Advanced Cryptography — Final Exam
Solution

Serge Vaudenay

25.6.2025

- duration: 3h
- any document allowed
- a pocket calculator is allowed
- communication devices are not allowed
- it is not allowed to write with a pencil
- it is not allowed to use the red color
- 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 Damgård's ElGamal Encryption

This exercise is inspired from Libert: Leveraging Small Message Spaces for CCA1 Security in Additively Homomorphic and BGN-Type Encryption, *EUROCRYPT 2025, LNCS vol. 15602, Springer.*

We define the following variant of the ElGamal cryptosystem. We assume a constant c. The key generation is an algorithm $\mathsf{Gen}(1^s) \rightarrow (\mathsf{pp}, G, G_1, G_2, H, x_1, x_2)$. It sets up some public parameters pp which include the security parameter s, some group parameters (allowing to make additive group operations), and the order q of the group (which is a prime number). It also generates uniformly three generators G, G_1, G_2 of the group, two scalars x_1 and x_2 in $\mathbf{Z}_q$, and $H = x_1G_1 + x_2G_2$. The public key is $\mathsf{pk} = (\mathsf{pp}, G, G_1, G_2, H)$ and the secret key is $\mathsf{sk} = (\mathsf{pp}, G, x_1, x_2)$. Algorithms are polynomially bounded in terms of s. The message space is $\{0, 1, \ldots, s^c - 1\}$. Encryption consists of picking a random $r \in \mathbf{Z}_q^*$ and setting

$$\mathsf{Enc}(\mathsf{pk}, m; r) = (mG + rH, rG_1, rG_2)$$

The decryption algorithm $\mathsf{Dec}(\mathsf{sk}, C_0, C_1, C_2)$ returns either m or an error $\perp$.

Q.1 Explain how decryption works and what is its complexity.

If $C_1 = 0$ or $C_2 = 0$, the algorithm returns $\perp$ as this is not a valid ciphertext. Otherwise, the algorithm computes $M = C_0 - x_1C_1 - x_2C_2$ which is equal to mG if (C_0, C_1, C_2) is the result of the encryption of m. Using the double-and-add algorithm, this computation of M runs in $\mathcal{O}(\log q)$ group additions.
Then, we compute the discrete logarithm of M in the message space. Since the message space is small (of cardinality s^c, which is polynomially bounded), this can be done by exhaustive search or with a dictionary in complexity $\mathcal{O}(s^c)$ group additions. We can also improve this with the baby-step giant-step algorithm in $\mathcal{O}(s^{\frac{c}{2}})$.
In total, the complexity is $\mathcal{O}(\log q + s^{\frac{c}{2}})$ group additions.
A very frequent error is to write $m = (C_0 - x_1C_1 - x_2C_2)G^{-1} \bmod q$. Division by a group element does not make sense. The modulo q is not appropriate either.

Q.2 We consider the INDCCA1 security which is defined by the following game.

Game $\Gamma_{\mathsf{INDCCA1}}(1^s, b)$
$\mathsf{Gen}(1^s) \to (\mathsf{pk}, \mathsf{sk})$
$\mathcal{A}^{\mathsf{ODec}}(\mathsf{pk}) \to (m_0, m_1, \mathsf{st})$
$\mathsf{Enc}(\mathsf{pk}, m_b; r) \to (C_0, C_1, C_2)$
$\mathcal{A}(\mathsf{st}, C_0, C_1, C_2) \to z$
return z

Oracle $\mathsf{ODec}(D_0, D_1, D_2)$
return $\mathsf{Dec}(\mathsf{sk}, D_0, D_1, D_2)$

The oracle ODec(input) which returns the result of Dec(sk, input).

Define the advantage of $\mathcal{A}$.

What is the difference with the normal INDCCA game?

Prove that the cryptosystem is not INDCCA secure.

The advantage is $\Pr[\Gamma_{\mathsf{INDCCA1}}(1^s, 1) \to 1] - \Pr[\Gamma_{\mathsf{INDCCA1}}(1^s, 0) \to 1]$.
The difference with INDCCA *security is that the second step of the adversary* $\mathcal{A}$ *is not given access to a decryption oracle. In the* INDCCA *notion, such access is given, with the restriction that the input must not be equal to* (C_0, C_1, C_2). *We cannot consider this notion because of the homomorphic property of the ElGamal encryption. Indeed, an adversary could query the decryption oracle in the second stage with* (C_0+G, C_1, C_2), *obtain* m_b+1, *deduce the value of* b, *and return* $z = b$. *The advantage would be one if* m_0 *and* m_1 *are chosen different from each other in the message space. (But* $\perp$ *could be returned instead of* s^c *if* $m_b = s^c - 1$.*) Hence, the following adversary has an advantage of 1 in the regular* INDCCA *game.*

$\mathcal{A}(\mathsf{pk})$:

1: *pick* m_0, m_1 *arbitrarily, but different in* $\{0, \ldots, s^c - 2\}$
2: $\mathsf{st} \leftarrow (\mathsf{pk}, m_1)$
3: ***return*** (m_0, m_1, st)

$\mathcal{A}(\mathsf{st}, C_0, C_1, C_2)$:

4: $\mathsf{st} \to (\mathsf{pk}, m_1)$
5: *get* pp *and* G *from* pk
6: $\mathsf{ODec}(C_0 + G, C_1, C_2) \to m$
7: ***return*** $1_{m=m_1+1}$

Q.3 Prove that the advantage in the INDCCA1 game is equal to the advantage of the following game.

Game $\Gamma_1(1^s, b)$

1: $\mathsf{Gen}(1^s) \to (\mathsf{pp}, G, G_1, G_2, H, x_1, x_2)$
2: $\mathcal{A}^{\mathsf{ODec}}(\mathsf{pp}, G, G_1, G_2, H) \to (m_0, m_1, \mathsf{st})$
3: pick $r \in \mathbf{Z}_q^*$, set $C_1 = rG_1$, $C_2 = rG_2$, $C_0 = m_bG + x_1C_1 + x_2C_2$
4: $\mathcal{A}(\mathsf{st}, C_0, C_1, C_2) \to z$
5: **return** z

The difference between $\Gamma_{\mathsf{INDCCA1}}(1^s, b)$ *and* $\Gamma_1(1^s, b)$ *happens during the encryption of* m_b, *with the computation of* C_0. *We can see that* $m_bG + x_1C_1 + x_2C_2 = m_bG + rH$, *so the two games obtain the same results, hence, produce the same output* z *with the same distribution. We deduce that they give the same advantage.*

Q.4 We consider the following game.

Game $\Gamma_2(1^s, b)$

1: $\mathsf{Gen}(1^s) \to (\mathsf{pp}, G, G_1, G_2, H, x_1, x_2)$
2: $\mathcal{A}^{\mathsf{ODec}}(\mathsf{pp}, G, G_1, G_2, H) \to (m_0, m_1, \mathsf{st})$
3: pick $r, r' \in \mathbf{Z}_q^*$, set $C_1 = rG_1$, $C_2 = r'G_2$
4: $C_0 = m_bG + x_1C_1 + x_2C_2$
5: $\mathcal{A}(\mathsf{st}, C_0, C_1, C_2) \to z$
6: **return** z

Formulate a standard security assumption under which the difference of the advantages of Γ_1 and Γ_2 is negligible.

> *The difference between $\Gamma_1(1^s, b)$ and $\Gamma_2(1^s, b)$ is in the selection of r and the computation of C_1 and C_2. The previous game computes $(C_1, C_2) = (rG_1, rG_2)$ with a random r which is not used after that, while the current game computes $(C_1, C_2) = (rG_1, r'G_2)$ with random r, r' which are not used after that. We define the following adversary.*
>
> $\mathcal{B}(\mathsf{pp}, G_1, G_2, C_1, C_2)$:
>
> *pick* $x_1, x_2 \in \mathbf{Z}_q$ *and a generator* G*, compute* $H = x_1G_1 + x_2G_2$
> *simulate* $\mathcal{A}^{\mathsf{ODec}}(\mathsf{pp}, G, G_1, G_2, H) \to (m_0, m_1, \mathsf{st})$
> $C_0 = m_bG + x_1C_1 + x_2C_2$
> *simulate* $\mathcal{A}(\mathsf{st}, C_0, C_1, C_2) \to z$
> ***return*** z
>
> *This adversary is a distinguisher between (G_1, G_2, rG_1, rG_2) and $(G_1, G_2, rG_1, r'G_2)$ which behaves exactly like a simulator of $\Gamma_1(1^s, b)$ or $\Gamma_2(1^s, b)$. Essentially, it is playing the decisional Diffie-Hellman game (DDH). Hence, under the hardness DDH assumption, the difference of advantages is negligible.*
> *Some students wanted to use the difference lemma with the failing event $F : r \neq r'$. The problem is that $\Pr[F]$ is not negligible.*

Q.5 Prove that $\Gamma_2(1^s, b)$ and the following game $\Gamma_3(1^s, b)$ give the same advantages.

Game $\Gamma_3(1^s, b)$

generate (pp, G, G_1) like with Gen
pick $\omega \in \mathbf{Z}_q^*$
pick $z, x_2 \in \mathbf{Z}_q$
set $x_1 = z - x_2\omega$
set $G_2 = \omega G_1$ and $H = zG_1$
$\mathcal{A}^{\mathsf{ODec}}(\mathsf{pp}, G, G_1, G_2, H) \to (m_0, m_1, \mathsf{st})$
pick $r, r' \in \mathbf{Z}_q^*$, set $C_1 = rG_1$, $C_2 = r'G_2$
$C_0 = m_bG + rH + x_2(r' - r)G_2$
$\mathcal{A}(\mathsf{st}, C_0, C_1, C_2) \to z$
return z

> *What is new in Γ_3 is that now G_2 is selected with a known discrete logarithm ω, then $H = (x_1 + x_2\omega)G_1$ is computed by picking $z = x_1 + x_2\omega$ instead of x_1. This produce (G_2, H, x_1) of same distribution. Also, the computation of C_0 is changed again but gives the same result. This does not change the result so give the same advantage.*

Q.6 Given an index i, we define the game Γ_i' as follows.

Game $\Gamma_i'(1^s, b)$

generate (pp, G, G_1) like with Gen
pick $\omega \in \mathbf{Z}_q^*$
pick $z, x_2 \in \mathbf{Z}_q$
set $x_1 = z - x_2\omega$
set $G_2 = \omega G_1$ and $H = zG_1$
set $\mathsf{ct} = 0$
$\mathcal{A}^{\mathcal{O}}(\mathsf{pp}, G, G_1, G_2, H) \to (m_0, m_1, \mathsf{st})$
pick $r, r' \in \mathbf{Z}_q^*$, set $C_1 = rG_1$, $C_2 = r'G_2$
$C_0 = m_b G + rH + x_2(r' - r)G_2$
$\mathcal{A}(\mathsf{st}, C_0, C_1, C_2) \to z$
return z

Oracle $\mathcal{O}(D_0, D_1, D_2)$

increment ct
if $\mathsf{ct} > i$ **then**
    return $\mathsf{Dec}(\mathsf{pp}, G, x_1, x_2, D_0, D_1, D_2)$
else
    if $D_2 \neq \omega D_1$ **then return** $\perp$
    if $D_1 = 0$ **then return** $\perp$
    set $M = D_0 - zD_1$
    find the discrete logarithm m of M in the message space (set $m = \perp$ if none)
    return m
end if

Prove that $\Gamma_0'(1^s, b)$ gives the same advantage as $\Gamma_3(1^s, b)$. Further define an event Bad_i which can occur during the execution of the games and such that $\Pr[\Gamma_{i-1}' \to 1 | \neg \mathsf{Bad}_i] = \Pr[\Gamma_i' \to 1 | \neg \mathsf{Bad}_i]$.

What is new in $\Gamma'_i(1^s, b)$ is the definition of the oracle. However, for $i = 0$, the oracle will do exactly the same computation. Hence, $\Gamma_3(1^s, b)$ is just the same as $\Gamma'_0(1^s, b)$.
The difference between Γ'_{i-1} and Γ'_i is in how the ith query to the oracle is computed.
*We note that whenever $D_2 = \omega D_1$, the two branches of the **if** statement in the oracle give exactly the same result. Furthermore, when $D_2 \neq \omega D_1$ and either $D_0 - x_1 D_1 - x_2 D_2 \notin \{mG; m = 0, \ldots, s^c - 1\}$ or $D_2 = 0$, the two branches give $\perp$ as a response.*
We let (D_0^i, D_1^i, D_2^i) denote the ith query to the oracle. We define

$$\mathsf{Bad}_i : \quad D_2^i \neq \omega D_1^i \;\wedge\; D_2^i \neq 0 \;\wedge\; D_0^i - x_1 D_1^i - x_2 D_2^i \in \{mG; m = 0, \ldots, s^c - 1\}$$

When this event occurs, the ith query to the oracle may give different results in Γ'_{i-1} and Γ'_i: it results to $\perp$ in Γ'_i but not in Γ'_{i-1}. As we have seen, when it does not occur, the two games give the same results.
When students defined an event which works for this question but has not a negligible probability to occur, they have got the points for this question but not for the next one.

Q.7 Prove that $\Pr[\mathsf{Bad}_i] \leq \frac{s^c}{q}$. Deduce that the difference between the advantages given by Γ'_{i-1} and Γ'_i is negligible.

Hint: when is the first time x_2 is used?

The value of x_2 is used for the first time when we want to check if the event Bad_i happens, at the ith query. Hence, the query itself is independent from the uniform selection of x_2. We made sure that x_2 is multiplied by some nonzero D_2^i. We deduce that $\Pr[\mathsf{Bad}_i] = \frac{s^c}{q}$.
We use the difference lemma to deduce that the difference between the advantages given by Γ'_{i-1} and Γ'_i is bounded by this negligible probability. Furthermore, s^c/q is negligible.

Q.8 Prove that there exists some polynomial $Q(s)$ such that the game $\Gamma'_{Q(s)}$ gives the advantage bounded by $\frac{1}{q}$.

We let Q be a polynomial which upper bounds the complexity of $\mathcal{A}$. The number of queries is bounded by Q. Hence, Γ'_Q never uses x_1, and uses x_2 only to compute C_0.

Except when the bad event $r = r'$ occurs (which occurs with probability $\frac{1}{q}$), we obtain that C_0 is the sum of something which depends on b ($m_b G + rH$) and something uniform and independent from everything else in the game ($x_2(r' - r)G_2$). Using the difference lemma again, we can switch to the following game, with a difference in the advantage which is negligible.

Game $\Gamma_4(1^s, b)$

generate (pp, G, G_1) *like with* Gen
pick $\omega \in \mathbf{Z}_q^*$
pick $z \in \mathbf{Z}_q$
set $G_2 = \omega G_1$ *and* $H = zG_1$
$\mathcal{A}^{\mathcal{O}}(\mathsf{pp}, G, G_1, G_2, H) \to (m_0, m_1, \mathsf{st})$
pick $r, r' \in \mathbf{Z}_q$
set $C_1 = rG_1$, $C_2 = r'G_2$
pick C_0 *uniform*
$\mathcal{A}(\mathsf{st}, C_0, C_1, C_2) \to z$
return z

Oracle $\mathcal{O}(D_0, D_1, D_2)$

if $D_2 \neq \omega D_1$ ***then return*** $\perp$
set $M = D_0 - zD_1$
find the discrete logarithm m *of* M *in the message space (set* $m = \perp$ *if none)*
return m

This game never uses b and gives an null advantage.

2 PMAC Security via Tweakable Block Ciphers

This exercise is inspired from Rogaway: Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC*, ASIACRYPT 2004, LNCS vol. 3329, Springer.*

A tweakable block cipher is a function pair defined by a block space $\{0,1\}^\ell$, a key space $\mathcal{K}$, and a tweak space $\mathcal{T}$. The functions are $\pi : \mathcal{K} \times \mathcal{T} \times \{0,1\}^\ell \to \{0,1\}^\ell$ and $\pi^{-1} : \mathcal{K} \times \mathcal{T} \times \{0,1\}^\ell \to \{0,1\}^\ell$. (The second function is denoted π^{-1} by abuse of notation. By abuse of notation, we also say that π is the tweakable block cipher.) They must be such that for every $k \in \mathcal{K}$ and $t \in \mathcal{T}$, the functions $x \mapsto \pi(k,t,x)$ and $y \mapsto \pi^{-1}(k,t,y)$ are permutations over $\{0,1\}^\ell$ which are inverse of each other. For more readability, we denote $\pi_k^t(x) = \pi(k,t,x)$.

Given an adversary $\mathcal{A}$ interacting with an oracle $\mathcal{O} : \mathcal{T} \times \{0,1\}^\ell \to \{0,1\}^\ell$, we define

$$\mathsf{Adv}_\pi^{\mathsf{PRP}}(\mathcal{A}) = \Pr[\mathsf{PRP}_\pi(\mathcal{A},1) \to 1] - \Pr[\mathsf{PRP}_\pi(\mathcal{A},0) \to 1]$$

where $\mathsf{PRP}_\pi(\mathcal{A},b)$ is the following game.

Game $\mathsf{PRP}_\pi(\mathcal{A},b)$
- pick $k \in \mathcal{K}$ at random
- pick a random function Π from $\mathcal{T}$ to the set of permutations of $\{0,1\}^\ell$
- $\mathcal{A}^{\mathcal{O}} \to z$
- **return** z

Oracle $\mathcal{O}(t,x)$:
- **if** $b = 1$ **then**
  - **return** $\pi_k^t(x)$
- **else**
  - **return** $(\Pi(t))(x)$
- **end if**

In what follows, we assume that $\{0,1\}^\ell$ is given a field structure with addition $\oplus$ and multiplication. We also consider an injective function mapping $t \in \mathcal{T}$ to a nonzero field element α_t.

Given a block cipher $C : \mathcal{K} \times \{0,1\}^\ell \to \{0,1\}^\ell$, we define $\pi = \mathsf{XE}_C$ the tweakable block cipher by

$$\pi_k^t(x) = C_k\left(x \oplus \alpha_t \cdot C_k(0)\right)$$

Below, we define a (simplified) version of PMAC. We consider the message space $\mathcal{M}$ of finite sequences of blocks in $\{0,1\}^\ell$ with a number of blocks bounded by B, and $\mathcal{T} = \{1,\ldots,B\} \times \{2,3\}$. We let $\pi = \mathsf{XE}_C$ and

$$\mathsf{PMAC}(k,(x_1,\ldots,x_n)) = \pi_k^{n,3}\left(\pi_k^{1,2}(x_1) \oplus \cdots \oplus \pi_k^{n-1,2}(x_{n-1}) \oplus x_n\right)$$

with $n \leq B$. That is, the inner tweaks are pairs $t = (i,2)$ with i being the block index and the outer tweak is the pair $t = (n,3)$ with n being the number of blocks. If we denote $L = C_k(0)$ and $\Delta_t = \alpha_t \cdot L$, we have

$$\mathsf{PMAC}(k,(x_1,\ldots,x_n)) = C_k\left(C_k(x_1 \oplus \Delta_{1,2}) \oplus \cdots \oplus C_k(x_{n-1} \oplus \Delta_{n-1,2}) \oplus x_n \oplus \Delta_{n,3}\right)$$

With the tag length ℓ, a complete last block x_n, and an appropriate definition for α_t, this is the standard PMAC authentication code.

Q.1 When $\mathcal{T}$ has a single element, prove that a tweakable block cipher π over the tweak space $\mathcal{T}$ is totally defined by a block cipher C over the same key space and block space, and that the PRP security of π is equivalent to the CPA security of C against distinguishers (i.e. the real-or-ideal cipher security which has been seen in the class).

Let t_0 denote the unique element of $\mathcal{T}$. We define $C_k(x) = \pi_k^{t_0}(x)$. Clearly, for every $k \in \mathcal{K}$, $x \mapsto C_k(x)$ is a permutation of $\{0,1\}^\ell$. So, C is a block cipher. We can conversely define π from C by $\pi_k^{t_0}(x) = C_k(x)$.
In the class, we have seen a general definition of CPA security against distinguishers for ciphers which may use nonces and be over a more general message space. When limited to block ciphers, this notion boils down to

pick $k \in \mathcal{K}$ at random
pick a random permutation σ of $\{0,1\}^\ell$
$\mathcal{A}^{\mathcal{O}} \to z$
***return** z*

Oracle $\mathcal{O}(x)$:
if** $b = 1$ **then
  ***return** $\pi_k(x)$*
else
  ***return** $\sigma(x)$*
end if

Clearly, a function Π from $\{t_0\}$ to the set of permutation of $\{0,1\}^\ell$ is equivalent to a permutation σ of $\{0,1\}^\ell$. So, the two games are totally equivalent.

Q.2 When the function mapping $t \in \mathcal{T}$ to α_t is not injective, prove that XE_C is not a secure tweakable block cipher by describing an adversary achieving a high advantage.

If the function is not injective, there exist two different tweaks t_1 and t_2 such that $\alpha_{t_1} = \alpha_{t_2}$. We define the following adversary.
$\mathcal{A}^{\mathcal{O}}$:
$y_1 \leftarrow \mathcal{O}(t_1, 0)$
$y_2 \leftarrow \mathcal{O}(t_2, 0)$
***return** $1_{y_1 = y_2}$*
In $\mathsf{PRP}_\pi(\mathcal{A}, 1)$, we always have

$$y_1 = C_k(\alpha_{t_1} \cdot C_k(0)) = C_k(\alpha_{t_2} \cdot C_k(0)) = y_2$$

so the game returns 1 with probability 100%. In $\mathsf{PRP}_\pi(\mathcal{A}, 0)$, $\sigma_1 = \Pi(t_1)$ and $\sigma_2 = \Pi(t_2)$ are independent random permutations, so $y_1 = \sigma_1(0)$ and $y_2 = \sigma_2(0)$ are independent random blocks, so the game returns 1 with probability $2^{-\ell}$. We have $\mathsf{Adv}_\pi^{\mathsf{PRP}}(\mathcal{A}) = 1 - 2^{-\ell}$.

Q.3 We consider the PRF security of PMAC over the key space $\mathcal{K}$, the input space $\mathcal{M}$, and the output space $\{0,1\}^\ell$. We denote by $\mathsf{PMAC}[\pi]$ the authentication code which is defined by the tweakable block cipher π. We also denote by π^* the ideal tweakable block cipher over the same tweak space and block space. (I.e., the key space of π^*

is the set of all functions from the tweak space to the set of block permutations and $\pi^*(k,t,x) = (k(t))(x)$.)

Given an adversary $\mathcal{A}$ against the PRF security of PMAC which is limited to a time complexity T (which include the running time and the size of the code of adversary) and to a number of queries with a total length of q blocks, prove that we can construct an adversary $\mathcal{B}$ against the PRP security of π with number of queries limited to q and a complexity of T plus a small overhead and such that $\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{PMAC}[\pi]}(\mathcal{A}) \leq \mathsf{Adv}^{\mathsf{PRP}}_{\pi}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{PMAC}[\pi^*]}(\mathcal{A})$.

The tweak space and block space of π and π^* are the same, so the ideal PRF games $\mathsf{PRF}_{\mathsf{PMAC}[\pi]}(\mathcal{A},0)$ and $\mathsf{PRF}_{\mathsf{PMAC}[\pi^*]}(\mathcal{A},0)$ are the same. Hence,

$$\begin{aligned}
\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{PMAC}[\pi]}(\mathcal{A}) &= \Pr[\mathsf{PRF}_{\mathsf{PMAC}[\pi]}(\mathcal{A},1) \to 1] - \Pr[\mathsf{PRF}_{\mathsf{PMAC}[\pi]}(\mathcal{A},0) \to 1] \\
&= \Pr[\mathsf{PRF}_{\mathsf{PMAC}[\pi]}(\mathcal{A},1) \to 1] - \Pr[\mathsf{PRF}_{\mathsf{PMAC}[\pi^*]}(\mathcal{A},0) \to 1] \\
&= \big(\Pr[\mathsf{PRF}_{\mathsf{PMAC}[\pi]}(\mathcal{A},1) \to 1] - \Pr[\mathsf{PRF}_{\mathsf{PMAC}[\pi^*]}(\mathcal{A},1) \to 1]\big) + \\
&\quad \big(\Pr[\mathsf{PRF}_{\mathsf{PMAC}[\pi^*]}(\mathcal{A},1) \to 1] - \Pr[\mathsf{PRF}_{\mathsf{PMAC}[\pi^*]}(\mathcal{A},0) \to 1]\big) \\
&= \big(\Pr[\mathsf{PRF}_{\mathsf{PMAC}[\pi]}(\mathcal{A},1) \to 1] - \Pr[\mathsf{PRF}_{\mathsf{PMAC}[\pi^*]}(\mathcal{A},1) \to 1]\big) + \\
&\quad \mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{PMAC}[\pi^*]}(\mathcal{A})
\end{aligned}$$

The two real games $\mathsf{PRF}_{\mathsf{PMAC}[\pi]}(\mathcal{A},1)$ and $\mathsf{PRF}_{\mathsf{PMAC}[\pi^*]}(\mathcal{A},1)$ can be simulated by a single adversary $\mathcal{B}^{\mathcal{O}}$ with access to an oracle which is implementing either π or π^*:

$\mathcal{B}^{\mathcal{O}}$:

1: *simulate $\mathcal{A}^{\mathsf{Sub}} \to z$ with subroutine $\mathsf{PMAC}[\mathcal{O}]$*
2: ***return** z*

The subroutine is just implementing the PMAC algorithm with oracle access to evaluate the tweakable block cipher. Hence, $\mathcal{B}$ has a small complexity overhead (the complexity of the PMAC construction) and same number of queries, but playing the PRP game. We have

$$\mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{PMAC}[\pi]}(\mathcal{A}) = \mathsf{Adv}^{\mathsf{PRP}}_{\pi}(\mathcal{B}) + \mathsf{Adv}^{\mathsf{PRF}}_{\mathsf{PMAC}[\pi^*]}(\mathcal{A})$$

Q.4 Given an adversary $\mathcal{B}$ against the PRP security of $\pi = \mathsf{XE}_C$ which is limited to a time complexity T' (which include the running time and the size of the code of adversary) and to a number of queries of q, prove that we can construct an adversary $\mathcal{C}$ against the PRP security of C with number of queries limited to $q+1$ and a complexity of T plus a small overhead and such that $\mathsf{Adv}^{\mathsf{PRP}}_{\pi}(\mathcal{B}) \leq \mathsf{Adv}^{\mathsf{PRP}}_{C}(\mathcal{C}) + \mathsf{Adv}^{\mathsf{PRP}}_{\pi'}(\mathcal{B})$, where $\pi' = \mathsf{XE}_{C^*}$ and C^* is an ideal block cipher over the same block space.

We apply the same method as in the previous question.
The and block space of C and C^* are the same, so the ideal PRF games $\mathsf{PRP}_\pi(\mathcal{B},0)$ *and* $\mathsf{PRP}_{\pi'}(\mathcal{B},0)$ *are the same. Hence,*

$$\begin{aligned}
\mathsf{Adv}_\pi^{\mathsf{PRP}}(\mathcal{B}) &= \Pr[\mathsf{PRP}_\pi(\mathcal{B},1)\to 1] - \Pr[\mathsf{PRP}_\pi(\mathcal{B},0)\to 1] \\
&= \Pr[\mathsf{PRP}_\pi(\mathcal{B},1)\to 1] - \Pr[\mathsf{PRP}_{\pi'}(\mathcal{B},0)\to 1] \\
&= (\Pr[\mathsf{PRP}_\pi(\mathcal{B},1)\to 1] - \Pr[\mathsf{PRP}_{\pi'}(\mathcal{B},1)\to 1]) + \\
&\quad\ (\Pr[\mathsf{PRP}_{\pi'}(\mathcal{B},1)\to 1] - \Pr[\mathsf{PRP}_{\pi'}(\mathcal{B},0)\to 1]) \\
&= (\Pr[\mathsf{PRP}_\pi(\mathcal{B},1)\to 1] - \Pr[\mathsf{PRP}_{\pi'}(\mathcal{B},1)\to 1]) + \\
&\quad\ \mathsf{Adv}_{\pi'}^{\mathsf{PRP}}(\mathcal{B})
\end{aligned}$$

The two real games $\mathsf{PRP}_\pi(\mathcal{B},1)$ *and* $\mathsf{PRP}_{\pi'}(\mathcal{B},1)$ *can be simulated by a single adversary* $\mathcal{C}^{\mathcal{O}}$ *with access to an oracle which is implementing either* C *or* C^**:*
$\mathcal{C}^{\mathcal{O}}$*:*

1: *simulate* $\mathcal{B}^{\mathsf{Sub}} \to z$ *with subroutine* $\mathsf{XE}_{\mathcal{O}}$
2: ***return*** z

The subroutine is just implementing the XE *algorithm with oracle access to evaluate the block cipher. The subroutine may call the oracle with input* 0 *only once to get* $L = C_k(0)$*. Hence,* $\mathcal{C}$ *has a small complexity overhead (the complexity of the* XE *construction) and a number of queries of* $q+1$ *(the extra one is to get* L*), but playing the PRP game. We have*

$$\mathsf{Adv}_\pi^{\mathsf{PRP}}(\mathcal{B}) = \mathsf{Adv}_C^{\mathsf{PRP}}(\mathcal{C}) + \mathsf{Adv}_{\pi'}^{\mathsf{PRP}}(\mathcal{B})$$

Q.5 Given an adversary $\mathcal{B}$ against the PRP security of $\pi' = \mathsf{XE}_{C^*}$ which is limited to a number of queries of q', prove that $\mathsf{Adv}_{\pi'}^{\mathsf{PRP}}(\mathcal{B}) \le \mathsf{Adv}_{\pi''}^{\mathsf{PRF}}(\mathcal{B}) + \frac{q'^2}{2^\ell}$, where $\pi'' = \mathsf{XE}_{F^*}$ and F^* is an ideal random function from $\{0,1\}^\ell$ to $\{0,1\}^\ell$.

By using the same method as in the previous question, we obtain

$$\mathsf{Adv}_{\pi'}^{\mathsf{PRP}}(\mathcal{B}) = \mathsf{Adv}_{C^*}^{\mathsf{PRF}}(\mathcal{C}) + \mathsf{Adv}_{\pi''}^{\mathsf{PRP}}(\mathcal{B})$$

where $\mathcal{C}$ *now tries to distinguish* C^* *from* F^**, which is the PRF security of* C^**. We have seen in the course (besides the Luby-Rackoff theorem) that* $\mathsf{Adv}_{C^*}^{\mathsf{PRF}}(\mathcal{C}) \le q'^2 2^{-1-\ell}$*. Essentially,* C^* *and* F^* *behave the same as long as the oracle* F^* *does not show a collision. This bad event occurs with probability bounded by* $\frac{q'^2}{2\cdot 2^\ell}$*. Then, the result comes from the difference lemma.*
Similarly, we have $\mathsf{Adv}_{\pi''}^{\mathsf{PRP}}(\mathcal{B}) \le \mathsf{Adv}_{\pi''}^{\mathsf{PRF}}(\mathcal{B}) + q'^2 2^{-1-\ell}$*.*

Q.6 Given an adversary $\mathcal{B}$ against the PRF security of $\pi'' = \mathsf{XE}_{F^*}$ which is limited to a number of queries of q', prove that $\mathsf{Adv}_{\pi''}^{\mathsf{PRF}}(\mathcal{B}) \le \frac{q'^2}{2^\ell}$.

We assume without loss of generality that $\mathcal{B}$ never queries the oracle twice with the same query. We consider the fail event that the $\mathsf{PRF}_{\pi''}(\mathcal{B}, 1)$ game evaluates F^* twice with the same input. The first F^* evaluation is to compute $L = F^*(0)$. The first time the fail event happens, the game wants to evaluate π'' on a block x with tweak t, where (t, x) was not evaluated before. There are two cases for the fail event to happen at this time: that $x \oplus \alpha_t \cdot L = 0$ or that $x \oplus \alpha_t \cdot L = x' \oplus \alpha_{t'} \cdot L$ for a previous evaluation for (t', x'). If this happens, it means that $L = \alpha_t^{-1} \cdot (x)$ or $L = (\alpha_t \oplus \alpha_{t'})^{-1} \cdot (x \oplus x')$, for one of the previous (t', x') evaluations. At the ith evaluation, it means that L belong to a set of i values that the adversary knows. As long as the fail event does not occur, the adversary receives no information about L except that it does not belong to those sets. Since L is random, it means that the fail event corresponds to a random guess of L when given ${q'}^2$ trials. So, the fail event occurs with probability bounded by ${q'}^2/2^\ell$ and we conclude with the difference lemma.