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

Week 1

密碼學導論：保密性、完整性與安全定義

完全學習教材 & Problem Set 1 詳解

什麼是密碼學 • 秘密金鑰加密（SKE） • 正確性與安全性 • 安全定義的兩個面向
訊息鑑別碼（MAC） • Encrypt-then-MAC • 重放 / 重排 / 反射攻擊 • 密碼學作為軍火



教材內容 講義逐頁精讀、延伸補充（Katz–Lindell、Rosulek）、圖解、Problem Set 1 全部題目詳解、重點整理、中英術語表、自我測驗

對應教材 First lecture（73 頁投影片）、Problem Set 1（Problems 1–3）、KL-IMC 第 1–5 章相關節次

適用對象 修習 6CCS3CIS（Level 6）之學生；不需任何密碼學先備知識，僅需基本離散數學與機率概念

依據 6CCS3CIS 第一週講義（Nguyen & Postlethwaite, 2025-09-11）與 Problem Set 1（Postlethwaite, 2025-09-29）編寫，並參考 Katz–Lindell《Introduction to Modern Cryptography》第三版、Rosulek《The Joy of Cryptography》及相關原始文獻

目錄

導讀：如何使用本教材 **3**

1 什麼是密碼學？(講義 pp. 4–6) **6**

1.1 七種定義的比較 7

1.2 從「通訊」出發，但不止於通訊 8

1.3 延伸補充：現代密碼學的三個原則（KL-IMC §1.4） 8

2 思想實驗：兩位領主的秘密通訊（講義 pp. 7–11） **9**

2.1 場景設定 9

3 秘密金鑰加密（SKE）：語法與正確性（講義 pp. 8, 11） **9**

3.1 語法（syntax） 10

3.2 正確性（correctness） 11

4 什麼是「好的」加密方案？安全定義的五次迭代（講義 pp. 12–17） **11**

4.1 五次迭代 12

4.2 安全定義的兩個面向：(∗) 與 (†) 13

4.3 延伸補充：如何把「學不到任何新資訊」形式化？ 14

5 保密性之外：完整性與訊息鑑別碼（講義 pp. 18–20） **15**

5.1 被收買的騎士：保密不等於不可竄改 15

5.2 達成完整性的兩條路：工程 vs. 密碼學 16

5.3 訊息鑑別碼（MAC）的語法（講義 p. 20） 17

5.4 延伸補充：MAC 在現實中的樣貌 17

6 保密性與完整性之外 (!)：重放、重排、反射、延遲、拒絕（講義 pp. 21–22） **18**

6.1 五種攻擊 18

6.2 從中世紀隱喻到現代（講義 p. 22） 19

7 歷史插曲：密碼學曾是「軍火」(講義 p. 23) **20**

7.1 法律框架：AECA、ITAR 與美國軍火清單 20

7.2 三個案例 21

7.3 為什麼這段歷史屬於密碼學課程？ 22

8 機器、敵手與通用安全定義（講義 pp. 24–28） **23**

8.1 從領主到「機器」 . . . 23
8.2 兩個通用的安全敘述（講義 pp. 25–26） . . . 23
8.3 為什麼要用這種框架思考安全？（講義 p. 27） . . . 24
8.4 延伸補充：安全定義在紙上長什麼樣子——以 MAC 為例 . . . 24
8.5 第一次 tutorial 與 Problem Set 預告（講義 p. 28） . . . 25

9 Problem 1：On integrity（論完整性） 27
9.1 (i) MAC 的正確性應該是什麼意思？ . . . 27
9.2 (ii) 設計一個正確（但不必安全）的 MAC . . . 28
9.3 (iii) 用 MAC 保護密文的完整性：Encrypt-then-MAC . . . 29
9.4 (iv) 三個候選定義的（∗）與（†） . . . 30
9.5 (v) 哪一個最強？能達成嗎？ . . . 31
9.6 (vi) 應該選哪一個作為 MAC 的安全定義？ . . . 33
9.7 (vii) 至少與 (D3) 一樣強、但仍可達成的定義：強不可偽造性 . . . 34
9.8 (viii) 給敵手驗證預言機，會讓它更強嗎？ . . . 36

10 Problem 2：On encryption（論加密） 38
10.1 (i) 確定性加密為何不可取？ . . . 38
10.2 (ii) 在訊息中加入什麼，可以防止重放、重排與反射？ . . . 40
10.3 (iii) 若敵手能選擇性地拒絕傳遞，對 (ii) 的答案有何影響？ . . . 42

11 Problem 3：On cryptography as munitions（論密碼學作為軍火） 44
11.1 (i) 閱讀指南 . . . 44
11.2 (ii) 寫作指南 . . . 46

12 本週重點整理 50
12.1 一頁看懂第一週 . . . 50
12.2 必背定義 . . . 50
12.3 常見誤區 . . . 51

13 中英術語對照表 51

14 自我測驗（附解答） 53

參考資料 56

導讀：如何使用本教材

課程資訊速覽

項目	內容
課程	6CCS3CIS Cryptography（三年級 / Level 6 版本）
授課教師	Eamonn W. Postlethwaite（前五週）、Ngoc Khanh Nguyen（後五週）
Office hours	以 email 預約：Eamonn 週一 16–17；Khanh 週五 13–14
上課時間	每週四 16–18 講課；週一或週二 tutorial
評量方式	100 % 一月筆試（capped resit）；期末前會提供 mock exam
討論平台	所有課程內容與行政問題一律使用 KEATS 論壇
參考書（非必買）	Katz–Lindell, *Introduction to Modern Cryptography*, 3rd ed.（簡稱 KL-IMC） Mike Rosulek, *The Joy of Cryptography*（簡稱 R-JC，線上免費：`https://joyofcryptography.com/`）

本週學習目標

讀完本教材，你應該能夠：

1. 用自己的話比較幾種「密碼學」的定義，並說明為何教師認為牛津字典的定義「不好」。
2. 寫出**秘密金鑰加密**（secret key encryption, SKE）與**訊息鑑別碼**（message authentication code, MAC）的**語法**（syntax），並清楚區分「語法」、「正確性」（correctness）與「安全性」（security）三件事。
3. 說明一個安全定義必然包含的兩個面向：**敵手不應做到的事**（$*$）與**敵手被賦予的能力**（$\dagger$），並據此分析、比較不同定義的強弱與可達成性。
4. 解釋為何「保密性」（secrecy / confidentiality）與「完整性」（integrity）是兩種不同的性質，需要不同的工具達成。
5. 描述重放（replay）、重排（reordering）、反射（reflection）、延遲（delay）與拒絕傳遞（denial）攻擊，並知道哪些能用密碼學手段防範、哪些屬於網路工程範疇。
6. 完整解答 Problem Set 1 的三大題，包括 MAC 安全定義 (D1)–(D3) 的比較、強不可偽造性、驗證預言機、確定性加密的風險，以及計數器與方向位元的設計。
7. 了解 1990 年代美國將密碼學視為軍火的歷史脈絡（PGP、Netscape、*Junger v. Daley*），並能就「技術物如何變成法律與社會爭議」寫出一頁反思。

本教材的結構與閱讀建議

- **第一部分｜講義精讀與延伸**：依照講義投影片的順序逐節講解，並以 KL-IMC、R-JC 與原始論文補充講義中「留白」的部分（例如安全定義的形式化、Encrypt-then-MAC 的理由、歷史細節）。每節末尾的 🎓 **考試重點**框標示可能在筆試出現的概念。

- **第二部分｜ Problem Set 1 詳解**：先原文重述題目（✏），再給出詳細解答（🔑），包含證明、反例與延伸討論。建議先自己嘗試作答，再對照解答。
- **第三部分｜總複習**：重點整理、中英術語對照表、自我測驗（附解答）與參考資料。

記號約定

- $x \leftarrow \mathsf{Alg}(\cdot)$ 表示 Alg 可能是**隨機化**（randomised）演算法，x 是其一次執行的輸出；$x = \mathsf{Alg}(\cdot)$ 則強調 Alg 是**確定性**（deterministic）的。KL-IMC 以 $x := \mathsf{Alg}(\cdot)$ 表示後者。
- Problem Set 中以方框標示通訊雙方必須保密的物件（通常是金鑰），例如 $\boxed{\mathsf{sk}}$、$\boxed{\mathsf{ik}}$。本教材正文中為了版面簡潔，多直接寫 sk、ik。
- $\mathsf{Alg}(k, \cdot)$ 表示一個「黑盒子」（oracle，預言機）：敵手可以輸入任何 x 並取得 $\mathsf{Alg}(k, x)$，但看不到 k。
- $\mathcal{K}, \mathcal{M}, \mathcal{C}, \mathcal{T}$ 分別為金鑰空間、訊息空間、密文空間與標籤空間；$\{0,1\}^n$ 表示長度為 n 的位元字串集合；$a \,\|\, b$ 表示字串串接。

第一部分

講義精讀與延伸

什麼是密碼學• 思想實驗• SKE 與正確性• 安全定義• 完整性與 MAC • 其他攻擊• 歷史插曲• 敵手模型

1　什麼是密碼學？（講義 pp. 4–6）

講義一開始並沒有急著給出公式，而是先問一個看似簡單的問題：「密碼學是什麼？」教師刻意列出七種來源各異的「定義」，目的不是要你背下其中一個，而是要你體會：**對同一個學科，不同的人會因為關心的面向不同而給出不同的界定**；而一旦下了定義，就等於劃出了界線（Wilde：「to define is to limit」）。

講義的開場提問

如果有人請你描述你對密碼學這門學問（或它所扮演的角色）的理解，分別在

- 學術（academic）場域、
- 產業或經濟（industrial / economic）場域、
- 文化或社會（cultural / societal）場域，

你會怎麼回答？（先自己想一分鐘再往下讀。）

1.1 七種定義的比較

來源	定義（原文大意）	著眼點／評註
Oxford English Dictionary	以代碼或密碼書寫的技藝或實務；加密的科學。	只談「加密」。教師明言「這是壞的定義」：它把密碼學等同於保密通訊，完全遺漏了完整性、鑑別、數位簽章、零知識證明、安全多方計算等現代密碼學的核心主題。
Katz–Lindell（KL-IMC）	研究用以保護數位資訊、系統與分散式計算、使其免受敵手攻擊的數學技術。	強調「數學」與「敵手」（adversarial）：安全是相對於攻擊者而言的，而且要能被嚴格論證。
Rosulek（R-JC）	研究如何控制資訊的存取（controlling access to information）。	最精簡。「存取」同時涵蓋「誰能讀」（保密性）與「誰能寫 / 改」（完整性、鑑別）。
Martin	研究計算的極限。	把密碼學放回理論計算機科學：安全性建立在「某些問題難以計算」之上（例如大數分解、離散對數、格問題）。
Eamonn（授課教師）	研究如何讓計算變得安全，以及——或許更重要的——「讓計算安全」到底是什麼意思。	本課程的主旋律：定義（definition）本身就是研究對象。第一週幾乎全部在談「怎麼定義安全」。
Wilde	「下定義就是設限。」	提醒：任何定義都有盲點；上面每一句都排除了某些東西。
Blanchette	密碼學是為了規範行為而寫的程式碼（code created with the sole purpose of regulating behaviour）。	社會學 / STS 的視角：密碼學不只是數學，它是「用技術手段強制某些規則」的工具，因此必然涉及權力、法律與政治（見第 7 節的出口管制歷史）。

為什麼教師說牛津的定義「BAD」？

一個好的定義應該能涵蓋這門學科實際在做的事。現代密碼學至少包括：

- **保密性**（confidentiality / secrecy）：加密。
- **完整性與鑑別**（integrity / authenticity）：MAC、數位簽章——這是本週後半的主題，而牛津定義完全沒提。
- **更進階的功能**：零知識證明（在不洩漏秘密的情況下證明你知道它）、安全多方計算（多方在不互相揭露輸入的情況下共同計算）、同態加密（直接對密文運算）等。

把密碼學等同於「加密」，就像把「醫學」定義成「開刀」。

1.2　從「通訊」出發，但不止於通訊

講義 p. 6 指出：被問到密碼學時，多數人會想到**安全通訊**（secure communications）。這確實是歷史上最悠久的用途——從斯巴達的 scytale、凱撒密碼、二戰的 Enigma 到今天的 TLS。因此講義也選擇從一個「兩位領主互相傳訊」的中世紀思想實驗出發（第 2 節）。但請記住講義最後一頁預告的討論題目之一就是「*cryptography beyond secure communications*」：磁碟加密（自己和未來的自己通訊）、密碼學貨幣、電子投票、匿名通訊等，都超出了「兩人傳訊」的框架。

1.3　延伸補充：現代密碼學的三個原則（KL-IMC §1.4）

講義提到「有興趣的人可以讀 KL-IMC 第 1、2 節」。KL-IMC 第一章把「現代密碼學」與「古典密碼學」的分水嶺總結為三個原則，它們正好解釋了為什麼本週要花那麼多篇幅談「定義」：

現代密碼學的三原則

1. **形式化定義**（Principle 1 – Formal Definitions）：在設計或分析任何方案之前，必須先精確地說清楚「安全」是什麼意思。沒有定義，就無法判斷一個方案是否達標，也無法比較兩個方案。
2. **精確的假設**（Principle 2 – Precise Assumptions）：多數方案的安全性依賴某個「某問題很難」的假設（例如大數分解困難）。假設必須明確陳述，才能被檢驗、被比較、被替換。
3. **安全證明**（Principle 3 – Proofs of Security）：在定義與假設之下，以嚴格的數學證明說明方案安全——通常是「若存在攻破方案的敵手，則可將其轉化為解決困難問題的演算法」（歸約，reduction）。

本週講義的「五次迭代」（第 4 節）與 Problem Set 1 全部都在練習第一原則：怎麼寫出一個既不太弱、又能達成的定義。

Kerckhoffs 原則（1883）

荷蘭語言學家 Auguste Kerckhoffs 在 1883 年提出：一個密碼系統即使除了金鑰以外的一切都被敵人知道，仍應是安全的。換言之，**演算法公開、只有金鑰保密**。理由有三：(1) 金鑰比演算法容易保密、容易更換；(2) 公開的演算法可以接受公眾檢驗；(3) 大量使用者可以共用同一套演算法而各持不同金鑰。講義中兩位領主「當面商定加密方法與解密方法」，但真正需要「耳語」保密的只有金鑰——這正是 Kerckhoffs 原則的體現。本週的所有安全定義也都預設敵手完全知道演算法 $\mathsf{Enc}, \mathsf{Dec}, \mathsf{Tag}, \mathsf{Vf}$。

考試重點

- 能列舉至少三種密碼學的定義並指出各自的著眼點；能說明為何「密碼學 = 加密」是不足的。
- 記住「to define is to limit」的寓意：定義是研究工具，而非真理；本課程大量工作就是

在挑選合適的定義。

- Kerckhoffs 原則：安全性只能依賴金鑰的保密，不能依賴演算法的保密。

2　思想實驗：兩位領主的秘密通訊（講義 pp. 7–11）

2.1　場景設定

兩位領主 Francis 與 Henry 各自駐紮在營帳（pavilion）中，想要「安全地」通訊。他們事先當面商定了三樣東西：

1. 一個**加密方法**：用一把秘密金鑰把訊息（message）轉成密文（ciphertext）；
2. 一個**解密方法**：用同一把秘密金鑰把密文還原成訊息；
3. 一把**秘密金鑰**（他們互相耳語，確保沒人聽見）。

之後 Francis 要傳訊給 Henry 時：加密 → 寫在羊皮紙（parchment）上 → 交給最信任的騎士（rider）送達 → Henry 解密閱讀。

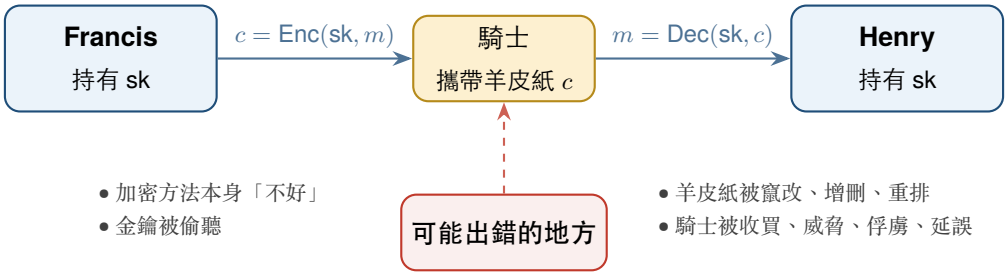


圖 1　兩位領主的思想實驗。講義 p. 10 的「Think...」要我們分三個層面思考「什麼會出錯」。

講義 p. 10：What can go wrong?

- 如果兩位領主不太聰明，設計出的加密方案很「壞」——這裡的「壞」可能是什麼意思？
- 羊皮紙上的密文可以被改動、增補、加長、縮短、重新排列嗎？改過之後解密出來還會「有意義」嗎？
- 騎士可能被俘、被殺、被收買、被威脅，或被迫延遲送達嗎？

這三組問題分別對應到本週接下來的三個主題：**保密性**（第 4 節）、**完整性**（第 5 節）、以及**保密性與完整性之外的性質**（第 6 節）。

3　秘密金鑰加密（SKE）：語法與正確性（講義 pp. 8, 11）

3.1 語法（syntax）

講義說「我們會不厭其煩地回到這個定義」（*ad nauseam*）。上面的思想實驗其實已經描述了一個**秘密金鑰加密方案**（secret key encryption scheme, SKE），又稱對稱式加密（symmetric encryption）或私鑰加密（private-key encryption）。

SKE 的語法

一個秘密金鑰加密方案是三個演算法的三元組 $\mathsf{SKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$：

- KGen（key generation）：不需輸入（或只輸入安全參數），輸出一把秘密金鑰 $\mathsf{sk} \in \mathcal{K}$：

$$\mathsf{sk} \leftarrow \mathsf{KGen}.$$

- Enc（encrypt）：輸入金鑰 sk 與訊息 $m \in \mathcal{M}$，輸出密文 $c \in \mathcal{C}$：

$$c = \mathsf{Enc}(\mathsf{sk}, m) \quad （若 \mathsf{Enc} 隨機化，則寫 c \leftarrow \mathsf{Enc}(\mathsf{sk}, m)）.$$

- Dec（decrypt）：輸入金鑰 sk 與密文 c，輸出訊息 m（或一個特殊符號 $\perp$ 表示「拒絕」）：

$$m = \mathsf{Dec}(\mathsf{sk}, c).$$

$\mathcal{K}, \mathcal{M}, \mathcal{C}$ 分別稱為**金鑰空間**、**訊息空間**與**密文空間**。

語法 ≠ 正確性 ≠ 安全性

講義特別強調：「上面的敘述完全沒有說到正確性或安全性，只有語法。」這是本週最重要的觀念之一。語法只規定「有哪些演算法、各吃什麼、吐什麼」；一個語法上合格的 SKE 可以完全沒用（解不回來）或完全不安全（密文就是明文）。三者的關係如圖 2。

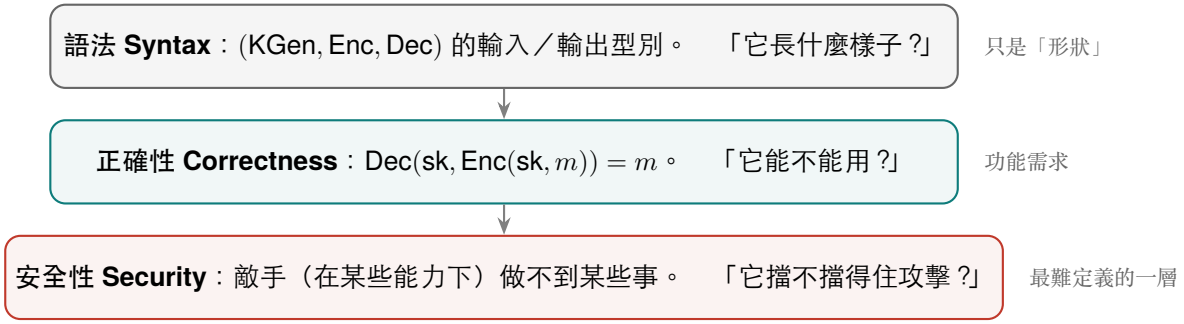


圖 2　語法、正確性、安全性是三個獨立的層次；本週講義依序建立這三層。

3.2　正確性（correctness）

SKE 的正確性（講義 p. 11）

若以秘密金鑰 sk 把訊息 m 加密成密文 $c = \mathsf{Enc}(\mathsf{sk}, m)$，則以同一把金鑰解密該密文會還原出同一個訊息：$m = \mathsf{Dec}(\mathsf{sk}, c)$。用量詞寫出來：

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

若 Enc 是隨機化的，則上式要求對 Enc 的每一種隨機選擇都成立（**完美正確性**，perfect correctness）；有些方案只要求「以壓倒性的機率成立」。

講義再次提醒：「這完全沒有說到安全性！」正確性只保證「合法的收件人解得回來」，對敵手能做什麼隻字未提。

正確但不安全 vs. 安全但不正確

- 「什麼都不做」方案（講義 p. 12）：$\mathsf{Enc}(\mathsf{sk}, m) = m$，$\mathsf{Dec}(\mathsf{sk}, c) = c$。顯然正確（$\mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{sk}, m)) = m$），但密文就是明文，毫無保密性可言。
- 「全部吞掉」方案：$\mathsf{Enc}(\mathsf{sk}, m) = 0$（不論 m 為何），Dec 隨便輸出。敵手從密文確實學不到任何關於 m 的資訊——但 Henry 也一樣，方案不正確，沒有任何用途。

這兩個極端告訴我們：正確性與安全性是**互相拉扯**的要求，好的方案要同時滿足兩者。

與 KL-IMC 記號的對照

KL-IMC 把金鑰寫成 k、演算法寫成 $\mathsf{Gen}, \mathsf{Enc}_k(m), \mathsf{Dec}_k(c)$，並加入安全參數 1^n（$\mathsf{Gen}(1^n)$ 輸出長度 $\geq n$ 的金鑰）。講義的 $\mathsf{sk} \leftarrow \mathsf{KGen}$、$\mathsf{Enc}(\mathsf{sk}, m)$ 與之完全對應，只是省略了安全參數。考試時兩種寫法都可接受，但請前後一致。

考試重點

- 能默寫 SKE 的語法（三個演算法、各自的輸入輸出與空間）。
- 能用量詞寫出正確性，並說明「正確性不蘊含安全性」（舉「什麼都不做」方案為例）。
- 能說明思想實驗中三類「會出錯的地方」分別對應哪個安全性質。

4　什麼是「好的」加密方案？安全定義的五次迭代（講義 pp. 12–17）

講義用蘇格拉底式的問答，連續提出五個候選定義，每提出一個就問「**Strong enough?**」，再用一個反例推翻（或修正）它。這段推演是理解「安全定義為何微妙」的最佳範例，也是 Problem Set 1 第一題的直接前身——那裡會對 MAC 做一模一樣的事。

4.1 五次迭代

(i) 敵手無法藉由觀察密文來恢復秘密金鑰。

夠強嗎？不夠。反例：「什麼都不做」方案（$\mathsf{Enc}(\mathsf{sk}, m) = m$）根本不使用金鑰，敵手當然無法從密文「恢復」金鑰——但訊息完全曝光。教訓：金鑰不是我們真正想保護的東西，訊息才是。以「金鑰不被恢復」作為目標，瞄準錯了對象。（形式名稱：key recovery security，KR。）

(ii) 敵手無法藉由觀察密文來恢復訊息。

夠強嗎？不夠。反例：想像一個方案讓敵手只能恢復訊息的第一個字母或第一個單字。「無法恢復（整個）訊息」被滿足了，但洩漏第一個字顯然不可接受——「ATTACK AT DAWN」與「RETREAT AT DAWN」的第一個字母就足以決定戰局。（形式名稱：one-wayness，OW。）

(iii) 敵手無法藉由觀察密文來恢復訊息的任何部分。

夠強嗎？不夠。反例：想像一個方案不洩漏訊息的任何一個位元，卻洩漏統計資訊——例如每個字母出現的次數。字母頻率足以攻破所有單表代換密碼；更一般地，「這兩封信是不是同一封」、「這封信比平常長很多」等資訊都可能致命。教訓：「部分」太狹隘；敵手學到的可能是關於訊息的函數值，而不是訊息的子字串。

(iv) 敵手無法藉由觀察密文來推論出關於訊息的任何進一步「有意義的」或「新的」資訊。

夠強嗎？接近了（Getting warmer）。講義承認這不是形式化的敘述（「有意義」、「新的」要怎麼定義？），但它抓到了正確的直覺：前三次迭代其實都是這句話的特例。形式化之後就是 Goldwasser–Micali（1982）的**語意安全**（semantic security），以及與之等價、更好操作的**不可區分性**（indistinguishability）——見第 4.3 小節。講義接著問：如果敵手的能力更強呢？

(v) 敵手即使能夠讓任意選定的訊息被加密、任意選定的密文被解密，仍無法推論出關於（其他）訊息的任何有意義的新資訊。

夠強嗎？是 *。這涵蓋了我們預期會遇到的敵手：**選擇明文攻擊**（chosen plaintext attack, CPA）與**選擇密文攻擊**（chosen ciphertext attack, CCA）。星號的意思是：這真的「合理」嗎？給敵手這麼大的能力是否過於悲觀？答案見下方的「現實中的預言機」。

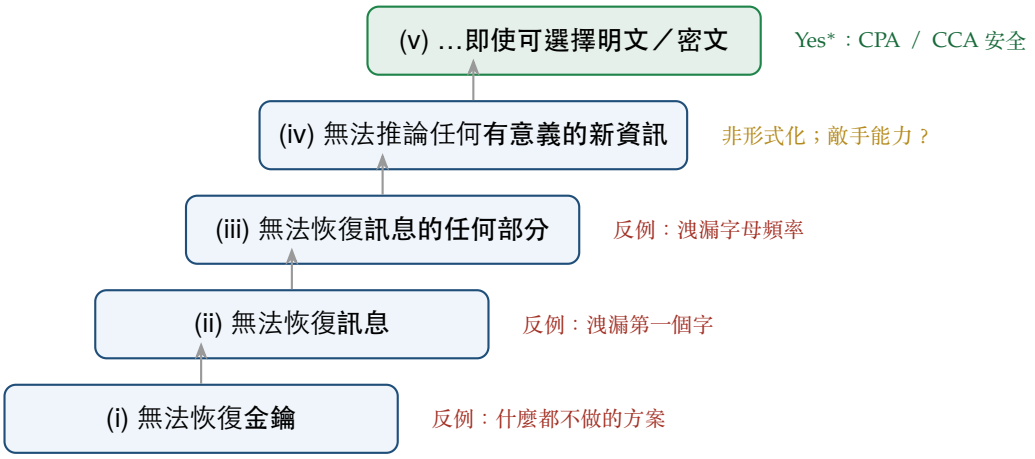


圖 3　「好的加密」定義的五次迭代：每一階都因為一個反例而被迫上修。

現實中的「加密預言機」與「解密預言機」

為什麼允許敵手「讓選定的訊息被加密」是合理的？因為現實系統常常主動替敵手加密：

- **中途島（1942）**：美軍懷疑日軍密電中的目標代號「AF」是中途島，便命令中途島守軍以明文發報「淡水蒸餾設備故障、缺水」。不久美軍截獲日軍情報「AF 缺水」——美軍成功讓敵方「加密了自己選定的明文」，證實 AF 即中途島。這是教科書級的**選擇明文攻擊**。
- **網頁 cookie**：伺服器把使用者的資料加密後存成 cookie；使用者（可能是敵手）可以藉由修改自己的帳號名稱、購物車內容等，讓伺服器加密幾乎任意的明文。
- **解密預言機**：伺服器收到密文後若回報「解密失敗 / 填充格式錯誤」，就等於洩漏了解密結果的一部分。Bleichenbacher（1998）對 RSA PKCS#1 v1.5 與 Vaudenay（2002）對 CBC 填充的**填充預言機攻擊**（padding oracle attack）都是用這種「一個位元的解密預言機」完整還原明文。

所以「Yes*」的星號答案是：**是的，這很現實**。寧可悲觀地假設敵手很強。

4.2　安全定義的兩個面向：(∗) 與 (†)

講義 p. 17 把上面的練習抽象成一句話：「定義安全是微妙的 !」並歸納出任何安全定義都隱含的兩個面向。這是全週（也是整門課）的核心框架，Problem Set 1 的 (iv) 小題會直接考它。

安全定義的兩個面向

一個安全定義由兩個部分組成：

- **(∗) 什麼事情不應該可能發生**（what should not be possible）——敵手的**目標**或「獲勝條件」。例如：推論出關於訊息的有意義資訊；偽造一個合法的標籤。
- **(†) 在什麼能力之下**（given what capabilities）——敵手的**能力**或「攻擊模型」。例如：只能觀察密文；或額外能讓選定的訊息被加密、選定的密文被解密。

用（∗）與（†）比較定義的強弱

固定其中一個面向，就能直覺地比較定義：

- （†）給敵手**越多**能力 ⇒ 定義**越強**（越難達成），因為方案必須抵擋更強的敵手。
- （∗）敵手的獲勝條件**越容易**達到（例如「恢復任何部分資訊」比「恢復整把金鑰」容易）⇒ 定義**越強**，因為方案必須連「小勝」都不讓敵手拿到。

五次迭代中，(i)→(iv) 是在放寬（∗）——讓敵手越來越容易「算贏」；(iv)→(v) 則是在加強（†）。

術語：confidentiality、secrecy、privacy

講義指出，這類「敵手學不到訊息內容」的性質常被統稱為**機密性／保密性**(confidentiality)，有些文獻偏好 **secrecy**。KL-IMC 多用 secrecy，Bellare 等人的論文多用 **privacy**。三者在本課程中視為同義。

4.3 延伸補充：如何把「學不到任何新資訊」形式化？

講義把形式化留給後續週次與 KL-IMC §§1–3，這裡先給一個「預告」，幫助你理解 (iv)、(v) 到底在說什麼。

完美保密（Shannon, 1949；KL-IMC 第 2 章）。 最強的版本：密文與訊息在統計上完全獨立——對任意訊息分布、任意 m 與 c，

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

敵手看到密文後對訊息的「後驗信念」與「先驗信念」一模一樣，等於什麼都沒學到。一次性密碼本（one-time pad）達成此性質，但 Shannon 證明任何完美保密方案的金鑰必須至少與訊息一樣長且不能重複使用——實務上太昂貴，於是現代密碼學改採「計算上」的版本。

不可區分性（indistinguishability；KL-IMC §3.2）。 把「學不到新資訊」改寫成一場遊戲（實驗，experiment）：

1. 挑戰者執行 $\mathsf{sk} \leftarrow \mathsf{KGen}$。
2. 敵手 $\mathcal{A}$ 選兩個等長的訊息 m_0, m_1 交給挑戰者。
3. 挑戰者擲一枚公平硬幣 $b \in \{0,1\}$，回傳 $c^* \leftarrow \mathsf{Enc}(\mathsf{sk}, m_b)$。
4. $\mathcal{A}$ 輸出猜測 b'；若 $b' = b$ 則 $\mathcal{A}$ 獲勝。

方案安全，若所有（計算能力有限的）敵手獲勝機率都只比亂猜的 $\frac{1}{2}$ 高出一個可忽略的量。直覺：如果連「這是 m_0 還是 m_1 的密文」這種一個位元的資訊都猜不準，那麼敵手就不可能學到任何更複雜的資訊。

- 只給敵手上面這些 ⇒ **EAV-security**（竊聽者安全；對應迭代 (iv)）。
- 另外給敵手一個加密預言機 $\mathsf{Enc}(\mathsf{sk},\cdot)$ ⇒ **IND-CPA**（KL-IMC §3.4）。
- 再給一個解密預言機 $\mathsf{Dec}(\mathsf{sk},\cdot)$（不得查詢 c^* 本身）⇒ **IND-CCA**（KL-IMC §5.1）。對應迭代 (v)。

⚠ 什麼是「允許」洩漏的？

再強的定義也必須允許某些事：(1) **訊息長度**（遊戲要求 $|m_0| = |m_1|$，因為一般的加密無法隱藏長度——隱藏長度需要填充，代價很高）；(2) **「有通訊發生」這件事本身**（流量分析屬於另一個領域）。把這些排除在外並不是漏洞，而是誠實地宣告定義的邊界——再次呼應「to define is to limit」。

🎓 考試重點

- 能依序說出五次迭代，**並為每一次給出講義的反例**（這類「為什麼 X 不夠強」的題目很適合筆試）。
- 能明確說出一個安全定義的兩個面向（$*$）、(†)，並用它們分析任何給定的定義。
- 知道 CPA / CCA 中「預言機」的現實對應（中途島、cookie、padding oracle）。
- 知道不可區分性遊戲的大致流程，以及為何要求 $|m_0| = |m_1|$。

5 保密性之外：完整性與訊息鑑別碼（講義 pp. 18–20）

5.1 被收買的騎士：保密不等於不可竄改

講義 p. 18 問：除了保密性之外，我們還想要什麼安全性質？設想騎士被收買，竄改羊皮紙上的密文。他不知道金鑰也不知道訊息，但他仍然可以重排、增加或刪除字元。Henry 解密之後會得到什麼？一團亂碼嗎？如果不是亂碼呢？

🔍 「保密」的方案可以被精準竄改：一次性密碼本的可延展性

一次性密碼本（one-time pad）是完美保密的：$c = m \oplus \mathsf{sk}$。但考慮 Francis 傳送

$$m = \texttt{PAY 100 GOLD}$$

騎士雖然讀不懂 c，卻知道（或猜到）訊息的格式。他把 c 中對應「1」的那個位元組 c_i 改成 $c_i' = c_i \oplus (1 \oplus 9)$。Henry 解密得到

$$c_i' \oplus \mathsf{sk}_i = m_i \oplus (1 \oplus 9) = 1 \oplus 1 \oplus 9 = 9,$$

也就是 `PAY 900 GOLD`——完全「有意義」，而且 Henry 毫無察覺。這個性質稱為**可延展性**（malleability）：敵手不必知道明文，就能對明文做可預測的修改。所有以 XOR 為基礎的串流加密（包括現代的 CTR 模式、ChaCha20）都有同樣性質；KL-IMC §4.1.2 另外指出，ECB 模式可以藉由重排密文區塊來重排明文區塊、刪除區塊來截短訊息。

保密性與完整性是兩種不同的性質（講義 p. 19）

- **保密性 $\not\Rightarrow$ 完整性**：一次性密碼本完美保密，卻任人竄改（上例）。
- **完整性 $\not\Rightarrow$ 保密性**：把訊息明文連同一個鑑別標籤一起送出，任何人都讀得到訊息，但沒人改得動它。

因此，「收到的訊息或密文就是對方送出的那一個」是一個**獨立且新的**安全性質，稱為**完整性**（integrity）。它必須用**不同的手段**達成——這正是 Problem Set 1 第一題的主題。

5.2　達成完整性的兩條路：工程 vs. 密碼學

講義 p. 19 列出 Francis 可以怎麼做：

- **工程方法**（engineering approach）：用擦不掉的墨水、填滿羊皮紙所有空白、使用防竄改（tamper-resistant）的羊皮紙……現代對應：防拆封裝、硬體安全模組（HSM）、Secure Enclave。這些有用，但每一種都是「針對特定竄改手法」的防禦，難以窮舉。
- **密碼學方法**（cryptographic approach）：
  1. 用另一把共享的秘密**完整性金鑰** ik，對密文計算一個**標籤**（tag）τ；
  2. 把密文與標籤一起寫在羊皮紙上；
  3. Henry 先用 ik 檢查標籤是否正確，通過之後才解密。

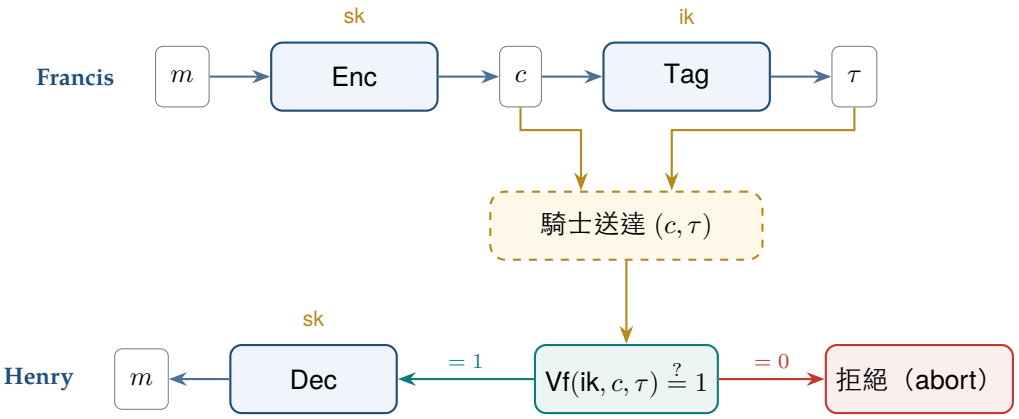


圖 4　講義的「密碼學方法」：先加密、再對密文貼標籤；收件端先驗證、通過才解密。這就是 **Encrypt-then-MAC**。

5.3 訊息鑑別碼（MAC）的語法（講義 p. 20）

MAC 的語法

一個**訊息鑑別碼**（message authentication code, MAC）是三個演算法的三元組 $\mathsf{MAC} = (\mathsf{KGen}, \mathsf{Tag}, \mathsf{Vf})$：

- KGen：輸出一把完整性金鑰 $\mathsf{ik} \in \mathcal{K}$：$\mathsf{ik} \leftarrow \mathsf{KGen}$。
- Tag：輸入完整性金鑰 ik 與訊息 $m \in \mathcal{M}$，輸出標籤 $\tau \in \mathcal{T}$：$\tau = \mathsf{Tag}(\mathsf{ik}, m)$（可隨機化：$\tau \leftarrow \mathsf{Tag}(\mathsf{ik}, m)$）。
- Vf（verify）：輸入 ik、訊息 m 與標籤 τ，輸出 0 或 1：$b = \mathsf{Vf}(\mathsf{ik}, m, \tau)$。輸出 1 時稱 τ 是 m 在 ik 下的**合法標籤**（valid tag），否則為不合法。

講義再度提醒：「同樣地，這裡沒有談正確性或安全性，只有語法。」——這兩者正是 Problem 1 (i) 與 (iv)–(viii) 要你補上的。

為什麼要用另一把金鑰 ik？——金鑰分離原則

講義明確寫「another shared secret integrity key」。原則上，**一把金鑰只做一件事**（key separation）。若加密與 MAC 共用同一把金鑰，兩個演算法之間可能產生意想不到的交互作用（例如用 CBC 加密、又用同一金鑰做 CBC-MAC，標籤會等於最後一個密文區塊，整個方案被瞬間攻破）。使用獨立金鑰 $(\mathsf{sk}, \mathsf{ik})$ 可以讓我們模組化地論證安全：只要 SKE 保密、MAC 不可偽造，組合就安全（Bellare–Namprempre 2000）。實務上通常由一把主金鑰透過金鑰衍生函數（KDF）衍生出 sk 與 ik。

5.4 延伸補充：MAC 在現實中的樣貌

- **確定性 MAC 與典範驗證**（canonical verification）：實務上幾乎所有 MAC（HMAC、CMAC、GMAC、Poly1305）都是確定性的，而 Vf 就是「重新計算 $\tau' = \mathsf{Tag}(\mathsf{ik}, m)$，比較是否等於 τ」。KL-IMC 稱之為**典範驗證**；Problem 1 (viii) 會證明在這種情況下，給敵手驗證預言機並不會讓它更強。
- **不只用於通訊**：KL-IMC §4.2 指出 MAC 的另一個典型場景是「自己與未來的自己」通訊——伺服器在使用者瀏覽器中存放的 cookie、存在不可信儲存裝置上的資料，都可用 MAC 防止被竄改（不一定需要加密）。
- **與數位簽章的差別**：MAC 是對稱的，驗證者也能產生標籤，所以 MAC 不能向第三方證明「這是 Alice 寫的」（Bob 也能偽造）。需要「不可否認性」時要用數位簽章（公鑰密碼學，課程後半）。

考試重點

- 能舉例說明「保密不蘊含完整性」（一次性密碼本的位元翻轉）與「完整性不蘊含保密性」（明文加標籤）。
- 能默寫 MAC 的語法；能畫出或描述 Encrypt-then-MAC 的傳送 / 接收流程，並強調「**先驗證、後解密**」。
- 知道為何加密與 MAC 要使用獨立的金鑰。

6 保密性與完整性之外（!）：重放、重排、反射、延遲、拒絕（講義 pp. 21–22）

假設 Francis 與 Henry 已經有了「完美」的加密與「完美」的 MAC：騎士讀不懂、也改不動任何一張羊皮紙。講義 p. 21 問：還有沒有我們想要的安全性質？有——因為騎士仍然可以對整張羊皮紙動手腳，而不必動其中的任何一個字。

6.1 五種攻擊

攻擊	騎士做了什麼	現代例子／後果
重放（replay）	把一張合法的羊皮紙複製並送達兩次（或留一份下次再送）。	「轉帳 1,000 鎊給 Bob」被重送十次，變成 10,000 鎊（KL-IMC §4.2 的例子）；重放「開門」指令；重放登入憑證。
重排（reordering）	先送第 2 封、再送第 1 封。	「賣出股票」與「買入股票」順序對調；「同意」與「撤回」顛倒；分段傳送的檔案被打亂。
反射（reflection）	把 Francis 送給 Henry 的羊皮紙退回給 *Francis*。	因為雙方共用同一把金鑰，Francis 會驗證通過、解密成功，誤以為 Henry 說了這句話。在挑戰–回應協定中，反射可以讓敵手「用對方的答案回答對方的挑戰」而冒充身分。
延遲（delay）	把羊皮紙扣住一陣子才送。	「現在賣出」在價格崩盤後才送達；時間敏感的指令失效。
拒絕傳遞（denial）	根本不送、或選擇性地丟掉某幾封。	阻斷服務（DoS）、截斷攻擊（truncation：丟掉最後的「結束」訊息）。

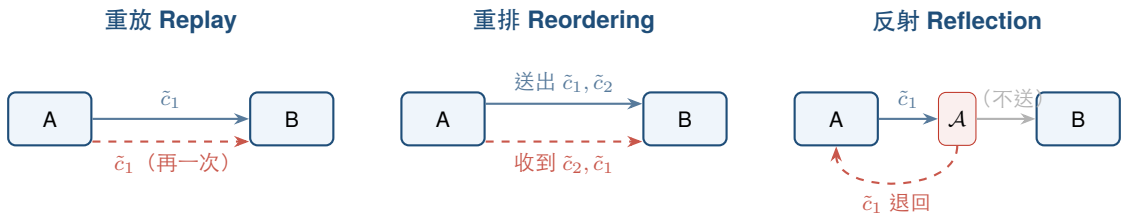


圖 5　三種「不需竄改任何一個位元」的攻擊。完美的加密與 MAC 都擋不住它們，因為每一張羊皮紙本身都是真的。

哪些屬於密碼學、哪些屬於網路工程？

講義把這五種攻擊分成兩組：

- **一般認為屬於密碼學範疇**：防止重放、防止重排、防止改變傳遞方向（反射）。這三者都可以藉由在訊息中加入被鑑別保護的額外資訊（計數器、方向位元、session 識別碼）來解決——Problem 2 (ii) 會詳細設計。
- **延伸到網路工程與安全**：防止延遲、防止拒絕傳遞。密碼學無法阻止敵手把封包丟掉或扣住——它控制的是通道，不是你。密碼學最多能讓這些行為被偵測（例如計數器出現缺口、逾時未收到回應），之後的處理（重傳、換路徑、中止連線）屬於協定與網路層的工作。Problem 2 (iii) 討論這點。

為什麼 MAC 本身不處理重放？因為 Vf 是**無狀態**（stateless）的：同一個合法的 (m, τ) 不管看幾次都會輸出 1。要擋重放，必須有人記得看過什麼，也就是需要狀態——而「哪些重複是可以接受的」是應用層才知道的事（KL-IMC §4.2）。

6.2　從中世紀隱喻到現代（講義 p. 22）

講義 p. 22 的提問

在現代，以下各項的合理對應是什麼？Francis 與 Henry？羊皮紙？騎士？保密性不足的攻擊？完整性的需求？重排、重放、反射、延遲與拒絕攻擊？

中世紀	現代對應（舉例）
Francis 與 Henry	你的瀏覽器與銀行伺服器；兩支手機上的 Signal / WhatsApp；兩個資料中心；一顆 IoT 感測器與雲端；也可以是「現在的你」與「明天要讀回硬碟的你」。
羊皮紙	一個 IP 封包、一筆 TLS record、一則訊息、一個檔案區塊。
騎士	整個網際網路：Wi-Fi 熱點、ISP、路由器、海底電纜、CDN——任何位於兩端之間、不受你控制的設備與組織。
保密性不足的攻擊	咖啡店 Wi-Fi 的封包嗅探；ISP 或政府的大規模監聽（2013 年 Snowden 揭露的計畫）；用字母頻率破解代換密碼；利用壓縮長度洩漏（CRIME / BREACH）。
完整性的需求	軟體更新不能被植入後門；網頁內容不能被 ISP 插入廣告或惡意程式；轉帳金額不能被改；韌體映像需驗證。
重放	重送已被授權的交易；重用攔截到的車鑰匙訊號（rolling code 就是為了防這個）；重放 Kerberos 票證。
重排	TCP 需要序號才能把亂序封包排回去；TLS record 層以隱含序號確保順序。
反射	早期的挑戰–回應認證協定（如 NTLM 的某些用法）可被反射攻擊；TLS 1.3 以不同方向用不同金鑰杜絕反射。
延遲與拒絕	DDoS 攻擊；BGP 劫持造成繞路；審查機制的封包丟棄；SSL 2.0 的截斷攻擊（敵手丟掉最後的封包，使接收端誤以為訊息已完整）。

考試重點

- 能定義並舉例五種攻擊，並區分「密碼學能處理」（重放、重排、反射）與「屬於網路工程」（延遲、拒絕）。
- 能說明為何「完美的加密 + 完美的 MAC」仍擋不住重放 / 重排 / 反射（每張羊皮紙都是真的；Vf 無狀態）。
- 能給出至少一組「中世紀 → 現代」的對應。

7　歷史插曲：密碼學曾是「軍火」（講義 p. 23）

講義中場休息的「fun fact」其實一點也不輕鬆：它是 Problem 3 的全部題材，也呼應 Blanchette 對密碼學的定義——為了規範行為而寫的程式碼。一旦程式碼能夠規範行為，政府就會想規範程式碼。

7.1　法律框架：AECA、ITAR 與美國軍火清單

- 美國《武器出口管制法》（Arms Export Control Act, **AECA**）授權總統管制「國防物品」的出口；實施細則是《國際武器貿易條例》（International Traffic in Arms Regulations, **ITAR**），由國務院主管。
- ITAR 的美國軍火清單（United States Munitions List, USML）第 **XIII** 類明列「密碼裝置與

軟體」，與飛彈、戰機、槍械並列。

- 「出口」的定義極廣：把軟體上傳到可被境外存取的網路、甚至對外國人揭露技術資料，都可能構成出口。違反者最高**十年徒刑或一百萬美元罰金**。
- 弔詭之處：印刷品不受管制（公開領域與出版品豁免）。同一份原始碼，印成書可以出口，存成磁片就是軍火。

7.2　三個案例

🏛 案例一：Netscape 的「美國版」與「國際版」

1990 年代中期，Netscape Navigator 內建的 SSL 讓線上刷卡成為可能。為了符合出口管制，Netscape 必須做兩個版本：

- **U.S. version**：完整金鑰長度——128 位元 RC4（或 3DES）對稱金鑰、1024 位元以上的 RSA。
- **International version**：刻意削弱——**40 位元**對稱金鑰、**512 位元** RSA（所謂的「export-grade」加密套件 `RSA_EXPORT`）。40 位元金鑰在當年用一台工作站幾天就能暴力破解。

取得美國版手續繁瑣，連許多美國使用者最後拿到的都是國際版。**後遺症延續了二十年**：2015 年公開的 **FREAK** 與 **Logjam** 攻擊，就是利用許多伺服器與瀏覽器仍然支援這些 1990 年代的 512 位元出口級套件，由中間人把連線「降級」到可破解的強度。刻意植入的弱點，比當初的政策活得更久。

🏛 案例二：Phil Zimmermann 與 PGP

- **1991**：軟體工程師 Philip Zimmermann 釋出 *Pretty Good Privacy*（PGP），第一個讓一般人用家用電腦做公鑰加密 email 的免費軟體；程式被（他人）張貼到 Usenet，數日內傳遍全球。
- **1993**：加州聖荷西的聯邦大陪審團展開刑事調查，審理 PGP 的全球散布是否違反 AECA / ITAR。Zimmermann 被告知他是調查目標。
- **1994 年 11 月 9 日**：Zimmermann 自歐洲演講返美，在華盛頓杜勒斯機場被海關特別探員攔下、搜查行李並在無律師在場下長時間訊問 PGP 的出口問題。（講義簡化為「被逮捕」；精確地說是被**攔查訊問**，當時並未遭正式逮捕或起訴。）
- **1995**：MIT Press 出版《*PGP Source Code and Internals*》——整本書就是 PGP 的完整原始碼，以 OCR 友善字體排印。因為書可以合法出口，境外讀者掃描後即可重建軟體。這是對「磁片是軍火、紙本是言論」這條界線的公開挑釁。
- **1996 年 1 月 11 日**：聯邦檢察官宣布結束歷時三年的調查，不予起訴，未說明理由。

Stay（1997）的法律評論即以此案為軸，檢討 ITAR 在第一修正案（言論自由）、第五修正案（法律明確性，vagueness）與「政治問題原則」（political question doctrine）下的合憲性。

詳見 Problem 3。

案例三：*Junger v. Daley*（6th Cir. 2000）——原始碼是受保護的言論

俄亥俄州 Case Western 大學法學教授 Peter Junger 想把自己寫的加密程式原始碼放到教學網站上，供「電腦與法律」課程使用，卻因此需要出口許可。

- **1996 年 11 月**：柯林頓簽署第 13026 號行政命令，把非軍用加密產品的出口管轄從國務院（ITAR）移轉到商務部（EAR），但明言管制理由是軟體的「**功能性**」而非「資訊價值」。
- **1998**：地方法院判 Junger 敗訴，認為加密原始碼「不具足夠表達性」，不受第一修正案保護。
- **2000 年 4 月**：第六巡迴上訴法院**逆轉**：「對熟悉程式語言的人而言，原始碼是傳達密碼學想法最有效、最精確的方式」，因此**原始碼是第一修正案保護的言論**；原始碼的功能性應在審查政府管制是否正當時納入考量（中度審查標準），而非用來否定其言論地位。案件發回重審。
- 幾乎同時（2000 年 1 月），商務部大幅放寬加密出口規則，爭議在實務上失去急迫性。

Harvard Law Review 的案例評析（Fromer, 2001）則批評第六巡迴法院「一切原始碼皆為言論」的結論過於寬泛，偏離最高法院對「象徵性行為」的判準（*Spence v. Washington*）：純粹為了執行而寫、無意與人溝通的程式碼，未必是言論。

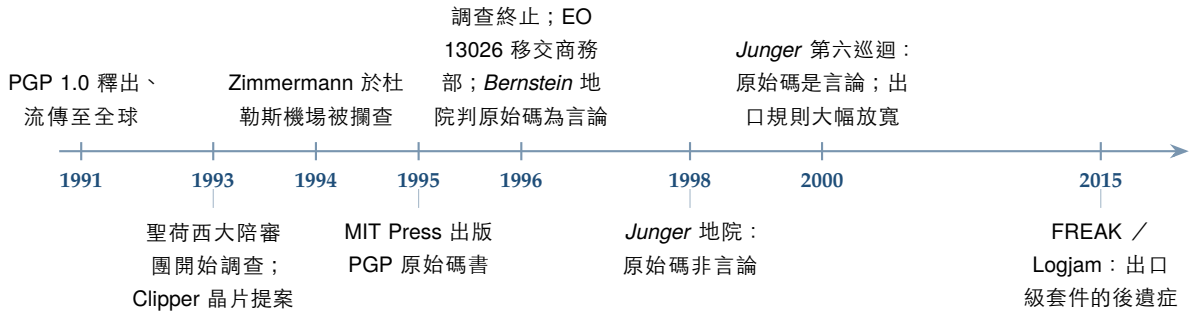


圖 6　「密碼戰爭」（Crypto Wars）第一回合的時間軸。

7.3　為什麼這段歷史屬於密碼學課程？

技術物不會自動保持「純技術」

- **密碼學是關於權力的**：誰能讀、誰能寫、誰能證明——這正是 Rosulek「控制資訊存取」的定義。控制權的分配從來不是純技術問題。
- **同一個物件有多重身分**：一段 RSA 程式碼同時是數學、是言論、是商品、是武器。法律必須決定用哪個身分看待它，而這個決定會改變人們能做什麼。

- **刻意的弱點是永久的**：40 位元與 512 位元的出口級套件在政策結束後仍存活了二十年，最終成為 FREAK / Logjam 的入口。這對今日所有「只給好人用的後門」提案都是直接的教訓。
- **同樣的辯論仍在進行**：2016 年 Apple 與 FBI 的 San Bernardino 手機解鎖之爭、澳洲 2018 年《協助與存取法》、英國 2023 年《線上安全法》與歐盟「Chat Control」提案中關於端對端加密與客戶端掃描的爭議，都是同一場辯論的續集。

考試重點

筆試較少直接考歷史細節，但可能要求你「舉例說明密碼學的社會 / 法律面向」或「說明刻意削弱密碼學的風險」。記住三個關鍵字：**ITAR ／軍火清單**、**Netscape 國際版 40 位元**、***Junger v. Daley*：原始碼是言論**；以及一個後果：**FREAK ／ Logjam**。

8　機器、敵手與通用安全定義（講義 pp. 24–28）

8.1　從領主到「機器」

講義 p. 24 把兩位領主抽象成兩台機器（machines）$M_1(\mathsf{sk},\mathsf{ik})$ 與 $M_2(\mathsf{sk},\mathsf{ik})$——文獻中也叫 players、agents、principals、entities——它們共享加密金鑰 sk 與完整性金鑰 ik，來回傳送 $(c_1,\tau_1),(c_2,\tau_2),\ldots$。通道上另有一個敵手 $\mathcal{A}$。

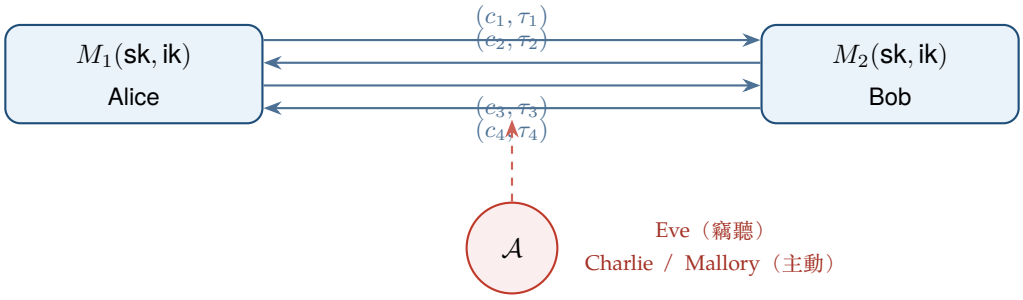


圖 7　講義 pp. 24–26 的通用圖像：兩台共享金鑰的機器，與一個位於通道上的敵手。

Alice、Bob 與那些名字

(M_1,M_2) 通常叫 **Alice** 與 **Bob**（源自 Rivest–Shamir–Adleman 1978 年的 RSA 論文）。敵手依能力取名：**Eve**（eavesdropper，只能竊聽）、**Charlie** 或 **Mallory**（malicious，能主動竄改、注入、重放）。文獻中還有 Trent（可信第三方）、Peggy / Victor（零知識證明中的 prover / verifier）等。

8.2　兩個通用的安全敘述（講義 pp. 25–26）

竊聽者安全：「Eve 無法推論出關於底層訊息的任何有意義資訊。」

若你確定敵手只能竊聽，這個性質在**任何場景**下都足夠——領主之間、你與朋友、你與銀

行、你與記者。這就是定義的「通用性」（generic）：同一條敘述不依賴應用的語意。但講義立刻補一句：「*Eve* 通常不被視為足夠強的敵手。」在真實網路上，能看到你封包的人幾乎都能改你的封包。

主動敵手安全：「**Charlie 可以讓選定的訊息被加密、選定的密文被解密，但仍無法推論出 Alice 與 Bob 之間底層訊息的任何有意義資訊。**」
這對應第 4 節的迭代 (v)，也就是 CPA / CCA 安全。

8.3 為什麼要用這種框架思考安全？（講義 p. 27）

安全定義框架的三大優點

1. **明確**：它明確地賦予敵手能力（†），並描述敵手能或不能達成什麼（∗）。沒有含糊地帶。
2. **共享的理解**：讓所有人對「安全」有同一個理解，而不是各自依賴模糊（且人人不同 !）的英文語意——「這個方案很安全」在沒有定義之前是一句空話。
3. **悲觀而通用**：允許我們對敵手悲觀（假設他們比實際上更強），因此證明出來的安全性可以通用地套用到各種場景。如果方案能擋住 Charlie，當然也能擋住 Eve。

8.4 延伸補充：安全定義在紙上長什麼樣子——以 MAC 為例

Problem Set 1 會要你比較好幾個 MAC 的安全定義。先看看一個標準定義的「長相」，之後讀題會輕鬆許多。KL-IMC Definition 4.2 把 MAC 的安全定義寫成一個實驗（experiment）$\mathsf{Mac\text{-}forge}_{\mathcal{A},\mathsf{MAC}}$：

實驗 $\mathsf{Mac\text{-}forge}_{\mathcal{A},\mathsf{MAC}}$（存在性不可偽造、適應性選擇訊息攻擊）

1. 產生金鑰 $\mathsf{ik} \leftarrow \mathsf{KGen}$。
2. 敵手 $\mathcal{A}$ 獲得預言機 $\mathsf{Tag}(\mathsf{ik}, \cdot)$ 的存取權（†）：可以反覆送入任何 m 並取得 $\mathsf{Tag}(\mathsf{ik}, m)$。令 $\mathcal{Q}$ 為 $\mathcal{A}$ 查詢過的所有訊息所成的集合。
3. $\mathcal{A}$ 最終輸出一對 (m^*, τ^*)。
4. $\mathcal{A}$ **獲勝**（實驗輸出 1）若且唯若（∗）：$\mathsf{Vf}(\mathsf{ik}, m^*, \tau^*) = 1$ 且 $m^* \notin \mathcal{Q}$。

定義：MAC 是安全的（existentially unforgeable under adaptive chosen-message attack，EUF-CMA），若對所有有效率的敵手 $\mathcal{A}$，$\Pr[\mathsf{Mac\text{-}forge}_{\mathcal{A},\mathsf{MAC}} = 1]$ 都是可忽略的（negligible）。

注意這個定義如何同時精確地寫出（†）（第 2 步）與（∗）（第 4 步）。Problem 1 的 (D3) 就是這個定義的非正式版本。「存在性」（existential）指敵手只要對任何一個新訊息偽造成功就算贏，哪怕那個訊息毫無意義——因為「有意義」是應用層才知道的事，定義不該依賴它。

8.5　第一次 tutorial 與 Problem Set 預告（講義 p. 28）

教師邀請同學帶著問題來 tutorial（也可以先寫在 KEATS），可討論的主題包括：安全性質的形式化、「合理」的敵手能力、通訊以外的密碼學、具體的真實世界場景等。**第一份 Problem Set 聚焦於不同場景中的保密性與完整性**——也就是本教材第二部分的全部內容。

考試重點

- 能說明 Eve 與 Charlie 的差別，以及為何「只防 Eve」通常不夠。
- 能複述安全定義框架的三大優點（明確、共享理解、悲觀而通用）。
- 能用（$*$）/（†）的語言讀懂（並寫出）一個以實驗形式呈現的安全定義。

第二部分

Problem Set 1 詳解

Problem 1：On integrity • Problem 2：On encryption • Problem 3：On cryptography as munitions

9 Problem 1：On integrity（論完整性）

題目背景

我們要把第一堂課中「以密碼學方法達成完整性」的草圖形式化。回憶：一個訊息鑑別碼（MAC）是三個演算法 $\mathsf{MAC} = (\mathsf{KGen}, \mathsf{Tag}, \mathsf{Vf})$：

- KGen 輸出完整性金鑰 $\boxed{\mathsf{ik}}$；
- $\mathsf{Tag}(\mathsf{ik}, m)$ 輸入 ik 與訊息 m，輸出標籤 τ；
- $\mathsf{Vf}(\mathsf{ik}, m, \tau)$ 輸入 ik、m、τ，輸出 0 或 1。

直覺：Alice 與 Bob 共享 ik。Alice 要送 m 時計算 $\tau = \mathsf{Tag}(\mathsf{ik}, m)$ 並送出 (m, τ)；Bob 檢查 $\mathsf{Vf}(\mathsf{ik}, m, \tau) = 1$。若 Vf 輸出 1，稱 τ 是 m 在 ik 下的**合法**（valid）標籤，否則為不合法。

9.1 (i) MAC 的正確性應該是什麼意思？

(i)

What should correctness mean for a MAC? （可參考第一堂課對 SKE 正確性的定義。）

解答

仿照 SKE 的正確性「用 sk 加密 m 得到 c，則用同一把 sk 解密 c 會得到 m」，MAC 的正確性應該是：

誠實產生的標籤，一定會被（持有同一把金鑰的）驗證者接受。

形式化敘述。對每一把可能由 KGen 輸出的金鑰 ik，以及每一個訊息 $m \in \mathcal{M}$：

$$\text{若} \tau = \mathsf{Tag}(\mathsf{ik}, m)\ (\text{或} \tau \leftarrow \mathsf{Tag}(\mathsf{ik}, m))\ , \quad \text{則} \mathsf{Vf}(\mathsf{ik}, m, \tau) = 1.$$

用一行寫：

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

幾點說明。

1. **若 Tag 是隨機化的**（同一個 m 可能有多個合法標籤），上式要求對 Tag 的每一種隨機選擇都成立；即 $\Pr\big[\mathsf{Vf}(\mathsf{ik}, m, \mathsf{Tag}(\mathsf{ik}, m)) = 1\big] = 1$，機率取自 KGen 與 Tag 的隨機性。較寬鬆的版本只要求此機率「壓倒性地接近 1」。
2. **正確性只談「合法的標籤被接受」**，完全沒有說「不合法的標籤會被拒絕」。後者是安全性（不可偽造性）的範疇，正是本題 (iv)–(viii) 要處理的。一個 Vf 永遠輸出 1 的 MAC 是完全正確的——也完全沒用。
3. **與 SKE 的對照**：SKE 的正確性是「$\mathsf{Dec} \circ \mathsf{Enc} = \mathrm{id}$」；MAC 的正確性是「$\mathsf{Vf}(\cdot, \cdot, \mathsf{Tag}(\cdot, \cdot)) \equiv 1$」。兩者都只保證**功能**（持有金鑰的合法使用者能正常運作），不保證安全。

9.2 (ii) 設計一個正確（但不必安全）的 MAC

(ii)

Design a correct MAC. It does not have to satisfy any security properties.

解答

由 (i) 可知，只要「Vf 接受 Tag 的輸出」即可。以下給三個由淺入深的例子（任何一個都是完整的答案）。

構造 A：常數標籤 MAC（「什麼都不做」的 MAC）。

- KGen：輸出任意 ik（例如均勻隨機的 $\mathsf{ik} \in \{0,1\}^n$，或乾脆 $\mathsf{ik} = \perp$）。
- $\mathsf{Tag}(\mathsf{ik}, m) = 0$（不論 ik、$m$ 為何都輸出常數 0）。
- $\mathsf{Vf}(\mathsf{ik}, m, \tau) = 1$ 若且唯若 $\tau = 0$。

正確性：對任何 ik、m，$\mathsf{Tag}(\mathsf{ik}, m) = 0$，而 $\mathsf{Vf}(\mathsf{ik}, m, 0) = 1$。✓
安全性：毫無。任何人對任何訊息都能「偽造」合法標籤 0。這是講義 p. 12「do nothing」加密方案在 MAC 世界的翻版；它在 (v) 會派上用場，因為它滿足 (D1)（金鑰根本沒被用到，當然無法被恢復）卻完全不滿足 (D3)。

構造 B：回聲 MAC。 $\mathsf{Tag}(\mathsf{ik}, m) = m$，$\mathsf{Vf}(\mathsf{ik}, m, \tau) = 1 \iff \tau = m$。正確；但敵手把訊息抄一遍就是合法標籤。

構造 C：一次性密碼本式 MAC（看起來有用到金鑰，但一樣不安全）。 令 $\mathcal{M} = \mathcal{T} = \mathcal{K} = \{0,1\}^n$。

- KGen：$\mathsf{ik} \leftarrow \{0,1\}^n$ 均勻隨機。
- $\mathsf{Tag}(\mathsf{ik}, m) = \mathsf{ik} \oplus m$。
- $\mathsf{Vf}(\mathsf{ik}, m, \tau) = 1 \iff \tau = \mathsf{ik} \oplus m$（典範驗證）。

正確性：$\mathsf{Vf}(\mathsf{ik}, m, \mathsf{ik} \oplus m)$ 檢查 $\mathsf{ik} \oplus m \stackrel{?}{=} \mathsf{ik} \oplus m$，恆為 1。✓
安全性：敵手只要看到一對 (m_1, τ_1) 就能算出 $\mathsf{ik} = \tau_1 \oplus m_1$，之後對任何 m 都能偽造 $\tau = \mathsf{ik} \oplus m$。它連最弱的 (D1) 都不滿足。

構造	正確？	滿足 (D1)？	滿足 (D3)？	備註
A：常數標籤	✓	✓	✗	金鑰未使用 ⇒ 無法恢復，但人人可偽造
B：回聲	✓	✓	✗	同上
C：$\mathsf{ik} \oplus m$	✓	✗	✗	一對訊息–標籤即洩漏金鑰

9.3 (iii) 用 MAC 保護密文的完整性：Encrypt-then-MAC

(iii)

In the first recorded lecture we informally used a MAC to ensure the integrity of ciphertexts. How would we achieve this in the syntax of $\mathsf{MAC} = (\mathsf{KGen}, \mathsf{Tag}, \mathsf{Vf})$ given above?

解答

給定一個 $\mathsf{SKE} = (\mathsf{KGen}_E, \mathsf{Enc}, \mathsf{Dec})$ 與一個 $\mathsf{MAC} = (\mathsf{KGen}_M, \mathsf{Tag}, \mathsf{Vf})$，其中 MAC 的訊息空間必須包含 SKE 的密文空間（$\mathcal{C} \subseteq \mathcal{M}_{\mathsf{MAC}}$，因為我們要對密文貼標籤）。組合方案如下：

Encrypt-then-MAC（EtM）

金鑰產生。獨立執行 $\mathsf{sk} \leftarrow \mathsf{KGen}_E$ 與 $\mathsf{ik} \leftarrow \mathsf{KGen}_M$；雙方共享 $(\mathsf{sk}, \mathsf{ik})$。

傳送訊息 m。

1. $c = \mathsf{Enc}(\mathsf{sk}, m)$； （先加密）
2. $\tau = \mathsf{Tag}(\mathsf{ik}, c)$； （再對密文貼標籤）
3. 送出 $\tilde{c} = (c, \tau)$。

接收 $\tilde{c} = (c, \tau)$。

1. 若 $\mathsf{Vf}(\mathsf{ik}, c, \tau) \neq 1$，**立即中止**（輸出 $\perp$，不做任何其他事）； （先驗證）
2. 否則輸出 $m = \mathsf{Dec}(\mathsf{sk}, c)$。 （通過才解密）

這恰好就是 Problem 2 題幹所給的流程，也是圖 4 畫的東西。若要把它寫成「一個新的 SKE」，令 $\mathsf{SKE}' = (\mathsf{KGen}', \mathsf{Enc}', \mathsf{Dec}')$：$\mathsf{KGen}'$ 輸出 $(\mathsf{sk}, \mathsf{ik})$；$\mathsf{Enc}'((\mathsf{sk}, \mathsf{ik}), m) = (\mathsf{Enc}(\mathsf{sk}, m), \mathsf{Tag}(\mathsf{ik}, c))$；$\mathsf{Dec}'((\mathsf{sk}, \mathsf{ik}), (c, \tau))$ 在 Vf 失敗時輸出 $\perp$、否則輸出 $\mathsf{Dec}(\mathsf{sk}, c)$。這種同時提供保密與完整性的 SKE 稱為**鑑別加密**（authenticated encryption, AE）。

三個設計要點。

1. **標籤算在密文上，不是明文上。**這樣接收端可以在碰觸任何未經鑑別的資料之前就把竄改過的密文丟掉；同時 τ 是密文的函數，不會額外洩漏明文資訊。
2. **兩把獨立的金鑰。**sk 與 ik 分開產生（或由主金鑰以 KDF 衍生），遵守金鑰分離原則（見第 5 節）。
3. **先驗證、後解密；失敗就中止。**任何「解密後再看看對不對」的做法都會把解密結果（或錯誤訊息、時間差）暴露給敵手，成為解密預言機。

延伸：為什麼是「Encrypt-*then*-MAC」，而不是其他順序？把一個加密方案與一個 MAC「黏」在一起有三種自然方式。Bellare 與 Namprempre（ASIACRYPT 2000）在「Enc 是 CPA 安全、MAC 不可偽造」的通用假設下，逐一分析：

組合	做法	通用結論（B–N 2000）	實務例子
Encrypt-and-MAC（E&M）	$c = \mathsf{Enc}(\mathsf{sk}, m)$，$\tau = \mathsf{Tag}(\mathsf{ik}, m)$，送 (c, τ)	❌ 連保密性都不保證：MAC 不必隱藏輸入，確定性 MAC 會洩漏「兩則訊息是否相同」；且須先解密才能驗證	SSH（原始設計；後來加上對策）
MAC-then-Encrypt（MtE）	$\tau = \mathsf{Tag}(\mathsf{ik}, m)$，$c = \mathsf{Enc}(\mathsf{sk}, m \Vert \tau)$	⛔ 有 CPA 保密與明文完整性，但不通用地保證密文完整性 / CCA 安全；必須先解密再驗證	SSL/TLS $\leq$ 1.2——導致 padding oracle、Lucky 13、POODLE 等攻擊
Encrypt-then-MAC（EtM）	$c = \mathsf{Enc}(\mathsf{sk}, m)$，$\tau = \mathsf{Tag}(\mathsf{ik}, c)$	✅ 只要 MAC 是強不可偽造（見 (vii)），就通用地得到密文完整性與 CCA 安全	IPsec ESP；TLS 1.3 的 AEAD（GCM、ChaCha20-Poly1305 內部即為此結構）

Krawczyk（CRYPTO 2001）進一步指出 MtE 在特定加密方式（如 CBC、串流加密）下可以安全，但 EtM 是唯一**不需看底層細節就安全**的做法——這正是「通用組合」的價值。

9.4 (iv) 三個候選定義的（$*$）與（†）

候選定義與 (iv)

假設 MAC 正確，且某把 ik 已由 KGen 產生。稱敵手能「**對訊息 m 偽造標籤**」，若它能在不知道 ik 的情況下產生某個 τ 使 $\mathsf{Vf}(\mathsf{ik}, m, \tau) = 1$。

- (D1) 敵手不應能藉由觀察 (m_i, τ_i)（其中 $\tau_i = \mathsf{Tag}(\mathsf{ik}, m_i)$）來恢復 ik。
- (D2) 讓敵手能存取 $\mathsf{Tag}(\mathsf{ik}, \cdot)$。敵手不應能對任何訊息 m 偽造標籤。
- (D3) 讓敵手能存取 $\mathsf{Tag}(\mathsf{ik}, \cdot)$，令 $\mathcal{Q}$ 為敵手輸入過 $\mathsf{Tag}(\mathsf{ik}, \cdot)$ 的訊息集合。敵手不應能對任何不在 $\mathcal{Q}$ 中的訊息偽造標籤。

$\mathsf{Tag}(\mathsf{ik}, \cdot)$ 表示敵手擁有一個黑盒函數：輸入 m、輸出 $\mathsf{Tag}(\mathsf{ik}, m)$，但看不到內部、不知道 ik。回憶：安全定義由兩個元素組成——敵手不應能做到的事（$*$），以及敵手能力的描述（†）。

(iv) What are ($*$) and (†) for each of (D1)–(D3)?

解答

	（∗）不應該可能的事（敵手的目標）	（†）敵手的能力（攻擊模型）
(D1)	輸出完整性金鑰 ik（金鑰恢復，key recovery）。	被動觀察若干訊息–標籤對 (m_i, τ_i)，$\tau_i = \mathsf{Tag}(\mathsf{ik}, m_i)$。題目寫的是「observing」，最自然的解讀是訊息由誠實雙方決定、敵手只能看（已知訊息攻擊，known-message attack）；若把它加強成敵手可選 m_i，則與 (D2)/(D3) 的能力相同。
(D2)	輸出任一對 (m, τ) 使 $\mathsf{Vf}(\mathsf{ik}, m, \tau) = 1$——對任何 m 偽造，包括曾經查詢過的訊息。	適應性選擇訊息攻擊（adaptive chosen-message attack）：可任意次、依先前結果適應性地查詢黑盒 $\mathsf{Tag}(\mathsf{ik}, \cdot)$。
(D3)	輸出 (m, τ) 使 $\mathsf{Vf}(\mathsf{ik}, m, \tau) = 1$ 且 $m \notin \mathcal{Q}$——對一個新的（未查詢過的）訊息偽造。	同 (D2)：適應性選擇訊息攻擊，並記錄查詢集合 $\mathcal{Q}$。

觀察。

- (D2) 與 (D3) 的（†）完全相同，只差在（∗）：(D2) 的獲勝條件比 (D3) 寬鬆（少了 $m \notin \mathcal{Q}$ 的限制），所以敵手在 (D2) 中更容易贏，(D2) 對方案的要求更嚴。
- (D1) 的（∗）是三者中對敵手最苛刻的目標（要拿到整把金鑰），（†）也最弱（只能看）。兩個面向都偏向「對方案寬鬆」，預示它是最弱的定義。
- 這正是第 4.2 節的原則：（†）越強或（∗）越容易達成，定義就越強。

9.5 (v) 哪一個最強？能達成嗎？

定義的強弱與 (v)

要「達成」一個定義，必須證明存在滿足它的物件（這裡是 MAC）；要證明一個定義「無法達成」，必須證明不存在這樣的物件。**(DX) 至少與 (DY) 一樣強**（at least as strong），若每一個滿足 (DX) 的 MAC 也滿足 (DY)。**(DX) 嚴格比 (DY) 強**（stronger），若 (DX) 至少與 (DY) 一樣強，且存在一個滿足 (DY) 但不滿足 (DX) 的 MAC。

(v) What is the strongest definition of (D1)–(D3)? Can it be achieved?

解答

結論先行：(D2) 是最強的定義；但它無法被任何正確的 MAC 達成。三者的強弱關係是

$$\text{(D2)} \succ \text{(D3)} \succ \text{(D1)},$$

其中 $\succ$ 表示「嚴格比...強」。以下逐步證明。

引理 9.1（(D2) 至少與 (D3) 一樣強） 每個滿足 (D2) 的 MAC 都滿足 (D3)。

證明. 兩者的敵手能力相同。若 MAC 滿足 (D2)，則沒有敵手能對任何訊息偽造標籤；特別地，沒有敵手能對 $\mathcal{Q}$ 之外的訊息偽造標籤，即滿足 (D3)。（(D3) 的獲勝條件是 (D2) 獲勝條件的子集。）■

引理 9.2（(D3) 至少與 (D1) 一樣強） 每個滿足 (D3) 的 MAC 都滿足 (D1)。

證明. 採**對換證明**（contrapositive）：假設 MAC 不滿足 (D1)，即存在敵手 $\mathcal{A}$ 能從觀察到的 $(m_1,\tau_1),\dots,(m_q,\tau_q)$ 恢復出 ik。我們用 $\mathcal{A}$ 建造一個攻破 (D3) 的敵手 $\mathcal{B}$：

1. $\mathcal{B}$ 向黑盒 $\mathsf{Tag}(\mathsf{ik},\cdot)$ 查詢 $m_1,\dots,m_q$（若 (D1) 的訊息由誠實方依某分布產生，$\mathcal{B}$ 就依同一分布抽樣；黑盒存取涵蓋了被動觀察的能力），把 (m_i,τ_i) 交給 $\mathcal{A}$。
2. $\mathcal{A}$ 輸出 ik'（若 $\mathcal{A}$ 成功則 $\mathsf{ik}'=\mathsf{ik}$）。
3. $\mathcal{B}$ 任選一個 $m^*\notin\mathcal{Q}=\{m_1,\dots,m_q\}$，自行計算 $\tau^*=\mathsf{Tag}(\mathsf{ik}',m^*)$（演算法 Tag 是公開的——Kerckhoffs 原則），輸出 (m^*,τ^*)。

只要 $\mathcal{A}$ 成功，$\mathsf{ik}'=\mathsf{ik}$，由**正確性**得 $\mathsf{Vf}(\mathsf{ik},m^*,\tau^*)=1$，且 $m^*\notin\mathcal{Q}$，故 $\mathcal{B}$ 攻破 (D3)。因此 $\Pr[\mathcal{B}\text{ 攻破(D3)}]\geq\Pr[\mathcal{A}\text{ 攻破(D1)}]$：若 (D3) 成立，(D1) 必成立。■

推論 9.3 (D2) 至少與 (D1) 一樣強（由引理 9.1、9.2 傳遞）。

命題 9.4（(D3) 嚴格比 (D1) 強） 存在一個滿足 (D1) 但不滿足 (D3) 的 MAC。

證明. 取 (ii) 的構造 **A**（常數標籤 MAC），其中 KGen 輸出均勻隨機的 $\mathsf{ik}\in\{0,1\}^n$。滿足 *(D1)*：Tag 與 Vf 完全不使用 ik，敵手看到的一切都與 ik 統計獨立，猜中 ik 的機率是 2^{-n}，可忽略。不滿足 *(D3)*：敵手不做任何查詢（$\mathcal{Q}=\emptyset$），直接輸出 $(m,0)$，$\mathsf{Vf}(\mathsf{ik},m,0)=1$，必勝。■

定理 9.5（(D2) 無法達成） **沒有任何**正確的 MAC 滿足 (D2)。

證明. 令 MAC 為任一正確的 MAC。考慮以下敵手 $\mathcal{A}$：任選一個訊息 $m\in\mathcal{M}$，向黑盒查詢得到 $\tau=\mathsf{Tag}(\mathsf{ik},m)$，然後輸出 (m,τ)。由**正確性**（(i)），$\mathsf{Vf}(\mathsf{ik},m,\tau)=1$。依題目對「偽造」的定義——「在不知道 ik 的情況下產生某個 τ 使 $\mathsf{Vf}(\mathsf{ik},m,\tau)=1$」——$\mathcal{A}$ 確實不知道 ik，卻對 m 產生了合法標籤。(D2) 要求對任何 m 都不可能偽造，故 MAC 不滿足 (D2)。因為 MAC 是任意的，結論對所有正確的 MAC 成立。■

命題 9.6（(D2) 嚴格比 (D3) 強） (D2) 至少與 (D3) 一樣強（引理 9.1），且存在滿足 (D3) 而不滿足 (D2) 的 MAC。

證明. 由定理 9.5，任何滿足 (D3) 的 MAC 都自動是「滿足 (D3) 但不滿足 (D2)」的例子。因此只需知道**滿足 (D3) 的 MAC 存在**：在標準假設下（偽隨機函數存在，等價於單向函數存在），$\mathsf{Tag}(\mathsf{ik},m)=F_{\mathsf{ik}}(m)$（$F$ 為偽隨機函數，實務上如 HMAC、CMAC）即滿足 (D3)（KL-IMC Construction 4.5 / Theorem 4.6，課程稍後會證明）；在「只用一次」的情境下甚至有不依賴任何假設的資訊理論式 MAC（KL-IMC §4.6）。■

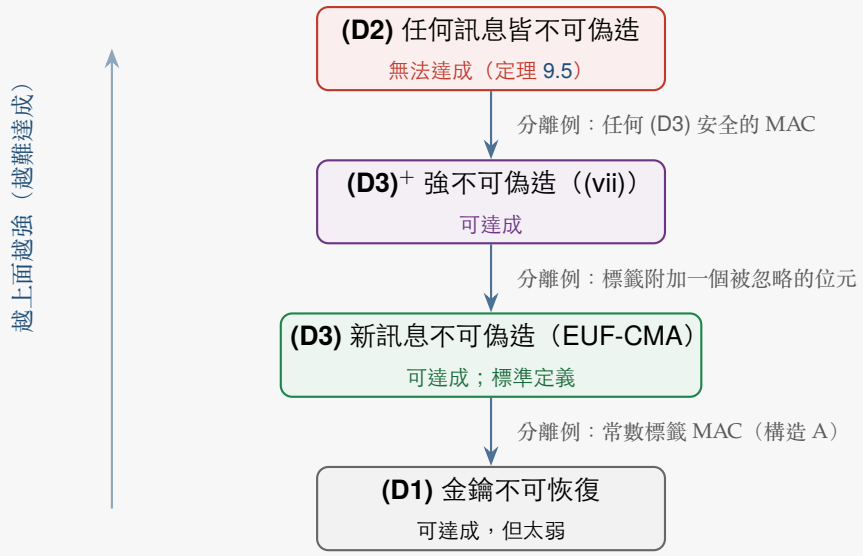


圖 8 四個 MAC 安全定義的強弱關係（Hasse 圖）。箭頭 $X \to Y$ 表示「滿足 X 蘊含滿足 Y」，箭頭旁為證明「嚴格」所需的分離例。

所以 (v) 的答案：最強的是 (D2)，但**無法達成**——不是因為我們不夠聰明，而是正確性本身就保證了「把黑盒的輸出照抄」是一次成功的偽造。這給我們一個重要教訓：

定義不能只追求「強」

一個定義若強到沒有任何物件能滿足它，就毫無用處。好的定義必須在「夠強（排除所有我們在意的攻擊）」與「可達成（存在滿足它的方案）」之間取得平衡。(D2) 把「重放黑盒的輸出」也算成攻擊，但重放是任何無狀態 MAC 都無法阻止的，所以必須由定義明確排除（(D3) 的 $m \notin \mathcal{Q}$），並交給更高層的協定處理（Problem 2 (ii)）。

9.6 (vi) 應該選哪一個作為 MAC 的安全定義？

(vi)

Which definition from (D1)–(D3) should we select as a security definition of a MAC? Why?

解答

選 **(D3)**。理由分三層：

1. **(D1) 太弱，而且瞄準錯了目標。** MAC 的任務是防止偽造，不是保護金鑰本身；金鑰只是達成目標的手段。命題 9.4 的常數標籤 MAC 完美滿足 (D1)，卻讓任何人都能偽造任何訊息——一個被 (D1) 認證為「安全」的方案在實際使用中毫無防護，這表示 (D1) 沒有捕捉到我們真正在意的事。這與講義迭代 (i)「無法恢復金鑰 ⇒ 什麼都不做的方案也算安全」是同一個教訓。

2. **(D2) 無法達成**（定理 9.5）。選一個沒有任何方案能滿足的定義，等於什麼都沒選。
3. **(D3) 是三者中最強的可達成定義**，而且它的兩個面向都「合理地悲觀」：
   - (†) 給敵手**適應性選擇訊息**的能力是現實的：敵手往往能影響誠實方要鑑別的內容（網頁 cookie、協定中由對方指定的欄位、中途島式的誘導），而且允許過強的能力不會有壞處——只會讓證明出來的安全性更通用。
   - (∗) 要求對**任何**新訊息都不可偽造（存在性不可偽造），而不只是「有意義的」訊息：什麼叫有意義是應用層才知道的事，定義若依賴它就無法通用（KL-IMC §4.2 的論證）。
   - 它恰好排除了唯一無法避免的「攻擊」——重放黑盒輸出。

(D3) 正是教科書的標準定義：**存在性不可偽造、適應性選擇訊息攻擊**（existential unforgeability under adaptive chosen-message attack，EUF-CMA；KL-IMC Definition 4.2；見第 8.4 節的實驗寫法）。
(D3) 刻意沒有涵蓋的兩件事（這不是缺陷，而是邊界）：

- **重放**：Vf 無狀態，同一個合法的 (m, τ) 永遠被接受。要防重放需要狀態（計數器、時間戳），由上層協定負責——Problem 2 (ii)。
- **對已查詢訊息產生新的標籤**：(D3) 不禁止敵手在看過 (m_1, τ_1) 後造出另一個 $\tau_1' \neq \tau_1$ 也能通過驗證。這在多數應用無害，但在某些組合（例如 Encrypt-then-MAC 要達到 CCA 安全）會成為漏洞——這就是 (vii) 的主題。

9.7 (vii) 至少與 (D3) 一樣強、但仍可達成的定義：強不可偽造性

(vii)

Can you think of a definition that is at least as strong as your answer to (vi.) but still achievable? （提示：Tag 可以是隨機化的，即對同一個訊息可能輸出多個合法標籤。考慮 (D3)。）

解答

提示點出關鍵：若 Tag 隨機化，同一個 m 可能有很多合法標籤。(D3) 只禁止敵手對新訊息偽造，卻允許敵手對舊訊息造出新標籤。把這個漏洞補起來，就得到：

(D3)$^+$：強不可偽造性（strong unforgeability，sEUF-CMA；KL-IMC Definition 4.3）

讓敵手存取 $\mathsf{Tag}(\mathsf{ik},\cdot)$，並令 $\mathcal{Q}^+$ 為敵手得到的所有**訊息–標籤對** (m_i,τ_i) 的集合（而不只是訊息）。敵手不應能輸出任何 (m,τ) 使得

$$\mathsf{Vf}(\mathsf{ik},m,\tau)=1 \quad \text{且} \quad (m,\tau)\notin\mathcal{Q}^+.$$

也就是：敵手不應能產生任何新的合法訊息–標籤對——包括對舊訊息的新標籤。

我們需要驗證三件事：它至少與 (D3) 一樣強、它嚴格更強、它可達成。

命題 9.7（(D3)$^+$ 至少與 (D3) 一樣強）

證明. 若 (m,τ) 是 (D3) 意義下的偽造（$\mathsf{Vf}=1$ 且 $m\notin\mathcal{Q}$），則 (m,τ) 顯然也不在 $\mathcal{Q}^+$ 中（$\mathcal{Q}^+$ 裡的訊息都在 $\mathcal{Q}$ 中），所以它也是 (D3)$^+$ 意義下的偽造。因此任何攻破 (D3) 的敵手都攻破 (D3)$^+$；對換即得「滿足 (D3)$^+$ $\Rightarrow$ 滿足 (D3)」。 ■

命題 9.8（(D3)$^+$ 嚴格比 (D3) 強） 存在滿足 (D3) 但不滿足 (D3)$^+$ 的 MAC。

證明. 取任一滿足 (D3) 的 MAC $(\mathsf{KGen},\mathsf{Tag},\mathsf{Vf})$，定義 MAC$'$：

$$\mathsf{Tag}'(\mathsf{ik},m)=\mathsf{Tag}(\mathsf{ik},m)\,\|\,b, \qquad b\leftarrow\{0,1\}\text{ 均勻隨機；}$$
$$\mathsf{Vf}'(\mathsf{ik},m,\tau\,\|\,b)=\mathsf{Vf}(\mathsf{ik},m,\tau) \qquad \text{（忽略最後一個位元）.}$$

MAC$'$ 仍滿足 (D3)：任何對 MAC$'$ 的新訊息偽造 $(m,\tau\,\|\,b)$，去掉 b 就是對 MAC 的新訊息偽造。但 MAC$'$ 不滿足 (D3)$^+$：敵手查詢一次得到 $(m_1,\tau_1\,\|\,b)$，輸出 $(m_1,\tau_1\,\|\,\bar{b})$（翻轉最後一位）。它通過驗證，且 $(m_1,\tau_1\,\|\,\bar{b})\notin\mathcal{Q}^+$，必勝。 ■

命題 9.9（(D3)$^+$ 可達成；KL-IMC Proposition 4.4） 若一個確定性的 MAC 使用典範驗證 $(\mathsf{Vf}(\mathsf{ik},m,\tau)=1 \iff \tau=\mathsf{Tag}(\mathsf{ik},m))$ 且滿足 (D3)，則它也滿足 (D3)$^+$。

證明. 在典範驗證下，每個訊息 m 恰有唯一的合法標籤 $\mathsf{Tag}(\mathsf{ik},m)$。設敵手輸出 (D3)$^+$ 意義下的偽造 (m,τ)：$\mathsf{Vf}(\mathsf{ik},m,\tau)=1$ 且 $(m,\tau)\notin\mathcal{Q}^+$。分兩種情況：

- 若 $m\notin\mathcal{Q}$：這已是 (D3) 意義下的偽造。
- 若 $m\in\mathcal{Q}$：敵手曾查詢 m 並得到 $\tau_m=\mathsf{Tag}(\mathsf{ik},m)$，故 $(m,\tau_m)\in\mathcal{Q}^+$。由唯一性，$\mathsf{Vf}(\mathsf{ik},m,\tau)=1$ 蘊含 $\tau=\mathsf{Tag}(\mathsf{ik},m)=\tau_m$，於是 $(m,\tau)=(m,\tau_m)\in\mathcal{Q}^+$，矛盾。此情況不可能發生。

所以每個 (D3)$^+$ 偽造都是 (D3) 偽造，兩者的攻破機率相等；(D3) 成立則 (D3)$^+$ 成立。 ■

因為滿足 (D3) 的確定性典範驗證 MAC 存在（HMAC、CMAC、基於 PRF 的 MAC），**(D3)$^+$ 是可達成的**。實務上幾乎所有 MAC 都是確定性的，所以「免費」得到強不可偽造性。

為什麼在意強不可偽造性？

在 Encrypt-then-MAC 中，若 MAC 只滿足 (D3) 而不滿足 $(D3)^+$，敵手可以把合法的 (c,τ) 改成 (c,τ')，收件端仍會接受並解密出同一個明文。這對完整性看似無害，卻破壞了**密文完整性**（INT-CTXT）與 **CCA 安全**：敵手得到了一個「與挑戰密文不同、卻解密成相同明文」的密文，可以合法地拿去問解密預言機。Bellare–Namprempre（2000）正是因此要求 EtM 的 MAC 必須強不可偽造；KL-IMC 習題 5.4 要你構造這個反例。

9.8 (viii) 給敵手驗證預言機，會讓它更強嗎？

(viii)

我們在所有候選定義中都沒有給敵手 Vf 的存取權。假設

- $\mathsf{Vf}(\mathsf{ik},m,\tau)$ 計算 $\tau'=\mathsf{Tag}(\mathsf{ik},m)$，若 $\tau=\tau'$ 回傳 1、否則回傳 0；且
- $\mathsf{Tag}(\mathsf{ik},\cdot)$ 是確定性的。

這稱為**確定性 MAC 的典範驗證**（canonical verification for a deterministic MAC）。在這些假設下，一個能存取 $\mathsf{Tag}(\mathsf{ik},\cdot)$ 的敵手，若再給它 $\mathsf{Vf}(\mathsf{ik},\cdot,\cdot)$ 的存取權，會變得更強大嗎？(提示：用敵手對 $\mathsf{Tag}(\mathsf{ik},\cdot)$ 的存取權「建造」典範驗證。)

解答

結論：不會（在可忽略的差距內）。在典範驗證 + 確定性 Tag 的假設下，Vf 預言機能告訴敵手的每一件事，敵手都能用 Tag 預言機自己算出來。

核心論證（依提示）。設 $\mathcal{A}$ 是一個同時使用 $\mathsf{Tag}(\mathsf{ik},\cdot)$ 與 $\mathsf{Vf}(\mathsf{ik},\cdot,\cdot)$ 的敵手。我們建造一個只使用 $\mathsf{Tag}(\mathsf{ik},\cdot)$ 的敵手 $\mathcal{B}$，它在內部執行 $\mathcal{A}$，並「模擬」Vf 預言機：

- 當 $\mathcal{A}$ 查詢 $\mathsf{Tag}(\mathsf{ik},m)$：$\mathcal{B}$ 轉送給自己的黑盒，把答案回給 $\mathcal{A}$（並記下 (m,τ_m)）。
- 當 $\mathcal{A}$ 查詢 $\mathsf{Vf}(\mathsf{ik},m,\tau)$：$\mathcal{B}$ 用黑盒取得 $\tau'=\mathsf{Tag}(\mathsf{ik},m)$，回答 1 若 $\tau=\tau'$、否則回答 0。

因為 Tag 是確定性的，黑盒回傳的 τ' 就是真正的 $\mathsf{Tag}(\mathsf{ik},m)$；因為驗證是典範的，真正的 Vf 做的也只是「算 $\mathsf{Tag}(\mathsf{ik},m)$、比對」。所以 $\mathcal{B}$ 的回答與真正的 Vf 預言機一模一樣——$\mathcal{A}$ 無法分辨自己是在和真的 Vf 互動，還是和 $\mathcal{B}$ 的模擬互動。於是 $\mathcal{A}$ 能做到的任何事，$\mathcal{B}$ 不用 Vf 也能做到：Vf 預言機沒有提供任何新的資訊。

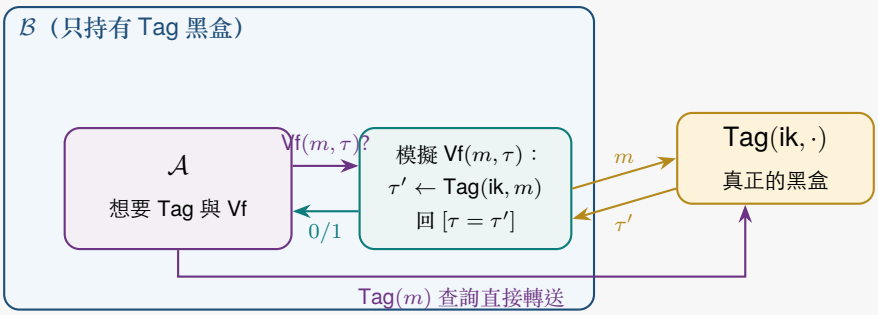


圖 9 (viii) 的模擬：$\mathcal{B}$ 把 $\mathcal{A}$ 包起來，用 Tag 黑盒替 $\mathcal{A}$「演出」一個完美的 Vf 預言機。

一個需要小心的細節 (進階)。上面的論證有一個記帳上的微妙之處：$\mathcal{B}$ 為了模擬 Vf(ik, m, τ) 而向黑盒查詢了 m，於是 m 進入了$\mathcal{B}$ 的查詢集合 $\mathcal{Q}_{\mathcal{B}}$。如果 $\mathcal{A}$ 最後偽造的訊息 m^* 恰好是它先前只拿去驗證、從未拿去貼標籤的訊息（$m^* \notin \mathcal{Q}_{\mathcal{A}}$ 但 $m^* \in \mathcal{Q}_{\mathcal{B}}$），那麼 $\mathcal{A}$ 的偽造在 (D3) 的規則下對 $\mathcal{B}$ 就「不算數」了。

解法：$\mathcal{B}$ 不需要為「新訊息」上的驗證查詢真的去問黑盒。精確的歸約如下。令 $\mathcal{A}$ 最多做 q_v 次驗證查詢。$\mathcal{B}$ 事先均勻選一個指標 $j^* \in \{0, 1, \ldots, q_v\}$，然後執行 $\mathcal{A}$：

1. Tag 查詢照常轉送並記錄 (m, τ_m)。
2. 對 Vf(ik, m, τ) 查詢：
   - 若 m 已被 $\mathcal{A}$ 查詢過 Tag（$\mathcal{B}$ 知道 τ_m）：直接回答 $[\tau = \tau_m]$，不碰黑盒。由典範驗證與確定性，這個回答是正確的。
   - 若 m 是新訊息：這是第 i 個「新訊息驗證查詢」。若 $i = j^*$，$\mathcal{B}$ 立即停止並輸出 (m, τ) **作為自己的偽造**；否則回答 0。
3. 若 $\mathcal{A}$ 結束並輸出 (m^*, τ^*)，$\mathcal{B}$ 也輸出 (m^*, τ^*)。

分析：令 F 為「$\mathcal{A}$ 在真實實驗中某次對新訊息的驗證查詢得到答案 1」的事件，j_0 為第一次發生的指標。

- 若 $\neg F$：$\mathcal{B}$ 對所有新訊息驗證查詢回答 0 都是正確的，模擬完美；當 $j^* = 0$ 時 $\mathcal{B}$ 跑完全程並輸出 $\mathcal{A}$ 的偽造，而 $\mathcal{Q}_{\mathcal{B}} = \mathcal{Q}_{\mathcal{A}}$，所以 $\mathcal{A}$ 贏 $\Rightarrow$ $\mathcal{B}$ 贏。
- 若 F 且 $j_0 = j$：在第 j 次新訊息驗證查詢之前模擬完美；當 $j^* = j$ 時 $\mathcal{B}$ 輸出的 (m, τ) 正是一個通過驗證、且 $m \notin \mathcal{Q}_{\mathcal{B}}$ 的偽造。

因此

$$\Pr[\mathcal{B} \text{ 贏}] \;\geq\; \frac{1}{q_v + 1}\Big(\Pr[\mathcal{A} \text{ 贏} \wedge \neg F] + \Pr[F]\Big) \;\geq\; \frac{\Pr[\mathcal{A} \text{ 贏}]}{q_v + 1}.$$

q_v 是多項式，所以若 $\mathcal{B}$ 的成功機率可忽略，$\mathcal{A}$ 的也可忽略：**有無 Vf 預言機，(D3) 的安全性等價**（只差一個多項式因子）。$\mathcal{B}$ 的黑盒查詢次數甚至不比 $\mathcal{A}$ 多。同樣的論證對 $(\text{D3})^+$ 也成立（KL-IMC p. 113：「任何強安全的 MAC 在允許驗證查詢時仍然強安全」）。

兩個重要的附註。

1. **假設是必要的**。若 Tag 是隨機化的、或 Vf 不是典範的（例如 Vf 會接受一些 Tag 不會輸出的標籤），Vf 預言機就可能洩漏 Tag 預言機給不了的資訊，驗證查詢可以造成差別。Bellare、Goldreich 與 Mityagin（2004，*The Power of Verification Queries in Message Authentication and Authenticated Encryption*）系統性地研究了這個問題。KL-IMC 坦言「含驗證預言機的定義或許才是『正確』的定義」，只是因為實務上的 MAC 幾乎都是典範的，才採用較簡單的版本。
2. **現實中的驗證預言機洩漏的不只一個位元**。收件端就是一個活生生的 Vf 預言機。若 Vf 的實作用一般的位元組比較（如 C 的 `memcmp` / `strncmp`，遇到第一個不同的位元組就提早返回），敵手可以從**拒絕所花的時間**逐位元組猜出正確標籤（KL-IMC §4.2「A potential timing attack」）：對每個位置試 256 種值，花最久才被拒絕的那個就是對的。這類**側通道攻擊**不在我們的定義之內（定義假設敵手只看到預言機的輸出），卻真實存在——所以 MAC 比對一定要用**常數時間**（constant-time）的實作。

Problem 1 的重點回顧

- MAC 正確性：$\forall \mathsf{ik}, m:\ \mathsf{Vf}(\mathsf{ik}, m, \mathsf{Tag}(\mathsf{ik}, m)) = 1$；它不談不合法標籤。
- 常數標籤 MAC 是「正確但不安全」與「滿足 (D1) 但不滿足 (D3)」的萬用反例。
- (D2) $\succ$ (D3) $\succ$ (D1)；(D2) 因正確性而不可達成；選 (D3)（EUF-CMA）。
- 強不可偽造性 $(\mathsf{D3})^+$：禁止任何新的 (m, τ) 對；確定性典範驗證 MAC 自動滿足；EtM 的 CCA 安全需要它。
- 典範驗證 + 確定性 $\Rightarrow$ Vf 預言機可被 Tag 預言機模擬（損失因子 $q_v + 1$）；但時間側通道是另一回事。

10 Problem 2：On encryption（論加密）

題目背景

回憶 $\mathrm{SKE} = (\mathsf{KGen}, \mathsf{Enc}, \mathsf{Dec})$：KGen 輸出 $\boxed{\mathsf{sk}}$；$\mathsf{Enc}(\mathsf{sk}, m)$ 輸出密文 c；$\mathsf{Dec}(\mathsf{sk}, c)$ 輸出訊息 m。以下假設 SKE 正確，且某把 sk 已由 KGen 產生。

10.1 (i) 確定性加密為何不可取？

(i)

假設敵手能存取 $\mathsf{Enc}(\mathsf{sk}, \cdot)$。描述一個場景，說明 Enc 是**確定性**的（同一訊息永遠輸出同一密文）為何不可取。（提示：想想常會出現相似、公式化（stereotyped）訊息的場景。）

解答

核心問題。確定性加密把「加密」變成一個固定的查表：$m \mapsto c$ 是一個固定的函數。於是

1. **相等性洩漏**：$c_i = c_j \iff m_i = m_j$。敵手不需金鑰就知道哪些密文對應同一則訊息、每則訊息出現幾次（講義迭代 (iii) 的「字母頻率」反例，在訊息層級重演）。
2. **字典攻擊**：若敵手有 $\mathsf{Enc}(\mathsf{sk}, \cdot)$ 的存取權，它可以把所有「可能的」訊息都拿去加密、建一本密文–明文對照字典，之後看到任何密文就直接查表「解密」——完全不需要知道金鑰。當訊息空間小、或訊息高度公式化時，這本字典很容易建完。

場景一：公式化的軍事／商業訊息。戰場上的回報往往只有少數幾種：「`NOTHING TO REPORT`」、「`ATTACK AT DAWN`」、「`RETREAT`」。敵手（有加密預言機時）把這幾句都加密一遍；之後攔到的每一封密文，一比對就知道內容。即使沒有預言機，光是「今天的密文與過去 30 天某一天相同」也足以推論「今天沒事」。二戰時盟軍破解 Enigma 的「crib」正是利用德軍每日固定的「*Keine besonderen Ereignisse*」（無特殊事件）與氣象報告格式；中途島的「AF 缺水」則是主動誘導對方加密指定明文（見第 4 節）。

場景二：投票或是／否回覆。電子投票系統把每張選票 $m \in \{\mathtt{YES}, \mathtt{NO}\}$ 加密後公布。敵手自己投一票 `YES`（這就是一次加密預言機查詢 !），看到自己的密文 $c_{\mathtt{YES}}$；從此所有與 $c_{\mathtt{YES}}$ 相同的選票都是贊成票，其餘都是反對票——全部選票被解密，金鑰仍然安全無虞。

場景三：金融交易。「`TRANSFER £100 FROM A TO B`」這類訊息格式固定、欄位值範圍小。敵手開一個帳戶、對自己做幾筆轉帳（取得密文），就能辨識出其他人的交易類型與金額級距；再配合重放（Problem 2 (ii)），後果更嚴重。

場景四：資料庫欄位加密。為了能用 `WHERE name = ?` 查詢，有些系統對欄位採確定性加密。2015 年 Naveed、Kamara 與 Wright 對這類系統（如 CryptDB 式設計）的研究顯示：只靠公開的統計資料（例如人口的姓氏分布、病歷中診斷碼的頻率）配合密文的頻率，就能還原大部分欄位內容——這正是「頻率洩漏」在現代的具體代價。

形式化的說法。任何**確定性且無狀態**的加密方案都不可能是 CPA 安全的（第 4.3 節的遊戲）：敵手先向預言機查詢 m_0 得到 c_0，再以 (m_0, m_1) 挑戰；若挑戰密文 $c^* = c_0$ 就猜 $b = 0$、否則猜 $b = 1$，必勝。這也是為什麼講義在迭代 (v) 中要給敵手「讓選定訊息被加密」的能力——它立刻排除掉所有確定性方案。

解法。加密必須是**隨機化**（每次加密抽一個新的隨機值 / IV）或**有狀態**（使用計數器作為 nonce）的，使同一訊息每次都產生不同的密文；KL-IMC §3.4 稱此為「加密必須是機率性的」。附註：確定性加密在少數場合仍被刻意使用（金鑰包裝、可搜尋加密、SIV 模式的「nonce 誤用抵抗」），但使用者必須清楚它一定洩漏相等性，並確認在該場景中可以接受。

10.2 (ii) 在訊息中加入什麼，可以防止重放、重排與反射？

題目設定與 (ii)

另外考慮一個「完美」的 MAC（非正式：在任何情況下都沒有敵手能偽造）。我們用下列方式對抗第一堂課介紹的重放、重排與反射攻擊。

給定 $(\boxed{\mathsf{sk}}, \boxed{\mathsf{ik}})$，傳送 m：令 $c = \mathsf{Enc}(\mathsf{sk}, m)$、$\tau = \mathsf{Tag}(\mathsf{ik}, c)$，送出 $\tilde{c} = (c, \tau)$。

接收 $\tilde{c} = (c, \tau)$：檢查 $\mathsf{Vf}(\mathsf{ik}, c, \tau) = 1$，否則中止；若未中止，令 $m = \mathsf{Dec}(\mathsf{sk}, c)$。

(ii) How might you add information to the message m to prevent replay, reordering and reflection attacks?（提示：考慮能明確定義「新鮮度」、「順序」與「方向」的資訊。）

解答

先弄清楚敵手還剩下什麼能力。MAC 完美，表示敵手無法送出任何「不是由 Alice 或 Bob 誠實產生」的 $\tilde{c}$——它改不了任何一個位元。它剩下的只有對整個 $\tilde{c}$ 的操作：重複送（重放）、調換順序（重排）、送回原發送者（反射）、扣住（延遲）、丟掉（拒絕）。前三者的共通點是：每一個 $\tilde{c}$ 本身都是真的，錯的是它出現的「時間」、「位置」或「方向」。所以對策也很清楚：**把「這則訊息應該在什麼時候、什麼位置、往哪個方向出現」寫進訊息裡**，讓接收端能檢查。因為這些資訊是 m 的一部分，它們會被 Enc 加密、再被 Tag（透過 c）保護——敵手無法竄改它們。

攻擊	要定義的概念	加入的資訊	接收端的檢查
重放	新鮮度（freshness）	每則訊息唯一的識別：**序號／計數器 ctr**（最常用）；或**時間戳** T；或由接收端事先提供的**隨機挑戰值**（nonce）	計數器：只接受 ctr 等於期望值（或在滑動視窗內且未見過）。時間戳：$\lvert T - T_{\text{now}}\rvert \le \Delta$ 且在視窗內未重複。nonce：等於自己剛送出的挑戰值。
重排	順序（order）	單調遞增的**序號 ctr**（與上一列共用同一個欄位）	嚴格要求 ctr = 期望值（在可靠通道上）；或接受視窗內的任意順序但由上層重組（不可靠通道）。
反射	方向（direction）	**方向位元** $b_{A\to B}$（例如 $b_{A\to B} = 0$ 若 A 的識別碼字典序小於 B，$b_{B\to A} = 1$）；或直接寫入**發送者／接收者身分** $\mathrm{id}_A \,\|\, \mathrm{id}_B$	只接受方向位元等於「對方 → 我」的訊息；收到標示為「我 → 對方」的訊息一律拒絕。

把三者合在一起，就是 KL-IMC §5.4 的「安全通訊 session」構造：

加入方向位元與計數器的 session 協定

狀態。雙方各自維護兩個計數器 $\mathsf{ctr}_{A\to B}, \mathsf{ctr}_{B\to A}$（初始為 0），並約定方向位元 $b_{A\to B}=0$、$b_{B\to A}=1$。

A 傳送 m 給 B。

1. $m' = b_{A\to B} \,\|\, \mathsf{ctr}_{A\to B} \,\|\, m$；（方向 ‖ 序號 ‖ 內容）
2. $c = \mathsf{Enc}(\mathsf{sk}, m')$，$\tau = \mathsf{Tag}(\mathsf{ik}, c)$，送出 (c, τ)；
3. $\mathsf{ctr}_{A\to B} \leftarrow \mathsf{ctr}_{A\to B} + 1$。

B 接收 (c, τ)。

1. 若 $\mathsf{Vf}(\mathsf{ik}, c, \tau) \neq 1$：拒絕。（竄改 / 偽造）
2. $m' = \mathsf{Dec}(\mathsf{sk}, c)$，解析為 $b \,\|\, \mathsf{ctr} \,\|\, m$（若格式不符：拒絕）。
3. 若 $b \neq b_{A\to B}$：拒絕。（反射：這是我自己送出的方向）
4. 若 $\mathsf{ctr} \neq \mathsf{ctr}_{A\to B}$：拒絕。（重放：ctr 太舊；重排 / 遺失：ctr 跳號）
5. 輸出 m，並令 $\mathsf{ctr}_{A\to B} \leftarrow \mathsf{ctr}_{A\to B} + 1$。

B 傳送給 A 時對稱地使用 $b_{B\to A}$ 與 $\mathsf{ctr}_{B\to A}$。

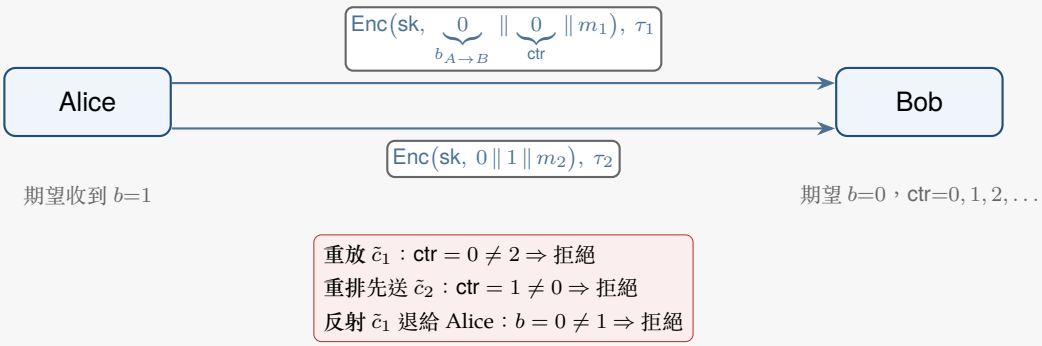


圖 10 方向位元擋反射、計數器擋重放與重排。三個欄位都在 m' 內，受到加密與 MAC 的雙重保護。

逐一驗證三種攻擊都失效。

- **重放**：敵手重送 $\tilde{c}_1$。B 驗證通過、解密得 $\mathsf{ctr}=0$，但 B 已把 $\mathsf{ctr}_{A\to B}$ 推進到 1（或更大），$0 \neq$ 期望值 $\Rightarrow$ 拒絕。
- **重排**：敵手先送 $\tilde{c}_2$（$\mathsf{ctr}=1$）。B 期望 $0 \Rightarrow$ 拒絕。（若之後 $\tilde{c}_1$ 送達則被接受，再來 $\tilde{c}_2$ 也會被接受——順序被強制了。）
- **反射**：敵手把 $\tilde{c}_1$ 送回 A。因為雙方共用 $(\mathsf{sk}, \mathsf{ik})$，$A$ 的 Vf 會通過、Dec 會成功——但解析出 $b = b_{A\to B} = 0$，而 A 只接受 $b_{B\to A} = 1 \Rightarrow$ 拒絕。

幾個設計上的補充。

1. **為什麼放進 m 而不是明文附在旁邊？**只要這些欄位受到 MAC 保護即可；放進 m 是最省事的做法（τ 透過 c 覆蓋了一切），而且順便隱藏了序號這類中繼資料。現代的 AEAD 介面另有「關聯資料」（associated data）欄位，可以把序號以明文但受鑑別的方式附上。**絕對不能做的是把序號放在 MAC 範圍之外**——那敵手改一改就能繞過檢查。
2. **隱含序號**：TLS 1.3 甚至不傳送序號，雙方各自計數，並把序號混入加密的 nonce；序號不對時解密 / 驗證會直接失敗。省頻寬，效果相同。
3. **另一種擋反射的方法**：兩個方向使用**不同的金鑰**（$\mathsf{sk}_{A\to B}, \mathsf{ik}_{A\to B}$ 與 $\mathsf{sk}_{B\to A}, \mathsf{ik}_{B\to A}$，TLS 1.3 的 `client_write_key` / `server_write_key` 即是）。反射回來的訊息會直接驗證失敗，不需要方向位元。KL-IMC 註腳指出這是實務上更常見的做法。
4. **跨 session 的重放**：計數器每次 session 都從 0 開始，若兩次 session 用同一組金鑰，上一次的 $\tilde{c}$ 就能在這一次重放。解法：每個 session 使用新鮮的金鑰（金鑰協商），或把一個隨機的 **session 識別碼**也放進 m'。
5. **時間戳 vs. 計數器**：時間戳不需雙方同步「狀態」，但需要同步「時鐘」，且在容許視窗 Δ 內仍可重放（除非接收端額外記住視窗內看過的訊息）。計數器精確，但雙方必須保持同步——這正是 (iii) 要討論的脆弱點。

10.3 (iii) 若敵手能選擇性地拒絕傳遞，對 (ii) 的答案有何影響？

(iii)

If the adversary is able to selectively deny delivery of some $\tilde{c}$, how does this affect your answer to (ii.)?

解答

一句話：敵手仍然無法讓 B 接受任何不該接受的訊息（保密性、完整性、防重放／重排／反射全都保持），但它獲得了對可用性（availability）的強大槓桿，而且某些「遺失」會變得與「惡意刪除」無法區分。具體影響取決於 (ii) 中接收端對計數器採取的政策。

1. 嚴格計數器政策（只接受 ctr = 期望值）。

- **放大的阻斷服務**：敵手只要丟掉一則 $\tilde{c}_i$，之後的 $\tilde{c}_{i+1}, \tilde{c}_{i+2}, \ldots$ 全部因為「ctr 跳號」而被拒絕——雙方計數器**失去同步**（desynchronisation），整個 session 停擺，直到重新協商。丟一封訊息的成本，換來癱瘓整條連線的效果。
- **好消息：遺失會被偵測**。KL-IMC 把「message-dropping attack」列為第三種攻擊，並說「我們至少希望這種行為能被偵測」。嚴格計數器恰好做到這點：B 看到 ctr 跳號，就知道中間少了東西。密碼學不能阻止丟包，但能讓丟包不會悄悄地發生。

- **安全性不受影響**：敵手丟掉 $\tilde{c}_i$ 後，仍然無法讓 B 接受 $\tilde{c}_{i+1}$ 當作第 i 則、也無法補送任何偽造的訊息。

2. 寬鬆政策（滑動視窗／「只要比上次大就接受」）。為了容忍自然的封包遺失（UDP、無線網路），DTLS 與 IPsec 採用滑動視窗：接受視窗內未見過的序號。此時：

- **選擇性刪除變得「合法」**：敵手丟掉 $\tilde{c}_1$、放行 $\tilde{c}_2$，B 接受 $\tilde{c}_2$ 並把視窗推進；$\tilde{c}_1$ 之後即使送達也可能落在視窗之外而被丟棄。對 B 而言，這與「網路剛好掉了一個封包」無法區分——敵手得到了一個「精準刪掉特定訊息」的能力（例如刪掉「取消交易」那一封）。
- **視窗內的重排被容忍**：這是設計上的取捨，上層應用必須自行處理順序。
- **重放仍被擋住**：視窗內的序號必須「未見過」，所以同一則訊息仍只會被接受一次。

3. 若 (ii) 使用時間戳。拒絕傳遞的近親是**延遲**：敵手扣住 $\tilde{c}$，在容許視窗 Δ 的最後一刻才放行——訊息仍「新鮮」、會被接受，但時效可能已過（「現在賣出」在崩盤後才到）。視窗越大越容易出事，越小越容易誤拒合法訊息；這是時間戳方案的固有張力。

4. 截斷攻擊（truncation）：連嚴格計數器也偵測不到的事。計數器能偵測中間的缺口，卻偵測不到結尾的缺失：若敵手丟掉 session 的最後 k 則訊息，B 只會以為「對方還沒說完」或「對話結束了」。SSL 2.0 就有這個問題；後來的 TLS 加入了受鑑別的 `close_notify` 訊息——沒有收到它就表示連線被截斷。一般原則：**需要一則受鑑別的「結束」訊息（或回條／心跳）**，拒絕傳遞才會變成可偵測的事件。

總結：這就是講義為何把「延遲」與「拒絕傳遞」劃入網路工程。密碼學控制的是訊息的內容與身分，不是通道的行為。在 (ii) 的設計下，敵手能做的「壞事」被壓縮到只剩**不送**與**晚送**——而這兩件事任何密碼學都無法阻止（敵手只要剪斷電纜即可）。我們能做的是：(a) 讓它**可被偵測**（計數器缺口、結束訊息、逾時）；(b) 讓系統**優雅地恢復**（重傳、重新協商、改走別的路徑）；(c) 用**冗餘與網路層防護**提高敵手的成本。這些都是協定與系統設計的工作，而不是加密或 MAC 的工作。

🎓 Problem 2 的重點回顧

- 確定性加密 ⇒ 相等性洩漏 + 字典攻擊；任何確定性無狀態方案都不是 CPA 安全。場景：公式化訊息、是 / 否投票、交易、資料庫欄位。
- 新鮮度 → 計數器 / 時間戳 / nonce；順序 → 計數器；方向 → 方向位元或每方向獨立金鑰。全部放進受 MAC 保護的範圍內。
- 拒絕傳遞：嚴格計數器 ⇒ 失同步的 DoS 但可偵測；滑動視窗 ⇒ 選擇性刪除與自然遺失無法區分；截斷需要受鑑別的結束訊息。密碼學只能偵測，不能阻止。

11 Problem 3：On cryptography as munitions（論密碼學作為軍火）

(i)、(ii)

(i) 閱讀第一堂課中場休息所引用的兩篇文章之一：

- Ronald J. Stay, *Cryptic Controversy: U.S. Government Restrictions on Cryptography Exports and the Plight of Philip Zimmermann*, 13 Ga. St. U. L. Rev. 581（1997；線上版標示 2012）。`https://readingroom.law.gsu.edu/gsulr/vol13/iss2/14`
- *Constitutional Law. Free Speech Clause. Sixth Circuit Classifies Computer Source Code as Protected Speech. Junger v. Daley, 209 F.3d 481 (6th Cir. 2000)*, 114 Harv. L. Rev. 1813（2001）。`https://doi.org/10.2307/1342657`

(ii) 用紙筆寫下你對所讀文章的想法，大約一面 A4。像密碼學與原始碼這樣「純技術」的物件，這麼快就變成社會與法律上的爭議，你感到意外嗎？

這一題沒有「標準答案」，評量的是閱讀與反思。以下提供：(1) 兩篇文章的**閱讀指南**（結構、論點、關鍵詞），讓你能在有限時間內讀懂法律評論；(2) **寫作架構**與思考提示；(3) 一篇**範例反思**供參考。**請務必自己動筆寫**——這題的價值在於你自己的思考，而且 tutor 分得出來。

11.1 (i) 閱讀指南

文章 A：Stay（1997），〈Cryptic Controversy〉

這是一篇法學院的學生評論（student note），約 24 頁，結構非常清楚，適合第一次讀法律文獻的人。

段落	內容與重點
開場	1994 年 11 月 9 日，Zimmermann 在杜勒斯機場被海關攔查、搜行李、長時間訊問——「危險武器」是一套叫 PGP 的軟體。1993 年起聖荷西大陪審團調查 PGP 的全球散布是否違反出口法。
I. 密碼學簡史	從斯巴達、Zimmermann 電報到二戰；個人電腦興起後出現「公眾密碼學」；無鑰 / 單鑰（DES）/ 公鑰（RSA、PGP）系統的簡介。（技術描述略為粗糙，讀者可用本課程所學修正。）
II. 法規框架	**AECA** 授權總統管制「國防物品」→ 國務院的 **ITAR** → **美國軍火清單第 XIII** 類列入密碼裝置與軟體。「技術資料」與「軟體」的定義含糊；「出口」定義極廣；印刷品與「公開領域」有豁免；罰則最高十年或一百萬美元。
III. 第一修正案	Zimmermann 免費散布 PGP 屬政治 / 科學性而非商業性言論，應受高度保護；ITAR 的許可制可能構成**事前限制**（prior restraint）與**過度廣泛**（overbreadth）；引用 *Bernstein v. Dept. of State*（1996 地院）認定原始碼是言論。
IV. 第五修正案	法律**明確性**（void-for-vagueness）：ITAR 對「軟體」、「技術資料」、「出口」的定義跟不上科技。著名例子：一位 Boeing 主管帶筆電到巴黎航空展、用 PGP 寄一封加密 email 回西雅圖，技術上就是未經許可出口軍火。
V. 政治問題原則	*United States v. Martinez*（11th Cir. 1990，Videocipher II 案）認為「哪些東西該列入軍火清單」是不可由法院審查的**政治問題**。作者引用 Harold Koh 的批評，並指出涉及《權利法案》個人自由時，法院不應迴避；*Bernstein* 法院也認為可審查。
結論	國會（Cantwell 法案）、商務部長、EFF 訴訟都在推動鬆綁；1996 年 1 月檢方不起訴且未說明理由。引用 Eben Moglen：「用密碼說話」的權利就像「說納瓦荷語的權利」。ITAR 可能既缺乏「迫切的政府利益」（因為好的密碼軟體早已遍布全球），又因定義過時而違憲地模糊。

文章 B：Harvard Law Review（2001）對 *Junger v. Daley* 的案例評析

六頁的案例評析（case comment，作者 Jeanne C. Fromer）。讀法：先弄清楚**案情**與**判決**，再看評析**批評什麼**。

- **案情**：法學教授 Peter Junger 想在課程網站公開自己寫的加密程式原始碼，依商務部 EAR 需申請出口許可。他主張原始碼是受第一修正案保護的言論，提起「表面違憲」（facial challenge）訴訟。
- **地院（1998）**：加密原始碼「偶爾有表達性」，但其出口不是受保護的行為；規範是內容中立的。
- **第六巡迴（2000 年 4 月）**：**逆轉**。原始碼兼具表達性與功能性；對懂程式語言的人而言，原始碼是傳達密碼學想法「最有效、最精確」的方式，故受第一修正案保護。功能性應在中度審查（intermediate scrutiny，*O'Brien* 標準）中衡量政府利益時考慮，而非用來否定言論地位。政府必須證明危害「真實而非臆測」。發回重審（同時規則已於 2000 年 1 月放寬）。

- **評析的批評**：法院「所有原始碼都是言論」的結論過於寬泛，偏離最高法院處理**象徵性行為**（symbolic conduct）的 *Spence v. Washington* 判準——必須有「傳達特定訊息的意圖」且「觀眾很可能理解」。純粹為了執行而寫、無意與人溝通的程式碼（例如只會被編譯運行的程式），未必通過這個判準。更忠於先例的做法是**個案判斷**原始碼是否為言論。

讀法律文獻的小技巧

- 法律評論的**註腳**通常佔一半篇幅，第一次讀可以全部跳過，只讀正文。
- 先讀**開頭與結論**，掌握作者的立場；再回頭看論證結構（通常以羅馬數字分節）。
- 常見術語：*First Amendment*（言論自由）、*prior restraint*（事前限制）、*overbreadth*（過度廣泛）、*vagueness*（模糊）、*intermediate scrutiny*（中度審查）、*content-neutral*（內容中立）、*political question doctrine*（政治問題原則）、*facial challenge*（表面違憲挑戰）、*remand*（發回）。
- 美國案件引註格式：*Junger v. Daley*, 209 F.3d 481 (6th Cir. 2000) = 聯邦判例彙編第三輯第 209 卷第 481 頁，第六巡迴上訴法院 2000 年判決。

11.2 (ii) 寫作指南

題目要什麼？一面 A4 的手寫反思（英文約 350–450 字），重點是**你的想法**，不是摘要。題目最後一句其實給了主題：「純技術」的物件為何、以及多快會變成社會與法律爭議？你意外嗎？

建議架構（四段）

1. **文章在說什麼**（3–4 句）：用自己的話說出主要論點與你印象最深的一個細節（例如機場攔查、Boeing 主管的例子、「磁片是武器、書不是」）。
2. **「意外」的時刻**（主體一）：哪一點讓你意外或不意外？為什麼？試著說出機制——為什麼法律範疇一碰到程式碼就失靈？（功能性 vs. 表達性；數學 vs. 商品 vs. 武器；「出口」在網路時代的意義。）
3. **連結課程概念**（主體二）：Blanchette 的定義（規範行為的程式碼必然被規範）；Rosulek 的「控制資訊存取」就是權力；Kerckhoffs 原則要求演算法公開、而出口管制正是在阻止公開；「to define is to limit」——法律也在下定義。
4. **今日的回聲與你的立場**（結尾）：Netscape 40 位元 → FREAK / Logjam 的教訓；現今關於後門、端對端加密、客戶端掃描的辯論。清楚說出你的看法，並承認對方的最佳論點。

動筆前的思考提示

- 如果原始碼是言論，那麼編譯後的執行檔呢？一個只會被機器執行、沒有人會「讀」的程式，是在「說」什麼？
- 政府的利益（情報、執法）是真實的。如果你是 1993 年的檢察官，你會怎麼論證？你的

論證在 2026 年還成立嗎？

- 「印成書就能合法出口」這條界線荒謬嗎？還是它反映了某種合理的直覺（紙本傳播慢、需要人工重建）？
- 刻意削弱的密碼學（40 位元、512 位元）在政策取消後存活了二十年。這對「只給好人用的後門」提案說了什麼？
- 你是否同意「以技術手段規範行為」本身就是一種政治行為？密碼學家該為此負什麼責任？

範例反思（英文，約 420 字；僅供參考架構，請勿照抄）

When code becomes a weapon: a reflection on Stay (1997)

Stay's note opens not with mathematics but with a luggage search. In November 1994 Philip Zimmermann, returning from a speaking tour in Europe, was pulled aside at Dulles airport and questioned about whether he had exported "munitions". The munition was PGP, a free program that let ordinary people encrypt e-mail. The rest of the note explains how this was possible: the Arms Export Control Act lets the State Department list "defense articles", and Category XIII of the Munitions List placed cryptographic software next to missiles and tanks. "Export" was read so broadly that uploading a file, or letting a foreign national read it, could count—with penalties of up to ten years in prison.

What surprised me was not that a government wanted to control cryptography. The lecture had already told us that cryptography decides who may read, write and prove, which is to say that it is about power. What surprised me was how quickly the legal categories broke down when applied to code. The same RSA routine was a defense article on a floppy disk and protected speech in a book; Zimmermann's supporters exploited exactly this gap by having MIT Press print the source code. Stay's vagueness argument—a Boeing engineer e-mailing his office from the Paris air show is technically an arms exporter—shows that the law simply had no stable concept of what a program *is*.

From the course's point of view this is Blanchette's definition in action: cryptography is code written to regulate behaviour, and code that regulates behaviour will itself be regulated. It also gives Kerckhoffs' principle a political edge. If security may rest only on the key, the algorithm must be public, and "public" is precisely what export control tried to prevent.

I am less convinced by the note's confidence that the First Amendment settles the question. *Junger v. Daley* eventually held that source code is speech, but the Harvard Law Review comment is right that not all code is written to communicate; a binary that only ever runs is hard to call expression. The honest position is that cryptographic software is dual-use in a way the legal categories were never designed for.

The most lasting lesson is empirical rather than constitutional. Export rules forced Netscape to ship a 40-bit "international" browser; twenty years later the FREAK and Logjam attacks exploited servers that still spoke those crippled cipher suites. Deliberately weakened cryptography does not expire when the policy does. That, more than any legal argument, is why today's proposals for

"exceptional access" should be read with Stay's history in mind.

範例反思的寫作要點解析

- **第一段**：用一個具體場景（機場）開頭，再濃縮法規框架（AECA → 軍火清單第 XIII 類 → 「出口」的廣義解釋 → 罰則）。展示你真的讀了文章，但不是逐段摘要。
- **第二段**：直接回答題目的「你意外嗎 ?」——而且給出有層次的答案（不意外 X，但意外 Y），並說明機制（法律範疇 vs. 程式碼的多重身分）。引用文章裡最生動的兩個例子（印成書、Boeing 主管）。
- **第三段**：把閱讀連回課程：Blanchette 的定義、Kerckhoffs 原則。這是「顯示你在這門課的脈絡下思考」的關鍵段落。
- **第四段**：展現批判性——不全盤接受作者立場，引入另一篇文章（HLR 評析）作為反方，提出自己的綜合判斷（dual-use）。
- **第五段**：拉到今天（FREAK / Logjam、後門辯論），以一個清楚的主張收尾。
- 全文約 420 字，手寫約一面 A4；沒有任何公式，但每一段都有具體的人名、案名、數字——這是「有讀」與「沒讀」的差別。

若你選讀的是文章 **B**（HLR 評析），同樣的架構可以改成：案情與判決 → 你對「原始碼是言論」直覺上同意與否 → 評析的 *Spence* 判準批評是否有說服力（以你寫過的程式為例：你寫程式時「想對誰說什麼」?）→ 連結課程（Kerckhoffs：公開演算法等於出版）→ 今日的對應（DMCA 反規避條款、Apple 與 FBI、歐盟 Chat Control）。

Problem 3 的重點

筆試若出現此主題，通常是申論題的一部分（例如「說明密碼學的社會或法律面向，並舉例」）。準備三個可以隨時寫出來的例子：**Zimmermann ／ PGP 調查（1993–96）**、**Netscape 國際版 40 位元與 FREAK ／ Logjam**、***Junger v. Daley*（原始碼是言論）**，以及一個你自己的論點。

第三部分

總複習

本週重點整理• 中英術語表• 自我測驗• 參考資料

12 本週重點整理

12.1 一頁看懂第一週

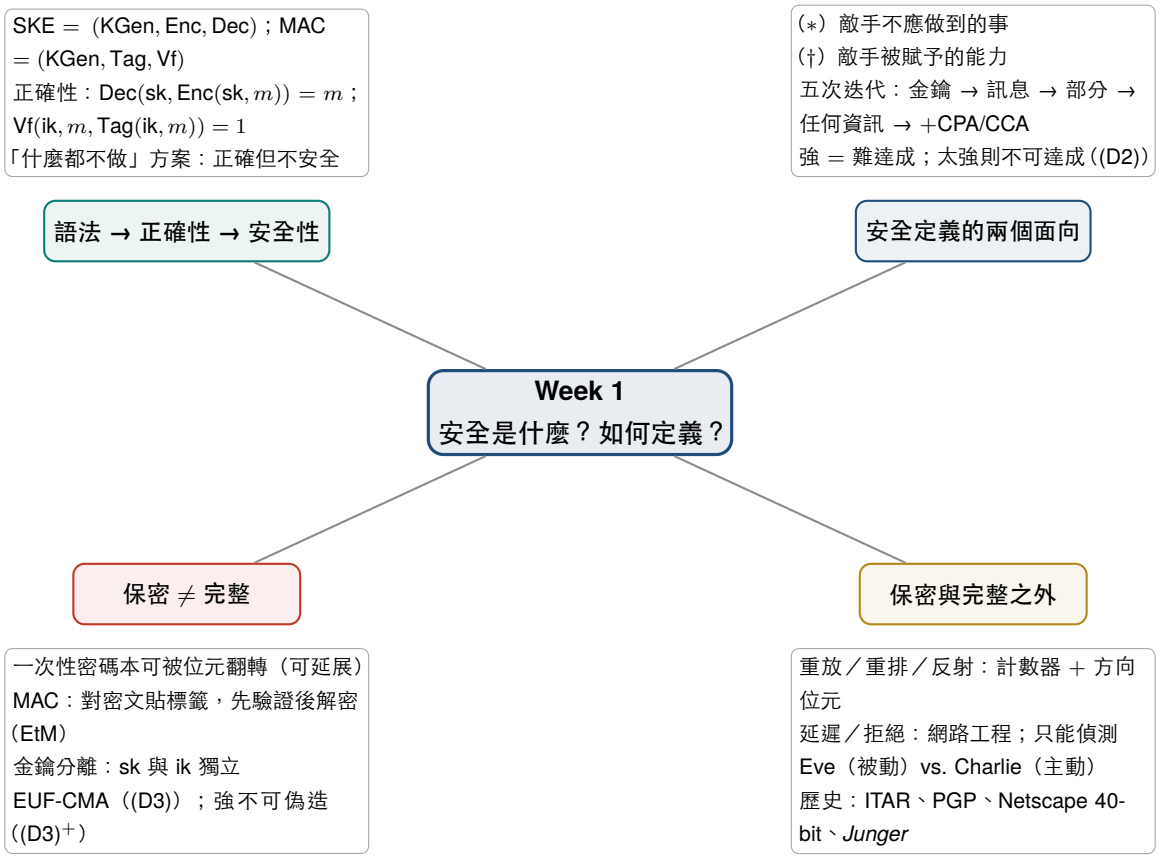


圖 11 第一週概念地圖。

12.2 必背定義

概念	一句話定義
SKE 語法	$\mathsf{sk} \leftarrow \mathsf{KGen}$；$c = \mathsf{Enc}(\mathsf{sk}, m)$；$m = \mathsf{Dec}(\mathsf{sk}, c)$；空間 $\mathcal{K}, \mathcal{M}, \mathcal{C}$。
SKE 正確性	$\forall \mathsf{sk} \leftarrow \mathsf{KGen}, \forall m \in \mathcal{M}:\ \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{sk}, m)) = m$。
MAC 語法	$\mathsf{ik} \leftarrow \mathsf{KGen}$；$\tau = \mathsf{Tag}(\mathsf{ik}, m)$；$\mathsf{Vf}(\mathsf{ik}, m, \tau) \in \{0, 1\}$。
MAC 正確性	$\forall \mathsf{ik} \leftarrow \mathsf{KGen}, \forall m \in \mathcal{M}:\ \mathsf{Vf}(\mathsf{ik}, m, \mathsf{Tag}(\mathsf{ik}, m)) = 1$。
安全定義的兩面向	(∗) 不應可能的事；(†) 敵手的能力。
EUF-CMA ((D3))	有 $\mathsf{Tag}(\mathsf{ik}, \cdot)$ 預言機的敵手，無法對任何 $m \notin \mathcal{Q}$ 輸出合法 (m, τ)。
強不可偽造 ((D3)+)	同上，但禁止任何 $(m, \tau) \notin \mathcal{Q}^+$（含舊訊息的新標籤）。
典範驗證	$\mathsf{Vf}(\mathsf{ik}, m, \tau) = 1 \iff \tau = \mathsf{Tag}(\mathsf{ik}, m)$（確定性 Tag）。
Encrypt-then-MAC	$c = \mathsf{Enc}(\mathsf{sk}, m)$，$\tau = \mathsf{Tag}(\mathsf{ik}, c)$；收到先 Vf、失敗即 $\perp$、通過才 Dec。
定義 X 至少與 Y 一樣強	每個滿足 X 的方案都滿足 Y。嚴格更強：另存在滿足 Y 不滿足 X 的方案。
可延展性	敵手不知明文仍可對明文做可預測修改（如 XOR 加密的位元翻轉）。
Kerckhoffs 原則	演算法公開，安全只依賴金鑰保密。

12.3 常見誤區

⚠ 考試時最容易犯的錯

1. **把正確性當成安全性。**「解得回來」與「敵手學不到東西」是兩回事。
2. **以為加密提供完整性。**一次性密碼本是完美保密的，但一個位元都擋不住。
3. **以為「金鑰安全」就夠了。**迭代 (i) 與 (D1)：什麼都不做的方案金鑰也「安全」。
4. **忘記 (D3) 的 $m \notin \mathcal{Q}$。**少了它就是 (D2)，而 (D2) 因正確性而不可能達成。
5. **MAC 算在明文上（E&M）或先解密再驗證（MtE）。**標準答案是 EtM：對密文貼標籤、先驗證。
6. **以為 MAC 擋得住重放。**Vf 無狀態；重放要靠計數器 / 時間戳等狀態。
7. **把序號放在 MAC 保護範圍外。**那敵手改一下序號就繞過了。
8. **以為密碼學能阻止丟包。**只能偵測與恢復，不能阻止。

13 中英術語對照表

English	中文	說明
cryptography	密碼學	本週給了七種定義；本課程採「研究如何（以及什麼叫）讓計算安全」。
secret key encryption (SKE)	秘密金鑰加密	又稱對稱式 / 私鑰加密；(KGen, Enc, Dec)。
key generation (KGen)	金鑰產生	輸出金鑰；通常是均勻隨機抽樣。
message / plaintext	訊息 / 明文	$m \in \mathcal{M}$。
ciphertext	密文	$c \in \mathcal{C}$。
key space / message space / ciphertext space	金鑰 / 訊息 / 密文空間	$\mathcal{K}$、$\mathcal{M}$、$\mathcal{C}$。
syntax	語法	演算法的介面（輸入 / 輸出型別），不含正確性與安全性。
correctness	正確性	誠實使用時功能正常：解得回來 / 標籤被接受。
security	安全性	敵手在某能力下做不到某事。
security definition	安全定義	由（$*$）與（†）組成。
adversary	敵手	$\mathcal{A}$；Eve（被動）、Charlie / Mallory（主動）。
eavesdropper	竊聽者	只能觀察通道的敵手。
confidentiality / secrecy / privacy	機密性 / 保密性	敵手學不到訊息內容；三詞在本課程同義。
integrity	完整性	收到的就是送出的；需另外的機制（MAC）。
authenticity	真實性 / 鑑別	訊息確實來自持有金鑰的一方；與完整性常並稱。

（續下頁）

English	中文	說明
message authentication code (MAC)	訊息鑑別碼	(KGen, Tag, Vf)。
tag	標籤	$\tau \in \mathcal{T}$；MAC 的輸出。
verify (Vf)	驗證	輸出 1（合法）或 0（不合法）。
valid tag	合法標籤	$\mathsf{Vf}(\mathsf{ik}, m, \tau) = 1$。
forgery	偽造	不知金鑰卻產生合法的 (m, τ)。
oracle	預言機 / 黑盒	$\mathsf{Tag}(\mathsf{ik}, \cdot)$：可查詢、看不到內部。
chosen-message attack (CMA)	選擇訊息攻擊	敵手可選訊息取得標籤；adaptive = 可依結果調整。
known-message attack	已知訊息攻擊	敵手只能被動觀察訊息–標籤對（(D1) 的自然解讀）。
existential unforgeability (EUF-CMA)	存在性不可偽造	對任何新訊息都無法偽造；(D3)。
strong unforgeability (sEUF-CMA)	強不可偽造	無法產生任何新的 (m, τ) 對；$(\mathsf{D3})^{+}$。
canonical verification	典範驗證	重算標籤再比對。
deterministic / randomised	確定性 / 隨機化	同輸入是否永遠同輸出。
chosen plaintext attack (CPA)	選擇明文攻擊	敵手有加密預言機。
chosen ciphertext attack (CCA)	選擇密文攻擊	敵手另有解密預言機。
indistinguishability (IND)	不可區分性	猜不出密文對應 m_0 或 m_1。
semantic security	語意安全	學不到任何可計算的新資訊；與 IND 等價。
perfect secrecy	完美保密	密文與明文統計獨立（Shannon）。
one-time pad	一次性密碼本	$c = m \oplus \mathsf{sk}$；完美保密但可延展。
malleability	可延展性	不知明文仍可預測地改動明文。
Encrypt-then-MAC (EtM)	先加密後鑑別	對密文貼標籤；先驗證後解密；通用安全。
MAC-then-Encrypt (MtE) / Encrypt-and-MAC (E&M)	先鑑別後加密 / 加密並鑑別	另兩種組合；非通用安全。
authenticated encryption (AE)	鑑別加密	同時保密與完整；GCM、ChaCha20-Poly1305。
key separation	金鑰分離	一把金鑰只做一件事。
replay / reordering / reflection attack	重放 / 重排 / 反射攻擊	計數器、計數器、方向位元（或每方向獨立金鑰）。
delay / denial (of delivery)	延遲 / 拒絕傳遞	網路工程範疇；密碼學只能偵測。
freshness	新鮮度	訊息是「新的」；由計數器、時間戳或 nonce 保證。
nonce	一次性數值	number used once；用於新鮮度或隨機化加密。
sequence number / counter	序號 / 計數器	單調遞增；擋重放與重排、偵測遺失。
timestamp	時間戳	需時鐘同步；視窗內仍可重放。

（續下頁）

English	中文	說明
direction bit	方向位元	$b_{A\to B} \neq b_{B\to A}$；擋反射。
sliding window	滑動視窗	容忍遺失與亂序的反重放機制（DTLS、IPsec）。
truncation attack	截斷攻擊	丟掉結尾訊息；需受鑑別的結束訊息偵測。
side-channel attack	側通道攻擊	利用時間、功耗等定義外的洩漏；如 MAC 比對的時間攻擊。
Kerckhoffs' principle	Kerckhoffs 原則	演算法公開，只有金鑰保密。
reduction	歸約	把攻破 X 的敵手轉成攻破 Y 的敵手，證明 Y $\Rightarrow$ X。
negligible	可忽略	比任何多項式的倒數都小（漸近意義）。
ITAR / munitions list	國際武器貿易條例 / 軍火清單	1990 年代美國把密碼軟體列為軍火的法規。
export-grade cryptography	出口級密碼學	刻意削弱（40 位元、512 位元）；FREAK / Logjam 的根源。

14 自我測驗（附解答）

建議先遮住解答、限時 40 分鐘作答。題型刻意模仿「定義- 反例- 論證」的筆試風格。

1. 定義與正確性（基礎）

寫出 SKE 的語法與正確性。接著給一個正確但完全不安全的 SKE，並說明它為何能通過「敵手無法恢復金鑰」這條檢驗。

解答

語法：$\mathsf{sk} \leftarrow \mathsf{KGen}$；$c = \mathsf{Enc}(\mathsf{sk}, m)$；$m = \mathsf{Dec}(\mathsf{sk}, c)$。正確性：$\forall \mathsf{sk}, m : \mathsf{Dec}(\mathsf{sk}, \mathsf{Enc}(\mathsf{sk}, m)) = m$。「什麼都不做」方案 $\mathsf{Enc}(\mathsf{sk}, m) = m$ 正確；因為 Enc 不使用 sk，密文與金鑰獨立，敵手恢復金鑰的機率只有亂猜的 $1/|\mathcal{K}|$——但訊息完全曝光，顯示「金鑰不被恢復」不是正確的安全目標。

2. 是非題（每題一句話說明理由）

(a) 完美保密的加密方案一定能偵測密文是否被竄改。

(b) 一個 MAC 若滿足 (D3)，則它一定滿足 (D1)。

(c) 存在滿足 (D2) 的 MAC，只是我們還沒找到。

(d) 確定性的 MAC 一定不安全，因為同一訊息永遠得到同一標籤。

(e) 確定性的加密一定不是 CPA 安全。

解答

(a) 否：一次性密碼本完美保密但可延展。(b) 是：能恢復金鑰就能對新訊息偽造（引理 9.2）。(c) 否：正確性保證「照抄黑盒輸出」即是偽造，(D2) 對所有正確的 MAC 都不成立（定理 9.5）。(d) 否：MAC 的目標是不可偽造，不是隱藏；同一訊息同一標籤不構成偽造。實務 MAC 幾乎都是確定性的，而且因此自動強不可偽造。(e) 是：查詢 m_0 得 c_0，挑戰 (m_0, m_1)，比對 $c^* \stackrel{?}{=} c_0$ 必勝。

3. 兩個面向

以下定義的（$*$）與（†）各是什麼？「敵手在觀察任意多個由誠實方產生的密文後，無法區分一個新密文對應的是 m_0 還是 m_1。」它比講義的迭代 (v) 強還是弱？為什麼？

解答

（$*$）：以明顯高於 $1/2$ 的機率區分 m_0 與 m_1 的密文。（†）：被動觀察誠實方產生的密文（已知 / 任意明文，但不能選擇）。比迭代 (v) 弱：(v) 允許敵手選擇被加密的明文與被解密的密文（CPA / CCA），能力更強，所以 (v) 的定義更難達成、更強。

4. 反例構造

給定一個滿足 (D3) 的 MAC $(\mathsf{KGen}, \mathsf{Tag}, \mathsf{Vf})$。構造一個 MAC MAC''，使它仍滿足 (D3) 但不是強不可偽造，且你的構造不能使用隨機位元（即 Tag'' 保持確定性）。

解答

令 $\mathsf{Tag}''(\mathsf{ik}, m) = \mathsf{Tag}(\mathsf{ik}, m) \,\|\, 0$，而 $\mathsf{Vf}''(\mathsf{ik}, m, \tau \,\|\, b) = \mathsf{Vf}(\mathsf{ik}, m, \tau)$ 對任何 $b \in \{0, 1\}$ 都接受。Tag'' 是確定性的；對新訊息的偽造仍對應到底層 MAC 的偽造，故滿足 (D3)；但敵手查詢一次得到 $\tau \,\|\, 0$，輸出 $(m, \tau \,\|\, 1)$ 即為新的合法對，違反 $(\mathrm{D3})^+$。注意這個 Vf'' 不是典範驗證（它接受 Tag'' 不會輸出的標籤）——與命題 9.9 一致：典範驗證是「(D3) $\Rightarrow$ $(\mathrm{D3})^+$」的關鍵前提。

5. Encrypt-and-MAC 的洩漏

Alice 使用 Encrypt-and-MAC：送出 $(\mathsf{Enc}(\mathsf{sk}, m), \mathsf{Tag}(\mathsf{ik}, m))$，其中 Enc 是 CPA 安全的隨機化加密、Tag 是確定性的安全 MAC。說明一個只會竊聽的敵手可以學到什麼。

解答

因為 Tag 確定性且算在明文上，$\mathsf{Tag}(\mathsf{ik}, m_i) = \mathsf{Tag}(\mathsf{ik}, m_j) \iff m_i = m_j$：敵手能看出哪些訊息相同、各出現幾次（相等性與頻率洩漏），即使加密本身是隨機化的。在是 / 否投票、公式化訊息的場景中這等於破解。這就是 E&M 不通用安全的原因（Bellare–Namprempre 2000）。

6. 反射攻擊的具體化

Alice 與 Bob 共用 $(\mathsf{sk}, \mathsf{ik})$，使用 Problem 2 的 EtM 流程但沒有方向位元。Alice 送出「$m =$ 請把密碼重設連結寄給我」。描述敵手如何讓 Alice 誤以為 Bob 向她提出同樣的請求；再說明方向位元為何能擋住它。

解答

敵手攔下 $\tilde{c} = (c, \tau)$ 並原封不動送回給 Alice。Alice 用同一把 ik 驗證通過、用同一把 sk 解密成功，得到「請把密碼重設連結寄給我」，以為是 Bob 的請求並照辦——敵手一個位元都沒改。加入方向位元後，c 內含 $b_{A\to B}$；Alice 只接受 $b_{B\to A}$，解析出 $b_{A\to B}$ 時立即拒絕。另一解法是兩個方向使用不同金鑰。

7. 計數器政策的取捨

某即時語音系統在 UDP 上使用 Problem 2 (ii) 的協定。工程師抱怨「只要掉一個封包整通電話就斷了」。問題出在哪？你會怎麼改？改了之後敵手多了什麼能力？

解答

嚴格計數器（只接受 ctr = 期望值）在有自然遺失的通道上會失同步。改用**滑動視窗**：接受視窗內未見過的序號、丟棄視窗外或重複者（DTLS / SRTP 的做法），並由上層處理亂序。代價：敵手可以**選擇性刪除**某些封包而與自然遺失無法區分，且視窗內的重排被容忍；重放仍被擋住。

8. 驗證預言機與側通道

(a) 在典範驗證、確定性 Tag 的假設下，用一句話說明為什麼給敵手 Vf 預言機不會讓它更強。(b) 若 Vf 的實作用逐位元組比較並在第一個不同處提早返回，說明敵手如何利用時間偽造一個 16 位元組的標籤，並估計所需的查詢次數。

解答

(a) $\mathsf{Vf}(\mathsf{ik}, m, \tau)$ 的輸出只是 $[\tau = \mathsf{Tag}(\mathsf{ik}, m)]$，而 $\mathsf{Tag}(\mathsf{ik}, m)$ 敵手本來就能向黑盒查到，所以 Vf 給不出任何新資訊（嚴謹版見 Problem 1 (viii)）。(b) 逐位元組：對第 i 個位元組嘗試 256 個值，拒絕時間最長者即正確（因為比較多進行了一個位元組）。16 個位元組約 $16 \times 256 = 4096$ 次查詢（期望值約一半），遠少於暴力的 2^{128}。對策：常數時間比較。

9. 綜合申論

「只要加密夠強，就不需要 MAC。」請以不超過 150 字反駁，至少引用本週的兩個概念。

解答

保密性與完整性是不同的性質，需要不同的工具：一次性密碼本是最強的保密（完美保密），卻可被位元翻轉任意竄改（可延展性）。敵手不需讀懂密文就能把「付 100」改成「付 900」。要讓收件端能偵測竄改，必須有一個依賴另一把秘密金鑰、且敵手無法偽造的標籤——這就是 MAC（EUF-CMA）。正確的組合是 Encrypt-then-MAC：對密文貼標籤、先驗證後解密。

10. 歷史與社會

用三句話說明「刻意削弱密碼學」在歷史上造成了什麼後果，並連結到 Blanchette 對密碼學的定義。

解答

1990 年代美國出口管制迫使 Netscape 等產品在國際版使用 40 位元對稱金鑰與 512 位元 RSA。這些「出口級」套件在 2015 年仍被許多伺服器支援，成為 FREAK 與 Logjam 降級攻擊的入口——弱點比政策活得更久。這印證 Blanchette 的觀點：密碼學是規範行為的程式碼，因此必然被政治與法律介入，而介入的技術後果會長期留存。

參考資料

課程教材

- N. K. Nguyen and E. W. Postlethwaite, *6CCS3CIS Cryptography, first lecture*, King's College London, 11 September 2025（73 頁投影片）。
- E. W. Postlethwaite, *Problem Set 1*, 6CCS3CIS Cryptography, 29 September 2025.

教科書

- J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 3rd ed., CRC Press, 2021. 本週相關節次：§1.2（加密的基本設定）、§1.4（現代密碼學三原則）、第 2 章（完美保密）、§3.2（計算安全的定義）、§3.4（CPA 安全）、§4.1–4.2（訊息完整性、MAC 定義、典範驗證、強不可偽造、驗證查詢、時間攻擊）、§5.2–5.4（鑑別加密、通用組合、安全通訊 session）。
- M. Rosulek, *The Joy of Cryptography*, 線上草稿，`https://joyofcryptography.com/`。相關章節：序言（「控制資訊存取」）、Chapter 1（Kerckhoffs 原則）、MAC 章節（以函式庫不可區分性定義不可偽造性；Encrypt-then-MAC 的 CCA 安全證明）。

原始文獻

- M. Bellare and C. Namprempre, "Authenticated Encryption: Relations among Notions and Analysis of the Generic Composition Paradigm," *ASIACRYPT 2000*, LNCS 1976; 完整版

Cryptology ePrint Archive 2000/025, https://eprint.iacr.org/2000/025.

- H. Krawczyk, "The Order of Encryption and Authentication for Protecting Communications (or: How Secure Is SSL?)," *CRYPTO 2001*; ePrint 2001/045.
- C. Namprempre, P. Rogaway and T. Shrimpton, "Reconsidering Generic Composition," *EUROCRYPT 2014*; ePrint 2014/206.
- M. Bellare, O. Goldreich and A. Mityagin, "The Power of Verification Queries in Message Authentication and Authenticated Encryption," ePrint 2004/309.
- S. Goldwasser and S. Micali, "Probabilistic Encryption," *J. Computer and System Sciences* 28(2), 1984（語意安全的原始論文）。
- C. E. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical Journal* 28(4), 1949.
- A. Kerckhoffs, "La cryptographie militaire," *Journal des sciences militaires*, 1883.
- M. Naveed, S. Kamara and C. V. Wright, "Inference Attacks on Property-Preserving Encrypted Databases," *ACM CCS 2015*（確定性加密欄位的頻率攻擊）。
- E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, RFC 8446, 2018；E. Rescorla, H. Tschofenig and N. Modadugu, *DTLS 1.3*, RFC 9147, 2022（序號、每方向金鑰、滑動視窗反重放）。

Problem 3 指定閱讀與相關史料

- R. J. Stay, "Cryptic Controversy: U.S. Government Restrictions on Cryptography Exports and the Plight of Philip Zimmermann," *Georgia State University Law Review* 13(2), Art. 14, 1997, pp. 581–604. https://readingroom.law.gsu.edu/gsulr/vol13/iss2/14
- J. C. Fromer, "Constitutional Law. Free Speech Clause. Sixth Circuit Classifies Computer Source Code as Protected Speech. *Junger v. Daley*, 209 F.3d 481 (6th Cir. 2000)," *Harvard Law Review* 114(6), 2001, pp. 1813–1818. https://doi.org/10.2307/1342657
- *Junger v. Daley*, 209 F.3d 481 (6th Cir. 2000); *Junger v. Daley*, 8 F. Supp. 2d 708 (N.D. Ohio 1998).
- S. Garfinkel, "The Continuing Investigation of Phil Zimmermann," *Wired*, March 1995（1994 年 11 月杜勒斯機場攔查）。
- P. Zimmermann, *PGP Source Code and Internals*, MIT Press, 1995.
- 美國司法部 1996 年 1 月 11 日不起訴聲明；見 https://philzimmermann.com/EN/news/PRZ_case_dropped.html。
- B. Beurdouche et al., "A Messy State of the Union: Taming the Composite State Machines of TLS," *IEEE S&P 2015*（FREAK）；D. Adrian et al., "Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice," *ACM CCS 2015*（Logjam）。

歷史案例

- National Security Agency, *The Battle of Midway: How Cryptology Enabled the United States to Turn the Tide in the Pacific War*（「AF is short of water」）。
- S. Vaudenay, "Security Flaws Induced by CBC Padding," *EUROCRYPT 2002*；D. Bleichenbacher, "Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1," *CRYPTO 1998*（解密預言機的現實性）。

本教材為學習輔助用途，內容依據公開講義與文獻整理；如與課程正式公告有出入，以 KEATS 公告為準。