Advanced Cryptography — Midterm Exam
Solution

Serge Vaudenay

17.4.2025

- duration: 1h45
- any document allowed
- a pocket calculator is allowed
- communication devices are not allowed
- the exam invigilators will **not** answer any technical question during the exam
- readability and style of writing will be part of the grade

The exam grade follows a linear scale in which each question has the same weight.

1 Alternate IND-CCA Security

Given a public-key cryptosystem $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, we define the following games for $b = 0, 1$:

```
$\Gamma_b(\mathcal{A}_1, \mathcal{A}_2)$:
  $\mathsf{Gen} \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$
  $\mathcal{A}_1^{\mathsf{ODec}_1}(\mathsf{pk}) \xrightarrow{\$} (\mathsf{pt}_0, \mathsf{pt}_1, \mathsf{st})$
  **if** $|\mathsf{pt}_0| \neq |\mathsf{pt}_1|$ **then return** $0$
  $\mathsf{ct}_0 \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, \mathsf{pt}_{1-b})$
  $\mathsf{ct}_1 \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, \mathsf{pt}_b)$
  $\mathcal{A}_2^{\mathsf{ODec}_2}(\mathsf{st}, \mathsf{ct}_0, \mathsf{ct}_1) \xrightarrow{\$} z$
  **return** $z$

Oracle $\mathsf{ODec}_1(\mathsf{ct})$:
  **return** $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$

Oracle $\mathsf{ODec}_2(\mathsf{ct})$:
  **if** $\mathsf{ct} = \mathsf{ct}_0$ or $\mathsf{ct} = \mathsf{ct}_1$ **then return** $\perp$
  **return** $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$
```

The advantage is defined as $\mathsf{Adv}(\mathcal{A}_1, \mathcal{A}_2) = \Pr[\Gamma_1 \rightarrow 1] - \Pr[\Gamma_0 \rightarrow 1]$. We say that the cryptosystem is IND^2-CCA-secure if for any PPT $(\mathcal{A}_1, \mathcal{A}_2)$, the advantage is negligible.

Q.1 What is the difference between this notion and IND-CCA security?

In IND-CCA security, only one of the two chosen plaintexts is encrypted and returned to the adversary. Here, the two are encrypted and given to the adversary in an order which depends on b. ODec_2 is now checking both challenge ciphertexts. Hence, the goal of the adversary is to figure out if the two messages have been swapped or not.

Q.2 Prove that the plain ElGamal cryptosystem is not IND^2-CCA-secure by specifying an adversary and proving that it has a high advantage.

We adapt the adversary which has been seen in class for OW-CCA security. We denote by pp the group parameters which include the prime order q and a generator g. We use multiplicative notations for the group.

```
$\mathcal{A}_1^{\mathsf{ODec}_1}(\mathsf{pp}, y)$:
  pick a random message $m_0$ in the group
  $m_1 \leftarrow m_0 \cdot g$
  $\mathsf{state} \leftarrow (\mathsf{pp}, m_0)$
  return $(m_0, m_1, \mathsf{state})$
$\mathcal{A}_2^{\mathsf{ODec}_2}(\mathsf{state}, (u_0, v_0), (u_1, v_1))$:
  $\mathsf{state} \rightarrow (\mathsf{pp}, m_0)$
  $m \leftarrow \mathsf{ODec}_2(u_0, v_0 \cdot g^2) \cdot g^{-2}$
  return $1_{m=m_0}$
```

Clearly, the adversary is PPT.

Normally, we have $u_0 \neq u_1$ so the query $(u_0, v_0 \cdot g^2)$ to ODec_2 can be equal to neither (u_0, v_0) nor (u_1, v_1). If it happens that $u_0 = u_1$, we have $v_0 = v_1 \cdot g^{\pm 1}$. Hence $(u_0, v_0 \cdot g^2)$ can be equal to neither (u_0, v_0) nor (u_1, v_1). Therefore, the query to ODec_2 always answers.

Due to the homomorphic property, the answer is m_{1-b}. Hence, the response from $\mathcal{A}_2$ is $z = b$. We deduce that the advantage is 1, which is the largest possible. Hence, the cryptosystem is not IND^2-CCA-secure.

When grading, some points were removed if the discussion about the case when ODec_2 was returning $\perp$ was missing. In the above solution, m_0 and m_1 were selected so that $m_i \cdot g^2$ is equal to no m_j so there is no chance that the queries ciphertext matches one of the challenged ones.

Q.3 We want to prove that IND^2-CCA security implies IND-CCA.

Q.3a We first consider the variant IND^2-CPA security of IND^2-CCA security where there is no decryption oracle. If the cryptosystem is IND^2-CPA-secure, prove that it is IND-CPA-secure.

We assume that the cryptosystem is IND^2-CPA-secure. To prove IND-CPA security, we take a PPT adversary $(\mathcal{B}_1, \mathcal{B}_2)$ playing the IND-CPA security game. Given this, we define an adversary $(\mathcal{A}_1, \mathcal{A}_2)$ playing Γ_b as follows. We set $\mathcal{A}_1 = \mathcal{B}_1$ and $\mathcal{A}_2(\mathsf{st}, \mathsf{ct}_0, \mathsf{ct}_1) = \mathcal{B}_2(\mathsf{st}, \mathsf{ct}_1)$. By inspection of the Γ_b and the IND-CPA game with same bit b, we can see that the two games are executing exactly the same operations. Hence, the advantage is the same. Due to IND^2-CPA-security, it is negligible. As this holds for any PPT adversary, we obtain that the cryptosystem is IND-CPA-secure.

Q.3b To extend the previous result to IND^2-CCA security implies IND-CCA security, show that we need to consider a failure case which reduces to the following game returning 1:

$\mathsf{Guess}(\mathcal{C})$:

```
$\mathsf{Gen} \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$
$\mathcal{C}^{\mathsf{ODec}}(\mathsf{pk}) \xrightarrow{\$} (\mathsf{pt}, \mathcal{L})$        ▷ $\mathcal{L}$ is a list of ciphertexts
$\mathsf{ct}_0 \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, \mathsf{pt})$
return $1_{\mathsf{ct}_0 \in \mathcal{L}}$
Oracle $\mathsf{ODec}(\mathsf{ct})$:
  return $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$
```

For this, construct a PPT adverdary $\mathcal{C}$ such that

$$\mathsf{Adv}(\mathcal{B}_1, \mathcal{B}_2) \leq \mathsf{Adv}(\mathcal{A}_1, \mathcal{A}_2) + \mathsf{Adv}(\mathcal{C})$$

Note that in the statement of the question, we forgot to say that $(\mathcal{B}_1, \mathcal{B}_2)$ is the starting IND adversary and that $(\mathcal{A}_1, \mathcal{A}_2)$ is the IND^2 adversary constructed in the previous question. This can have made the question harder than expected. In the previous question, what changes with the decryption oracles is that the ODec_2 oracle has the extra restriction that the query shall not be ct_0, which is an encryption of pt_{1-b} selected by the IND^2-CCA game. This is a failure event in the reduction. The probability that this occurs is the probability that $\mathsf{Guess}(\mathcal{C})$ returns 1 with the following $\mathcal{C}$:

```
$\mathcal{C}^{\mathsf{ODec}}(\mathsf{pk})$:
  set $\mathcal{L}$ to the empty list
  $\mathcal{B}_1^{\mathsf{ODec}}(\mathsf{pk}) \xrightarrow{\$} (\mathsf{pt}_0, \mathsf{pt}_1, \mathsf{st})$
  if $|\mathsf{pt}_0| \neq |\mathsf{pt}_1|$ then make the game return 0
  $\mathsf{ct}_1 \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, \mathsf{pt}_b)$
  $\mathcal{B}_2^{\mathsf{SDec}}(\mathsf{st}, \mathsf{ct}_1) \xrightarrow{\$} z$
  return $(\mathsf{pt}_{1-b}, \mathcal{L})$
Subroutine $\mathsf{SDec}(\mathsf{ct})$:
  add $\mathsf{ct}$ in $\mathcal{L}$
  if $\mathsf{ct} = \mathsf{ct}_1$ then make $\mathcal{C}$ return $(\mathsf{pt}_{1-b}, \mathcal{L})$
  return $\mathsf{ODec}(\mathsf{ct})$
```

The execution or $\mathsf{Guess}(\mathcal{C})$ simulates perfectly the IND-CCA and IND^2-CCA games with bit b, with two differences: the ciphertext ct_0 is computed in the end instead of between $\mathcal{A}_1$ and $\mathcal{A}_2$; the return is 1 in the failure event that ODec_2 is called with input ct_0. Due to the difference Lemma, we obtain

$$\mathsf{Adv}(\mathcal{B}_1, \mathcal{B}_2) \leq \mathsf{Adv}(\mathcal{A}_1, \mathcal{A}_2) + \mathsf{Adv}(\mathcal{C})$$

Q.3c Construct an IND^2-CCA adversary $(\mathcal{D}_1, \mathcal{D}_2)$ making no ODec_2 oracle access and such that

$$\mathsf{Adv}(\mathcal{C}) = \mathsf{Adv}(\mathcal{D}_1, \mathcal{D}_2)$$

Then, conclude about the IND-CCA security of the cryptosystem.

```
$\mathcal{D}_1^{\mathsf{ODec}_1}(\mathsf{pk})$:
  $\mathcal{C}^{\mathsf{ODec}_1}(\mathsf{pk}) \xrightarrow{\$} (\mathsf{pt}_0, \mathcal{L})$
  for each $\mathsf{ct} \in \mathcal{L}$ do
      $\mathsf{pt} \leftarrow \mathsf{ODec}_1(\mathsf{ct})$
      if $\mathsf{pt} \neq \mathsf{pt}_0$ then remove $\mathsf{ct}$ from $\mathcal{L}$
  end for
  pick $\mathsf{pt}_1$ at random, different from $\mathsf{pt}_0$, of same length
  $\mathsf{st} \leftarrow \mathcal{L}$
  return $(\mathsf{pt}_0, \mathsf{pt}_1, \mathsf{st})$
$\mathcal{D}_2^{\mathsf{ODec}_2}(\mathsf{st}, \mathsf{ct}_0, \mathsf{ct}_1)$:
  $\mathsf{st} \rightarrow \mathcal{L}$
  return $1_{\mathsf{ct}_0 \in \mathcal{L}}$
```

With bit $b = 1$, the game Γ perfectly simulates the game $\mathsf{Guess}(\mathcal{C})$. For that, we just observe that removing from the list all ciphertexts which do not decrypt to pt_0 do not change the outcome of the game, due to the correctness of the cryptosystem.

With $b = 0$, we observe that $\mathcal{D}_2$ always returns 0. Hence, $\mathsf{Adv}(\mathcal{C}) = \mathsf{Adv}(\mathcal{D}_1, \mathcal{D}_2)$.

We conclude that

$$\mathsf{Adv}(\mathcal{B}_1, \mathcal{B}_2) \leq \mathsf{Adv}(\mathcal{A}_1, \mathcal{A}_2) + \mathsf{Adv}(\mathcal{D}_1, \mathcal{D}_2)$$

Due to IND^2-CCA security, both terms are negligible. Hence, $\mathsf{Adv}(\mathcal{B}_1, \mathcal{B}_2)$ is negligible. This holds for any PPT $(\mathcal{B}_1, \mathcal{B}_2)$. Therefore, the cryptosystem is IND-CCA secure.

Strictly speaking, we should treat the case that pt_0 returned by $\mathcal{C}$ is of length 0, because we cannot pick any different plaintext pt_1 of same length. If we go back to our reductions in the previous questions, we can see that $\mathcal{C}$ returning such plaintext would imply that $\mathcal{B}_1$ selects that plaintext. But then, it is clear that the advantage of $(\mathcal{B}_1, \mathcal{B}_2)$ is zero: either the two challenge plaintexts have different length and the game answers 0 no matter b, or the two challenge plaintexts are equal and the advantage is zero. So, this case is eliminated in the reduction.

Q.4 We now want to prove that IND-CCA security implies IND^2-CCA. For that, we consider the following intermediary game:

```
$\Gamma_{b,b'}(\mathcal{A}_1, \mathcal{A}_2)$:
  $\mathsf{Gen} \xrightarrow{\$} (\mathsf{pk}, \mathsf{sk})$
  $\mathcal{A}_1^{\mathsf{ODec}_1}(\mathsf{pk}) \xrightarrow{\$} (\mathsf{pt}_0, \mathsf{pt}_1, \mathsf{st})$
  if $|\mathsf{pt}_0| \neq |\mathsf{pt}_1|$ then return $0$
  $\mathsf{ct}_0 \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, \mathsf{pt}_{1-b'})$
  $\mathsf{ct}_1 \xleftarrow{\$} \mathsf{Enc}(\mathsf{pk}, \mathsf{pt}_b)$
  $\mathcal{A}_2^{\mathsf{ODec}_2}(\mathsf{st}, \mathsf{ct}_0, \mathsf{ct}_1) \xrightarrow{\$} z$
  return $z$
```

```
Oracle $\mathsf{ODec}_1(\mathsf{ct})$:
    return $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$
```

```
Oracle $\mathsf{ODec}_2(\mathsf{ct})$:
    if $\mathsf{ct} = \mathsf{ct}_0$ or $\mathsf{ct} = \mathsf{ct}_1$ then return $\perp$
    return $\mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$
```

Q.4a Construct an IND-CCA adversary $(\mathcal{B}_1, \mathcal{B}_2)$ such that

$$\Pr[\mathit{\Gamma}_{1,1}(\mathcal{A}_1, \mathcal{A}_2) \to 1] - \Pr[\mathit{\Gamma}_{1,0}(\mathcal{A}_1, \mathcal{A}_2) \to 1] \leq \mathsf{Adv}(\mathcal{B}_1, \mathcal{B}_2)$$

```
$\mathcal{B}_1^{\mathsf{ODec}_1}(\mathsf{pk})$:
    $\mathcal{A}_1^{\mathsf{ODec}_1}(\mathsf{pk}) \to (\mathsf{pt}_0, \mathsf{pt}_1, \mathsf{st})$
    $\mathsf{st}' \leftarrow (\mathsf{st}, \mathsf{pk}, \mathsf{pt}_1)$
    return $(\mathsf{pt}_1, \mathsf{pt}_0, \mathsf{st}')$
$\mathcal{B}_2^{\mathsf{ODec}_1}(\mathsf{st}', \mathsf{ct}^*)$:
    $\mathsf{st}' \to (\mathsf{st}, \mathsf{pk}, \mathsf{pt}_1)$
    $\mathsf{ct}_0 \leftarrow \mathsf{ct}^*$
    $\mathsf{ct}_1 \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{pt}_1)$
    $\mathcal{A}_2^{\mathsf{ODec}_2}(\mathsf{st}, \mathsf{ct}_0, \mathsf{ct}_1) \to z$
    return $z$
```

The simulation is perfect. We just moved steps from the game to the adversary.

Q.4b Construct an IND-CCA adversary $(\mathcal{C}_1, \mathcal{C}_2)$ such that

$$\Pr[\mathit{\Gamma}_{1,0}(\mathcal{A}_1, \mathcal{A}_2) \to 1] - \Pr[\mathit{\Gamma}_{0,0}(\mathcal{A}_1, \mathcal{A}_2) \to 1] \leq \mathsf{Adv}(\mathcal{C}_1, \mathcal{C}_2)$$

Then, conclude about the IND^2-CCA security of the cryptosystem.

```
$\mathcal{C}_1^{\mathsf{ODec}_1}(\mathsf{pk})$:
  $\mathcal{A}_1^{\mathsf{ODec}_1}(\mathsf{pk}) \to (\mathsf{pt}_0, \mathsf{pt}_1, \mathsf{st})$
  $\mathsf{ct}_0 \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{pt}_1)$
  $\mathsf{st}' \leftarrow (\mathsf{st}, \mathsf{ct}_0)$
  return $(\mathsf{pt}_0, \mathsf{pt}_1, \mathsf{st}')$
$\mathcal{C}_2^{\mathsf{ODec}_1}(\mathsf{st}', \mathsf{ct}^*)$:
  $\mathsf{st}' \to (\mathsf{st}, \mathsf{ct}_0)$
  $\mathsf{ct}_1 \leftarrow \mathsf{ct}^*$
  $\mathcal{A}_2^{\mathsf{ODec}_2}(\mathsf{st}, \mathsf{ct}_0, \mathsf{ct}_1) \to z$
  return $z$
```

The simulation is perfect. We just moved steps from the game to the adversary. We deduce

$$\begin{aligned}
\mathsf{Adv}(\mathcal{A}_1, \mathcal{A}_2) &= \Pr[\Gamma_{1,1}(\mathcal{A}_1, \mathcal{A}_2) \to 1] - \Pr[\Gamma_{0,0}(\mathcal{A}_1, \mathcal{A}_2) \to 1] \\
&= (\Pr[\Gamma_{1,1}(\mathcal{A}_1, \mathcal{A}_2) \to 1] - \Pr[\Gamma_{1,0}(\mathcal{A}_1, \mathcal{A}_2) \to 1]) + \\
&\quad\ (\Pr[\Gamma_{1,0}(\mathcal{A}_1, \mathcal{A}_2) \to 1] - \Pr[\Gamma_{0,0}(\mathcal{A}_1, \mathcal{A}_2) \to 1]) \\
&\le \mathsf{Adv}(\mathcal{B}_1, \mathcal{B}_2) + \mathsf{Adv}(\mathcal{C}_1, \mathcal{C}_2)
\end{aligned}$$

Due to IND-CCA security, both terms are negligible. Hence, $\mathsf{Adv}(\mathcal{A}_1, \mathcal{A}_2)$ *is negligible. This holds for any PPT* $(\mathcal{A}_1, \mathcal{A}_2)$*. Therefore, the cryptosystem is IND*^{2}*-CCA secure.*

2 Proofs for ElGamal

In this exercise, we consider the plain ElGamal cryptosystem from the course. We use a multiplicatively denoted group of prime order q with generator g.

Q.1 After a sender computes $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{pt}; r)$ with a random $r \in \mathbf{Z}_q$, we define an "instance" $x = (\mathsf{pk}, \mathsf{pt}, \mathsf{ct})$ and a "witness" $w = r$. They are connected by a relation $R(x, w)$ to express that ct is the correct encryption of pt. Specify R, propose a Σ-protocol for this relation R, and prove that it is a Σ-protocol.

During encryption, ct *is computed by* $\mathsf{ct}_1 = g^r$ *and* $\mathsf{ct}_2 = \mathsf{pt} \cdot \mathsf{pk}^r$*. So, we use a DLEQ proof for* $\mathsf{ct}_1 = g^r$ *and* $\mathsf{ct}_2/\mathsf{pt} = \mathsf{pk}^r$*.*

$$R((\mathsf{pk}, \mathsf{pt}, \mathsf{ct}), r) \Longleftrightarrow \mathsf{ct}_1 = g^r \ \wedge \ \mathsf{ct}_2/\mathsf{pt} = \mathsf{pk}^r$$

The prover picks ρ *and sends* $a = g^\rho$ *and* $b = \mathsf{pk}^\rho$*. The verifier selects a random challenge* $c \in \mathbf{Z}_q$*. The prover responds by* $z = \rho + c \cdot r \bmod q$*. The verifier checks that* $g^z = a \cdot \mathsf{ct}_1^c$ *and* $\mathsf{pk}^z = b \cdot (\mathsf{ct}_2/\mathsf{pt})^c$*.*
We can see that it follows the structure of a Σ*-protocol and that it is correct.*
Given two responds z_1, z_2 *for two different challenges* c_1, c_2 *with the same* a, b*, the extractor computes* $r' = \frac{z_2 - z_1}{c_2 - c_1} \bmod q$*. Since* $g^{z_2 - z_1} = \mathsf{ct}_1^{c_2 - c_1}$ *and* $\mathsf{pk}^{z_2 - z_1} = (\mathsf{ct}_2/\mathsf{pt})^{c_2 - c_1}$*, we deduce that* $R(x, r')$ *holds.*
For the simulator, we verify the usual structure so that from c*, by picking* z *at random, we can set* $a = g^z \cdot \mathsf{ct}_1^{-c}$ *and* $b = \mathsf{pk}^z \cdot (\mathsf{ct}_2/\mathsf{pt})^{-c}$*.*

Q.2 After a receiver computes $\mathsf{pt} \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$, we define an "instance" $x = (\mathsf{pk}, \mathsf{pt}, \mathsf{ct})$ and a "witness" $w' = \mathsf{sk}$. They are connected by a relation $R'(x, w)$ to express that pt is the correct decryption of ct. Specify R', propose a Σ-protocol for this relation R', and prove that it is a Σ-protocol.

We have $\mathsf{pk} = g^{\mathsf{sk}}$ and $\mathsf{ct}_2/\mathsf{pt} = \mathsf{ct}_1^{\mathsf{sk}}$. So, we use a DLEQ proof again.

$$R'((\mathsf{pk}, \mathsf{pt}, \mathsf{ct}), \mathsf{sk}) \iff \mathsf{pk} = g^{\mathsf{sk}} \ \wedge \ \mathsf{ct}_2/\mathsf{pt} = \mathsf{ct}_1^{\mathsf{sk}}$$

The prover picks ρ and sends $a = g^\rho$ and $b = \mathsf{ct}_1^\rho$. The verifier selects a random challenge $c \in \mathbf{Z}_q$. The prover responds by $z = \rho + c \cdot \mathsf{sk} \bmod q$. The verifier checks that $g^z = a \cdot \mathsf{pk}^c$ and $\mathsf{ct}_1^z = b \cdot (\mathsf{ct}_2/\mathsf{pt})^c$.

We can see that it follows the structure of a Σ-protocol and that it is correct. Given two responds z_1, z_2 for two different challenges c_1, c_2 with the same a, b, the extractor computes $r'' = \frac{z_2 - z_1}{c_2 - c_1} \bmod q$. Since $g^{z_2 - z_1} = \mathsf{pk}^{c_2 - c_1}$ and $\mathsf{ct}_1^{z_2 - z_1} = (\mathsf{ct}_2/\mathsf{pt})^{c_2 - c_1}$, we deduce that $R'(x, r'')$ holds.

For the simulator, we verify the usual structure so that from c, by picking z at random, we can set $a = g^z \cdot \mathsf{ct}_1^{-c}$ and $b = \mathsf{pk}^z \cdot (\mathsf{ct}_2/\mathsf{pt})^{-c}$.

Several students incorrectly answered with the relation $R'((\mathsf{pk}, \mathsf{pt}, \mathsf{ct}), \mathsf{sk}) \iff \mathsf{ct}_2/\mathsf{pt} = \mathsf{ct}_1^{\mathsf{sk}}$ and ended up with a Schnorr proof. Although the protocol is coherent with this R', it does not match the fact that pt is a correct decryption of ct for the public key pk. It rather show the knowledge of one key which makes the decryption correct but which not necessarily corresponding to pk.

Q.3 In the last proof of knowledge of w' such that $R'((\mathsf{pk}, \mathsf{pt}, \mathsf{ct}), w')$, we want to adapt it into a batch proof for $R'((\mathsf{pk}, \mathsf{pt}_i, \mathsf{ct}_i), w')$ for $i = 1, \ldots, n$. For that, we pick a random seed and use a pseudorandom generator set up with seed in order to generate some $\alpha_1, \ldots, \alpha_n \in \mathbf{Z}_q$. We define $\mathsf{pt} = \prod_{i=1}^n \mathsf{pt}_i^{\alpha_i}$ and $\mathsf{ct} = \prod_{i=1}^n \mathsf{ct}_i^{\alpha_i}$. (The product of ct_i pairs is done component-wise.) We modify the previous Σ-protocol by sending seed in the first message and proving $R'((\mathsf{pk}, \mathsf{pt}, \mathsf{ct}), w')$ only. Prove that it is a Σ-protocol for the relation $\forall i = 1, \ldots, n \quad R'((\mathsf{pk}, \mathsf{pt}_i, \mathsf{ct}_i), w')$.

Note: it is recommended not to loose too much time on soundness as it is quite tricky.

This still follows the structure of the Σ-protocol with a first message (a, b, seed), a challenge c, a response z, and a final verification. Correctness is straightforward.

The simulator works similarly: we pick z and seed at random then deduce a and b.

For soundness, the previous extractor extracts r'' such that $\mathsf{pk} = g^{r''}$ and $\mathsf{ct}_2/\mathsf{pt} = \mathsf{ct}_1^{r''}$. This implies $r'' = \mathsf{sk}$. If there exists i such that $\mathsf{ct}_{i2}/\mathsf{pt}_i \neq \mathsf{ct}_{i1}^{\mathsf{sk}}$ then, assuming that the α_i are uniform and independent, the probability that $\mathsf{ct}_2/\mathsf{pt} = \mathsf{ct}_1^{\mathsf{sk}}$ is $\frac{1}{q}$, which is negligible. Hence, except with negligible probability, we have $\mathsf{ct}_{i2}/\mathsf{pt}_i = \mathsf{ct}_{i1}^{r''}$ for every i.

There is a problem in the soundness reasoning: we need to assume the good distribution of the α_i. However, a malicious prover can choose seed as they want and the distribution of α_i is not taken for granted. We can however get around by assuming that the pseudorandom generator is based on a random oracle. (We only expected handwaving arguments for the soundness.)

Some students proposed and AND composition here. It gives a protocol which is much less efficient than the proposed one.

Q.4 Can we do the same with the protocol of the first question?

No. This is because the exponent r_i is not the same for all instances, unlike sk which is the same for all.