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?

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"*.

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.

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

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

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

The corresponding IND-CPA definition is defined as follows.

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.

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.

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.

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.

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

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

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).

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

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