]$$

where query is the event *"$\mathcal{A}$ queries pt^* to H"*. Now, let's consider the following OW-CPA adversary $\mathcal{B}$.

$\mathcal{B}^{\mathcal{A}}_{\mathsf{PKC}}(\mathsf{pk}, \mathsf{ct}^*)$

$K^* \leftarrow\!\!\$\ \{0,1\}^{\ell_2}$
initialize a RO oracle H with a list $\mathcal{L}$ as in Question 1
run $\mathcal{A}^H(\mathsf{pk}, \mathsf{ct}^*, K^*)$ and replies to RO queries with simulated RO H
$\mathsf{pt}' \leftarrow\!\!\$\ \mathcal{L}$

$\mathcal{B}$ perfectly simulates $\mathcal{A}$'s view in the $\text{IND-CPA}^1_{\mathsf{KEM}}$ game. Moreover, if query happens, pt^* will be contained in $\mathcal{B}$'s list $\mathcal{L}$ and $\mathcal{B}$ will output it with probability $\frac{1}{q}$, where q is the number of RO queries made by $\mathcal{A}$. Therefore,

$$\Pr[\text{OW-CPA}_{\mathsf{PKC}}(\mathcal{B})] = \Pr[\mathcal{B}^{\mathcal{A}} \Rightarrow \mathsf{pt}^*] = \frac{1}{q}\Pr[\mathsf{query}]\ .$$

Hence, for any adversary ppt IND-CPA KEM adversary $\mathcal{A}$, we can build a PKC adversary $\mathcal{B}$ s.t.

$$\begin{aligned}\left|\Pr[\text{IND-CPA}^1{}_{\mathsf{KEM}}(\mathcal{A}) \Rightarrow 1] - \Pr[\text{IND-CPA}^0{}_{\mathsf{KEM}}(\mathcal{A}) \Rightarrow 1]\right| &\leq \Pr[\mathsf{query}] \\ &= q\Pr[\text{OW-CPA}_{\mathsf{PKC}}(\mathcal{B})]\end{aligned}$$

which concludes the proof.
Note that $\mathcal{B}$ cannot check $\mathsf{enc}(\mathsf{pk}, \mathsf{pt}') = \mathsf{ct}^*$ for all $\mathsf{pt}' \in \mathcal{L}$ to find the correct pt^*, as the encryption might be randomized.

2 Relations between primitives

In this exercise, you will show some relations between cryptographic primitives.

Question 1. We consider a 2-message Key Agreement (KA) protocol (see slide 84 for the general definition of KA). In a 2-message KA, the protocol runs

1. $st_A, m_A \leftarrow\!\!\$\ A(1^\lambda)$
2. $K_B, m_B \leftarrow\!\!\$\ B(1^\lambda, m_A)$
3. The derived key at A is computed as $K_A \leftarrow A(1^\lambda, st_A, m_B)$.

Show how one can build a KEM from any 2-message KA, prove that the corresponding KEM is correct.

(3pts) We define the KEM algorithms as follows.

- $\mathsf{Gen}(1^\lambda) := A(1^\lambda)$. I.e., we run $st_A, m_A \leftarrow\!\!\$\ A(1^\lambda)$ and we set $\mathsf{sk} \leftarrow st_A$ and $\mathsf{pk} \leftarrow m_A$.
- $\mathsf{Encaps}(\mathsf{pk}) := B(1^\lambda, \mathsf{pk})$. I.e., we run $K_B, m_B \leftarrow\!\!\$\ B(\mathsf{pk}) = B(m_A)$ and we set $K \leftarrow K_B$ and $\mathsf{ct} \leftarrow m_B$.
- $\mathsf{Decaps}(\mathsf{sk}, \mathsf{ct}) := A(\mathsf{sk}, \mathsf{ct})$. I.e., we run $K_A \leftarrow A(\mathsf{sk}, \mathsf{ct}) = A(st_A, m_B)$ and we set $K \leftarrow K_A$.

Clearly, if the KA is correct and runs in polynomial time so is the KEM. In addition, the decapsulation is deterministic.

Question 2. Prove that if the underlying KA is secure against key distinguisher in a passive attack setting (slide 86), the KEM resulting from the construction of Question 1 is IND-CPA.

(3pts) Nothing complicated here. In short, for any IND-CPA KEM adversary $\mathcal{A}(\mathsf{pk}, \mathsf{ct}^*, K^*)$ that can distinguish between the real and random key K^*, one can define the KA-distinguishing adversary $\mathcal{B} := \mathcal{A}$ that has the same advantage as $\mathcal{A}$. Indeed, $(\mathsf{pk}, \mathsf{ct}^*, K^*)$ in the IND-CPA KEM is exactly $(\text{transcript}, K_b)$ in the KA distinguishing game in slide 86.

Question 3. Show how one can build a PKC by combining a KEM **and** a symmetric block cipher. Prove that the resulting primitive is correct.
Hint: Think of the way ElGamal is built.

(3pts) We build a PKC $\mathsf{PKC} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ from a KEM $\mathsf{KEM} = (\mathsf{gen}, \mathsf{encaps}, \mathsf{decaps})$ and a block cipher $\Pi = (\mathsf{Enc}_S, \mathsf{Dec}_S)$ as follows.

$\underline{\text{IND-CPA}^0_\Pi(\mathcal{A})}$	$\underline{\text{IND-CPA}^1_\Pi(\mathcal{A})}$
$K \leftarrow\!\!\$\, \mathcal{K}$	$K \leftarrow\!\!\$\, \mathcal{K}$
$m_0, m_1, st \leftarrow\!\!\$\, \mathcal{A}$	$m_0, m_1, st \leftarrow\!\!\$\, \mathcal{A}$
$\mathsf{ct}^* \leftarrow\!\!\$\, \mathsf{Enc}(m_0)$	$\mathsf{ct}^* \leftarrow\!\!\$\, \mathsf{Enc}(m_1)$
$b' \leftarrow\!\!\$\, \mathcal{A}(st, \mathsf{ct}^*)$	$b' \leftarrow\!\!\$\, \mathcal{A}(st, \mathsf{ct}^*)$
return b'	**return** b'

Figure 3: OT-CPA game for a symmetric cipher.

- $\mathsf{Gen} := \mathsf{gen}$.
- $\mathsf{Enc}(\mathsf{pk}, \mathsf{pt})$: Run $K, \mathsf{ct}_1 \leftarrow\!\!\$\, \mathsf{encaps}(\mathsf{pk})$ and $\mathsf{ct}_2 \leftarrow\!\!\$\, \mathsf{Enc}_S(K, \mathsf{pt})$. Output $\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2)$.
- $\mathsf{Decaps}(\mathsf{sk}, (\mathsf{ct}_1, \mathsf{ct}_2))$: Decrypt $K' \leftarrow \mathsf{decaps}(\mathsf{sk}, \mathsf{ct}_1)$, then decrypt $\mathsf{pt}' \leftarrow \mathsf{Dec}_S(K', \mathsf{ct}_2)$. Output pt'.

Clearly, if KEM and Π are correct and run in polynomial time, it is also the case of PKC. In addition, the decryption function is deterministic as decaps and Dec_S are deterministic as well.

Definition 3. We say a symmetric cipher $\Pi = (\mathsf{Enc}, \mathsf{Dec})$ is one-time secure against passive attacks (OT-CPA) if for all ppt adversary $\mathcal{A}$ the advantage

$$\mathsf{Adv}^{\text{ot}-\text{cpa}}_{\mathcal{A},\Pi} := \left|\Pr[\text{OT-CPA}^1{}_\Pi(\mathcal{A}) \Rightarrow 1] - \Pr[\text{OT-CPA}^0{}_\Pi(\mathcal{A}) \Rightarrow 1]\right|$$

is negligible in λ and the games OT-CPA^b are defined in Figure 3, where $\mathcal{K}$ is the key space.

Question 4. Prove that if the underlying KEM is IND-CPA **and** the underlying block cipher is OT-CPA, the PKC resulting from the construction in Question 3 is IND-CPA (slide 90).

(4pts) We proceed by constructing hybrid games. We start with the PKC IND-CPA^0 game and we modify it s.t. the symmetric key is sampled uniformly at random. We call this game Γ, which is detailed in Fig. 4. Now for any $\mathcal{A}$ we can build $\mathcal{B}$ s.t.

$$\left|\Pr[\text{IND-CPA}^0{}_{\mathsf{PKC}}(\mathcal{A}) \Rightarrow 1] - \Pr[\Gamma_{\mathsf{PKC}}(\mathcal{A}) \Rightarrow 1]\right| \leq \mathsf{Adv}^{\text{ind}-\text{cpa}}_{\mathcal{B},\mathsf{KEM}} .$$

The adversary $\mathcal{B}$ is as follows.

$\underline{\mathcal{B}^{\mathcal{A}}(\mathsf{pk}, \mathsf{ct}^*, K^*)}$
$m_0, m_1, st \leftarrow\!\!\$\, \mathcal{A}(\mathsf{pk})$
$\mathsf{ct}^*_2 \leftarrow\!\!\$\, \mathsf{Enc}_S(K^*, m_0)$
$b' \leftarrow\!\!\$\, \mathcal{A}(st, (\mathsf{ct}^*, \mathsf{ct}^*_2)$
return b'

If the key is real, $\mathcal{B}$ simulates $\mathcal{A}$'s view in the IND-CPA^0 game, and if it is random it perfectly simulates $\mathcal{A}$'s view in the Γ game. This proves the inequality above.

Next, we modify the game Γ into another Γ' (see Fig. 4), where we encrypt the message m_1 instead of the message m_1. Again, for any adversary $\mathcal{A}$, one can build a $\mathcal{C}$ adversary playing the OT-CPA game against Π s.t.

$$\left|\Pr[\Gamma_{\mathsf{PKC}}(\mathcal{A}) \Rightarrow 1] - \Pr[\Gamma'_{\mathsf{PKC}}(\mathcal{A}) \Rightarrow 1]\right| \leq \mathsf{Adv}^{\mathrm{ot-cpa}}_{\mathcal{C},\Pi} \ .$$

The adversary $\mathcal{C}$ is as follows.

$\mathcal{C}^{\mathcal{A}}$	$\mathcal{C}^{\mathcal{A}}(st, \mathsf{ct}^*)$
$\mathsf{sk}, \mathsf{pk} \leftarrow\!\!\$\ \mathsf{gen}$	$K, \mathsf{ct}_1 \leftarrow\!\!\$\ \mathsf{encaps}(\mathsf{pk})$
$m_0, m_1, st \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk})$	$b' \leftarrow\!\!\$\ \mathcal{A}(st, (\mathsf{ct}_1, \mathsf{ct}_2^*))$
return st, m_0, m_1	**return** b'

If $\mathcal{C}$ plays the OT-CPA^0 game, it will simulate $\mathcal{A}$'s view in the Γ game, otherwise it perfectly simulates $\mathcal{A}$'s view in the Γ' game. Note that $\mathcal{C}$ can simulate $\mathcal{A}$ environment because ct_1^* is independent of ct_2^* in games Γ and Γ'. Finally, we can transform Γ' into IND-CPA^1 by replacing the random symmetric key with the one output by the encapsulation (i.e. we revert the modification made in the first step of the proof). As in the first step, we have

$$\left|\Pr[\Gamma'_{\mathsf{PKC}}(\mathcal{A}) \Rightarrow 1] - \Pr[\text{IND-CPA}^1{}_{\mathsf{PKC}}(\mathcal{A}) \Rightarrow 1]\right| \leq \mathsf{Adv}^{\mathrm{ind-cpa}}_{\mathcal{B},\mathsf{KEM}} \ .$$

Hence, collecting the inequalities, we have

$$\begin{aligned}
&\left|\Pr[\text{IND-CPA}^0{}_{\mathsf{PKC}}(\mathcal{A}) \Rightarrow 1] - \Pr[\text{IND-CPA}^1{}_{\mathsf{PKC}}(\mathcal{A}) \Rightarrow 1]\right| \\
&\leq \left|\Pr[\text{IND-CPA}^0{}_{\mathsf{PKC}}(\mathcal{A}) \Rightarrow 1] - \Pr[\Gamma_{\mathsf{PKC}}(\mathcal{A}) \Rightarrow 1]\right| \\
&+ \left|\Pr[\Gamma_{\mathsf{PKC}}(\mathcal{A}) \Rightarrow 1] - \Pr[\Gamma'_{\mathsf{PKC}}(\mathcal{A}) \Rightarrow 1]\right| \\
&+ \left|\Pr[\Gamma'_{\mathsf{PKC}}(\mathcal{A}) \Rightarrow 1] - \Pr[\text{IND-CPA}^1{}_{\mathsf{PKC}}(\mathcal{A}) \Rightarrow 1]\right| \\
&\leq 2\mathsf{Adv}^{\mathrm{ind-cpa}}_{\mathcal{B},\mathsf{KEM}} + \mathsf{Adv}^{\mathrm{ot-cpa}}_{\mathcal{C},\Pi}
\end{aligned}$$

which concludes the proof.

$\Gamma(\mathcal{A})$	$\Gamma'(\mathcal{A})$
$\mathsf{sk}, \mathsf{pk} \leftarrow\!\!\$\ \mathsf{gen}$	$\mathsf{sk}, \mathsf{pk} \leftarrow\!\!\$\ \mathsf{gen}$
$m_0, m_1, st \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk})$	$m_0, m_1, st \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk})$
$K^*, \mathsf{ct}_1^* \leftarrow\!\!\$\ \mathsf{encaps}(\mathsf{pk})$	$K^*, \mathsf{ct}_1^* \leftarrow\!\!\$\ \mathsf{encaps}(\mathsf{pk})$
$K^* \leftarrow\!\!\$\ \{0,1\}^{\ell_2}$	$K^* \leftarrow\!\!\$\ \{0,1\}^{\ell_2}$
$\mathsf{ct}_2^* \leftarrow\!\!\$\ \mathsf{Enc}_S(K^*, m_0)$	$\mathsf{ct}_2^* \leftarrow\!\!\$\ \mathsf{Enc}_S(K^*, m_1)$
$b' \leftarrow\!\!\$\ \mathcal{A}(st, (\mathsf{ct}_1^*, \mathsf{ct}_2^*))$	$b' \leftarrow\!\!\$\ \mathcal{A}(st, (\mathsf{ct}_1^*, \mathsf{ct}_2^*))$
return b'	**return** b'

Figure 4: Intermediate games for the proof in Q.4.