Notes of Cryptography

Squirrel

April 15, 2025

Preface

Course

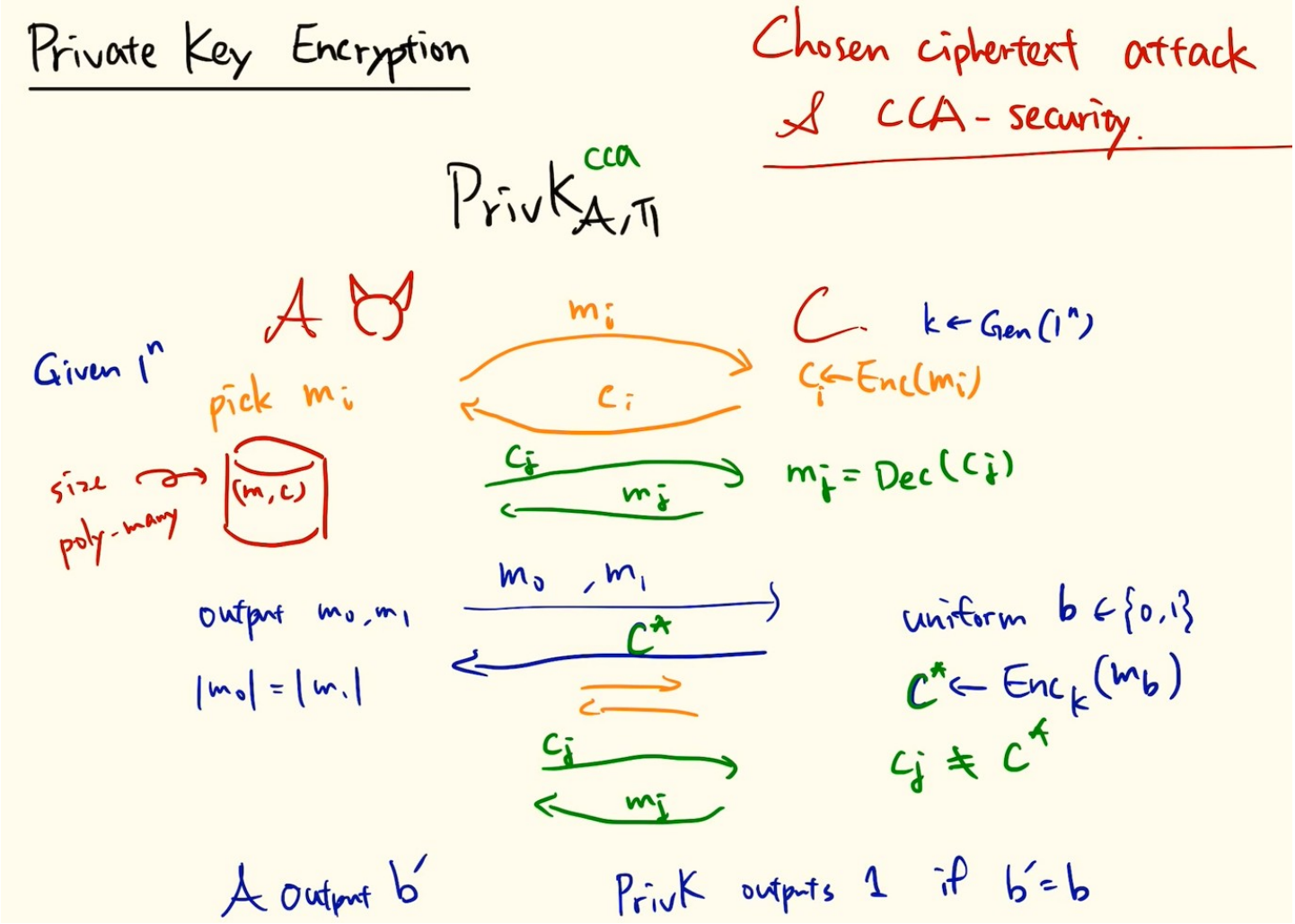
密碼學設計與分析 Cryptography Design and Analysis (11320IIS500900) in NTHU

7 L7

7.1 CCA-security

CCA = Chosen Ciphertext Attack

允許攻擊者使用 decryption oracle，給予其密文，它會回傳明文。但限制攻擊者不可使用欲 challenge 的密文 c^*。



§ Remark on CCA-security

CCA => Lunch time attack

Is CCA realistic（現實可行）?
No, but still have weak decryption oracle which only leak 1-bit message from decrypted ciphertext, which is suffice to learn the entire message (plaintext).

§ Padding for Arbitrary Length

Assuming block size is L bytes.
If message length = $L(t-1)+2$, then we need padding which is $L-2$ bytes.

One of the pratical padding solution is PKCS #5:

- Block length: L byte
- b bytes to apppend the message to a multiple of L, where $1 \leqslant b \leqslant L$. Note that $b \neq 0$.
- Append b (encoded 1 byte), b time(s).
  i.e., b=3 $\Rightarrow$ 0x03 03 03, where underlines indicate 1 byte.

Quiz

當最後一個 block 本來就是滿的，應該如何進行 padding？

Ans：
額外補上一個 block，並在每個 byte 填入 block size 大小的數值。
E.g. 設 block size 為 8 bytes，則額外新增一個 block，並在八個 bytes 中填入 0x08。

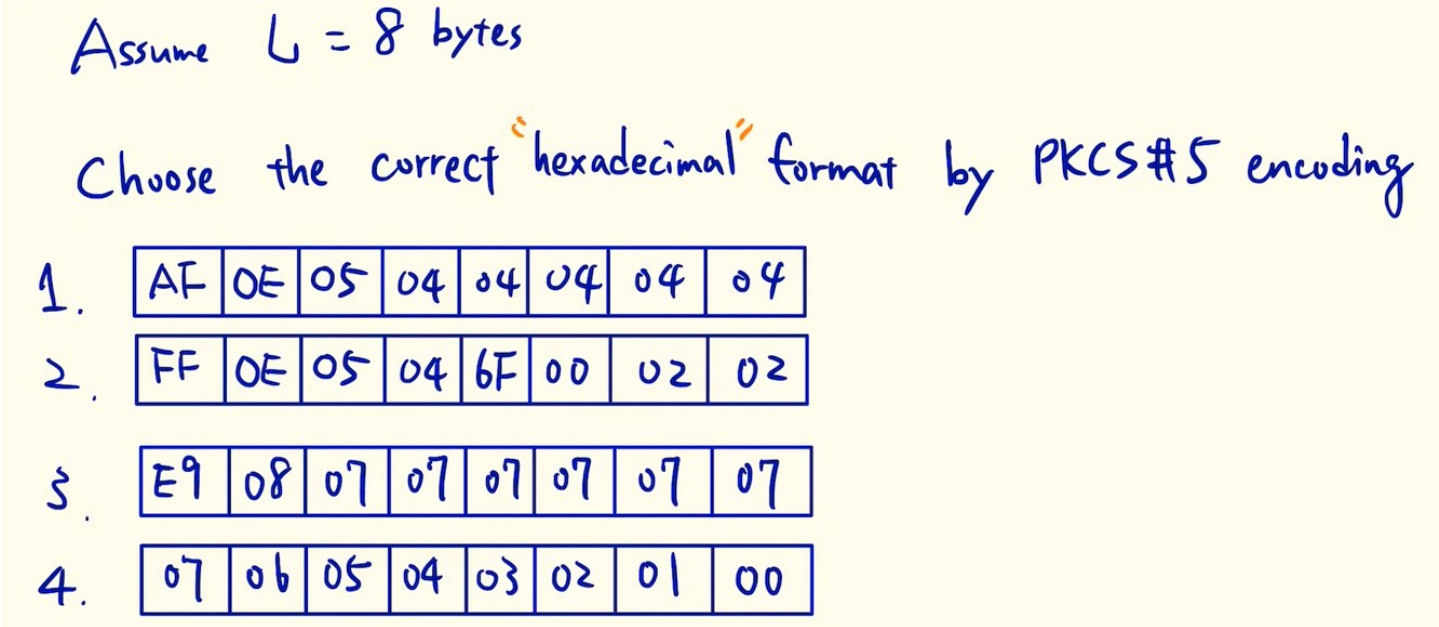
§ Decryption

使用 CBC mode 解密。

在 decryption 後檢查 encoded data，設最後一個 byte 為 b:

- 若 $b = 0$ 或 $b > L$，return error
- 若最後 b 個 bytes 並不全都等於 b，return error
- 否則，去除 padding 的部分，並 return message

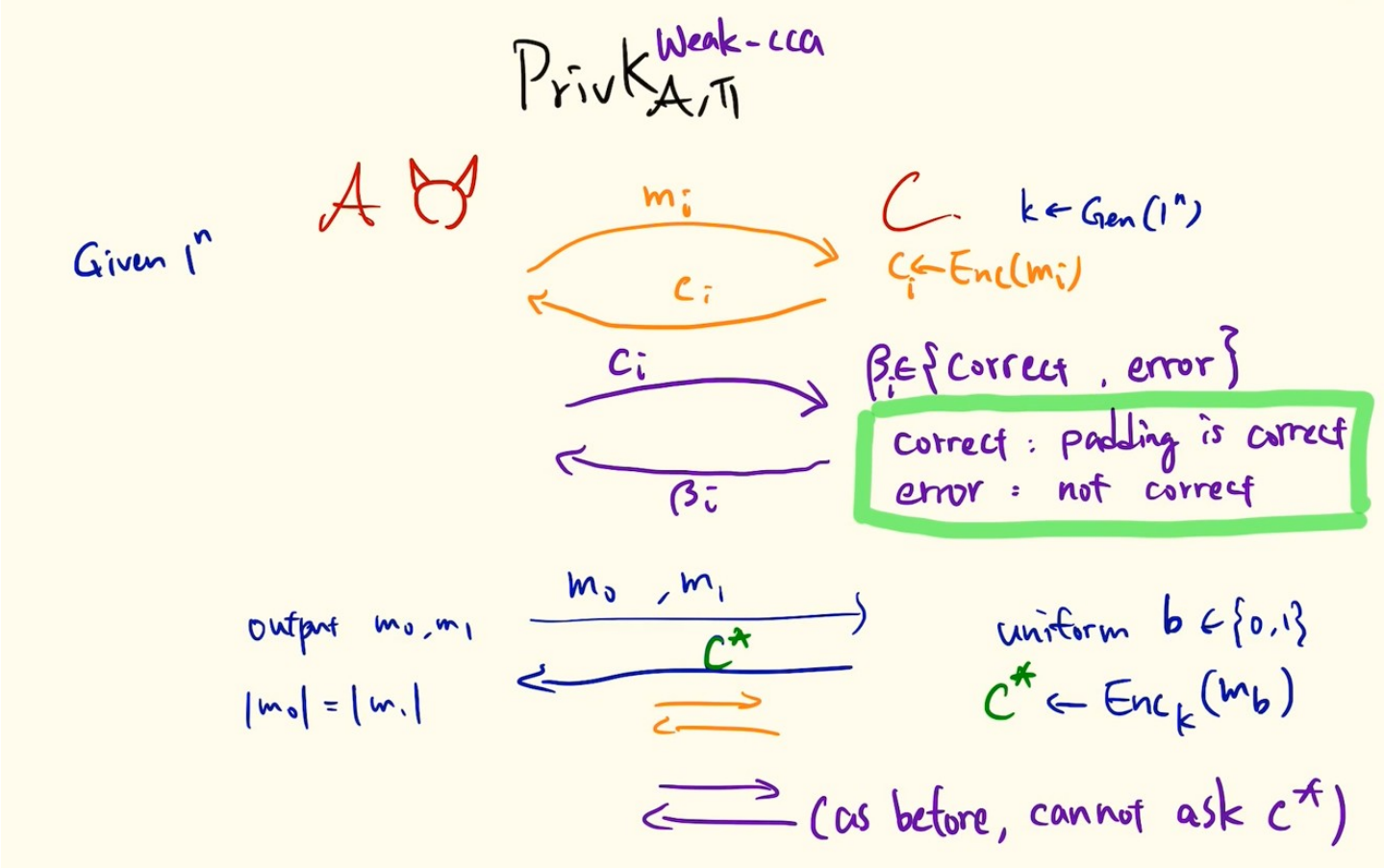
Quiz



Ans: (2)

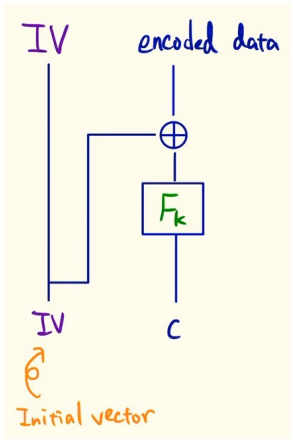
§ Weak CCA with Padding Oracle

這裡出現了一種新的 oracle。給定 ciphertext，它會 return padding 是正確或錯誤。



§ Padding Oracle Attack

基本原理

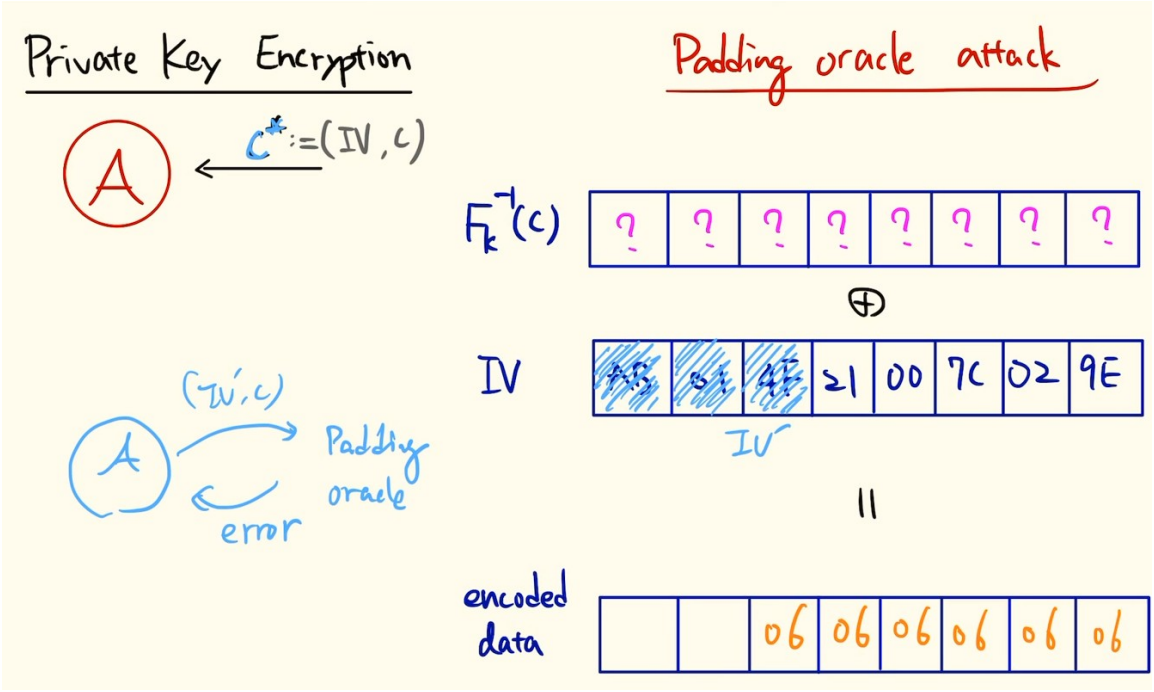


其中 encoded data = $F_k^{-1}(c) \oplus IV$

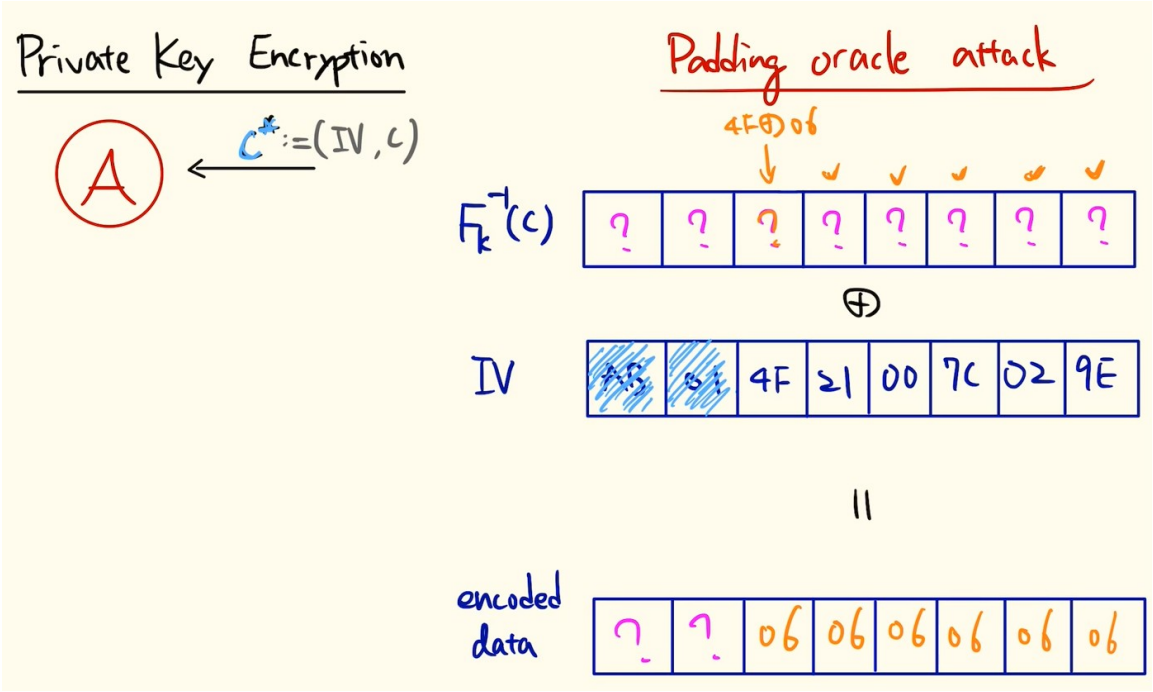
我們可以觀察到，若 attacker A 修改了 IV 的第 i 個 byte，這個動作只會影響到 encoded data 的第 i 個 byte。(∵ CBC 使用 XOR 運算)

攻擊過程

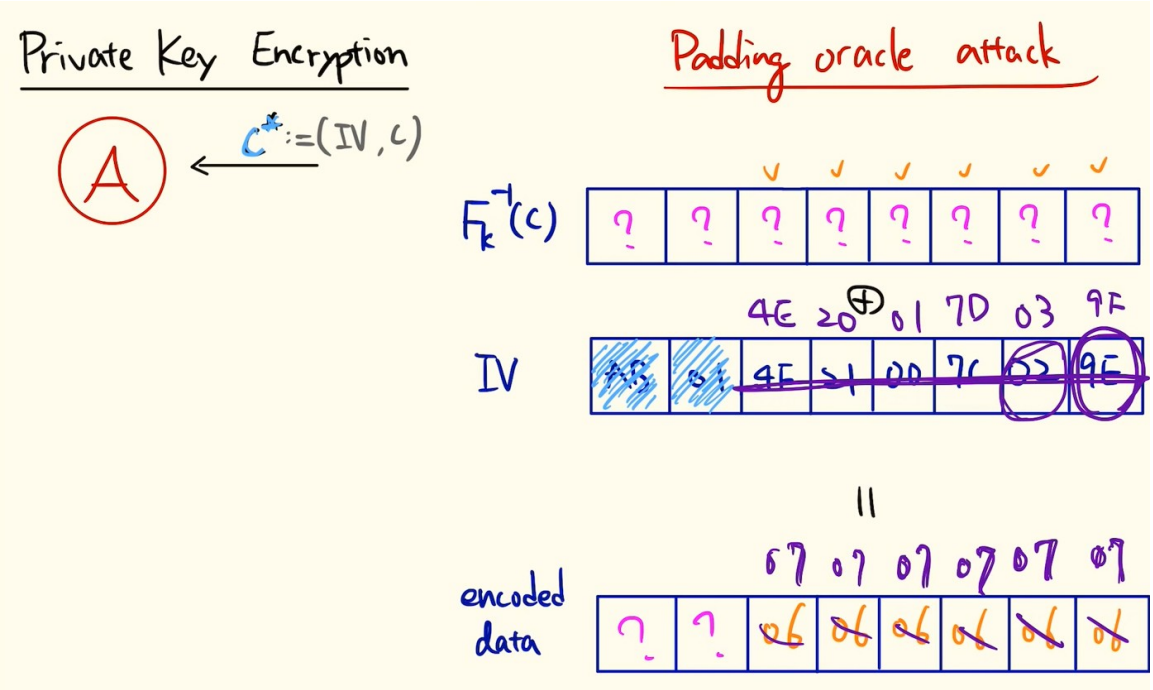
Attacker 會先由左至右逐個修改 byte，並在每次修改完之後都詢問 oracle。直到 oracle return error，該 byte 到最右邊即為 padding bytes：



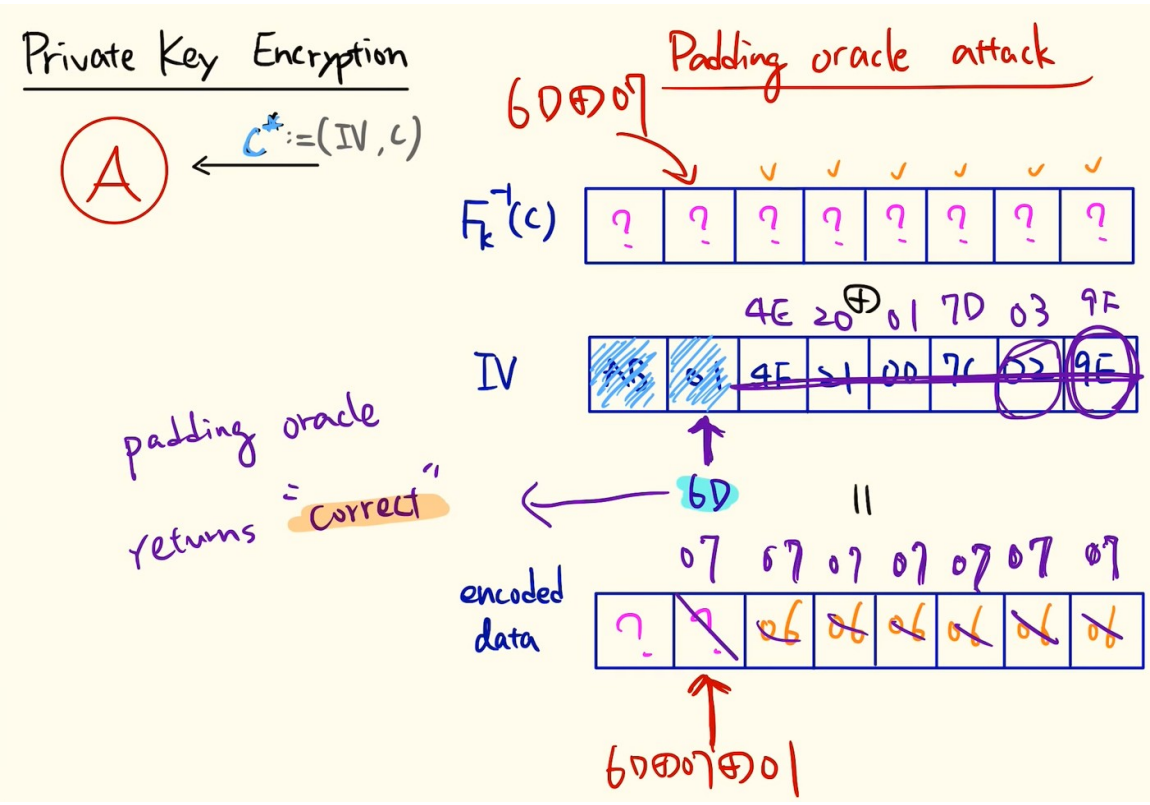
之後 attacker 便可藉由 IV 和 encoded data 反推 $F_k^{-1}(c)$ 的 padding 部分為何：



為了得到非 padding 部分的 message 為何，attacker 可以藉由修改 padding 部分為原本的值再加一，並重新計算新 IV 的 padding 部分：



再來是持續修改非 padding 部分的最後（最右）一個 byte，直到 padding oracle return correct，attacker 就可以知道此時的 encoded data 中的對應 byte 為原本的 padding 值再加一。最後依序計算 $F_k^{-1}(c)$ 得到密鑰，再將密鑰與 IV 做 XOR 得到原本的 encoded data。



持續進行這些步驟就可以得到完整的 encoded data 為何。

Remark on Padding Oracle Attack

- # of pading bytes: $< L$ padding oracle queries（確定 padding byte 的數量所需的次數）
- contain of one byte of the message: $\leqslant 2^8 = 256$ padding oracle queries（最多嘗試 256 次即可猜到 encoded data 中非 padding 部分的一個 byte）
- In $PrivK_{A,\Pi}^{weak-cca}$ with padding oracle, A choose m_0, m_1 such that $|m_0| = |m_1|$ and last significant byte of $m_0$is different from correspondence of m_1. And it only needs $\leqslant L + 2^8$ padding

oracle queries to finish it.

7.2 Message Authentication Code (MAC)

Secrecy：由 Enc 提供、adversary 無法知道訊息內容、不能涵蓋所有的 concerns（例如：訊息篡改）
Integrity：確保訊息不被篡改 (tampering)、驗證訊息的正確性

MAC = Message Authentication Code

§ Syntax

Alice 傳 message m 及一個可以驗證 message 的 tag t 給 Bob，而 Bob 在收到訊息後，透過 t 來驗證 m。

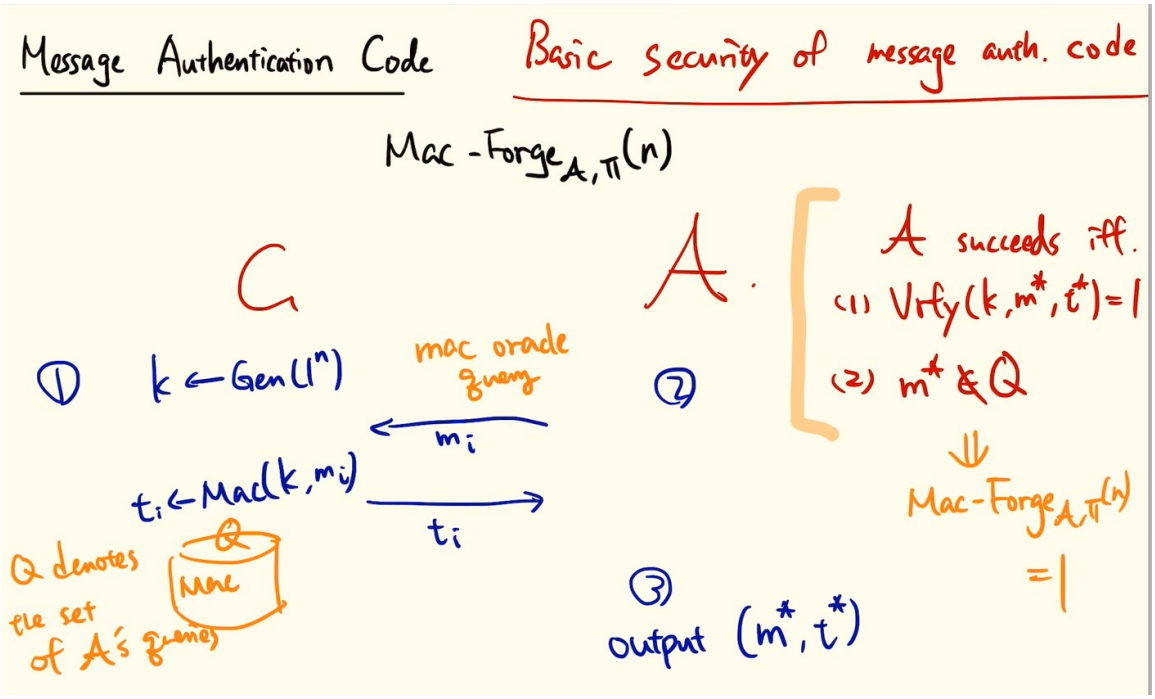
Π is a MAC construction. $\Pi = (\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrfy})$

- Key generation $\mathrm{Gen}(1^n) \to k$: a key
- Message authentication code $\mathrm{Mac}(k, m) \to t$: a tag
- Verification $\mathrm{Vrfy}(k, m, t) := 0$ or 1, where 0 stands for rejection, 1 stands for acceptance.

Remark

- m is not hidden for Vrfy.
- 對於一個 deterministic MAC，Vrfy 在做的事情就是重新建立一次 t，並比較其是否與接收到的 t 相同。
- If $m \in \{0,1\}^{l(n)}$ where $l(n)$ is a polynomial, then MAC is fixed-length MAC.

§ Experiment $\mathrm{MAC\text{-}Forge}_{A,\Pi}(n)$



有 challenger C 和 adversary A：

Step 1： C 產生 key，即 $k \leftarrow \mathrm{Gen}(1^n)$

Step 2： A 選擇一個 message m_i，以此詢問 C，而 C 再 return 一個計算過後的 tag $t_i \leftarrow \mathrm{Mac}(k, m_i)$ 給 A。此外，C 可以使用一個 list Q 來收集所有來自 A 的 queries。

Step 3： A outputs 他的偽造 (m^*, t^*)

A 成功的條件 (即 $\mathrm{MAC\text{-}Forge}_{A,\Pi}(n) = 1$) 有兩個 (都要符合，if and only if) :
(1) $\mathrm{Vrfy}(k, m^*, t^*) = 1$
(2) $m^* \notin Q$

Remark

$\mathrm{MAC\text{-}Forge}$ 的 strong 版本是 $\mathrm{MAC\text{-}sForge}$。
$\mathrm{MAC\text{-}sForge}$：比 $\mathrm{MAC\text{-}Forge}$ 寬鬆一點，只要求 (m^*, t^*) 也不在 Q 中，即 $(m^*, t^*) \notin Q$。

If a deterministic MAC satisfies existential unforgeability in $\mathrm{MAC\text{-}Forge}$, it also satisifeis strong unforgeability in $\mathrm{MAC\text{-}sForge}$.
解釋：因為 deterministic MAC 會將一個 m 只對應到一個 t，所以如果 m^* 不在 Q 中，則與之對應的 t^* 也不會在 Q 中。因此 $(m^*, t^*) \notin Q$。

§ Strong Verison of Previous Experiment $\mathrm{MAC\text{-}sForge}_{A,\Pi}(n)$

大致和之前相同，唯一不同處在於 A 成功的第二個條件改成 $(m^*, t^*) \notin Q$。這意味著 A 可以向 C query m^*，但只要可以偽造另一組不在 Q 中的 (m^*, t^*) 並且仍可以使用它通過 Vrfy，那麼就算 A 成功了。

只要在 $\mathrm{MAC\text{-}sForge}$ 是安全的，那它在 $\mathrm{MAC\text{-}Forge}$ 也是安全的。反之則不成立。

Definition 9 (Strong Security)

$$\Pr[\mathrm{MAC\text{-}sForge}_{A,\Pi}(n) = 1] \leqslant \mathrm{negl}(n)$$

§ Security Definition of MAC

Definition 10
A message authentication code $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Vrfy})$ is existentially unforgeable under adaptive chosen message attacks, if for all PPT adversaries A there is a ngeligible function negl such that

$$\Pr[\mathrm{MAC\text{-}Forge}_{A,\Pi}(n) = 1] \leqslant \mathrm{negl}(n)$$

§ Construction

Pseudorandom Function

Pseudorandom function (PRF) 的定義請參見 6.1 CPA-secure Encryption。

Construction of Fixed-length MAC

Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a PRF.
$\Pi = (\mathrm{Gen}, \mathrm{Mac}, \mathrm{Vrfy})$ is a fixed-length MAC for message fo length n.

- $\mathrm{Gen}(1^n)$: uniform $k \in \{0,1\}^n$
- $\mathrm{Mac}(k, m)$: on input k and message $m \in \{0,1\}^n$, outputs a tag $t = F_k(m)$ where t is t-bit tag.
- $\mathrm{Vrfy}(k, m, t)$: on input (k, m, t), outputs 1 if and only if $t = F_k(m)$; otherwise, outputs 0.

8 L8

接著 L7 的 MAC。

8.1 MAC

§ Security of Fixed-length MAC

Theorem 7
If F is PRF, then Π is existentially unforgeable under adaptive chosen message attack (in $\mathrm{MAC\text{-}Forge}$ experiment).

Proof (Security proof of PRF-based construction)

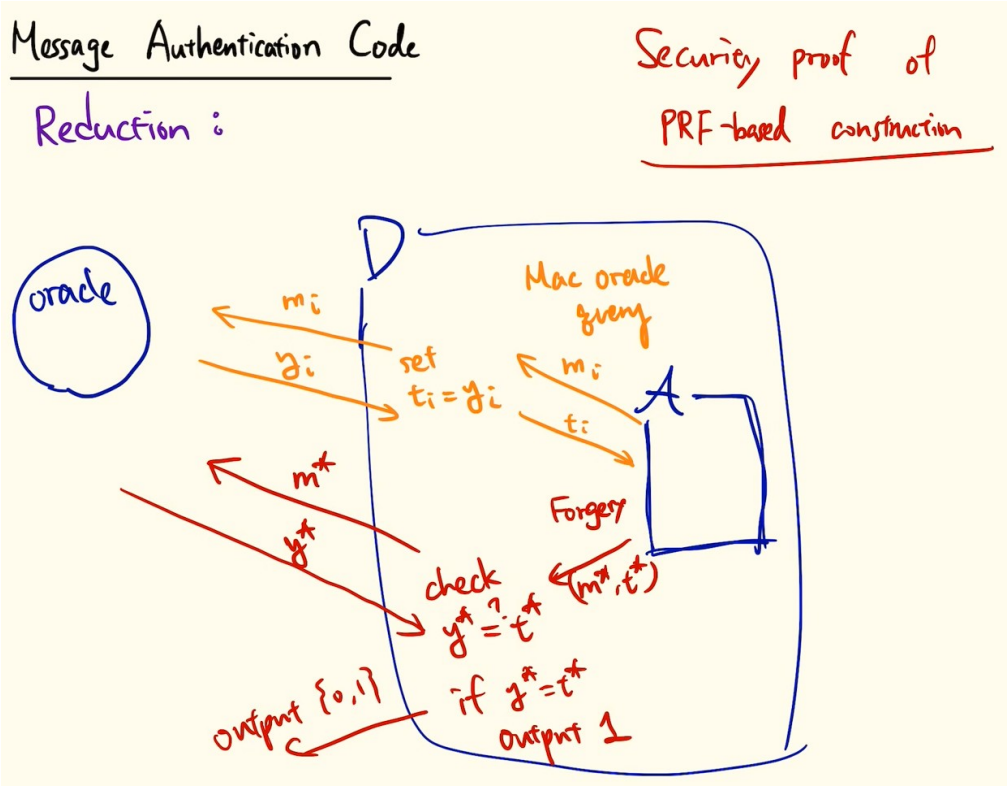
Consider $\widetilde{\Pi} = (\widetilde{\mathrm{Gen}}, \widetilde{\mathrm{Mac}}, \widetilde{\mathrm{Vrfy}})$ constructed by the random function.
Then we know that $\Pr[\mathrm{MAC\text{-}Forge}_{A,\tilde{\Pi}}(n) = 1] \leqslant 2^{-n}$
because for any $m \notin Q$, the tag $t = f(m)$ is uniformly distributed in $\{0,1\}^n$ from A's view.

Our goal of proof:

$$
\begin{aligned}
& |\Pr[\mathrm{MAC\text{-}Forge}_{A,\Pi}(n) = 1] - \Pr[\mathrm{MAC\text{-}Forge}_{A,\tilde{\Pi}}(n) = 1]| \leqslant \mathrm{negl}(n) \\
\Rightarrow \quad & \Pr[\mathrm{MAC\text{-}Forge}_{A,\Pi}(n) = 1] \leqslant 2^{-n} + \mathrm{negl}(n) \\
\Rightarrow \quad & \Pr[\mathrm{MAC\text{-}Forge}_{A,\Pi}(n) = 1] \leqslant \mathrm{negl}(n)
\end{aligned}
$$

We then contruct a distringuisher D who want to break PRF security. D can do oracle access to some functions, and finally determines such function is pseudorandom (i.e. F_k for uniform $k \in \{0,1\}^n$) or truly random (i.e. f for uniform $f \in \mathrm{Func}_n$).

Reduction:



(i) If D's oracle is PRF,

$$\Pr[D^{F_k(\cdot)}(1^n) = 1] = \Pr[\text{MAC-Forge}_{A,\Pi}(1^n) = 1]$$

(ii) If D's oracle is random function,

$$\Pr[D^{f(\cdot)}(1^n) = 1] = \Pr[\text{MAC-Forge}_{A,\tilde{\Pi}}(n) = 1] \leqslant 2^{-n}$$

(iii) If By assumption,

$$|\Pr[D^{F_k(\cdot)}(1^n) = 1] - \Pr[D^{f(\cdot)}(1^n) = 1]| \leqslant \text{negl}(n)$$

綜合上面三點，

$$\Pr[\text{MAC-Forge}_{A,\Pi}(n) = 1] \quad \leqslant 2^{-n} + \text{negl}(n) \quad \leqslant \text{negl}(n)$$

□

Quiz

Say $\Pi = (\text{Gen}, \text{Mac}, \text{Vrfy})$ is a MAC with tag size $t(n)$.
Show that if $t(n) = O(\log(n))$, then Π is not secure MAC.
(Hint: PPT adversary runs **random guess**)

§ MAC for Arbitrary-length Messages (domain extension of MAC)

Let $\Pi' = (\text{Gen}', \text{Mac}', \text{Vrfy}')$ be a fixed-length MAC for message size n, and $\Pi = (\text{Gen}, \text{Mac}, \text{Vrfy})$ is a MAC for arbitrary-length messages.

- Gen: identical to $\text{Gen}'(1^n) \to k$, where $k \in \{0,1\}^n$
- Mac: on input $k \in \{0,1\}^n$ and a message $\{0,1\}^*$ of length $l < \frac{n}{4}$.
  (i) Parse m into d blocks $m_1, m_2, \ldots, m_d$ (each of length $\frac{n}{4}$).
  The final block is padded with 0s.
  (ii) choose a uniform $r \in \{0,1\}^{\frac{n}{4}}$

(iii) For $i = 1, 2, \ldots, d$, compute $t_i \leftarrow \mathrm{Mac}'_k(r \,||\, l \,||\, i \,||\, m_i)$ where all the length of r, l, i, m are $\frac{n}{4}$ bits respectively.
(iv) Output $t = (r, t_1, t_2, \ldots, t_d)$.

- Vrfy: On input k, t, m,
  (i) Parse m and t, such that $m = (m_1, m_2, \ldots, m_d)$ and $t = (r, t_1, t_2, \ldots, t_{d'})$.
  (ii) Output 1 if the below are both satisfied:
  - $d' = d$
  - For $1 \leqslant i \leqslant d$, $\mathrm{Vrfy}'(r \,||\, l \,||\, i \,||\, m_i) = 1$.

Remark of Mac

It seems r, l, i are important for security.

如果 Mac' 中沒有 l，則 $t_i \leftarrow \mathrm{Mac}'_k(r \,||\, i \,||\, m_i \,||0^{\frac{n}{4}})$。
此時 adversary 的可以這樣進行攻擊：

Step 1： 選取一個 m 來詢問 MAC oracle，並從 MAC oracle 收到 $t = (r, t_1, \ldots, t_d)$

Step 2： 偽造 $m^* = (m_1, m_2, \ldots, m_{d-1})$ 和 $t^* = (r, t_1, t_2, \ldots, t_{d-1})$。

由於 m^* 之前並沒有被拿來問 MAC oracle，所以 adversary 是成功的。

這說明 l 是重要的。

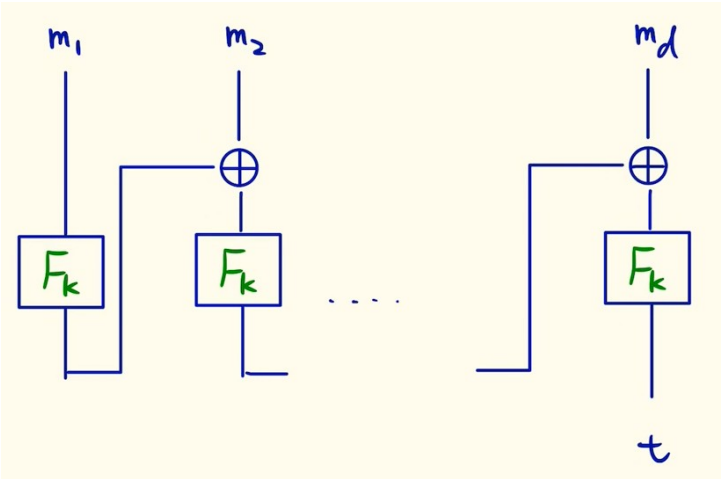
Quiz

In domain extension of MAC, it is in secure if there is no r or i.
(1) Show an attack if no r.
(2) Show an attack if no i.

Theorem 8
If MAC for fixed length Π' is secure, then MAC for arbitrary length Π is secure.

§ CBC-MAC

每個 block 計算完後就跟下一個 block 做 xor 再繼續計算，最終只會得到一個 tag t。



Quiz

Analyze advantages and disadvantages of CBC-MAC and domain extension.
(Hint: compare in performance, security, adjustment of parameters, etc.)