Lecture 1
Overview of Cryptographic Techniques

Jason Lin

Outline

- 密碼學的基本概念
- 秘密金鑰密碼系統
- 公開金鑰密碼系統
- 單向雜湊函數
- 電子商務應用
- 其他分支研究領域

何謂密碼學? (1/2)

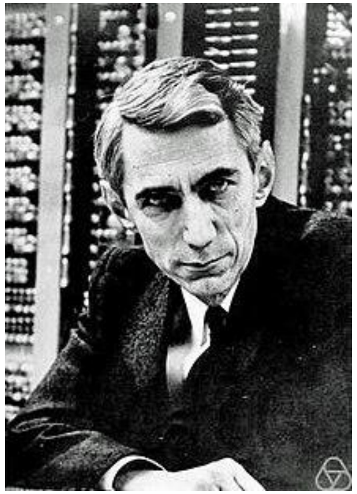
Cryptology (研究密碼的學門)
From the Greek for "Greek *Kruptos* (隱藏) and *Logos* (訊息),"

Cryptography (密碼學)
Code-making (創造密碼)

Cryptanalysis (破密分析)
Code-breaking (破解密碼)

何謂密碼學? (2/2)

- 密碼學 (Cryptography)：研究如何透過數學方法達到資訊傳輸的秘密性 (Secrecy) 與鑑定性 (Authenticity) 之科學
- 西元 1949 年資訊理論的創始者 Claude Shannon 提出第一篇討論密碼系統通訊理論之論文
- 歷史實例：
  - 東方，中國清朝大學士紀曉嵐題藏頭詩
  - 西方，美軍利用納瓦霍密碼 (Navajo code) 來傳送軍事情報



Claude Shannon

C. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, 28 (4): 662, 1949.

清朝密碼學家

「精神炯炯
老貌堂堂
烏巾白鬚
龜鶴呈祥」---紀曉嵐

納瓦荷族密碼兵

- 第二次世界大戰

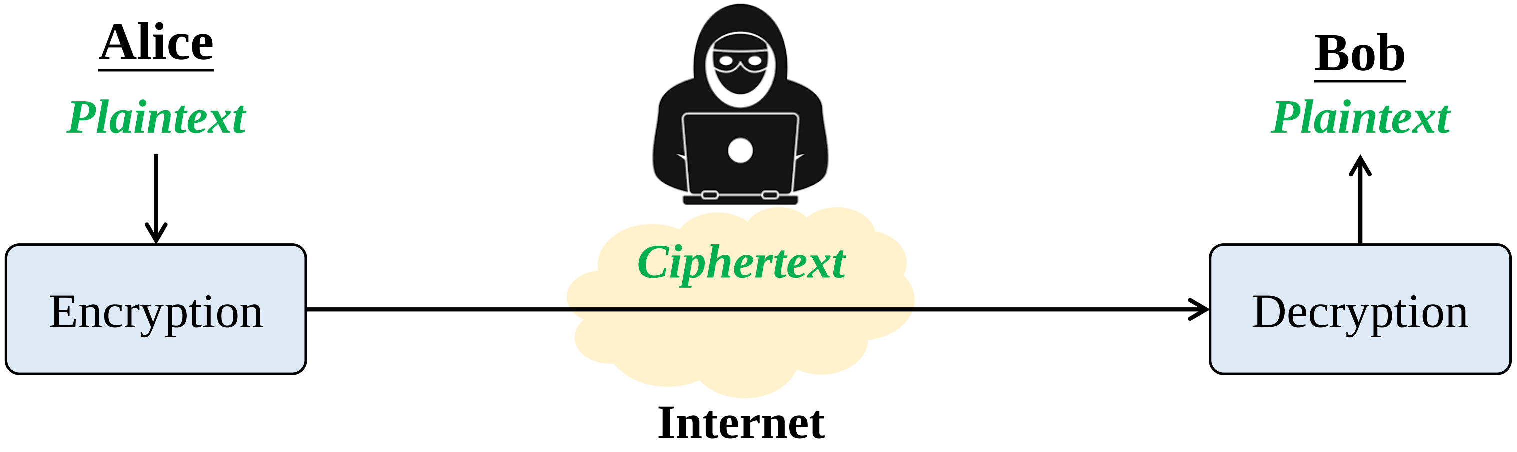
密碼學之術語

- 明文 (Plaintext)：原始可辨識的訊息 (Message)
- 密文 (Ciphertext)：轉換過難以辨識的訊息
- 金鑰 (Key)：使用在加密器中相當重要且關鍵性的資訊，只有傳送方與接收方知道的資訊
- 加密 (Encipher, Encrypt, Encode)：使用加密器與金鑰去轉換明文成密文的過程
- 解密 (Decipher, Decrypt, Decode)：使用解密器與金鑰去轉換密文回明文的過程

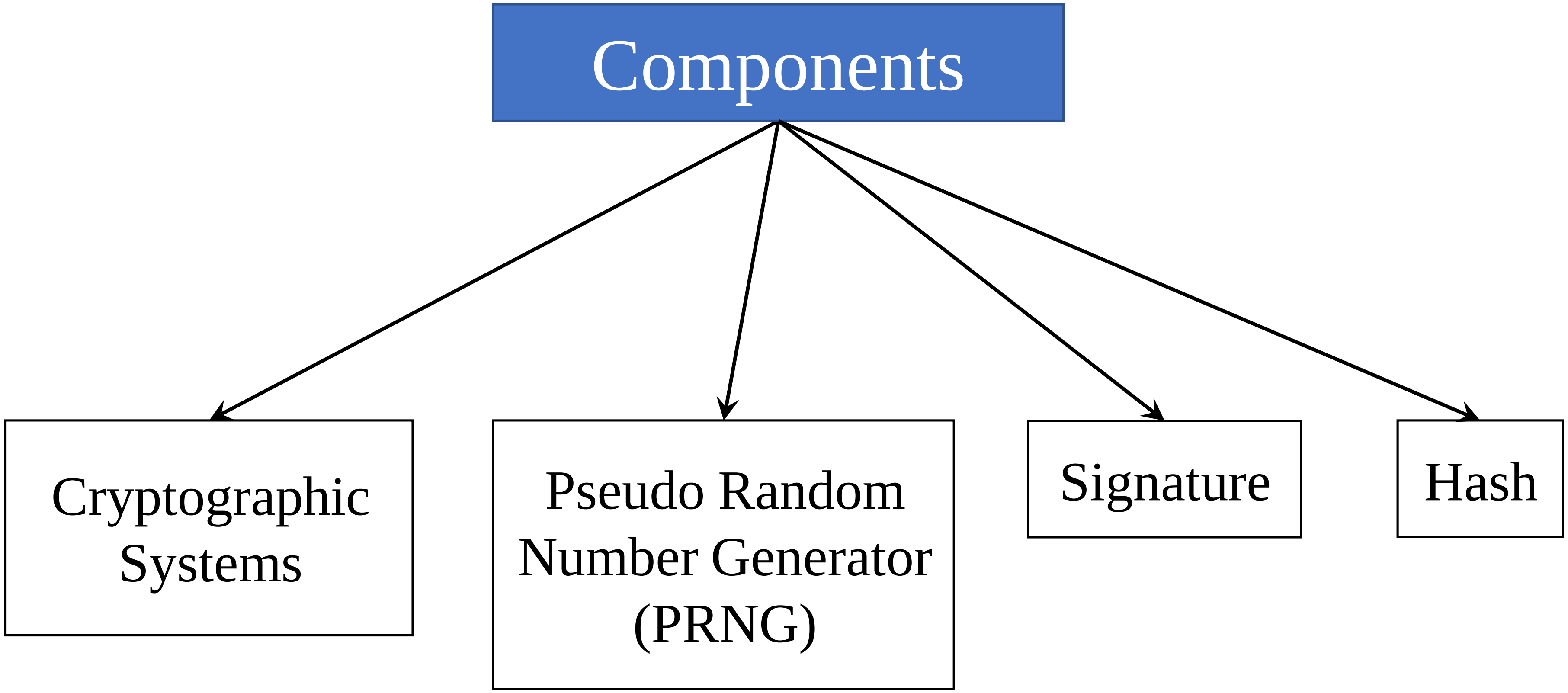
為何需要密碼學？

- 一般在網路參考模型 OSI 的第七層–應用層中的超文本傳輸安全協定 (Hypertext Transfer Protocol，HTTP)，預設採用「明文」來進行傳輸，這對網頁上的資料交換過程來說缺乏安全性
- 因此，在 HTTP 上增加一層 SSL/TLS 以將傳輸的資料進行加密，將傳輸的資料變成「密文」，成為 HTTPS (Secure HTTP)，這樣即使有心人士在中途攔截這些資料，也無法直接破譯

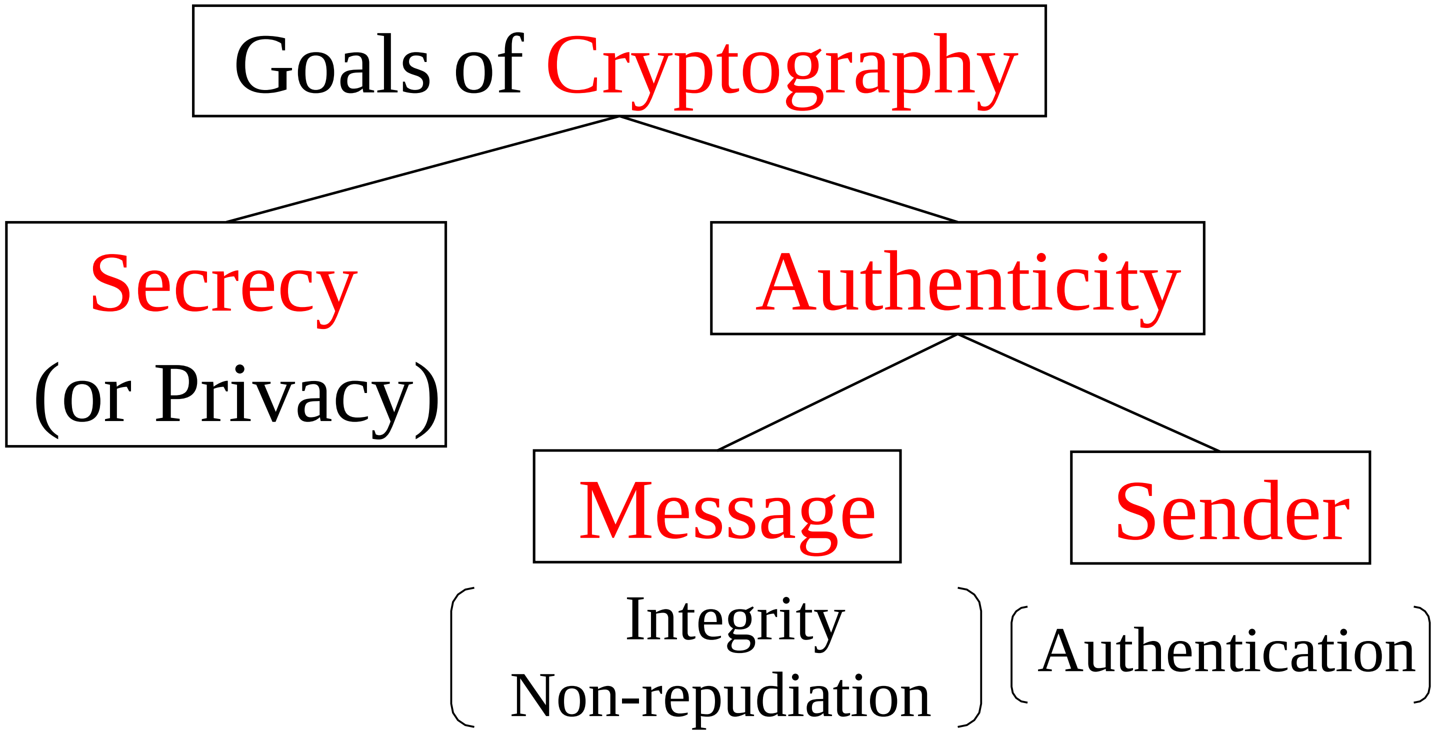
密碼系統之簡圖


Alice
Plaintext
Encryption
Ciphertext
Internet
Bob
Plaintext
Decryption

密碼元件


Components
Cryptographic Systems
Pseudo Random Number Generator (PRNG)
Signature
Hash

密碼學之目標



We shall see that secrecy and authenticity are **<u>independent</u>** attributes of a cryptographic system.

Kerchoff's (1835-1903) 原理之安全假設

- 密碼系統之安全性必須僅依賴其解密金鑰，而非其加/解密演算法。在這種給破密者最大知識的假設中，我們欲分析一套密碼系統之安全性，必須假設破密者除了解密金鑰及其直接相關資訊外，其密文、加解密演算法，甚至是某些密文所對應的明文皆為破密者所知。而在這種假設前提下，如果仍可以抵抗所有的演算法，則此密碼系統才可被稱之為安全。
- 美國數學家 Claude Shannon (資訊理論的先驅) 將此詮釋為 "the enemy knows the system"

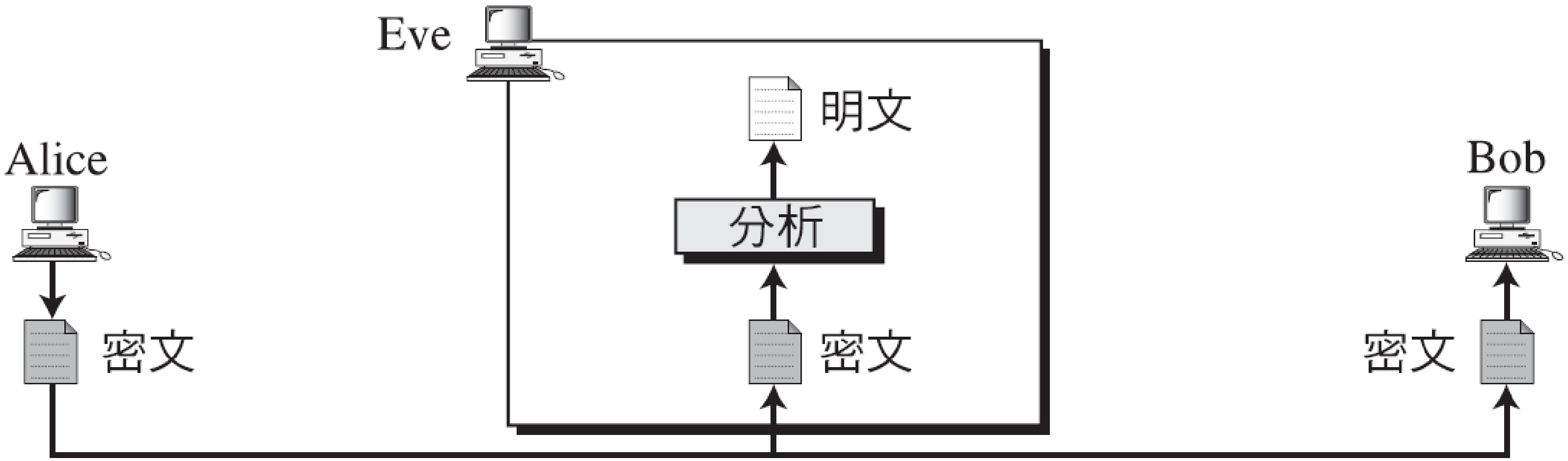
C. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, 28 (4): 662, 1949.

破密分析 (1/5)

- 破密者依在密碼系統中所收集的資訊，依層次有下列四種可能的破密分析攻擊：
  1. 只知密文攻擊法 (Ciphertext-Only Attack)
  2. 已知明文攻擊法 (Known-Plaintext Attack)
  3. 選擇密文攻擊法 (Chosen-Ciphertext Attack)
  4. 選擇明文攻擊法 (Chosen-Plaintext Attack)
- 一般密碼系統必須至少禁得起「已知明文攻擊法」，而公開金鑰密碼系統需禁得起「選擇明文攻擊法」

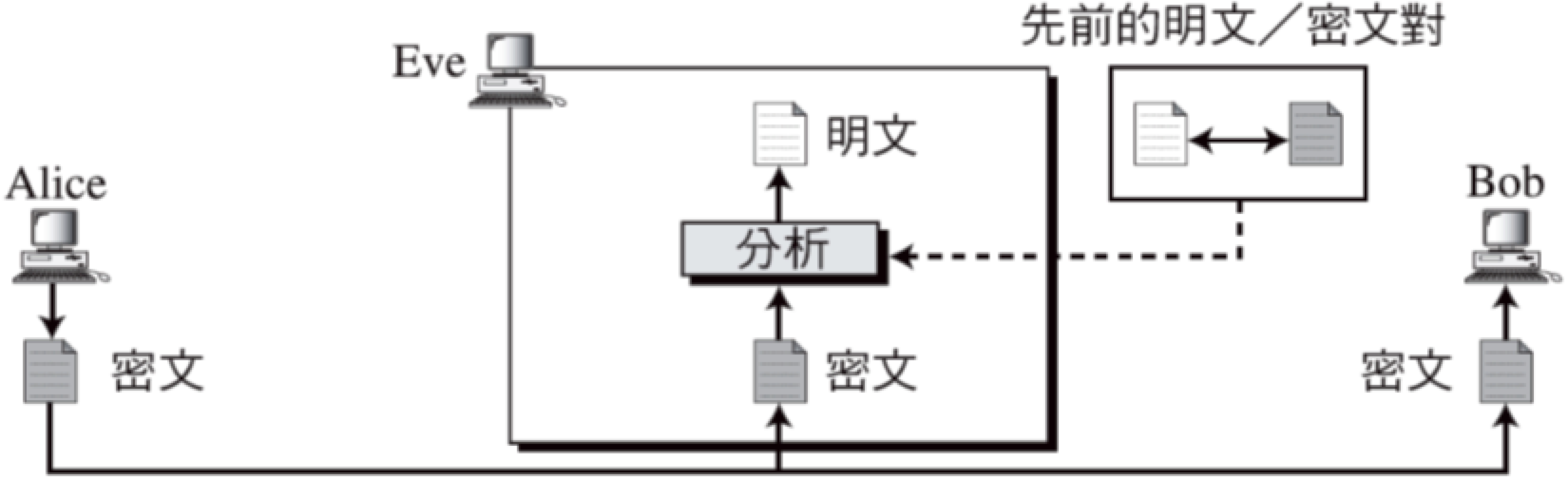
破密分析 (2/5)

- 只知密文攻擊法
  - The cryptanalyst want to find $m_1, m_2, \ldots, m_n$ or K from $c_1, c_2, \ldots, c_n$.



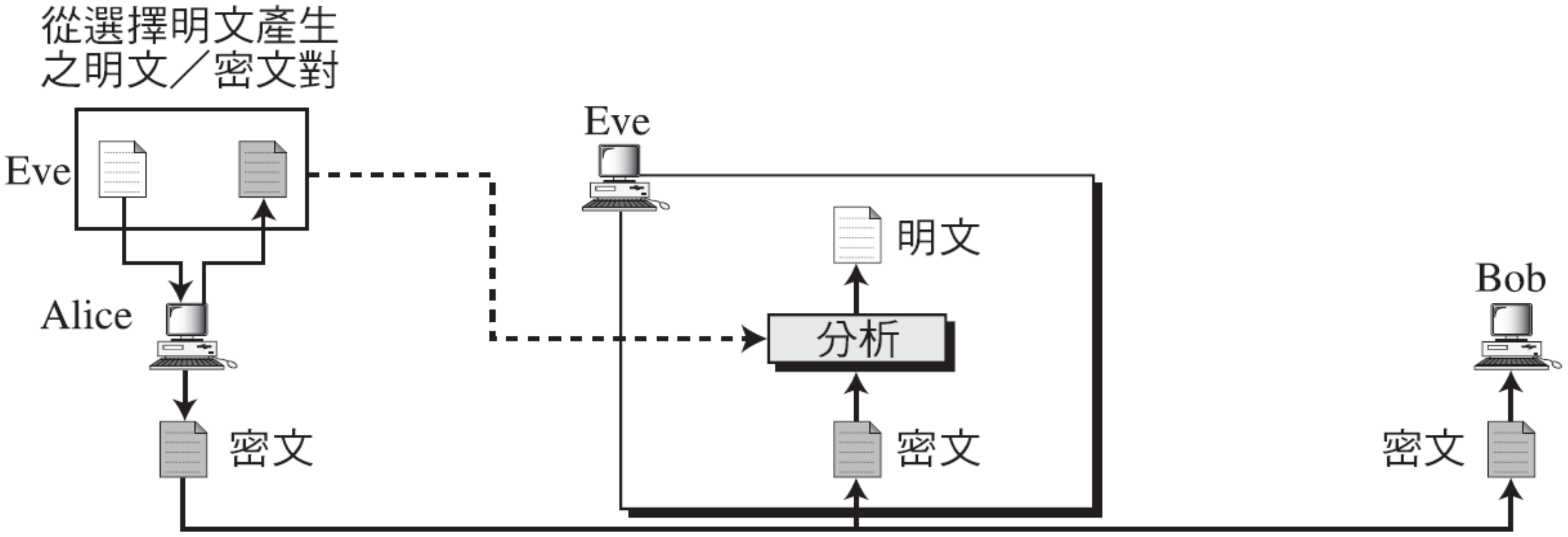
破密分析 (3/5)

- 已知明文攻擊法
  - Given the pairs $\{m_1, c_1\}$, $\{m_2, c_2\}$, …, $\{m_n, c_n\}$, the cryptanalyst wants to derive K or m_{n+1} from c_{n+1}. However, the pairs cannot be controlled by the cryptanalyst.



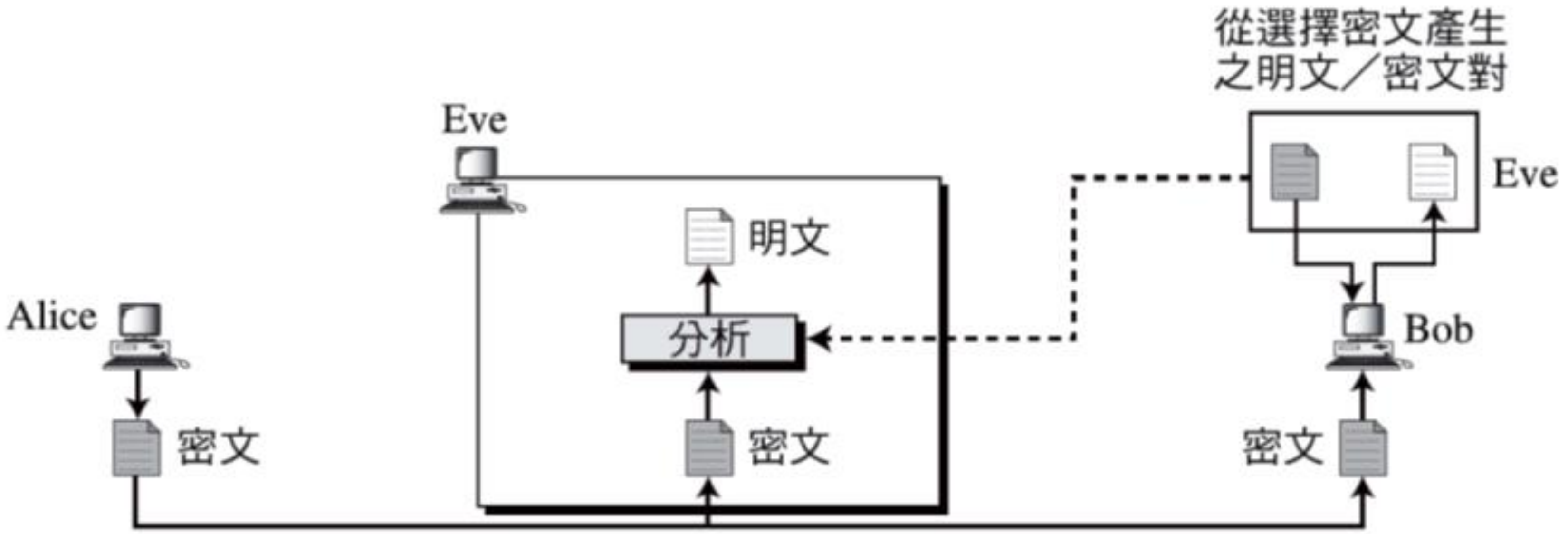
破密分析 (4/5)

- 選擇明文攻擊法
  - The cryptanalyst can gather information by obtaining the encryptions of chosen plaintexts.

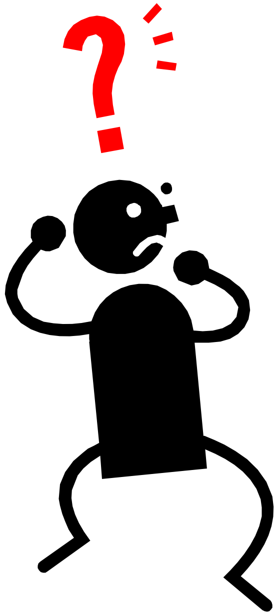


破密分析 (5/5)

- 選擇密文攻擊法
  - The cryptanalyst can gather information by obtaining the decryptions of chosen ciphertexts.



密碼系統之安全性



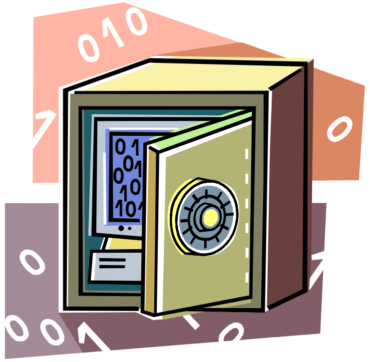
Guess directly

$$\text{Plaintext: 1 bit}\begin{cases}\text{Boy}\\ \text{Girl}\end{cases}$$

Boy/Girl
50%

Boy/Girl
50%

理論安全與實際安全 (1/2)



- 理論安全 (Theoretical Security or Unconditional Security)
  - 不管破密者有多強大的計算能力或時間皆無法解出明文，只有 one-time pad system 可達成，但並無真正 one-time pad system 存在。
  - 假設密碼元件金鑰為 k 位元，對 n 位元的明文加密，若破密得到明文之機率＝直接猜對明文之機率，則稱此密碼元件為理論安全。
  - 達到理論安全之必要條件為 $k \geq n$。
- 實際安全 (Practical Security or Computational Security)
  - 若密碼系統無法達到理論安全，但卻無法在合理的範圍內破解，比如說所付出的代價超出破解所得到的好處，或者所需的時間超出明文可利用的時間，則稱此系統為實際安全。

理論安全與實際安全 (2/2)

- 理論工作函數 (Work Characteristic, $W(n)$)
  - 破解某一密碼系統，在理論上所需之最少代價
- 歷史理論工作函數 (Historical Work Characteristic, $Wh(n)$)
  - 破解某一密碼系統，現在已知所需之最少代價
- 某一密碼系統被宣稱為安全，是指其歷史工作函數無法在合理範圍內達成。換言之，某一密碼系統目前為安全並不保證未來仍為安全

利用金鑰搜尋窮舉法破解系統所需時間表

Key Size (bits)	Number of Alternative Keys	Time required at 1 decryption/µs	Time required at 10^6 decryptions/µs
32	$2^{32} = 4.3 \times 10^9$	2^{31} µs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} µs = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} µs = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} µs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} µs = 6.4×10^{12} years	6.4×10^6 years

µs（微秒）：10^{-6} 秒

金鑰搜尋窮舉法破解系統範例

- 可利用 Office 軟體或 WinZip 密碼加密保護檔案的方法建立密文檔，再利用下面連結中，Home Users 分類底下的軟體 Advanced Office Password Recovery 或 Advanced Archive Password Recovery (Free trial version, 30 days)，採用金鑰搜尋窮舉法破解密文檔得到明文及金鑰
- 網站：https://www.elcomsoft.com/products.html

密碼系統之分類 (1/2)

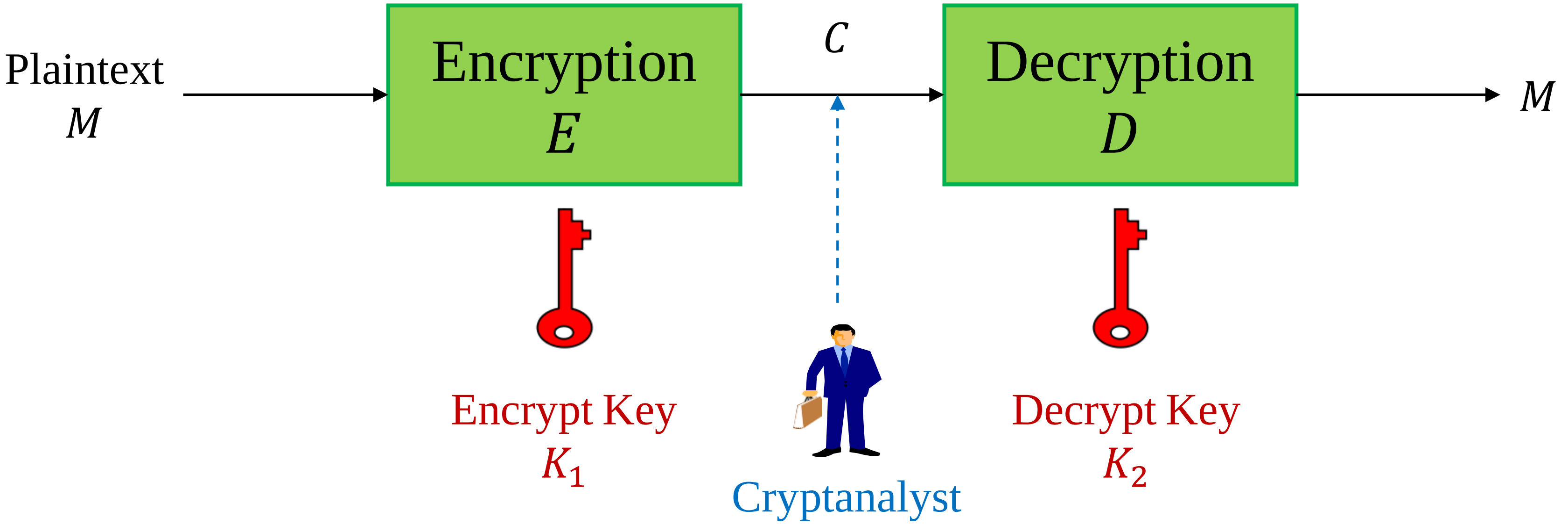
- 密碼系統可用以下三種規則來分類：
  1. 依轉換明文成密文的動作分類
     - Substitution
     - Transposition
  2. 依金鑰的數目分類
     - Secret-key encryption
     - Public-key encryption
  3. 依處理明文的方式分類
     - Block cipher
     - Stream cipher
- 一般通常以第二種規則來分類密碼系統

密碼系統之分類 (2/2)

- Secret-key encryption (秘密金鑰加密):
  - It is also called symmetric encryption (對稱式加密), which uses the same key for encryption and decryption.
  - Security depends on the sender and receiver possessing some common secret that is unknown to the enemy cryptanalyst.
- Public-key encryption (公開金鑰加密):
  - It is also called asymmetric encryption (非對稱式加密), which uses separate keys for encryption and decryption.
  - Security depends on the sender and receiver possessing some common trusted information, which one assumes that the enemy cryptanalyst also knows.

秘密金鑰與公開金鑰加密之比較 (1/3)

- When $K_1 = K_2$, the system is called symmetric encryption, where both K_1 and K_2 are called secret keys (秘鑰).
- When $K_1 \neq K_2$, the system is called asymmetric encryption, where K_1 is called public key (公鑰) and K_2 is private key (私鑰).



秘密金鑰與公開金鑰加密之比較 (2/3)

	功能	**優點**	**缺點**
秘密金鑰加密	1. 保護機密資訊 2. 鑑定收送方之身份 3. 確保資訊完整性	1. 速度快 2. 秘密金鑰長度短 3. 計算能力弱的機器亦可執行	1. 金鑰分配問題 2. 為能與多人秘密通訊，需保存的金鑰數目太多 3. 無法達到不可否認性
公開金鑰加密	1. 保護機密資訊 2. 可做數位簽章	1. 簡化金鑰分配及管理 2. 可達到不可否認性	1. 速度慢 2. 金鑰長度長 3. 計算能力弱的機器執行吃力

秘密金鑰與公開金鑰加密之比較 (3/3)

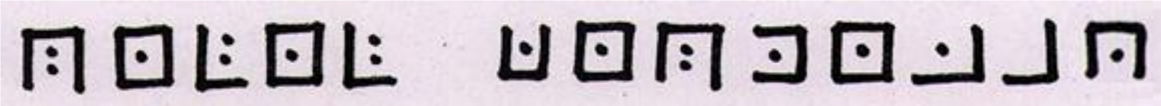
- DES 如用硬體製作可達每秒 45M 位元之加解密，而 RSA 差不多每秒 50K 位元，相差 1000 倍
- 故常用混合型密碼系統 (Hybrid Cryptosystem)，就是用公開金鑰密碼系統分配秘密金鑰，再用此秘密金鑰來加解密訊息

秘密金鑰加密之古典技術 (1/3)

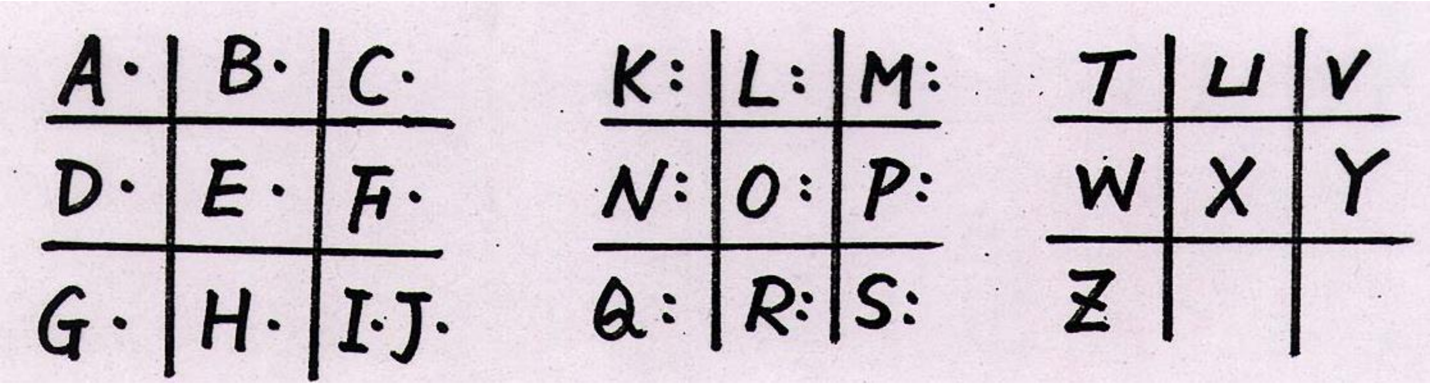
- 使用兩種基本元件
  - 代換法 (Substitution)
    - 原字母用其他字母來取代
    - 如：豬圈加密法
  - 換位法 (Transposition)
    - 原字母用不同的順序排列
    - 如：鐵軌柵欄加密法

秘密金鑰加密之古典技術 (2/3)

- 豬圈加密法 (Pigpen Cipher)
  - 紐約州，Trinity 市一座教堂旁的墓碑文

密文：

依照下列代換法則取代而成



明文：REMEMBERDEATH

秘密金鑰加密之古典技術 (3/3)

- 鐵軌柵欄加密法 (Rail Fence Cipher)
  - 軌道長度為 2
  - Plaintext : meet me after the toga party

m e m a t r h t g p r y

e t e f e t e o a a t

  - Ciphertext : MEMATRHTGPRYETEFETEOAAT

秘密金鑰加密之現代技術 (1/6)

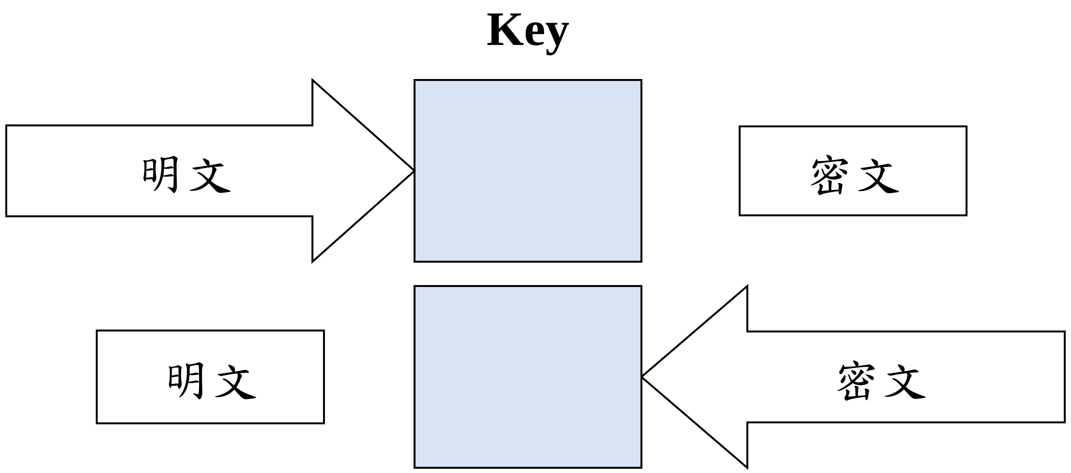
- 現代秘密金鑰加密主要以代換 (Substitution) 及換位 (Transposition) 為基本轉換方式，其運算速度快，適合用於大量資料之加解密。
- 分類
  - 區塊加密器 (Block Cipher)
    - Ex., DES, Triple DES, IDEA, AES
  - 串流加密器 (Stream Cipher)
    - Ex., RC4
  - Block Cipher Mode of Operation
    - Ex., ECB, CBC, CFB, OFB
  - Password-based Cryptography
    - Ex., PKCS #5
  - …

秘密金鑰加密之現代技術 (2/6)

- 現代秘密金鑰加密以區塊加密器的使用最為廣泛
- 一般區塊密文 (Block Cipher) 加密器之設計，須注意到以下特性
  - 迷惑性：明文及密文對金鑰的關係無法用數學式分析
  - 擴散性：改變明文或金鑰的任一位元，密文之所有位元均可能影響
  - 規則性：以利實現
  - 簡單性：快速運算
  - 相似性：加密器解密器相同以節省成本

秘密金鑰加密之現代技術 (3/6)

- 1976 年美國國家標準局 NIST (National Institute of Standards and Technology) 公佈為資料加密標準 DES (Data Encryption Standard)
- 十六回合：擴充轉換、二進位加法、代換轉換、排列轉換

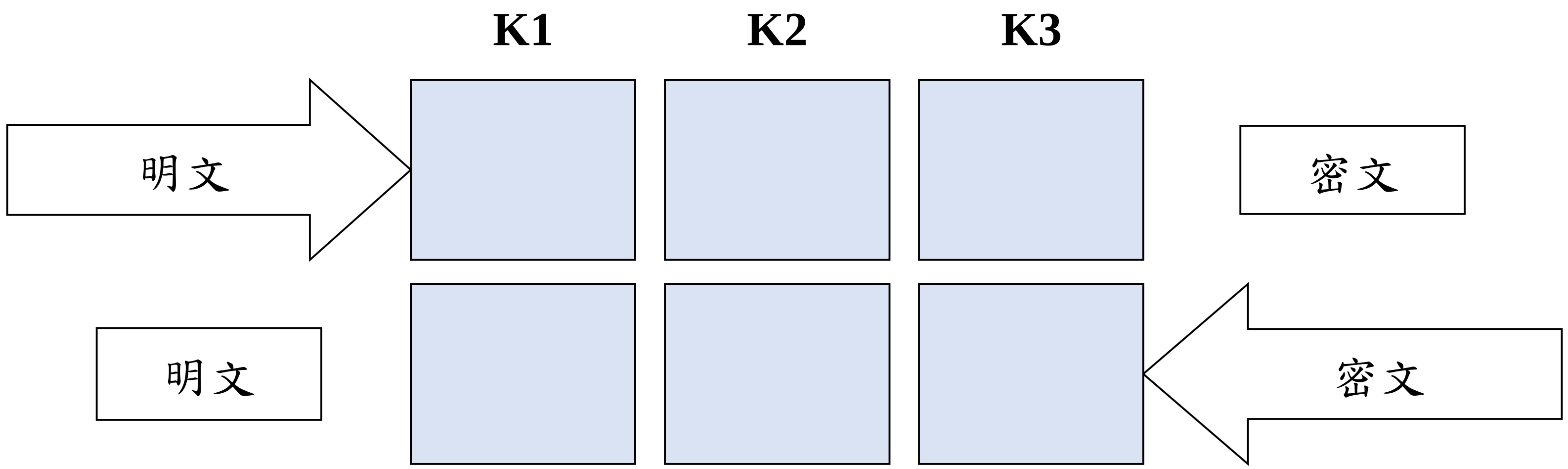


秘密金鑰加密之現代技術 (4/6)

- 安全?
  - 金鑰 56 bits ($2^{56} \approx 10^{17}$)
  - 若破解速度 10^6 key/sec ≈ 3×10^{13} keys/year
  - 使用暴力搜尋需要約 3000 年。
  - 但如果使用 10^6 台電腦同時運轉， 則只需 3000年/$10^6 \approx 26.28$ 小時即可破解。

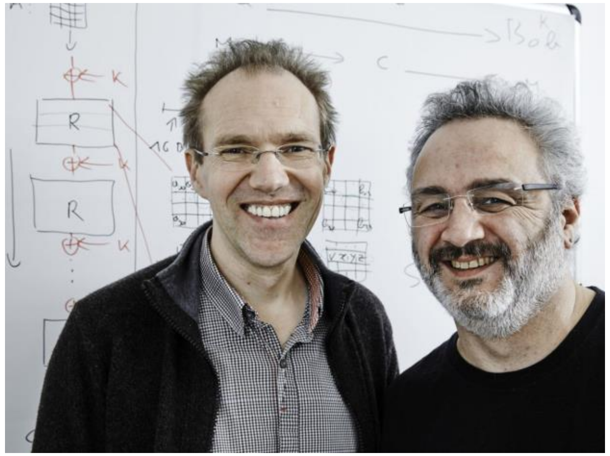
秘密金鑰加密之現代技術 (5/6)

- Triple DES
  - $K_1 \neq K_2 \neq K_3$, 168 bits
  - $K_1 = K_3 \neq K_2$, 112 bits



秘密金鑰加密之現代技術 (6/6)

- 2000 年由比利時密碼學家 Vincent Rijmen 和 Joan Daemen 所開發
  - Rijndael Block Cipher (AES的前身)
  - 128, 160, 192, 224, 256
- 2002 年美國國家標準局公佈為資料加密標準 AES (Advanced Encryption Standard)。
  - AES-128 (10), AES-192 (12), AES-256 (14)
  - Less memory
  - High efficiency



Joan Daemen (left) and Vincent Rijmen (right)

公開金鑰加密 (1/2)

- 公開金鑰加密是植基於數學難題，其運算速度慢，但無須考慮金鑰之分配，用於較少量之資料加解密，如保護對話金鑰或使用者之個人資訊。
  - Ex., RSA , ElGamal
- NOTE：公開金鑰加密與數位簽章 (Digital Signature) 均為一組 Public Key 與 Private Key，若是使用者的 Public Key 有認證憑據 (Certification)，則此 Public Key 可用作身分鑑別 (Authentication)。

公開金鑰加密 (2/2)

- Diffie, Hellman, 1976
- Rivest, Shamir, Adleman (RSA), 1977
  - Factorization Problem
- ElGamal, 1985
  - Discrete Logarithm Problem

公開金鑰加密之基礎

- 常見的數學運算
- 有限體
- 主要的數學難題
  - 背包問題
  - 質因數分解問題
  - 離散對數問題
  - 橢圓曲線的離散對數問題
  - …

許多公鑰密碼系統植基於一些數學難題，若能具備相關之背景，將可收事半功倍之效果。

常見的數學運算

- 模運算 (Modular Reduction)
  - $b = a \pmod{n}$
- 模乘法 (Modular Multiplication)
  - $c = a \times b \pmod{n}$
- 模指數運算 (Modular Exponentiation)
  - $c = a^b \pmod{n}$
- 求模乘法反元素 (Inverse of Modulus)
  - 求 b (一般記為 a^{-1})，使得 $a \times b = 1\ (mod\ n)$

有限體 (1/2)

- 代數 (Algebra) 的定義及性質
  - 群 (Group) → 在單一運算中具有封閉性、結合性且有單位元素及反元素之集合
  - 交換群 (Commutative Group) → 具有交換性之 Group
  - 體 (Field) → 對於某兩個運算 (如 + 及 ∗)，符合以下三條件之集合
    - 對於加法 (+) 而言，為一 Commutative Group
    - 對於乘法 (∗) 而言，扣除加法之單位元素，為一 Group
    - 乘法對加法具有分配性，也就是 $a * (b + c) = (a * b) + (a * c)$
- 有限體 (Finite Field，又稱 Galois Field) → 有限個元素之 Field，Galois Field 之元素個數稱為它的階 (Order)。

有限體 (2/2)

- 許多密碼元件常用之運算皆定義在 Galois Field (GF) 下
- 對於每個質數 p 與每個正整數 n，存在唯一的有限體 $GF(p^n)$
- 常用之 Galois Field 主要包括 $GF(p)$ 及 $GF(2^n)$
- 植基於 $GF(p)$ 之運算，可視為 mod p 之運算
- 植基於 $GF(2^n)$ 之運算，可視為 mod $(a_{n-1}x^{n-1} + \cdots + a_1x + a_0)$ 之運算，其中 $a_{n-1}, \ldots, a_1$ 及 $a_0 \in \{0,1\}$

主要的數學難題 (1/2)

- 許多密碼系統(尤其公鑰密碼系統)之安全基礎植基於數學難題之上，這些數學難題須具有以下特徵之一
  - 單向函數 (One-way Function)
    - 一函數 $y = f(x)$，求 $f(x)$ 容易但是求 $f^{-1}(y)$ 困難。
  - 單向暗門函數 (One-way Trapdoor Function)
    - 一函數 $y = f(x)$，求 $f(x)$ 容易但是求 $f^{-1}(y)$ 困難，但若已知某額外資訊，則不難求出 $f^{-1}(y)$。
- 以下介紹不重定理說明，注重做法及應用說明。
- 相關密碼系統說明，容後配合各相關部份依序介紹。

主要的數學難題 (2/2)

- 數學難題的類型
  - 背包問題 (Knapsack Problem)
    - $S = m_1B_1 + m_2B_2 + \cdots + m_nB_n$ → 計算 S 易，求 $m_1, m_2, \ldots, m_n$ 難
  - 質因數分解問題 (Prime Factorization)
    - $n = p \times q$ → 計算兩個質因數的乘積 $p \times q$ 易，分解 n 難
  - 離散對數問題 (Discrete Logarithm Problem, DLP)
    - $z = x^y \ (mod\ p)$ → 計算 z 易，求得 y 難
- 橢圓曲線密碼系統 (Elliptic Curve Cryptosystem, ECC)
  - 可將橢圓曲線配合上述解離散對數問題來產生密碼系統

金鑰交換協定

- 目標
  - 使用者透過公用通道交換秘密資訊
- 使用時機
  - 兩位以上之使用者，透過公用通道協議彼此對話金鑰 (Session Key) 之程序，此對話金鑰多為對稱式加密系統之秘鑰 (Secret Key)，可於每次使用時更換，以確保其安全性
- 代表性系統
  - Diffie-Hellman (DH) Key Exchange System，是基於 Hellman 在史丹佛大學的博士生 Ralph Merkle 所提出的概念，又可稱 Key Agreement 或者是 Key Determination

Merkle Puzzle (1/5)

- 起源
  - 金鑰交換協定最早是由大學時期的 Ralph Merkle 於 1974 年在一場演講中所提出的想法，並在 1978 年讀研究所時發表
- 目標
  - 在公眾網路下進行一連串通訊，Alice 與 Bob 可以獲得一個共同的 Common Secret Key，破密者 Charlie 亦可獲得此 Key，但會需要很多的時間來破解
- 假設
  - Alice 與 Bob 沒有 Common Secret Key
  - Alice 與 Bob 有相同的安全對稱式加/解密演算法 ($|K| \geq 32$)
  - 運算一次加解密需要 10^{-4} 秒

Ralph Merkle

Merkle Puzzle (2/5)

- Main tool: puzzles
- Problems that can be solved with some efforts
  - Ex., $puzzle(key) = E(key, \text{“}message\text{”})$ where $key = 0^{96}||b_1 \dots b_{32}$ and $E(k, m)$ is an agreed symmetric cipher (e.g., AES) with $k \in \{0,1\}^{128}$
  - Goal: find key by trying all 2^{32} possibilities

Merkle Puzzle (3/5)

Alice

Bob

- She prepares 2^{32} puzzles

$M_{AB} \parallel x_i \parallel s_i \Rightarrow \boxed{E} \Rightarrow puzzle_i$

k_i

M_{AB}^i為雙方事先約定好的明文："Puzzle # "

- For $i = 1, \dots, 2^{32}$, choose random $k_i \in \{0,1\}^{32}$ and $x_i, s_i \in \{0,1\}^{128}$

Puzzle ID

Secret

$$\xrightarrow{puzzle_1, \dots, puzzle_{2^{32}}}$$

Merkle Puzzle (4/5)

Alice

Bob

- Choose $puzzle_j$ from all puzzles

$$M_{AB} \,||\, x_j \,||\, s_j \Leftarrow \boxed{D} \Leftarrow puzzle_j$$

k_i (Try all possible values)

$$\xleftarrow{\quad x_j \quad}$$

- Lookup puzzle with the identifier number x_j

完成後 s_j 即為雙方協議的一把共同密鑰

Merkle Puzzle (5/5)

- 複雜度分析
  - Alice has to prepare n puzzles ($n = 2^{32}$)
  - Bob has to solve one puzzle (2^{32} keys)
  - Charlie's eavesdropping work: $(2^{32})^2 = 2^{64}$ times

User	Alice	Bob	Charlie
$O(.)$	n	n	n^2

Diffie-Hellman 金鑰交換協定 (1/4)

- 1976 年由 Diffie 與 Hellman 所提出，也是最早提出的公開金鑰概念的非對稱式加密演算法
  - 植基於 DLP
- 在一般協定標準中，使用固定之系統參數，以下兩個協定均使用 Diffie-Hellman Key Agreement System
  - Oakley Key Determination Protocol (RFC 2412)
  - Simple Key Management for Internet Protocol (SKIP)

Whitfield Diffie (left) and Martin Hellman (right)

Diffie-Hellman 金鑰交換協定 (2/4)

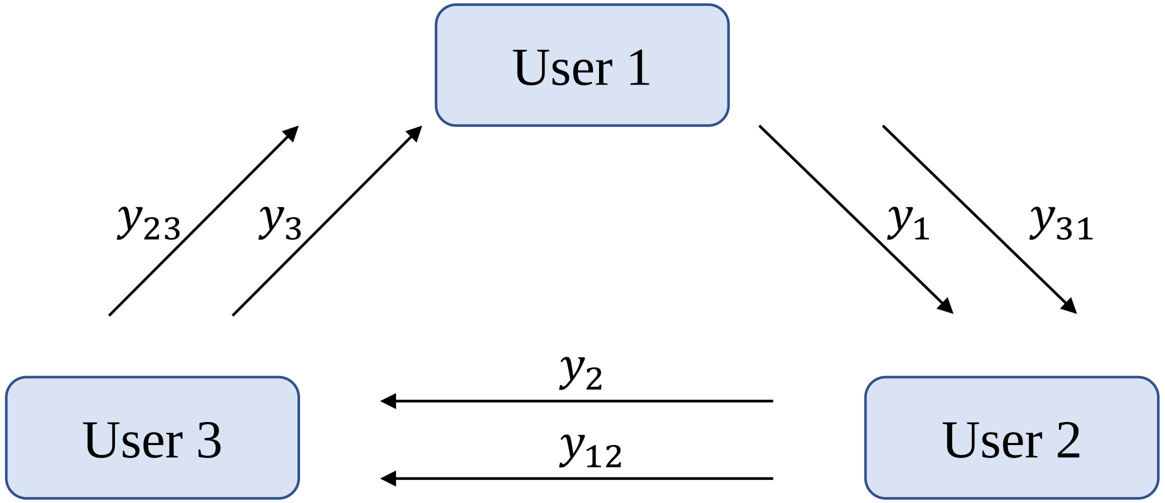
- 系統參數 (所有使用者事先所協議出的共用系統參數)
  - p : 一足夠大之質數，1024 位元以上。
  - g : mod p 之一原根 (Primitive Root)，即任一 $x \in [1, p-1]$ 皆可找到一個對應的 y，使得 $y = g^x \pmod{p}$
- 使用者參數
  - 密鑰: $x, x \in [1, p-1]$
  - 公鑰: $y, y = g^x \pmod{p}$

Diffie-Hellman 金鑰交換協定 (3/4)

- 兩人 (Alice, Bob) 協商一把共用密鑰
  1. Alice 挑選一秘密值 x_a，計算 $y_a = g^{x_a} \pmod p$
  2. Bob 挑選一秘密值 x_b，計算 $y_b = g^{x_b} \pmod p$
  3. Alice 將 y_a 送給 Bob，Bob 將 y_b 送給 Alice
  4. Alice 得到 $k_{ab} = y_b^{x_a} \pmod p$
  5. Bob 得到 $k_{ba} = y_a^{x_b} \pmod p$
- $k_{ab} = y_b^{x_a} = g^{x_b x_a} = g^{x_a x_b} = y_a^{x_b} = k_{ba} \pmod p$
- 第三者得知 y_a 與 y_b 並無助於得到 k_{ab} (k_{ba})

Diffie-Hellman 金鑰交換協定 (4/4)

- 若有 N 位參與者，則需循環交換 $(N-1)$ 次資料始可得到一共用之密鑰
- Ex: 三人協商一共用密鑰 (2 Passes)



Pass 1

$U_1: y_{31} = y_3^{x_1} = g^{x_3 x_1}$

$U_2: y_{12} = y_1^{x_2} = g^{x_1 x_2}$

$U_3: y_{23} = y_2^{x_3} = g^{x_2 x_3}$

Pass 2

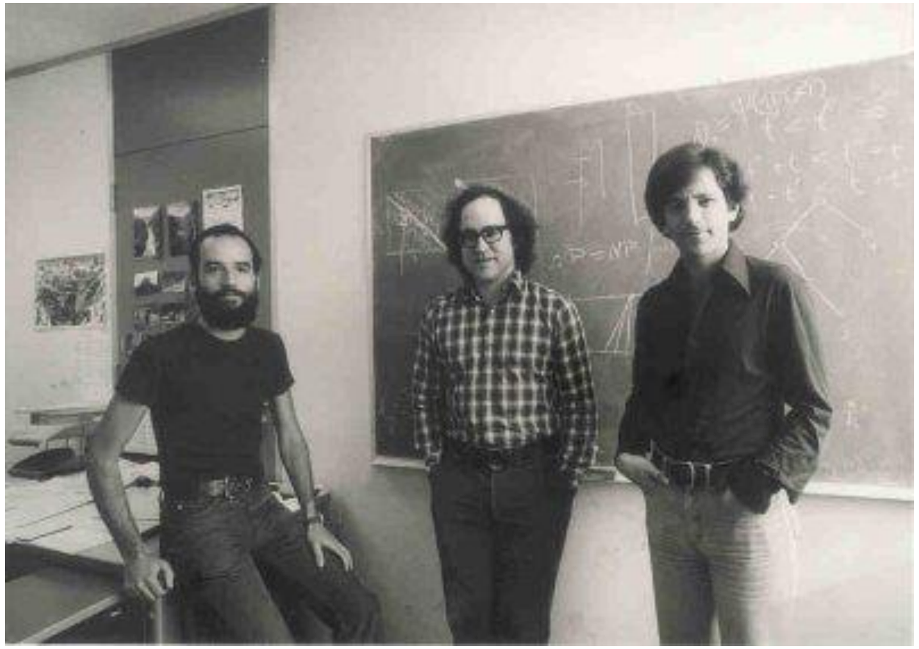
$U_1: k_{231} = y_{23}^{x_1} = g^{x_2 x_3 x_1}$

$U_2: k_{312} = y_{31}^{x_2} = g^{x_3 x_1 x_2}$

$U_3: k_{123} = y_{12}^{x_3} = g^{x_1 x_2 x_3}$

RSA 加密演算法 (1/2)

- RSA 是由 Ron Rivest、Adi Shamir 和 Leonard Adleman 於 1977 年在美國的麻省理工學院工作時所共同提出的，並由他們姓式的第一個字母命名
- 主要是基於質因數分解的難題，如果金鑰的長度足夠長，依靠目前的電腦還無法實際破解



Adi Shamir (left), Ron Rivest (middle), and Len Adleman (right)

RSA 加密演算法 (2/2)

- 取兩個質數 $p = 17, q = 11$
- 計算 $N = p \times q = 187$，$\psi(N) = (p-1)(q-1) = 160$
- 選定公鑰 $e = 7$，(e 必須與 $\psi(N)$ 互質)
- 計算出私鑰 d，滿足 $ed = 1 \, mod \, \psi(N)$，可得 $d = 23$
- 加密明文 M 得密文 $C = M^e \, mod \, N$

  $C = 88^7 \bmod 187 = 11$
- 解密密文 C 得明文 $M = C^d \, mod \, N$

  $M = 11^{23} \bmod 187 = 88$

安全基礎—質因數分解問題

- $\psi(N) = (p-1)(q-1)$，如果知悉公開參數 N 的兩個質因數 (p,q)，則 $\psi(N)$ 的值將無秘密可言。
- 由於e為公開訊息，並且滿足 $ed = 1\, mod\, \psi(N)$，因此如果 $\psi(N)$ 為外人知悉，則可利用歐式演算法計算出私鑰 d 的值。
- 如果可以有效率地執行因數分解，則 RSA 密碼系統毫無安全性可言。

質因數分解問題

- 15 =
- 91 =
- 2173 =
- 3776111 = ?

公開金鑰系統的延伸應用

- 數位簽章與憑證
- 盲簽章
- 模糊傳輸
- 零知識證明
- 同態加密

數位簽章與憑證 (RSA)

- Certificate Authority (CA) → Alice
  - $\{ed = 1 \bmod \psi(N)\}$
  - Public key (e, N); private key d
  - Certificate: algorithm, issuer, period, …
- Alice: $s = m^d \bmod N$.
- Alice sends (s, m) to Bob.
- Bob uses (e, N): $m = s^e \bmod N$.

盲簽章 (1/2)

- 由 David Chaum 在 1983 年首度提出，屬於一種數位簽章的方式。
- 訊息的內容在簽名之前對簽名者是不可見的。
- 所得到的盲簽章 (Blind Signature) 可以對原始的非盲訊息以常規數位簽章的方式公開驗證。
- 盲簽章可以有效地保護隱私，其中簽名者和訊息作者為不同人，例子包括電子投票和數位現金。

盲簽章 (2/2)

- Alice:
  - $ed \equiv 1 \pmod{\psi(N)}$
- Bob: message m, random number r
  - $m' = mr^e \bmod N$
- Alice:
  - $s' = (m')^d = m^d r \bmod N$
- Bob:
  - $s = s' r^{-1} \bmod N = \left(m^d r\right) r^{-1} \bmod N = m^d \bmod N$

模糊傳輸 (Oblivious Transfer)

- Michael Rabin 在 1981 年率先提出的一個概念，讓通訊雙方其中的一方先傳送一個位元給另一方，而接收方只有一半的機率能夠獲得傳送方的位元，另一半的機率則是什麼也得不到
- 這個 1-out-of-2 的模糊傳輸協定主要是建立在 RSA 與二次剩餘的基礎上，加上只有本身知道的隨機亂數值

Michael Rabin

M. O. Rabin, "How to exchange secrets with oblivious transfer," Technical Report TR-81, Aiken Computation Laboratory, Harvard University, 1981.

模糊傳輸：1-out-of-n

Request one message

n messages

Alice

Bob

1. Correctness
2. Privacy of Bob
3. Privacy of Alice

Stocks

Message query

M. Naor and B. Pinkas, “Oblivious transfer and polynomial evaluation,” In Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, pp. 245-254. 1999.

模糊傳輸：t-out-of-n

Alice

Request t messages

n messages

Bob

1. Correctness
2. Privacy of Bob
3. Privacy of Alice

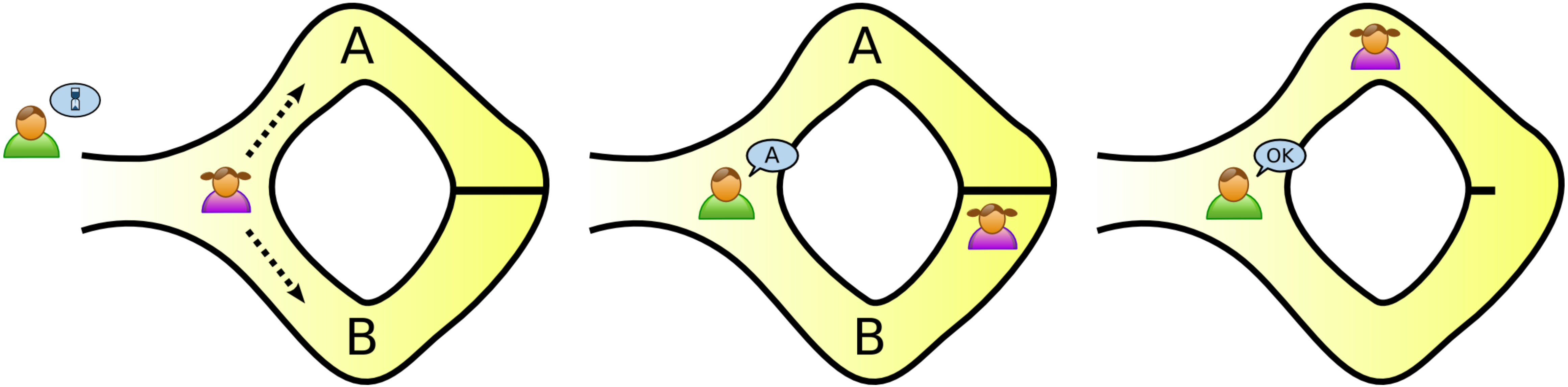
Stocks

Message query

O. Wakaha and S. Ryota, "k out of n Oblivious transfer without random oracles," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 87(1): 147-151, 2004.

零知識證明 (Zero-knowledge Proof)

- One party (the prover) can prove to another party (the verifier) that he/she possesses knowledge of certain information without revealing the actual content to the verifier.



J.-J. Quisquater, L. C. Guillou, and T. A. Berson, “How to explain zero-knowledge protocols to your children,” CRYPTO' 89 Proceedings, Lecture Notes in Computer Science, vol. 435, pp. 628–631, 1990.

同態加密 (Homomorphic Encryption)

- 最早是由美國密碼學家 Craig Gentry 於 2009 年所提出，允許明文在密文的狀態下進行某些運算，而不需要先解密成明文
- 有以下三種主要的類型
  1. 完全同態加密 (Fully Homomorphic Encryption)：允許對密文進行任何的加法與乘法運算，包括多次操作，並在最後解密得到結果
  2. 部分同態加密 (Partially Homomorphic Encryption)：只允許對密文進行某一類型的運算，通常是加法或者乘法，而不是兩者都支援
  3. 屬性同態加密 (Attribute-Based Homomorphic Encryption)：根據數據的屬性執行計算，而不是直接針對特定的密文進行操作
- 主要應用於雲端上的安全計算、隱私保護的資料分析與安全多方計算等領域

基於 RSA 的部分同態加密

- 基於 RSA 的架構，我們可以很輕易地實現同態加密的乘法運算
- 若 RSA 系統有模數 N 以及加密用的公鑰 e，則已知針對訊息 m 進行加密的結果為 $E(m) = m^e$。現有兩組明文 m_1 與 m_2，則其同態性質如下：

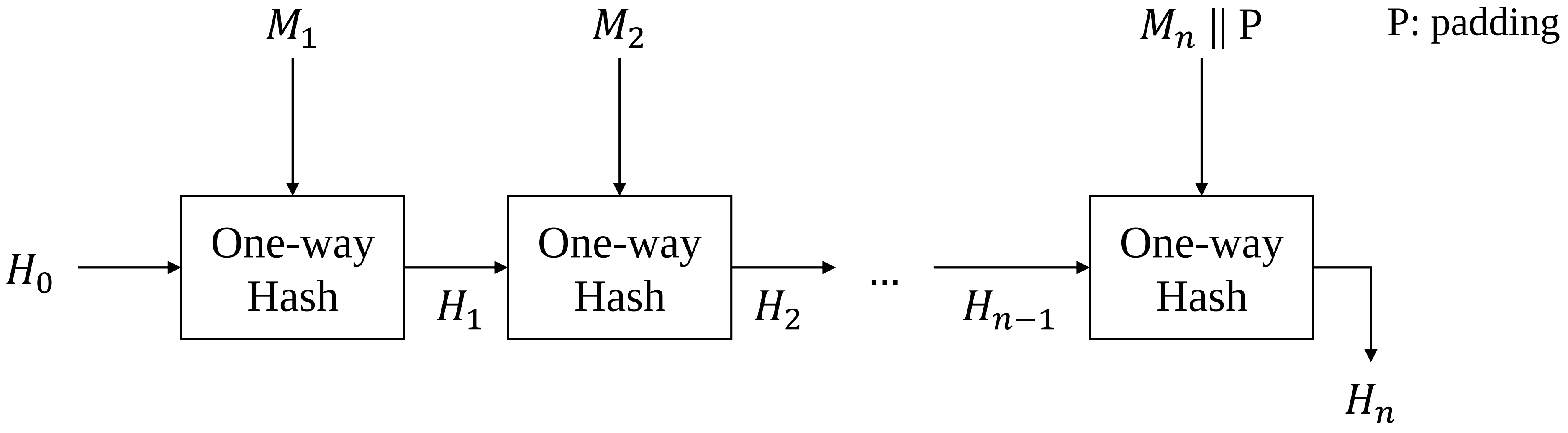
$$
\begin{aligned}
&E(m_1) \cdot E(m_2) \\
&= m_1^e m_2^e \bmod N \\
&= (m_1 \cdot m_2)^e \bmod N \\
&= E(m_1 \cdot m_2)
\end{aligned}
$$

單向雜湊函數 (1/3)

- 功能
  - 已給任意長度之輸入 x，可求得固定長度之輸出 $h(x)$。
- 用途
  - 配合數位簽章使用以加快數位簽章之速度及增加安全性。
- 常用之單向雜湊函數 (One-way Hash Function)
  - MD5、SHA-1、SHA-2、SHA-3 …

單向雜湊函數 (2/3)

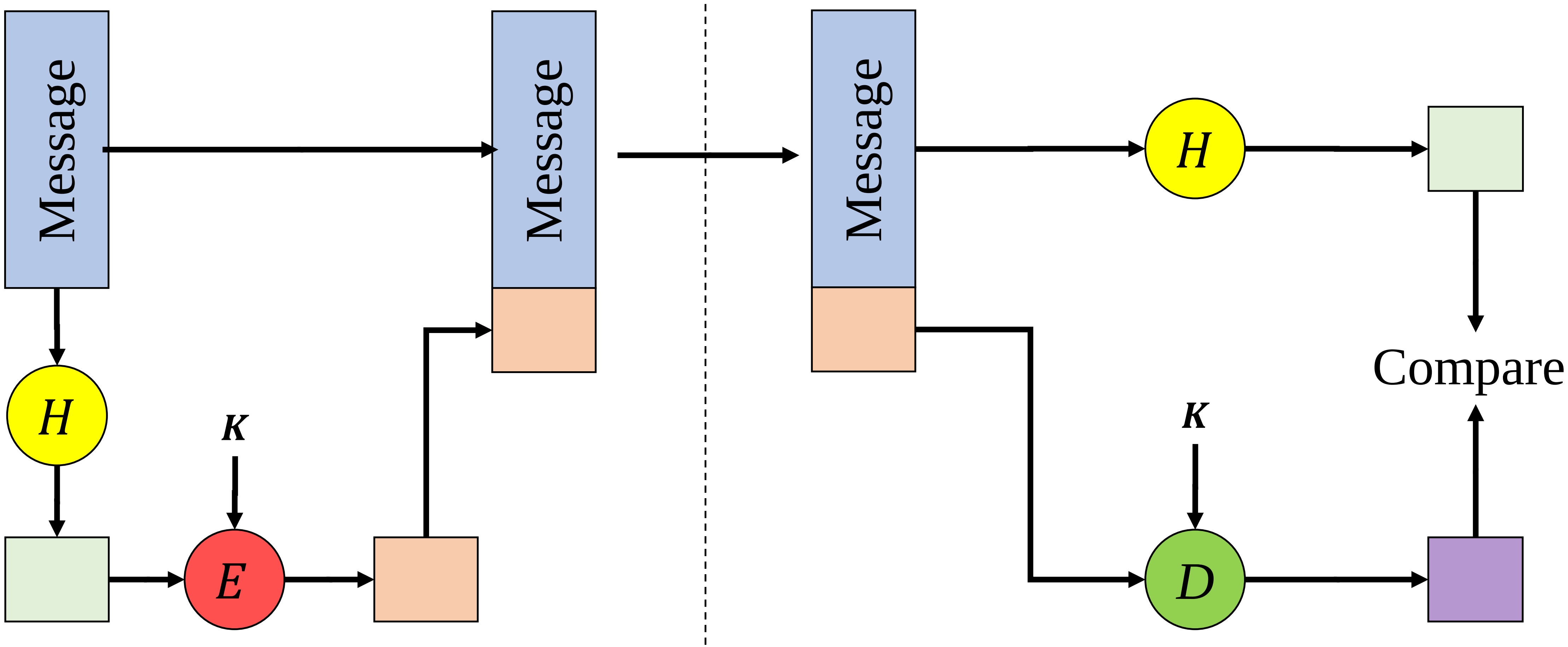
- One-way Hash Algorithm
  - Construction method called Merkle–Damgård
  - Used by algorithms like MD5, SHA-1, and SHA-2



單向雜湊函數 (3/3)

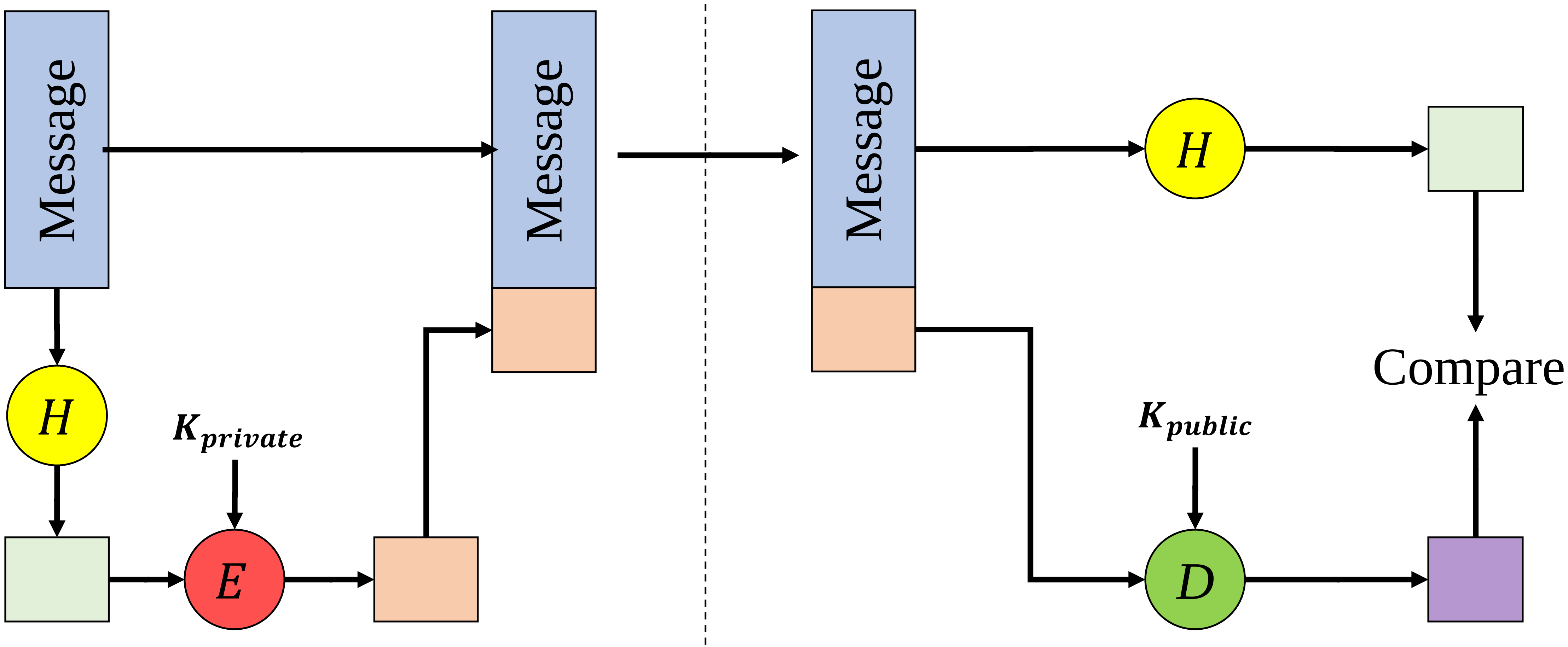
- A hash function H must have the following properties:
  - H can be applied to a block of data of any size.
  - H produces a fixed-length output.
  - $H(x)$ is relatively easy to compute for any given x, making both hardware and software implementations practical.
  - For any given code h, it is computationally infeasible to find x such that $h(x) = h$.
  - For any given block x, it is computationally infeasible to find $y \neq x$ with $h(y) = h(x)$.
  - It is computationally infeasible to find any pair (x, y) such that $h(x) = h(y)$.

Message Authentication Using a One-way Hash Function (1)



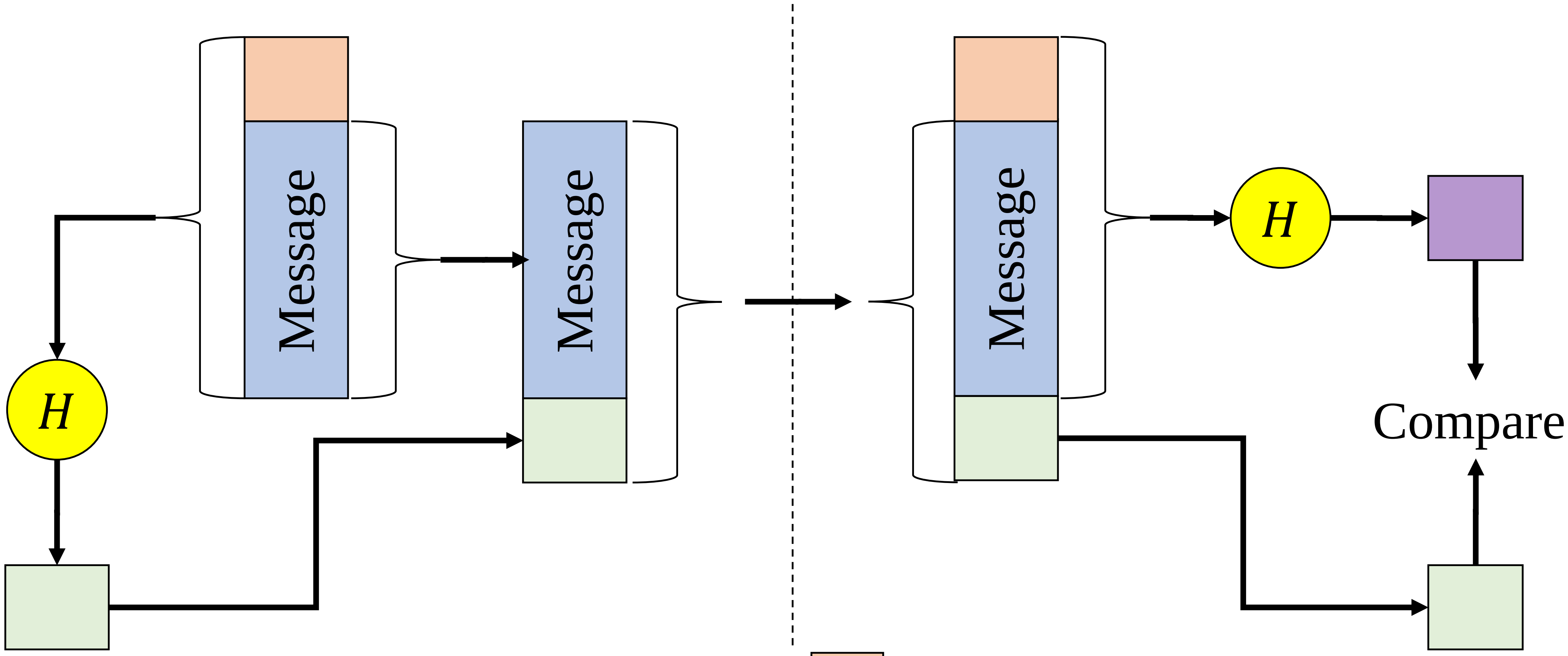
Using conventional encryption

Message Authentication Using a One-way Hash Function (2)



Using public-key encryption (Digital Signature)

Message Authentication Using a One-way Hash Function (3)



Using secret key (HMAC)

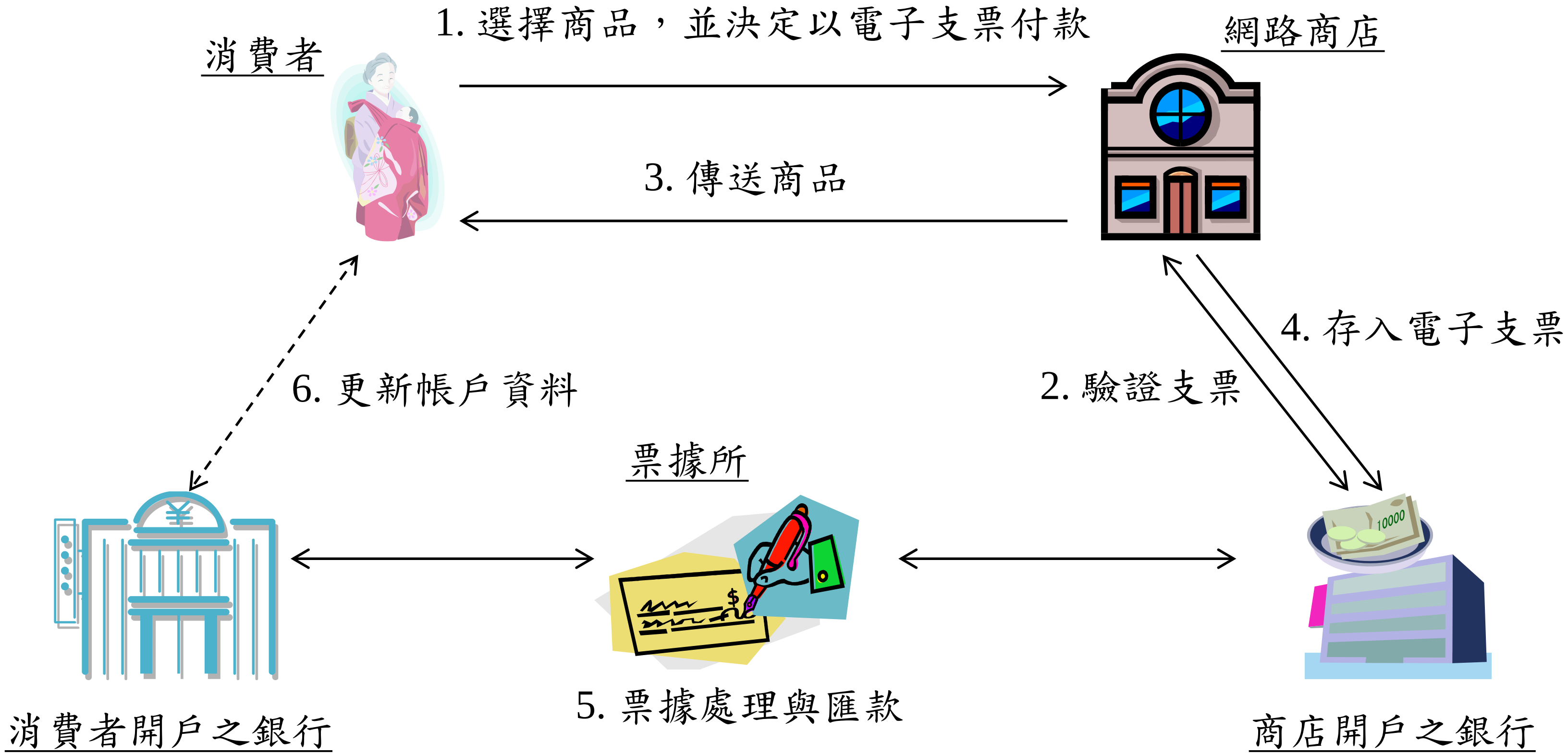
電子商務應用

- 電子支票
- 電子競標
- 電子投票

電子支票 (1/2)

- 先交易後付款
- 金額較大的交易
- 簽名效果
  - 公開金鑰系統的數位簽章
- 驗證付款者和銀行的身份
  - 數位憑證

電子支票 (2/2)

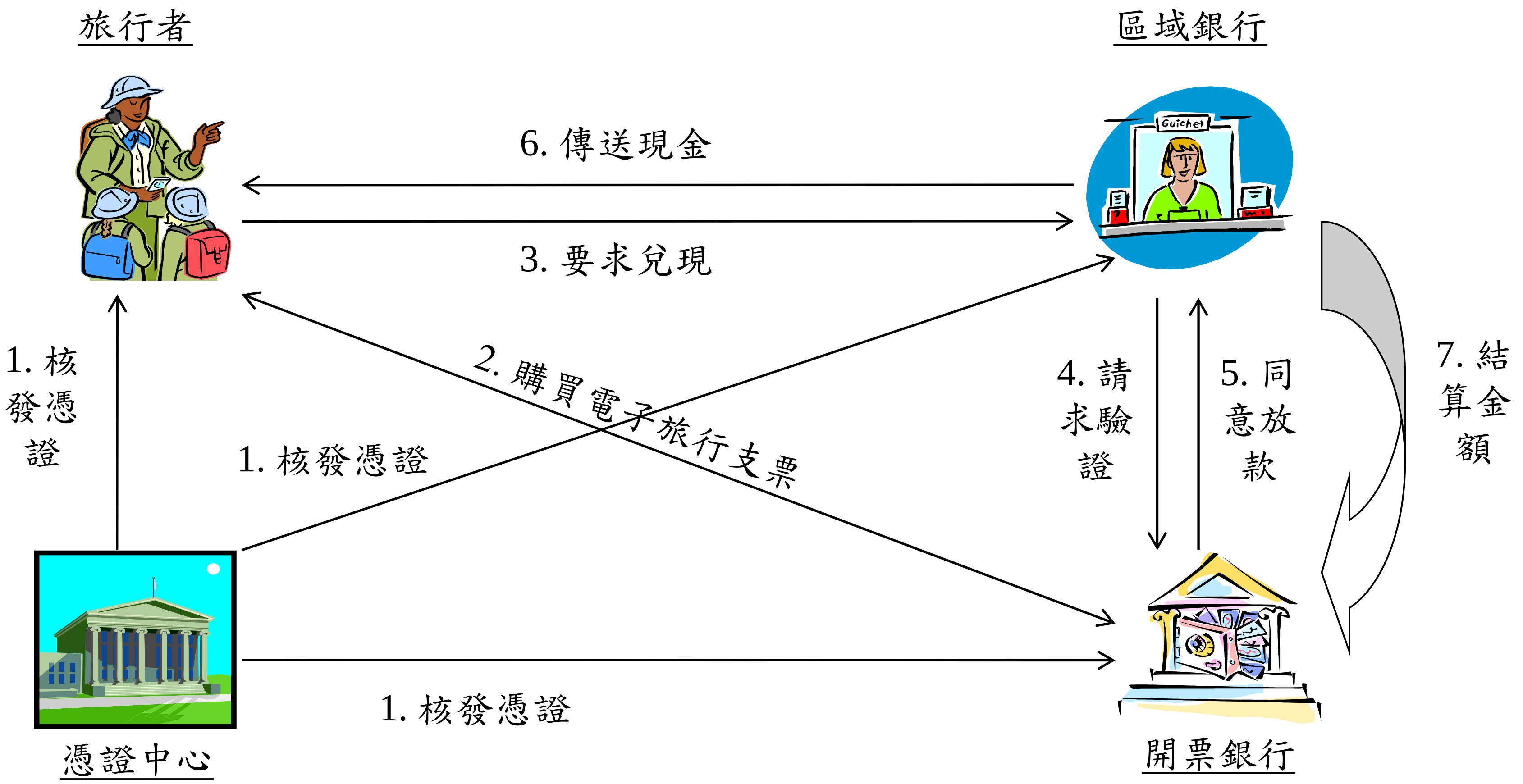


消費者
1. 選擇商品，並決定以電子支票付款
網路商店
3. 傳送商品
4. 存入電子支票
2. 驗證支票
6. 更新帳戶資料
票據所
5. 票據處理與匯款
消費者開戶之銀行
商店開戶之銀行

電子旅行支票 (1/2)

- 傳統旅行支票
  - 個人使用
  - 護照
  - 簽章
  - 仿冒護照 ➔ 盜領
- 電子
  - 智慧卡 (Smart card)
  - 生物辨識：聲紋、指紋
  - 先付款
  - 公開金鑰系統的數位簽章

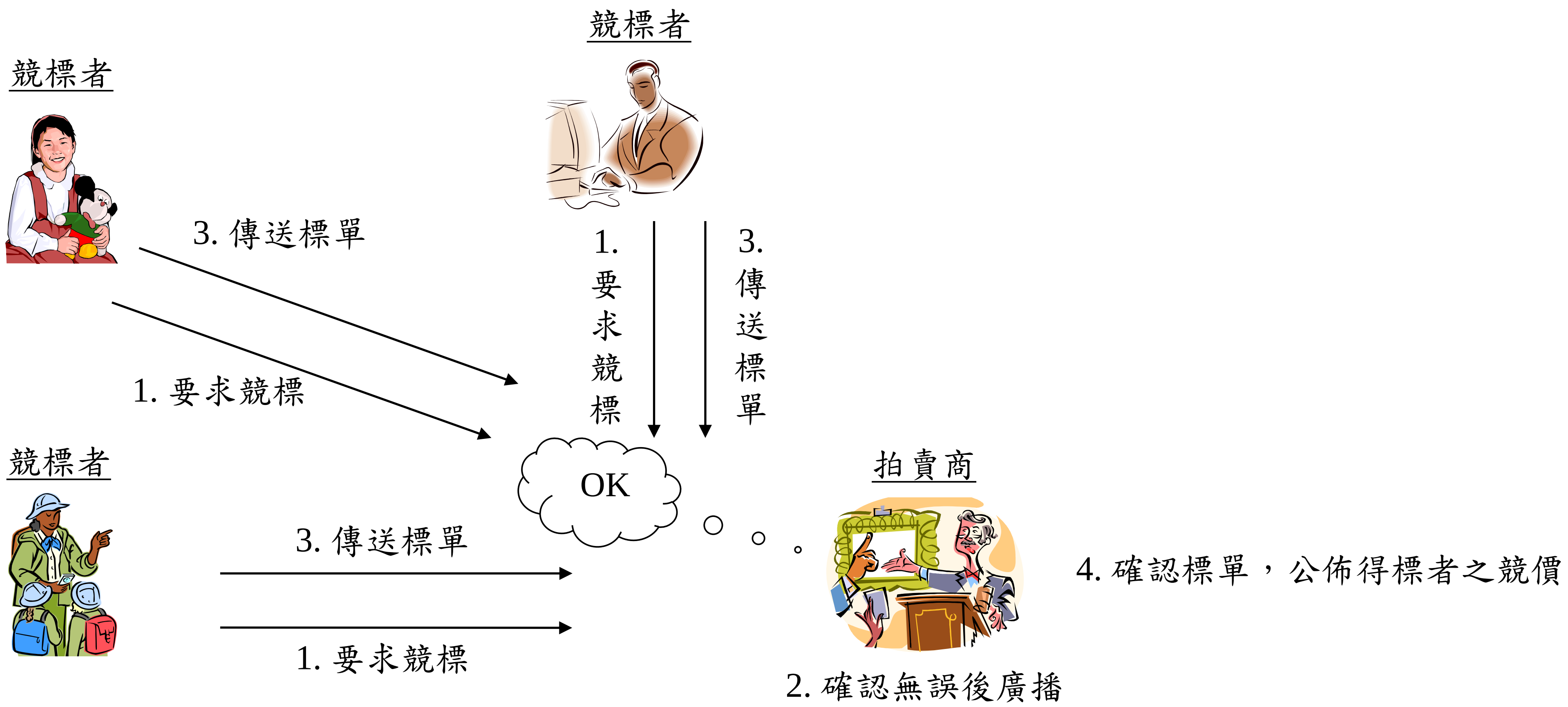
電子旅行支票 (2/2)


旅行者
區域銀行
Guichet
6. 傳送現金
3. 要求兌現
1. 核發憑證
2. 購買電子旅行支票
1. 核發憑證
4. 請求驗證
5. 同意放款
7. 結算金額
1. 核發憑證
憑證中心
開票銀行

電子競標 (1/2)

- 英式拍賣 (English Auction)：eBay, Amazon Auctions, …
  - 競價由低至高、依次遞增，當到達拍賣截止時間，出價最高者成為競買的贏家。拍賣前，賣家可設定保留價，當最高競價低於保留價時，賣家有權不出售此拍賣品
- 荷式拍賣 (Dutch Auction)
  - 拍賣人先將價格設定在足以阻止所有競拍者的水平，然後由高價往低價喊，第一個應價的競拍者獲勝，並支付當時所喊到的價格
- 封閉式拍賣 (Sealed-bid Auction)
  - 數位簽章技術
  - 通訊金鑰

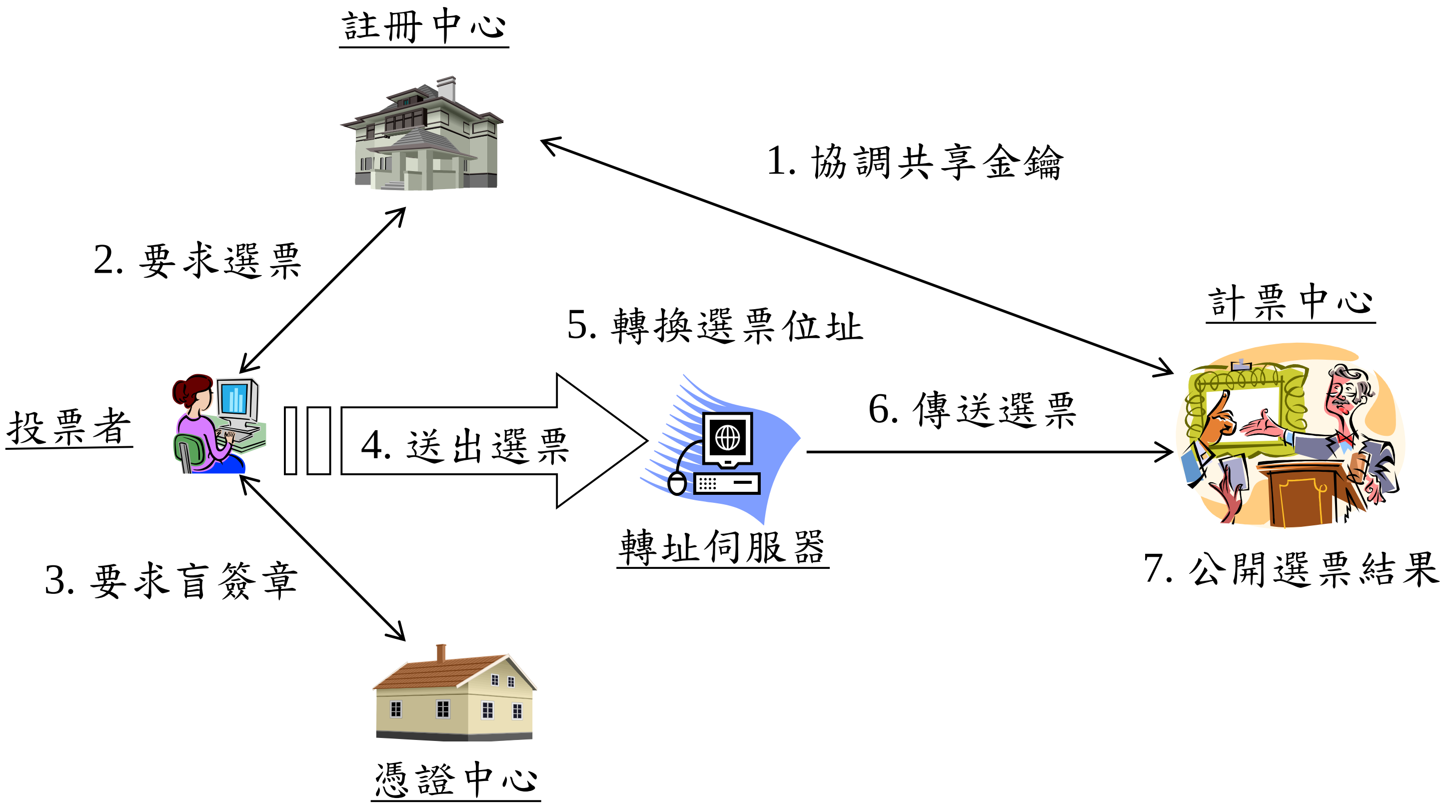
電子競標 (2/2)


競標者
競標者
競標者
拍賣商
1. 要求競標
3. 傳送標單
1. 要求競標
3. 傳送標單
1. 要求競標
3. 傳送標單
OK
2. 確認無誤後廣播
4. 確認標單，公佈得標者之競價

電子投票 (1/2)

- 傳統
  - 時間限制
  - 地理限制
  - 開票慢
- 電子
  - 任何時間
  - 任何地點
  - 高效率
- 合法性---公開金鑰技術
- 正確性---AES
- 匿名性---盲簽章

電子投票 (2/2)



註冊中心
1. 協調共享金鑰
2. 要求選票
5. 轉換選票位址
計票中心
投票者
4. 送出選票
6. 傳送選票
轉址伺服器
7. 公開選票結果
3. 要求盲簽章
憑證中心

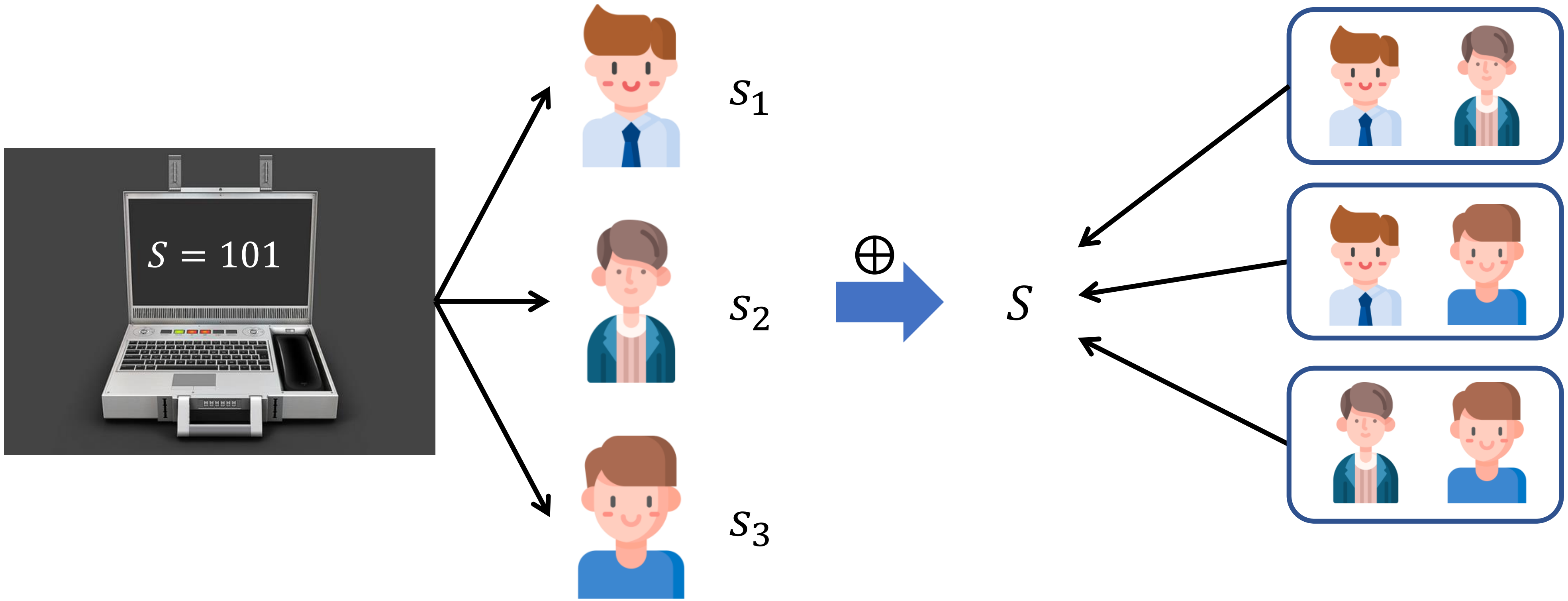
其他分支研究領域

- 秘密分享
- 視覺密碼學
- 資訊偽裝學

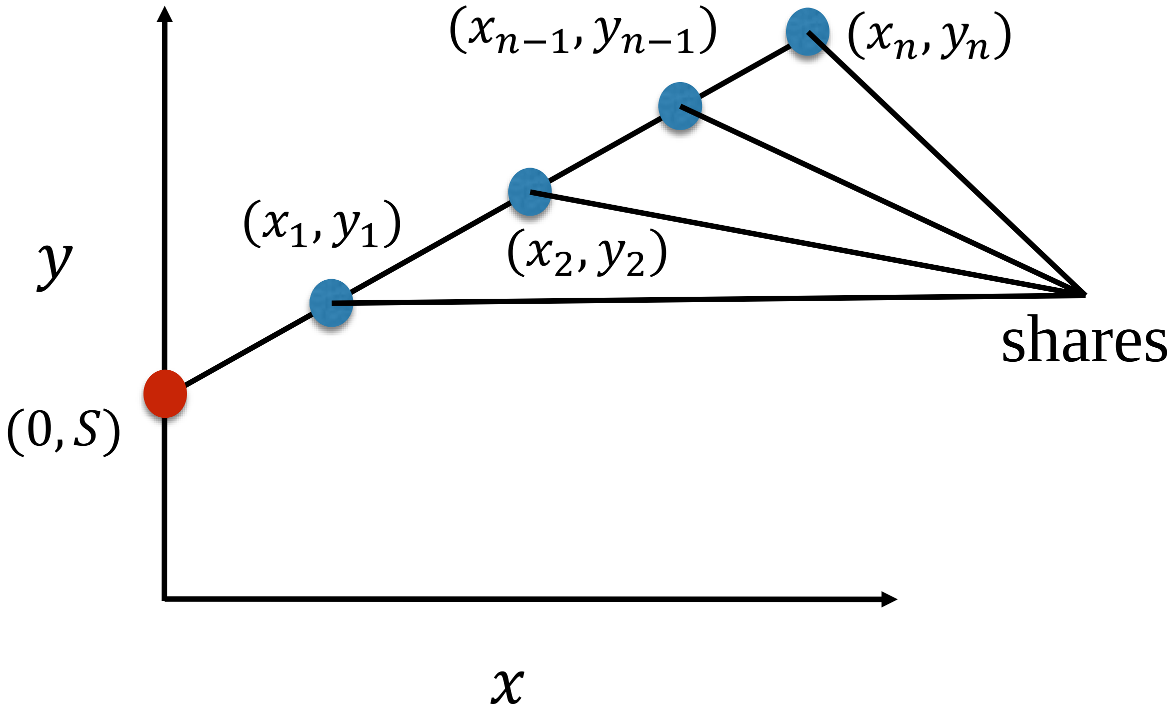
秘密分享 (Secret Sharing)

- Given a secret S and n agents
  - Any t or more agents can recover S
  - Less than t players have no information about S

(2,3) Secret Sharing



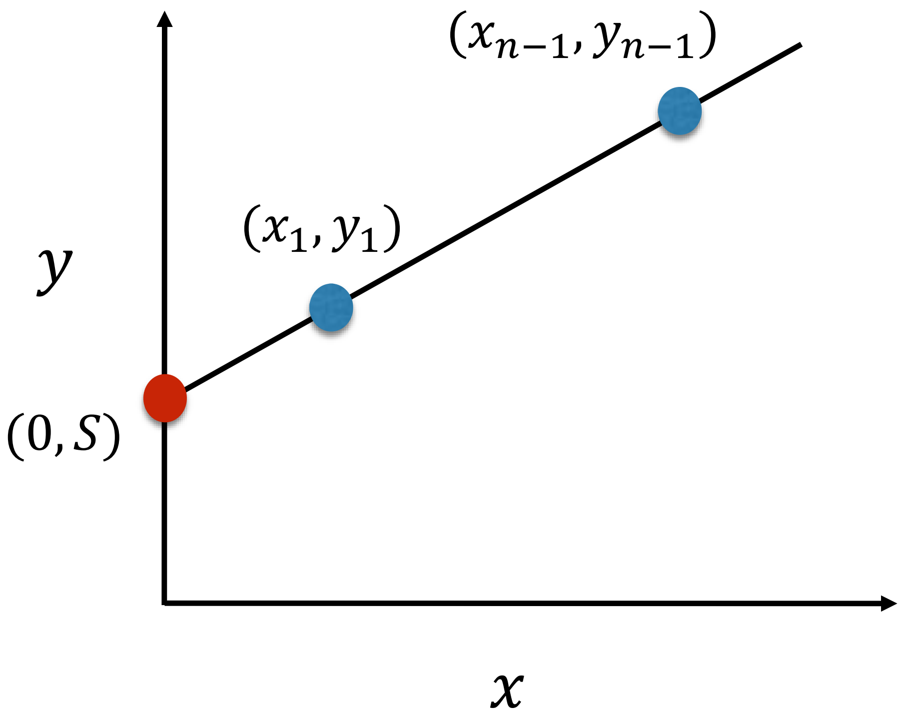
(t, n) 秘密分享



Secret S is y intercept

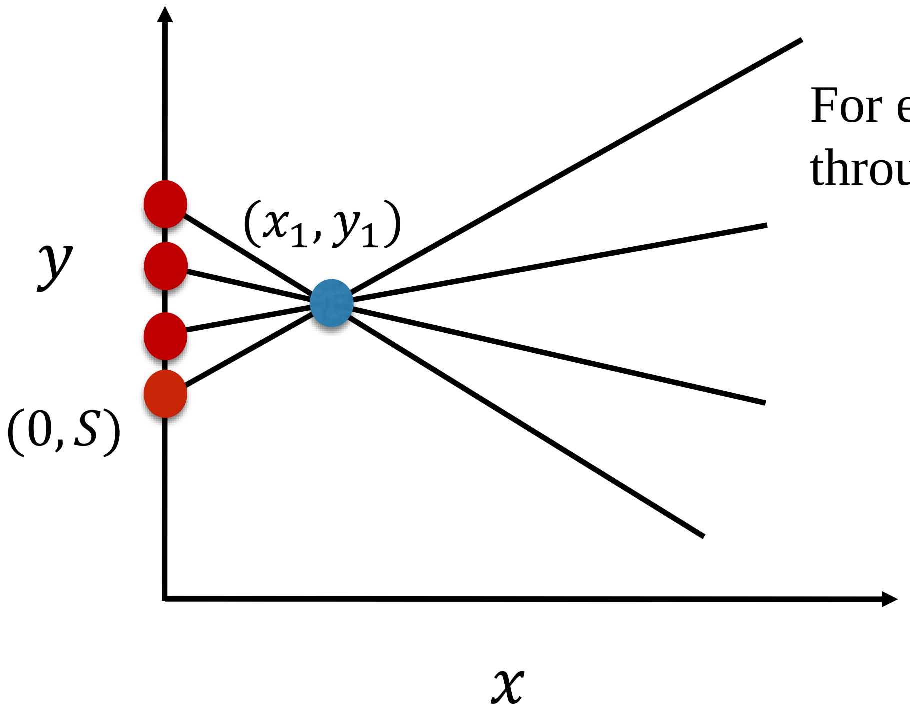
$(2, n)$ 秘密分享

One share does not suffice



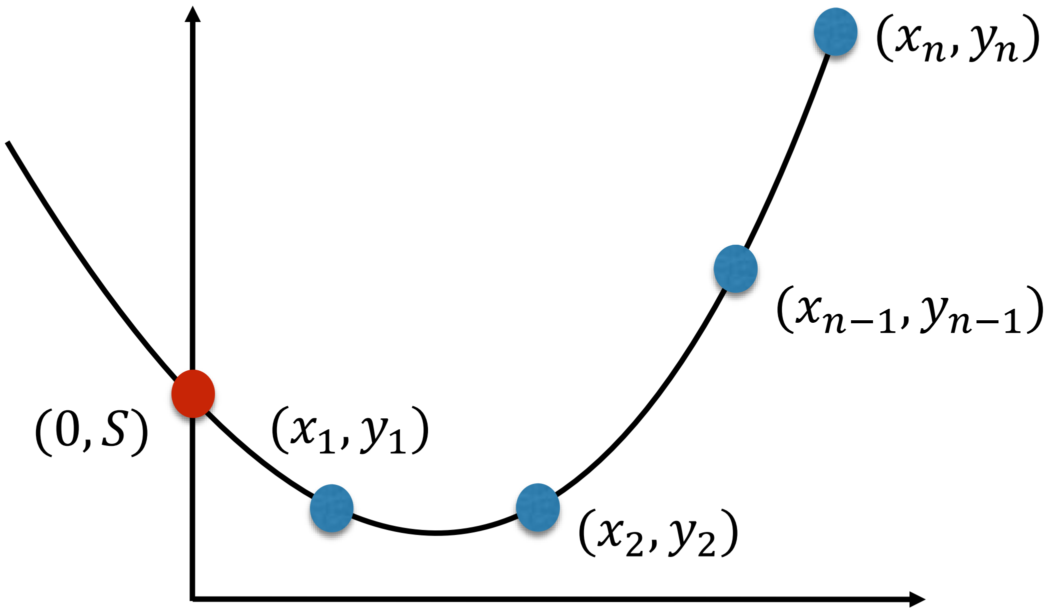
$(2, n)$ 秘密分享

One share does not suffice



For every secret S, there is a line through x_1, y_1

$(3, n)$ 秘密分享

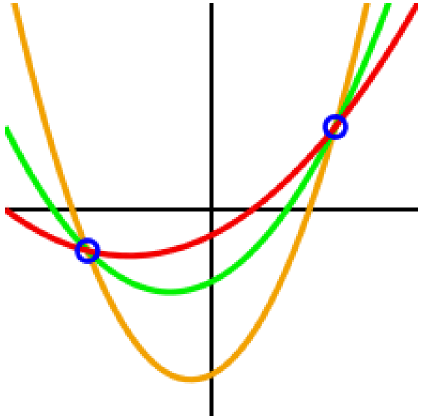


Three points determine a quadratic polynomial

Shamir 秘密分享

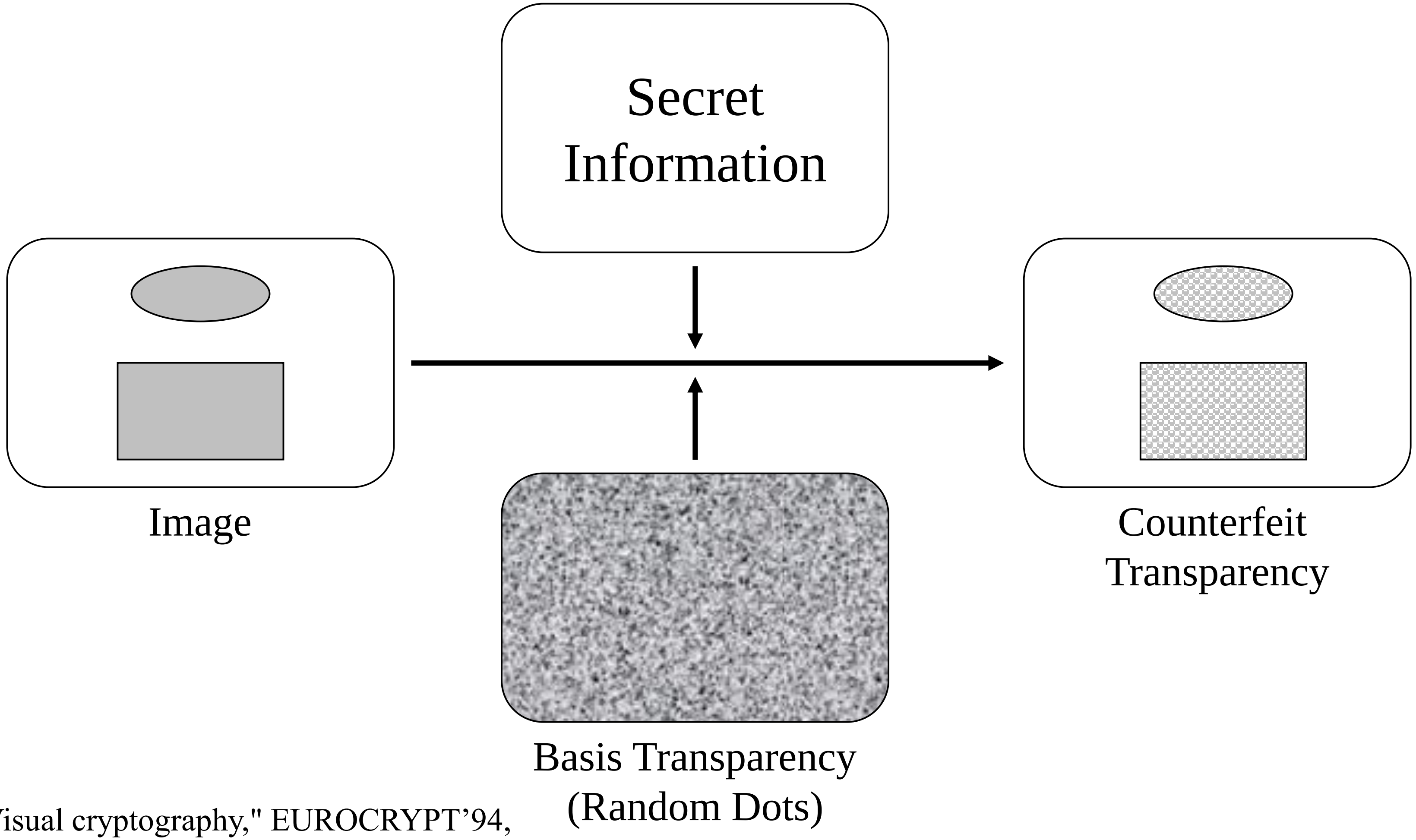
- It takes t points to define a polynomial of degree $t-1$
  - Create a degree-$(t-1)$ polynomial with secret as the constant coefficient and the remaining coefficients chosen at random
  - Find n points on the curve (not at $x = 0$) and give one to each participant
  - At least t points are required to fit the polynomial and hence to recover secret (and any t points will suffice)

$$\boxed{y = a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + \cdots + a_1 x + a_0}$$



A. Shamir, "How to share a secret," *Communications of the ACM*, 22(11): 612-613, 1979.

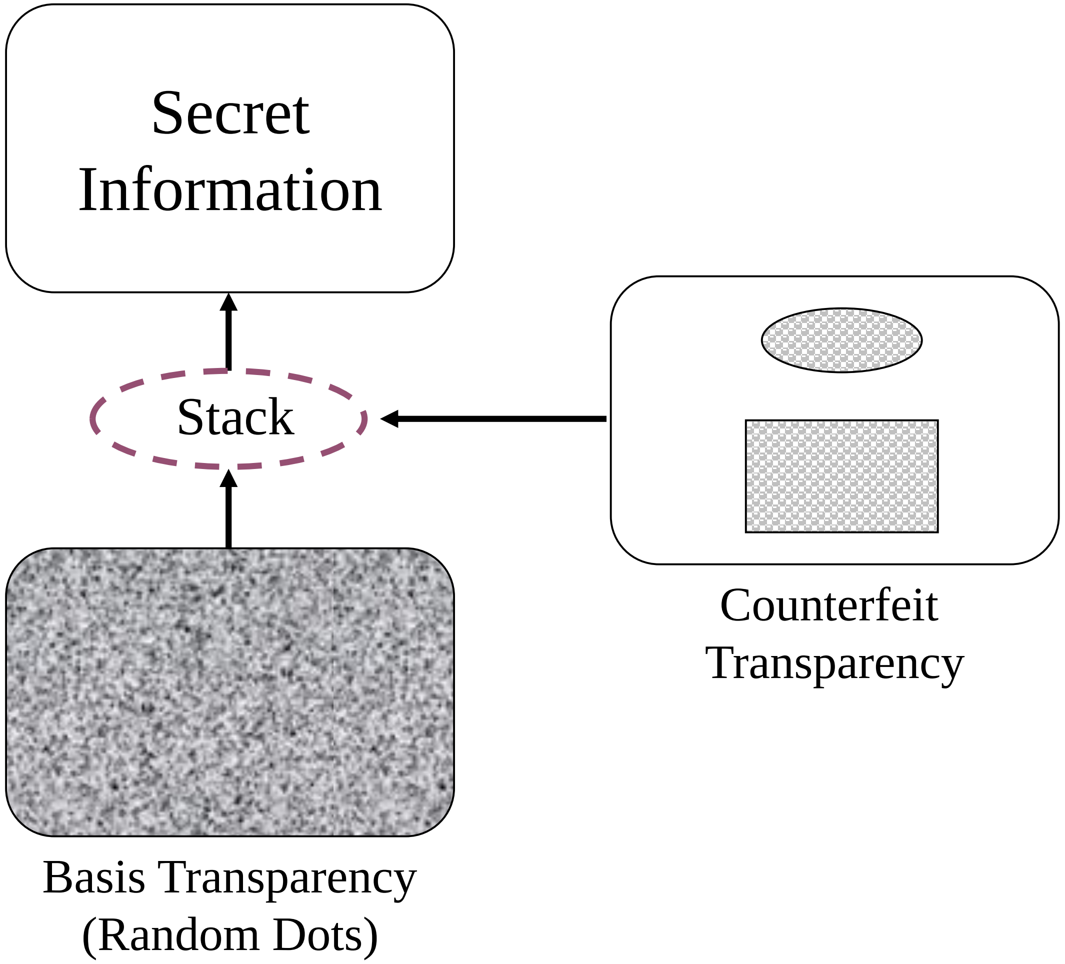
視覺密碼學 (1/2)



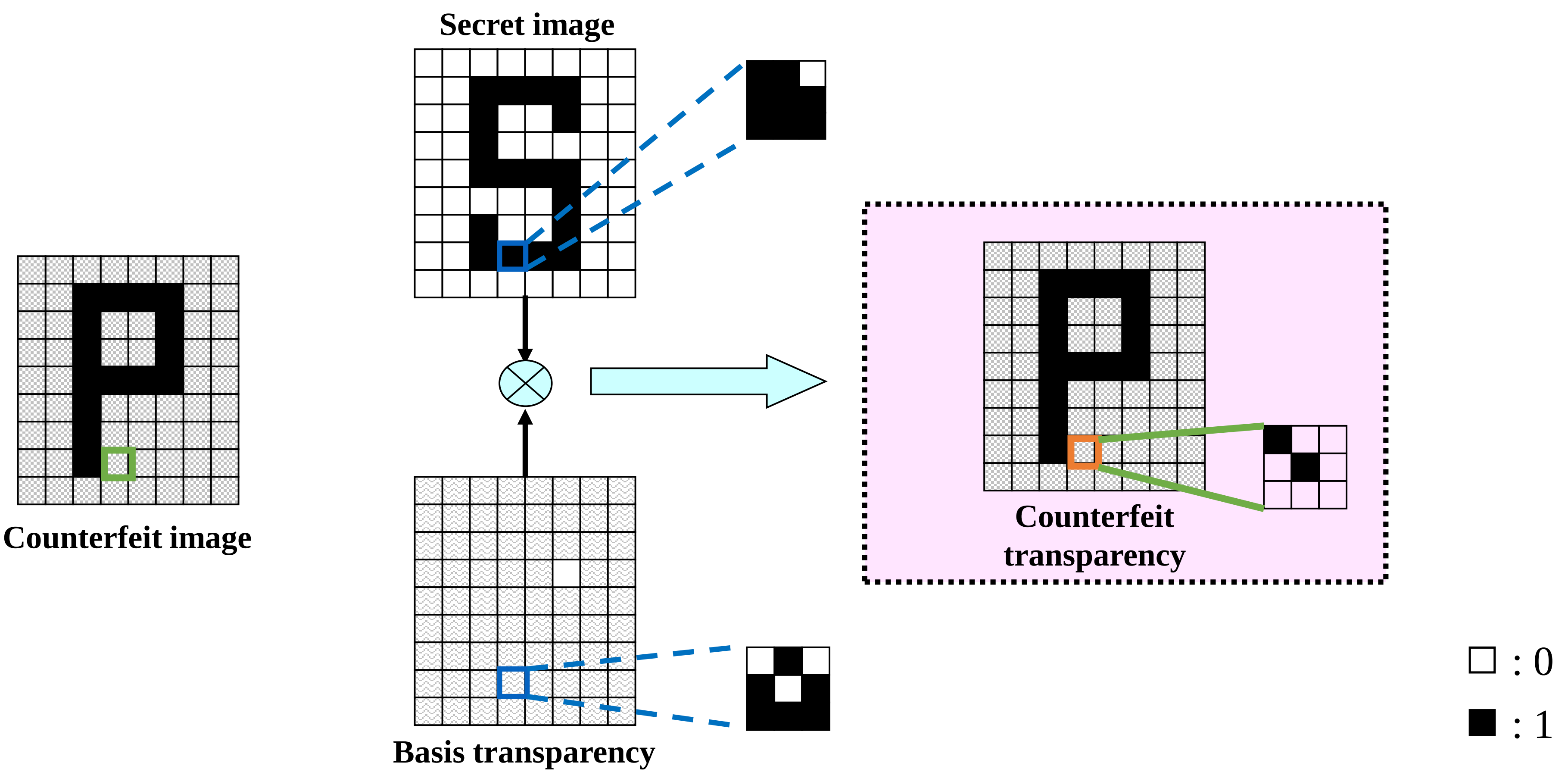
M. Naor and A. Shamir, "Visual cryptography," EUROCRYPT’94, Perugia, Italy, May 9–12, 1994.

視覺密碼學 (2/2)

像素疊在一起是作 OR 的運算

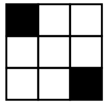
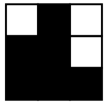
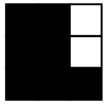
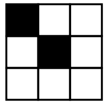
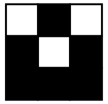
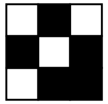
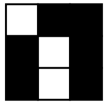
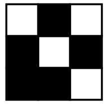
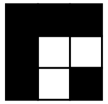


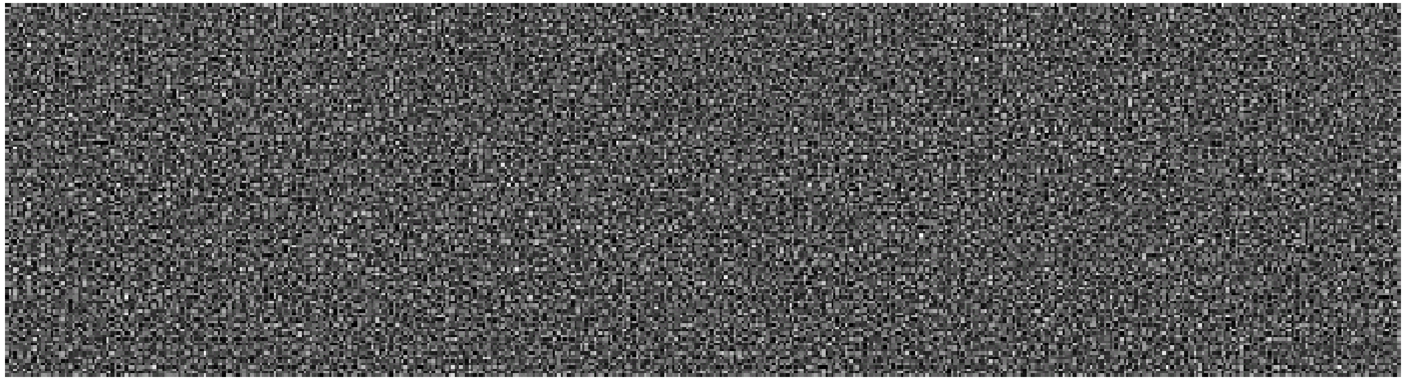
3×3 像素擴充法 (1/2)



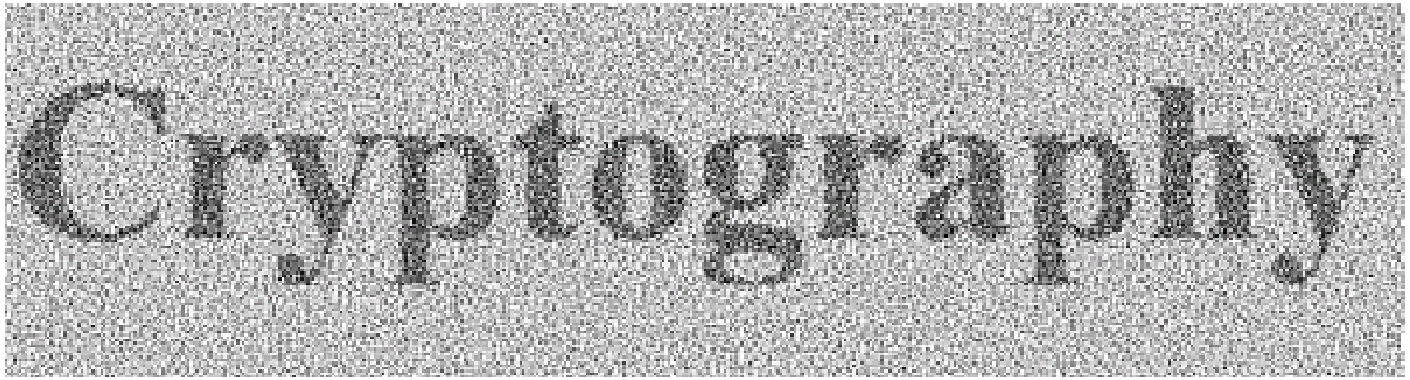
Secret image
Counterfeit image
Basis transparency
Counterfeit transparency
□ : 0
■ : 1

3×3 像素擴充法 (2/2)

attributes / cases	Counterfeit image	Counterfeit transparency	Basis transparency	Secret image	Counterfeit transparency +Basis transparency
Case 1	White	白 		White	白 
Case 2	White	白 		Black	黑 
Case 3	Black	黑 		White	白 
Case 4	Black	黑 		Black	黑 



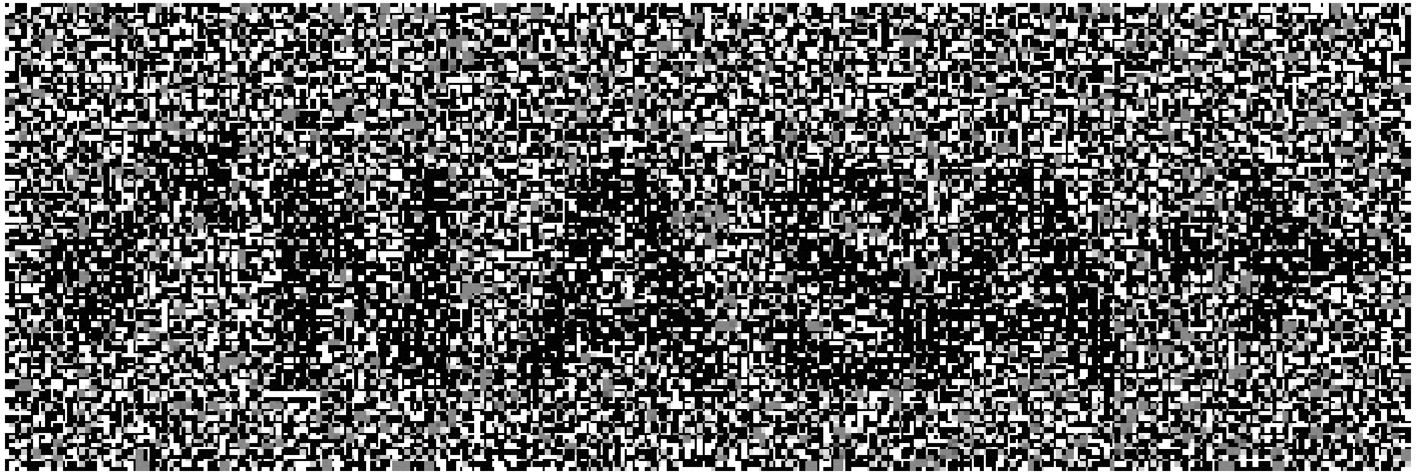
Basis Transparency



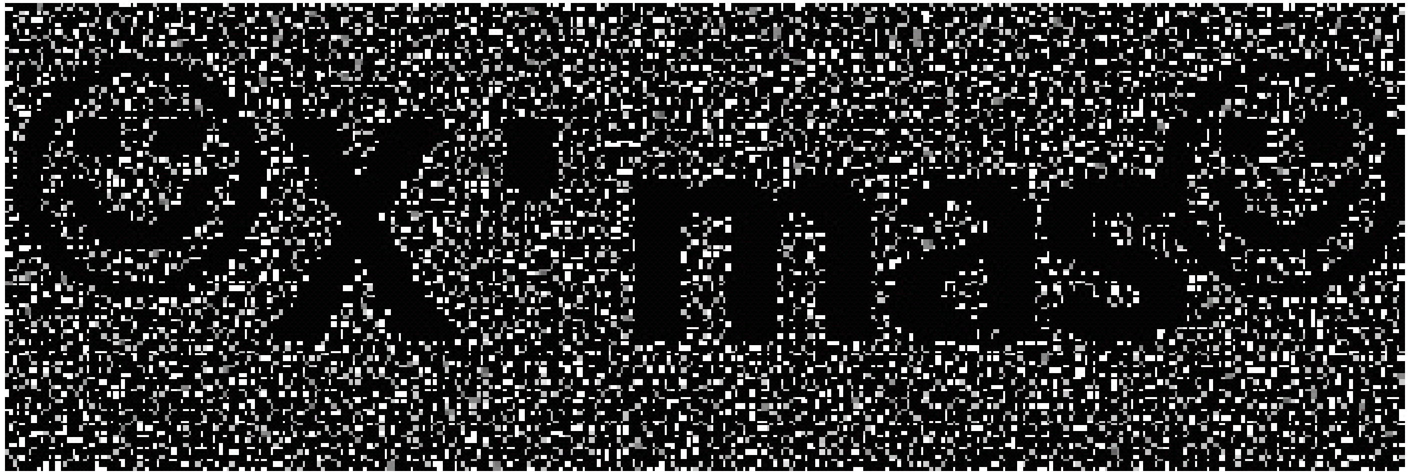
Counterfeit Transparency 1

3×3 像素擴充法 (有意義)

Original secret pixel	White				Black			
Counterfeit 1	白	白	黑	黑	白	白	黑	黑
Counterfeit 2	白	黑	白	黑	白	黑	白	黑
Stacked pixel	白	白	白	白	黑	黑	黑	黑



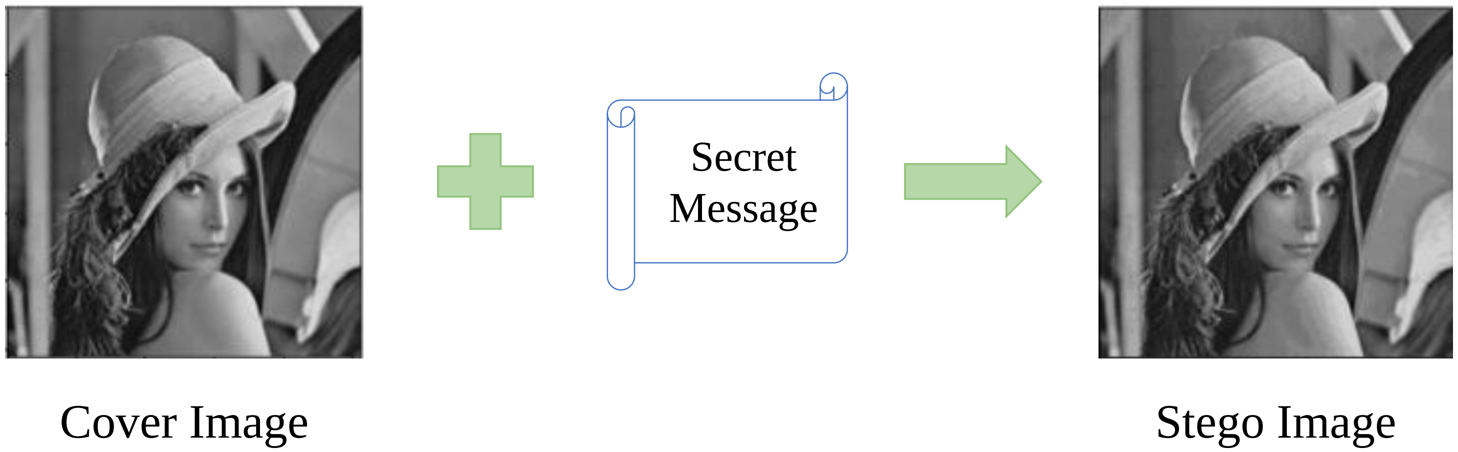
Counterfeit 1



Counterfeit 2

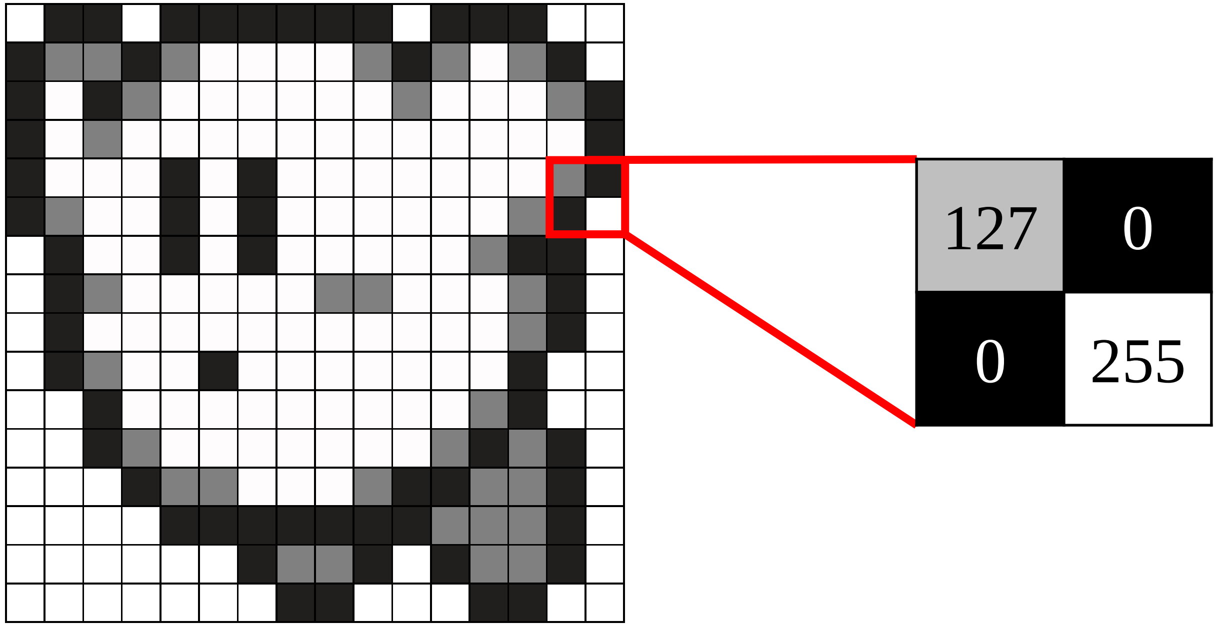
資訊偽裝學

- Steganography: the study of concealing the secret message with meaningful multimedia content such as image, video, audio, and text



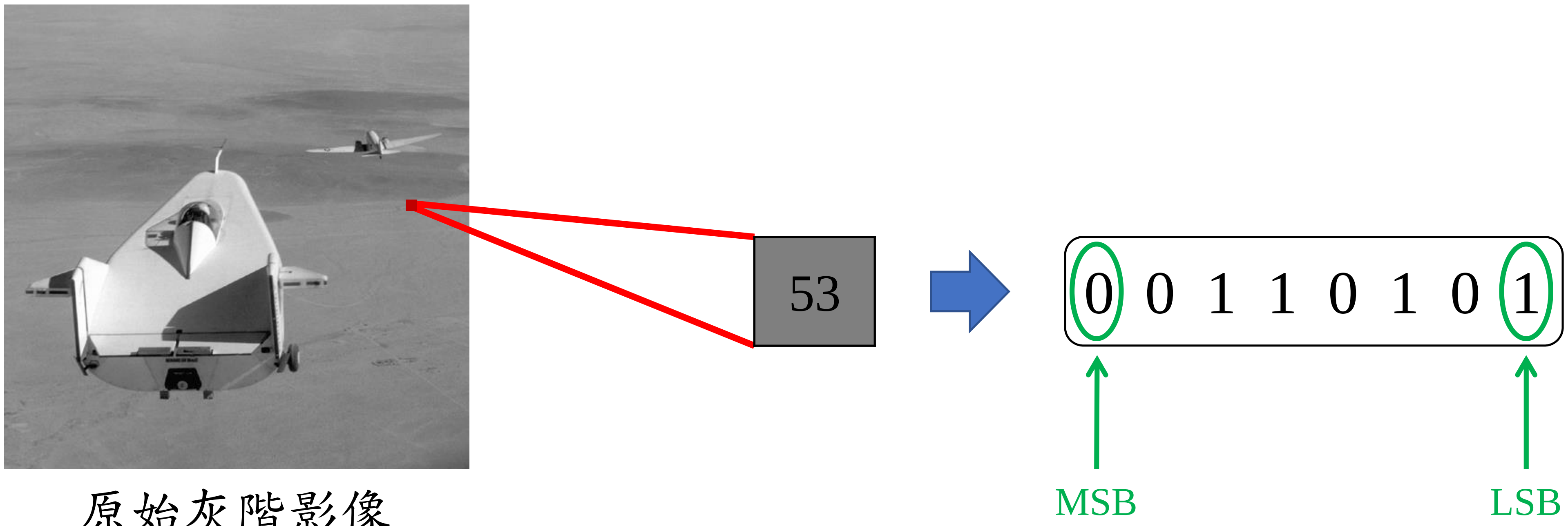
空間域資訊隱藏

- 空間域的研究是以數位影像為載體來進行資訊偽裝
- 其中又以灰階影像 (Grayscale Image) 較為常見，由數值 0 ~ 255 的像素所組成



LSB 資訊嵌入法 (1/2)

- LSB (Least Significant Bit)：最低有效位元
- MSB (Most Significant Bit)：最高有效位元



原始灰階影像

LSB 資訊嵌入法 (2/2)

- 由左至右，由上至下的方式依序嵌入訊息至每一個像素值的最低位元

1 0 0 1 1 0 0 0 1 1 …

Secret Message

Cover Image

20	31	25	40
34	51	39	42
37	55	53	60

Stego Image

21	30	24	41
35	50	38	42
37	55	52	60

嵌入在第一個 LSB

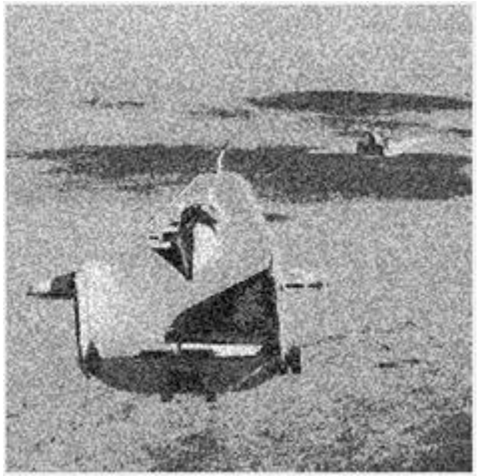
嵌入在第二個 LSB

嵌入在第三個 LSB

嵌入在第四個 LSB

嵌入在第五個 LSB

嵌入在第六個 LSB



嵌入在第七個 LSB

嵌入在第八個 LSB

OPAP 資訊嵌入法 (1/2)

- LSB 資訊嵌入法是否已經是最低誤差？
  - 假設我要嵌入二進制訊息 111 在像素值 153

151	10010111
152	10011000
153	10011001
154	10011010
…	
159	10011111

誤差 6

151	10010111
152	10011000
153	10011001
154	10011010
…	
159	10011111

誤差 2

OPAP 資訊嵌入法 (2/2)

- Optimal Pixel Adjustment Process (OPAP)：透過調整嵌入後像素的其他高階位元值來降低嵌入後的誤差值
  - p：原始像素值
  - p'：利用 LSB 嵌入 n 個位元後的像素值
  - p^*：利用 OPAP 嵌入 n 個位元後的像素值

$$p^* \begin{cases} p + 2^n & \text{if } p^n - p' > 2^{n-1} \text{ and } p + 2^n \leq 255 \\ p - 2^n & \text{if } p^n - p' < -2^{n-1} \text{ and } p - 2^n \geq 0 \\ p & otherwise \end{cases}$$

C.-K. Chan and L.-M. Cheng, *Pattern Recognition* 37(3): 469-474, 2004.

Conclusions

- Goal: Secrecy and Authenticity
- No “Perfect” Security
- Cracked = Insecure？