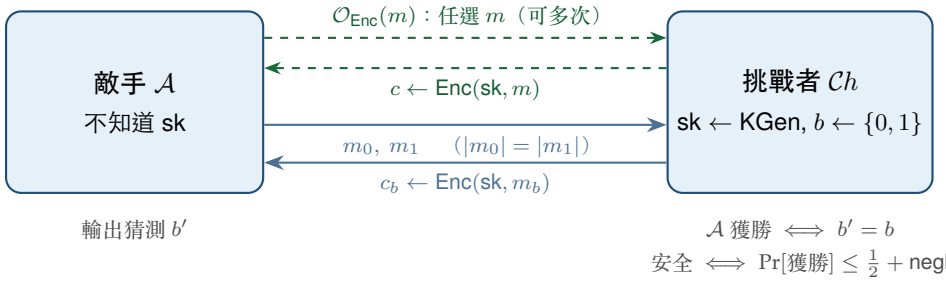
King's College London · 6CCS3CIS Cryptography（2025/26）

Week 2

形式化定義、Kerckhoffs 原則與安全遊戲

完全學習教材 & Problem Set 2 詳解

SKE 與 MAC 的正式定義 • 空間、字母表與函數 • 單射 / 滿射 / 雙射
Kerckhoffs 六原則 • IND-EAV / IND-CPA / IND-CCA • EUF-CMA • PPT 與可忽略函數



教材內容	講義逐頁精讀、延伸補充（Katz–Lindell、Kerckhoffs 原典、真實案例）、圖解、Problem Set 2 全部題目詳解（含程式驗證的 Caesar 密碼資料）、重點整理、中英術語表、自我測驗
對應教材	Second lecture（60 頁投影片）、Problem Set 2（Problems 1–3）、KL-IMC 第 1、3、4、5 章相關節次
適用對象	修習 6CCS3CIS（Level 6）之學生；建議先讀 Week 1 教材；需基本離散數學（集合、函數、機率）

依據 6CCS3CIS 第二週講義（Nguyen & Postlethwaite, 2025-10-09）與 Problem Set 2（Postlethwaite, 2025-10-06）編寫，並參考 Katz–Lindell《Introduction to Modern Cryptography》第三版、Kerckhoffs《La cryptographie militaire》（1883）及相關原始文獻

目錄

導讀：如何使用本教材 **4**

1 SKE 的正式定義（講義 pp. 2–4） **8**

1.1 語法（syntax） 8

1.2 「隨機化」是什麼意思？（講義 p. 3） 8

1.3 正確性（correctness，講義 p. 4） 9

1.4 為什麼需要 $\perp$？ 9

2 MAC 的正式定義（講義 pp. 5–7） **10**

2.1 語法 10

2.2 確定性 Tag 與典範驗證（講義 p. 6） 10

2.3 正確性（講義 p. 7） 11

3 空間 $\mathcal{K}, \mathcal{M}, \mathcal{C}, \mathcal{T}$ 到底是什麼？（講義 pp. 8–9） **12**

3.1 形式上只是集合 12

3.2 字母表、字串與 Kleene 星號 12

3.3 講義要你記住的兩個常見設定（講義 p. 9） 12

4 把 Enc 與 Dec 當成函數（講義 pp. 10–11） **13**

4.1 單射、滿射、雙射 13

4.2 E_{sk} 與 D_{sk}（講義 p. 10） 14

4.3 有多少個函數？（講義 p. 11） 14

4.4 這個計數在密碼學裡的意義 15

5 Kerckhoffs 原則（講義 p. 12） **16**

5.1 六條原則：原文、英譯與中譯 17

5.2 「Kerckhoffs 原則」= 第二條 17

5.3 從 1883 到今天 18

6 SKE 的形式化安全：IND-EAV、IND-CPA、IND-CCA（講義 pp. 13–18） **19**

6.1 直覺與困難（講義 p. 13） 19

6.2 IND-EAV：對竊聽者的不可區分性（講義 p. 14） 19

6.3 IND-CPA：選擇明文攻擊（講義 p. 15） 20

6.4 IND-CCA：選擇密文攻擊（講義 p. 16） 21

6.5 為什麼是「不可區分」？（講義 p. 17） . . . 22
6.6 為什麼是二分之一？（講義 p. 18） . . . 23
6.7 三個概念的強弱關係 . . . 23
6.8 為什麼要求 $|m_0| = |m_1|$？ . . . 23

7 MAC 的形式化安全：EUF-CMA（講義 pp. 19–22） 24
7.1 直覺（講義 p. 19） . . . 24
7.2 EUF-CMA 遊戲（講義 p. 20） . . . 24
7.3 為什麼是「存在性不可偽造」？（講義 p. 21） . . . 25
7.4 為什麼是零？（講義 p. 22） . . . 25
7.5 與 Week 1 的連結 . . . 25

8 哪些敵手？PPT 與可忽略函數（講義 p. 23） 26
8.1 安全參數 . . . 26
8.2 有效率 = PPT . . . 26
8.3 明顯異於 = 不可忽略 . . . 27
8.4 漸近安全 vs. 具體安全 . . . 28

9 Problem 1：On message and ciphertext spaces（函數與無歧義解密） 30
9.1 (i) 判斷五個函數 . . . 30
9.2 (ii) 單射、滿射與集合大小 . . . 31
9.3 題目背景：無歧義解密 . . . 32
9.4 (iii) 滿射但有歧義 . . . 32
9.5 (iv) 滿射且無歧義 . . . 33
9.6 (v) 無歧義 ⇒ 單射 ⇒ $|\mathcal{C}| \geq |\mathcal{M}|$ . . . 33
9.7 (vi) 單射、非滿射、無歧義：最小的必要假設 . . . 34

10 Problem 2：On brute force attacks（暴力破解的極限） 35
10.1 (i) 密文 ck 有幾種讀法？ . . . 35
10.2 (ii) 一則密文一定夠嗎？ . . . 37
10.3 (iii) 最長的「雙解」Caesar 密文 . . . 38
10.4 (iv) 有加密預言機：一次查詢就夠 . . . 39

11 Problem 3：On Kerckhoffs' principles（碼書與 Kerckhoffs 原則） 41
11.1 先把碼書寫成 SKE . . . 41
11.2 (i) 逐條檢視 Kerckhoffs 原則 . . . 42

11.3 (ii) 讓解密更有效率：把金鑰加倍 42
11.4 (iii) 用現代語言重寫 Kerckhoffs 原則 43
11.5 (iv) 違反 Kerckhoffs 原則的災難 45

12 本週重點整理 **49**
12.1 一頁看懂第二週 49
12.2 必背定義 50
12.3 常見誤區 50

13 中英術語對照表 **51**

14 自我測驗（附解答） **53**

參考資料 **55**

導讀：如何使用本教材

課程資訊

課程	6CCS3CIS Cryptography（King's College London，2025/26）
講師	Ngoc Khanh Nguyen、Eamonn W. Postlethwaite
本週講義	Second lecture（2025 年 10 月 9 日，60 張投影片含逐步顯示，實際內容 24 頁）
本週習題	Problem Set 2（2025 年 10 月 6 日，三大題共 14 小題）
課本	Katz & Lindell, *Introduction to Modern Cryptography*, 3rd ed.（簡稱 KL-IMC）
本週對應章節	KL-IMC §1.3（Kerckhoffs 原則）、§3.1（計算安全、PPT、negligible）、§3.2.1（IND-EAV）、§3.4.2（IND-CPA）、§4.2（EUF-CMA）、§5.1.2（IND-CCA）

本週在整門課的位置

第一週用故事（Francis 與 Henry 的羊皮紙）建立直覺：**語法 → 正確性 → 安全性**三層分開、保密不等於完整、安全定義要同時說清「敵手不能做什麼」與「敵手能做什麼」。第二週把這些直覺全部寫成數學：

- SKE 與 MAC 的正式語法（哪些演算法是隨機的、哪些是確定的、$\perp$ 是什麼）；
- 空間 $\mathcal{K}, \mathcal{M}, \mathcal{C}, \mathcal{T}$ 到底是什麼（集合、字母表、Kleene 星號）；
- 把 $\mathsf{Enc}(\mathsf{sk}, \cdot)$ 當成函數來看：單射、滿射、雙射，以及它們的計數；
- 1883 年 Kerckhoffs 的六條原則，以及其中第二條為何成為現代密碼學的基石；
- 用遊戲（game）定義安全：IND-EAV、IND-CPA、IND-CCA（加密）與 EUF-CMA（MAC）；
- 「有效率的敵手」（PPT）與「可忽略的機率」（negligible）這兩個詞的精確意義。

下週開始的歷史密碼（Caesar、Vigenère 等）會不斷回到這些定義：它們為什麼不安全？用哪一個遊戲可以精確說明？

學習目標

讀完本教材你應該能夠：

1. 默寫 SKE 與 MAC 的正式定義（含空間、隨機性、正確性），並解釋為何 Dec 與 Vf 是確定性的、為何需要 $\perp$。
2. 判斷一個函數是否單射 / 滿射 / 雙射，計算 $\mathcal{M} \to \mathcal{C}$ 的函數、單射與雙射個數，並證明「解密無歧義 $\Rightarrow E_{\mathsf{sk}}$ 單射 $\Rightarrow |\mathcal{C}| \geq |\mathcal{M}|$」。

3. 列出 Kerckhoffs 六原則，分辨「Kerckhoffs 原則」（第二條）與其他五條，並用現代語言重寫。
4. 畫出 IND-EAV、IND-CPA、IND-CCA、EUF-CMA 四個遊戲，解釋每一行規則存在的理由（$|m_0| = |m_1|$、$c \neq c_b$、$m' \notin \mathcal{Q}$、為何是 $\frac{1}{2}$ 與 0）。
5. 證明「確定性加密不可能 IND-CPA 安全」，並說明三個 IND 概念的強弱關係與分離例子。
6. 解釋 PPT 與 negligible 的定義、舉例，並說明漸近安全與具體安全的差別。
7. 完整解答 Problem Set 2 的全部 14 小題，包括用程式驗證 Caesar 密碼的暴力破解極限。

教材結構

- **第一部分（§1–§8）**：依講義順序精讀，每節標明對應的講義頁碼（講義右下角頁碼，共 24 頁），並補充 KL-IMC 與原始文獻的內容。
- **第二部分（§9–§11）**：Problem Set 2 逐題詳解。每題先完整引述原題，再給出可直接寫在作業或考卷上的「標準答案」，最後補充延伸討論。
- **第三部分**：一頁重點整理、中英術語對照表、自我測驗與參考資料。

記號慣例

記號	意義
$x \leftarrow X$	從集合 X 均勻隨機抽出 x；若 X 是隨機演算法，表示執行它並把輸出記為 x
$c \leftarrow \mathsf{Enc}(\mathsf{sk}, m)$ vs. $c = \mathsf{Enc}(\mathsf{sk}, m)$	箭頭表示 Enc 可能是隨機的；等號表示確定性
$\mathsf{Enc}(\mathsf{sk}, m; \rho)$	把隨機性 $\rho \in \mathcal{R}$ 明寫成輸入後，Enc 變成確定性函數
$E_{\mathsf{sk}}(\cdot)$、$D_{\mathsf{sk}}(\cdot)$	固定金鑰後的函數 $E_{\mathsf{sk}}\colon \mathcal{M} \to \mathcal{C}$、$D_{\mathsf{sk}}\colon \mathcal{C} \to \mathcal{M} \cup \{\perp\}$
α^n、α^*、ε	字母表 α 上長度 n 的字串、任意長度的字串（Kleene 星號）、空字串
$\lvert X\rvert$、$\lvert m\rvert$	有限集合的元素個數；字串的長度
$\mathcal{O}_{\mathsf{Enc}}$, $\mathcal{O}_{\mathsf{Dec}}$, $\mathcal{O}_{\mathsf{Tag}}$	加密 / 解密 / 標籤預言機（oracle）：敵手可呼叫但看不到內部的黑盒
IND-EAV 等	安全遊戲的名稱；「SKE 是 IND-CPA 安全的」指沒有 PPT 敵手能以明顯高於 $\frac{1}{2}$ 的機率獲勝
$\mathsf{negl}(n)$	可忽略函數：比任何多項式的倒數都小（§8）
(P1)–(P6)	Kerckhoffs 的第一至第六條原則（§5）

建議讀法

講義本身很精簡（許多頁只有一兩句話），真正的內容在「為什麼這樣定義」。建議讀每一節時先遮住解釋，自己回答每個「Why?」（為什麼 Dec 要確定？為什麼 $\frac{1}{2}$？為什麼 0？），再對照教材。第二部分的 Problem 2 附有可執行的 Python 程式，請務必自己跑一次。

第一部分

講義精讀與延伸

對應講義 pp. 2–23：SKE 與 MAC 的正式定義、空間與函數、Kerckhoffs 原則、IND 與 EUF-CMA 安全遊戲、PPT 與可忽略函數

1　SKE 的正式定義（講義 pp. 2–4）

第一週我們用「金鑰、加密、解密」三個直覺描述了秘密金鑰加密。本節把它寫成嚴格的定義，每個字都有用意。

1.1　語法（syntax）

秘密金鑰加密方案 SKE（講義 p. 2）

一個**秘密金鑰加密方案**（secret key encryption scheme）是三個演算法的三元組 $\mathsf{SKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$，並附帶三個集合：**金鑰空間**（key space）$\mathcal{K}$、**訊息空間**（message space）$\mathcal{M}$、**密文空間**（ciphertext space）$\mathcal{C}$。

- KGen 是**隨機化**演算法，抽樣並輸出秘密金鑰 $\mathsf{sk} \in \mathcal{K}$，記作 $\mathsf{sk} \leftarrow \mathsf{KGen}$。
- Enc 是**可能隨機化**的演算法，輸入 $\mathsf{sk} \in \mathcal{K}$ 與 $m \in \mathcal{M}$，輸出 $c \in \mathcal{C}$：

$$c \leftarrow \mathsf{Enc}(\mathsf{sk}, m)\text{（若隨機化）} \qquad \text{或} \qquad c = \mathsf{Enc}(\mathsf{sk}, m)\text{（若確定性）}.$$

- Dec 是**確定性**演算法，輸入 $\mathsf{sk} \in \mathcal{K}$ 與 $c \in \mathcal{C}$，輸出 $m \in \mathcal{M}$ 或特殊的**失敗符號**（failure symbol）$\perp$：

$$m = \mathsf{Dec}(\mathsf{sk}, c) \qquad \text{或} \qquad \perp = \mathsf{Dec}(\mathsf{sk}, c).$$

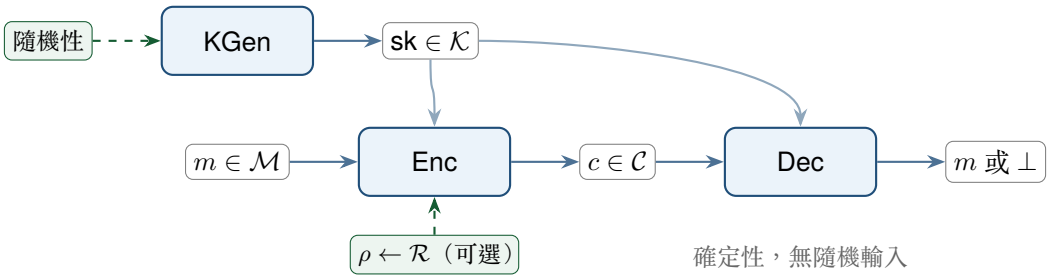


圖 1　SKE 的三個演算法與資料流。KGen 一定用到隨機性；Enc 可選；Dec 不用。

1.2　「隨機化」是什麼意思？（講義 p. 3）

講義問：一個函數是「隨機化的」是什麼意思？答案是：**它隱含地多拿了一個隨機的輸入。**

把隨機性寫出來：$\mathsf{Enc}(\mathsf{sk}, m; \rho)$

取一個**隨機性空間**（randomness space）$\mathcal{R} = \{0,1\}^r$，先抽 $\rho \leftarrow \mathcal{R}$，再計算

$$c = \mathsf{Enc}(\mathsf{sk}, m; \rho).$$

分號後面是隨機帶（random coins）。一旦把 ρ 也當成輸入，Enc 就是一個**完全確定的函數** $\mathcal{K} \times \mathcal{M} \times \mathcal{R} \to \mathcal{C}$；「隨機化演算法」只是「確定性演算法 + 由系統幫你抽好的 ρ」。實

作上 ρ 來自作業系統的亂數來源，例如 Python 的 `os.urandom`、Rust 的 `getrandom`、Linux 的 `/dev/urandom`。

這個觀點有三個用處：

1. **分析**：所有「機率」都可以明確地說成「對 ρ（以及 sk、敵手的隨機帶）取機率」。
2. **正確性的量化**：對隨機化加密，「正確」可以要求對每一個 ρ 都成立（完美正確），或以壓倒性的機率成立（KL-IMC 允許後者）。
3. **安全性**：§6 會證明**確定性加密不可能通過 IND-CPA**。講義 p. 3 的評語「歷史密碼的一個根本問題是 Enc 是確定性的」就是這個意思——同一個明文每次都加密成同一個密文，重複與模式會洩漏。

為什麼 Dec 是確定性的，而 Vf 也是？

收件人需要一個明確的答案：這個密文到底是什麼訊息？如果 Dec 擲硬幣，「正確性」就只能是機率性的，而且沒有任何好處——任何隨機化的 Dec 都可以把隨機帶固定成全零而不失正確性。同理，Vf 的輸出是「接受 / 拒絕」的判決，必須確定。反過來，KGen 必須隨機：若金鑰是確定地產生，人人都能算出同一把金鑰（這也是 Kerckhoffs 第二條原則的要求：演算法公開，秘密只在金鑰）。

1.3　正確性（correctness，講義 p. 4）

SKE 正確性

$\mathsf{SKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ 是**正確的**（correct），若

$$\forall\, \mathsf{sk} \in \mathcal{K},\ \forall\, m \in \mathcal{M}: \qquad \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{sk}, m)) = m.$$

若 Enc 是隨機化的，等式應對所有隨機性 ρ 成立：$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{sk}, m; \rho)) = m$。

注意量詞：是所有金鑰、所有訊息，不是「大部分」。這條性質講的是「用同一把 sk 加密再解密會回到原訊息」；它完全沒有提到敵手。

正確但愚蠢：正確性 ≠ 安全性（講義 p. 4）

$\mathsf{Enc}(\mathsf{sk}, m) = m$、$\mathsf{Dec}(\mathsf{sk}, c) = c$ 滿足正確性定義的每一個字，卻什麼都沒保護。第一週的「什麼都不做」方案在這裡再次出現，提醒我們：**語法與正確性只是入場券，安全性需要另外定義**（§6）。

1.4　為什麼需要 $\perp$？

正式定義允許 Dec 輸出 $\perp$。這不是裝飾，有三個理由：

- **不在像集裡的密文**：$E_{\mathsf{sk}}\colon \mathcal{M} \to \mathcal{C}$ 不一定是滿射（Problem 1 (vi)）。對一個從未被任何訊息加密出來的 $c \in \mathcal{C}$，Dec 必須說點什麼——$\perp$ 就是「這不是合法密文」。
- **格式錯誤**：長度不對、填充（padding）不合法、版本欄位不認得。
- **完整性失敗**：第一週的 Encrypt-then-MAC 中，Vf 失敗時「先驗證、後解密」的 Dec′ 就輸出 $\perp$。到了 §6 的 IND-CCA 遊戲，解密預言機對挑戰密文也回 $\perp$。

正確性只約束「合法產生」的密文：對 $c \leftarrow \mathsf{Enc}(\mathsf{sk}, m)$ 必須回 m；對其他 c，Dec 可以回 $\perp$，也可以回任何東西（但 §6 會看到「回任何東西」常常是安全漏洞的來源）。

考試重點

- 能一字不差地寫出 SKE 的語法與正確性，並標出哪個演算法「一定隨機」「可能隨機」「一定確定」。
- 能解釋 $\mathsf{Enc}(\mathsf{sk}, m; \rho)$ 記號：隨機化 = 確定性演算法 + 隨機輸入 $\rho \in \mathcal{R} = \{0,1\}^r$。
- 能說明 $\perp$ 的用途，以及為何正確性只對「由 Enc 產生的密文」有要求。

2　MAC 的正式定義（講義 pp. 5–7）

2.1　語法

訊息鑑別碼 MAC（講義 p. 5）

一個**訊息鑑別碼**（message authentication code）是三元組 $\mathsf{MAC} = (\mathsf{KGen}, \mathsf{Tag}, \mathsf{Vf})$，附帶金鑰空間 $\mathcal{K}$、訊息空間 $\mathcal{M}$ 與**標籤空間**（tag space）$\mathcal{T}$。

- KGen 是隨機化演算法，輸出**完整性金鑰**（integrity key）$\mathsf{ik} \in \mathcal{K}$：$\mathsf{ik} \leftarrow \mathsf{KGen}$。
- Tag 是可能隨機化的演算法，輸入 $\mathsf{ik} \in \mathcal{K}$、$m \in \mathcal{M}$，輸出標籤 $\tau \in \mathcal{T}$：$\tau \leftarrow \mathsf{Tag}(\mathsf{ik}, m)$ 或 $\tau = \mathsf{Tag}(\mathsf{ik}, m)$。
- Vf 是確定性演算法，輸入 $\mathsf{ik} \in \mathcal{K}$、$m \in \mathcal{M}$、$\tau \in \mathcal{T}$，輸出 0（拒絕）或 1（接受）。

與 SKE 對照：Enc $\leftrightarrow$ Tag 產生東西、Dec $\leftrightarrow$ Vf 檢查東西；差別在 Vf 不需要「還原」什麼，只需判決。注意 MAC 的輸出 τ 是附加在 m 旁邊傳送的，訊息本身不被隱藏。

2.2　確定性 Tag 與典範驗證（講義 p. 6）

本課程只考慮確定性的 Tag，即 $\tau = \mathsf{Tag}(\mathsf{ik}, m)$。此時可以「不失一般性」地假設 Vf 是典範驗證（canonical verification）：

典範驗證

$\mathsf{Vf}(\mathsf{ik}, m, \tau)$ 的做法是：計算 $\tau' = \mathsf{Tag}(\mathsf{ik}, m)$；若 $\tau = \tau'$ 輸出 1，否則輸出 0。

講義問：「典範驗證需要確定性的 Tag，為什麼？」

命題 2.1（典範驗證 ⇔ 確定性 Tag） 設 MAC 使用典範驗證。則 MAC 正確（對所有 ik, m，$\mathsf{Vf}(\mathsf{ik}, m, \mathsf{Tag}(\mathsf{ik}, m)) = 1$）若且唯若對每一組 (ik, m)，$\mathsf{Tag}(\mathsf{ik}, m)$ 只有一個可能的輸出值，亦即 Tag 是確定性的。

證明.（⇐）若 Tag 確定，驗證時重新計算得到同一個 $\tau' = \mathsf{Tag}(\mathsf{ik}, m) = \tau$，故輸出 1。

（⇒）反之，若存在 (ik, m) 使 $\mathsf{Tag}(\mathsf{ik}, m)$ 可能輸出兩個不同值 $\tau_1 \neq \tau_2$（各以正機率出現），則發送端可能產生 τ_1，驗證端重算卻可能得到 τ_2，此時 $\tau_1 \neq \tau_2$，Vf 輸出 0：合法標籤被拒絕，正確性以正機率失敗。 ■

為什麼說「不失一般性」？

若 Tag 確定且 MAC 正確，任何 Vf 都必須接受 $\tau = \mathsf{Tag}(\mathsf{ik}, m)$（這是正確性的要求）；典範的 Vf 則只接受這一個值。換言之，在所有正確的驗證演算法中，典範驗證接受的集合最小，因此它永遠不會比其他 Vf 更不安全。第一週 Problem 1 (vii) 已經證明：典範驗證的 MAC 一旦 EUF-CMA 安全，就自動是強不可偽造（SUF-CMA）的。

隨機化 Tag 在理論上存在（例如標籤附帶一個隨機 nonce，Vf 用 nonce 重算），但 HMAC、CMAC、Poly1305 等實務 MAC 都是確定性的，典範驗證就是它們實際的驗證方式——實作上要用常數時間比較（Week 1 §4.4），否則 $\tau = \tau'$ 的逐位元組比較會洩漏時間側通道。

2.3 正確性（講義 p. 7）

MAC 正確性

$\mathsf{MAC} = (\mathsf{KGen}, \mathsf{Tag}, \mathsf{Vf})$ 正確，若

$$\forall\, \mathsf{ik} \in \mathcal{K},\ \forall\, m \in \mathcal{M}: \qquad \mathsf{Vf}(\mathsf{ik}, m, \mathsf{Tag}(\mathsf{ik}, m)) = 1.$$

正確但愚蠢：Vf ≡ 1

永遠輸出 1 的 Vf 滿足正確性，但任何人都能「偽造」任何訊息。第一週 Problem 1 (v) 證明過：正是因為正確性保證「合法標籤一定被接受」，所以「對任何訊息都不可偽造」(D2) 是不可能達成的目標——我們只能要求新訊息不可偽造（§7）。

考試重點

- 寫出 MAC 的語法、正確性；知道本課程假設 Tag 確定、Vf 典範。
- 能用一句話回答「為什麼典範驗證需要確定性 Tag」：否則重算的 τ' 可能不等於合法的 τ，正確性失敗。

3 空間 $\mathcal{K}, \mathcal{M}, \mathcal{C}, \mathcal{T}$ 到底是什麼？（講義 pp. 8–9）

3.1　形式上只是集合

定義裡的「空間」在數學上就是集合（set）：一堆沒有重複的東西。貓的照片、位元字串、一顆超新星——什麼都可以（講義 p. 8）。這種抽象有好處（定理對任何集合都成立），但常常不夠具體。密碼學裡，這些集合幾乎總是**某個字母表上的字串**。

3.2　字母表、字串與 Kleene 星號

字母表與字串

一個**字母表**（alphabet）α 是一個有限非空集合，其元素稱為**符號**（symbol）。對 $n \in \mathbb{N} = \{0, 1, 2, \dots\}$：

- α^n 是所有長度為 n 的符號序列（字串）所成的集合；$|\alpha^n| = |\alpha|^n$。
- $\alpha^0 = \{\varepsilon\}$，其中 ε 是**空字串**（empty string）（長度 0 的字串，恰有一個）。
- **Kleene 星號**（Kleene star）：$\alpha^* = \bigcup_{i \in \mathbb{N}} \alpha^i$，所有有限長度字串的集合（包含 ε）。它是可數無限集合。

二元字母表 $\alpha = \{0, 1\}$（講義 p. 8）

$$\alpha^0 = \{\varepsilon\}, \quad \alpha^1 = \{0, 1\}, \quad \alpha^2 = \{00, 01, 10, 11\}, \quad \alpha^3 = \{000, \dots, 111\}, \quad \alpha^* = \bigcup_{i \in \mathbb{N}} \alpha^i.$$

α^1 的元素是一個**位元**（bit）；α^n 有 2^n 個元素，所以「n 位元金鑰」的金鑰空間大小是 2^n——這是為什麼 128 位元金鑰的暴力搜尋需要約 $2^{128} \approx 3.4 \times 10^{38}$ 次嘗試。

英文字母表 $\alpha = \{\mathrm{a}, \dots, \mathrm{z}\}$（講義 p. 9）

歷史密碼通常使用傳統意義下的字母表。常把字母編碼為數字 $\mathrm{a} = 0, \mathrm{b} = 1, \dots, \mathrm{z} = 25$，於是字母運算變成 $\mathbb{Z}_{26} = \{0, \dots, 25\}$ 上的模 26 算術——Problem 2 的 Caesar 密碼 $c_i = m_i + \mathsf{sk} \bmod 26$ 正是如此。$|\alpha^n| = 26^n$；$26! \approx 4.0 \times 10^{26} \approx 2^{88.4}$ 是「26 個字母的全部排列數」，下週單字母替換密碼會用到。

3.3　講義要你記住的兩個常見設定（講義 p. 9）

令 $\alpha_{\mathcal{K}}, \alpha_{\mathcal{M}}, \alpha_{\mathcal{C}}$ 分別是金鑰、訊息、密文空間的字母表。

設定	空間	典型例子
歷史密碼	$\alpha_{\mathcal{K}} = \alpha_{\mathcal{M}} = \alpha_{\mathcal{C}} = \{\mathrm{a}, \ldots, \mathrm{z}\}$；$\mathcal{K} = \alpha_{\mathcal{K}}^{n}$（固定長度金鑰），$\mathcal{M} = \alpha_{\mathcal{M}}^{*}$、$\mathcal{C} = \alpha_{\mathcal{C}}^{*}$（任意長度）	Caesar（$n = 1$，金鑰是一個位移量）、Vigenère（n 個字母的關鍵字）、單字母替換（金鑰是一個排列）
現代密碼	$\alpha_{\mathcal{K}} = \alpha_{\mathcal{M}} = \{0,1\}$，$\alpha_{\mathcal{C}} = \{0,1\}^{\ell}$；$\mathcal{K} = \{0,1\}^{n}$，$\mathcal{M} = \{0,1\}^{*}$，$\mathcal{C} = (\{0,1\}^{\ell})^{*}$	區塊加密模式：金鑰 $n = 128$ 或 256 位元，密文由 $\ell = 128$ 位元的區塊串成；AES-GCM、ChaCha20-Poly1305

兩個觀察：

- 金鑰空間 $\mathcal{K} = \alpha^{n}$ 是**固定長度、有限的**，大小 $|\alpha|^{n}$；訊息與密文空間則是 α^{*}，**無限**。因此「暴力搜尋金鑰」在原則上總是可行（有限），只是可能不可行（太大）——Problem 2 (ii) 討論何時連這都不夠。
- 現代設定裡密文字母表是 $\{0,1\}^{\ell}$ 而非 $\{0,1\}$，表示密文以「區塊」為單位；密文長度是區塊長度的倍數，這會透露訊息長度的粗略資訊——這就是為什麼 IND 遊戲要求 $|m_0| = |m_1|$（§6）。

數一數：這些集合有多大？

- $|\{0,1\}^{128}| = 2^{128}$；$|\{\mathrm{a..z}\}^{5}| = 26^{5} = 11{,}881{,}376$。
- $|\alpha^{\leq n}| = \sum_{i=0}^{n} |\alpha|^{i} = \dfrac{|\alpha|^{n+1} - 1}{|\alpha| - 1}$（長度不超過 n 的字串數）。
- α^{*} 無限但可數（把字串依長度、再依字典序排成一列）。

4　把 Enc 與 Dec 當成函數（講義 pp. 10–11）

固定一把金鑰 sk 之後，加密就是一個普通的數學函數。這一節複習函數的三個性質，並算一算「有多少個這樣的函數」——這是 Problem 1 的全部數學工具。

4.1　單射、滿射、雙射

函數與其三種性質（Problem Set 2 前言）

一個**函數**（function）$f\colon X \to Y$ 把每一個 $x \in X$ 對應到恰好一個 $y \in Y$，記 $f(x) = y$。$f(X) = \{f(x) : x \in X\} \subseteq Y$ 稱為 f 的**像**（image）。

- f 是**單射**（injective, one-to-one）：不同的輸入對應不同的輸出，即 $f(x) = f(x') \Rightarrow x = x'$（等價地 $x \neq x' \Rightarrow f(x) \neq f(x')$）。
- f 是**滿射**（surjective, onto）：每個 $y \in Y$ 至少被一個 x 打到，即 $f(X) = Y$。
- f 是**雙射**（bijective）：既單射又滿射；此時每個 y 恰有一個原像，反函數 $f^{-1}\colon Y \to X$

存在。

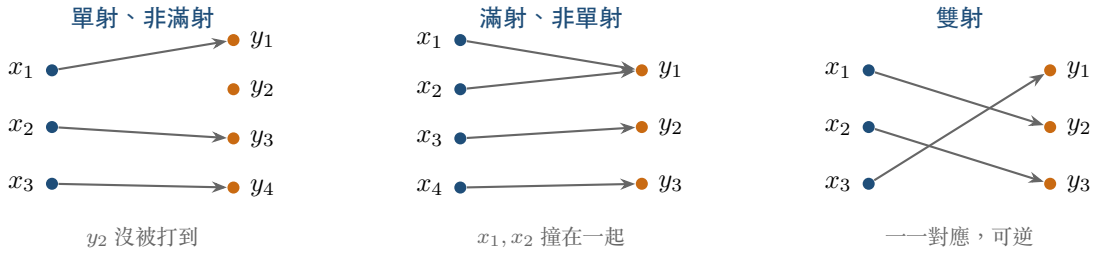


圖 2　三種函數。單射要「不撞」、滿射要「全打到」，雙射兩者皆是。

一句話記憶

單射看輸入：不同輸入、不同輸出（箭頭不會合流）。滿射看輸出：每個輸出都有人指著。Problem 1 (ii) 會證明：單射 $\Rightarrow |Y| \geq |X|$，滿射 $\Rightarrow |X| \geq |Y|$，雙射 $\Rightarrow |X| = |Y|$（鴿籠原理）。

4.2　E_{sk} 與 D_{sk}（講義 p. 10）

令 $\mathcal{M} = \{m_1, \ldots, m_s\}$、$\mathcal{C} = \{c_1, \ldots, c_t\}$，$t \geq s$，並固定 $\mathsf{sk} \in \mathcal{K}$。寫 $E_{\mathsf{sk}}(\cdot) = \mathsf{Enc}(\mathsf{sk}, \cdot)$、$D_{\mathsf{sk}}(\cdot) = \mathsf{Dec}(\mathsf{sk}, \cdot)$。依定義

$$E_{\mathsf{sk}} \colon \mathcal{M} \to \mathcal{C}, \qquad D_{\mathsf{sk}} \colon \mathcal{C} \to \mathcal{M} \cup \{\perp\}.$$

講義註腳預告：Problem Set 2 會證明 E_{sk} **必須是單射**，因此 $t \geq s$ 是必要的（Problem 1 (v)）。此時正確性 $D_{\mathsf{sk}}(E_{\mathsf{sk}}(m)) = m$ 說的是：D_{sk} 是 E_{sk} 的**左反函數**——$D_{\mathsf{sk}} \circ E_{\mathsf{sk}} = \mathrm{id}_{\mathcal{M}}$。在像 $E_{\mathsf{sk}}(\mathcal{M})$ 之外，D_{sk} 不受約束（可輸出 $\perp$）。

4.3　有多少個函數？（講義 p. 11）

命題 4.1（函數的計數）　設 $|X| = s$、$|Y| = t$。

(a) $X \to Y$ 的函數共有 t^s 個。

(b) 若 $t \geq s$，$X \to Y$ 的單射共有 $t(t-1)\cdots(t-s+1) = \dfrac{t!}{(t-s)!} = \dbinom{t}{s} \cdot s!$ 個；若 $t < s$ 則一個也沒有。

(c) 若 $t = s$，單射即雙射，共有 $s!$ 個。

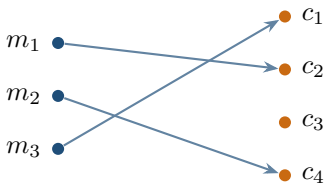
證明. (a) 依序決定 $f(x_1), \ldots, f(x_s)$，每個有 t 種選擇且互不影響，乘法原理給出 t^s。(b) 單射要求選過的值不能再選：$f(x_1)$ 有 t 種，$f(x_2)$ 有 $t-1$ 種，$\cdots$，$f(x_s)$ 有 $t-s+1$ 種。另一種算法：先選像集（s 個不同的 y，$\binom{t}{s}$ 種），再決定 X 到像集的雙射（$s!$ 種）。若 $t < s$，鴿籠原理說必有兩個輸入撞到同一輸出。(c) 在 (b) 中代入 $t = s$ 得 $\binom{s}{s} \cdot s! = s!$。 ■

講義的圖：$s=3$、$t=4$

$\mathcal{M}=\{m_1,m_2,m_3\}$、$\mathcal{C}=\{c_1,c_2,c_3,c_4\}$：

- 函數個數 $t^s=4^3=64$；
- 單射個數 $\binom{4}{3}\cdot 3!=4\cdot 6=24$；
- 若去掉 c_4 使 $t=s=3$，單射就是雙射，共 $3!=6$ 個。

所以在 64 個「可能的 E_{sk}」中，只有 24 個能讓解密無歧義；其餘 40 個都有兩個訊息撞到同一密文。



其中一個單射 E_{sk}：$m_1\mapsto c_2$，$m_2\mapsto c_4$，$m_3\mapsto c_1$。
D_{sk}：$c_2\mapsto m_1$，$c_4\mapsto m_2$，$c_1\mapsto m_3$，$c_3\mapsto\perp$。
像 $E_{\mathsf{sk}}(\mathcal{M})=\{c_1,c_2,c_4\}$；$c_3$ 永遠不會是合法密文。

圖 3　講義 p. 11 的設定（$s=3,t=4$）之一例。

4.4　這個計數在密碼學裡的意義

金鑰 = 挑選一個函數

一把金鑰 sk 就是從「所有單射 $\mathcal{M}\to\mathcal{C}$」這個巨大集合中挑出一個 E_{sk}。從敵手的角度，知道「系統」（Kerckhoffs !）等於知道這個集合，不知道的只是「哪一個」。

- 單字母替換密碼（下週）：$\mathcal{M}=\mathcal{C}=\{\text{a..z}\}$ 逐字母加密，E_{sk} 是 26 個字母的一個排列，$|\mathcal{K}|=26!\approx 2^{88.4}$。暴力搜尋不可行——但它仍然會被頻率分析攻破。**金鑰空間大只是必要條件，不是充分條件**（Kerckhoffs (P1) 說的是「對所有攻擊」不可破譯）。
- 現代區塊加密（例如 AES-128）：$\mathcal{M}=\mathcal{C}=\{0,1\}^{128}$，每把金鑰決定一個 $\{0,1\}^{128}$ 上的排列。所有排列有 $(2^{128})!$ 個，而金鑰只有 2^{128} 把——真實密碼只「覆蓋」了極小一部分排列。安全性的目標是：敵手分不出「AES 的某把金鑰」與「隨機挑的排列」（**偽隨機排列**（pseudorandom permutation），之後幾週的主題）。

考試重點

- 三個定義要能用符號寫、也要能用「箭頭圖」判斷。
- 計數公式 t^s、$\binom{t}{s}s!=\frac{t!}{(t-s)!}$、$s!$ 及其推導（乘法原理、鴿籠）。
- 知道 D_{sk} 是 E_{sk} 的左反函數，且只在像集上被正確性約束。

5　Kerckhoffs 原則（講義 p. 12）

🏛 Auguste Kerckhoffs 與《La cryptographie militaire》

Auguste Kerckhoffs（1835–1903）是出生於荷蘭林堡的語言學家，在巴黎高等商業學院任教，精通多種語言（也是人造語言 Volapük 的推廣者）。1883 年 1、2 月他在《Journal des sciences militaires》分兩期發表〈La cryptographie militaire〉（軍事密碼學），檢討當時法軍（普法戰爭 1870–71 戰敗後）的密碼實務。他在文中區分兩種情境：「少數孤立者之間暫時交換信件的密寫系統」與「要長期規範各軍團指揮官之間通訊的密碼方法」。後者才需要滿足下面六個「desiderata」（期望條件）。原文掃描版見 Fabien Petitcolas 的網站 `https://petitcolas.net/kerckhoffs/`。

5.1 六條原則：原文、英譯與中譯

#	法文原文（1883）	英譯（Petitcolas；講義採用）	中譯與關鍵詞
(P1)	Le système doit être matériellement, sinon mathématiquement, indéchiffrable ;	The system must be substantially, if not mathematically, indecipherable;	系統必須在實務上（即使不是數學上）不可破譯。→ 計算安全 vs. 資訊論安全
(P2)	Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi ;	The system must not require secrecy and can be stolen by the enemy without causing trouble;	系統不得依賴保密；落入敵手亦無妨。→「Kerckhoffs 原則」本體
(P3)	La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants ;	It must be easy to communicate and retain the key without the aid of written notes, and easy to change or modify the key at the discretion of the correspondents;	金鑰要能不靠紙本傳遞與記憶，且可隨時更換。→ 金鑰短、可更換、金鑰管理
(P4)	Il faut qu'il soit applicable à la correspondance télégraphique ;	The system ought to be compatible with telegraph communication;	須相容於電報通訊。→ 相容於通訊媒介 / 協定
(P5)	Il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes ;	The system must be portable, and its use must not require more than one person;	須可攜，一人即可操作。→ 可實作於受限裝置、可自動化
(P6)	Enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer.	Finally, given the circumstances in which the system is applied, it must be easy to use and must neither stress the mind or require the knowledge of a long series of rules.	須易用，不費心神、不需記一長串規則。→ 可用性、不易誤用（misuse resistance）

5.2 「Kerckhoffs 原則」= 第二條

今天說「Kerckhoffs 原則」（單數）幾乎都指 (P2)：**系統的安全只能依賴金鑰的保密，不能依賴演算法的保密**。Shannon 在 1949 年〈Communication Theory of Secrecy Systems〉把它濃縮成一句 **Shannon 格言**（Shannon's maxim）：「*the enemy knows the system*」——假設敵人知道系統。

KL-IMC §1.3 給的三個理由

1. **短的金鑰比演算法容易保密。**演算法要寫進每一台裝置、每一份程式碼、每一位工程師的腦袋裡；金鑰只是 128 位元，可以放在硬體安全模組裡。
2. **洩漏後，換金鑰比換演算法容易得多。**金鑰外洩就重新產生一把（(P3)）；演算法外洩等於整個系統、所有使用者都要重新設計與部署。
3. **公開才能標準化與受公眾檢驗。**公開的演算法可以被全世界的密碼學家攻擊、改進、互通；秘密演算法只被少數人看過，錯誤往往要到被攻破時才被發現。

Kerckhoffs 原則沒有說什麼

- 它沒有要求演算法一定要公開發表——它要求的是「即使公開也安全」。Petitcolas 的註解：方法不必公開，但「設計時必須被當成公開的」。把演算法保密作為額外的一層（defence in depth）並不違反原則；依賴它才違反。
- 它沒有說金鑰可以公開。恰恰相反：全部的秘密都集中到金鑰上，所以 (P3) 的金鑰管理變得至關重要。
- 「安全靠隱匿」（security through obscurity）是它的反面：把演算法藏起來並以此為安全依據。Problem 3 (iv) 會看到一連串的災難。

5.3 從 1883 到今天

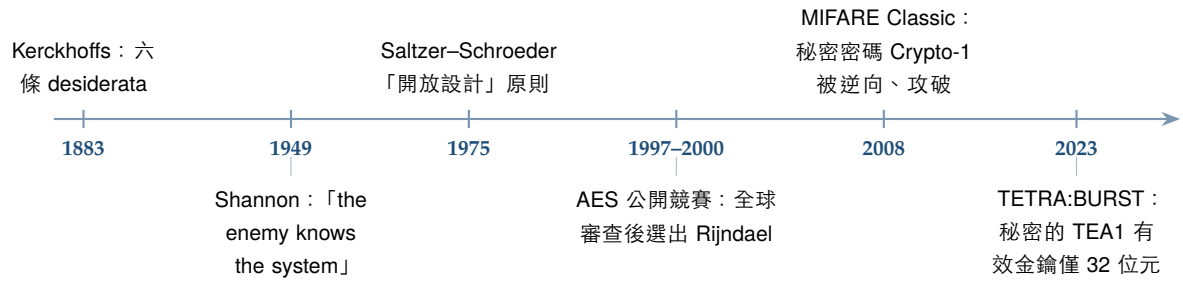


圖 4 Kerckhoffs 第二條原則的接受史與「安全靠隱匿」的反面教材。

現代密碼學把 (P2) 做到了極致：AES（1997–2000）、SHA-3（2007–2012）、後量子標準 ML-KEM / ML-DSA（2016–2024）都由 NIST 公開競賽選出，候選演算法在數年間被全世界公開攻擊；TLS、Signal 等協定的規格與程式碼都公開。安全定義本身（§6）也內建了 Kerckhoffs：遊戲裡敵手知道 $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$，不知道的只有 sk。

其他五條過時了嗎？

(P4) 的「電報」當然過時了，但把它讀成「相容於當代通訊媒介」就不過時；(P5)、(P6) 是今日「可用性安全」（usable security）與「不易誤用的 API」的祖先；(P3) 是金鑰管理學；(P1) 則演變成了本週後半的計算安全定義。Problem 3 (iii) 會請你逐條重寫。

6　SKE 的形式化安全：IND-EAV、IND-CPA、IND-CCA（講義 pp. 13–18）

6.1　直覺與困難（講義 p. 13）

直覺：不知道 sk 的敵手，不應該能「學到」任何關於訊息的「有意義」資訊，即使它能

(i) 觀察密文（被動竊聽，eavesdropping）；

(ii) 選擇訊息讓對方加密、並看到對應密文（選擇明文，chosen plaintext）；

(iii) 選擇訊息讓對方加密，並且選擇（某些）密文讓對方解密、看到結果（選擇密文，chosen ciphertext）。

問題是「學到」與「有意義」都不是數學。Week 1 的教訓是：安全定義必須同時說清（∗）敵手不能做什麼與（†）敵手能做什麼。本週的工具是**密碼學遊戲**（cryptographic game）：一個**挑戰者**（challenger）$\mathcal{Ch}$ 依規則與**敵手**（adversary）$\mathcal{A}$ 互動，最後判定 $\mathcal{A}$ 是否「獲勝」。(†) 由 $\mathcal{A}$ 可用的**預言機**（oracle）決定，(∗) 由獲勝條件決定。

Kerckhoffs 內建於遊戲

在每個遊戲裡，演算法 (KGen, Enc, Dec) 都是**公開的**（$\mathcal{A}$ 知道、甚至可以自己執行），只有 sk 是秘密。這就是 (P2)。

6.2　IND-EAV：對竊聽者的不可區分性（講義 p. 14）

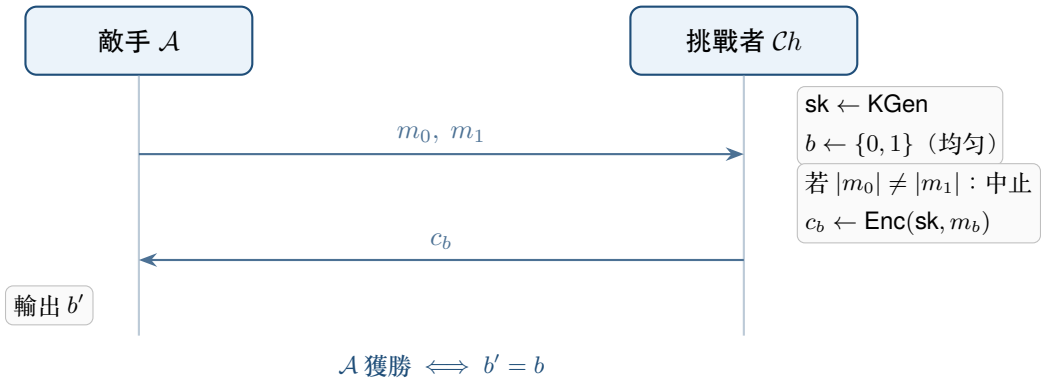


圖 5　IND-EAV 遊戲（INDistinguishability against EAVesdroppers；KL-IMC §3.2.1）。

IND-EAV 安全

SKE 是 IND-EAV 安全的，若對每一個有效率的（PPT）敵手 $\mathcal{A}$，

$$\Pr\left[\mathcal{A} \text{ 在 IND-EAV 遊戲中獲勝}\right] \leq \tfrac{1}{2} + \mathsf{negl},$$

其中機率對 sk、b、Enc 與 $\mathcal{A}$ 的隨機性取。等價地，優勢（advantage）$\mathsf{Adv}^{\text{ind-eav}}_{\mathsf{SKE}}(\mathcal{A}) = \left|\Pr[b' = b] - \tfrac{1}{2}\right|$ 是可忽略的。

讀法：$\mathcal{A}$ 自己挑兩個等長訊息（它認為最容易被分辨的兩個），挑戰者隨機加密其中一個，$\mathcal{A}$ 只看到一個密文，要猜是哪一個。如果連 $\mathcal{A}$ 自選的最有利情況都猜不準，它對任何訊息都學不到任何事。

6.3 IND-CPA：選擇明文攻擊（講義 p. 15）

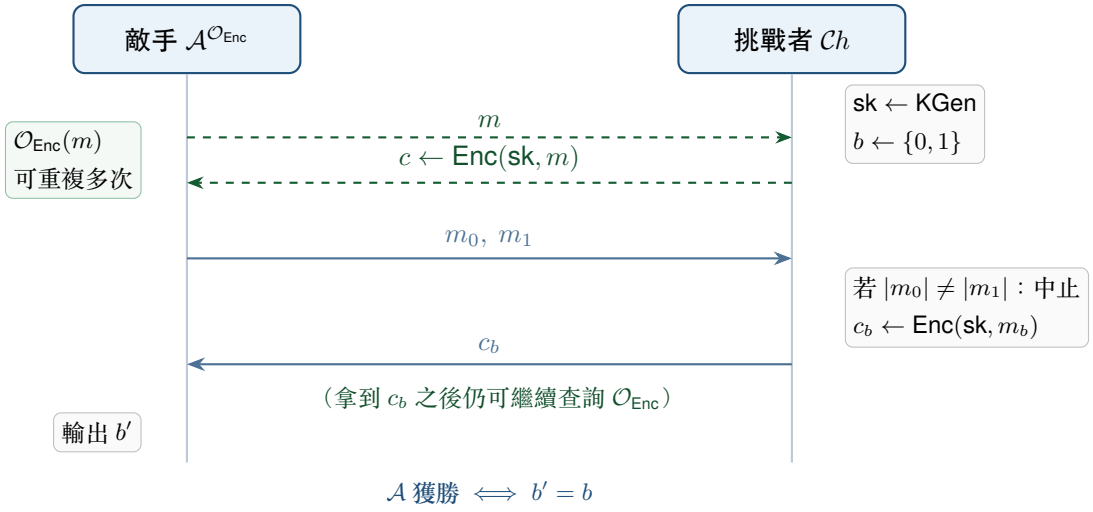


圖 6　IND-CPA 遊戲（KL-IMC §3.4.2）。綠色虛線是加密預言機 $\mathcal{O}_{\mathsf{Enc}}$，挑戰前後皆可呼叫、次數任意（多項式）。

IND-CPA 安全

SKE 是 IND-CPA 安全的，若對每一個 PPT 敵手 $\mathcal{A}$（可存取 $\mathcal{O}_{\mathsf{Enc}}(\cdot) = \mathsf{Enc}(\mathsf{sk}, \cdot)$），$\Pr[\mathcal{A} \text{ 獲勝}] \leq \tfrac{1}{2} + \mathsf{negl}$。

為什麼要給敵手加密預言機？因為現實中敵手常能讓你加密它選的內容：Week 1 的中途島故事（誘使日軍發出含「AF」的電報）、網頁瀏覽器替攻擊者的 JavaScript 加密含有 cookie 的請求（BEAST、CRIME 攻擊的前提）、加密資料庫替使用者上傳的資料加密。IND-CPA 說：即使敵手能看到任意多個「自選明文」的密文，仍然分不出挑戰密文。

定理 6.1（確定性（無狀態）加密不可能 IND-CPA 安全）　若 Enc 是確定性的（且無內部狀態），則存在一個只做一次預言機查詢、必勝的 IND-CPA 敵手。

證明. $\mathcal{A}$ 任選兩個等長且相異的訊息 $m_0 \neq m_1$。先查詢 $\mathcal{O}_{\mathsf{Enc}}(m_0)$，得 $c^* = \mathsf{Enc}(\mathsf{sk}, m_0)$。再送

出挑戰 (m_0, m_1)，收到 c_b。若 $c_b = c^*$ 輸出 $b' = 0$，否則輸出 $b' = 1$。因為 Enc 確定，$b = 0$ 時 $c_b = \mathsf{Enc}(\mathsf{sk}, m_0) = c^*$，必輸出 0；$b = 1$ 時 $c_b = \mathsf{Enc}(\mathsf{sk}, m_1)$，而由正確性 E_{sk} 是單射（Problem 1 (v)），$\mathsf{Enc}(\mathsf{sk}, m_1) \neq \mathsf{Enc}(\mathsf{sk}, m_0) = c^*$，必輸出 1。故 $\Pr[b' = b] = 1$。 ■

推論 6.2 Caesar、Vigenère、單字母替換、Enigma……所有確定性的歷史密碼，以及任何「直接用區塊加密逐塊加密」的 ECB 模式，都不是 IND-CPA 安全的。要通過 IND-CPA，Enc 必須隨機化（$\rho \leftarrow \mathcal{R}$）或有狀態（例如維護一個不重複的計數器 nonce）。

事實上許多歷史密碼連更弱的 IND-EAV 都過不了：

Caesar 密碼不是 IND-EAV 安全的

$\mathcal{A}$ 送出 $m_0 = \mathrm{aa}$、$m_1 = \mathrm{ab}$。收到 $c_b = c_1 c_2$ 後，若 $c_1 = c_2$ 輸出 0，否則輸出 1。Caesar 對每個字母加同一個位移，aa 的兩個密文字母相同、ab 的不同，故 $\mathcal{A}$ 必勝。$\mathcal{A}$ 甚至不需要知道金鑰——這正是「不可區分」的威力：**不用恢復金鑰、不用恢復訊息，只要洩漏任何一點點可分辨的東西就算輸**。

6.4 IND-CCA：選擇密文攻擊（講義 p. 16）

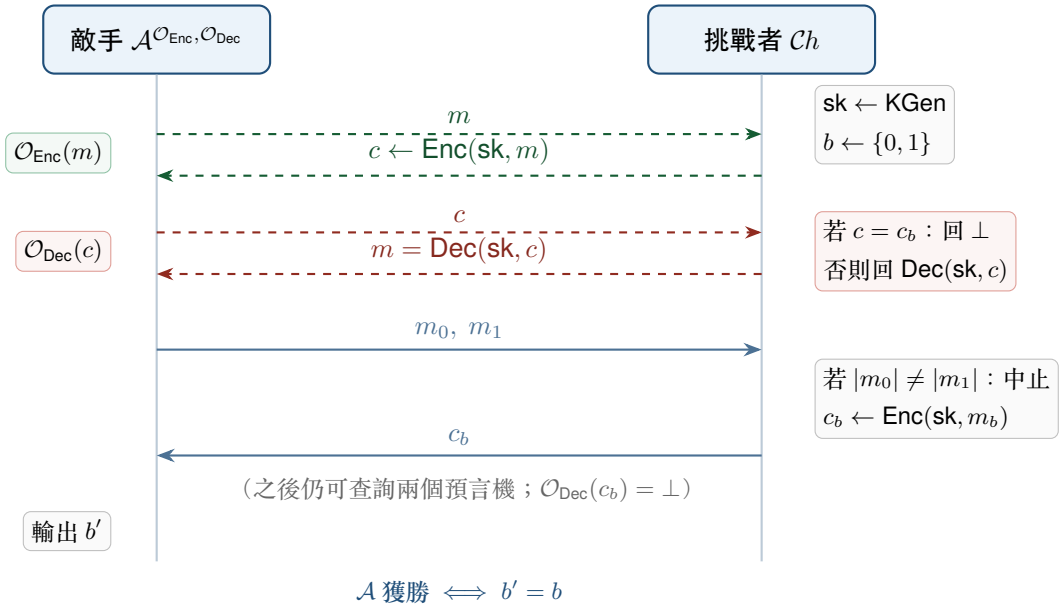


圖 7 IND-CCA 遊戲（KL-IMC §5.1.2）。紅色虛線是解密預言機 $\mathcal{O}_{\mathsf{Dec}}$；唯一的限制是不能拿挑戰密文 c_b 本身去解密。

IND-CCA 安全

SKE 是 IND-CCA 安全的，若對每一個 PPT 敵手 $\mathcal{A}$（可存取 $\mathcal{O}_{\mathsf{Enc}}$ 與受限的 $\mathcal{O}_{\mathsf{Dec}}$），$\Pr[\mathcal{A}\text{ 獲勝}] \leq \frac{1}{2} + \mathsf{negl}$。

為什麼限制 $\mathcal{O}_{\mathsf{Dec}}(c_b) = \perp$？

不限制的話遊戲就無聊了：$\mathcal{A}$ 把 c_b 丟給 $\mathcal{O}_{\mathsf{Dec}}$ 拿回 m_b，必勝。這與 Week 1 EUF-CMA 的「$m' \notin \mathcal{Q}$」是同一種「排除平凡勝利」的規則：定義要排除所有我們在意的攻擊，又不能把不可能避免的事算成攻擊。注意只有 c_b 本身被禁止——$\mathcal{A}$ 可以把 c_b 改一個位元再送去解密。因此 IND-CCA 安全蘊含**不可延展性**（non-malleability）：改動密文必須導致 $\perp$ 或與 m_b 無關的結果。

CCA 捕捉什麼現實？ Week 1 提過的 padding oracle（Vaudenay 2002）與 Bleichenbacher (1998) 攻擊：伺服器對「格式錯」與「格式對」的密文回應不同（錯誤訊息、時間），這就是一個（部分的）解密預言機；只要洩漏一個位元的解密資訊，就可能逐步解出整個訊息。

IND-CPA 安全但不是 IND-CCA 安全：可延展的串流加密

考慮「隨機 nonce + 金鑰串流」的加密（CTR 模式或 ChaCha20 不帶 MAC）：$c = (\nu,\ m \oplus G(\mathsf{sk}, \nu))$，$\nu$ 每次新抽。它是 IND-CPA 安全的（之後幾週會證明）。但 $\mathcal{A}$ 在 CCA 遊戲中可以：收到 $c_b = (\nu, z)$ 後，送 $c' = (\nu,\ z \oplus 10\cdots0) \neq c_b$ 給 $\mathcal{O}_{\mathsf{Dec}}$，拿回 $m_b \oplus 10\cdots0$，翻回第一個位元就得到 m_b，必勝。這就是 Week 1「一次性密碼本可延展」的遊戲版。補救：Encrypt-then-MAC（Week 1 Problem 1 (iii)）——Bellare–Namprempre 證明 IND-CPA 的 Enc 加上 SUF-CMA 的 MAC 給出 IND-CCA。

6.5 為什麼是「不可區分」？（講義 p. 17）

講義的例子：設 $\mathcal{M} = \{0,1\}^*$，而 $\mathcal{A}$ 能從 $c \leftarrow \mathsf{Enc}(\mathsf{sk}, m)$ 判斷 m 的**漢明重量**（Hamming weight）（1 的個數）。它如何贏 IND-EAV？送出 $m_0 = 0$、$m_1 = 1$：兩者等長、漢明重量分別為 0 與 1，看 c_b 判斷重量就知道 b。

IND ⇒ 什麼都學不到

把例子一般化：假設 $\mathcal{A}$ 能從密文計算任何關於訊息的函數 $f(m)$（第一個位元、是否含某字、長度以外的任何統計量……），只要存在兩個等長訊息 $f(m_0) \neq f(m_1)$，$\mathcal{A}$ 就把它們當挑戰，用 f 分辨 b。逆否命題：**IND 安全 ⇒ 沒有任何（可有效計算的）訊息函數能從密文學到**。KL-IMC 定理 3.13 把這件事嚴格化：IND-EAV 等價於語意安全（semantic security）——「有密文」與「沒有密文」時，敵手對 $f(m)$ 的猜測能力相同。若 $\mathcal{A}$ 需要加密預言機才能算出 f（例如要先收集統計資料），同樣的論證給出 IND-CPA；需要解密預言機則給出 IND-CCA（講義 p. 17 最後一行）。

為什麼讓 $\mathcal{A}$ 自己挑 m_0, m_1？因為我們要的是**最壞情況保證**：對每一對訊息都分不出來，自然對 $\mathcal{A}$ 自選的最有利一對也分不出來，反之亦然。

6.6 為什麼是二分之一？（講義 p. 18）

IND-{EAV, CPA, CCA} 安全（講義版）

SKE 是 IND-{EAV, CPA, CCA} 安全的，若沒有有效率的敵手 $\mathcal{A}$ 能以明顯異於二分之一的機率贏得對應的遊戲。

為什麼是 $\frac{1}{2}$？因為任何人都可以**直接猜** b（擲硬幣），獲勝機率恰為 $\frac{1}{2}$；這是「什麼都沒學到」的基準線。注意兩件事：

- 「明顯異於」是雙向的：若某 $\mathcal{A}$ 以 $\frac{1}{2}-0.3$ 的機率獲勝，把它的輸出 b' 反過來就以 $\frac{1}{2}+0.3$ 獲勝。所以優勢定義成 $\left|\Pr[b'=b]-\frac{1}{2}\right|$。
- 「明顯」的精確意義是「不可忽略」（non-negligible），「有效率」是 PPT——§8 說明。

6.7 三個概念的強弱關係

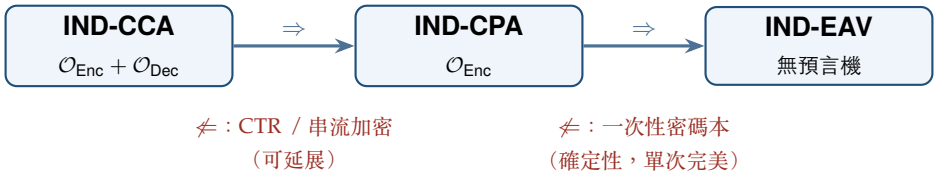


圖 8　給敵手愈多預言機，定義愈強。箭頭方向是「蘊含」；反向都不成立（分離例子）。

- **蘊含**：任何 IND-EAV 敵手也是一個（不用預言機的）IND-CPA 敵手，任何 IND-CPA 敵手也是一個（不用 $\mathcal{O}_{\mathsf{Dec}}$ 的）IND-CCA 敵手。因此 CCA 安全 ⇒ CPA 安全 ⇒ EAV 安全。
- **分離**：一次性密碼本（金鑰只用一次）是完美的 IND-EAV（任何敵手恰好 $\frac{1}{2}$），卻因確定性而不是 IND-CPA（定理 6.1）；隨機化串流加密是 IND-CPA 卻不是 IND-CCA（上面的可延展例子）。

遊戲	預言機（†）	獲勝條件（∗）	現實對應
IND-EAV	無	猜中 b；基準 $\frac{1}{2}$	被動竊聽一則訊息
IND-CPA	$\mathcal{O}_{\mathsf{Enc}}$（挑戰前後）	同上	敵手能誘使你加密它選的內容
IND-CCA	$\mathcal{O}_{\mathsf{Enc}}$、$\mathcal{O}_{\mathsf{Dec}}$ $(c\neq c_b)$	同上	padding oracle、錯誤訊息、任何解密結果的回饋
EUF-CMA	$\mathcal{O}_{\mathsf{Tag}}$	偽造新訊息的合法標籤；基準 0	竄改 / 偽造訊息（§7）

6.8 為什麼要求 $|m_0|=|m_1|$？

因為**加密無法隱藏長度**（在 $\mathcal{M}=\alpha^*$ 的一般設定下）：一則 1 KB 與一則 1 GB 的訊息，密文長度必然不同，否則所有訊息都得被填充到最大長度——而 α^* 沒有最大長度。所以定義誠實地把長度排除在保護範圍外。這是一個真實的洩漏來源：

- 流量分析：加密的語音封包長度洩漏說話內容的音節結構；加密網頁的大小洩漏你在看哪一頁。
- CRIME（2012）與 BREACH（2013）：TLS / HTTP 在加密前先壓縮，攻擊者注入自選字串，觀察密文長度是否變短，逐字猜出 cookie——完全只用「長度」就攻破了 IND-CPA 安全的加密。對策是不要在加密前壓縮秘密與攻擊者可控資料的混合物。

若系統需要隱藏長度，必須在協定層做填充（padding）到固定長度桶，這是加密定義之外的事。

考試重點

- 能畫出三個 IND 遊戲，並說明每一行：$\mathsf{sk} \leftarrow \mathsf{KGen}$、$b \leftarrow \{0,1\}$、$|m_0| = |m_1|$、$c_b$、$\mathcal{O}_{\mathsf{Dec}}(c_b) = \perp$、輸出 b'、獲勝條件。
- 能寫出安全定義的句型：「沒有 PPT 敵手能以明顯異於 $\frac{1}{2}$ 的機率獲勝」，並解釋 $\frac{1}{2}$ 的來源。
- 能證明確定性加密不是 IND-CPA；能給出 Caesar 不是 IND-EAV 的敵手。
- 能說明 CCA $\Rightarrow$ CPA $\Rightarrow$ EAV 與兩個分離例子；能解釋漢明重量例子如何說明「IND 捕捉一切洩漏」。

7 MAC 的形式化安全：EUF-CMA（講義 pp. 19–22）

7.1 直覺（講義 p. 19）

不知道 ik 的敵手，不應該能替一個它沒看過標籤的訊息偽造出合法標籤，即使它可以要求對任意自選的訊息產生標籤。講義註明：像 IND-EAV / CPA / CCA 一樣，MAC 也有不同強度的定義（KL-IMC §4.2「Strong unforgeability」），本課程只用其中一個。Week 1 的 Problem 1 其實已經把整個光譜走過一遍——(D1) 金鑰不可恢復（太弱）、(D2) 任何訊息不可偽造（不可能）、(D3) 新訊息不可偽造（標準）、$(D3)^+$ 強不可偽造（典範驗證下自動成立）。本節的 EUF-CMA 就是 (D3)。

7.2 EUF-CMA 遊戲（講義 p. 20）

EUF-CMA 安全（講義 p. 22）

$\mathsf{MAC} = (\mathsf{KGen}, \mathsf{Tag}, \mathsf{Vf})$ 是 EUF-CMA 安全的，若沒有有效率的敵手 $\mathcal{A}$ 能以明顯異於零的機率贏得 EUF-CMA 遊戲；即對每個 PPT $\mathcal{A}$，$\Pr[\mathcal{A} \text{ 獲勝}] \leq \mathsf{negl}$。

逐字解讀名字：**Existential**（存在性）——$\mathcal{A}$ 只要對某個（它自己挑的、哪怕毫無意義的）新訊息偽造成功就算贏，不必是特定的有意義訊息；**Unforgeability**——不可偽造；**Chosen Message Attack**——$\mathcal{A}$ 可以選訊息要標籤（$\mathcal{O}_{\mathsf{Tag}}$），而且是適應性的（看到前面的回答再決定下一個查詢）。

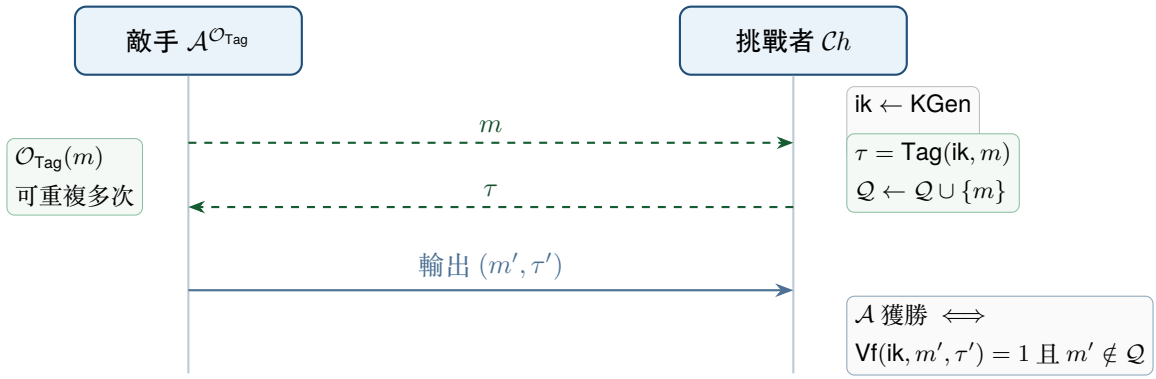


圖 9 EUF-CMA 遊戲（Existential UnForgeability against Chosen Message Attack；KL-IMC §4.2）。$\mathcal{Q}$ 記錄所有查詢過的訊息。

7.3 為什麼是「存在性不可偽造」？（講義 p. 21）

講義問：為什麼 EUF 捕捉了「不能替新訊息偽造標籤」？假設 $\mathcal{A}$ 看過 $(m_1, \tau_1), \ldots, (m_n, \tau_n)$ 之後能偽造 $m' \notin \{m_1, \ldots, m_n\}$ 的標籤 τ'。它如何贏 EUF-CMA？——把 $m_1, \ldots, m_n$ 依序送給 $\mathcal{O}_{\mathsf{Tag}}$ 拿到 $\tau_1, \ldots, \tau_n$，然後輸出 (m', τ')。$\mathcal{Q} = \{m_1, \ldots, m_n\} \not\ni m'$，且 $\mathsf{Vf}(\mathsf{ik}, m', \tau') = 1$，獲勝。

所以：**任何**「看過一些合法訊息–標籤對後能偽造新訊息」的攻擊，都對應一個 EUF-CMA 敵手；EUF-CMA 安全就排除了所有這類攻擊。而且因為是「存在性」，連「偽造一個亂碼訊息的標籤」都不允許——這看似過嚴，實則必要：我們不知道上層應用會把哪些位元串當成有意義的訊息（Week 1 的教訓：定義要夠強、對應用不做假設）。

7.4 為什麼是零？（講義 p. 22）

IND 遊戲的基準是 $\frac{1}{2}$，因為敵手是在**兩個可能性中分辨**，隨便猜就對一半。EUF-CMA 要敵手**創造**一個東西（合法的偽造），沒有「亂猜就對一半」的情況，所以基準是 0。

零不是真的零：猜標籤的機率

敵手總可以隨便挑一個新訊息 m' 與一個隨機的 $\tau' \in \mathcal{T}$，獲勝機率是 $\Pr[\mathsf{Tag}(\mathsf{ik}, m') = \tau']$——對一個好的 MAC 約為 $1/|\mathcal{T}|$。若標籤是 128 位元，這是 2^{-128}，可忽略；若為了省頻寬把標籤截成 32 位元，就是 2^{-32}，不可忽略（約 43 億次嘗試必中一次，線上協定可能真的被試這麼多次）。這就是「明顯異於零」裡「明顯」二字的意義：定義允許可忽略的機率，而標籤長度決定了這個機率的下限。

7.5 與 Week 1 的連結

- **強不可偽造（SUF-CMA）**：把獲勝條件改成「$(m', \tau') \notin \mathcal{Q}'$」，其中 $\mathcal{Q}'$ 記錄訊息–標籤對。這允許「對舊訊息給一個新標籤」也算偽造。Week 1 Problem 1 (vii) 證明：典範驗證 + 確定性 Tag 時兩者等價，所以本課程不必區分。
- **驗證預言機**：真實敵手還能把 (m, τ) 丟給接收端看是否被接受。Week 1 Problem 1 (viii) 證明：這最多讓獲勝機率乘上 $(q_v + 1)$，仍可忽略，所以定義裡不給 $\mathcal{O}_{\mathsf{Vf}}$ 也無妨。

- **為什麼不是 (D1)**「金鑰不可恢復」：常數標籤 $\mathsf{Tag}(\mathsf{ik}, m) = 0$ 的金鑰永遠恢復不了，卻人人可偽造。EUF-CMA 直指我們真正在意的事：偽造。

考試重點

- 能畫出 EUF-CMA 遊戲、說明 $\mathcal{Q}$ 的作用與 $m' \notin \mathcal{Q}$ 的必要性（否則重放合法標籤就算贏，而這對任何正確的 MAC 都做得到）。
- 能解釋「為什麼是 0 而非 $\frac{1}{2}$」：創造 vs. 分辨。
- 能說出名稱 E / UF / CMA 各自的意義。

8 哪些敵手？PPT 與可忽略函數（講義 p. 23）

所有安全定義都長這樣：「**沒有有效率的敵手能以明顯異於** $\frac{1}{2}$（或 0）**的機率獲勝**」。講義 p. 23 把兩個關鍵詞翻譯成術語：**有效率** = PPT，**明顯異於** = 不可忽略。講義說本課程不要求形式定義，但理解它們能避免許多誤解。

8.1 安全參數

現代密碼學不是定義「一個」方案，而是定義由**安全參數**（security parameter）n 索引的一族方案：n 通常就是金鑰長度（$\mathcal{K} = \{0,1\}^n$）。所有東西都是 n 的函數：敵手的執行時間、獲勝機率、金鑰空間大小。慣例上把 1^n（n 個 1 的字串）當輸入交給演算法，使「多項式時間」以 n 為基準。

8.2 有效率 = PPT

機率多項式時間 PPT

演算法 $\mathcal{A}$ 是**機率多項式時間**（probabilistic polynomial time）的，若它可以使用隨機性（擲硬幣），且存在多項式 p 使得對所有輸入，$\mathcal{A}$ 的執行步數至多 $p(n)$，n 是輸入長度（安全參數）。

- **為什麼多項式？**它與計算模型無關（圖靈機、RAM、C 程式互相模擬只差多項式），且在組合下封閉：PPT 呼叫 PPT 多項式次仍是 PPT。這讓「歸約」（用敵手 $\mathcal{A}$ 造出敵手 $\mathcal{B}$）保持在同一個類別裡——Week 1 Problem 1 (viii) 的 $\mathcal{B}$ 就是這樣。
- **暴力搜尋不是 PPT**：嘗試全部 2^n 把金鑰需要 2^n 步，不是任何多項式。這正是為什麼「金鑰空間有限所以理論上總能破」不構成對計算安全的威脅。
- **警告**：n^{100} 也是多項式。漸近定義不在乎常數與次數，所以「PPT 敵手無法攻破」在實務上必須搭配具體的數字（見 §8.4）。

8.3　明顯異於 = 不可忽略

可忽略函數（KL-IMC 定義 3.4）

函數 $f\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ 是**可忽略的**（negligible），若對每一個正多項式 p，存在 N 使得對所有 $n > N$，

$$f(n) < \frac{1}{p(n)}.$$

也就是：f 終究（eventually）比任何多項式的倒數都小。記作 $f(n) = \mathsf{negl}(n)$。

函數	可忽略？	理由
2^{-n}	✅	對任何 p，2^n 終究超過 $p(n)$
$2^{-\sqrt{n}}$、$n^{-\log_2 n}$	✅	比任何多項式倒數小，但比 2^{-n} 大得多（「次指數」）
$1/n^2$、$1/n^{100}$	❌	取 $p(n) = n^{101}$ 就不成立
2^{-30}（常數）	❌	常數不隨 n 變小；在實務上 2^{-30} 也不夠小
$1/\mathrm{poly}(n)$	❌	定義就是要排除這類
$\mathsf{negl}_1(n) + \mathsf{negl}_2(n)$	✅	引理 8.1
$p(n) \cdot \mathsf{negl}(n)$（$p$ 多項式）	✅	引理 8.1——這就是為什麼允許敵手查詢多項式次

引理 8.1（可忽略函數的封閉性）　若 f, g 可忽略、p 是正多項式，則 $f + g$ 與 $p \cdot f$ 都可忽略。

證明. 任取正多項式 q。對 f 用 $2q$、對 g 用 $2q$ 得 N_1, N_2；當 $n > \max(N_1, N_2)$ 時 $f(n) + g(n) < \frac{1}{2q(n)} + \frac{1}{2q(n)} = \frac{1}{q(n)}$。對 $p \cdot f$：對 f 用多項式 $p \cdot q$ 得 N；當 $n > N$ 時 $p(n)f(n) < p(n) \cdot \frac{1}{p(n)q(n)} = \frac{1}{q(n)}$。 ■

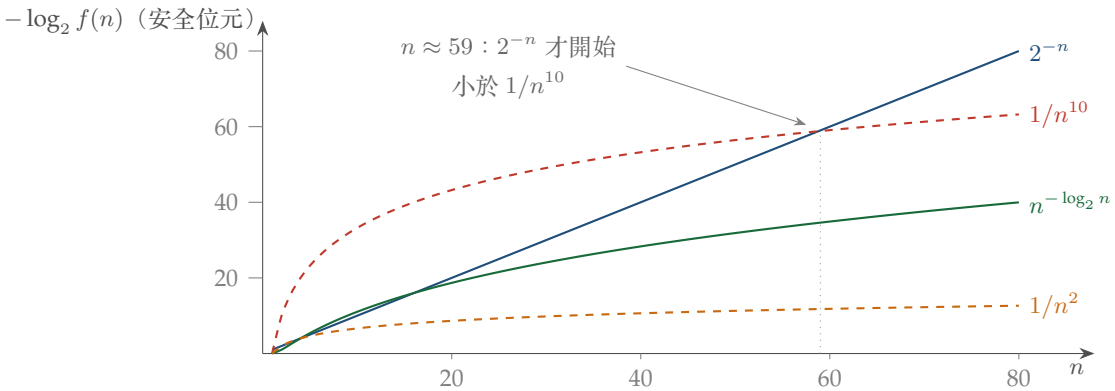


圖 10　實線是可忽略函數，虛線是多項式倒數（縱軸愈高表示函數值愈小）。注意 $1/n^{10}$ 在 $n < 59$ 時其實比 2^{-n} 還小——「可忽略」是漸近性質，只保證「終究」。

8.4 漸近安全 vs. 具體安全

	漸近（asymptotic）	具體（concrete）
句型	對每個 PPT $\mathcal{A}$，$\mathsf{Adv}(\mathcal{A}) \le \mathsf{negl}(n)$	任何執行時間 $\le t$ 的 $\mathcal{A}$，$\mathsf{Adv}(\mathcal{A}) \le \varepsilon$（「$(t,\varepsilon)$-安全」）
例子	「AES 是偽隨機排列」（以 n 為金鑰長度的一族）	「AES-128：$t = 2^{100}$ 時 $\varepsilon \le 2^{-28}$」（即 $t/\varepsilon \approx 2^{128}$）
優點	乾淨、與機器無關、組合容易	可以設定金鑰長度、估計真實風險
缺點	隱藏常數、說不出「幾位元安全」	數字依附於計算模型與年代（摩爾定律、GPU、量子）

課堂定義採漸近版本，但實務上每個標準都附具體數字：「128 位元安全」意指最佳已知攻擊的 $t/\varepsilon \approx 2^{128}$。Kerckhoffs 的 (P1)「*matériellement, sinon mathématiquement*」恰好對應這個區分：**數學上不可破譯 = 資訊論安全**（即使無限算力也破不了，如一次性密碼本）；**實務上不可破譯 = 計算安全**（PPT 敵手破不了）。

⚠ 三個常見誤解

1. 「安全 = 不可能被破」。錯：定義允許可忽略的機率與超多項式的時間。暴力搜尋總是存在。
2. 「PPT 敵手不能贏 = 任何真實程式不能贏」。錯：n^{100} 是多項式；真實保證要看具體數字。
3. 「$\frac{1}{2} + 2^{-30}$ 的勝率沒什麼」。錯（在實務上）：2^{-30} 是常數，不可忽略；且敵手重複 2^{30} 次就能放大優勢。

🎓 考試重點

- PPT：有隨機性、時間是輸入長度（安全參數）的多項式。
- negligible：對每個多項式 p 終究小於 $1/p(n)$；舉得出可忽略與不可忽略的例子各兩個。
- 能把講義的兩句話翻譯成形式：「$\Pr[\text{win}] \le \frac{1}{2} + \mathsf{negl}(n)$」「$\Pr[\text{win}] \le \mathsf{negl}(n)$」。

第二部分

Problem Set 2 詳解

Problem 1：On message and ciphertext spaces · Problem 2：On brute force attacks · Problem 3：On Kerckhoffs' principles

9 Problem 1：On message and ciphertext spaces（函數與無歧義解密）

題目背景

A function $f\colon X \to Y$ is a map that associates an element $y \in Y$ to each element $x \in X$. We write $f(x) = y$ and $f(X) = \{f(x) : x \in X\} \subseteq Y$ for the image of f in Y. We say f is

- **injective** when each $x \in X$ maps to a distinct $y \in Y$, i.e. $f(x) = f(x') \iff x = x'$,
- **surjective** when for each $y \in Y$ there exists at least one $x \in X$ that maps to y, i.e. $f(X) = Y$,
- **bijective** when it is both injective and surjective.

9.1 (i) 判斷五個函數

(i)

For each function below determine whether it is injective, surjective, bijective or none.

a. $f\colon \{1,2\} \to \{1,2,3\}$, $f(1) = 1$, $f(2) = 3$.

b. $g\colon \{1,2\} \to \{1,2,3\}$, $g(1) = 1$, $g(2) = 1$.

c. $h\colon \{1,2,3\} \to \{1,2\}$, $h(1) = 1$, $h(2) = 2$, $h(3) = 1$.

d. $i\colon \{1,2,3\} \to \{1,2,3\}$, $i(1) = 1$, $i(2) = 3$, $i(3) = 2$.

e. $j\colon \{1,2,3\} \to \{1,2\}$, $j(1) = 1$, $j(2) = 1$, $j(3) = 1$.

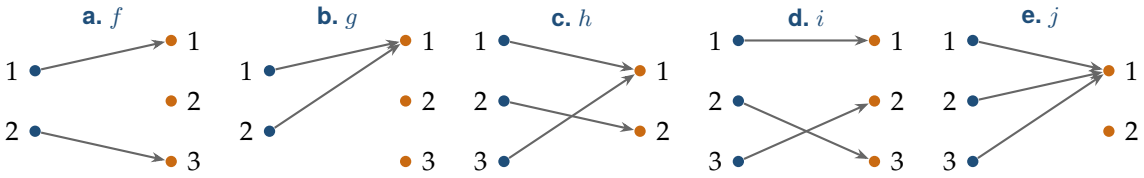


圖 11　(i) 的五個函數。單射：箭頭不合流；滿射：右側每點都被指到。

解答

	單射？	滿射？	結論	理由
a. f	✅ 是	❌ 否	單射	$f(1)=1\neq 3=f(2)$；$2\in Y$ 沒有原像，$f(X)=\{1,3\}\neq Y$
b. g	❌ 否	❌ 否	都不是	$g(1)=g(2)=1$ 但 $1\neq 2$；$2,3$ 無原像
c. h	❌ 否	✅ 是	滿射	$h(1)=h(3)=1$；$h(X)=\{1,2\}=Y$
d. i	✅ 是	✅ 是	雙射	三個輸出 $1,3,2$ 互異且恰好用完 Y；i 是對調 $2,3$ 的排列，$i^{-1}=i$
e. j	❌ 否	❌ 否	都不是	常數函數：三個輸入撞在 1；2 無原像

觀察：a 與 b 的定義域比對應域小（$2<3$），所以不可能滿射；c 與 e 的定義域比對應域大（$3>2$），所以不可能單射；只有 d 的兩邊一樣大，才有機會雙射。這就是 (ii) 要證明的一般定理。

9.2 (ii) 單射、滿射與集合大小

(ii)

Let X and Y be finite sets. Prove that if $f\colon X\to Y$ is injective then $|Y|\geq|X|$. Prove that if f is surjective then $|X|\geq|Y|$. Prove that if f is bijective then $|X|=|Y|$.

解答

令 $X=\{x_1,\ldots,x_s\}$（s 個互異元素）、$|Y|=t$。

(1) 單射 $\Rightarrow |Y|\geq|X|$。考慮 $f(x_1),f(x_2),\ldots,f(x_s)\in Y$。因為 f 單射，$x_a\neq x_b\Rightarrow f(x_a)\neq f(x_b)$，所以這 s 個元素**兩兩相異**。Y 至少包含這 s 個相異元素，故 $t=|Y|\geq s=|X|$。（等價說法：f 把 X 一一對應到它的像 $f(X)$，所以 $|f(X)|=|X|$；又 $f(X)\subseteq Y$，所以 $|X|=|f(X)|\leq|Y|$。）

(2) 滿射 $\Rightarrow |X|\geq|Y|$。因為 f 滿射，每個 $y\in Y$ 至少有一個原像；對每個 y **選定**一個原像，記為 $g(y)\in X$，使 $f(g(y))=y$。這定義了函數 $g\colon Y\to X$。g 是單射：若 $g(y)=g(y')$，兩邊套 f 得 $y=f(g(y))=f(g(y'))=y'$。對 g 套用（1），得 $|X|\geq|Y|$。（直接的鴿籠說法：Y 的 t 個元素各需要至少一個不同的「箭頭」，而每個 x 只射出一支箭，所以箭至少 t 支，即 $s\geq t$。）

(3) 雙射 $\Rightarrow |X|=|Y|$。雙射同時單射與滿射，由（1）$|Y|\geq|X|$、由（2）$|X|\geq|Y|$，故相等。

兩個好用的推論

- **鴿籠原理**（(1) 的逆否）：若 $|X| > |Y|$，任何 $f\colon X \to Y$ 都不是單射——必有兩個輸入撞在一起。
- **有限等大集合上三者等價**：若 $|X| = |Y| < \infty$，則 f 單射 $\iff$ 滿射 $\iff$ 雙射。（單射時像有 $|X|$ 個元素，填滿 Y；滿射時若有兩輸入相撞，像至多 $|X| - 1 < |Y|$ 個，矛盾。）這在 (iv) 很好用。注意對無限集合不成立：$n \mapsto 2n$ 在 $\mathbb{N}$ 上單射但不滿射。

9.3 題目背景：無歧義解密

背景

Decryption is **unambiguous** when, given the secret key sk, it is possible to recover the encrypted message from the ciphertext. This is required for correctness. We consider necessary conditions on $\mathsf{Enc}(\mathsf{sk}, \cdot)$ as a function. When decryption is not unambiguous we say it is **ambiguous**. Let $\mathsf{SKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ with message and ciphertext spaces $\mathcal{M}, \mathcal{C}$. For a fixed sk let $E_{\mathsf{sk}}\colon \mathcal{M} \to \mathcal{C}$, $E_{\mathsf{sk}}(m) = \mathsf{Enc}(\mathsf{sk}, m) = c$.

把「無歧義」翻成數學：對每個合法密文 $c \in E_{\mathsf{sk}}(\mathcal{M})$，**恰有一個** m 使 $E_{\mathsf{sk}}(m) = c$，於是「知道 sk 就能還原 m」有意義；否則，即使知道 sk，看到 c 也不知道是哪一個 m——Dec 是確定性函數，只能輸出一個值，對另一個訊息正確性必然失敗。

9.4 (iii) 滿射但有歧義

(iii)

Give an example of a message space $\mathcal{M}$, a ciphertext space $\mathcal{C}$ and a surjective E_{sk} for which decryption is ambiguous. (Hint: imagine a situation where, even knowing the secret key, it is impossible to know which message was encrypted.)

解答

最小例子：$\mathcal{M} = \{0,1\}$，$\mathcal{C} = \{0\}$，$E_{\mathsf{sk}}(m) = 0$（對任何金鑰、任何訊息都輸出同一個密文）。

- 滿射：$\mathcal{C}$ 唯一的元素 0 被打到，$E_{\mathsf{sk}}(\mathcal{M}) = \{0\} = \mathcal{C}$。
- 有歧義：收到 $c = 0$，即使知道 sk，訊息可能是 0 也可能是 1。確定性的 $\mathsf{Dec}(\mathsf{sk}, 0)$ 只能回其中一個，對另一個訊息 $\mathsf{Dec}(\mathsf{sk}, E_{\mathsf{sk}}(m)) \neq m$，正確性失敗。

稍微「像加密」的例子：$\mathcal{M} = \{0,1\}^2$，$\mathcal{C} = \{0,1\}$，$\mathcal{K} = \{0,1\}$，$E_{\mathsf{sk}}(m_1 m_2) = m_1 \oplus \mathsf{sk}$（把第二個位元丟掉）。滿射（$\mathsf{sk} \oplus 0$ 與 $\mathsf{sk} \oplus 1$ 給出 $0, 1$ 兩個密文），但 $m_1 0$ 與 $m_1 1$ 加密成同一個 c，第二個位元的資訊永遠消失。

有歧義的本質是**不單射**：兩個訊息撞到同一個密文。滿射與此無關——(iii) 與 (iv) 合起來說明「滿射」既不保證也不妨礙無歧義。

9.5 (iv) 滿射且無歧義

(iv)

Give an example of a message space $\mathcal{M}$, a ciphertext space $\mathcal{C}$ and a surjective E_{sk} for which decryption is unambiguous.

解答

例 1（單字母 Caesar）：$\mathcal{M} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26} = \{0, \ldots, 25\}$，$E_{\mathsf{sk}}(m) = m + \mathsf{sk} \bmod 26$。

- 滿射：對任何 c，$m = c - \mathsf{sk} \bmod 26$ 滿足 $E_{\mathsf{sk}}(m) = c$。
- 無歧義：$E_{\mathsf{sk}}(m) = E_{\mathsf{sk}}(m') \Rightarrow m + \mathsf{sk} \equiv m' + \mathsf{sk} \Rightarrow m \equiv m' \pmod{26}$，單射。$\mathsf{Dec}(\mathsf{sk}, c) = c - \mathsf{sk} \bmod 26$ 對每個 c 恰好還原唯一的 m。

例 2（一次性密碼本）：$\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^n$，$E_{\mathsf{sk}}(m) = m \oplus \mathsf{sk}$。它是對合（$E_{\mathsf{sk}} \circ E_{\mathsf{sk}} = \mathrm{id}$），因此是雙射：滿射且單射。

兩個例子裡 $|\mathcal{M}| = |\mathcal{C}|$ 有限，由 (ii) 的推論，滿射自動蘊含單射，所以無歧義。

9.6 (v) 無歧義 $\Rightarrow$ 單射 $\Rightarrow |\mathcal{C}| \geq |\mathcal{M}|$

(v)

Prove that if decryption is unambiguous then E_{sk} is injective. What can you conclude about the sizes of $\mathcal{M}$ and $\mathcal{C}$ if they are both finite?

定理 9.1（無歧義解密的必要條件） 若 SKE 對金鑰 sk 的解密無歧義（等價地，正確性對 sk 成立），則 $E_{\mathsf{sk}} \colon \mathcal{M} \to \mathcal{C}$ 是單射。因此若 $\mathcal{M}, \mathcal{C}$ 有限，$|\mathcal{C}| \geq |\mathcal{M}|$。

解答

證明. 用反證法。假設 E_{sk} 不是單射：存在 $m \neq m'$ 使 $E_{\mathsf{sk}}(m) = E_{\mathsf{sk}}(m') = c$。解密無歧義（正確性）要求 $\mathsf{Dec}(\mathsf{sk}, E_{\mathsf{sk}}(m)) = m$ 且 $\mathsf{Dec}(\mathsf{sk}, E_{\mathsf{sk}}(m')) = m'$。但兩者的輸入都是 (sk, c)，而 Dec 是確定性函數，對同一輸入只能有一個輸出，所以 $m = \mathsf{Dec}(\mathsf{sk}, c) = m'$，與 $m \neq m'$ 矛盾。故 E_{sk} 單射。

若 $\mathcal{M}, \mathcal{C}$ 有限，對單射 $E_{\mathsf{sk}} \colon \mathcal{M} \to \mathcal{C}$ 套用 (ii) 的（1），得 $|\mathcal{C}| \geq |\mathcal{M}|$：**密文空間至少要和訊息空間一樣大**。這就是講義 p. 10 註腳預告的 $t \geq s$。 ■

三個延伸

1. **隨機化加密膨脹得更多。**若 $\mathsf{Enc}(\mathsf{sk}, m; \rho)$ 隨機化，無歧義要求不同訊息的密文集合互不相交：$E_{\mathsf{sk}}(m, \mathcal{R}) \cap E_{\mathsf{sk}}(m', \mathcal{R}) = \varnothing$。於是 $|\mathcal{C}| \geq \sum_m |E_{\mathsf{sk}}(m, \mathcal{R})|$。若每個 ρ 給不同密文（例如密文包含 nonce），$|\mathcal{C}| \geq |\mathcal{M}| \cdot |\mathcal{R}|$——這是隨機化加密「密文比明文長」（ciphertext expansion）的根本原因，也是 IND-CPA 的代價。
2. **單射是「每把金鑰各自」的要求。**定理只說對固定 sk，$m \mapsto \mathsf{Enc}(\mathsf{sk}, m)$ 單射；它沒有說 $(\mathsf{sk}, m) \mapsto \mathsf{Enc}(\mathsf{sk}, m)$ 單射。事實上安全性需要跨金鑰的碰撞：一次性密碼本裡每個 c 都能由每個 m（配上某把金鑰）產生，正是這一點讓敵手無從分辨。
3. **計數。**由命題 4.1，合法的 E_{sk} 只能是 $\binom{t}{s} s!$ 個單射之一，而非全部 t^s 個函數。

9.7 (vi) 單射、非滿射、無歧義：最小的必要假設

(vi)

Give an example of a message space $\mathcal{M}$, a ciphertext space $\mathcal{C}$ and an injective E_{sk} that is not surjective for which decryption is unambiguous. Conclude with a minimal necessary assumption for unambiguous decryption.

解答

例子：$\mathcal{M} = \{0,1\}$，$\mathcal{C} = \{0,1\}^2$，$\mathcal{K} = \{0,1\}$，

$$E_{\mathsf{sk}}(m) = (m \oplus \mathsf{sk}) \,\|\, 0, \qquad \mathsf{Dec}(\mathsf{sk},\ c_1c_2) = \begin{cases} c_1 \oplus \mathsf{sk} & \text{若} c_2 = 0, \\ \bot & \text{若} c_2 = 1. \end{cases}$$

- 單射：$E_{\mathsf{sk}}(m) = E_{\mathsf{sk}}(m') \Rightarrow m \oplus \mathsf{sk} = m' \oplus \mathsf{sk} \Rightarrow m = m'$。
- 非滿射：像 $E_{\mathsf{sk}}(\mathcal{M}) = \{00, 10\}$；01 與 11 永遠不會被產生。
- 無歧義：對合法密文 $c = (m \oplus \mathsf{sk})0$，Dec 回 $c_1 \oplus \mathsf{sk} = m$，正確；對非法密文回 $\bot$——這正是定義裡 $\bot$ 的用途。

結論（最小必要假設）：無歧義解密所需的全部條件是

> 對每一把金鑰 $\mathsf{sk} \in \mathcal{K}$，$E_{\mathsf{sk}} = \mathsf{Enc}(\mathsf{sk}, \cdot) \colon \mathcal{M} \to \mathcal{C}$ **是單射**（若空間有限，等價地蘊含 $|\mathcal{C}| \geq |\mathcal{M}|$）。

它是**必要的**（定理 9.1），也是**充分的**：給定單射 E_{sk}，定義 $\mathsf{Dec}(\mathsf{sk}, c) = E_{\mathsf{sk}}^{-1}(c)$（當 $c \in E_{\mathsf{sk}}(\mathcal{M})$，原像唯一）、否則 $\bot$，即得正確的 Dec。因此它是「最小」的：不能再弱（去掉就錯），也不需再強。不需要滿射、不需要 $|\mathcal{C}| = |\mathcal{M}|$、不需要跨金鑰的單射。

數學上可解密 ≠ 有效率地可解密

定理說單射保證 E_{sk}^{-1} 存在，但沒說它容易算。一個單射函數可能在知道 sk 的情況下仍然難以反算——這聽起來奇怪，卻是 Problem 3 (ii) 的主題（碼書的右欄沒有排序，反查要線性搜尋），也是「單向函數」概念的雛形：正向容易、反向困難的單射。好的加密方案要求 Dec 和 Enc 一樣有效率，這是定義裡「演算法」一詞隱含的要求（PPT）。

10 Problem 2：On brute force attacks（暴力破解的極限）

題目背景：Caesar 密碼的 SKE 形式

The Caesar cipher is a historical example of an SKE. Let $\mathcal{K} = \{0, 1, \ldots, 25\}$ and $\mathcal{M} = \mathcal{C} = \{0, 1, \ldots, 25\}^*$. Encode lowercase letters as $\mathrm{a} = 0, \mathrm{b} = 1, \ldots, \mathrm{z} = 25$. Define $\mathsf{SKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$ as

- KGen: select sk uniformly from $\mathcal{K}$,
- $\mathsf{Enc}(\mathsf{sk}, m)$: parse $m = m_1 m_2 \cdots m_\ell$; for $i \in \{1, \ldots, \ell\}$ let $c_i = m_i + \mathsf{sk} \bmod 26$ and return $c = c_1 c_2 \cdots c_\ell$,
- $\mathsf{Dec}(\mathsf{sk}, c)$: parse $c = c_1 c_2 \cdots c_\ell$; for $i \in \{1, \ldots, \ell\}$ let $m_i = c_i - \mathsf{sk} \bmod 26$ and return $m = m_1 m_2 \cdots m_\ell$.

先確認它是合法的 SKE：KGen 隨機；Enc、Dec 確定；正確性 $\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{sk}, m))_i = (m_i + \mathsf{sk}) - \mathsf{sk} \equiv m_i \pmod{26}$ 對所有 sk, m 成立。對固定 sk，E_{sk} 逐字母是 $\mathbb{Z}_{26}$ 上的雙射 (Problem 1 (iv))，對整個 $\mathcal{M} = \mathbb{Z}_{26}^*$ 也是長度保持的雙射。金鑰空間只有 26 把——暴力破解 (brute force：試遍所有金鑰、看哪個解密結果「像話」) 顯然可行。本題問的是：可行的暴力破解一定成功嗎？

10.1 (i) 密文 ck 有幾種讀法？

(i)

You receive the ciphertext $c = \mathrm{ck}$. If you allow acronyms and abbreviations, how many human readable messages are there that can encrypt to c using the Caesar cipher, and what are the respective keys?

解答

$c = \mathrm{ck} = (2, 10)$。對每把金鑰 $\mathsf{sk} \in \{0, \ldots, 25\}$ 計算 $m = \mathsf{Dec}(\mathsf{sk}, c) = (2 - \mathsf{sk},\ 10 - \mathsf{sk}) \bmod 26$，得到下表（✅ 一般英文單字；⛔ 常見縮寫 / 首字母縮略詞；❌ 難以視為可讀）：

sk	m		說明	sk	m		說明
0	ck	⊖	Calvin Klein；*check* 的縮寫	13	px	⊖	pixel；Post Exchange（美軍福利社）
1	bj	⊖	姓名縮寫（勉強）	14	ow	✓	感嘆詞「哎喲」
2	ai	✓	三趾樹懶；也是 AI	15	nv	⊖	Nevada（州名縮寫）
3	zh	⊖	中文的 ISO 639-1 語言代碼	16	mu	✓	希臘字母 μ
4	yg	✗	—	17	lt	⊖	Lt.（lieutenant）
5	xf	⊖	車款（Jaguar XF）；勉強	18	ks	⊖	Kansas
6	we	✓	代名詞	19	jr	⊖	Jr.（junior）
7	vd	⊖	venereal disease	20	iq	⊖	IQ
8	uc	⊖	University of California；「you see」	21	hp	⊖	HP；horsepower
9	tb	⊖	tuberculosis；terabyte	22	go	✓	動詞
10	sa	⊖	South Africa / Australia	23	fn	⊖	function；鍵盤 Fn 鍵
11	rz	✗	—	24	em	✓	印刷單位 em；'em（them）
12	qy	✗	—	25	dl	⊖	download；decilitre

計數：

- 只算一般英文單字：**6** 個——`ai`（sk = 2）、`we`（6）、`ow`（14）、`mu`（16）、`go`（22）、`em`（24）。
- 題目允許首字母縮略詞與縮寫：再加上 `ck, bj, zh, xf, vd, uc, tb, sa, px, nv, lt, ks, jr, iq, hp, fn, dl` 共 17 個，合計 **23** 個；若剔除最勉強的 `bj`、`xf`，約 **20** 個。只有 `yg`、`rz`、`qy` 三個幾乎無法閱讀。（用一份常見網路字彙表機械檢查，26 個候選中有 22 個出現在表中。）

結論：對這麼短的密文，暴力破解列得出全部 26 個候選，卻選不出正確的那一個：幾乎每把金鑰都給出「可讀」的訊息。「人類可讀」的定義愈寬（允許縮寫），存活的金鑰愈多。

10.2 (ii) 一則密文一定夠嗎？

(ii)

Is a single ciphertext always sufficient to mount a brute force attack against an SKE?

解答

不一定。暴力破解的流程是「對每把 $\mathsf{sk} \in \mathcal{K}$ 計算 $\mathsf{Dec}(\mathsf{sk}, c)$，保留看起來正確的」。它要成功，需要同時滿足幾個條件，而單一密文可能讓任何一個失敗：

1. **金鑰空間可枚舉**。$|\mathcal{K}| = 26$ 當然可以；2^{128} 則不行。這是計算上的限制（§8）。
2. **要有辦法認出正確的明文**。若明文沒有結構（隨機位元、已壓縮資料、另一層加密的密文、金鑰材料），每把金鑰解出來都一樣「像話」，攻擊者無從判斷。這就是為什麼暴力破解通常搭配已知明文（知道某段 m）或明文的冗餘（語言、檔頭、格式）。
3. **密文要夠長**。即使明文是英文，太短的密文會讓多把金鑰都解出可讀結果——(i) 就是例子。所需長度由 Shannon 的**唯一解距離**（unicity distance）刻畫（下方觀念框）。
4. **金鑰空間不能大到「什麼明文都可能」**。極端情形是一次性密碼本：$|\mathcal{K}| \geq |\mathcal{M}|$，對任何 c，每一個同長度的 m 都對應某把金鑰。暴力破解會列出全部可能的明文——等於什麼都沒學到。這是完美保密（Week 1）的另一種說法。

反過來，當密文夠長、明文有冗餘、金鑰空間小，單一密文就足夠：Caesar 密文只要約 2–3 個以上的英文字母，通常就只剩一個可讀解。

Shannon 的唯一解距離（unicity distance）

設語言每個字母的**冗餘**（redundancy）為 $D = \log_2|\alpha| - r$ 位元，其中 r 是語言的熵率（英文約 1.5 位元 / 字母，$\log_2 26 \approx 4.7$，故 $D \approx 3.2$）。對長度 L 的密文，錯誤金鑰解出「像英文」的明文的期望個數約為

$$\bar{s} \;\approx\; (|\mathcal{K}| - 1) \cdot 2^{-D\,L},$$

而使 $\bar{s} \approx 1$ 的長度 $U = \dfrac{\log_2|\mathcal{K}|}{D}$ 稱為唯一解距離：超過 U 之後，解通常唯一。

密碼	$\log_2\lvert\mathcal{K}\rvert$	U（字母）	備註
Caesar	$\log_2 26 \approx 4.7$	≈ 1.5	2 個字母理論上就該唯一——但見下文
單字母替換	$\log_2 26! \approx 88.4$	≈ 28	下週：約 30 字母以上才能唯一決定
AES-128（英文明文）	128	≈ 40	單一區塊（16 字母）不夠；三個區塊有餘
一次性密碼本	$\geq L \log_2 26$	∞	永遠不唯一：完美保密

為什麼 (i) 的結果（6 到 23 個候選）遠多於公式的 $\bar{s} \approx 25 \cdot 2^{-6.4} \approx 0.3$？因為 $D \approx 3.2$ 是長文

本的平均冗餘；極短字串的冗餘低得多——676 個雙字母組合中約 100 個是英文單字（密度 ≈ 0.15，對應 $D \approx 1.4$），而若允許縮寫，幾乎任何兩個字母都「可讀」（密度 $\to 1$，$D \to 0$，$U \to \infty$）。**冗餘愈低，暴力破解愈無力。**

10.3 (iii) 最長的「雙解」Caesar 密文

(iii)

What is the longest ciphertext you can find that can decrypt to at least two human readable messages using the Caesar cipher?

解答

這是一個可以用程式窮舉的問題：兩個訊息 m, m' 能加密成同一個密文，若且唯若 m' 是 m 的某個位移 $k \in \{1, \ldots, 25\}$（此時密文可以取 m 本身，金鑰 0 與 $26 - k$ 都給出可讀解）。所以只要在字典裡找「互為 Caesar 位移」的單字對，取最長者。

```
import string
AL = string.ascii_lowercase
def shift(w, k):
    return "".join(AL[(AL.index(ch) + k) % 26] for ch in w)

words = {w.strip().lower() for w in open("/usr/share/dict/words")
         if w.strip().isalpha() and w.strip().islower()}
pairs = set()
for w in words:
    for k in range(1, 26):
        v = shift(w, k)
        if v in words and v != w:
            pairs.add((len(w),) + tuple(sorted((w, v))) + (k,))
for p in sorted(pairs, reverse=True)[:12]:
    print(p)
```

結果（macOS 內建的 Webster's Second 字典，約 23.5 萬字）——最長為 7 個字母：

長度	單字對	位移	備註
7	`abjurer` ↔ `nowhere`	13（ROT13）	最著名的 ROT13 單字對；兩者皆為一般英文字
7	`primero` ↔ `sulphur`	3	primero 是一種紙牌遊戲（OED 有收）
7	`unfiber` ↔ `bumpily`	7	兩者皆罕見
6	`fusion` ↔ `layout`	6	**兩者皆為日常用字**
6	`cheery` ↔ `purrel`	13	purrel 罕見
6	`becuna` ↔ `orphan`	13	becuna（一種魚）罕見
5	`sleep` ↔ `bunny`	9	日常用字
5	`river` ↔ `arena`	9	日常用字
5	`cheer` ↔ `jolly`	7	日常用字
5	`wheel` ↔ `dolls`	7	日常用字
5	`green` ↔ `terra`	13	—
4	`cold` ↔ `frog` ; `hide` ↔ `stop` ; `meet` ↔ `wood`	3 ; 11 ; 10	日常用字

若只用一份 2 萬字的常用字表，最長是 6 字母的 `fusion` / `layout`，其次是上表的 5 字母對。還有一個三解的例子：密文 `jolly` 可解成 `jolly`（$\mathsf{sk}=0$）、`cheer`（$\mathsf{sk}=7$）與口語的 `diffs`（$\mathsf{sk}=6$）。

建議的作答：「以單一英文單字而言，最長的雙解密文是 7 個字母，例如 $c=$ nowhere 可解為 `nowhere`（$\mathsf{sk}=0$）或 `abjurer`（$\mathsf{sk}=13$）；若限制為日常用字，則為 6 個字母的 $c=$ layout，可解為 `layout`（$\mathsf{sk}=0$）或 `fusion`（$\mathsf{sk}=6$）。」

兩點討論：

- **多字訊息可以無限延長。**若允許多個單字，只要把同一位移的單字對串起來即可：位移 9 的「`sleep river`」↔「`bunny arena`」（11 個字母）、ROT13 的「`green abjurer`」↔「`terra nowhere`」……只是語句愈長愈不像「人類會寫的訊息」。這說明「人類可讀」沒有精確定義，暴力破解的成敗取決於攻擊者對明文的語言模型有多準。
- **為什麼不會更長？**依唯一解距離的估計，長度 L 的英文字串有一個 Caesar 位移也是英文字的機率約 $25\cdot 2^{-3.2L}$：$L=7$ 時約 5×10^{-6}，而英文單字只有幾十萬個，所以 7 字母的例子已屬鳳毛麟角，8 字母以上的單字對預期不存在（在這份字典中確實沒有）。

10.4 (iv) 有加密預言機：一次查詢就夠

(iv)

If an adversary has access to $\mathsf{Enc}(\mathsf{sk},\cdot)$ describe a (non brute force) attack against the Caesar cipher that does not need any ciphertexts.

解答

敵手向預言機查詢 $\mathsf{Enc}(\mathsf{sk}, \mathrm{a})$。因為 $\mathrm{a} = 0$，

$$c = \mathsf{Enc}(\mathsf{sk}, \mathrm{a}) = 0 + \mathsf{sk} \bmod 26 = \mathsf{sk}.$$

一次查詢，直接讀出金鑰，之後所有密文都能解。更一般地，查詢任何已知的 m（哪怕只有一個字母）得 c，則 $\mathsf{sk} = c_1 - m_1 \bmod 26$——不必嘗試 26 把金鑰，一步算出。

這是一個**選擇明文攻擊**（chosen plaintext attack），而且是最強的那種結果：**金鑰恢復**（Week 1 的 (D1)）。它甚至不需要「選擇」：任何一對已知的 (m, c)（known plaintext）都夠，所以就算沒有預言機，只要猜中一個字（例如電報開頭的日期），Caesar 也立刻瓦解。

把 (iv) 放回安全定義

- Caesar 的 Enc 是確定性的，由定理 6.1 它不可能 IND-CPA 安全；(iv) 給出的攻擊比定理的通用攻擊更強——不只分辨，還恢復金鑰。
- Caesar 甚至不是 IND-EAV 安全的（§6 的例子：$m_0 = \mathrm{aa}$、$m_1 = \mathrm{ab}$，看密文兩字母是否相同）。注意這個攻擊不需要預言機也不需要暴力破解——這就是為什麼安全定義不以「金鑰不可恢復」為標準：Caesar 在 (i) 的情境下金鑰確實恢復不了，但它仍然處處洩漏。
- 對任何確定性加密，加密預言機都讓敵手建立一本「碼書」（明文 → 密文對照表）。Caesar 特別慘是因為它對金鑰是線性的：一個對照就決定全部。現代加密用隨機化（nonce）與非線性避免這兩點。

Problem 2 的三個帶走訊息

1. 暴力破解需要：可枚舉的金鑰空間、可辨認的明文（冗餘或已知明文）、超過唯一解距離的密文長度。
2. 冗餘愈低（縮寫、壓縮、隨機資料）暴力破解愈無力；一次性密碼本是極限（任何明文皆可能）。
3. 有加密預言機時，確定性密碼可被建碼書；對 Caesar 一次查詢即恢復金鑰。

11 Problem 3：On Kerckhoffs' principles（碼書與 Kerckhoffs 原則）

題目背景：以碼書加密

Encryption based on codebooks can be shoehorned into the formalism of an SKE. If Alice and Bob wish to communicate they take the Oxford English Dictionary (OED) and for each word in order, e.g. starting with *aardvark*, they choose another word, e.g. *cryptography*, that has not been chosen before. They each write *aardvark* in the left hand column of their codebook and *cryptography* in the same row of the right hand column. (Formally, this is a permutation of the words of the English language.) This represents key generation with the codebook being sk. To encrypt a word, find it in the left hand column and write the word in the same row of the right hand column. This is easy, as hopefully the left hand column is in alphabetical order. To decrypt a word, find it in the right hand column and write the word in the same row of the left hand column...（註：OED 第二版有二十冊、21,730 頁。）

11.1　先把碼書寫成 SKE

碼書加密的形式化

令 $\mathcal{W}$ 為 OED 的單字集合，$N = |\mathcal{W}|$（約 17 萬個現行詞條）。

- $\mathcal{K} = \mathrm{Sym}(\mathcal{W})$：$\mathcal{W}$ 上所有排列 π 的集合，$|\mathcal{K}| = N!$。KGen 依題述程序逐字選擇未用過的字，恰好抽出一個排列（若每次均勻選擇，則 π 均勻）。碼書就是 π 的「表列」：左欄 w、右欄 $\pi(w)$。
- $\mathcal{M} = \mathcal{C} = \mathcal{W}^*$（單字序列）。$\mathsf{Enc}(\pi, w_1 \cdots w_\ell) = \pi(w_1) \cdots \pi(w_\ell)$；$\mathsf{Dec}(\pi, v_1 \cdots v_\ell) = \pi^{-1}(v_1) \cdots \pi^{-1}(v_\ell)$。
- 正確性：$\pi^{-1}(\pi(w)) = w$。E_{sk} 逐字是雙射（Problem 1 (iv)），解密無歧義。

金鑰的資訊量約 $\log_2 N! \approx N \log_2(N/e) \approx 2.7 \times 10^6$ 位元——約 340 KB，遠大於任何現代金鑰。

本質上這是**以單字為字母的單字母替換密碼**（monoalphabetic substitution over a huge alphabet）：確定性、逐符號、金鑰空間巨大。下週分析替換密碼的所有結論（頻率分析）都適用。

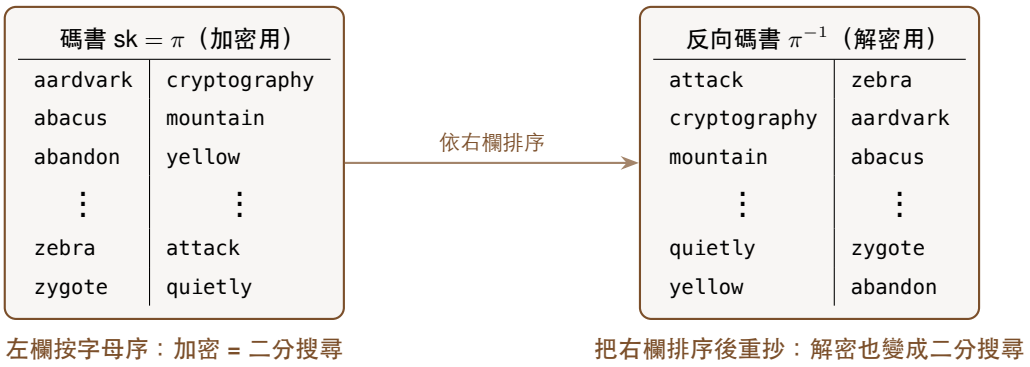


圖 12　碼書與反向碼書：(ii) 的解法是把金鑰加倍。

11.2 (i) 逐條檢視 Kerckhoffs 原則

(i)

Consider the above with respect to each of Kerckhoffs' principles. Which are satisfied and which are not?

解答

	原則	判定	理由
(P1)	實務上不可破譯	⊖	金鑰空間 $N!$ 大到無法暴力破解，但系統是確定性的單字替換：高頻字（the、of、and）、固定句型（「致……司令部」）、已知明文、上下文都讓敵手逐步重建碼書。對「長期、大量」的軍事通訊（Kerckhoffs 的設定）不滿足。
(P2)	系統不需保密	✓	「用碼書做單字替換」這個方法可以完全公開；秘密只在 π。形式上滿足。但金鑰本身就是整個系統的全部內容——碼書被繳獲 = 全盤皆輸，而這會牽動 (P3)。
(P3)	金鑰可不靠筆記傳遞與記憶、可隨時更換	✗✗	**致命缺陷**。金鑰是二十冊、21,730 頁的書：不可能記憶（違反「sans le secours de notes écrites」），分發要靠實體運送，更換金鑰 = 重新產生並印製整套碼書給所有通訊者。一旦疑似外洩，無法「au gré des correspondants」即時更換。
(P4)	相容於電報	⊖	密文是英文單字，可以打電報（歷史上商用電碼本正是為電報省費用而生）。但：(a) 單字沒有冗餘，電報一個字母傳錯整個字就錯、無法更正；(b) 不在 OED 裡的人名、地名、數字需要額外的拼字表。
(P5)	可攜、一人可用	✗	一人可以操作，但二十冊書談不上「portatif」。
(P6)	易用、不費心神、無冗長規則	⊖	加密容易（左欄按字母序，翻查即可）；**解密困難**——右欄是隨機順序，找一個字要線性掃過十幾萬列。規則本身簡單，但操作負擔不對稱。→ (ii)

總結：形式上滿足 (P2)，勉強滿足 (P4)，部分滿足 (P1)、(P6)，嚴重違反 (P3)、(P5)。Kerckhoffs 當年要批評的正是這類「秘密在一本厚書裡」的系統。

11.3 (ii) 讓解密更有效率：把金鑰加倍

(ii)

How might we make decryption more efficient? (Hint: sk is already huge, why not double its size?)

解答

製作反向碼書：把碼書的每一列 $(w, \pi(w))$ 依右欄 $\pi(w)$ 的字母序重新排序並抄成第二本書（圖 12）。第二本書就是 π^{-1} 的表列：左欄是密文字（按字母序），右欄是對應的明文字。解密時在第二本書中二分搜尋，和加密一樣快（$O(\log N)$ 次翻頁，而非 $O(N)$ 的線性掃描）。代價：金鑰（兩人各自持有的秘密）從一本變成兩本——「double its size」——而且兩本必須一致、都要保密。

一部碼 vs. 二部碼

密碼史上這正是**一部碼**（one-part code）與**二部碼**（two-part code）的區別。一部碼的明文與碼字同時按順序排列（例如字母序對應數字遞增），一本書即可雙向查詢，但敵手只要破出幾個碼字，就能靠「順序」內插出鄰近的字，極為脆弱。二部碼把碼字隨機指派（即本題的隨機排列 π），安全得多，但必須印兩本書：加密本（encoding book）與解密本（decoding book）。德國外交部的 0075、13040 碼與各國海軍碼都是二部碼。安全與效率的代價就是**金鑰加倍**——Kerckhoffs 在 1883 年批評的，就是這種體積龐大、難以更換的碼書。

現代觀點：Dec 必須和 Enc 一樣有效率

- 問題 1 (vi) 說「單射保證 E_{sk}^{-1} 存在」；本題說明存在不等於好算。SKE 定義裡 Dec 是「演算法」，隱含必須有效率（PPT）。
- 碼書的解法是**空間換時間**：明寫 π^{-1}。現代區塊加密走另一條路：金鑰只有 128 位元，E_{sk} 與 E_{sk}^{-1} 都由同一把短金鑰透過結構化的回合函數「算出來」（例如 Feistel 結構天生可逆、AES 每個步驟各有反運算），不必儲存任何表。
- 無論哪種做法，**金鑰決定了一個排列及其反排列**；差別只在「存」還是「算」。

11.4 (iii) 用現代語言重寫 Kerckhoffs 原則

(iii)

How might we rewrite Kerckhoffs' principles in the modern day? Do we need to rewrite them all?

解答

	1883 年	現代重寫
(P1)	實務上（即使非數學上）不可破譯	方案必須對**所有有效率（PPT）的敵手滿足明確定義**的安全目標（IND-CPA / IND-CCA、EUF-CMA……），且安全性最好能**歸約**到公認困難的問題；「實務上 vs. 數學上」= 計算安全 vs. 資訊論安全。安全必須對所有攻擊成立，不只是暴力破解。
(P2)	系統不需保密	**不變**。演算法、協定、程式碼都假設公開（Shannon：the enemy knows the system）；安全只依賴金鑰。進一步：應該公開並接受公眾審查與標準化（AES、SHA-3、PQC 競賽）；不要自創密碼。
(P3)	金鑰可不靠筆記記憶、可隨時更換	金鑰應短（128–256 位元）、**均勻隨機產生**、**易於更換**（金鑰輪替、每次會話新金鑰、前向保密）、由**金鑰交換協定**安全地建立，並妥善儲存（硬體安全模組、金鑰派生）。「不靠筆記」的現代版是「不要把金鑰寫死在程式碼或貼在螢幕上」。
(P4)	相容於電報	相容於**當代通訊媒介與協定**：以位元組為介面、低延遲、低頻寬開銷、可在不可靠通道上運作（封包遺失、重排——Week 1 的計數器）、可與現有協定（TLS、IP）整合。
(P5)	可攜、一人可用	可在**受限裝置**上有效率地實作（智慧卡、IoT、手機），可**自動化**（不需人工介入、不需多方同時在場），軟硬體皆可實作且抗側通道。
(P6)	易用、不費心神、無冗長規則	**難以誤用**（misuse resistance）：API 不讓使用者選錯模式、重用 nonce、忘記驗證；預設值安全（secure by default）；對使用者透明（HTTPS 的鎖頭）。「一長串規則」的現代版是「開發者必須記得的 15 個注意事項」——好的函式庫把它們藏起來。

需要全部重寫嗎？

- (P2) 幾乎一字不改——它是唯一被升格為「原則」的一條，而且被整個現代密碼學內化（安全遊戲裡敵手知道演算法）。
- (P3) 的精神完全不變（金鑰小、可換、可管理），只是從「人腦記憶」換成「機器產生與儲存」。
- (P1) 需要實質重寫：Kerckhoffs 只有模糊的「indéchiffrable」，現代有精確的定義與歸約證明。這是 1883 與今日最大的差距。
- (P4)–(P6) 字面過時（電報、可攜、心神），但它們指向的**工程與可用性**面向不但沒過時，反而是當代多數真實漏洞的來源（Heartbleed、nonce 重用、padding oracle 都不是「數學被破」）。它們值得重寫，而非刪除。

11.5 (iv) 違反 Kerckhoffs 原則的災難

(iv)

Find an example where the failure to apply one or more of Kerckhoffs' principles has been catastrophic.

解答

主例：**MIFARE Classic 與秘密密碼 Crypto-1（違反 (P2)、(P1)）**。

- **系統**：NXP（前 Philips）自 1994–95 年起銷售的非接觸式智慧卡晶片，用於倫敦 Oyster 卡、荷蘭 OV-chipkaart、澳洲 SmartRider、全球無數門禁與票務系統；2008 年時累計售出逾 10 億顆、約 2 億張在用，占非接觸卡市場約 85%。
- **違反 (P2)**：卡片的加密與認證依賴專有串流密碼 **Crypto-1**，設計保密、金鑰僅 **48** 位元。48 位元在 1998 年（EFF 的 DES 破解機）就已知不足；NXP 實際上依賴的是「沒人知道演算法」。
- **被逆向**：2007 年 12 月 Karsten Nohl 與 Henryk Plötz 在 24C3 以晶片切片顯微攝影還原大部分電路；2008 年 3 月 Radboud 大學（Garcia、de Koning Gans、Jacobs 等）純粹從卡片與讀卡機的通訊紀錄完整還原 Crypto-1 與認證協定。
- **災難性的結果**：Crypto-1 本身有嚴重的密碼學弱點（線性回饋移位暫存器搭配弱的濾波函數、16 位元狀態的弱亂數產生器），加上協定缺陷，Radboud 的第二個攻擊**只需一到兩次對讀卡機的部分認證、不到一秒、約 8 MB 記憶體**即可還原金鑰；卡片可被完整複製，Oyster 卡被示範免費搭車，荷蘭政府大樓的門禁卡被複製。
- **法律與後果**：NXP 向法院聲請禁止 Radboud 在 ESORICS 2008 發表，2008 年 7 月 18 日 Arnhem 法院以言論自由駁回，並指出「損害並非來自發表，而是來自卡片明顯的缺陷」。荷蘭 OV-chipkaart 計畫大幅修改、各系統陸續遷移到使用公開演算法（3DES / AES）的 MIFARE DESFire，耗費數年與鉅資。Garcia 等人的結論一語中的：「*Security by obscurity is volatile. Do not develop your own crypto but use standards.*」

其他案例 A：演算法保密（違反 (P2)，連帶 (P1)）

案例	違反	發生了什麼	後果
GSM 的 COMP128、A5/1、A5/2（1998–2003）	(P2)、(P1)	SIM 認證演算法 COMP128 與語音加密 A5 設計保密。1998 年 4 月 Briceno、Goldberg、Wagner 由外洩文件與逆向還原 COMP128，一天內找到致命弱點（「narrow pipe」），約 2^{17} 次查詢即可複製 SIM；1999 年從手機逆向出 A5/1、A5/2，A5/2（出口弱化版）當場被破，2003 年 Barkan–Biham–Keller 給出即時唯密文攻擊。	全球數億用戶的 SIM 可被複製、通話可被即時竊聽；A5/2 的存在還讓攻擊者能對 A5/1 用戶發動降級攻擊。
DVD 的 CSS（1999）	(P2)、(P1)	內容擾亂系統使用保密的 40 位元串流密碼。1999 年因某播放器未保護其金鑰而被逆向（DeCSS），隨後 Frank Stevenson 證明可在 2^{25} 步內破解。	整個 DVD 版權保護體系失效；訴訟與「程式碼是否為言論」之爭（Week 1）。
KeeLoq 汽車遙控器（2006–08）	(P2)、(P1)	Microchip 的專有區塊加密（64 位元金鑰）用於多家車廠的遙控門鎖與防盜器；演算法 2006 年外洩，2007–08 年出現實用的代數攻擊與側通道攻擊。	遙控器可被複製、車廠主金鑰可被還原。
TETRA 的 TEA1（1995–2023）	(P2)、(P1)	歐洲警用、軍用與關鍵基礎設施無線電標準的加密演算法 TEA1–TEA4 由 ETSI 保密、僅簽 NDA 者可見。2023 年 Midnight Blue（TETRA:BURST）從 Motorola 無線電韌體逆向出全部演算法，發現 TEA1 把 80 位元金鑰**內部縮減為 32 位元**——筆電一分鐘可暴力破解。	全球使用 TEA1 的警察、電網、管線通訊二十餘年來實際上可被竊聽與注入；ETSI 直到 2023–25 年才公開演算法。

其他案例 B：金鑰管理與操作（違反 (P3)、(P6)）

案例	違反	發生了什麼	後果
Zimmermann 電報（1917）	(P3)	德國外交部的二部碼碼書（0075、13040）體積龐大、難以更換；舊碼 13040 早被英國 Room 40 大量重建。駐華府大使館把電報改用舊碼 **13040** 重新編碼轉發墨西哥，英國遂取得完整明文。	電報公開後美國於 1917 年 4 月參戰；碼書「不可隨時更換」的代價是戰略級的。
Enigma 的金鑰表（1939–45）	(P3)、(P6)	機器本身的設計波蘭人 1932 年已重建（符合 (P2) 的預期），但每月**紙本金鑰表**（違反「不靠筆記」）屢次被繳獲（U-110、氣象船 München、Lauenburg），加上繁瑣的指示程序導致操作員犯錯（重複指示、固定開頭）。	盟軍長期讀取 U 艇通訊，大西洋戰役轉向。

共同教訓：違反 (P2) 的系統往往同時違反 (P1)——因為沒有公眾審查，設計者看不到自己的弱點（Crypto-1 的濾波函數、COMP128 的窄管線、TEA1 的金鑰縮減）；而一旦演算法被逆向（晶片切片、韌體分析、文件外洩——總是會發生），就沒有任何東西剩下。違反 (P3) 的系統則敗在金鑰無法更換：被繳獲的碼書與金鑰表會一直被利用下去。

延伸思考

- TETRA 的 TEA1 弱化據稱是為了符合出口管制（Wassenaar）。這與 Week 1 的 Netscape 40 位元出口版是同一個故事：**刻意的弱點 + 保密的設計 = 二十年後的災難**。
- 「演算法保密」在上述每個案例中都延後了攻擊幾年，但也延後了修補：公開的 DES 在 1970 年代就被徹底分析並在設計時就知道壽命；Crypto-1 用了 13 年才被外界檢視。保密買到的是「不知道自己有多弱」。
- 如果你是 1995 年 ETSI 的設計者，Kerckhoffs 會建議你怎麼做？（公開演算法、由金鑰長度而非演算法保密控制出口強度、讓學界審查。）

第三部分

總複習

本週重點整理 · 中英術語對照表 · 自我測驗 · 參考資料

12　本週重點整理

12.1　一頁看懂第二週

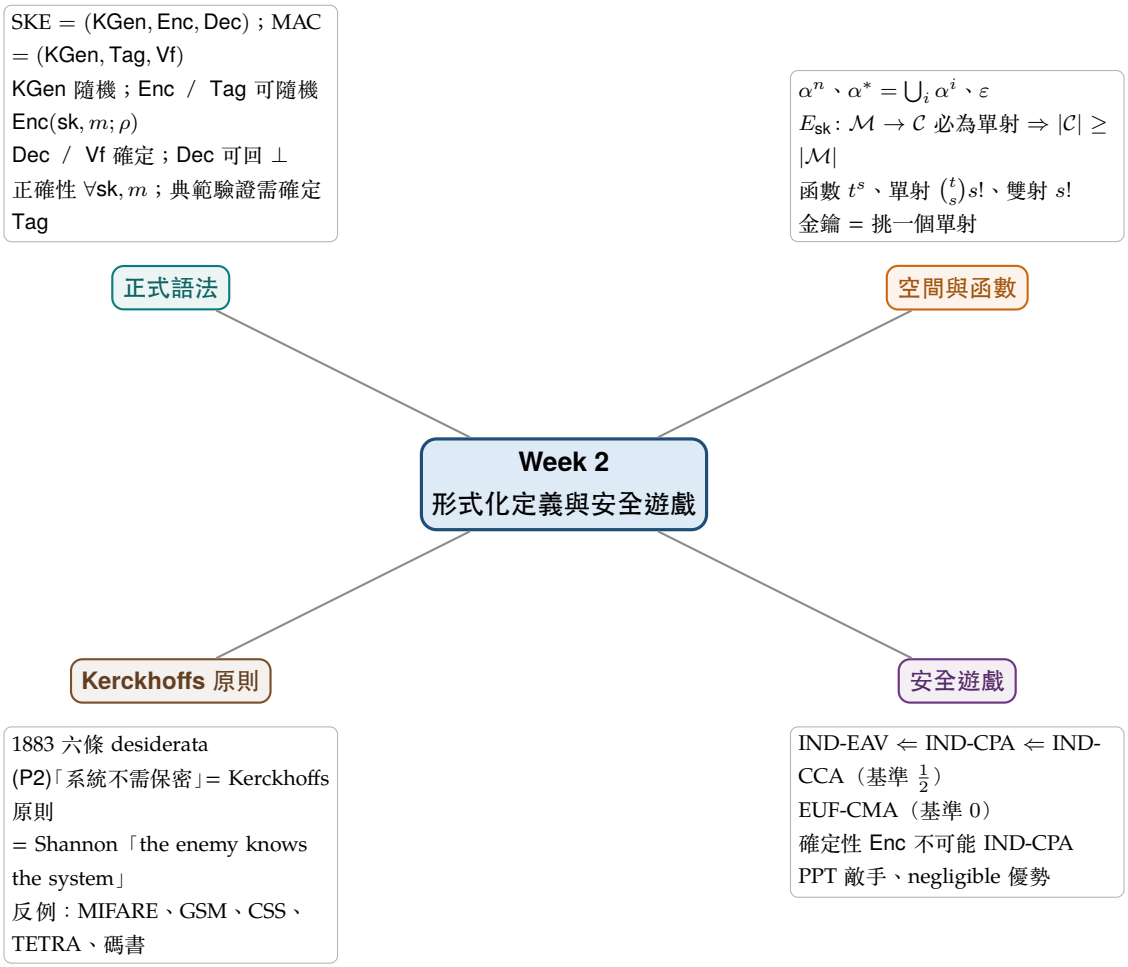


圖 13　第二週概念地圖。

12.2　必背定義

概念	一句話定義
SKE 語法	$\mathsf{sk} \leftarrow \mathsf{KGen}$；$c \leftarrow \mathsf{Enc}(\mathsf{sk}, m)$（或 $=$）；$m/\perp = \mathsf{Dec}(\mathsf{sk}, c)$；空間 $\mathcal{K}, \mathcal{M}, \mathcal{C}$
隨機化演算法	確定性演算法 + 隨機輸入 $\rho \in \mathcal{R} = \{0,1\}^r$：$\mathsf{Enc}(\mathsf{sk}, m; \rho)$
SKE 正確性	$\forall \mathsf{sk} \in \mathcal{K}, \forall m \in \mathcal{M}$：$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{sk}, m)) = m$
MAC 語法 / 正確性	$\mathsf{ik} \leftarrow \mathsf{KGen}$；$\tau = \mathsf{Tag}(\mathsf{ik}, m)$；$\mathsf{Vf}(\mathsf{ik}, m, \tau) \in \{0,1\}$；$\mathsf{Vf}(\mathsf{ik}, m, \mathsf{Tag}(\mathsf{ik}, m)) = 1$
典範驗證	Vf 重算 $\tau' = \mathsf{Tag}(\mathsf{ik}, m)$ 並比對；需要確定性 Tag
字母表 / Kleene 星號	α^n 長度 n 字串、$\lvert\alpha^n\rvert = \lvert\alpha\rvert^n$；$\alpha^* = \bigcup_{i \in \mathbb{N}} \alpha^i \ni \varepsilon$
單射 / 滿射 / 雙射	$f(x) = f(x') \Rightarrow x = x'$；$f(X) = Y$；兩者皆是
無歧義解密	E_{sk} 單射（對每把 sk）；有限時 $\lvert\mathcal{C}\rvert \geq \lvert\mathcal{M}\rvert$；單射亦充分
Kerckhoffs 原則	安全只能依賴金鑰保密，不能依賴演算法保密（第二條）
IND-EAV	$\mathcal{A}$ 送等長 m_0, m_1，收 $c_b \leftarrow \mathsf{Enc}(\mathsf{sk}, m_b)$，猜 b；$\Pr[\text{win}] \leq \frac{1}{2} + \mathsf{negl}$
IND-CPA / IND-CCA	同上，加 $\mathcal{O}_{\mathsf{Enc}}$ / 加 $\mathcal{O}_{\mathsf{Enc}}, \mathcal{O}_{\mathsf{Dec}}$（$\mathcal{O}_{\mathsf{Dec}}(c_b) = \perp$）
EUF-CMA	$\mathcal{A}^{\mathcal{O}_{\mathsf{Tag}}}$ 輸出 (m', τ')，贏若 $\mathsf{Vf} = 1$ 且 $m' \notin \mathcal{Q}$；$\Pr[\text{win}] \leq \mathsf{negl}$
PPT	有隨機性、執行時間是安全參數 n 的多項式
negligible	對每個多項式 p，終究 $f(n) < 1/p(n)$；例 2^{-n}；反例 $1/n^2$
唯一解距離	$U = \log_2\lvert\mathcal{K}\rvert / D$；Caesar ≈ 1.5 字母、替換 ≈ 28 字母

12.3　常見誤區

⚠ 考試時最容易犯的錯

1. **把「Enc 可能隨機」寫成「Dec 可能隨機」**。Dec 與 Vf 永遠確定性。
2. **以為正確性只要「大部分」訊息成立**。是對所有 sk、所有 m。
3. **混淆單射與滿射的方向**。單射 $\Rightarrow \lvert Y\rvert \geq \lvert X\rvert$（箭頭不合流，右邊夠大）；滿射 $\Rightarrow \lvert X\rvert \geq \lvert Y\rvert$。
4. **以為無歧義解密需要滿射或 $\lvert\mathcal{C}\rvert = \lvert\mathcal{M}\rvert$**。只需要單射；$\perp$ 處理像集之外的密文。
5. **把 Kerckhoffs 原則說成「演算法必須公開」**。是「即使公開也要安全」；也不是「金鑰可以公開」。
6. **IND 遊戲忘記 $\lvert m_0\rvert = \lvert m_1\rvert$ 或 CCA 忘記 $\mathcal{O}_{\mathsf{Dec}}(c_b) = \perp$**。前者因長度無法隱藏，後者排除平凡勝利。
7. **以為 IND-EAV 安全的方案可以重複使用金鑰**。一次性密碼本是 IND-EAV（單次）卻不是 IND-CPA；多訊息安全需要隨機化。
8. **把「基準 $\frac{1}{2}$」套到 EUF-CMA**。偽造是創造，不是二選一，基準是 0。
9. **以為 $1/n^{10}$ 可忽略、2^{-30} 可忽略**。多項式倒數與常數都不可忽略。

10. **以為金鑰空間大就安全。**26! 的替換密碼與 $N!$ 的碼書都敗給頻率分析；(P1) 是對所有攻擊。

13 中英術語對照表

English	中文	說明
secret key encryption (SKE)	秘密金鑰加密	(KGen, Enc, Dec) 與 $\mathcal{K}, \mathcal{M}, \mathcal{C}$
message authentication code (MAC)	訊息鑑別碼	(KGen, Tag, Vf) 與 $\mathcal{K}, \mathcal{M}, \mathcal{T}$
randomised algorithm	隨機化演算法	可擲硬幣；等價於多一個隨機輸入 ρ
deterministic algorithm	確定性演算法	同輸入必同輸出；Dec、Vf、本課的 Tag
randomness space $\mathcal{R}$	隨機性空間	$\rho \leftarrow \mathcal{R} = \{0,1\}^r$；$\mathsf{Enc}(\mathsf{sk}, m; \rho)$
random coins	隨機帶	演算法用到的隨機位元
failure symbol $\perp$	失敗符號	Dec 對非法密文的輸出；讀作「bottom」
correctness	正確性	$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{sk}, m)) = m$；$\mathsf{Vf}(\mathsf{ik}, m, \mathsf{Tag}(\mathsf{ik}, m)) = 1$
canonical verification	典範驗證	重算標籤並比對；需確定性 Tag
integrity key ik	完整性金鑰	MAC 的金鑰，與 sk 獨立
tag space $\mathcal{T}$	標籤空間	標籤所在集合；長度決定猜中機率 $1/\lvert\mathcal{T}\rvert$
alphabet	字母表	有限符號集合 α
string / word	字串	符號的有限序列
empty string ε	空字串	長度 0；$\alpha^0 = \{\varepsilon\}$
Kleene star α^*	Kleene 星號	任意有限長度字串的集合
natural numbers $\mathbb{N}$	自然數	本課含 0：$\{0, 1, 2, \dots\}$
function	函數	每個輸入恰對應一個輸出
image $f(X)$	像	所有輸出所成的集合
injective (one-to-one)	單射	$f(x) = f(x') \Rightarrow x = x'$
surjective (onto)	滿射	$f(X) = Y$
bijective	雙射	單射且滿射；可逆
inverse function f^{-1}	反函數	雙射才有；D_{sk} 是 E_{sk} 的左反函數
pigeonhole principle	鴿籠原理	$\lvert X\rvert > \lvert Y\rvert \Rightarrow$ 無單射 $X \to Y$
permutation	排列	集合到自身的雙射；n 個元素有 $n!$ 個
unambiguous decryption	無歧義解密	知 sk 可唯一還原 m；$\Longleftrightarrow$ E_{sk} 單射
ciphertext expansion	密文膨脹	隨機化加密 $\lvert\mathcal{C}\rvert \geq \lvert\mathcal{M}\rvert \cdot \lvert\mathcal{R}\rvert$
Kerckhoffs' principle	Kerckhoffs 原則	安全不得依賴演算法保密（第二條）
desiderata	期望條件	Kerckhoffs 對六條原則的原稱
Shannon's maxim	Shannon 格言	「the enemy knows the system」
security through obscurity	安全靠隱匿	以保密設計為安全依據；反面教材
open design	開放設計	Saltzer–Schroeder（1975）的安全設計原則
codebook	碼書	單字（或片語）到碼字的對照表

（續下頁）

English	中文	說明
one-part / two-part code	一部碼 / 二部碼	碼字有序 / 隨機；後者需兩本書
cryptographic game	密碼學遊戲	挑戰者與敵手依規則互動，定義安全
challenger	挑戰者	持有秘密、執行遊戲規則的一方
adversary	敵手	試圖獲勝的（PPT）演算法
oracle	預言機	敵手可呼叫的黑盒：$\mathcal{O}_{\mathsf{Enc}}, \mathcal{O}_{\mathsf{Dec}}, \mathcal{O}_{\mathsf{Tag}}$
eavesdropper	竊聽者	只觀察密文的被動敵手
chosen plaintext attack (CPA)	選擇明文攻擊	敵手可取得自選明文的密文
chosen ciphertext attack (CCA)	選擇密文攻擊	敵手可取得自選密文的解密（$c \neq c_b$）
chosen message attack (CMA)	選擇訊息攻擊	敵手可取得自選訊息的標籤
indistinguishability (IND)	不可區分性	分不出兩個自選訊息中哪個被加密
challenge ciphertext c_b	挑戰密文	遊戲中要被分辨的那一個密文
advantage	優勢	$\lvert\Pr[\text{win}] - \frac{1}{2}\rvert$（IND）或 $\Pr[\text{win}]$（EUF）
semantic security	語意安全	密文不幫助計算任何訊息函數；等價 IND-EAV
Hamming weight	漢明重量	位元串中 1 的個數
malleability	可延展性	改密文可預測地改明文；IND-CCA 排除之
existential unforgeability (EUF)	存在性不可偽造	對任何新訊息皆不可偽造
strong unforgeability (SUF)	強不可偽造	連舊訊息的新標籤也不可偽造
forgery	偽造	通過 Vf 的非合法產生之 (m', τ')
security parameter n	安全參數	通常為金鑰長度；一切以 n 為基準
probabilistic polynomial time (PPT)	機率多項式時間	「有效率」的形式定義
negligible function	可忽略函數	終究小於任何多項式倒數
non-negligible / noticeable	不可忽略 / 明顯	至少是某個多項式的倒數（無窮多次）
asymptotic security	漸近安全	以 $n \to \infty$ 敘述；PPT 與 negligible
concrete security	具體安全	(t, ε)-安全：時間 $\le t$ 則優勢 $\le \varepsilon$
information-theoretic security	資訊論安全	對無限算力敵手仍安全（一次性密碼本）
computational security	計算安全	僅對有效率敵手安全
brute force attack	暴力破解	枚舉所有金鑰並辨認明文
known / chosen plaintext	已知 / 選擇明文	攻擊者握有 (m, c) 對 / 可指定 m
unicity distance	唯一解距離	$U = \log_2\lvert\mathcal{K}\rvert / D$：解唯一所需最短密文
redundancy	冗餘	語言每字母多餘的位元數 $D = \log_2\lvert\alpha\rvert - r$
spurious key	偽金鑰	解出可讀明文但錯誤的金鑰
Caesar cipher	Caesar 密碼	$c_i = m_i + \mathsf{sk} \bmod 26$；$\lvert\mathcal{K}\rvert = 26$
monoalphabetic substitution	單字母替換	金鑰為字母表的排列；26! 把金鑰
ROT13	ROT13	位移 13 的 Caesar；加密即解密

14 自我測驗（附解答）

建議先不看解答，每題寫下自己的答案再比對。

1. 語法

在 KGen, Enc, Dec, Tag, Vf 五個演算法中，哪些一定隨機、哪些可以隨機、哪些一定確定？為什麼 Dec 不能隨機？$\perp$ 在定義裡扮演什麼角色？

解答

KGen 一定隨機（否則人人算得出同一把金鑰）；Enc、Tag 可以隨機（本課的 Tag 取確定性）；Dec、Vf 一定確定。Dec 若隨機，正確性只能是機率性的且毫無好處——可把隨機帶固定而不失正確性；收件人需要明確答案。$\perp$ 讓 Dec 能對像集之外、格式錯誤或驗證失敗的密文給出「非法」判決，使 E_{sk} 不必滿射（Problem 1 (vi)）。

2. 計數與單射

令 $f\colon \{0,1\}^3 \to \{0,1\}^2$ 為「丟掉最後一個位元」。f 單射嗎？滿射嗎？$\{0,1\}^2 \to \{0,1\}^3$ 的函數有幾個？其中單射有幾個？f 能是某個正確 SKE 的 E_{sk} 嗎？

解答

f 不單射（$000, 001 \mapsto 00$）、滿射（每個 2 位元串都有原像）。$\{0,1\}^2 \to \{0,1\}^3$ 的函數有 $8^4 = 4096$ 個，單射有 $8 \cdot 7 \cdot 6 \cdot 5 = 1680$ 個（$\binom{8}{4} \cdot 4! = 70 \cdot 24$）。$f$ 不單射，故不可能是正確 SKE 的 E_{sk}（定理 9.1）；方向相反的單射 $\{0,1\}^2 \to \{0,1\}^3$ 則可以。

3. 跨金鑰的單射？

證明或反駁：若對每把 sk，E_{sk} 都是單射，則 $(\mathsf{sk}, m) \mapsto \mathsf{Enc}(\mathsf{sk}, m)$ 也是單射。

解答

反駁。一次性密碼本 $\mathsf{Enc}(\mathsf{sk}, m) = \mathsf{sk} \oplus m$（$\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^n$）：每個 E_{sk} 是雙射，但 $\mathsf{Enc}(\mathsf{sk}, m) = \mathsf{Enc}(\mathsf{sk} \oplus 1^n, m \oplus 1^n)$，不單射。事實上跨金鑰的碰撞是保密的來源：同一個 c 可由任何 m（配適當金鑰）產生。單射只是「每把金鑰各自」的要求。

4. 分類 Kerckhoffs 原則

下列各情況違反了哪一條原則？(a) 金鑰印在卡片上放在皮夾；(b) 系統需要三名操作員同時在場；(c) 加密只能透過傳真機運作；(d) 廠商宣稱「我們的演算法保密，所以更安全」；(e) 操作手冊有 40 頁必須逐條遵守的規則。

解答

(a) (P3)（金鑰應不靠筆記）；(b) (P5)（一人可用）；(c) (P4)（相容於通訊媒介）；(d) (P2)（安全不得依賴演算法保密；若同時用強演算法且不依賴保密，則不違反，但缺乏公眾審查通常也連帶危及 (P1)）；(e) (P6)（不應要求一長串規則）。

5. 設計一個 IND-EAV 敵手

方案把兩個 n 位元區塊的訊息 $m = m_1 \| m_2$ 加密為 $\mathsf{Enc}(\mathsf{sk}, m) = (m_1 \oplus \mathsf{sk}) \| (m_2 \oplus \mathsf{sk})$，$\mathsf{sk} \in \{0,1\}^n$。寫出一個必勝的 IND-EAV 敵手。

解答

送 $m_0 = 0^n \| 0^n$、$m_1 = 0^n \| 1^n$。收到 $c_b = c_1 \| c_2$：若 $c_1 = c_2$ 輸出 0，否則輸出 1。$b = 0$ 時 $c_1 = c_2 = \mathsf{sk}$；$b = 1$ 時 $c_2 = \mathsf{sk} \oplus 1^n \neq \mathsf{sk} = c_1$。獲勝機率 1。教訓：重複使用金鑰串流讓「兩段明文是否相同」洩漏——這就是 Caesar「aa vs. ab」攻擊的位元版。

6. 隨機性不夠多

假設 Enc 只使用 $r = 10$ 位元的隨機性 ρ（其餘確定）。說明一個用約 $2^{10} \cdot 10$ 次 $\mathcal{O}_{\mathsf{Enc}}$ 查詢、幾乎必勝的 IND-CPA 敵手。這對實務上 nonce 的長度有何啟示？

解答

固定 $m_0 \neq m_1$。向 $\mathcal{O}_{\mathsf{Enc}}(m_0)$ 查詢約 $2^{10} \ln 2^{10} \approx 7000$ 次（優惠券收集），以高機率收集到 m_0 的全部 $\leq 2^{10}$ 個可能密文，記為集合 S。送挑戰 (m_0, m_1)，若 $c_b \in S$ 輸出 0，否則 1。$b = 0$ 時 $c_b \in S$（已收集完整）；$b = 1$ 時 c_b 是 m_1 的密文，由無歧義解密不可能等於任何 m_0 的密文，故 $\notin S$。因此隨機性空間必須大到無法枚舉：這就是 nonce 要 96–128 位元、且絕不能重複的原因。

7. 為什麼不乾脆禁止挑戰後的解密查詢？

IND-CCA 遊戲只禁止 $\mathcal{O}_{\mathsf{Dec}}(c_b)$，而非禁止挑戰後的一切解密查詢。後者是另一個（較弱的）定義，為什麼講義採用前者？

解答

只允許挑戰前查詢的版本稱為 CCA1（「午餐時間攻擊」）；講義的版本是 CCA2（適應性）。真實攻擊幾乎都是適應性的：padding oracle 攻擊者先拿到目標密文 c_b，再送出它的變體 $c' \neq c_b$ 給伺服器觀察反應。CCA1 擋不住這類攻擊，CCA2 可以；而 $\mathcal{O}_{\mathsf{Dec}}(c_b) = \perp$ 是唯一必要的例外（否則平凡獲勝）。

8. 截短的標籤

某系統為省頻寬，把 HMAC 的輸出截成 16 位元當標籤。它是 EUF-CMA 安全的嗎？漸近與具體兩個角度各怎麼說？

解答

敵手隨便挑新訊息 m' 與隨機 $\tau' \in \{0,1\}^{16}$，獲勝機率 2^{-16}。若標籤長度固定為 16（不隨安全參數增長），2^{-16} 是常數、不可忽略——漸近上不安全。具體上，線上協定被試 $2^{16} = 65{,}536$ 次完全可行，所以也不安全。標籤長度應隨 n 增長，實務上至少 64、通常 128 位元（$\tau \in \mathcal{T}$ 的大小決定「明顯異於零」裡的下限）。

9. 可忽略嗎？

判斷下列函數是否可忽略：(a) $2^{-n/2}$；(b) n^{-3}；(c) $2^{-(\log_2 n)^2}$；(d) $1/n!$。另證明：若 f 可忽略，則 $\sqrt{f}$ 也可忽略。

解答

(a) 是（$2^{n/2}$ 終究超過任何多項式）；(b) 否（取 $p(n) = n^4$）；(c) 是：$2^{-(\log_2 n)^2} = n^{-\log_2 n}$，指數 $\log_2 n \to \infty$ 超過任何固定次數；(d) 是（$n! \geq 2^n$ 對 $n \geq 4$）。證明：任取多項式 p，對 f 用多項式 p^2 得 N，當 $n > N$ 時 $f(n) < 1/p(n)^2$，開根號得 $\sqrt{f(n)} < 1/p(n)$。

10. 唯一解距離

(a) 金鑰為 5 個字母的 Vigenère 密碼（$|\mathcal{K}| = 26^5$），用 $D \approx 3.2$ 估計唯一解距離。(b) 若極短字串的有效冗餘只有 $D \approx 1.4$，長度 2 的 Caesar 密文預期有幾把偽金鑰？這與 Problem 2 (i) 的結果一致嗎？

解答

(a) $\log_2 26^5 = 5 \times 4.70 \approx 23.5$ 位元，$U \approx 23.5/3.2 \approx 7.3$ 個字母：約 8 個字母的密文就足以唯一決定金鑰（不過找到金鑰仍需 Kasiski / 頻率分析，下週）。(b) $\bar{s} \approx 25 \cdot 2^{-1.4\times 2} \approx 25 \cdot 0.144 \approx 3.6$ 把，與 (i) 找到的 6 個一般單字（扣掉正確的一個約 5 個偽解）同一數量級；允許縮寫時 $D \to 0$，$\bar{s} \to 25$，與「23 個可讀」一致。

參考資料

課程教材

- N. K. Nguyen, E. W. Postlethwaite, *6CCS3CIS Cryptography, second lecture*, King's College London, 9 October 2025.
- E. W. Postlethwaite, *Problem Set 2*, 6CCS3CIS Cryptography, 6 October 2025.

課本與延伸閱讀

- J. Katz, Y. Lindell, *Introduction to Modern Cryptography*, 3rd ed., CRC Press, 2021. 本週對應：§1.3（Kerckhoffs 原則與三個理由）、§3.1（計算安全、PPT、negligible，定義 3.4）、§3.2.1（IND-EAV 與語意安全，定理 3.13）、§3.4.2（IND-CPA）、§4.2（EUF-CMA、強不可偽造）、§5.1.2（IND-CCA）。
- M. Rosulek, *The Joy of Cryptography*, 2021（免費線上）。第 1–2 章：加密語法、Kerckhoffs 原則、不可區分性的「函式庫」觀點。`https://joyofcryptography.com`
- D. Stinson, M. Paterson, *Cryptography: Theory and Practice*, 4th ed., CRC Press, 2018. §3.5（Shannon 理論、唯一解距離、偽金鑰）。
- C. E. Shannon, "Communication Theory of Secrecy Systems", *Bell System Technical Journal* 28(4), 1949, pp. 656–715.（完美保密、唯一解距離、「the enemy knows the system」）

Kerckhoffs 原則

- A. Kerckhoffs, "La cryptographie militaire", *Journal des sciences militaires*, vol. IX, pp. 5–38 (Jan. 1883), pp. 161–191 (Feb. 1883). 掃描與英譯：F. Petitcolas, `https://petitcolas.net/kerckhoffs/`
- J.-C. Caraco, R. Géraud-Stewart, D. Naccache, "Kerckhoffs' Legacy", IACR ePrint 2020/556.
- J. H. Saltzer, M. D. Schroeder, "The Protection of Information in Computer Systems", *Proc. IEEE* 63(9), 1975.（「開放設計」原則）

Problem 3 (iv) 的案例原始文獻

- F. D. Garcia, G. de Koning Gans, R. Muijrers, P. van Rossum, R. Verdult, R. Wichers Schreur, B. Jacobs, "Dismantling MIFARE Classic", *ESORICS 2008*, LNCS 5283. `https://sos.cs.ru.nl/applications/rfid/2008-esorics.pdf`
- K. Nohl, D. Evans, Starbug, H. Plötz, "Reverse-Engineering a Cryptographic RFID Tag", *USENIX Security 2008*.
- Rechtbank Arnhem, kort geding vonnis LJN BD7578, 18 juli 2008（NXP v. Radboud Universiteit）。
- M. Briceno, I. Goldberg, D. Wagner, "GSM Cloning", ISAAC, UC Berkeley, April 1998. `http://www.isaac.cs.berkeley.edu/isaac/gsm.html`
- E. Barkan, E. Biham, N. Keller, "Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication", *CRYPTO 2003*; *J. Cryptology* 21, 2008.
- A. Biryukov, A. Shamir, D. Wagner, "Real Time Cryptanalysis of A5/1 on a PC", *FSE 2000*.

- S. Indesteege, N. Keller, O. Dunkelman, E. Biham, B. Preneel, “A Practical Attack on KeeLoq”, *EUROCRYPT 2008*.
- C. Meijer, W. Bokslag, J. Wetzels, “All cops are broadcasting: TETRA under scrutiny”, *USENIX Security 2023*.
  `https://www.usenix.org/system/files/usenixsecurity23-meijer.pdf`
  TETRA:BURST 專案網站：`https://tetraburst.com`
- J. von zur Gathen, “Zimmermann Telegram: The Original Draft”, *Cryptologia* 31(1), 2007.
- W. F. Friedman, C. J. Mendelsohn, *The Zimmermann Telegram of January 16, 1917 and its Cryptographic Background*, War Department, 1938（NSA 解密文件）。

Problem 2 使用的資料

- macOS 內建字典 `/usr/share/dict/words`（Webster’s Second International, 1934；約 23.5 萬字）。
- 常用字表：J. Kaufman, *google-10000-english*（取自 Google Trillion Word Corpus），GitHub。

本教材為學習輔助資料，依據公開講義、習題與文獻整理撰寫；Problem Set 的解答為作者之理解與推導，
如與課堂公布之標準答案有出入，請以課堂為準。