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

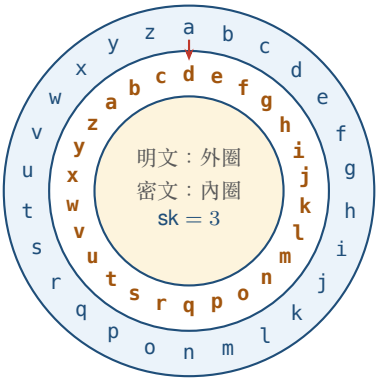
Week 3

歷史密碼、破譯方法與完美保密

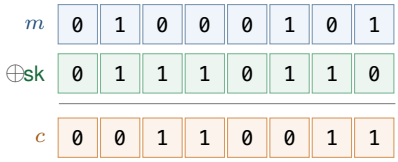
完全學習教材 & Problem Set 3 詳解

模算術與 XOR • Caesar / 單字母替換 / Vigenère • 暴力搜尋與頻率分析

Kasiski 與重合指數 • trace • 完美保密 • 一次性密碼本及其限制



$c_i = m_i + \mathsf{sk} \bmod 26$



一次性密碼本：$c = m \oplus \mathsf{sk}$

金鑰均勻、與訊息等長、只用一次

$\Rightarrow$ 完美保密：$\Pr[M = m \mid C = c] = \Pr[M = m]$

教材內容 講義逐頁精讀、延伸補充（Katz–Lindell 第 2 章、Shannon 1949、Kasiski / Friedman、一次性密碼本的歷史）、圖解、Problem Set 3 全部題目詳解（含三段電影台詞密文的完整破譯過程與程式）、重點整理、中英術語表、自我測驗

對應教材 Third lecture（104 頁投影片，實際內容 30 頁）、Problem Set 3（Problems 1–3，共 16 小題）、KL-IMC §1.3 與第 2 章

適用對象 修習 6CCS3CIS（Level 6）之學生；建議先讀 Week 1–2 教材；需基本離散數學與機率

依據 6CCS3CIS 第三週講義（Nguyen & Postlethwaite, 2025-10-09）與 Problem Set 3（Postlethwaite, 2025-10-09）編寫，並參考 Katz–Lindell《Introduction to Modern Cryptography》第三版、Shannon（1949）、Kasiski（1863）、Bellovin（2011）及 NSA 解密文獻

目錄

導讀：如何使用本教材 **3**

1 模算術（講義 pp. 2–6） **7**
1.1 $\mathbb{Z}/n\mathbb{Z}$：會「繞回」的整數（講義 p. 2） . . . 7
1.2 模算術像整數算術（講義 p. 3） . . . 7
1.3 $n = 26$：字母的算術（講義 p. 4） . . . 8
1.4 $n = 2$：位元的算術與 $\oplus$（講義 p. 5） . . . 8
1.5 逐元素的模算術（講義 p. 6） . . . 8

2 Caesar 密碼（講義 pp. 7–8） **9**
2.1 正式定義（講義 p. 7） . . . 9
2.2 討論：如何「均勻抽樣」金鑰？（講義 p. 8） . . . 10
2.3 討論：潛在弱點（講義 p. 8） . . . 11

3 單字母替換密碼（講義 pp. 9–12） **11**
3.1 正式定義（講義 pp. 9–10） . . . 11
3.2 討論：金鑰空間的大小（講義 p. 11） . . . 12
3.3 討論：如何均勻抽出一個排列？（講義 p. 11） . . . 12
3.4 討論：潛在弱點（講義 p. 12） . . . 13

4 Vigenère 密碼（講義 pp. 13–15） **14**
4.1 正式定義（講義 p. 13） . . . 14
4.2 討論：j 個交錯的 Caesar 密碼（講義 p. 14） . . . 15
4.3 討論：潛在弱點（講義 p. 15） . . . 16

5 反思：從「設計—破解」循環到定義（講義 pp. 16–17） **17**
5.1 三個密碼的共同模式（講義 p. 16） . . . 17
5.2 換位密碼不會改變結論（講義 p. 17） . . . 17

6 歷史密碼的密碼分析（講義 pp. 18–22） **18**
6.1 先說清楚：以下方法「不是必要的」（講義 p. 18） . . . 19
6.2 暴力搜尋（講義 p. 19） . . . 19
6.3 頻率分析（講義 pp. 20–21） . . . 20
6.4 Bonus weakness：trace（講義 p. 22） . . . 22

7 完美保密（講義 pp. 23–26） **23**
7.1 直覺（講義 p. 23） . . . 23
7.2 正式定義 . . . 24
7.3 講義的例子：$\mathbb{Z}/26\mathbb{Z}$ 上的一次性密碼本（講義 pp. 24–25） . . . 25
7.4 限制（講義 p. 26） . . . 26

8 一次性密碼本（講義 pp. 27–29） **27**
8.1 定義（講義 p. 27） . . . 27
8.2 千萬不要重用金鑰（講義 p. 28） . . . 27
8.3 歷史：Miller、Vernam、Mauborgne（與 Shannon、Kotelnikov） . . . 28

8.4 反思：完美但脆弱（講義 p. 29） 29

9 Problem 1：On the insufficient security of historical ciphers 32
9.1 (i) 以 SKE 框架定義三個密碼 32
9.2 (ii) 用 Enc(sk, ·) 取回金鑰 33
9.3 (iii) 一對 m_0, m_1 區分出 b 33

10 Problem 2：On the one time pad 34
10.1 (i) 定義 OTP（訊息長度至多 n 位元） 35
10.2 (ii) 用 Enc(sk, ·) 取回 sk 35
10.3 (iii) 為何 (ii) 不與完美保密矛盾？ 36
10.4 (iv) 抽到全零金鑰的機率 36
10.5 (v) 全零金鑰存在，為何仍完美保密？ 37
10.6 (vi) 單一位元、金鑰重用：如何猜 sk？ 38
10.7 (vii) 可以用同一把金鑰加密多則訊息嗎？ 39
10.8 (viii) OTP 滿足哪些 Kerckhoffs 原則？ 39
10.9 (ix) 一次性密碼本的歷史使用 40

11 Problem 3：On practical use and cryptanalysis of historical ciphers 42
11.1 (i) 用三個密碼加密與解密 42
11.2 (ii) Caesar 密文 43
11.3 (iii) 單字母替換密文 45
11.4 (iv) Vigenère 密文（金鑰長度 3） 46

12 本週重點整理 51
12.1 一頁看懂第三週 51
12.2 必背定義與結論 51
12.3 常見誤區 52
12.4 與前後週的連結 53

13 中英術語對照表 53

14 自我測驗（附解答） 55

參考資料 59

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	Third lecture（2025 年 10 月 9 日，104 張投影片含逐步顯示，實際內容 30 頁）
本週習題	Problem Set 3（Eamonn W. Postlethwaite，三大題共 16 小題，其中 Problem 3 含三段待破譯的電影台詞密文）
課本	Katz & Lindell, *Introduction to Modern Cryptography*, 3rd ed.（簡稱 KL-IMC）
本週對應章節	KL-IMC §1.3（歷史密碼與其破譯）、第 2 章（完美保密：§2.1 定義、§2.2 一次性密碼本、§2.3 限制、§2.4 Shannon 定理）

本週在整門課的位置

前兩週建立了語言：SKE 的語法與正確性（Week 1–2）、Kerckhoffs 原則、以及用遊戲寫成的安全定義 IND-EAV / IND-CPA / IND-CCA（Week 2）。第三週第一次把這套語言用在**具體的方案**上：

- 先補足數學工具——**模算術**（$\mathbb{Z}/n\mathbb{Z}$）與它在 $n = 26$（字母）和 $n = 2$（位元，$\oplus$）的兩個特例；
- 把三個**歷史密碼**（Caesar、單字母替換、Vigenère）寫成正式的 $(\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$，並逐一指出它們在 Week 2 的定義下為何立刻不安全——不需要任何「聰明的」破譯；
- 介紹真正的**破譯方法**（暴力搜尋、頻率分析、Kasiski / 重合指數、trace），這些方法在歷史上花了幾百年才出現，但今天只是「額外的羞辱」；
- 反思「發現漏洞 → 加複雜度 → 再被破」的無原則循環，引出現代密碼學「假設 ⇒ 定義」的證明式思維；
- 最後給出一個**真正安全**的方案：**一次性密碼本**（one time pad）與 Shannon 的**完美保密**，並說明它為什麼「不能讓大家回家」——金鑰必須與訊息等長、只能用一次。

這個「完美但不實用」的結論正是下一階段的出發點：如何用短金鑰得到計算上安全的加密。

學習目標

讀完本教材你應該能夠：

1. 在 $\mathbb{Z}/26\mathbb{Z}$ 與 $\mathbb{Z}/2\mathbb{Z}$ 中正確計算（含負數與「繞回」），並說明加法群的四條性質為什麼讓「加金鑰 / 減金鑰」成為一對可逆操作。

2. 把 Caesar、單字母替換、Vigenère 寫成完整的 SKE（$\mathcal{K}, \mathcal{M}, \mathcal{C}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec}$），算出各自的金鑰空間大小，並說明如何均勻抽樣金鑰（拒絕抽樣、Fisher–Yates）。
3. 對每一個歷史密碼給出：一次 $\mathsf{Enc}(\mathsf{sk}, \cdot)$ 查詢即可還原金鑰的方法、一對等長訊息 m_0, m_1 使 IND-EAV 敵手必勝。
4. 解釋並執行暴力搜尋、頻率分析（含重合指數與 χ^2 檢定）、Vigenère 的分流攻擊、以及 Caesar 的 trace 不變量。
5. 寫出 Shannon 完美保密的定義與其等價形式，證明一次性密碼本是完美保密的，並證明 $|\mathcal{K}| \geq |\mathcal{M}|$。
6. 說明一次性密碼本的三個實務限制（金鑰長度、單次使用、金鑰配送），解釋「全零金鑰」為何不是問題、「重複使用金鑰」為何是災難，並舉出歷史實例（VENONA、華盛頓–莫斯科熱線）。
7. 完整解答 Problem Set 3 的全部 16 小題，包括親手（與用程式）破譯三段電影台詞密文。

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 30 頁），並補充 KL-IMC、Shannon 原始論文與歷史文獻的內容。
- **第二部分（§9–§11）**：Problem Set 3 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。Problem 3 的三段密文附完整的手動破譯步驟與可重現的 Python 程式。
- **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
$\mathbb{Z}/n\mathbb{Z} = \{0, 1, \ldots, n-1\}$	模 n 的整數；$x \bmod n$ 為 x 除以 n 的非負餘數
$\mathtt{a} \leftrightarrow 0$, $\mathtt{b} \leftrightarrow 1, \ldots, \mathtt{z} \leftrightarrow 25$	字母與 $\mathbb{Z}/26\mathbb{Z}$ 的對應；字母運算一律指對應數字的模 26 運算
$\oplus$	模 2 加法（XOR），逐位元作用於等長位元串
α、α^n、$\alpha^{\leq n}$、α^*	字母表；長度 n 的字串；長度至多 n 的字串；任意長度的字串
$[\ell] = \{1, 2, \ldots, \ell\}$	索引集合；$m = m_1 m_2 \cdots m_\ell$ 表示逐字元拆解
$\mathrm{Perm}(\alpha)$	α 上所有排列（雙射 $\alpha \to \alpha$）的集合，$\lvert\mathrm{Perm}(\alpha)\rvert = \lvert\alpha\rvert!$
$x \leftarrow X$	從有限集合 X 均勻隨機抽出 x
$\Pr[M = m \mid C = c]$	看到密文 c 之後，訊息為 m 的事後機率（完美保密的核心量）
$\mathrm{trace}(w)$	字串 w 相鄰字母的差分序列（§6）
IC	重合指數（index of coincidence）
`typewriter`	明文與密文字串一律以等寬字體排印

建議讀法

本週的講義有大量「Sample sk and try encrypting and decrypting a short message of your choice」這類動手指令，請真的拿紙筆做一次——Problem 3 (i) 就是這件事。三段電影台詞的密文（Problem 3 (ii)–(iv)）建議先不看解答，依 §6 的方法自己破一次，再對照 §11 的逐步過程；教材附的程式可以用來驗證。

第一部分

講義精讀與延伸

對應講義 pp. 2–29：模算術、三種歷史密碼的形式化、暴力搜尋與頻率分析、trace、完美保密、一次性密碼本及其限制

1　模算術（講義 pp. 2–6）

1.1　$\mathbb{Z}/n\mathbb{Z}$：會「繞回」的整數（講義 p. 2）

模 n 的整數

令 n 為正整數。**模 n 的整數**（integers modulo n）是集合

$$\mathbb{Z}/n\mathbb{Z} = \{0, 1, \ldots, n-1\},$$

其上的加法與減法和整數 $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ 一樣，只是結果要再取**除以 n 的餘數**（記作 mod$n$），使它落回 $\{0, \ldots, n-1\}$。更一般地，兩個整數 a, b 稱為**模 n 同餘**（congruent modulo n），記 $a \equiv b \pmod{n}$，若 $n \mid (a-b)$；$\mathbb{Z}/n\mathbb{Z}$ 的每個元素就是一整類同餘的整數的「代表」。

講義的例子（$n = 5$）：$1 - 4 = -3 \equiv 2$、$2 + 2 = 4$、$3 + 4 = 7 \equiv 2 \pmod 5$。要點是**負數也要繞回**：$-3$ 加上 5 得 2。一般地，x mod n 永遠取 $\{0, \ldots, n-1\}$ 中的那個代表。

程式語言的 % 不一定是數學的 mod

Python 的 `(-3) % 26` 得 `23`（與數學一致），但 C、C++、Java、JavaScript 的 `-3 % 26` 得 `-3`（餘數跟著被除數的正負號走）。寫解密 $m_i = c_i - \mathsf{sk}$ 時，這是最常見的 bug 來源；安全寫法是 `((x % n) + n) % n`。

1.2　模算術像整數算術（講義 p. 3）

講義列出 $\mathbb{Z}/n\mathbb{Z}$ 上加法的四條性質，它們合起來就是代數裡「**交換群**（abelian group）」的定義：

性質	式子（對所有 $x, y, z \in \mathbb{Z}/n\mathbb{Z}$）	對密碼的意義
單位元（identity）	$x + 0 = 0 + x = x$	金鑰 0 是「什麼都不做」的金鑰（但它仍是合法金鑰，見 §10 (v)）
反元素（inverse）	對每個 x 存在 $-x$ 使 $x + (-x) = 0$；$-x = n - x$	加金鑰可以被「減金鑰」完全抵銷：這就是 Dec 存在的理由
交換律（commutative）	$x + y = y + x$	先加訊息再加金鑰、或反過來，結果一樣
結合律（associative）	$(x + y) + z = x + (y + z)$	多次加密 / 解密可以任意結合，例如 $(m + \mathsf{sk}) - \mathsf{sk} = m + (\mathsf{sk} - \mathsf{sk}) = m$

為什麼密碼學要用模算術？

三個理由：(1) **封閉性**——字母加字母還是字母，所以 $\mathcal{C} = \mathcal{M}$，密文不會跑出字母表；(2) **可逆性**——每個加法都有反元素，所以「加 sk」是 $\mathbb{Z}/n\mathbb{Z}$ 上的雙射（Week 2：E_{sk} 必須單射），解密就是加 $-\mathsf{sk}$；(3) **均勻性**——若 sk 在 $\mathbb{Z}/n\mathbb{Z}$ 上均勻，則對任何固定的 m，$m + \mathsf{sk}$ 也在 $\mathbb{Z}/n\mathbb{Z}$ 上均勻。第三點是 §7 完美保密證明的全部內容。

1.3　$n = 26$：字母的算術（講義 p. 4）

講義約定 a $\leftrightarrow 0$, b $\leftrightarrow 1, \ldots,$ z $\leftrightarrow 25$，於是字母可以相加減：

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12
n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

講義的三個例子：

- a $+$ c $=$ c，即 $0 + 2 = 2$；
- b $-$ d $=$ y，即 $1 - 3 = -2 \equiv 24$；
- f $-$ f $=$ a，即 $5 - 5 = 0$。

四條群性質照樣成立：$x - x = 0$、$x + y = y + x$、$x + 0 = 0 + x$、$(x + y) + z = x + (y + z) \pmod{26}$。

心算模 26 的訣竅

把字母想成時鐘上的 26 個刻度（封面的密碼盤）。「加 k」就是順時針轉 k 格，「減 k」逆時針轉 k 格，轉過 z 就接回 a。負數 $-k$ 與 $26 - k$ 是同一個位置：$-3 \equiv 23$、$-8 \equiv 18$。

1.4　$n = 2$：位元的算術與 $\oplus$（講義 p. 5）

模 2 加法有專屬符號 $\oplus$（XOR，exclusive or）：

$$0 \oplus 0 = 0, \qquad 0 \oplus 1 = 1, \qquad 1 \oplus 0 = 1, \qquad 1 \oplus 1 = 0.$$

讀法：「兩個位元不同時得 1，相同時得 0」。四條性質變成：

$$a \oplus a = 0, \qquad a \oplus b = b \oplus a, \qquad a \oplus 0 = a, \qquad (a \oplus b) \oplus c = a \oplus (b \oplus c).$$

$\oplus$ 的特別之處：自己是自己的反元素

在 $\mathbb{Z}/2\mathbb{Z}$ 中 $-1 \equiv 1$，所以「減」和「加」是同一件事：$a \oplus a = 0$ 說明每個位元的反元素就是它自己。後果：一次性密碼本的 Dec 與 Enc 是同一個演算法（§8）：

$$(m \oplus \mathsf{sk}) \oplus \mathsf{sk} = m \oplus (\mathsf{sk} \oplus \mathsf{sk}) = m \oplus 0 = m.$$

這一行用到了結合律、自反元素、單位元三條性質——模算術的性質不是裝飾，而是正確性證明的每一步。

1.5　逐元素的模算術（講義 p. 6）

對字串（向量）做運算時，**逐位置獨立進行、不進位**（schoolbook addition without carries）。講義的兩個例子：

$n = 2$：

$$\begin{array}{r} 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1 \\ \oplus\, 0\ 1\ 1\ 1\ 0\ 1\ 1\ 0 \\ \hline 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1 \end{array}$$

$n = 26$：

$$\begin{array}{r} \mathtt{h\ e\ l\ l\ o} \\ -\,\mathtt{a\ b\ c\ d\ e} \\ \hline \mathtt{h\ d\ j\ i\ k} \end{array} \qquad \begin{array}{r} 7\ 4\ 11\ 11\ 14 \\ -\,0\ 1\ 2\ 3\ 4 \\ \hline 7\ 3\ 9\ 8\ 10 \end{array}$$

「不進位」正是重點：第 i 個位置的結果只和第 i 個位置的輸入有關。這讓加密可以逐字元定義（Caesar、Vigenère、一次性密碼本都是 $c_i = m_i + k_i$ 的形式，差別只在 k_i 怎麼來），也讓敵手可以逐字元攻擊——這是 §6 所有方法的共同基礎。

考試重點

- 能算 $\mathtt{b} - \mathtt{d}$、$-3 \bmod 26$、$(1 \oplus 1) \oplus 1$ 這類題目，且知道負數要加 n。
- 能說出 $(\mathbb{Z}/n\mathbb{Z}, +)$ 的四條性質，並指出 Dec 的正確性用到哪幾條。
- 知道 $\oplus$ 就是模 2 加法、XOR 的真值表、以及 $a \oplus a = 0$ 使加解密同演算法。

2 Caesar 密碼（講義 pp. 7–8）

2.1 正式定義（講義 p. 7）

Caesar 密碼

令 $\alpha = \{\mathtt{a}, \ldots, \mathtt{z}\}$（與 $\mathbb{Z}/26\mathbb{Z}$ 對應）。

$$\mathcal{K} = \{0, 1, \ldots, 25\} = \mathbb{Z}/26\mathbb{Z}, \qquad \mathcal{M} = \mathcal{C} = \alpha^*.$$

- KGen：從 $\mathcal{K}$ 均勻抽出 sk。
- $\mathsf{Enc}(\mathsf{sk}, m)$：把 $m = m_1 m_2 \cdots m_\ell$ 拆成單一字元；對 $i \in [\ell]$ 令 $c_i = m_i + \mathsf{sk} \bmod 26$，回傳 $c = c_1 c_2 \cdots c_\ell$。
- $\mathsf{Dec}(\mathsf{sk}, c)$：把 $c = c_1 c_2 \cdots c_\ell$ 拆成單一字元；對 $i \in [\ell]$ 令 $m_i = c_i - \mathsf{sk} \bmod 26$，回傳 $m = m_1 m_2 \cdots m_\ell$。

正確性：$(m_i + \mathsf{sk}) - \mathsf{sk} = m_i$（反元素與結合律）。

幾點觀察：

- 固定 sk 後，單一字母上的映射 $x \mapsto x + \mathsf{sk}$ 是 α 上的**雙射**（反函數是 $x \mapsto x - \mathsf{sk}$），因此 E_{sk} 在每個長度 ℓ 上都是 $\alpha^\ell \to \alpha^\ell$ 的雙射——符合 Week 2 的「E_{sk} 必須單射」。
- Enc 是**確定性、無狀態的**：同一 (sk, m) 永遠給同一個 c。Week 2 定理 6.1 已經告訴我們：這種方案不可能 IND-CPA 安全。
- $\mathsf{sk} = 0$ 是合法金鑰，此時 $c = m$。這不是 bug——見 §10 (v) 的討論：把它拿掉反而會洩漏資訊。

- 據傳 Caesar 最愛用的金鑰是 d ↔ 3；sk = 13 的特例叫 ROT13，因為 $13 + 13 = 26 \equiv 0$，加密與解密是同一個操作。

動手做：講義 p. 7 的指令

取 $m =$ veni vidi vici、sk $= 3$：

m:	v	e	n	i	v	i	d	i	v	i	c	i
	21	4	13	8	21	8	3	8	21	8	2	8
+3:	24	7	16	11	24	11	6	11	24	11	5	11
c:	y	h	q	l	y	l	g	l	y	l	f	l

得 $c =$ yhql ylgl ylfl（空白保留不加密）。解密把每個數字減 3：$24 - 3 = 21 =$ v，依此類推。

Suetonius 的記載

Caesar 密碼的名字來自 Suetonius《羅馬十二帝王傳》〈神聖的尤利烏斯〉第 56 章：Caesar 寫機密信件時「把字母表的順序改變，使人讀不出任何一個字；若要解讀，須以第四個字母 D 代替 A，其餘類推」。Suetonius 也記載其繼承人 Augustus 用位移 1（B 代 A、C 代 B ……），且最後一個字母不繞回而寫成 AA。當時並無任何破譯單字母密碼的技術記載；已知最早的頻率分析出自 9 世紀巴格達的 al-Kindī（§3）。

2.2 討論：如何「均勻抽樣」金鑰？（講義 p. 8）

若 $|\mathcal{K}| = N$，均勻抽樣指每把金鑰被選中的機率恰為 $1/N$。講義的提示「擲五枚硬幣」：五個公平位元給出 $2^5 = 32$ 個等機率結果，但我們只要 26 個。

拒絕抽樣（rejection sampling）

重複執行：擲五枚硬幣得 $x \in \{0, \ldots, 31\}$；若 $x \le 25$ 則輸出 sk $= x$，否則丟棄重來。每一輪被接受的機率是 $26/32 = 81.25\%$，期望輪數 $32/26 \approx 1.23$；在「被接受」的條件下，$0, \ldots, 25$ 每個的機率都是 $\frac{1/32}{26/32} = \frac{1}{26}$，所以輸出**恰好均勻**。

模偏差（modulo bias）

一個常見的錯誤是直接取 sk $= x \bmod 26$：此時 $0, 1, 2, 3, 4, 5$ 各有兩個 x（x 與 $x + 26$）對應，機率 $2/32$，其餘 20 個只有 $1/32$。六把金鑰比別人兩倍可能，敵手暴力搜尋時先試這六把，平均成本立刻下降。對 Caesar 這只是小傷；對 26! 把金鑰的替換密碼，有偏差的抽樣可能讓「有效金鑰空間」縮水好幾個數量級（§3）。一般原則：**從 2^k 個等機率結果得到 N 個等機率結果，若 $N \nmid 2^k$，必須拒絕一部分結果**——沒有不丟棄又均勻的方法（因為 $1/N$ 不是 $1/2^k$ 的整數倍）。

2.3 討論：潛在弱點（講義 p. 8）

講義列出五個弱點，每一個都對應到 Week 2 的某個概念：

弱點（講義原文）	說明與對應概念
small $\mathcal{K}$ implies efficient brute force, if one can check guesses for sk	$\lvert\mathcal{K}\rvert = 26$：敵手把 26 把金鑰全試一遍（§6）。「能檢查猜測」是前提：需要知道明文大概長什麼樣（Week 2 的唯一解距離：Caesar 只需約 1.5 個字母）或有一對 (m, c)。
message, ciphertext pairs give sk	一對 (m, c) 中任一個位置就給出 $\mathsf{sk} = c_i - m_i$。這是「已知明文攻擊」的極端：一個字母就完全破解。
structure, e.g. 'gaps' between consecutive letters, preserved	相鄰字母的差 $c_{i+1} - c_i = m_{i+1} - m_i$ 不受金鑰影響（講義 p. 22 的 trace，§6）。
frequency analysis	每個明文字母都被同一個位移送到同一個密文字母，字母頻率分布只是整體平移（§6）。
(more fundamentally) deterministic encryption	Week 2 定理 6.1：確定性加密不可能 IND-CPA。事實上連 IND-EAV 都不過：$m_0 = \mathsf{aa}$、$m_1 = \mathsf{ab}$，看密文兩字母是否相同即知 b（Week 2 §6.3、本週 Problem 1 (iii)）。

大金鑰空間是必要條件，不是充分條件

講義強調：五個弱點中只有第一個可以靠把 $\mathcal{K}$ 變大來「修補」，而且這是**必要而非充分的**。下一節的單字母替換密碼有 $26! \approx 2^{88}$ 把金鑰，暴力搜尋完全不可行，卻仍被頻率分析輕易攻破；其他四個弱點一個也沒有消失。安全性不是「金鑰夠多」就好，而是要對一個明確的定義（IND-EAV / CPA / CCA）給出論證。

3 單字母替換密碼（講義 pp. 9–12）

3.1 正式定義（講義 pp. 9–10）

排列與單字母替換密碼

集合 X 上的**排列**（permutation）是雙射 $\pi\colon X \to X$；所有排列的集合記為 $\mathrm{Perm}(X)$（Week 2 寫成 $\mathrm{Sym}(X)$）。**單字母替換密碼**（monoalphabetic substitution cipher）：字母表 α，

$$\mathcal{K} = \mathrm{Perm}(\alpha), \qquad \mathcal{M} = \mathcal{C} = \alpha^*.$$

- KGen：從 $\mathcal{K}$ 均勻抽出 sk（一個排列）。
- $\mathsf{Enc}(\mathsf{sk}, m)$：拆 $m = m_1 \cdots m_\ell$；對 $i \in [\ell]$ 令 $c_i = \mathsf{sk}(m_i)$；回傳 $c = c_1 \cdots c_\ell$。
- $\mathsf{Dec}(\mathsf{sk}, c)$：拆 $c = c_1 \cdots c_\ell$；對 $i \in [\ell]$ 令 $m_i = \mathsf{sk}^{-1}(c_i)$；回傳 $m = m_1 \cdots m_\ell$。

正確性：$\mathsf{sk}^{-1}(\mathsf{sk}(x)) = x$，因為 sk 是雙射所以反函數存在且唯一。

金鑰最方便的寫法是兩列表（講義 p. 9）：上列是 α 的自然順序，下列是它們的像；加密「由

上往下」查，解密「由下往上」查——這就是 π^{-1}。

明文	a b c d e f g h i j k l m n o p q r s t u v w x y z
密文	f b z　　⋯　　a v

動手做：以一把具體金鑰加解密

用 §3.3 的 Fisher–Yates 程式（種子 2025）抽出一把金鑰：

明文	a b c d e f g h i j k l m n o p q r s t u v w x y z
密文	x i w v e s j k n d z t b o g y h m l a q f p u c r

則 Enc(sk, the quick brown fox jumps over the lazy dog) = ake hqnwz imgpo sgu dqbyl gfem ake txrc vgj。解密時查下列找上列：a ↦ t、k ↦ h、e ↦ e（e 恰好是這把金鑰的不動點）……注意兩個 the 都變成 ake：同一個明文片段永遠給同一個密文片段。

3.2 討論：金鑰空間的大小（講義 p. 11）

Caesar 的 26 把金鑰每一把都是一個排列（位移），所以 Caesar 的金鑰空間是 Perm(α) 的子集；替換密碼只是「允許任意排列」。

$$|\mathcal{K}| = |\mathrm{Perm}(\alpha)| = |\alpha|!, \qquad 26! = 403{,}291{,}461{,}126{,}605{,}635{,}584{,}000{,}000 \approx 4.03 \times 10^{26} \approx 2^{88.4}.$$

（第一個字母有 26 種像，第二個剩 25 種……，乘法原理；Week 2 命題 4.1 (c)。）

2^{88} 有多大？

若每秒能試 10^{12} 把金鑰（一台很強的 GPU 叢集），試完 26! 把需要 4×10^{14} 秒 $\approx 1.3 \times 10^{7}$ 年。暴力搜尋確實不可行——但 §6 會看到，69 個字母的密文用頻率分析幾分鐘就能手動破解。「金鑰空間大」只封住了講義 p. 8 的第一個弱點。

3.3 討論：如何均勻抽出一個排列？（講義 p. 11）

這次不能只擲五枚硬幣了。標準做法是 **Fisher–Yates 洗牌**（Fisher & Yates 1938 的統計表；Durstenfeld 1964 給出就地版本，Knuth《The Art of Computer Programming》稱為 Algorithm P）：

```
Input : alphabet as an array A[0..25] = (a, b, ..., z)
for i = 25 down to 1:
    j <- uniform integer in {0, 1, ..., i}      # rejection sampling!
    swap A[i] and A[j]
Output: sk = the permutation  x_k -> A[k]        # A is now a uniform permutation
```

命題 3.1（Fisher–Yates 是均勻的） 若每一步的 j 都在 $\{0, \ldots, i\}$ 上均勻且各步獨立，則輸出的排列在 Perm(α) 上均勻。

證明. 演算法共做 25 次選擇，可能的選擇序列有 $26 \cdot 25 \cdots 2 = 26!$ 條，每條機率 $\prod_{i=1}^{25} \frac{1}{i+1} = \frac{1}{26!}$。不同的選擇序列給出不同的排列（第 i 步決定「哪個元素落在位置 i」，之後不再動它），而排列總數也是 $26!$，所以選擇序列與排列一一對應，每個排列機率恰為 $1/26!$。 ■

⚠ 三個會破壞均勻性的「捷徑」

1. **錯誤的洗牌**：「對每個位置 i，和整個陣列中隨機一個位置交換」。這給出 26^{26} 條選擇序列，而 $26! \nmid 26^{26}$，所以不可能每個排列等機率（Week 2 的鴿籠原理）。
2. **模偏差**：用 `rand() % (i+1)` 取 j。與 §2 相同的問題。
3. **關鍵字字母表**：許多教科書用一個關鍵字（如 `cryptogahbdef…`）生成排列。好記，但金鑰空間從 $26!$ 縮成「英文單字的數量」，而且敵手猜出前幾個對應就能推出其餘——這違反 (P1)。

```python
import random
def fisher_yates(seed=None):
    rng = random.Random(seed)            # demo only; use `secrets` for real keys
    A = list("abcdefghijklmnopqrstuvwxyz")
    for i in range(25, 0, -1):
        j = rng.randint(0, i)            # Python's randint is unbiased
        A[i], A[j] = A[j], A[i]
    return "".join(A)                    # image of a, b, ..., z in order

print(fisher_yates(2025))                # -> xiwvesjkndztbogyhmlaqfpucr
```

3.4 討論：潛在弱點（講義 p. 12）

弱點	說明
message, ciphertext pairs give *information* about sk	講義刻意把 Caesar 的「give sk」改成「give information about sk」：一對 (m_i, c_i) 只透露排列表的一格。要拿到整把金鑰，需要 26 個明文字母都出現過；但實務上拿到十幾格就足以讀懂其餘密文。
frequency analysis	每個明文字母仍被送到固定的密文字母，所以字母的頻率分布只是被重新貼標籤：最常出現的密文字母大概就是 e（§6）。
(more fundamentally) deterministic encryption	與 Caesar 完全相同：$m_0 =$ aa、$m_1 =$ ab 即可區分。確定性是結構性的缺陷，換再大的金鑰空間也救不了。

講義最後一句話是下一節的動機：「如果能在維持大金鑰空間的同時，讓頻率分析變得更困難呢 ?」——答案（Vigenère）會告訴我們，「更困難」和「安全」之間還差得很遠。

單一字母的替換密碼其實是「完美」的

若把訊息空間限制成單一字母 $\mathcal{M} = \alpha$，則對均勻的 $\mathsf{sk} \in \mathrm{Perm}(\alpha)$，無論 m 是什麼，$c = \mathsf{sk}(m)$ 在 α 上都是均勻分布（每個 c 恰有 25! 把金鑰把 m 送到它）。依 §7 的定義，這是完美保密的（KL-IMC 習題 2.9）！Caesar 限制成單一字母也一樣（Shannon 定理：$|\mathcal{K}| = |\mathcal{M}| = |\mathcal{C}| = 26$，每對 (m, c) 恰有一把金鑰）。所以三個歷史密碼的問題都不在「一個字母怎麼加密」，而在**同一把金鑰被用在許多字母上**——金鑰重用，正是本週反覆出現的主題。

從 al-Kindī 到福爾摩斯

- 約 9 世紀，巴格達的博學者 al-Kindī（約 801–873）在《解讀密文手稿》（*Risāla fī Istikhrāj al-Muʿammā*）中描述了頻率分析——已知最早的破譯方法，比 Caesar 晚了九百年、比下一個突破（Kasiski）早了一千年。
- 1586 年，蘇格蘭女王 Mary 與 Babington 密謀的信件使用替換密碼加上代碼字（nomenclator），被 Walsingham 的密碼專家 Phelippes 破譯，成為她被處決的證據。
- 文學裡的替換密碼：Poe〈金甲蟲〉（1843）、Conan Doyle〈跳舞的人〉（1903）都以頻率分析為情節核心——主角第一步都是「最多的符號是 e」。

4 Vigenère 密碼（講義 pp. 13–15）

4.1 正式定義（講義 p. 13）

想法：不再對每個字母用同一個位移，而是依一把**重複使用的金鑰字串**輪流使用不同的位移。

Vigenère 密碼

令 $\alpha = \{\mathsf{a}, \ldots, \mathsf{z}\} \leftrightarrow \mathbb{Z}/26\mathbb{Z}$，$\mathcal{M} = \mathcal{C} = \alpha^*$，並固定正整數 n（金鑰長度上限）。

$$\mathcal{K} = \{0, 1, \ldots, 25\}^{\le n} = \mathbb{Z}/26\mathbb{Z} \cup (\mathbb{Z}/26\mathbb{Z})^2 \cup \cdots \cup (\mathbb{Z}/26\mathbb{Z})^n.$$

- KGen：從 $\mathcal{K}$ 均勻抽出 sk。
- $\mathrm{Enc}(\mathsf{sk}, m)$：拆 $m = m_1 \cdots m_\ell$ 與 $\mathsf{sk} = \mathsf{sk}_1 \cdots \mathsf{sk}_j$；對 $i \in [\ell]$ 令

$$c_i = m_i + \mathsf{sk}_{((i-1) \bmod j)+1} \pmod{26},$$

  回傳 $c = c_1 \cdots c_\ell$。
- $\mathrm{Dec}(\mathsf{sk}, c)$：同樣拆解；對 $i \in [\ell]$ 令 $m_i = c_i - \mathsf{sk}_{((i-1) \bmod j)+1} \pmod{26}$。

（Problem Set 3 允許假設 sk 的長度恰為 n，即 $\mathcal{K} = (\mathbb{Z}/26\mathbb{Z})^n$。）

⚠ 索引的小陷阱

講義寫 $c_i = m_i + \mathsf{sk}_{i \bmod j}$。若 sk 的下標從 1 開始，當 $j \mid i$ 時 $i \bmod j = 0$ 會指到不存在的 sk_0；正確的意思是「第 i 個字母用第 $((i-1) \bmod j)+1$ 個金鑰字母」（或把金鑰下標改成從 0 開始：$\mathsf{sk}_0, \ldots, \mathsf{sk}_{j-1}$，$c_i = m_i + \mathsf{sk}_{(i-1) \bmod j}$）。寫程式時請用 0 起始的下標。

4.2 討論：j 個交錯的 Caesar 密碼（講義 p. 14）

講義的例子 $n = 5$、$j = 4$、$\ell = 7$：

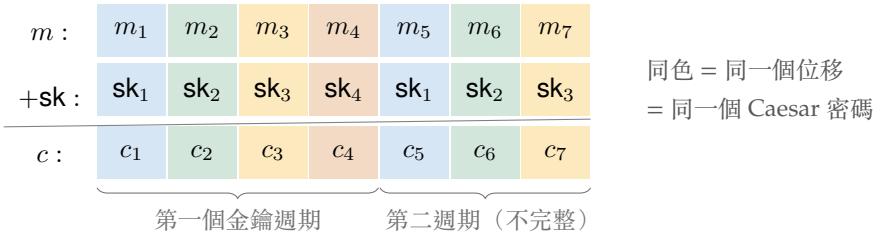


圖 1　金鑰長度 $j = 4$ 的 Vigenère 加密。位置 $1, 5, 9, \ldots$ 用 sk_1，位置 $2, 6, 10, \ldots$ 用 sk_2……：整段密文是四個「交錯」的 Caesar 密碼。

這個觀察（講義：「four 'distinct' Caesar ciphers」）就是全部破譯方法的基礎：只要知道 j，把密文依位置 mod$j$ 分成 j 條子密文，每條都是普通的 Caesar，各自用頻率分析解決（§6）。攻擊成本是 $26 \cdot j$，不是 26^j——金鑰可以被「分而治之」地逐段攻擊，這是金鑰空間大小不等於安全性的又一個例子。

金鑰空間大小。 $|\mathcal{K}| = 26 + 26^2 + \cdots + 26^n = \dfrac{26^{n+1} - 26}{25} \geq 26^n$。例如 $n = 3$：18,278 $\approx 2^{14.2}$；$n = 5$：約 $1.2 \times 10^7 \approx 2^{23.6}$；$n = 10$：約 2^{47}。一個有趣的副作用：長度恰為 n 的金鑰佔了 $\mathcal{K}$ 的 $26^n/|\mathcal{K}| \approx 25/26 \approx 96\%$，所以「從 $\mathcal{K}$ 均勻抽樣」幾乎總是抽到最長的金鑰——金鑰的長度並不均勻。

🔍 動手做：金鑰 `kcl`

$\mathsf{sk} = \texttt{kcl} = (10, 2, 11)$，$m = \texttt{kings college london}$（空白不加密、不消耗金鑰）：

m:	k	i	n	g	s	c	o	l	l	e	g	e	l	o	n	d	o	n
sk:	k	c	l	k	c	l	k	c	l	k	c	l	k	c	l	k	c	l
c:	u	k	y	q	u	n	y	n	w	o	i	p	v	q	y	n	q	y

例如第一個字母 k + k = 10 + 10 = 20 = u，第四個 g + k = 6 + 10 = 16 = q。注意明文 `ll` 變成 `nw`（不同位移），但第 1、4 個字母和第 13、16 個字母都用 sk_1——週期性仍然存在。

4.3 討論：潛在弱點（講義 p. 15）

弱點	說明
message, ciphertext pairs give information about sk	一對 (m, c) 的第 i 個位置給出 $\mathsf{sk}_{((i-1) \bmod j)+1} = c_i - m_i$；只要已知明文長度 $\geq j$，整把金鑰就出來了。
harder frequency analysis	同一個明文字母會依位置被送到 j 個不同的密文字母，整體頻率分布被「抹平」（重合指數下降）。但只要先找出 j（Kasiski 1863、Friedman 1920），分流之後每條子密文都是 Caesar。講義註腳：Babbage 早已知道，Kasiski 1863 明確發表。
(more fundamentally) deterministic encryption	仍是確定性、無狀態。IND-EAV 的區分對只要長一點：$m_0 = \mathsf{a}^{n+1}$、$m_1 = \mathsf{a}^n\mathsf{b}$（位置 1 與 $n+1$ 用同一個位移；Problem 1 (iii)）。

Vigenère 的唯一問題是「週期」

如果金鑰不重複——長度 $j \geq \ell$，而且只用一次——那麼 $c_i = m_i + \mathsf{sk}_i$ 每個位置都用一個獨立均勻的位移，這正是 §7 的完美保密方案（Vernam 式一次性密碼本的 $\mathbb{Z}/26\mathbb{Z}$ 版本）。Vigenère 的一切弱點都來自「金鑰比訊息短，所以必須重複」。Shannon 的定理（§7）會證明這不是巧合：想要完美保密，金鑰就必須至少和訊息一樣長。

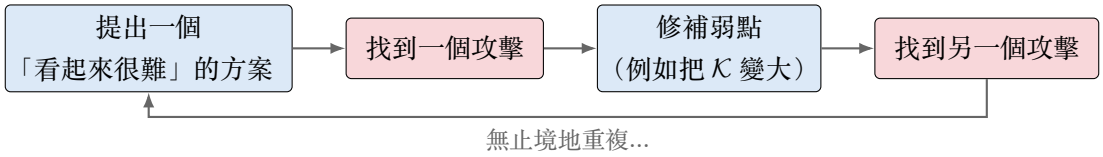
「不可破解的密碼」三百年

- **1553**：Giovan Battista Bellaso 在《La cifra del Sig. Giovan Battista Bellaso》中描述了用關鍵字選取位移的多字母表密碼——也就是今天叫「Vigenère」的方案。
- **1586**：Blaise de Vigenère 的《Traicté des chiffres》提出的其實是更強的自動金鑰（autokey）密碼；19 世紀的作者張冠李戴，把 Bellaso 的方案冠上他的名字。Kahn 在《The Codebreakers》中為此抱不平。
- **1854 前後**：Charles Babbage 私下破解了 Vigenère（回應 Thwaites 的挑戰），但從未發表——一說是因為他在克里米亞戰爭期間為英國海軍擔任密碼顧問。
- **1863**：普魯士軍官 Friedrich W. Kasiski 出版《Die Geheimschriften und die Dechiffrir-Kunst》，第一次公開「找重複片段、取距離的公因數」求金鑰長度的方法（講義 p. 15 的註腳）。
- **1868、1917**：Lewis Carroll 在童書專欄稱它「不可破解」；《Scientific American》1917 年仍稱它「impossible of translation」——這時 Kasiski 的書已經出版 54 年。它的綽號 *le chiffre indéchiffrable* 的真正意思是「沒有人發表過破法」，而非「不存在破法」。
- **1920/1922**：William F. Friedman 的《The Index of Coincidence and Its Applications in Cryptography》（Riverbank 第 22 號出版品）給出統計方法求金鑰長度，被譽為密碼分析史上最重要的單篇著作。

5　反思：從「設計—破解」循環到定義（講義 pp. 16–17）

5.1　三個密碼的共同模式（講義 p. 16）

回顧三節：每一個方案的生命週期都是



講義稱這是「不講原則的」（unprincipled）：

- **Caesar**：金鑰太少 ⇒ 暴力搜尋。「修補」：換成替換密碼。
- **替換密碼**：26! 把金鑰，但同一字母永遠映到同一字母 ⇒ 頻率分析。「修補」：Vigenère。
- **Vigenère**：頻率被抹平，但金鑰週期性 ⇒ Kasiski / Friedman 分流後逐段頻率分析。「修補」：…… ?

每一次「修補」都只針對上一個攻擊，沒有人問「我們到底想抵擋哪一類敵手 ?」。在這種循環裡，一個方案「安全」的唯一證據是「目前還沒有人破解它」——Vigenère 頂著「不可破解」的名號三百年，就是這種證據有多薄弱的最佳示範。

Week 2 的三大原則如何打破循環

Katz–Lindell 的現代密碼學三原則（Week 2 §1）正是對這種循環的回應：

1. **精確定義**：先說清楚「安全」是什麼（例如 IND-CPA）。這週會看到歷史上第一個這樣的定義——Shannon 的完美保密（§7）。
2. **明確假設**：若需要計算假設，必須清楚陳述（下週起：PRG、PRF）。本週的完美保密不需要任何假設。
3. **安全性證明**：證明「若假設成立，則方案滿足定義」。對完美保密，證明是無條件的、純機率論的。

有了定義，我們就不必等待下一個攻擊：一個被證明滿足 IND-CPA 的方案，任何符合模型的攻擊都被排除了。而且一旦定義是對的，「修補」也有方向：Vigenère 之所以失敗，是因為它連「兩個訊息不可區分」這麼弱的要求都達不到。

5.2　換位密碼不會改變結論（講義 p. 17）

講義 p. 17 提醒：本週只討論了替換（substitution）式的密碼——字母被換成別的字母，但位置不變。另一大類歷史密碼是**換位密碼**（transposition cipher）：字母本身不變，只是位置被打亂。

三個換位密碼

- **密碼棒（scytale，斯巴達，約西元前 5 世紀）**：把羊皮紙條纏在直徑固定的棒子上橫著寫，拆下來後字母順序就被打亂；金鑰是棒子的直徑（等價於「每列幾個字母」）。例：$m =$ `helpmeiamunderattack`（20 字母），以 4 列 × 5 行寫入、按行讀出：

h	e	l	p	m
e	i	a	m	u
n	d	e	r	a
t	t	a	c	k

$\longrightarrow$ $c =$ `hent eidt laea pmrc muak`

- **柵欄密碼（rail fence）**：金鑰是列數 r，明文沿鋸齒線寫在 r 列上，再逐列讀出。$r = 3$、$m =$ `wearediscoveredfleeatonce` 得 $c =$ `wecrlteerdsoeefeaocaivden`。
- **欄位換位（columnar transposition）**：金鑰是一個關鍵字，明文按列寫成矩陣，依關鍵字字母的字典序重排欄位後逐欄讀出。一戰德軍的 ÜBCHI、二戰 SOE 的雙重欄位換位都是它的變形。

為什麼換位不會改變我們的結論？

- **字母頻率完全不變**：密文只是明文的重排，e 還是最多；這恰好是辨認「這是一個換位密碼」的指紋（密文字母分布與英文幾乎相同，但讀不通）。攻擊者接著以 anagramming——依常見雙字母、三字母組合試著重組——還原順序。
- **同樣是確定性加密**：$m_0 =$ `aaaa`、$m_1 =$ `aaab`——換位後 m_0 的密文仍是 `aaaa`，m_1 的密文含一個 `b`，一次查詢就能區分。更直接地，$m_0 =$ `ab`$\cdots$ 與 $m_1 =$ `ba`$\cdots$ 的密文是同一多重集合，但只要密文與選定明文不同就知道發生了換位。
- **組合起來也一樣**：替換。換位（乘積密碼，product cipher）仍是確定性、無狀態的函數，Week 2 定理 6.1 依然適用。乘積密碼是現代分組密碼（DES、AES）的設計思想，但那些方案的安全性來自金鑰相依的、經過大量輪數的混淆與擴散，而且必須搭配隨機化的操作模式——不是「換位加替換」本身。

考試時如何「舉反例」

講義 p. 16 的核心訊息是：我們需要定義而非「換一個更複雜的方案」。遇到「某某古典密碼安全嗎？」這類問題，標準答案結構是：(1) 指出它是確定性加密，寫下 Week 2 的 IND-EAV（或 IND-CPA）遊戲；(2) 給出一對具體的 m_0, m_1 與敵手的判斷規則；(3) 計算敵手獲勝的機率（通常是 1，即優勢 $\frac{1}{2}$ 達到最大）。這比描述頻率分析更簡潔，也更「有原則」。頻率分析、Kasiski 等方法是更強的攻擊（直接還原金鑰、而且是唯密文攻擊），考試中若被要求「破解」才需要。

6　歷史密碼的密碼分析（講義 pp. 18–22）

6.1 先說清楚：以下方法「不是必要的」（講義 p. 18）

講義 p. 18 開宗明義：要斷定這些歷史密碼不安全，**完全不需要**暴力搜尋或頻率分析。四個理由，每個都是 Week 2 某個定義的直接應用：

講義的理由	對應的安全概念	本教材中的例子
可以挑同長度的 m_0, m_1，從 c_b 分辨出哪個被加密	IND-EAV（單次挑戰、無預言機）	Problem 1 (iii)：aa vs. ab
一對 (m, c) 就洩漏 sk 的資訊	已知明文攻擊（KPA）	Caesar：$\mathsf{sk} = c_1 - m_1$
能存取 $\mathsf{Enc}(\mathsf{sk}, \cdot)$ 就能直接取回 sk	IND-CPA 的預言機	Problem 1 (ii)：查詢 abc…z
它們都是確定性的	Week 2 定理 6.1	任何重複的明文片段都看得到

反過來說，**抵擋這些攻擊只是「最低門檻」**：一個方案要有希望達到 IND-EAV / CPA / CCA，至少不能被暴力搜尋或頻率分析打倒，但做到這點遠遠不夠。

攻擊模型回顧

模型	敵手拿到什麼	本週的對應
唯密文（COA）	只有密文（與訊息分布的先驗知識）	頻率分析、Kasiski、trace、Problem 3
已知明文（KPA）	若干 (m_i, c_i)	暴力搜尋的「檢查」步驟
選擇明文（CPA）	$\mathsf{Enc}(\mathsf{sk}, \cdot)$ 預言機	Problem 1 (ii)、Problem 2 (ii)
選擇密文（CCA）	另加 $\mathsf{Dec}(\mathsf{sk}, \cdot)$ 預言機	OTP 的可塑性（§8）

歷史密碼在最弱的 COA 下就倒了；現代定義要求在 CPA/CCA 下仍安全。

6.2 暴力搜尋（講義 p. 19）

暴力搜尋（brute force）就是窮舉某個空間——通常是 $\mathcal{K}$——直到找到「對的」金鑰。講義的虛擬碼：

```
Input : message, ciphertext pairs {(m_i, c_i)}_{i=1..N}
for k in K:
    for i in [N]:
        if Dec(k, c_i) != m_i:        # k is wrong
            try next k
    return k                           # k is consistent with every pair
```

兩個講義提出的問題：

「is wrong」是什麼意思？ 需要二選一：

- **已知明文對**：有 (m_i, c_i) 可以比對，如上方虛擬碼。這時只要密文夠長，錯的金鑰幾乎不可能通過所有檢查。
- **足夠的「語境」**：只有密文（COA）時，改成「Dec(k, c) 看起來像英文嗎？」。這需要關於訊息分布的先驗知識——以及夠長的密文，否則會有多把金鑰都給出合理的明文。

唯一解距離（unicity distance）：要多少密文才夠？

Shannon 定義 $U = H(\mathcal{K})/D$，其中 D 是語言的冗餘度（英文每字母約 $\log_2 26 - 1.5 \approx 3.2$ 位元）。粗略地說，密文長度超過 U 之後，「能解出合理英文」的錯誤金鑰期望數小於 1：

$$U_{\text{Caesar}} \approx \frac{4.7}{3.2} \approx 1.5, \qquad U_{\text{mono}} \approx \frac{88.4}{3.2} \approx 28, \qquad U_{\text{Vigenère},n} \approx \frac{4.7n}{3.2} \approx 1.5n.$$

Problem 3 的三段密文（46、69、180 個字母）都遠超過各自的 U，所以解是唯一的——這也是為什麼 (ii) 的 26 種位移只有一種讀得通。

為什麼用 Dec 而不是 Enc？

1. **唯密文攻擊時根本沒有 m_i 可以加密**：手上只有 c，唯一能做的是對每把 k 算 Dec(k, c)，再判斷結果是否「像訊息」。
2. **隨機化加密時 Enc 不會重現 c_i**：若 Enc 使用隨機數（Week 2 的 IND-CPA 方案都必須如此），Enc(k, m_i) 即使 k 正確也幾乎不會等於 c_i；但 Dec$(k, c_i) = m_i$ 對正確的 k 永遠成立（正確性）。
3. **Dec 能一次排除整類金鑰**：許多方案的 Dec 會先檢查格式 / 填充 / 標籤，不合法就立刻拒絕，比較便宜。

對本週的三個確定性密碼，兩者剛好等價（Enc$(k, m_i) = c_i \iff$ Dec$(k, c_i) = m_i$），但正確的習慣是寫 Dec。

Caesar 的暴力搜尋

Problem 3 (ii) 的密文 `w jtcm kwum …`：對 sk $= 0, 1, \ldots, 25$ 各算一次 Dec，只有 sk $= 8$ 給出英文（§11）。成本：26 次解密，每次 $O(\ell)$。對替換密碼則是 26! 次——不可行；但 §11 會示範頻率分析如何在幾十步內完成。

6.3　頻率分析（講義 pp. 20–21）

頻率分析的前提

講義 p. 20：假設 (1) 對訊息分布有先驗知識（例如：英文散文），且 (2) 加密沒有消除這個資訊。那麼比較密文的統計量與語言的統計量就是合法的攻擊。Caesar / 替換 / 換位密碼都滿足 (2)：它們把每個字母送到固定的字母（或固定的位置），統計結構只是被重新標籤。

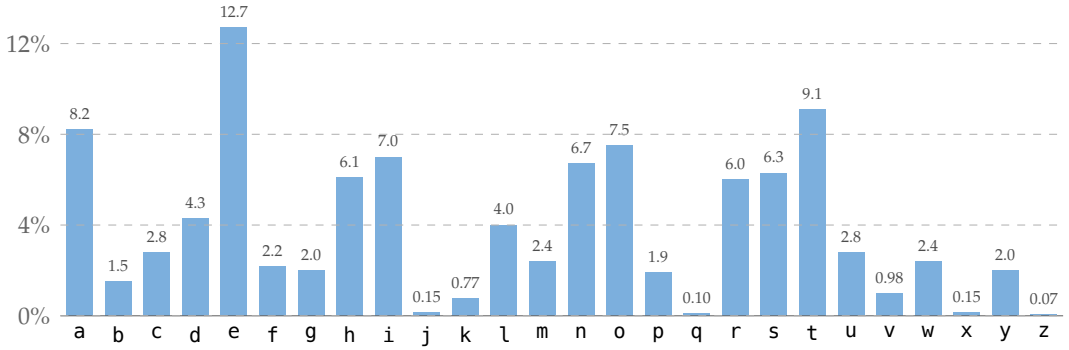


圖 2　英文字母的相對頻率（%，典型語料值；講義 p. 20 列出前五名 e 12.7、t 9.1、a 8.2、o 7.5、i 7.0）。

還有哪些統計量可用（講義 p. 20）？

- **雙字母**（bigram）：th、he、in、er、an、re、on、at、en、nd；**三字母**（trigram）：the、and、ing、her、ere、ent。
- **字首頻率**：英文單字以 t、a、o、i、s、w 開頭最多（t 約 16%）；**字尾**：e、s、d、t、n。
- **雙寫字母**：ll、ee、ss、oo、tt、ff。
- **單字母單字**：只有 a 和 i（偶爾 o）——Problem 3 (ii)、(iii) 都靠它。
- **單字長度與重複單字**：題目保留空白，等於多送了一份結構。

講義強調：「需要足夠但出奇地少的密文」——Problem 3 (iii) 只有 69 個字母。

用數字取代肉眼：重合指數與 χ^2

手動頻率分析要「看起來像英文」，寫程式則需要可計算的分數：

- **重合指數**（Friedman 1920）：從文本中隨機抽兩個不同位置，字母相同的機率

$$\mathrm{IC}(x) = \frac{\sum_{a\in\alpha} n_a(n_a-1)}{N(N-1)},$$

  其中 n_a 是字母 a 的出現次數、N 是總長度。英文約 0.0655；均勻隨機文字為 $1/26 \approx 0.0385$。Caesar / 替換 / 換位都保持 IC（重新標籤不改變 $\sum n_a(n_a-1)$），Vigenère 則把它壓向 0.0385——這正是判斷金鑰長度的依據（§11）。

- **χ^2 統計量**：對每個候選位移 k，把密文平移 $-k$ 後計算

$$\chi^2_k = \sum_{a\in\alpha} \frac{(n_a - Np_a)^2}{Np_a},$$

  p_a 是英文頻率。正確的 k 給出最小的 χ^2，而且通常遠小於其他 25 個——比「最多的字母是 e」穩健得多（§11 (iv) 的第二條子密文就是「最多的字母不是 e」的例子）。

- **內積法**：最大化 $\sum_a p_a \cdot n_{a+k}/N$，效果類似。

Vigenère 的頻率分析（講義 p. 21）。 給定 $c = c_1c_2\cdots c_\ell$ 與金鑰長度上限 n：

1. 對每個候選長度 $i \in [n]$，把密文分成 i 條子密文：

$$c_1c_{1+i}c_{1+2i}\cdots,\quad c_2c_{2+i}c_{2+2i}\cdots,\quad \ldots,\quad c_ic_{2i}c_{3i}\cdots$$

（講義寫成 $c_{i-1}c_{2i-1}c_{3i-1}\cdots$ 是最後一條的另一種寫法；共 i 條，每條是一個 Caesar 密碼）。

2. 對每條子密文做頻率分析求出位移；若「沒有一條讀得通」就換下一個 i。

兩個更快求 i 的方法：

- **Kasiski 檢驗（1863）**：密文中重複出現的三字母以上片段，多半來自「同一段明文碰到同一段金鑰」，因此它們的距離是金鑰長度的倍數；取所有距離的最大公因數。短密文時雜訊很多（§11 (iv) 中 180 個字母只有兩組重複三字母，距離 90 與 16，gcd $= 2$，是錯的 !）。
- **Friedman 的 IC 檢驗（1920）**：對每個 i 算 i 條子密文的平均 IC；正確的 i（及其倍數）會跳回 ≈ 0.065，其餘留在 ≈ 0.045。對短密文更可靠。

總成本：至多 n 個長度 $\times$ 每長度 $26 \cdot i$ 個位移 $= O(26n^2)$，與 26^n 天差地遠。

6.4 Bonus weakness：trace（講義 p. 22）

講義 p. 22 的標題是「Veni vidi explicavi」（我來、我見、我解釋）。Caesar 密碼「特別糟糕」：相鄰字母之間的間距（gap）完全不變。

trace

對字串 $w = w_1w_2\cdots w_\ell \in \alpha^\ell$（視為 $(\mathbb{Z}/26\mathbb{Z})^\ell$），定義

$$\mathrm{trace}(w) = (w_2 - w_1,\ w_3 - w_2,\ \ldots,\ w_\ell - w_{\ell-1}) \in (\mathbb{Z}/26\mathbb{Z})^{\ell-1}.$$

例：$\mathrm{trace}(\texttt{hello}) = (\texttt{e}-\texttt{h}, \texttt{l}-\texttt{e}, \texttt{l}-\texttt{l}, \texttt{o}-\texttt{l}) = (-3, 7, 0, 3) = (23, 7, 0, 3)$。

引理 6.1（Caesar 保持 trace） 對任意 $\mathsf{sk} \in \mathbb{Z}/26\mathbb{Z}$ 與 $m \in \alpha^\ell$，$\mathrm{trace}(\mathsf{Enc}(\mathsf{sk}, m)) = \mathrm{trace}(m)$。

證明. $c_{i+1} - c_i = (m_{i+1} + \mathsf{sk}) - (m_i + \mathsf{sk}) = m_{i+1} - m_i$，金鑰互相抵消。 ■

講義的例子：$\mathsf{sk} = 1$，$\mathsf{Enc}(1, \texttt{hello}) = \texttt{ifmmp}$，$\mathrm{trace}(\texttt{ifmmp}) = (\texttt{f}-\texttt{i}, \texttt{m}-\texttt{f}, \texttt{m}-\texttt{m}, \texttt{p}-\texttt{m}) = (23, 7, 0, 3)$ ✓。

命題 6.2（trace 刻畫了 Caesar 的所有可能明文） 密文 c 有可能解密成 m（即存在 sk 使 $\mathsf{Dec}(\mathsf{sk}, c) = m$）若且唯若 $\mathrm{trace}(c) = \mathrm{trace}(m)$（且長度相同）。

證明. ($\Rightarrow$) 由引理。($\Leftarrow$) 設 $\mathrm{trace}(c) = \mathrm{trace}(m)$。令 $\mathsf{sk} := c_1 - m_1$。歸納證明 $c_i = m_i + \mathsf{sk}$ 對所有 i 成立：$i = 1$ 由定義；若 $c_i = m_i + \mathsf{sk}$，則 $c_{i+1} = c_i + (c_{i+1} - c_i) = m_i + \mathsf{sk} + (m_{i+1} - m_i) = m_{i+1} + \mathsf{sk}$，其中第二個等號用了 trace 相等。故 $\mathsf{Dec}(\mathsf{sk}, c) = m$。 ■

如何用 trace 破解

把常見單字的 trace 背起來或列表：

單字	trace	單字	trace
the	(14, 23)	and	(13, 16)
hello	(23, 7, 0, 3)	yes	(6, 14)
no	(1)	you	(16, 6)
blue	(10, 9, 10)	come	(12, 24, 18)

在密文中找到長度相同、trace 相同的單字，$\mathsf{sk} = c_1 - m_1$ 就出來了。Problem 3 (ii) 的 `jtcm` 的 trace 是 $(10, 9, 10)$——與 `blue` 相同；$\mathsf{sk} = \mathtt{j} - \mathtt{b} = 9 - 1 = 8$。這比暴力搜尋更快：不必嘗試 26 把金鑰，只需比對一次。

trace 的推廣

- **Vigenère（金鑰長度 j）**：相鄰字母的差不再守恆，但相距 j 的字母的差守恆：$c_{i+j} - c_i = m_{i+j} - m_i$。這是 Kasiski 檢驗的本質——同一段明文在相距 j 的倍數處會給出相同密文。
- **替換密碼**：差值不守恆，但「哪些位置的字母相同」這個模式守恆：`that`（模式 ABCA）加密後仍是 ABCA。Problem 3 (iii) 中 `dtroqzgk` 與 `zit` 的重複就是這種「同構」（isomorph）資訊。
- **OTP**：$c_{i+1} - c_i = m_{i+1} - m_i + (\mathsf{sk}_{i+1} - \mathsf{sk}_i)$，而 $\mathsf{sk}_{i+1} - \mathsf{sk}_i$ 本身均勻且獨立於訊息——什麼都不守恆。這就是 §7 的完美保密。

考試重點

- 能寫出暴力搜尋虛擬碼，並說明為何用 Dec（唯密文時沒有 m；隨機化加密時 Enc 不重現 c）。
- 能解釋「is wrong」需要先驗知識或明文對，以及為何密文要夠長（唯一解距離）。
- 頻率分析的兩個前提；Vigenère 要先分流再逐條分析；知道 Kasiski 與 IC 的原理。
- trace 的定義、守恆引理、「若且唯若」命題的證明（關鍵：令 $\mathsf{sk} = c_1 - m_1$ 再歸納）。
- 最重要的：這些攻擊都不是證明不安全的必要手段——IND-EAV 的一對 m_0, m_1 就夠了。

7 完美保密（講義 pp. 23–26）

7.1 直覺（講義 p. 23）

歷史密碼的問題是密文「洩漏了關於訊息的統計資訊」。能不能做到一點都不洩漏？講義 p. 23 的答案是可以：

直覺上：密文不含任何關於訊息的資訊（長度除外）。一點也沒有！
更精確地說：任何同長度的訊息都可能是被加密的那一個，而且每一個都一樣可能。

這是 Shannon 在 1949 年〈Communication Theory of Secrecy Systems〉中提出的**完美保密**（perfect secrecy），也是史上第一個嚴格的密碼學安全定義——正是 §5 說的「先有定義」。講義隨即潑冷水：「那我們為什麼不直接回家？」因為完美保密有不可避免而且嚴重的實際限制（§7.4）。

7.2 正式定義

令 M 為訊息空間 $\mathcal{M}$ 上的隨機變數（任意分布，代表敵手對訊息的先驗知識），K 為 KGen 輸出的金鑰（與 M 獨立），$C = \mathsf{Enc}(K, M)$。

完美保密（Shannon 1949；KL-IMC Def. 2.3）

SKE Π 在訊息空間 $\mathcal{M}$ 上是**完美保密**的，若對 $\mathcal{M}$ 上的每個機率分布、每個 $m \in \mathcal{M}$、每個使 $\Pr[C = c] > 0$ 的 $c \in \mathcal{C}$，

$$\Pr[M = m \mid C = c] = \Pr[M = m].$$

白話：看到密文之後，敵手對訊息的信念（後驗）與看之前（先驗）完全相同。注意定義對敵手的計算能力**沒有任何限制**——這是資訊論式的安全。

引理 7.1（等價刻畫（KL-IMC Lemma 2.4）） Π 完美保密 $\iff$ 對所有 $m, m' \in \mathcal{M}$ 與 $c \in \mathcal{C}$：

$$\Pr_K\big[\mathsf{Enc}(K, m) = c\big] = \Pr_K\big[\mathsf{Enc}(K, m') = c\big].$$

證明概要.（$\Leftarrow$）若密文分布不依賴訊息，則 $\Pr[C = c \mid M = m] = \Pr[C = c]$，由 Bayes 得 $\Pr[M = m \mid C = c] = \Pr[M = m]$。（$\Rightarrow$）取 $\mathcal{M}$ 上在 $\{m, m'\}$ 各 $1/2$ 的分布；由定義與 Bayes 推得 $\Pr[C = c \mid M = m] = \Pr[C = c] = \Pr[C = c \mid M = m']$。 ■

這個刻畫最好用：「**對每個固定的 m，$\mathsf{Enc}(K, m)$ 的分布都一樣**」。證明完美保密時幾乎都走這條路。

完美不可區分（KL-IMC Def. 2.5）

考慮 Week 2 的 IND-EAV 遊戲 $\mathsf{PrivK}^{\mathrm{eav}}_{\mathcal{A},\Pi}$，但允許 $\mathcal{A}$ 的計算能力無限。Π 是完美不可區分的，若對每個（不受限的）$\mathcal{A}$：$\Pr[\mathsf{PrivK}^{\mathrm{eav}}_{\mathcal{A},\Pi} = 1] = \frac{1}{2}$，即優勢恰為 0。

KL-IMC Lemma 2.6：完美保密 $\iff$ 完美不可區分。所以完美保密就是 Week 2 的 IND-EAV 去掉兩個「讓步」：敵手不再受多項式時間限制、優勢不再是「可忽略」而是「恰好 0」。

7.3　講義的例子：$\mathbb{Z}/26\mathbb{Z}$ 上的一次性密碼本（講義 pp. 24–25）

$\mathbb{Z}/26\mathbb{Z}$ 上的一次性金鑰方案

令 $\mathcal{K} = \{0, \dots, 25\}^n$，$\mathcal{M} = \mathcal{C} = \{\mathsf{a}, \dots, \mathsf{z}\}^{\leq n}$，$n$ 為正整數。

- KGen：從 $\mathcal{K}$ 均勻抽出 $\mathsf{sk} = \mathsf{sk}_1 \cdots \mathsf{sk}_n$。
- $\mathsf{Enc}(\mathsf{sk}, m)$：拆 $m = m_1 \cdots m_\ell$；對 $i \in [\ell]$ 令 $c_i = m_i + \mathsf{sk}_i$；回傳 $c = c_1 \cdots c_\ell$。
- $\mathsf{Dec}(\mathsf{sk}, c)$：拆 $c = c_1 \cdots c_\ell$；對 $i \in [\ell]$ 令 $m_i = c_i - \mathsf{sk}_i$；回傳 m。

與 Vigenère 唯一的差別：下標是 sk_i 而非 $\mathsf{sk}_{i \bmod j}$——**金鑰的每個元素只用一次**，因此 sk 至少與 m 一樣長。

講義 p. 25：$c = \mathsf{aaaaaaa}$ 是誰的密文？

取 $n = 7$。每一個 7 字母字串都是可能的明文，而且對每個明文恰有一把金鑰把它送到 aaaaaaa（$\mathsf{sk}_i = \mathsf{a} - m_i = -m_i$）：

明文 m	數字	唯一的 $\mathsf{sk} = -m$	sk（字母）
leopard	$(11, 4, 14, 15, 0, 17, 3)$	$(15, 22, 12, 11, 0, 9, 23)$	pwmlajx
caracal	$(2, 0, 17, 0, 2, 0, 11)$	$(24, 0, 9, 0, 24, 0, 15)$	yajayap
lioness	$(11, 8, 14, 13, 4, 18, 18)$	$(15, 18, 12, 13, 22, 8, 8)$	psmnwii
cheetah	$(2, 7, 4, 4, 19, 0, 7)$	$(24, 19, 22, 22, 7, 0, 19)$	ytwwhat
panther	$(15, 0, 13, 19, 7, 4, 17)$	$(11, 0, 13, 7, 19, 22, 9)$	lanhtwj

講義的檢查：$\mathsf{Enc}(\mathtt{yajayap}, \mathtt{caracal})$：$2 + 24 = 26 \equiv 0$、$0 + 0 = 0$、$17 + 9 = 26 \equiv 0 \cdots\cdots = \mathsf{aaaaaaa}$ ✓。由於每把金鑰被抽到的機率相同（26^{-7}），每個明文都「一樣可能」：敵手看到 aaaaaaa 之後，對「是 leopard 還是 panther」的信念與沒看到之前一模一樣。

定理 7.2（$\mathbb{Z}/26\mathbb{Z}$ 一次性方案是完美保密的）　對任意固定長度 $\ell \leq n$，上述方案在 $\mathcal{M}_\ell = \alpha^\ell$ 上完美保密。

證明. 用引理的刻畫。固定 $m, c \in \alpha^\ell$。$\mathsf{Enc}(K, m) = c$ 若且唯若 $K_i = c_i - m_i$ 對所有 $i \in [\ell]$ 成立。因為 $K_1, \dots, K_\ell$ 獨立且在 $\mathbb{Z}/26\mathbb{Z}$ 上均勻，

$$\Pr_K[\mathsf{Enc}(K, m) = c] = \prod_{i=1}^{\ell} \Pr[K_i = c_i - m_i] = 26^{-\ell},$$

與 m 無關。故對所有 m, m'：$\Pr[\mathsf{Enc}(K, m) = c] = \Pr[\mathsf{Enc}(K, m') = c]$。（$K_{\ell+1}, \dots, K_n$ 沒被用到，不影響。） ■

長度會洩漏

若 $\mathcal{M}$ 包含不同長度的訊息，定義就不成立：看到 7 個字母的密文，敵手就知道 m 不是 5 個字母。這是講義「other than length」的意思，也是為什麼 Week 2 的 IND 遊戲要求

$|m_0| = |m_1|$。實務上可用填充（padding）到固定長度來處理。

7.4 限制（講義 p. 26）

講義 p. 26 說：這個方案的強項是「對任何明文與密文，恰有一把金鑰連接它們」；但這同時是它的弱點。

定理 7.3（金鑰至少和訊息空間一樣大（KL-IMC Thm. 2.10）） 若 Π 在 $\mathcal{M}$ 上完美保密，則 $|\mathcal{K}| \geq |\mathcal{M}|$。

證明. 取 $\mathcal{M}$ 上的均勻分布，任取一個 $\Pr[C = c] > 0$ 的 c。令 $\mathcal{M}(c) = \{\mathsf{Dec}(k, c) : k \in \mathcal{K}\}$ 為「c 可能解密成的訊息」集合，顯然 $|\mathcal{M}(c)| \leq |\mathcal{K}|$。若 $|\mathcal{K}| < |\mathcal{M}|$，則存在 $m' \in \mathcal{M} \setminus \mathcal{M}(c)$。對這個 m'：沒有金鑰能把 m' 加密成 c（否則由正確性 $\mathsf{Dec}(k, c) = m'$），所以 $\Pr[M = m' \mid C = c] = 0 \neq \frac{1}{|\mathcal{M}|} = \Pr[M = m']$，違反完美保密。 ■

講義的句子「need at least as much key as message」就是這個定理：想加密 ℓ 個字母的任意訊息，至少需要 26^ℓ 把金鑰，即 ℓ 個均勻的金鑰字母。

定理 7.4（Shannon 定理（KL-IMC Thm. 2.11）） 設 $|\mathcal{K}| = |\mathcal{M}| = |\mathcal{C}|$。則 Π 完美保密 $\iff$ (1) KGen 以機率 $1/|\mathcal{K}|$ 選每把金鑰，且 (2) 對每對 (m, c) 恰有一把 k 使 $\mathsf{Enc}(k, m) = c$。

這給了一個「驗收」完美保密的檢查表，也解釋了講義 p. 25 為何強調「唯一的 sk」與「每把 sk 等機率」兩件事都要成立。Problem 2 (v) 會用到：拿掉全零金鑰會破壞 (1)。

⚠ 金鑰重用：講義 p. 26 的計算

若用同一把 sk 加密 m_1, m_2 得 c_1, c_2，則

$$c_2 - c_1 = (m_2 + \mathsf{sk}) - (m_1 + \mathsf{sk}) = m_2 - m_1.$$

敵手不需知道 sk 就拿到兩個訊息的差——對英文這幾乎等於拿到兩個訊息（§8 的 crib dragging）。Vigenère 就是「同一把短金鑰反覆使用」的極端：每隔 j 個字母就重用一次。結論：**單次使用的金鑰，而且至少和訊息一樣長**（講義註腳：可嚴格證明，見 KL-IMC §2.3、§2.4）。

完美保密 vs. Week 2 的定義

- 完美保密 $\Rightarrow$ IND-EAV（優勢恰為 0，對無限算力的敵手）。這比 Week 2 要求的更強。
- 完美保密 $\not\Rightarrow$ IND-CPA：一次性方案是確定性的，同一把金鑰用兩次就被區分（Week 2 定理 6.1）。這不矛盾——完美保密只承諾一則訊息。多則訊息的完美保密需要每則訊息各自的新鮮金鑰（KL-IMC §2.2 的討論）。
- 完美保密只談保密，不談完整性：翻轉 c_i 的一個位元就翻轉 m_i（可塑性），這在 Week

1 的 MAC 討論中已經出現。

8　一次性密碼本（講義 pp. 27–29）

8.1　定義（講義 p. 27）

把上一節的方案從 $\mathbb{Z}/26\mathbb{Z}$ 換成 $\mathbb{Z}/2\mathbb{Z} = \{0,1\}$，加法變成 XOR，就是**一次性密碼本**（one-time pad, OTP）。講義的標題寫 Miller–Vernam–Mauborgne，是它的三位「發明人」（§8.3）。

一次性密碼本

令 $\mathcal{K} = \{0,1\}^n$，$\mathcal{M} = \mathcal{C} = \{0,1\}^{\le n}$。

- KGen：從 $\{0,1\}^n$ 均勻抽出 sk（擲 n 枚公平硬幣）。
- $\mathsf{Enc}(\mathsf{sk}, m)$：若 $m = m_1 \cdots m_\ell$，回傳 $c = m \oplus \mathsf{sk}_1 \cdots \mathsf{sk}_\ell$（逐位元 XOR 金鑰的前 ℓ 位）。
- $\mathsf{Dec}(\mathsf{sk}, c)$：若 $c = c_1 \cdots c_\ell$，回傳 $m = c \oplus \mathsf{sk}_1 \cdots \mathsf{sk}_\ell$。

正確性：$(m \oplus k) \oplus k = m \oplus (k \oplus k) = m \oplus 0^\ell = m$，依序用了結合律、XOR 的自反性、單位元——§1 的群性質。加密與解密是同一個函數。

封面的例子

$m =$ `01000101`（ASCII 的 `E`），$\mathsf{sk} =$ `01110110`：

$m =$	0	1	0	0	0	1	0	1
$\mathsf{sk} =$	0	1	1	1	0	1	1	0
$c = m \oplus \mathsf{sk} =$	0	0	1	1	0	0	1	1

解密：$c \oplus \mathsf{sk} =$ `00110011` $\oplus$ `01110110` $=$ `01000101` $= m$ ✓。

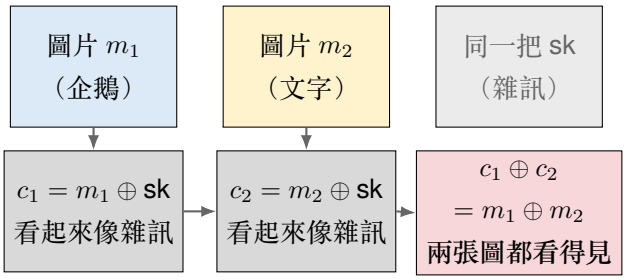
定理 8.1（OTP 是完美保密的（KL-IMC Thm. 2.9））　對任意固定長度 $\ell \le n$，OTP 在 $\{0,1\}^\ell$ 上完美保密。

證明. 固定 $m, c \in \{0,1\}^\ell$。$\mathsf{Enc}(K, m) = c \iff K_1 \cdots K_\ell = m \oplus c$，而 K 的前 ℓ 位在 $\{0,1\}^\ell$ 上均勻，所以 $\Pr_K[\mathsf{Enc}(K,m) = c] = 2^{-\ell}$，與 m 無關；由 KL-IMC Lemma 2.4 得證。 ■

等價的「遊戲」說法：在 IND-EAV 遊戲中，不論 b 為何，挑戰密文 $c = m_b \oplus K$ 都是 $\{0,1\}^\ell$ 上的均勻隨機字串，與 b 獨立；任何敵手（再聰明、算力再大）猜中 b 的機率都恰好是 $\frac{1}{2}$。

8.2　千萬不要重用金鑰（講義 p. 28）

講義 p. 28 的整頁只有一張圖（來源：davidlowryduda 與 crypto.stackexchange）：



單獨看，c_1 和 c_2 都是完美的均勻雜訊；但

$$c_1 \oplus c_2 = (m_1 \oplus \mathsf{sk}) \oplus (m_2 \oplus \mathsf{sk}) = m_1 \oplus m_2,$$

金鑰完全消失，兩張圖片疊在一起清晰可見。圖片的例子特別有說服力，因為人眼本身就是極好的「訊息分布先驗」。

⚠ 重用金鑰之後敵手能做什麼

1. **已知其中一則**：$m_2 = c_1 \oplus c_2 \oplus m_1$——另一則立刻全部暴露。
2. **兩則都不知道（crib dragging）**：猜 m_1 的某一小段是常見字（如 `"the "`），把它 XOR 到 $m_1 \oplus m_2$ 的每個位置上；若某個位置出現可讀的英文，猜測大概正確，而且得到 m_2 的一段，再回頭推 m_1 的下一段……如此交替「拖行」直到兩則全部還原。VENONA 計畫（§10）本質上就是對數千對重用的密碼本做這件事。
3. **IND-CPA 攻擊**：敵手查詢 $\mathsf{Enc}(\mathsf{sk}, 0^\ell) = \mathsf{sk}_1 \cdots \mathsf{sk}_\ell$，直接拿到金鑰（Problem 2 (ii)）。或用 Week 2 的兩次查詢：先查 m_0 得 c'，挑戰 (m_0, m_1) 得 c；$c = c' \iff b = 0$。
4. **可塑性**：不必知道金鑰也能把 c 改成 $c \oplus \Delta$，解密成 $m \oplus \Delta$——例如把匯款金額的某幾位翻轉。OTP 對完整性毫無保障。
5. **「幾乎」重用也不行**：若金鑰不是真均勻（例如由打字員「隨手亂打」、或由確定性機器產生），統計偏差就會洩漏訊息；二戰德國外交部的 GEE 系統用機器產生的「偽隨機」密碼本，1944 年被美方破解。

8.3 歷史：Miller、Vernam、Mauborgne（與 Shannon、Kotelnikov）

🏛 一次性密碼本是誰發明的？

- **1882**：加州銀行家 Frank Miller 出版《Telegraphic Code to Insure Privacy and Secrecy in the Transmission of Telegrams》，描述用「只用一次、用後銷毀」的隨機位移數列加密電報代碼——OTP 的全部要素。此事 2011 年才由 Steven Bellovin 重新發現（*Cryptologia* 35(3)）。
- **1917–1919**：AT&T 工程師 Gilbert Vernam 發明用打孔紙帶把金鑰與明文的 Baudot 碼逐位元 XOR 的電傳打字機加密（美國專利 1,310,719，1918 年 9 月 13 日申請、1919 年 7 月 22 日核准）。Vernam 原本的金鑰帶是會循環的。
- **約 1918–1920**：美國陸軍的 Joseph Mauborgne 指出金鑰必須真隨機且永不重複，否則

可破（他 1914 年便出版過密碼分析小冊，深知重複金鑰的弱點；Parker Hitt 1916 年的《Manual for the Solution of Military Ciphers》也系統化了這類攻擊）。「Vernam 的機器 + Mauborgne 的要求」= 一次性密碼本。

- **1941**：蘇聯的 V. A. Kotelnikov 在 1941 年 6 月 19 日的機密報告中獨立證明了 OTP 的不可破性。
- **1945/1949**：Claude Shannon 於 1945 年的機密報告（1949 年公開發表為〈Communication Theory of Secrecy Systems〉）嚴格定義完美保密、證明 OTP 達成它，並證明任何完美保密方案的金鑰都至少和訊息一樣長——即 §7 的兩個定理。

講義投影片把 Mauborgne 拼成「Maugborgne」，是筆誤。

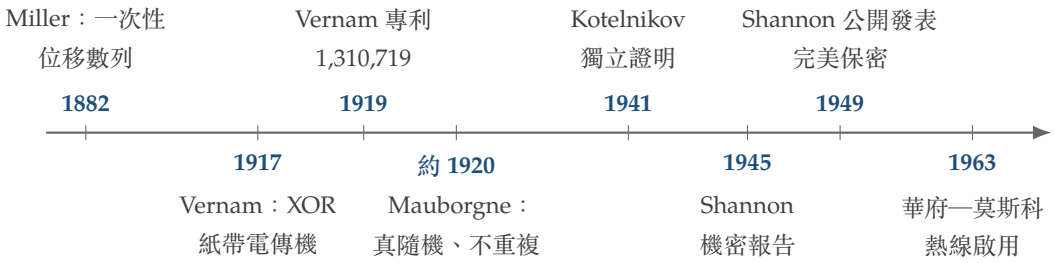


圖 3　一次性密碼本的時間線。

8.4　反思：完美但脆弱（講義 p. 29）

講義 p. 29 的三個要點：

1. 完美保密做得到，但「脆弱」。 它的安全性證明建立在三個全部必須成立的條件上：金鑰真正均勻、至少與訊息一樣長、絕不重用。違反任何一條，不只是「安全性下降一點」，而是像 VENONA 那樣整批崩潰。

2. 邊緣用例：華府—莫斯科「紅色電話」。 訊息量極小、只有兩個通訊方、而且雙方都有外交管道可以安全遞送成卷的金鑰紙帶——一般通訊都不具備這些條件。

🏛 熱線的真相

- 古巴飛彈危機（1962 年 10 月）期間，外交訊息的傳遞與解碼耗時數小時，促成 1963 年 6 月 20 日美蘇在日內瓦簽署備忘錄；熱線於 1963 年 8 月 30 日啟用。
- 它從來不是電話，而是**電傳打字機線路**（經倫敦、哥本哈根、斯德哥爾摩、赫爾辛基到莫斯科的有線電報，另有一條無線電備援）。
- 加密設備是挪威 Standard Telefon og Kabelfabrik 製造的 ETCRRM II（Electronic Teleprinter Cryptographic Regenerative Repeater Mixer）——一台 Vernam 式一次性

紙帶混合器。雙方各自產生自己方向的金鑰紙帶，透過大使館交換。

- 1986 年升級為傳真、2008 年改為加密電子郵件鏈路。

3. 金鑰分發是密碼學的難題。 OTP 需要「和所有未來訊息加起來一樣長」的金鑰，並且要事先安全地送到對方手上。但如果有辦法安全地送這麼長的金鑰，為什麼不直接送訊息？我們真正想要的是短金鑰、長訊息——與完美保密所允許的恰恰相反。Shannon 的定理說這在資訊論意義下不可能；出路是放寬定義：

- 只要求對多項式時間的敵手安全（Week 2 的 IND 定義），
- 並允許可忽略的優勢。

下週起的主角——偽隨機生成器（PRG）——正是用一把短金鑰「偽造」一條長得像均勻的金鑰流，把 OTP 變成串流密碼。它的安全性不再是無條件的，而是依賴一個明確陳述的計算假設——這正是 §5 描述的現代路線。

考試重點

- 能完整寫出 OTP 的 $(\mathcal{K}, \mathcal{M}, \mathcal{C}, \mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$，用群性質證明正確性，並證明完美保密 $(\Pr[\mathsf{Enc}(K, m) = c] = 2^{-\ell})$。
- 金鑰重用：推導 $c_1 \oplus c_2 = m_1 \oplus m_2$，並說明 crib dragging 與已知明文的後果。
- $|\mathcal{K}| \geq |\mathcal{M}|$ 的證明思路（$\mathcal{M}(c)$ 的大小論證）。
- 三個限制（長金鑰、單次使用、金鑰分發）及它們為何「不可避免」。
- 歷史：Miller / Vernam / Mauborgne、Shannon 1949、VENONA、熱線。

第二部分

Problem Set 3 詳解

Problem 1：歷史密碼的不安全性 · Problem 2：一次性密碼本 · Problem 3：實作與破譯三段電影台詞

9 Problem 1：On the insufficient security of historical ciphers

題目背景

Let HISTORICAL-CIPHERS $=$ {Caesar, monoalphabetic substitution, Vigenère}. In each question below you may assume sk for Vigenère has length n, a positive integer.
In this question we will examine ways in which historical examples of SKE fail to achieve security definitions demanded of contemporary cryptography *without requiring sometimes tedious cryptanalysis*.

題目的最後一句話就是講義 p. 18 的訊息：證明不安全只需要 Week 2 的定義，不需要頻率分析。全題使用 $\alpha = \{\mathsf{a}, \ldots, \mathsf{z}\} \leftrightarrow \mathbb{Z}/26\mathbb{Z}$，所有加法在 $\mathbb{Z}/26\mathbb{Z}$ 中進行。

9.1 (i) 以 SKE 框架定義三個密碼

(i)

For each cipher in HISTORICAL-CIPHERS define it in the framework of SKE $=$ (KGen, Enc, Dec). In particular define the message, ciphertext and key spaces; $\mathcal{M}$, $\mathcal{C}$ and $\mathcal{K}$. Then define Enc and Dec. (If the historical description does not specify KGen you may assume a key is uniformly sampled from $\mathcal{K}$.)

解答

三個密碼共用 $\mathcal{M} = \mathcal{C} = \alpha^*$（任意長度的字母串），KGen 都是「從 $\mathcal{K}$ 均勻抽樣」。令 $m = m_1 \cdots m_\ell$、$c = c_1 \cdots c_\ell$。

	Caesar	**單字母替換**	**Vigenère（長度 n）**
$\mathcal{K}$	$\mathbb{Z}/26\mathbb{Z} = \{0, \ldots, 25\}$	$\mathrm{Perm}(\alpha)$（α 上所有雙射）	$(\mathbb{Z}/26\mathbb{Z})^n$
$\lvert\mathcal{K}\rvert$	26	$26! \approx 2^{88.4}$	26^n
$\mathsf{Enc}(\mathsf{sk}, m)$	$c_i = m_i + \mathsf{sk}$	$c_i = \mathsf{sk}(m_i)$	$c_i = m_i + \mathsf{sk}_{((i-1) \bmod n)+1}$
$\mathsf{Dec}(\mathsf{sk}, c)$	$m_i = c_i - \mathsf{sk}$	$m_i = \mathsf{sk}^{-1}(c_i)$	$m_i = c_i - \mathsf{sk}_{((i-1) \bmod n)+1}$
正確性	$(m_i + \mathsf{sk}) - \mathsf{sk} = m_i$	$\mathsf{sk}^{-1}(\mathsf{sk}(m_i)) = m_i$	同 Caesar，逐位置

每個方案都對 $i \in [\ell]$ 逐字母運算，最後回傳 $c = c_1 \cdots c_\ell$（或 $m = m_1 \cdots m_\ell$）。備註：

- Caesar 是替換密碼的特例（位移是排列），也是 Vigenère 的特例（$n = 1$）。
- 若不假設 Vigenère 金鑰長度恰為 n，講義的定義是 $\mathcal{K} = \{0, \ldots, 25\}^{\leq n}$，$\lvert\mathcal{K}\rvert = \sum_{i=1}^{n} 26^i$，Enc 中的 n 換成實際長度 $j = \lvert\mathsf{sk}\rvert$。
- 三者都是**確定性、無狀態**的 SKE——這一點是 (iii) 的根源。

9.2 (ii) 用 Enc(sk, ·) 取回金鑰

(ii)

For each cipher in HISTORICAL-CIPHERS describe how to recover sk given access to $\mathsf{Enc}(\mathsf{sk}, \cdot)$.

解答

這是 Week 2 IND-CPA 模型中的「加密預言機」。每個密碼都只需要一次查詢：

Caesar 查詢 $m = \mathsf{a}$（即 0），得 $c = 0 + \mathsf{sk} = \mathsf{sk}$。密文本身就是金鑰（以字母表示：$c$ 是第 sk 個字母）。任何單字母 m 也行：$\mathsf{sk} = c - m$。

單字母替換 查詢 $m = \mathtt{abcdefghijklmnopqrstuvwxyz}$（26 個字母各一次），得 $c = \mathsf{sk(a)sk(b)}\cdots\mathsf{sk(z)}$——這正是金鑰兩列表的下列，整張排列表一次到手。若每次只能查一個字母，26 次查詢亦可。

Vigenère 查詢 $m = \mathsf{a}^n$（n 個 a），得 $c_i = 0 + \mathsf{sk}_i = \mathsf{sk}_i$，$i = 1, \ldots, n$，即 $c = \mathsf{sk}$。若不知道 n（只知上限），查詢 a^n 後，c 會呈現週期 $j = |\mathsf{sk}|$ 的重複；取最短週期即得 sk。

一般原則：對確定性的「加法型」密碼，加密單位元 0 就直接吐出金鑰；對「查表型」密碼，加密整張表的輸入就吐出整張表。

9.3 (iii) 一對 m_0, m_1 區分出 b

(iii)

For each cipher in HISTORICAL-CIPHERS suppose sk is chosen according to KGen and $b \in \{0, 1\}$ is chosen uniformly. Give two messages $m_0, m_1 \in \mathcal{M}$ of the same length such that if you are given $c = \mathsf{Enc}(\mathsf{sk}, m_b)$ you can determine b.

解答

這是 Week 2 的 IND-EAV 遊戲（沒有預言機、只有一個挑戰密文）。想法：挑一個「有重複字母」與一個「沒有重複字母」的訊息，因為確定性的逐字母加密會保留「兩個位置的字母是否相同」——只要這兩個位置用的是同一個位移 / 同一張表。

Caesar 與單字母替換。 取 $m_0 = \mathsf{aa}$、$m_1 = \mathsf{ab}$。

- 若 $b = 0$：$c = \mathsf{Enc}(\mathsf{sk}, \mathsf{aa}) = (\mathsf{sk(a)}, \mathsf{sk(a)})$，兩個字母相同。
- 若 $b = 1$：$c = (\mathsf{sk(a)}, \mathsf{sk(b)})$，因為 sk 是單射且 $\mathsf{a} \neq \mathsf{b}$，兩個字母不同。

規則：$c_1 = c_2$ 則輸出 0，否則輸出 1。對每一把金鑰都正確，所以 $\Pr[\text{猜對}] = 1$，優勢 $|\Pr[b' = b] - \frac{1}{2}| = \frac{1}{2}$（最大值）。（對 Caesar，$\mathsf{sk}(x) = x + \mathsf{sk}$ 也是單射，論證相同。）

Vigenère（金鑰長度 n）。 aa 不再管用：位置 1、2 用不同的 $\mathsf{sk}_1, \mathsf{sk}_2$，$c_1 = c_2$ 與否取決於 $\mathsf{sk}_1 = \mathsf{sk}_2$ 與否，而非 b。要挑相距 n 的位置——它們用同一個 sk_1：

$$m_0 = \mathsf{a}^{n+1} = \underbrace{\mathsf{aa}\cdots\mathsf{a}}_{n+1}, \qquad m_1 = \mathsf{a}^{n}\mathsf{b} = \underbrace{\mathsf{aa}\cdots\mathsf{a}}_{n}\,\mathsf{b}.$$

- $b = 0$：$c_1 = 0 + \mathsf{sk}_1$，$c_{n+1} = 0 + \mathsf{sk}_1$，所以 $c_1 = c_{n+1}$。
- $b = 1$：$c_1 = \mathsf{sk}_1$，$c_{n+1} = 1 + \mathsf{sk}_1 \neq c_1$。

規則：$c_1 = c_{n+1}$ 則輸出 0，否則 1；$\Pr[\text{猜對}] = 1$，優勢 $= \frac{1}{2}$。若覺得 $n+1$ 不夠對稱，也可取 $m_0 = \mathsf{a}^{2n}$、$m_1 = \mathsf{a}^n\mathsf{b}^n$：$b = 0$ 時 c 的前半等於後半，$b = 1$ 時每個位置都差 1。

若金鑰長度未知（$\leq n$）。 令 $L = \mathrm{lcm}(1, 2, \ldots, n)$，取 $m_0 = \mathsf{a}^{L+1}$、$m_1 = \mathsf{a}^{L}\mathsf{b}$：不論實際長度 $j \leq n$ 為何，$j \mid L$，位置 1 與 $L+1$ 都用 sk_1，論證不變。

⚠ 為什麼訊息長度 $\leq n$ 時做不到？

若 $|m_0| = |m_1| = \ell \leq n$，Vigenère 每個位置用獨立均勻的 sk_i、每個都只用一次——這正是 §7 的 $\mathbb{Z}/26\mathbb{Z}$ 一次性方案，對長度 $\leq n$ 的單一訊息是完美保密的，任何 m_0, m_1 都無法區分。所以區分攻擊必須用到長度 $> n$ 的訊息，讓金鑰被重用。這恰好說明：Vigenère 的弱點就是「金鑰重用」。

題目沒問、但值得注意： 這些攻擊都只需要一個密文、不需要任何計算假設、而且對 Week 2 的任何版本（IND-EAV、IND-CPA、IND-CCA）都成立——講義 p. 18 的第一點。

通用模板

遇到「給出 m_0, m_1 區分確定性密碼」：(1) 找兩個位置 $i \neq i'$，它們在加密時用到同一份金鑰材料；(2) m_0 在這兩個位置放相同字母，m_1 放不同字母，其餘位置任意但相同；(3) 敵手看 $c_i \overset{?}{=} c_{i'}$。對 Caesar / 替換密碼，任意 $i \neq i'$ 都行；對週期 j 的 Vigenère，取 $i' = i + j$；對 OTP，不存在這樣的 i, i'——這就是完美保密。

10　Problem 2：On the one time pad

題目背景

In this question assume the message, ciphertext and key alphabet are binary, i.e. $\alpha_{\mathcal{M}} = \alpha_{\mathcal{C}} = \alpha_{\mathcal{K}} = \{0, 1\}$. This means $\mathcal{M}, \mathcal{C}, \mathcal{K} \subseteq \{0, 1\}^*$.

10.1　(i) 定義 OTP（訊息長度至多 n 位元）

(i)

Define the one time pad in the framework of SKE $=$ (KGen, Enc, Dec) for messages of length at most n bits. Here n is an arbitrary positive integer.

解答

$$\mathcal{K} = \{0,1\}^n, \qquad \mathcal{M} = \mathcal{C} = \{0,1\}^{\le n} = \bigcup_{\ell=0}^{n} \{0,1\}^\ell.$$

- KGen()：輸出均勻的 $\mathsf{sk} \leftarrow \{0,1\}^n$（每把金鑰機率 2^{-n}）。
- Enc(sk, m)：令 $\ell = |m|$；輸出 $c = m \oplus (\mathsf{sk}_1 \mathsf{sk}_2 \cdots \mathsf{sk}_\ell)$，即 $c_i = m_i \oplus \mathsf{sk}_i$，$i \in [\ell]$。
- Dec(sk, c)：令 $\ell = |c|$；輸出 $m = c \oplus (\mathsf{sk}_1 \cdots \mathsf{sk}_\ell)$。

正確性：$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{sk}, m)) = (m \oplus \mathsf{sk}_{1..\ell}) \oplus \mathsf{sk}_{1..\ell} = m \oplus 0^\ell = m$。完美保密（對每個固定長度 ℓ）：$\Pr[\mathsf{Enc}(K, m) = c] = \Pr[K_{1..\ell} = m \oplus c] = 2^{-\ell}$，與 m 無關（§8）。

- 密文與明文等長，所以長度會洩漏；這是定義允許的（Week 2 的 IND 遊戲要求 $|m_0| = |m_1|$）。
- 數一數：$|\mathcal{M}| = \sum_{\ell=0}^{n} 2^\ell = 2^{n+1} - 1 > 2^n = |\mathcal{K}|$。這不違反 $|\mathcal{K}| \ge |\mathcal{M}|$ 的定理——該定理針對的是「完美保密」成立的訊息空間，而這裡完美保密只對每個固定長度的子空間 $\{0,1\}^\ell$ 成立（$|\{0,1\}^\ell| = 2^\ell \le 2^n$）。
- 若只要加密恰好 n 位元的訊息，就是 KL-IMC Construction 2.8：$\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^n$。

10.2　(ii) 用 Enc$(\mathsf{sk}, \cdot)$ 取回 sk

(ii)

An adversary with access to Enc$(\mathsf{sk}, \cdot)$ can recover sk for the one time pad. How?

解答

查詢 $m = 0^n$：$\mathsf{Enc}(\mathsf{sk}, 0^n) = 0^n \oplus \mathsf{sk} = \mathsf{sk}$。一次查詢、密文就是金鑰。更一般地，查詢任何 n 位元的 m，得 $c = m \oplus \mathsf{sk}$，則 $\mathsf{sk} = c \oplus m$——這就是 XOR 的自反性：知道三個量中的任兩個就知道第三個。即使只允許查詢較短的訊息，查 0^ℓ 得到前 ℓ 位金鑰，拼起來即可。

10.3 (iii) 為何 (ii) 不與完美保密矛盾？

(iii)

Why does (ii.) not contradict the perfect secrecy of the one time pad? (Hint: what conditions on sk does the proper operation of the one time pad demand?)

解答

因為 (ii) 的敵手**違反了 OTP 的使用條件**：每把金鑰（的每個位元）只能用一次。

- 完美保密的定義（KL-IMC Def. 2.3 / Lemma 2.4）只談一則訊息的一個密文：$\Pr[M = m \mid C = c] = \Pr[M = m]$。定義裡沒有加密預言機；敵手看到的就只有那一個 c。OTP 對這個定義的證明完全正確，(ii) 沒有推翻它。
- 在 (ii) 中，敵手拿到 $\mathsf{Enc}(\mathsf{sk}, 0^n)$ 之後，sk 已經「用掉」了。正確操作的 OTP 此時應該**銷毀** sk，之後的任何訊息都要用新的金鑰加密；敵手知道一把已經不會再用的金鑰，毫無價值。若之後仍然用同一把 sk 加密真正的訊息，那就不是 OTP，而是「一次性密碼本用兩次」——完美保密從沒對它做任何承諾。
- 用 Week 2 的語言：(ii) 是 IND-CPA 攻擊（有 $\mathsf{Enc}(\mathsf{sk}, \cdot)$ 預言機）。OTP 是完美保密 / IND-EAV 安全，但**不是** IND-CPA 安全——這兩個定義不同，一個方案滿足前者而不滿足後者沒有任何矛盾（Week 2 的定義層級）。而且由 Week 2 定理 6.1，任何確定性方案都不可能 IND-CPA，OTP 當然也不例外。

題目提示的「proper operation 對 sk 的條件」就是：**均勻、至少與訊息一樣長、只用一次**。(ii) 破壞了第三條。

10.4 (iv) 抽到全零金鑰的機率

(iv)

Let KGen sample uniformly from $\mathcal{K} = \{0,1\}^n$. With what probability does KGen sample $\mathsf{sk} = 00\cdots0$, i.e. the all zeros secret key?

解答

$\mathcal{K}$ 有 2^n 個元素，均勻抽樣每個機率 $1/2^n$，所以

$$\Pr[\mathsf{sk} = 0^n] = 2^{-n}.$$

這與任何其他特定金鑰（例如 1^n、$0101\cdots$）的機率完全相同——全零金鑰沒有任何特殊地位。

n	1	8	128	256	1024
2^{-n}	0.5	$\approx 3.9 \times 10^{-3}$	$\approx 2.9 \times 10^{-39}$	$\approx 8.6 \times 10^{-78}$	$\approx 5.6 \times 10^{-309}$

10.5 (v) 全零金鑰存在，為何仍完美保密？

(v)

The one time pad is perfectly secure even though the all zeros secret key of (iv.) exists and is sampled with positive probability. Why?

解答

直覺上「全零金鑰把明文原封不動送出去」聽起來很糟。但完美保密問的不是「密文是否等於明文」，而是「**敵手看到密文後，能否縮小明文的可能範圍**」。

為什麼不洩漏。 假設敵手看到 c。對每一個可能的明文 $m \in \{0,1\}^\ell$，恰有一把金鑰 $k = m \oplus c$ 會把 m 送到 c，而每把金鑰機率相同。特別地，「$m = c$（金鑰全零）」只是 2^ℓ 個同樣可能的解釋中的一個——敵手無法知道發生的是這一個，還是 $m = c \oplus k$（$k \neq 0$）中的某一個。形式上，完美保密的證明 $\Pr[\mathsf{Enc}(K,m) = c] = 2^{-\ell}$ 對所有 m 成立，包括 $m = c$；全零金鑰的存在正是讓「$m = c$」這個可能性與其他可能性一樣可能所必需的。

反過來：把全零金鑰拿掉才會出問題（KL-IMC 習題 2.7）。 令 $\mathcal{K}' = \{0,1\}^n \setminus \{0^n\}$，均勻抽樣。則對任何 m：

$$\Pr[\mathsf{Enc}(K,m) = m] = \Pr[K = 0^n] = 0,$$
$$\Pr[\mathsf{Enc}(K,m') = m] = \Pr[K = m \oplus m'] = \tfrac{1}{2^n - 1} \quad (m' \neq m).$$

兩者不等，Lemma 2.4 失敗：敵手看到 c 後確定 $m \neq c$，排除了一個可能的明文。具體的 IND-EAV 敵手：送出 m_0, m_1；收到 c，若 $c = m_0$ 輸出 1，若 $c = m_1$ 輸出 0，否則擲硬幣。計算：

$$\Pr[\text{贏}] = \tfrac{1}{2} + \tfrac{1}{2(2^n-1)}, \qquad \mathsf{Adv} = \left|\Pr[\text{贏}] - \tfrac{1}{2}\right| = \tfrac{1}{2(2^n-1)} > 0.$$

$n = 1$ 時 $\Pr[\text{贏}] = 1$（金鑰只能是 1，$c = \bar{m}$，敵手必勝）；n 很大時優勢雖小，但完美保密要求的是恰好 0。Shannon 定理（§7）的角度：$|\mathcal{K}'| = 2^n - 1 < 2^n = |\mathcal{M}|$，條件 (2)「每對 (m,c) 恰有一把金鑰」對 $c = m$ 這一對失敗。

Enigma 的「反射器」：同樣的錯誤

Enigma 的反射器設計保證沒有任何字母會被加密成自己。德方以為這是優點（加密與解密對稱），Bletchley Park 卻把它當成最重要的槓桿之一：把猜測的明文（crib，例如 `WETTERVORHERSAGE`）沿著密文滑動時，**任何位置只要出現同一個字母對齊，就可以立刻排除**。「排除某些可能」正是上面 $\mathcal{K}'$ 的洩漏——歷史上真實發生、而且代價極高。

10.6 (vi) 單一位元、金鑰重用：如何猜 sk？

(vi)

Let $n = 1$ so messages are a single bit. Assume messages are equal to 0 with probability $1/3$ and equal to 1 with probability $2/3$. If the same $\mathsf{sk} \in \{0,1\}$ is mistakenly used to encrypt many such one bit messages $m_1, m_2, \ldots, m_\ell$ to $c_1, c_2, \ldots, c_\ell$, how should we guess sk? (Hint: each $c_i = m_i \oplus \mathsf{sk}$ will be either 0 or 1, if there are e.g. many more c_i equal to one than zero then which sk do we expect?)

解答

分析密文的分布。 $c_i = m_i \oplus \mathsf{sk}$，且各 m_i 獨立同分布。

- 若 $\mathsf{sk} = 0$：$c_i = m_i$，所以 $\Pr[c_i = 1] = 2/3$——密文中 **1** 偏多。
- 若 $\mathsf{sk} = 1$：$c_i = \bar{m}_i$，所以 $\Pr[c_i = 1] = 1/3$——密文中 **0** 偏多。

猜測規則（多數決）。 令 $N_1 = \#\{i : c_i = 1\}$、$N_0 = \ell - N_1$。

$$\text{若} N_1 > N_0 \text{ 猜} \mathsf{sk} = 0; \qquad \text{若} N_1 < N_0 \text{ 猜} \mathsf{sk} = 1; \qquad \text{平手則擲硬幣。}$$

猜到 sk 之後，$m_i = c_i \oplus \mathsf{sk}$ 把所有訊息一次解開。

為什麼這是最佳的（最大概似）。

$$\frac{\Pr[c_1 \cdots c_\ell \mid \mathsf{sk} = 0]}{\Pr[c_1 \cdots c_\ell \mid \mathsf{sk} = 1]} = \frac{(2/3)^{N_1}(1/3)^{N_0}}{(1/3)^{N_1}(2/3)^{N_0}} = 2^{N_1 - N_0}.$$

概似比只取決於 $d = N_1 - N_0$；$d > 0$ 時 $\mathsf{sk} = 0$ 更可能。若對 sk 的先驗是均勻的（KGen 確實如此），後驗為

$$\Pr[\mathsf{sk} = 0 \mid c] = \frac{2^d}{1 + 2^d},$$

例如 $d = 1$: 0.667、$d = 2$: 0.8、$d = 3$: 0.889、$d = 5$: 0.970、$d = 10$: 0.999。每多一個「多數票」，信心就多一倍的賠率。

錯誤機率隨 ℓ 下降。 多數決猜錯的機率（$\mathsf{sk} = 0$ 但 $N_1 \le N_0$，平手算一半；$\mathsf{sk} = 1$ 對稱）：

ℓ	1	3	5	7	11	15	21	31	51	101
$\Pr[\text{錯}]$	$\frac{1}{3}$	$\frac{7}{27} \approx .259$	$\frac{17}{81} \approx .210$	.173	.122	.088	.056	.027	.007	.0003

（由二項分布 $N_1 \sim \mathrm{Bin}(\ell, 2/3)$ 直接計算；Hoeffding 不等式給出 $\Pr[\text{錯}] \le e^{-\ell/18}$ 的指數衰減。）只要 ℓ 約 30，金鑰就幾乎確定；$\ell = 1$ 時 1/3 的錯誤率等於「永遠猜 $\mathsf{sk} = 0$」——單一訊息下 OTP 的確什麼都沒洩漏。

教訓。 完美保密的證明用到了「金鑰只與一則訊息 XOR」。一旦同一位元金鑰被用在多則非均勻的訊息上，密文的統計分布就會把金鑰「投票」出來——這是 VENONA 攻擊的一位元縮影。注意若訊息是均勻的（$\Pr[m = 0] = 1/2$），多數決無效；但重用仍洩漏 $c_i \oplus c_j = m_i \oplus m_j$，即「哪些訊息相同」。

10.7 (vii) 可以用同一把金鑰加密多則訊息嗎？

(vii)

Should you use the same key to encrypt many messages via the one time pad?

解答

不可以。名字裡的「one time」就是使用說明。理由（由弱到強）：

1. (vi)：訊息分布有偏時，重用直接洩漏金鑰。
2. 講義 p. 26 / §8：$c_1 \oplus c_2 = m_1 \oplus m_2$，即使訊息分布未知也洩漏兩則訊息的關係；已知其中一則就得到另一則；crib dragging 可同時還原兩則。
3. (ii)：若敵手能影響被加密的內容（選擇明文），一則已知訊息就給出金鑰，之後所有訊息全部暴露。
4. 形式上：重用金鑰的 OTP 是確定性方案，由 Week 2 定理 6.1 不可能 IND-CPA；更強地，任何確定性方案都不可能達到「多則訊息的完美保密」（KL-IMC §2.2 的討論 / 習題 2.3）。

正確做法：每則訊息用金鑰帶上新的一段，用過即銷毀（VENONA 的失敗正是來自重複印製密碼本）。

10.8 (viii) OTP 滿足哪些 Kerckhoffs 原則？

(viii)

Which of Kerckhoffs' principles does the one time pad satisfy? Which does it not?

解答

Kerckhoffs（1883）的六條原則（Week 2 §5）逐條檢查：

	原則		OTP
(P1)	系統即使不是數學上不可破，至少實務上不可破	✅	唯一數學上不可破的方案（完美保密，對無限算力敵手亦然）——超額達標。
(P2)	系統落入敵手也不應造成不便；安全只繫於金鑰	✅	演算法就是 XOR，完全公開；Shannon 的證明只假設金鑰保密。
(P3)	金鑰必須不靠書面筆記即可傳達與記住，並可隨意更換	❌	金鑰與所有訊息總長相同且真隨機，**不可能記住**，必須以紙本 / 紙帶 / 膠卷形式存放並實體遞送；「隨意更換」意味著重新分發整卷密碼本。這是 OTP 最根本的違反。
(P4)	適用於電報通訊	✅	Vernam 的設計就是電傳打字機；VENONA 的數字密碼本正是用電報傳送；熱線是電傳線路。
(P5)	可攜、單人即可操作	⚠	一本密碼本或一捲紙帶可以攜帶（SOE 的絲質密碼本、間諜的微縮膠卷），單人可操作；但金鑰材料量隨通訊量線性成長，大流量就需要機器與專人押運。
(P6)	易於使用，不需費神或記長規則	⚠	加解密本身極簡單（逐字加減 / XOR）；但「絕不重用、用後銷毀、雙方保持同步、保持均勻」這些紀律負擔沉重，而且一出錯就災難性崩潰（VENONA）。

總結：OTP 完美滿足 (P1)、(P2)、(P4)，明確違反 (P3)，(P5)、(P6) 視規模而定。Kerckhoffs 的原則本來就是為了「野戰可用」而寫；OTP 在安全性上無可挑剔，卻在可用性上失分，這與講義 p. 29「完美但脆弱、只有邊緣用例」的結論一致。

10.9 (ix) 一次性密碼本的歷史使用

(ix)

Research and describe a historical use of the one time pad.

解答

最著名、也最能說明「違反條件就崩潰」的案例是蘇聯的外交 / 情報密碼本與美國的 **VENONA** 計畫。

系統。 1930 年代起，蘇聯 NKVD / GRU 與外交部的電報採兩層加密：先用代碼簿把字詞換成 4 位數的代碼組，再把代碼組逐位加上一次性密碼本上的隨機 5 位數字（**模 10 不進位加法**——§1 的元素級運算）。密碼本每頁只用一次、用後銷毀。只要密碼本真隨機且不重用，這就是 $\mathbb{Z}/10\mathbb{Z}$ 上的完美保密方案。

錯誤。 1941–1942 年德軍入侵造成的壓力下，密碼本製造單位為了趕工，把同一批約 35,000 頁的金鑰印了兩份，分發給不同的使用單位（例如紐約的貿易代表團與華府的大使館）。重複的頁面在 1942–1948 年間被用掉。

攻擊。

- 1943 年 2 月 1 日，美國陸軍信號情報局（Arlington Hall）的 Gene Grabeel 開始系統性檢視蘇聯電報。1943 年 10 月，Richard Hallock 發現貿易電報中金鑰重用的統計證據；1944 年 11 月，Cecil Phillips 與 Genevieve Feinstein 找到剝離金鑰的方法；1946 年起語言學家 Meredith Gardner 開始還原代碼簿。
- 原理就是 §8 的 $c_1 - c_2 = m_1 - m_2$（模 10），再對「代碼組的差」做 crib dragging：已知的固定格式（電報開頭、常見代碼組、公開文件的譯文）給出突破口。
- 成果：約 3,000 份電報被（至少部分）翻譯（1995 年解密時公布的數字是 2,200，之後持續重工）；紐約 KGB 站發往莫斯科的電報中，1944 年約 49% 可讀，1943 年 15%，1942 年只有 1.8%——重複頁面越多的年份越容易破。揭露了 Rosenberg 夫婦、Klaus Fuchs、Cambridge Five 等間諜網；計畫一直持續到 1980 年，1995 年解密。

它示範了什麼。 (1) OTP 的安全性證明沒有錯——被攻破的電報全都是金鑰重用的那一部分，一次使用的部分至今未破；(2) 兩層加密（代碼簿 + OTP）沒有拯救它，因為重用讓敵手拿到 $m_1 - m_2$，而代碼組的分布高度非均勻（(vi) 的放大版）；(3) 金鑰分發與管理的操作失誤才是真正的弱點——這正是講義 p. 29 的結論。

其他歷史用例（簡述）。

用例	要點
Vernam / AT&T（1917–）	第一個 OTP 的機械實作：Baudot 紙帶 XOR；美軍一戰末期試用。
德國外交部（1920s–1945）	1920 年代起用 OTP（「i-Wurm」）；二戰的 GEE 系統改用機器產生的偽隨機密碼本，被美方於 1944 年破解——「幾乎隨機」不是隨機。
英國 SOE 絲質密碼本（1943–45）	Leo Marks 推動以印在絲綢上的一次性字母表（WOK / LOP）取代易被刑求逼出的詩歌密碼；用過的部分剪掉燒毀。
華府—莫斯科熱線（1963–1986）	ETCRRM II 一次性紙帶混合器；雙方互換金鑰紙帶（§8）。
冷戰間諜與數字電臺	微縮膠卷密碼本（如 1957 年被捕的 Rudolf Abel 所持）、Che Guevara 與古巴的通訊、至今仍可收到的短波「數字電臺」廣播，一般認為都是 OTP。

11 Problem 3：On practical use and cryptanalysis of historical ciphers

11.1 (i) 用三個密碼加密與解密

(i)

For each cipher in HISTORICAL-CIPHERS use it to encrypt and decrypt a message of your choice.

解答

三個例子已分別在 §2、§3、§4 逐字母算過，這裡彙整並附上可直接執行的程式。

	金鑰	明文 → 密文	解密檢查
Caesar	$\mathsf{sk} = 3$	`veni vidi vici` → `yhql ylgl ylfl`	每字母 -3：`y`→`v`、`h`→`e`⋯
替換	$\mathsf{sk} =$ `xiwvesjkndztb ogyhmlaqfpucr`（`a..z` 的像）	`the quick brown fox jumps over the lazy dog` → `ake hqnwz imgpo sgu dqbyl gfem ake txrc vgj`	查反表 sk^{-1}：`a`→`t`、`k`→`h`、`e`→`e`⋯
Vigenère	$\mathsf{sk} = \mathtt{kcl} = (10, 2, 11)$	`kings college london` → `ukyqu nynwoip vqynqy`	第 i 個字母減 $\mathsf{sk}_{((i-1) \bmod 3)+1}$

```python
import string
A = string.ascii_lowercase
def num(ch): return A.index(ch)

def caesar(text, k, dec=False):
    s = -k if dec else k
    return "".join(A[(num(ch)+s) % 26] if ch in A else ch for ch in text)

def mono(text, key, dec=False):          # key: 26-letter string, image of a..z
    table = dict(zip(key, A)) if dec else dict(zip(A, key))
    return "".join(table.get(ch, ch) for ch in text)

def vigenere(text, key, dec=False):
    # spaces are kept as-is and do NOT consume key letters
    ks = [num(ch) for ch in key]; out = []; j = 0
    for ch in text:
        if ch in A:
            s = -ks[j % len(ks)] if dec else ks[j % len(ks)]
            out.append(A[(num(ch)+s) % 26]); j += 1
        else:
            out.append(ch)
    return "".join(out)

c = caesar("veni vidi vici", 3)                  # -> yhql ylgl ylfl
assert caesar(c, 3, dec=True) == "veni vidi vici"

k = "xiwvesjkndztbogyhmlaqfpucr"
c = mono("the quick brown fox jumps over the lazy dog", k)
assert mono(c, k, dec=True).startswith("the quick")

c = vigenere("kings college london", "kcl")     # -> ukyqu nynwoip vqynqy
assert vigenere(c, "kcl", dec=True) == "kings college london"
```

11.2 (ii) Caesar 密文

(ii)

```
w jtcm kwum nwzbp w jtcm izqam w jtcm iakmvl w jtcm kwum qv
```

解答

46 個字母、13 個單字。三種方法都能在一分鐘內解開，依「越省力越前面」排列：

方法 A：單字母單字。 `w` 單獨成字出現四次。英文的單字母單字只有 `a`、`i`（詩句中偶有 `o`），所以

$$\mathsf{sk} = \mathtt{w} - \mathtt{a} = 22, \qquad \mathsf{sk} = \mathtt{w} - \mathtt{i} = 22 - 8 = 14, \qquad \mathsf{sk} = \mathtt{w} - \mathtt{o} = 22 - 14 = 8.$$

只需試三把金鑰。$\mathsf{sk} = 22$ 給 `a nxgq oayq…`、$\mathsf{sk} = 14$ 給 `i vfoy wigy…`，都不是英文；$\mathsf{sk} = 8$ 給出答案。

方法 B：trace（講義 p. 22）。 `jtcm` 出現四次，是最值得攻擊的單字。$\mathrm{trace}(\mathtt{jtcm}) = (\mathtt{t}-\mathtt{j}, \mathtt{c}-\mathtt{t}, \mathtt{m}-\mathtt{c}) = (10, -17, 10) = (10, 9, 10)$。在常見四字母單字中找 trace $(10, 9, 10)$：`blue` 的 trace 是 $(\mathtt{l}-\mathtt{b}, \mathtt{u}-\mathtt{l}, \mathtt{e}-\mathtt{u}) = (10, 9, -16) = (10, 9, 10)$ ✓。於是 $\mathsf{sk} = \mathtt{j}-\mathtt{b} = 9-1 = 8$，一次比對就得到金鑰。

方法 C：暴力搜尋（講義 p. 19）。 26 把金鑰全試，用「像不像英文」當 Dec 的檢查：

sk	$\mathrm{Dec}(\mathsf{sk}, c)$	sk	$\mathrm{Dec}(\mathsf{sk}, c)$
0	`w jtcm kwum nwzbp w jtcm izqam …`	13	`j wgpz xjhz ajmoc j wgpz vmdnz …`
1	`v isbl jvtl mvyao v isbl hypzl …`	14	`i vfoy wigy zilnb i vfoy ulcmy …`
2	`u hrak iusk luxzn u hrak gxoyk …`	15	`h uenx vhfx yhkma h uenx tkblx …`
3	`t gqzj htrj ktwym t gqzj fwnxj …`	16	`g tdmw ugew xgjlz g tdmw sjakw …`
4	`s fpyi gsqi jsvxl s fpyi evmwi …`	17	`f sclv tfdv wfiky f sclv rizjv …`
5	`r eoxh frph iruwk r eoxh dulvh …`	18	`e rbku secu vehjx e rbku qhyiu …`
6	`q dnwg eqog hqtvj q dnwg ctkug …`	19	`d qajt rdbt udgiw d qajt pgxht …`
7	`p cmvf dpnf gpsui p cmvf bsjtf …`	20	`c pzis qcas tcfhv c pzis ofwgs …`
8	**`o blue come forth o blue arise …`**	21	`b oyhr pbzr sbegu b oyhr nevfr …`
9	`n aktd bnld enqsg n aktd zqhrd …`	22	`a nxgq oayq radft a nxgq mdueq …`
10	`m zjsc amkc dmprf m zjsc ypgqc …`	23	`z mwfp nzxp qzces z mwfp lctdp …`
11	`l yirb zljb cloqe l yirb xofpb …`	24	`y lveo mywo pybdr y lveo kbsco …`
12	`k xhqa ykia bknpd k xhqa wneoa …`	25	`x kudn lxvn oxacq x kudn jarbn …`

只有一列讀得通——46 個字母遠超過 Caesar 的唯一解距離（約 1.5 個字母）。

答案。 $\mathsf{sk} = 8$（字母 `i`），明文：

`o blue come forth o blue arise o blue ascend o blue come in`

出自 Derek Jarman 的電影《*Blue*》（1993）——全片畫面只有一片 International Klein Blue，旁白是 Jarman 失明前寫下的文字；這句是開場的吟誦「O Blue come forth / O Blue arise / O Blue ascend / O Blue come in」。

11.3 (iii) 單字母替換密文

(iii)

```
itqr qfr iqfrl fttr q dtroqzgk zit dtroqzgk wtzvttf itqr qfr iqfrl dxlz wt
zit itqkz
```

解答

69 個字母、16 個單字、14 個不同的密文字母。暴力搜尋 26! 不可行，用講義 p. 20 的頻率分析，一步一步來。

第 0 步：統計。

密文字母	t	q	r	i	z	f	l	d	k	o	g	w	v	x
次數	13	10	9	7	7	6	3	3	3	2	2	2	1	1
%	18.8	14.5	13.0	10.1	10.1	8.7	4.3	4.3	4.3	2.9	2.9	2.9	1.4	1.4

單字長度：$4,3,5,4,1,8,3,8,7,4,3,5,4,2,3,5$。重複的單字：`itqr`（×2）、`qfr`（×2）、`iqfrl`（×2）、`dtroqzgk`（×2）、`zit`（×2）。

第 1 步：最多的字母是 e。 `t` 佔 18.8%，遠高於其他，猜 `t` ↦ `e`。

第 2 步：單字母單字。 `q` 單獨成字，而且是第二高頻（14.5%）：`q` ↦ `a`（a 是最常見的單字母單字且頻率 8.2%；`i` 也可能，先記著）。

第 3 步：三字母單字 `zit`。 出現兩次、結尾是 t=e：最常見的三字母單字 the 完全吻合 ⇒ `z` ↦ `t`、`i` ↦ `h`。檢查頻率：`z` 10.1%、`i` 10.1% 與 `t`、`h` 的高頻一致。

第 4 步：用已知字母擴散。 現在密文變成（大寫為已知）：

```
HEqr qfr Hqfrl fEEr A dEroAToK THE dEroAToK wETvEEf HEqr qfr Hqfrl dxlT wE THE HEAkT
```

- `itqkz` = `HEA?T`：只能是 heart ⇒ `k` ↦ `r`。
- `qfr` = `A??`，出現兩次且夾在兩個同樣的單字之間：and ⇒ `f` ↦ `n`、`r` ↦ `d`。頻率檢查：`r` 13%、`f` 8.7%——`d` 在這段文字裡偏高，但短文本本來就會偏離平均（見下方警告）。
- `itqr` = `HEAD` ✓（兩次）；`iqfrl` = `HAND?` ⇒ hands，`l` ↦ `s`；`fttr` = `NEED` ✓。
- `dtroqzgk` = `?EDIATOR`（o 介於 d、a 之間，g 介於 t、r 之間）：mediator ⇒ `d` ↦ `m`、`o` ↦ `i`、`g` ↦ `o`。
- `wtzvttf` = `?ET?EEN`：between ⇒ `w` ↦ `b`、`v` ↦ `w`。`wt` = `be` ✓。
- `dxlz` = `M?ST`：must ⇒ `x` ↦ `u`。

答案。

head and hands need a mediator the mediator between head and hands must be the heart

出自 Fritz Lang 的《*Metropolis*》（大都會，1927）的題詞與結尾字幕：「Mittler zwischen Hirn und Händen muss das Herz sein」——「頭腦與雙手之間的中介必須是心」。

還原的金鑰（部分）。

明文	a b c d e f g h i j k l m n o p q r s t u v w x y z
密文	q w ? r t ? ? i o ? ? ? d f g ? ? k l z x ? v ? ? ?

只確定了 14/26 格。其餘 12 個明文字母 {c,f,g,j,k,l,p,q,v,x,y,z} 與 12 個未出現的密文字母 {a,b,c,e,h,j,m,n,p,s,u,y} 之間有 $12! \approx 4.8 \times 10^8$ 種對應都與密文一致——**訊息唯一，金鑰不唯一**。這是替換密碼「一對 (m, c) 只給出 sk 的資訊」（講義 p. 12）的具體寫照；若要完整金鑰，需要更多密文。

⚠ 短文本的頻率會偏離平均

這段只有 69 個字母，第二、三高頻的密文字母對應到 a（14.5%）與 d（13%），而非英文平均的 t、a。單純把「頻率排名」一一對應會失敗（只有 e 可靠）。有效的做法是**頻率 + 單字結構 + 重複模式交替使用**，每猜一個字母就回頭驗證全文——這就是講義「需要足夠但出奇地少的密文」的實際含義。

11.4 (iv) Vigenère 密文（金鑰長度 3）

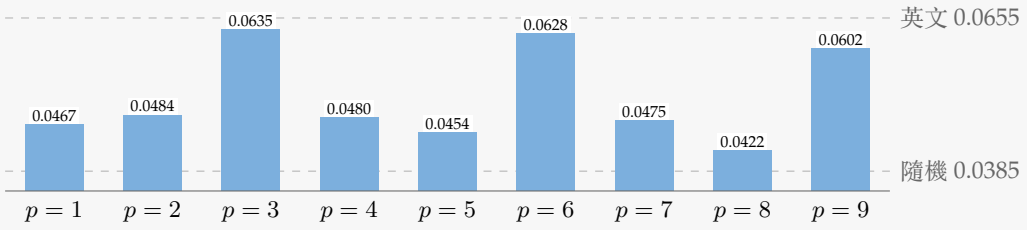
(iv)

```
zjc jsce hfzbej mml dcfdjv kmlzbeh zvzgvjc rhrrqi jvggg me tgis mwt rys
qycscrci cd ffgfb g nortvcu q zvokj ujzhrvf ge hfv ryiy lvop kvc kolevylgci
uyks ycz rycqv amdslkg uzzj ss jfgr zb rzac cwiv hcrfq zb prwl kwkv hm uwc
```

解答

180 個字母。題目已給金鑰長度 3，但我們先示範如何自己找出來，再依講義 p. 21 分流攻擊。

第 1 步：確認金鑰長度（Friedman 的 IC 檢驗）。 整段密文的 IC $= 0.0467$——介於英文（0.0655）與隨機（0.0385）之間，是多字母表密碼的指紋。對每個候選週期 p，把密文分成 p 條子密文，計算平均 IC：



$p=3$（及其倍數 6、9）明顯跳回英文水準，其餘停在 0.045 左右 ⇒ 金鑰長度 3。

⚠ Kasiski 在短密文上不可靠

Kasiski 檢驗找重複的三字母以上片段：只有 ehf（位置 6、96，距離 90）與 zbe（位置 9、25，距離 16）。$\gcd(90,16)=2$——錯誤答案！距離 $90=3\times30$ 確實支持 3，但 16 是巧合。180 個字母太短，巧合重複淹沒了真正的訊號；IC 檢驗用所有字母的統計量，穩健得多。

第 2 步：分成三條子密文（講義 p. 21）。 忽略空白，第 $1,4,7,\ldots$ 個字母為子密文 1，第 $2,5,8,\ldots$ 為子密文 2，第 $3,6,9,\ldots$ 為子密文 3，各 60 個字母。各自的字母計數：

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
子密文 1			3	1	5	4	1		4	4	5	3		1				4	1	1	2	**10**	1		3	7
子密文 2	2	4	3	2		3	4	**7**		2	1		1		4		2	2	6	2	2	4	4		1	4
子密文 3		1	**11**	1	1	2	6		2	4	2	4	6			2	3	7	1		1				4	2

第 3 步：每條子密文都是 Caesar，求位移。

- **子密文 1**：最多的是 v（10 次）。若 v=e，位移 $=21-4=17=$ r。χ^2 檢驗：$k=17$ 的 $\chi^2=9.0$，第二名 $k=23$ 高達 215——毫無懸念。
- **子密文 2**：最多的是 h（7 次）。若 h=e，位移 $=7-4=3=$ d，但 $\chi^2_3=365$，非常差。χ^2 最小的是 $k=14=$ o（$\chi^2=11.5$，第二名 237）。檢查：以 $k=14$ 解密後 h→t（7 次）、s→e（6 次）——這條子密文裡 t 恰好比 e 多一次。**「最多的字母就是 e」在 60 個字母的樣本上會失誤；χ^2 用全部 26 個頻率，不會。**
- **子密文 3**：最多的是 c（11 次）。c=e 給位移 $2-4=-2=24=$ y；$\chi^2_{24}=19.1$，第二名 185 ✓。

金鑰 sk $=(17,14,24)=$ **roy**。

第 4 步：解密。 注意題目「為了方便保留了空白」：空白不佔金鑰位置——第 i 個字母（跳過空白計數）用 $\text{sk}_{((i-1)\bmod 3)+1}$。前幾個字母：z $-$ r $=25-17=8=$ i、j $-$ o $=9-14=-5=21=$ v、c $-$ y $=2-24=-22=4=$ e、j $-$ r $=9-17=18=$ s……

ive seen things you people wouldnt believe attack ships on fire off the shoulder of orion i watched c beams glitter in the dark near the tannhauser gate all those moments will be lost in time like tears in rain time to die

出自《*Blade Runner*》（銀翼殺手，1982）結尾、複製人 Roy Batty（Rutger Hauer 飾）臨終的「Tears in rain」獨白——金鑰 `roy` 正是他的名字。
（若誤把空白也算進金鑰位置，會得到 `ive vulq qrbkql you flrfsh…`：開頭的 `ive` 與 `you` 剛好在位置對齊時正確，其餘是亂碼——這種「偶爾對、大多錯」的輸出正是計數方式弄錯的徵兆。）
程式化的完整流程。

```
import string, collections
A = string.ascii_lowercase
ENG = [8.17,1.49,2.78,4.25,12.70,2.23,2.02,6.09,6.97,0.15,0.77,4.03,2.41,
       6.75,7.51,1.93,0.10,5.99,6.33,9.06,2.76,0.98,2.36,0.15,1.97,0.07]

def ioc(seq):
    n = len(seq); cnt = collections.Counter(seq)
    return sum(v*(v-1) for v in cnt.values()) / (n*(n-1))

def chi2(seq):                          # how far is seq from English?
    n = len(seq); cnt = collections.Counter(seq)
    return sum((cnt[ch] - n*p/100)**2 / (n*p/100) for ch, p in zip(A, ENG))

def crack_vigenere(ct, max_len=10):
    L = [ch for ch in ct if ch in A]                        # drop spaces
    period = max(range(1, max_len+1),
                 key=lambda p: sum(ioc(L[i::p]) for i in range(p)) / p)
    key = []
    for i in range(period):
        stream = L[i::period]
        best = min(range(26), key=lambda k: chi2([A[(A.index(ch)-k) % 26] for ch
     in stream]))
        key.append(best)
    out, j = [], 0
    for ch in ct:
        if ch in A:
            out.append(A[(A.index(ch) - key[j % period]) % 26]); j += 1
        else:
            out.append(ch)
    return "".join(A[k] for k in key), "".join(out)

print(crack_vigenere("zjc jsce hfzbej mml dcfdjv ..."))   # -> ('roy', 'ive seen
     things ...')
```

三段密文告訴我們的事

- **Caesar**：任何一點結構（單字母單字、重複單字、trace）都足以一擊致命；暴力搜尋只是保底。
- **替換**：2^{88} 的金鑰空間毫無幫助；69 個字母就夠用頻率 + 單字模式破解。而且攻擊者不需要整把金鑰。
- **Vigenère**：先用統計量（IC）找週期，再分而治之；用 χ^2 而非「最多的就是 e」；小心空白等格式細節。
- 三者都只用到**密文與語言先驗**——講義 p. 20 的兩個前提——這是最弱的攻擊模型（COA）。

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　一頁看懂第三週

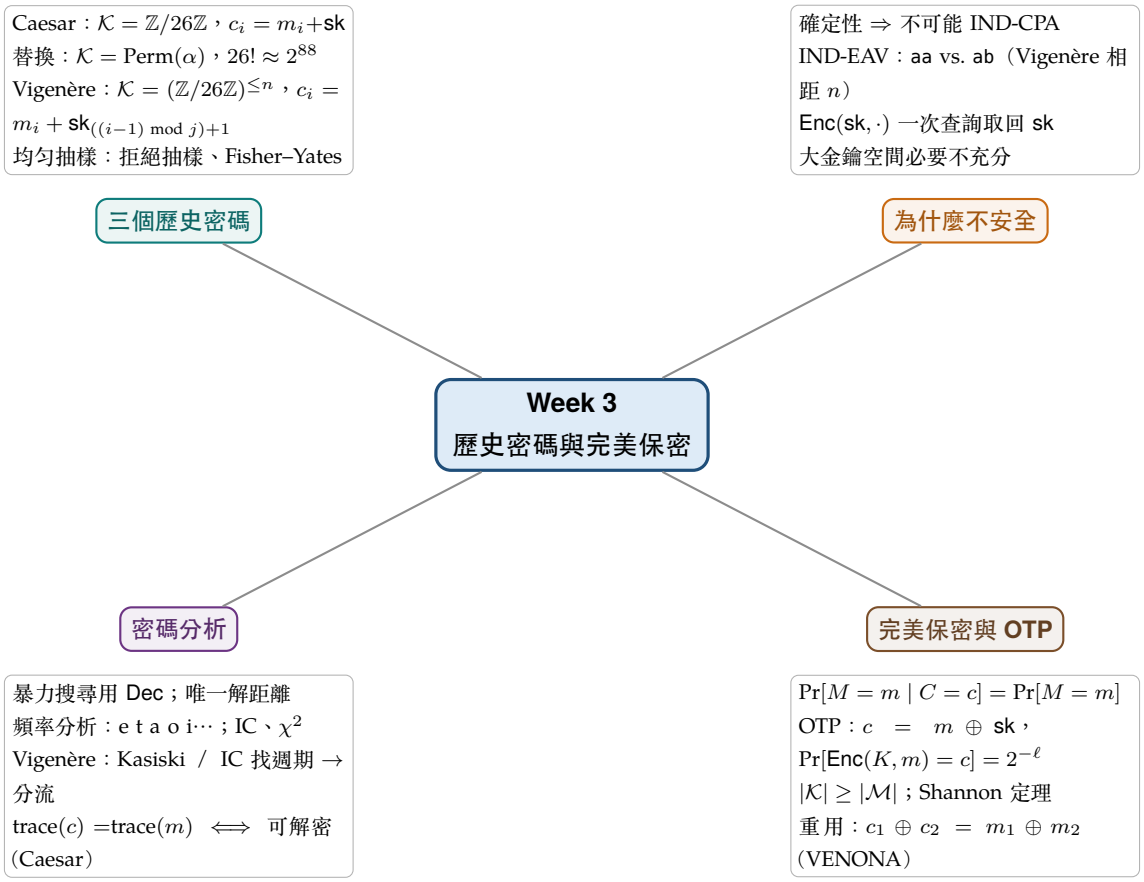


圖 4　第三週概念地圖。

12.2　必背定義與結論

密碼、抽樣與弱點。

概念	一句話版本
$\mathbb{Z}/n\mathbb{Z}$	$\{0,\dots,n-1\}$ 配模 n 加法：封閉、結合、單位元 0、反元素 $-a=n-a$（交換群）
XOR	$\mathbb{Z}/2\mathbb{Z}$ 的加法；$a\oplus a=0$（自反），加密 = 解密
Caesar	$\mathcal{K}=\mathbb{Z}/26\mathbb{Z}$，$\mathcal{M}=\mathcal{C}=\alpha^*$，$c_i=m_i+\mathsf{sk}$、$m_i=c_i-\mathsf{sk}$
單字母替換	$\mathcal{K}=\mathrm{Perm}(\alpha)$，$c_i=\mathsf{sk}(m_i)$、$m_i=\mathsf{sk}^{-1}(c_i)$；$\lvert\mathcal{K}\rvert=26!$
Vigenère	$\mathcal{K}=\{0..25\}^{\le n}$，$c_i=m_i+\mathsf{sk}_{((i-1) \bmod j)+1}$；$=j$ 個交錯的 Caesar
均勻抽樣	拒絕抽樣（26/32 接受）；排列用 Fisher–Yates；小心模偏差
五個弱點（Caesar）	小 $\mathcal{K}$ / (m,c) 給 sk / 間距保留 / 頻率分析 / 確定性

密碼分析與完美保密。

概念	一句話版本
暴力搜尋	對每個 $k \in \mathcal{K}$ 檢查 $\mathsf{Dec}(k, c_i) = m_i$；用 Dec 因 COA 無 m、隨機化 Enc 不重現 c
唯一解距離	$U \approx \log_2\lvert\mathcal{K}\rvert / 3.2$：Caesar 1.5、替換 28、Vigenère $1.5n$ 個字母
頻率分析	前提：訊息分布先驗 + 加密不消除統計；e 12.7、t 9.1、a 8.2、o 7.5、i 7.0
IC	$\sum n_a(n_a-1)/N(N-1)$；英文 0.0655、隨機 0.0385；Vigenère 分流找週期
trace	$(w_2 - w_1, \ldots, w_\ell - w_{\ell-1})$；Caesar 保持；$c$ 可解成 $m \iff \mathrm{trace}(c) = \mathrm{trace}(m)$
完美保密	$\forall$ 分布、m、c：$\Pr[M = m \mid C = c] = \Pr[M = m] \iff \Pr[\mathsf{Enc}(K, m) = c]$ 與 m 無關
OTP	$\mathcal{K} = \{0,1\}^n$，$c = m \oplus \mathsf{sk}_{1..\ell}$；完美保密因 $\Pr[\mathsf{Enc}(K, m) = c] = 2^{-\ell}$
$\lvert\mathcal{K}\rvert \geq \lvert\mathcal{M}\rvert$	否則存在 m' 不在 $\{\mathsf{Dec}(k, c)\}$ 中，$\Pr[M = m' \mid C = c] = 0 \neq \Pr[M = m']$
Shannon 定理	$\lvert\mathcal{K}\rvert = \lvert\mathcal{M}\rvert = \lvert\mathcal{C}\rvert$ 時：完美 $\iff$ 金鑰均勻且每對 (m, c) 恰一把金鑰
金鑰重用	$c_1 \oplus c_2 = m_1 \oplus m_2$；有偏訊息時多數決洩漏金鑰；VENONA
限制	金鑰 $\geq$ 訊息、單次使用、金鑰分發 $\Rightarrow$ 轉向計算安全（PRG）

12.3 常見誤區

⚠ 考試時最容易犯的錯

1. 「**金鑰空間夠大就安全**」。$26! \approx 2^{88}$ 的替換密碼 69 個字母就被破。大金鑰空間是必要條件。
2. **Vigenère 的區分對寫成 aa ／ ab**。位置 1、2 用不同位移；要用相距 n 的位置（a^{n+1} vs. $\mathsf{a}^n\mathsf{b}$）。
3. **Vigenère 的下標寫成 $\mathsf{sk}_{i \bmod j}$ 而不處理 0**。正確是 $\mathsf{sk}_{((i-1) \bmod j)+1}$（或 0 起始下標）。
4. **暴力搜尋用 Enc**。唯密文時沒有 m；隨機化加密時 Enc 不會重現 c。
5. 「**最多的字母就是 e**」**當成定理**。60 個字母的樣本會失誤（Problem 3 (iv) 子密文 2）；用 χ^2 / 內積。
6. **Kasiski 的 gcd 當成金鑰長度**。短密文的巧合重複會污染；要搭配 IC。
7. **完美保密寫成「密文均勻」**。正確敘述是「密文分布與訊息無關」 / 「後驗 = 先驗」。均勻是 OTP 的證明手段，不是定義。
8. 「**全零金鑰會洩漏明文，所以要拿掉**」。拿掉才會洩漏（$\Pr[\mathsf{Enc}(K, m) = m] = 0$）；KL 習題 2.7、Enigma 反射器。
9. 「**OTP 完美保密，所以 IND-CPA**」。OTP 確定性，一次 $\mathsf{Enc}(\mathsf{sk}, 0^n)$ 查詢就拿到金鑰；完美保密只談一則訊息。

10. $|\mathcal{K}| \geq |\mathcal{M}|$ 與 $\mathcal{M} = \{0,1\}^{\leq n}$ 的「矛盾」。完美保密只對固定長度子空間成立；長度本來就洩漏。
11. **忘記 Vigenère 解密時空白不佔金鑰位置**。Problem 3 (iv) 計數錯會得到「偶爾對、多數錯」的亂碼。
12. **把 Vigenère 歸給 Vigenère**。Bellaso 1553 發明；Vigenère 1586 的是自動金鑰密碼。

12.4 與前後週的連結

Week 1	保密性 vs. 完整性：OTP 只給保密、完全可塑（翻轉 c_i 即翻轉 m_i）；MAC 的必要性。
Week 2	SKE 語法、E_{sk} 單射、Kerckhoffs 六原則、IND-EAV / CPA / CCA、定理 6.1（確定性 $\Rightarrow$ 非 IND-CPA）、唯一解距離——本週全部用上。
Week 3	歷史密碼作為 SKE 的失敗；第一個嚴格定義（完美保密）；OTP 達成它；Shannon 證明代價不可避免。
Week 4+	放寬為計算安全：PRG 把短金鑰拉長成「看起來均勻」的金鑰流（串流密碼）；PRF / 分組密碼與隨機化模式達成 IND-CPA；安全性依賴明確假設 + 歸約證明——§5 的三原則。

13 中英術語對照表

English	中文	說明
modular arithmetic	模算術	在 $\mathbb{Z}/n\mathbb{Z} = \{0,\dots,n-1\}$ 中運算，結果取除以 n 的餘數
$\mathbb{Z}/n\mathbb{Z}$, integers modulo n	模 n 整數	配模加法成交換群；$n = 26$ 對應字母表
congruent (modulo n)	同餘	$a \equiv b \pmod n \iff n \mid (a-b)$
abelian group	交換群	封閉、結合、單位元、反元素、交換
identity element	單位元	$a + 0 = a$；全零金鑰就是單位元
inverse	反元素	$a + (-a) = 0$；$\mathbb{Z}/26\mathbb{Z}$ 中 $-a = 26 - a$
exclusive or (XOR), $\oplus$	互斥或	$\mathbb{Z}/2\mathbb{Z}$ 的加法；$a \oplus a = 0$
self-inverse	自反 / 自逆	XOR 的加密與解密是同一運算
elementwise / componentwise	逐元素	對字串每個位置獨立運算、不進位
Caesar (shift) cipher	凱撒（位移）密碼	$c_i = m_i + \mathsf{sk}$，$\mathcal{K} = \mathbb{Z}/26\mathbb{Z}$
ROT13	ROT13	Caesar 的 $\mathsf{sk} = 13$：加密 = 解密
monoalphabetic substitution cipher	單字母替換密碼	固定一個排列 $\mathsf{sk} \in \mathrm{Perm}(\alpha)$ 替換每個字母
permutation	排列	集合到自身的雙射；$\lvert\mathrm{Perm}(\alpha)\rvert = \lvert\alpha\rvert!$

（續下頁）

English	中文	說明
Fisher–Yates shuffle	Fisher–Yates 洗牌	線性時間均勻產生隨機排列（Durstenfeld 1964、Knuth P）
rejection sampling	拒絕抽樣	從 2^k 個等機率結果得到 N 個等機率結果：超出就重抽
modulo bias	模偏差	直接取 $x \bmod N$ 造成的不均勻
polyalphabetic cipher	多字母表密碼	同一明文字母依位置用不同替換表
Vigenère cipher	維吉尼亞密碼	以重複的金鑰字串做位移；j 個交錯的 Caesar
autokey cipher	自動金鑰密碼	Vigenère 1586 的真正發明：金鑰接上明文本身
key length / period	金鑰長度 / 週期	Vigenère 位移重複的間隔 j
tabula recta	密碼表	26×26 的位移表，手算 Vigenère 用
transposition (permutation) cipher	換位密碼	字母不變、位置重排；頻率分布不變
scytale	密碼棒	斯巴達的換位裝置
rail fence cipher	柵欄密碼	鋸齒狀寫、逐列讀
columnar transposition	欄位換位	依關鍵字重排矩陣的欄
product cipher	乘積密碼	替換與換位的組合；現代分組密碼的雛形
unprincipled	不講原則的	講義 p. 16：找漏洞—修補—再找漏洞的循環
cryptanalysis	密碼分析	在不知金鑰的情況下還原訊息或金鑰
ciphertext-only attack (COA)	唯密文攻擊	只有密文與語言先驗
known-plaintext attack (KPA)	已知明文攻擊	有若干 (m_i, c_i)
brute force / exhaustive search	暴力搜尋 / 窮舉	對每個 $k \in \mathcal{K}$ 檢查 $\mathsf{Dec}(k, c_i) = m_i$
unicity distance	唯一解距離	$U = H(\mathcal{K})/D$：超過此長度解幾乎唯一
redundancy (of a language)	冗餘度	英文約 3.2 位元 / 字母
frequency analysis	頻率分析	比較密文與語言的字母（雙字母、字首）頻率
bigram / trigram	雙字母組 / 三字母組	th, he, in⋯ / the, and, ing⋯
index of coincidence (IC)	重合指數	隨機兩位置字母相同的機率；英文 0.0655
chi-squared statistic	卡方統計量	$\sum (n_a - Np_a)^2/Np_a$，越小越像英文
Kasiski examination	Kasiski 檢驗	重複片段距離的公因數 ≈ 金鑰長度
Friedman test	Friedman 檢驗	用 IC 估計金鑰長度
crib / crib dragging	已知片段 / 拖行	猜測明文片段並沿密文滑動驗證
isomorph / word pattern	同構 / 單字模式	重複字母的位置模式，替換密碼保持
trace (of a word)	trace	相鄰字母差的序列；Caesar 保持
perfect secrecy	完美保密	$\Pr[M = m \mid C = c] = \Pr[M = m]$（Shannon 1949）
information-theoretic security	資訊論安全	對無限算力敵手仍安全
computational security	計算安全	只對多項式時間敵手、可忽略優勢

（續下頁）

English	中文	說明
prior / posterior distribution	先驗 / 後驗分布	看密文前 / 後對訊息的信念
perfect indistinguishability	完美不可區分	IND-EAV 遊戲中任何敵手恰 $1/2$
Shannon's theorem	Shannon 定理	$\lvert\mathcal{K}\rvert = \lvert\mathcal{M}\rvert = \lvert\mathcal{C}\rvert$ 時完美保密的充要條件
one-time pad (OTP)	一次性密碼本	$c = m \oplus \mathsf{sk}$，金鑰均勻、等長、單次
Vernam cipher	Vernam 密碼	OTP 的電傳打字機實作（1917–1919）
key reuse / two-time pad	金鑰重用 / 二次密碼本	$c_1 \oplus c_2 = m_1 \oplus m_2$
depth	重疊（密碼分析術語）	多段密文以同一段金鑰加密的狀態（VENONA）
malleability	可塑性	不知金鑰也能有意義地修改密文
maximum likelihood	最大概似	選使觀測最可能的假設；多數決
majority vote	多數決	Problem 2 (vi) 的猜金鑰規則
key distribution	金鑰分發	把金鑰安全送到對方；OTP 的致命瓶頸
hotline (Washington–Moscow)	華府—莫斯科熱線	1963 年起的電傳線路，用一次性紙帶加密
VENONA	VENONA 計畫	美方利用蘇聯密碼本重用破譯（1943–1980）
numbers station	數字電臺	短波廣播數字組，推測搭配 OTP
pseudorandom generator (PRG)	偽隨機生成器	下週：用短金鑰產生「看似均勻」的長金鑰流
stream cipher	串流密碼	PRG 輸出當 OTP 金鑰

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. 模算術

在 $\mathbb{Z}/26\mathbb{Z}$（$\mathtt{a}=0$）中計算 $\mathtt{y}+\mathtt{h}$、$\mathtt{b}-\mathtt{d}$、$-\mathtt{q}$；再計算 $\mathtt{10110} \oplus \mathtt{01101}$。解釋為何 XOR 的「解密」與「加密」是同一個函數，而 $\mathbb{Z}/26\mathbb{Z}$ 的不是。

解答

$\mathtt{y}+\mathtt{h} = 24+7 = 31 \equiv 5 = \mathtt{f}$；$\mathtt{b}-\mathtt{d} = 1-3 = -2 \equiv 24 = \mathtt{y}$；$-\mathtt{q} = -16 \equiv 10 = \mathtt{k}$（檢查 $16+10 = 26 \equiv 0$）。$\mathtt{10110} \oplus \mathtt{01101} = \mathtt{11011}$。在 $\mathbb{Z}/2\mathbb{Z}$ 中每個元素是自己的反元素（$1+1=0$），所以「加 k」與「減 k」相同；在 $\mathbb{Z}/26\mathbb{Z}$ 中 $-\mathsf{sk} \neq \mathsf{sk}$（除非 $\mathsf{sk} \in \{0, 13\}$），所以解密要用 $-\mathsf{sk}$。ROT13 是 $\mathbb{Z}/26\mathbb{Z}$ 中唯一「加密 = 解密」的非平凡位移。

2. 均勻抽樣

用五枚公平硬幣做拒絕抽樣產生 Caesar 金鑰。(a) 期望要擲幾輪？(b) 若改成直接取 $x \bmod 26$，金鑰落在 $\{0, \ldots, 5\}$ 的機率是多少？均勻時應是多少？(c) 為什麼不存在「不丟

棄任何結果」的均勻方法？

解答

(a) 每輪接受機率 $26/32$，期望輪數 $32/26 \approx 1.23$（幾何分布）。(b) $0, \ldots, 5$ 各有兩個原像，機率 $6 \times 2/32 = 12/32 = 37.5\%$；均勻時應為 $6/26 \approx 23.1\%$。(c) 任何以 k 枚硬幣決定輸出的確定性方法，每個輸出的機率都是（整數）$/2^k$；$1/26$ 不是這種形式，因為 $26 \nmid 2^k$。所以必須允許「重擲」（輪數不固定）。

3. 替換密碼的金鑰

(a) $\mathrm{Perm}(\alpha)$ 中把 e 映到自己的排列有幾個？(b) 若金鑰改成「用一個英文單字生成的關鍵字字母表」，違反了哪條 Kerckhoffs 原則？(c) 寫出一個 IND-EAV 敵手並說明它對每一把金鑰都成功。

解答

(a) 固定 e，其餘 25 個字母任意排列：$25! \approx 1.55 \times 10^{25}$，佔全部的 $1/26$。(b) (P1) / (P2) 的精神：有效金鑰空間從 $26!$ 縮成「英文單字數」（$\sim 10^5$），敵手知道系統（含「金鑰來自單字」）後就能窮舉；也與 (P3) 的初衷（好記）互相拉扯。(c) $m_0 = \mathsf{aa}$、$m_1 = \mathsf{ab}$；收到 $c = c_1c_2$，若 $c_1 = c_2$ 輸出 0，否則 1。任何排列都是單射：$\mathsf{sk(a)} = \mathsf{sk(a)}$ 恆成立、$\mathsf{sk(a)} \neq \mathsf{sk(b)}$ 恆成立，故 $\Pr[b' = b] = 1$。

4. Vigenère

金鑰長度恰為 4。(a) 給出一對長度 5 的 m_0, m_1 區分出 b，並說明為何長度 4 的訊息辦不到。(b) 證明對任意位置 i，$c_{i+4} - c_i = m_{i+4} - m_i$。(c) 若金鑰長度未知但 ≤ 4，給出一對通用的 m_0, m_1。

解答

(a) $m_0 = \mathsf{aaaaa}$、$m_1 = \mathsf{aaaab}$：位置 1 與 5 都用 sk_1，$b = 0$ 時 $c_1 = c_5$，$b = 1$ 時 $c_5 = c_1 + 1$。長度 ≤ 4 的訊息每個位置用獨立均勻的位移且各用一次，是完美保密的（§7），任何 m_0, m_1 都無法區分。(b) $c_{i+4} - c_i = (m_{i+4} + \mathsf{sk}_r) - (m_i + \mathsf{sk}_r) = m_{i+4} - m_i$，其中 $r = ((i-1) \bmod 4) + 1 = ((i+3) \bmod 4) + 1$。(c) $L = \mathrm{lcm}(1,2,3,4) = 12$：$m_0 = \mathsf{a}^{13}$、$m_1 = \mathsf{a}^{12}\mathsf{b}$，位置 1 與 13 對任何 $j \in \{1,2,3,4\}$ 都用 sk_1。

5. trace

(a) 計算 trace(yes)。(b) ekx 可能是 yes 的 Caesar 密文嗎？eky 呢？若可能，金鑰是多少？(c) 對 Vigenère（長度 j）trace 還成立嗎？什麼東西取而代之？

解答

(a) $\mathsf{trace}(\mathsf{yes}) = (\mathsf{e}-\mathsf{y}, \mathsf{s}-\mathsf{e}) = (4-24, 18-4) = (6, 14)$。(b) $\mathsf{trace}(\mathsf{ekx}) = (10-4, 23-10) = (6, 13) \neq (6, 14)$，不可能。$\mathsf{trace}(\mathsf{eky}) = (6, 14)$ ✓，$\mathsf{sk} = \mathsf{e} - \mathsf{y} = 4 - 24 \equiv 6$；檢查 $\mathsf{y} + 6 = 30 \equiv 4 = \mathsf{e}$、$\mathsf{e} + 6 = \mathsf{k}$、$\mathsf{s} + 6 = 24 = \mathsf{y}$ ✓。(c) 相鄰差不再守恆（$\mathsf{sk}_{i+1} - \mathsf{sk}_i$ 混進來），但相距 j 的差守恆（第 4 題 (b)）；替換密碼則只保留「相同字母的位置模式」。

6. 暴力搜尋與 Dec

某 IND-CPA 安全的方案以 $\mathsf{Enc}(\mathsf{sk}, m; \rho)$ 加密，ρ 是 128 位元隨機數。敵手有 (m, c) 一對，想暴力搜尋 sk。(a) 若用「$\mathsf{Enc}(k, m) \stackrel{?}{=} c$」檢查，會發生什麼？(b) 改用 Dec 為何正確？(c) 唯密文情形下，密文至少要多長才「值得」暴力搜尋替換密碼？

解答

(a) 即使 $k = \mathsf{sk}$，$\mathsf{Enc}(k, m)$ 用新的隨機數，與 c 相等的機率約 2^{-128}；正確金鑰會被誤判為錯，搜尋失敗。(b) 正確性保證 $\mathsf{Dec}(\mathsf{sk}, c) = m$ 對任何曾被用的 ρ 都成立，所以正確金鑰必通過檢查；錯誤金鑰通過的機率由密文長度控制。(c) 唯一解距離 $U \approx 88.4/3.2 \approx 28$ 個字母；少於此長度會有多把金鑰都給出「合理英文」。（但 26! 次嘗試本身不可行——這題的重點是「檢查」的邏輯，不是可行性。）

7. 單字母 Caesar 是完美的

令 $\mathcal{M} = \mathcal{C} = \alpha$（單一字母）、$\mathcal{K} = \mathbb{Z}/26\mathbb{Z}$ 均勻。(a) 證明 Caesar 完美保密。(b) 令 $\mathcal{M} = \alpha^2$，給出 m, m', c 使 $\Pr[\mathsf{Enc}(K, m) = c] \neq \Pr[\mathsf{Enc}(K, m') = c]$。

解答

(a) 對任意 $m, c \in \alpha$，$\mathsf{Enc}(K, m) = c \iff K = c - m$，機率 $1/26$，與 m 無關；由 Lemma 2.4 得證。亦可用 Shannon 定理：$|\mathcal{K}| = |\mathcal{M}| = |\mathcal{C}| = 26$、金鑰均勻、每對 (m, c) 恰一把金鑰。(b) $m = \mathsf{aa}$、$m' = \mathsf{ab}$、$c = \mathsf{bb}$：$\Pr[\mathsf{Enc}(K, \mathsf{aa}) = \mathsf{bb}] = \Pr[K = 1] = 1/26$，但 $\Pr[\mathsf{Enc}(K, \mathsf{ab}) = \mathsf{bb}] = 0$（需要 $K = 1$ 且 $K = 0$）。兩個字母就不完美——金鑰被用了兩次。

8. 不均勻的金鑰

OTP 的 KGen 改為：以機率 $\frac{1}{2}$ 輸出 0^n，否則從其餘 $2^n - 1$ 把金鑰中均勻選一把。此方案完美保密嗎？給出一個優勢為正的 IND-EAV 敵手並計算其獲勝機率。

解答

不完美。$\Pr[\mathsf{Enc}(K, m) = m] = \Pr[K = 0^n] = \frac{1}{2}$，而對 $m' \neq m$，$\Pr[\mathsf{Enc}(K, m') = m] = \frac{1}{2} \cdot \frac{1}{2^n - 1}$。敵手：送 $m_0 \neq m_1$；若 $c = m_0$ 猜 0、若 $c = m_1$ 猜 1、否則擲硬幣。$b = 0$ 時：$c = m_0$ 機率 $\frac{1}{2}$（猜對）、$c = m_1$ 機率 $\frac{1}{2(2^n-1)}$（猜錯）、其餘擲硬幣。$\Pr[\text{贏}] = \frac{1}{2} \cdot 1 + \frac{1}{2(2^n-1)} \cdot 0 + \left(1 - \frac{1}{2} - \frac{1}{2(2^n-1)}\right)\frac{1}{2} = \frac{3}{4} - \frac{1}{4(2^n-1)}$，對 $n \geq 2$ 嚴格大於 $\frac{1}{2}$（$n = 8$：約 0.749；$n = 1$ 時這

個 KGen 恰好是均勻的，獲勝機率正是 $\frac{1}{2}$）。Shannon 定理的條件 (1)「金鑰均勻」被破壞。

9. 二次密碼本

同一把 8 位元金鑰加密兩則訊息得 $c_1 = \texttt{00010010}$、$c_2 = \texttt{00010011}$。已知 $m_1 = \texttt{01001000}$（ASCII `H`）。求 m_2 與金鑰。若 m_1 未知，$c_1 \oplus c_2$ 仍透露了什麼？

解答

$\mathsf{sk} = c_1 \oplus m_1 = \texttt{00010010} \oplus \texttt{01001000} = \texttt{01011010}$；$m_2 = c_2 \oplus \mathsf{sk} = \texttt{00010011} \oplus \texttt{01011010} = \texttt{01001001}$（ASCII `I`）。或直接 $m_2 = c_1 \oplus c_2 \oplus m_1 = \texttt{00000001} \oplus \texttt{01001000} = \texttt{01001001}$。未知 m_1 時，$c_1 \oplus c_2 = \texttt{00000001}$ 告訴敵手「兩則訊息只差最低位元」——若訊息是 ASCII 大寫字母，就是相鄰的兩個字母（H/I、J/K⋯）。

10. 金鑰長度的下界

某人宣稱：$\mathcal{K} = \{0,1\}^n$、$\mathcal{M} = \{0,1\}^{n+1}$ 的方案 Π 完美保密。不看 Π 的細節，說明為什麼不可能，並描述一個直接的攻擊思路。

解答

$|\mathcal{K}| = 2^n < 2^{n+1} = |\mathcal{M}|$，違反定理「完美保密 $\Rightarrow |\mathcal{K}| \geq |\mathcal{M}|$」。攻擊：取任一 $\Pr[C = c] > 0$ 的密文 c，計算 $\mathcal{M}(c) = \{\mathsf{Dec}(k, c) : k \in \mathcal{K}\}$，至多 2^n 個訊息；至少有 2^n 個訊息不在其中。敵手看到 c 後就知道這些訊息不可能是明文——後驗與先驗不同。IND-EAV 版本：挑 $m_0 \in \mathcal{M}(c)$、$m_1 \notin \mathcal{M}(c)$⋯⋯更簡單地，隨機挑 m_0, m_1，收到 c 後檢查哪一個在 $\mathcal{M}(c)$ 裡（需要無限算力，但完美保密正是對這種敵手的承諾）。

11. Kasiski 與 IC

某 Vigenère 密文中，三字母片段 `qxr` 出現在位置 $5, 29, 53$（1 起始）。(a) Kasiski 給出哪些候選金鑰長度？(b) 分流後計算得平均 IC：$p = 2$: 0.044、3: 0.046、4: 0.063、6: 0.047、8: 0.065、12: 0.064。金鑰長度最可能是多少？為什麼不選 8 或 12？

解答

(a) 距離 $24, 24$，gcd $= 24$，候選長度為 24 的因數：$2, 3, 4, 6, 8, 12, 24$。(b) $p = 4$ 第一次跳回英文水準（0.063），8 與 12 也高是因為它們是 4 的倍數（分流更細仍是單一 Caesar，只是樣本變少）；$p = 2, 3, 6$ 低，排除。取最小的高 IC 週期：4。確認方式：分成 4 條子密文各做 χ^2，若都能得到明確的位移且合起來讀得通，即為答案。

12. VENONA 的教訓

蘇聯的電報先用代碼簿把字詞換成數字代碼組，再加上一次性密碼本（模 10）。為何「代碼簿 + OTP」兩層保護在密碼本重用之後仍然被破？這對「完美保密的證明到底證明了什麼」有何啟示？

解答

重用給出 $c_1 - c_2 = m_1 - m_2 \pmod{10}$，其中 m_i 是代碼組序列。代碼組的分布高度不均（常用詞、固定格式、流水號），所以「差」的分布也不均，可用 crib dragging 逐段還原兩則代碼組序列，再重建代碼簿（Meredith Gardner 的工作）。代碼簿只是另一個確定性替換，並不提供額外的隨機性。啟示：完美保密的證明是**條件式**的——「若金鑰均勻、等長、只用一次，則……」。證明沒有錯，錯的是操作違反了前提；安全性論證的前提與實際部署之間的落差，正是講義 p. 29「完美但脆弱」的意思。

參考資料

課程教材

- N. K. Nguyen, E. W. Postlethwaite, *6CCS3CIS Cryptography, third lecture*, King's College London, 2025/26.
- E. W. Postlethwaite, *Problem Set 3*, 6CCS3CIS Cryptography, 9 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（Caesar、位移、單字母替換、Vigenère 與其破譯）、第 2 章（完美保密：定義 2.3、引理 2.4、定義 2.5、引理 2.6、構造 2.8、定理 2.9、§2.3 定理 2.10、§2.4 Shannon 定理 2.11；習題 2.3、2.7、2.9）。
- C. E. Shannon, "Communication Theory of Secrecy Systems", *Bell System Technical Journal* 28(4), 1949, pp. 656–715.（完美保密、OTP、$|\mathcal{K}| \geq |\mathcal{M}|$、唯一解距離）
- D. Stinson, M. Paterson, *Cryptography: Theory and Practice*, 4th ed., CRC Press, 2018. 第 2 章（古典密碼與其分析：Kasiski、IC、χ^2 / 內積法）、第 3 章（Shannon 理論）。
- M. Rosulek, *The Joy of Cryptography*, 第 1–2 章（OTP、一次性性質、Kerckhoffs）。`https://joyofcryptography.com`
- D. Kahn, *The Codebreakers*, rev. ed., Scribner, 1996.（al-Kindī、Bellaso / Vigenère、Babbage、Kasiski、Friedman、Vernam / Mauborgne 的標準史料）
- S. Singh, *The Code Book*, Fourth Estate, 1999.（通俗版：Mary Queen of Scots、Vigenère 的破解、OTP）

古典密碼與密碼分析

- Suetonius, *De vita Caesarum*, Divus Iulius 56; Divus Augustus 88.（Caesar 位移 3、Augustus 位移 1）
- F. W. Kasiski, *Die Geheimschriften und die Dechiffrir-Kunst*, Berlin: E. S. Mittler und Sohn, 1863.（講義 p. 15 註腳）
- W. F. Friedman, *The Index of Coincidence and Its Applications in Cryptography*, Riverbank Publication No. 22, 1922（撰於 1920）。
- G. B. Bellaso, *La cifra del Sig. Giovan Battista Bellaso*, Venice, 1553；B. de Vigenère, *Traicté des chiffres*, Paris, 1586.
- R. A. Fisher, F. Yates, *Statistical Tables for Biological, Agricultural and Medical Research*, 1938（Example 12）；R. Durstenfeld, "Algorithm 235: Random permutation", *Comm. ACM* 7(7), 1964；D. E. Knuth, *TAOCP* vol. 2, §3.4.2, Algorithm P.
- 英文字母頻率：R. Lewand, *Cryptological Mathematics*, MAA, 2000（表 1.1）；P. Norvig, "English Letter Frequency Counts: Mayzner Revisited", 2013. https://norvig.com/mayzner.html
- 換位密碼史：MIT Press Reader, "A History of Cryptography from the Spartans to the FBI"（講義 p. 17 引用）。

一次性密碼本的歷史

- S. M. Bellovin, "Frank Miller: Inventor of the One-Time Pad", *Cryptologia* 35(3), 2011, pp. 203–222.
- G. S. Vernam, "Secret Signaling System", U.S. Patent 1,310,719, filed 13 Sep. 1918, granted 22 Jul. 1919；G. S. Vernam, "Cipher Printing Telegraph Systems for Secret Wire and Radio Telegraphic Communications", *J. AIEE* 45, 1926.
- S. M. Bellovin, "Vernam, Mauborgne, and Friedman: The One-Time Pad and the Index of Coincidence", in *The New Codebreakers*, LNCS 9100, 2016.
- NSA Center for Cryptologic History, *VENONA: Soviet Espionage and the American Response 1939–1957*（R. L. Benson, M. Warner eds., 1996）；R. L. Benson, *The Venona Story*；NSA, "VENONA: An Overview"（解密文件）。https://www.nsa.gov/Helpful-Links/NSA-FOIA/Declassification-Transparency-Initiatives/Historical-Releases/Venona/
- CIA, DCI speech on VENONA release, 11 July 1995. https://irp.fas.org/cia/product/dci_speech_71195.html
- Crypto Museum, "ETCRRM" 與 "Washington–Moscow Hotline". https://www.cryptomuseum.com/crypto/hotline/

- “Memorandum of Understanding Regarding the Establishment of a Direct Communications Link”, Geneva, 20 June 1963（美國國務院條約文本）。
- L. Marks, *Between Silk and Cyanide: A Codemaker's War 1941–1945*, HarperCollins, 1998.（SOE 絲質一次性密碼本）
- 講義 p. 28 圖片來源：D. Lowry-Duda, “Visualizing the one-time pad” 與 Crypto Stack Exchange 問答 “Taking advantage of one-time pad key reuse?”. `https://crypto.stackexchange.com/q/59`

Problem 3 的電影引文

- D. Jarman, *Blue*, 1993（Basilisk Communications / Channel 4）；劇本收錄於 D. Jarman, *Chroma*, 1994.
- F. Lang (dir.), T. von Harbou (scr.), *Metropolis*, UFA, 1927.（題詞 “Mittler zwischen Hirn und Händen muss das Herz sein”）
- R. Scott (dir.), *Blade Runner*, Warner Bros., 1982；Roy Batty 的「Tears in rain」獨白（Rutger Hauer 於拍攝時改寫的版本）。

本教材計算所用工具

- 所有解密、頻率統計、IC / χ^2 計算、多數決錯誤機率與金鑰空間大小均以 Python 3 標準函式庫（`collections`、`fractions`、`math`、`random`）重新計算；程式片段見 §11。

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。