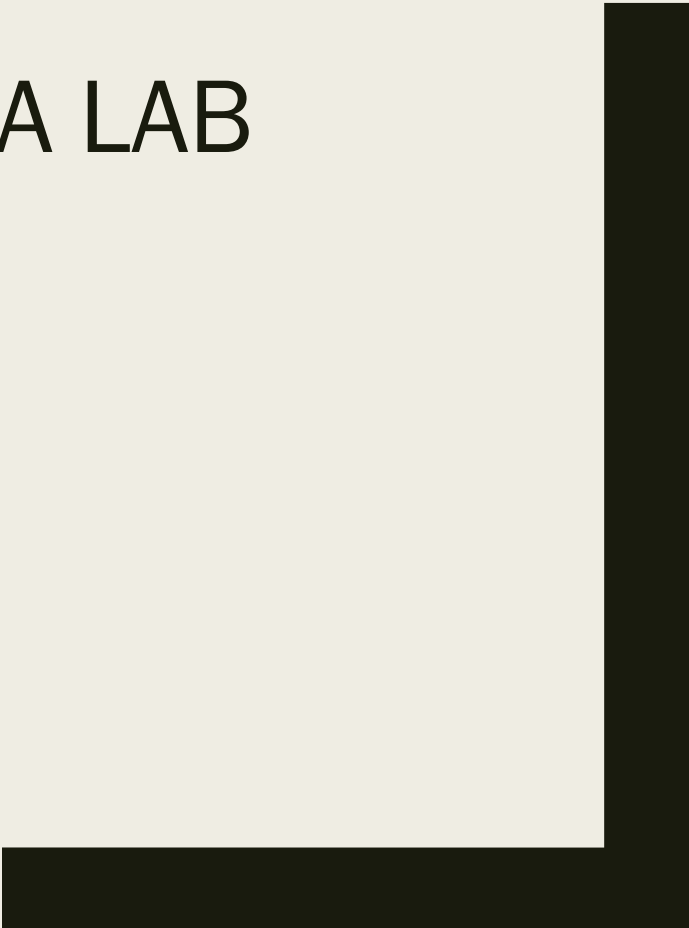




NETWORK & MULTIMEDIA LAB

CRYPTOGRAPHY

Fall 2020



何謂密碼學

Ronald Rivest（發明RC4、MD5，RSA共同發明者）解釋道：
「密碼學是關於如何在敵人存在的環境中通訊」。

- 古典密碼學

- 主要關注資訊的**保密書寫**和**傳遞**，以及與其相對應的破譯方法。

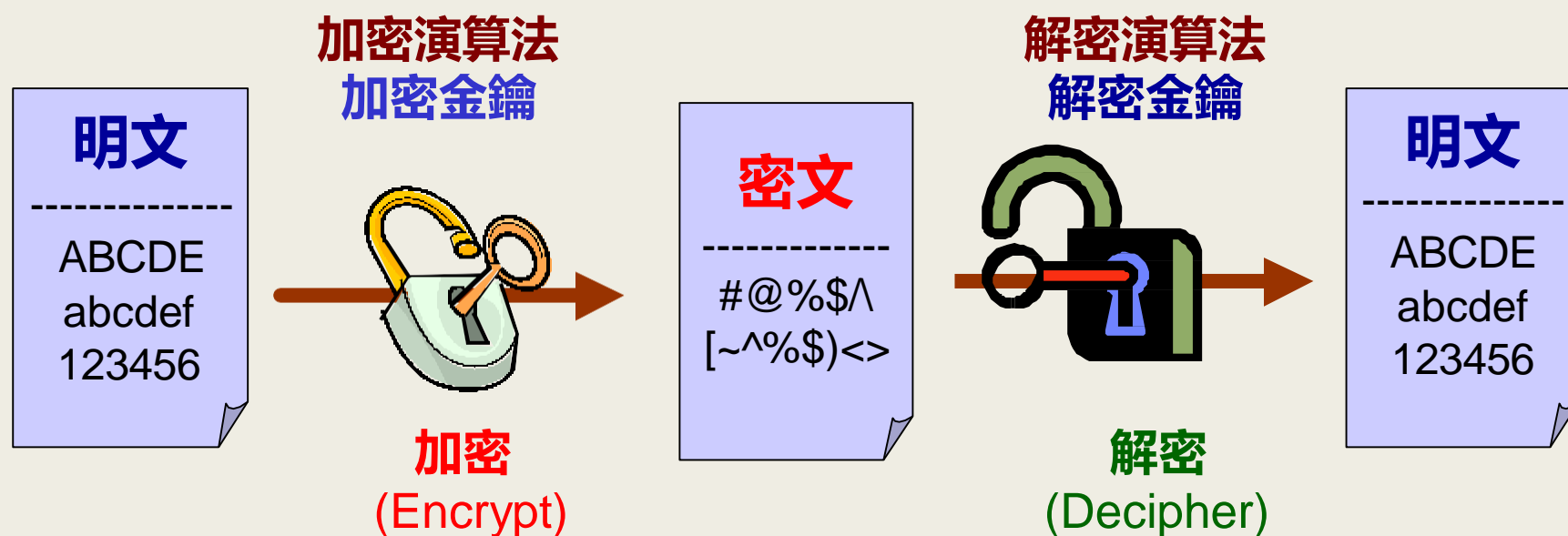
- 現代密碼學

- 不只關注資訊保密問題，還同時涉及資訊**完整性**驗證（訊息驗證碼）、資訊發布的**不可否認性**（數位簽章）、以及在分散式計算中產生的來源於外部和內部的攻擊的所有資訊安全問題（拜占庭將軍問題）。

何謂密碼學

簡單來說，密碼學為一種利用數學方法來對資料加密和解密的科學。

密碼系統包含：明文、密文、加解密演算法、加解密金鑰



密碼學基本名詞

- 明文 (Plaintext)
 - 加密前的原始資料，為加密演算法的輸入，解密演算法的輸出。
- 密文 (Ciphertext)
 - 加密之後的資料，為加密演算法的輸出，解密演算法的輸入。
- 加密演算法 (Encryption Algorithm)
 - 利用密鑰對明文進行加密的編碼動作的演算法。
- 解密演算法 (Decryption Algorithm)
 - 利用金鑰對密文進行解密的解碼動作的演算法。
- 密碼破解 (Cryptanalysis)
 - 不需經由解密金鑰或使用偽造金鑰，將密文還原成明文。

為什麼需要密碼學

- 訊息隱密性（ Confidentiality ）：指訊息不會遭截取、窺竊而洩漏資料內容致損害其秘密性。
- 訊息完整性（ Integrity ）：指訊息內容不會遭篡改而造成資料不正確，即訊息如遭篡改時，該筆訊息無效。
- 訊息來源辨識性（ Authentication ）：指傳送方無法冒名傳送資料。
- 訊息不可重複性（ Non-duplication ）：指訊息內容不得重複。
 - 防止重放攻擊 (Replay attack)
- 訊息不可否認性（ Non-repudiation ）：指無法否認其傳送或接收訊息行為。
 - A 傳訊息給 B，之後就不能否認曾經傳過此訊息

加密技術的強度

- 加密技術的強度指的是密碼破解所需要花費的時間與資源。
- 加密技術強度的高低通常牽涉到下列的因素：
 - 演算法強度
 - 金鑰的長度
 - 金鑰保護機制
- 柯克霍夫原則（ Kerckhoffs's principle ）
 - 密碼系統的安全性不在演算法的保密，而是取決於金鑰的保密。
 - 即使密碼系統的任何細節已為人悉知，只要密鑰（ key ）未洩漏，它也應是安全的。

金鑰 (Key)

金鑰是一組相當長度的數字或符號字串，其大小通常以位元(bit)為單位。

金鑰通常是演算法則內的一個變數，所以不同的金鑰會產生不一樣的密文。

就密碼學而言金鑰長度越長，密文就越不容易被破解。

密碼系統的安全

評估密碼系統安全性主要有三種方法：

1. 無條件安全性：
 - 假定攻擊者擁有無限的計算資源，仍然無法破譯該密碼系統。
2. 計算安全性：
 - 使用最好的方法破譯它所需要的計算資源，遠遠超出攻擊者所擁有的。
3. 可證明安全性：
 - 將密碼系統的安全性歸結為某個經過深入研究的數學難題（如大整數質因數分解、計算離散對數等）。這種評估方法只給出它們的等價性證明，沒有完全證明密碼方法本身的安全性。

計算安全性（實際安全性）

密碼系統要達到實際安全性，需要滿足以下任一準則：

1. 破譯該密碼系統的實際計算量（包括計算時間或費用）十分巨大，以致於在實際上是無法實現的。
2. 破譯該密碼系統所需要的計算時間超過被加密訊息有用的生命周期。
例如，戰爭中發起戰鬥攻擊的作戰命令只需要在戰鬥打響前需要保密；重要新聞消息在公開報道前需要保密的時間往往也只有幾個小時。
3. 破譯該密碼系統的成本超過被加密訊息本身的價值。

Classical Cryptography

Caesar Cipher

- Encryption: $C = E(P) = (P+k) \bmod (26)$
- Decryption: $P = D(C) = (C-k) \bmod (26)$
- only 26 possibilities of keys

```
Plain:  ABCDEFGHIJKLMNOPQRSTUVWXYZ  
Cipher: XYZABCDEFGHIJKLMNOPQRSTUVWXYZ
```

```
HELLO -> EBIIL
```

Affine Cipher

- Encryption: $C = E(P) = a * P + b \pmod{26}$, where $\gcd(a, n) = 1$
- Decryption: $P = D(C) = a^{-1} * (C - b) \pmod{26}$
- Caesar Cipher is an Affine Cipher with $a = 1$
- Still very easy to break, only $12*26$ possibilities of keys

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Substitution Cipher (替換式密碼)

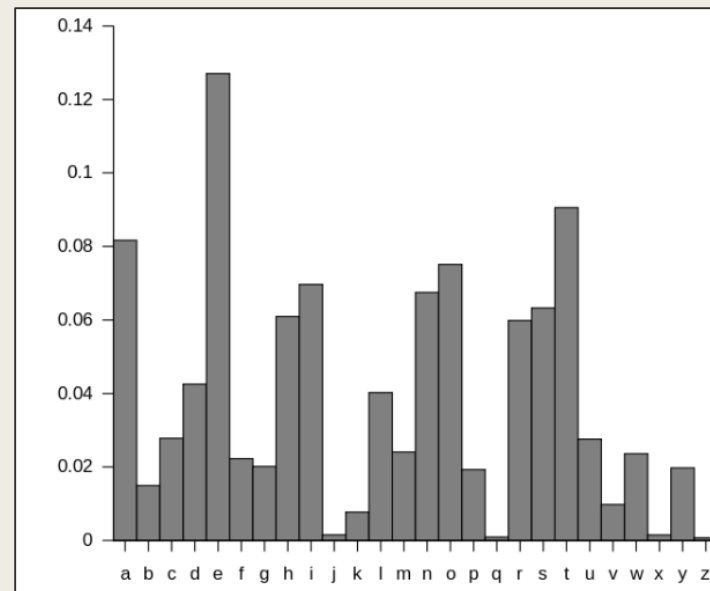
- Substitute each character with another
- Stronger than affine cipher, $26!$ possibilities of keys, still easy to break

```
Plaintext:  ABCDEFGHIJKLMNOPQRSTUVWXYZ  
Ciphertext: ZEBRASCDFGHIJKLMNOPQTUVWXY
```

```
HELLO -> DAIIL
```

Breaking Substitution Cipher

- 無法隱藏明文的統計特徵: Use frequency analysis
- Online tool: [quip quip](#)
- Try this Ciphertext:
 - *GEPV PV FS XFVJ XYFHOWX DI GEX NXFLSXVV DI VRQVGPGRGPDS BPOEXK SHWFQ*



Breaking Substitution Cipher

■ Try this Ciphertext:

- *GEPV PV FS XFVJ XYFHOWX DI GEX NXFLSXVV DI VRQVGPRGPD S BPOEXK SHWFQ*

quipqiup

BETA

quipqiup is a fast and automated cryptogram solver by [Edwin Olson](#). It can solve simple substitution ciphers often found in newspapers, including puzzles like cryptoquips (in which word boundaries are preserved) and patristocrats (inwhi chwor dboun darie saren t).

Puzzle:

GEPV PV FS XFVJ XYFHOWX DI GEX NXFLSXVV DI VRQVGPRGPD S BPOEXK SHWFQ

Clues: For example G=R QVW=THE

auto

Solve

0

-0.943

THIS IS AN EASY EXAMPLE OF THE WEAKNESS OF SUBSTITUTION CIPHER NMLAB

1

-1.249

THIS IS AN EASY EXAMPLE OF THE WEAKNESS OF S??STIT?TION CIPHER NMLA?

2

-1.367

THIS IS AN EASY EXAMPLE DO THE WEAKNESS DO S??STIT?TIDN CIPHER NMLA?

3

-1.454

THIS IS AG EASY EXAMPLE NO THE ?EA?GESS NO SUBSTITUTING CIPHER GMLAB

4

-1.619

THIS IS AN EASY EXAGCRE OF THE WEAKNESS OF SUBSTITUTION MICHEL NGRAB

5

-1.642

THIS IS AG EAS? EXAMPLE NO THE ?EA?GESS NO SUBSTITUTING CIPHER GMLAB

密碼破解技術

- 唯密文攻擊(Ciphertext Only Attack)
 - 破解者藉由蒐集所有可能的密文以找出明文或金鑰。
- 已知明文攻擊(Known Plaintext Attack)
 - 破解者藉由已知的明文與其相對應的密文以找出金鑰。
- 選擇明文攻擊(Chosen Plaintext Attack)
 - 攻擊者利用特殊方法將明文發送給傳送端，再由傳送者取得加密後的密文(即破解者可以控制明文與其相對應的密文)，以找出金鑰。
- 選擇密文攻擊(Chosen Ciphertext Attack)
 - 攻擊者利用特殊方法將密文發送給接收端，再由接收者取得解密後的明文(即破解者可以控制密文與其相對應的明文)，以找出金鑰。
- 窮舉法(Brute-Force Attack，暴力破解)
 - 破解者嘗試所有可能的金鑰來攻擊密碼系統。

Vigenère Cipher


- Block version of Caesar Cipher

```
Plaintext:  An apple a day keeps the doctor away  
Key:       ci pherc i phe rciph erc ipherc iphe  
Ciphertext: Cv pwtcg i shc bgmez xyg ldjxft ilhc
```

Breaking Vigenère Cipher

- Find repeated cipher text
- [English bigrams and trigrams frequency](#)

```
Plaintext:  An apple a day keeps the doctor away  
Key:       ci pherc i phe rciph erc ipherc iphe  
Ciphertext: Cv pwtcg i s hc bgmez xyg ldjxft il hc
```



Width is 18=> possible block
size: 1, 2, 3, 6

Rail Fence Cipher

- A kind of transposition ciphers
- Example:
 - $m = WE ARE DISCOVERED. FLEE AT ONCE$
 - $c = WECRL TEERD SOEEF EAOCA IVDEN$

W	.	.	.	E	.	.	.	C	.	.	.	R	.	.	.	L	.	.	.	T	.	.	.	E
.	E	.	R	.	D	.	S	.	O	.	E	.	E	.	F	.	E	.	A	.	O	.	C	.
.	.	A	.	.	.	I	.	.	.	V	.	.	.	D	.	.	.	E	.	.	.	N	.	.

- Online tool: <https://www.geocachingtoolbox.com/index.php?page=railFenceCipher>
- Try this Ciphertext:
 - $AaY--rpyfneJBeaaX0n-,ZZcs-uXeeSVJ-sh2tioaZ\}slrg,-ciE-anfGt.-eClyss-TzprttFlora\{GcouhQladctm0ltt-FYluuezTyorZ-$

Rail Fence Cipher

- Try this Ciphertext:

- *AaY--rpyfneJBeaaX0n-,ZZcs-uXeeSVJ-sh2tioaZ}slrg,-ciE-anfGt.-eClyss-TzprttFlora{GcouhQladctm0ltt-FYluuezTyorZ-*

Rail Fence cipher

Enter the number of rails and the offset, if any. Choose the method, either encrypt or decrypt, and enter the text. Optionally add a check to show the rail fence.

Number of rails (>1):	<input type="text" value="21"/>
Offset:	<input type="text" value="0"/>
Show rail fence:	<input type="checkbox"/> Delimiter: <input type="text" value="."/>
Method:	<input type="button" value="Decrypt"/>
Text:	<div><div>AaY--rpyfneJBeaaX0n-,ZZcs-uXeeSVJ-sh2tioaZ}slrg,-ciE-anfGt.-eClyss-TzprttFlora{GcouhQladctm0ltt-FYluuezTyorZ-</div><div></div></div>
	<input type="button" value="Reset fields"/>
Result:	<div><div>A-fence-is-a-structure-that-encloses-an-area,-SharifCTF{QmFzZTY0IGlzIGEgZ2VuZXJpYyB0ZXJt},-typically-outdoors.</div><div></div></div>

One Time Pad

- Encrypt: $P \oplus \text{key} = C$
- Decrypt: $C \oplus \text{key} = P$
- Theoretically unbreakable (if the key is randomly generated)

```
pt: "HELLO WORLD"  
key: "01230 98765"  
ct: "xt~y{ nwezq"
```

Modern Cryptography

對稱式加密 vs. 非對稱式加密



對稱加密技術的優缺點

■ 優點：

- 較快速
- 如果使用足夠大的金鑰，將難以破解

■ 缺點：

- 需要有一個安全性機制將金鑰安全的分送至交易的雙方
- 提供隱密性(Confidential)的安全性能力，無法提供不可否認的功能

非對稱加密技術的優缺點

- 每個使用者擁有一對金鑰: 公開金鑰(public key)和私密金鑰(private key)
- 訊息由其中一把金鑰加密後，必需由另一把金鑰解密
- 公開金鑰可以被公開發佈，而私密金鑰必需隱密的加以保存
- 優點：
 - 公開鑰匙可以公開發送
 - 提供隱密性、來源辨識與不可否認性等功能
- 缺點：
 - 效率較差

對稱式加密 vs. 非對稱式加密

- 二者各有優劣，實務上經常合併使用

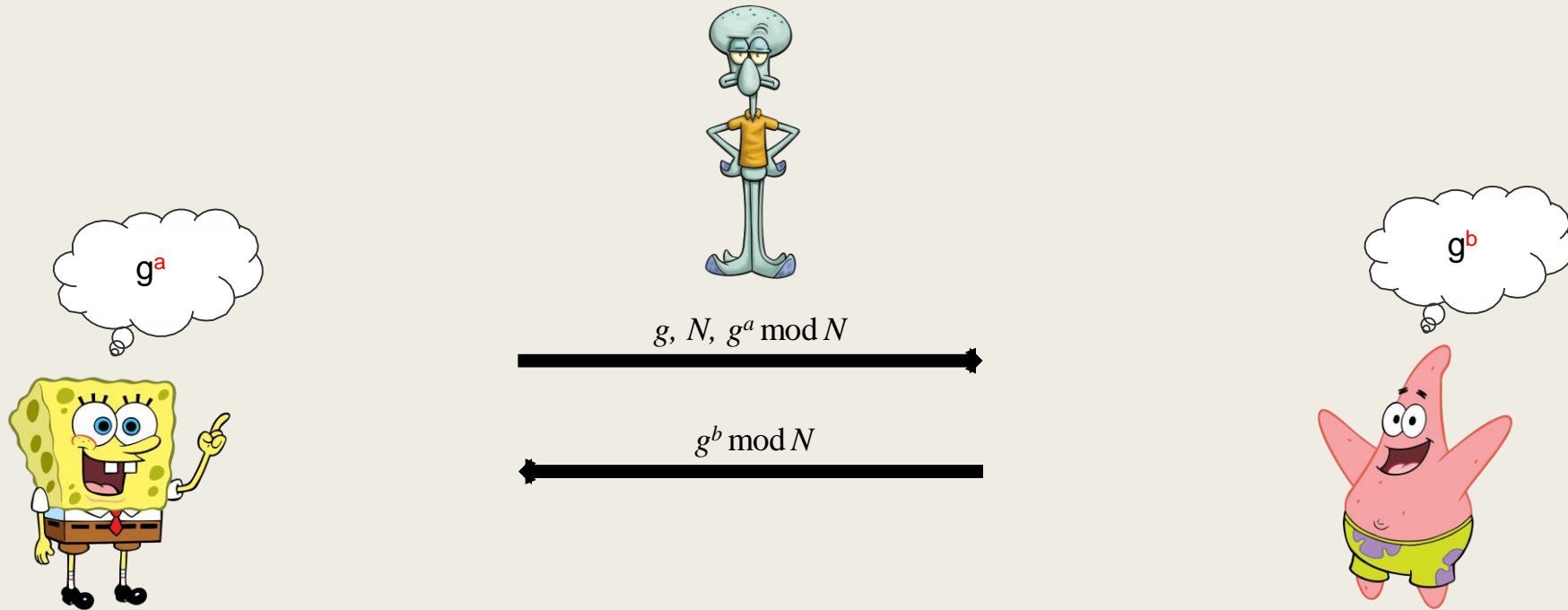
	對稱式加密	非對稱式加密
加解密的key是否相同	相同	不同
key可否公開	不可公開	公鑰可以公開 私鑰不可公開
key保管問題	如果與N個人交換訊息， 需保管好N把密鑰	無論與多少人交換訊息 只需保管自己的私鑰
加解密速度	快	慢
應用	常用於加密長度較長的資料 例：email	常用於加密長度較短的資料 例：數位簽章
功能	隱密性	隱密性、來源辨識性、 不可否認性

Diffie-Hellman Key Exchange

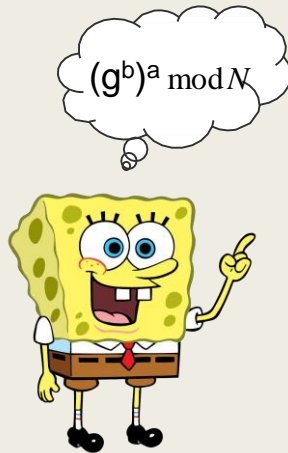
(迪菲-赫爾曼密鑰交換，D-H)

- 雖然本身是一個匿名（無認證）的金鑰交換協定，它卻是很多認證協定的基礎
- 這個金鑰可以在後續的通訊中作為對稱金鑰來加密通訊內容

Diffie-Hellman Key Exchange



Diffie-Hellman Key Exchange



Discrete Logarithm Problem

- Given N , g , x , find e such that $x = g^e \pmod{N}$
- e.g. $N = 23$, $g = 5$, $x = 3$, the answer is 16 · ($3 = 5^{16} \pmod{23}$)
- Really hard to solve

常見對稱式加密

- Stream Cipher (流密碼)
 - RC4 (WEP和WPA中採用的加密算法， WEP已被WPA、WPA2取代)
- Block Cipher (塊密碼)
 - RC5、RC6
 - IDEA
 - Blowfish、Twofish
 - AES (Advanced Encryption Standard)：Rijndael
 - DES (Data Encryption Standard)、Triple DES (3DES)

Stream Cipher

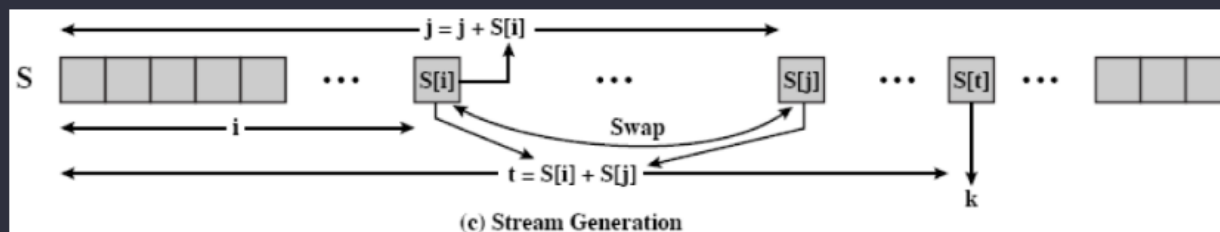
- 流密碼一般逐個 byte/bit 處理訊息
- 一般來說
 - 流密碼的密鑰長度會與明文的長度相同
 - 流密碼的密鑰派生自一個較短的密鑰，派生算法通常為一個偽隨機數生成算法
- 流加密目前來說都是對稱加密
- 偽隨機數生成算法生成的序列的隨機性越強，明文中的統計特徵被隱藏的更好

偽隨機數生成器 (PRNG , pseudo random number generator)

- 大多數使用PRNG的加密算法之安全性是基於以下假設：PRNG和真隨機序列的分辨是不可行的
- 一般來說，偽隨機數生成器的基本構造模塊為反饋移位暫存器
- 當然，也有一些特殊設計的流密碼，比如RC4

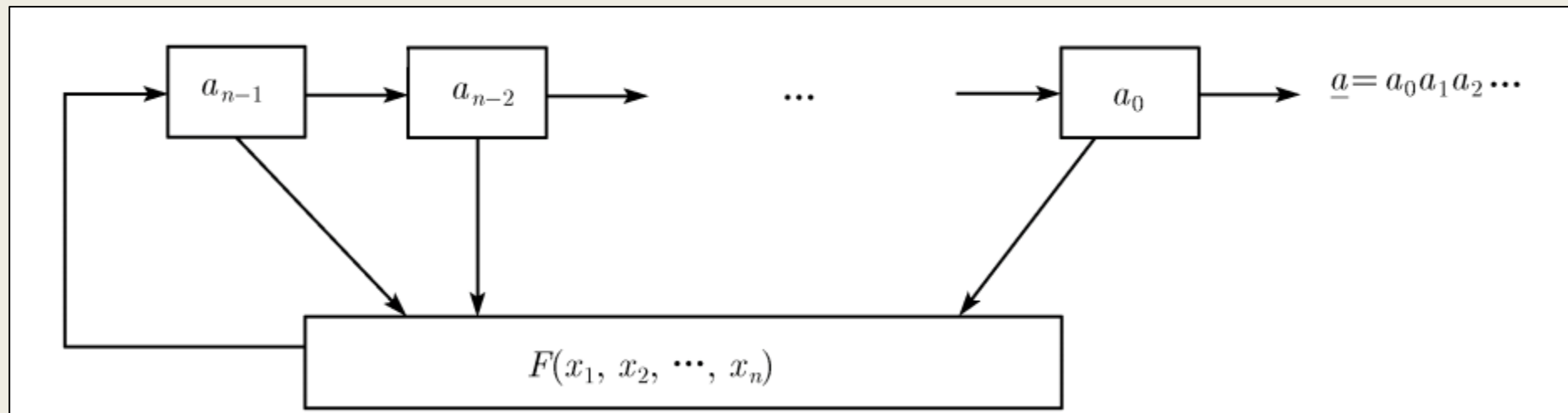
生成流密鑰

```
i = j = 0
for each message byte b
  i = (i + 1) (mod 256)
  j = (j + S[i]) (mod 256)
  swap(S[i], S[j])
  t = (S[i] + S[j]) (mod 256)
  print S[t]
```



反饋移位暫存器 (Feedback Shift Register)

- N-bit FSR:



- 初始值(種子): $a_0 \sim a_{n-1}$
- F 為反饋函數。如果 F 為線性函數，那麼我們稱其為線性反饋移位暫存器 (LFSR)，否則我們稱其為非線性反饋移位暫存器 (NFSR)

其他 PRNG

- 線性同餘生成器 (Linear Congruential Generator, LCG)
 - Java仍然使用LCG來實現PRNG。然而LCGs的質量很差。

它是根據遞歸公式：

$$N_{j+1} \equiv (A \times N_j + B) \pmod{M}$$

其中 A, B, M 是產生器設定的常數。

類別	隨機性	不可預測性	不可重現性
弱偽隨機數	✓	✗	✗
強偽隨機數	✓	✓	✗
真隨機數	✓	✓	✓

- 梅森旋轉算法 (Mersenne twister)
 - 是R、Python、Ruby、IDL、Free Pascal、PHP、Maple、Matlab、SageMath、Microsoft Excel、GNU多重精度運算庫和GSL的默認偽隨機數產生器
 - 有許多其它算法具有更強的安全性，但這些算法的速度非常緩慢，對於許多應用是不實際的
 - 最為廣泛使用Mersenne Twister的一種變體是MT19937
- xorshift、WELL family of generators (Well Equidistributed Long-period Linear)

PRNG問題

- 通常來說，偽隨機數產生器可能會有以下問題
 - 某些種子狀態的周期比預期的短(在這種情況下，這種種子狀態可以稱為「弱」)
 - 生成序列時數字分布不均勻
 - 某些值出現的位置之間的距離與隨機序列分布的距離不同
 - 連續值的關聯性

Block Cipher

- 每次加密一塊明文
- 明文可長可短，因此在塊加密時需要用padding輔助，即padding 到指定分組長度
- Shannon 提出的兩大基本策略：
 - 混淆
 - 擴散

基本策略(1) 混淆

- Confusion，將密文與密鑰之間的統計關係變得盡可能複雜，使得攻擊者即使獲取了密文的一些統計特性，也無法推測密鑰。
- 一般使用複雜的非線性變換可以得到很好的混淆效果，常見的方法如下
 - 乘法
 - 替換: S-box (Substitution-box，替換盒)

S-box (Substitution-box , 替換盒)

- S-Box接受特定數量的輸入位元 m ，並將其轉換為特定數量的輸出位元 n ，其中 n 不一定等於 m
- S-Box通常是固定的 (例如DES和AES加密演算法)
- 也有一些加密演算法的S-Box是基於金鑰動態生成的 (例如Blowfish和Twofish)

DES的6×4位元S盒 (S₅) 是一個很好的例子：

S ₅		中間四個位元															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
首尾位元	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

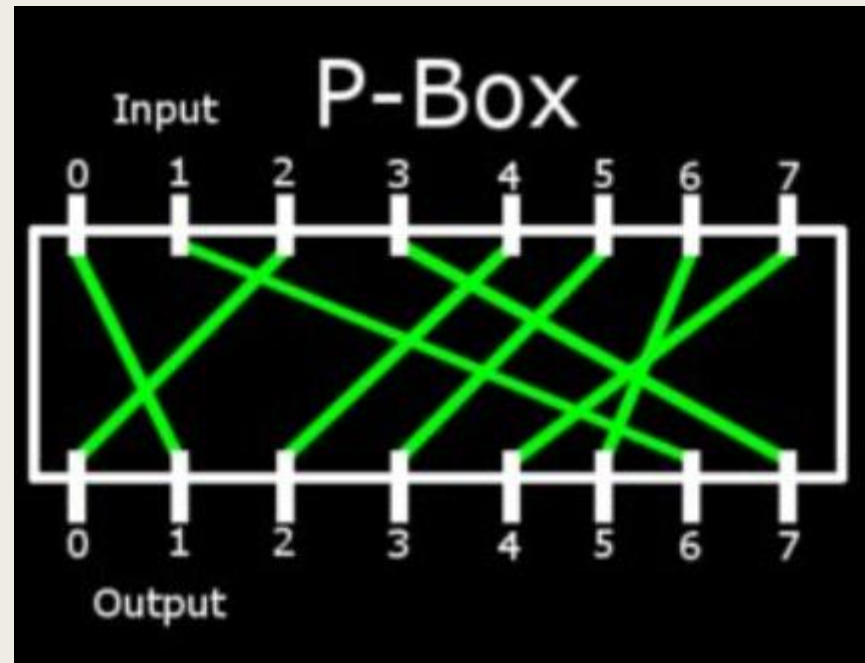
給定6位元輸入，將首尾兩個位元作為行條件、中間四個位元 (inner four bits) 作為列條件進行查表，最終獲得4位元輸出。例如，輸入「011011」，通過首尾兩個位元「01」和中間的位元「1101」進行查表，最終的輸出應該是「1001」^[2]。

基本策略(2) 擴散

- Diffusion，使得明文中的每一位影響密文中的許多位。常見的方法有
 - 線性變換
 - 移位，循環移位
 - 置換: P-box (Permutation-box，置換盒)

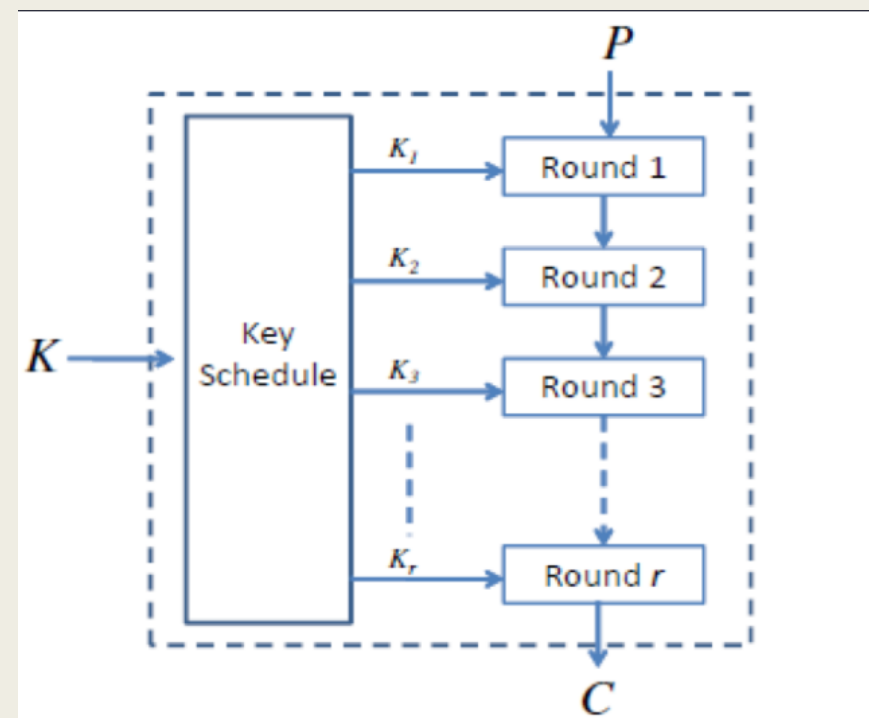
P-box (Permutation-box , 置換盒)

- 進行位元洗牌的方法，造成擴散



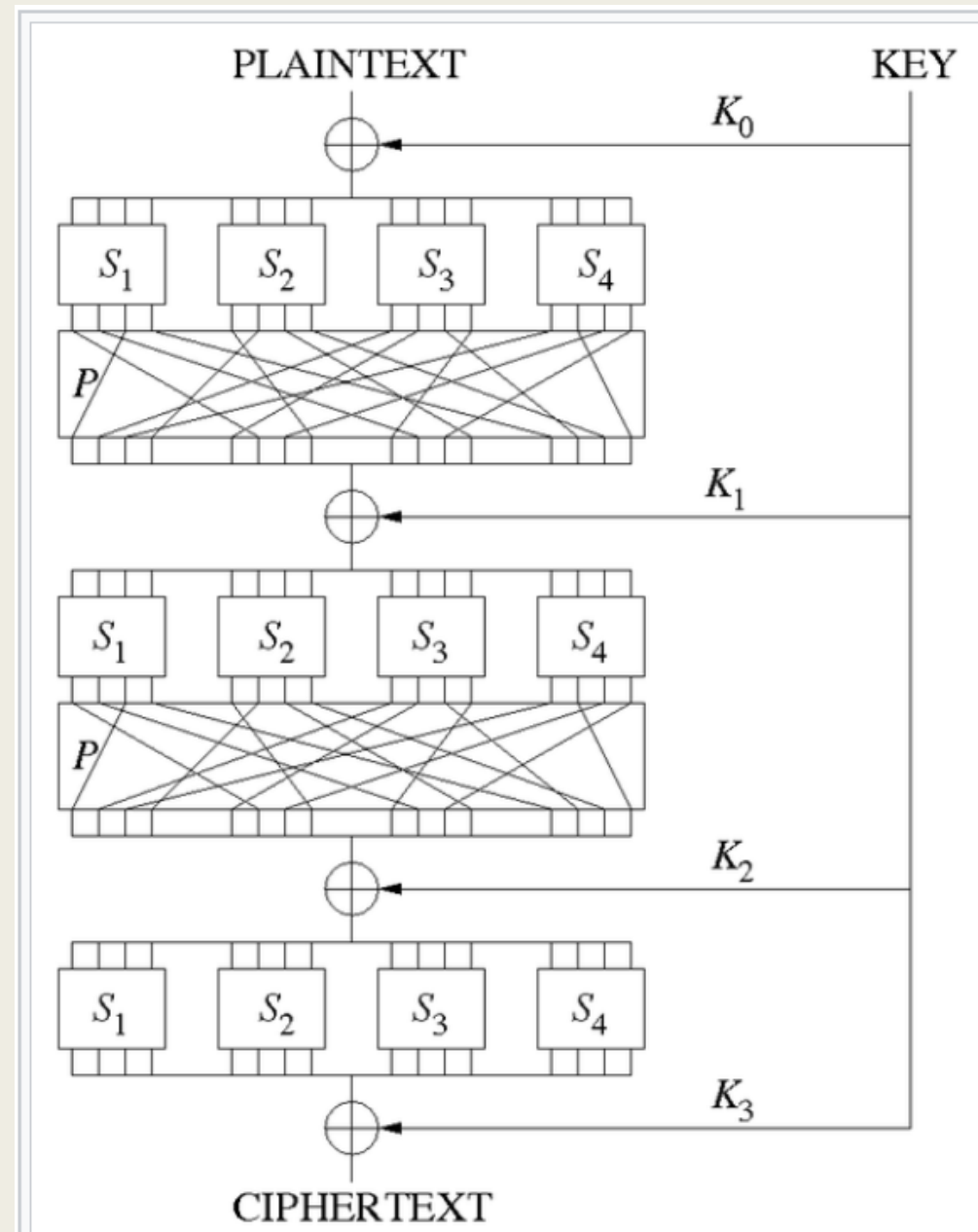
Block Cipher常見加解密結構: 疊代結構

- 因為疊代結構便於設計與實現，同時方便安全性評估
- 疊代結構一般包括三個部分
 - 金鑰排程
 - 輪加密函數
 - 輪解密函數
- 金鑰排程(金鑰擴展)
 - 目前金鑰擴展的方法有很多，沒有見到什麼完美的方法，基本原則是使得金鑰的每一個 bit 盡可能影響多輪的輪金鑰
- 目前輪函數主要有以下設計方法
 - SP-Network (AES)
 - Feistel Network (DES)
 - 其他方案



Substitution-Permutation Network (SP-network , SPN)

- 由於其實施於硬體的高效性，SPN的應用十分廣泛
 - AES (Rijndael), 3-Way, Kalyna, Kuznyechik, PRESENT, SAFER, SHARK, and Square



Feistel Network

- 由 DES 設計者之一 Horst Feistel 所發明

擴張置換

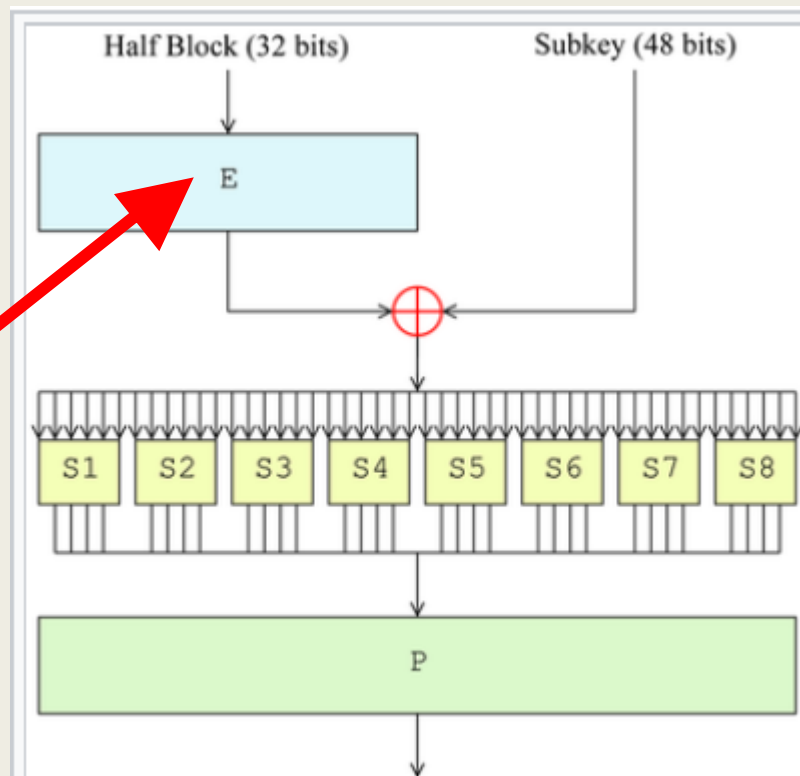
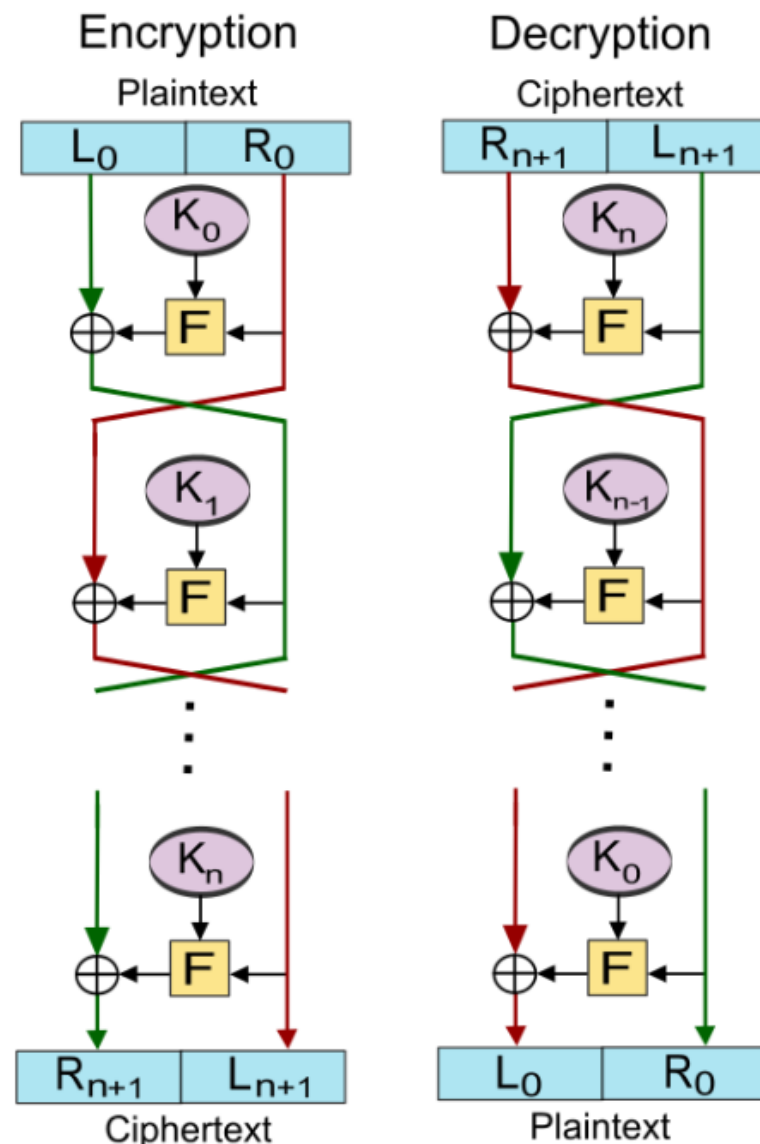
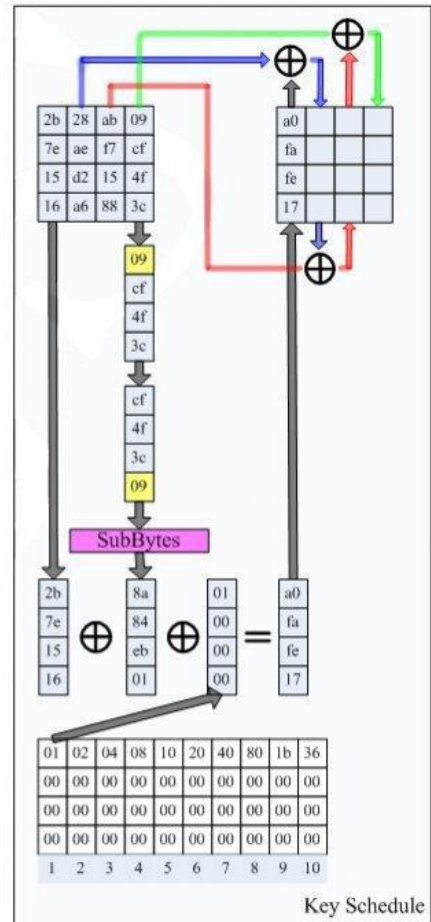
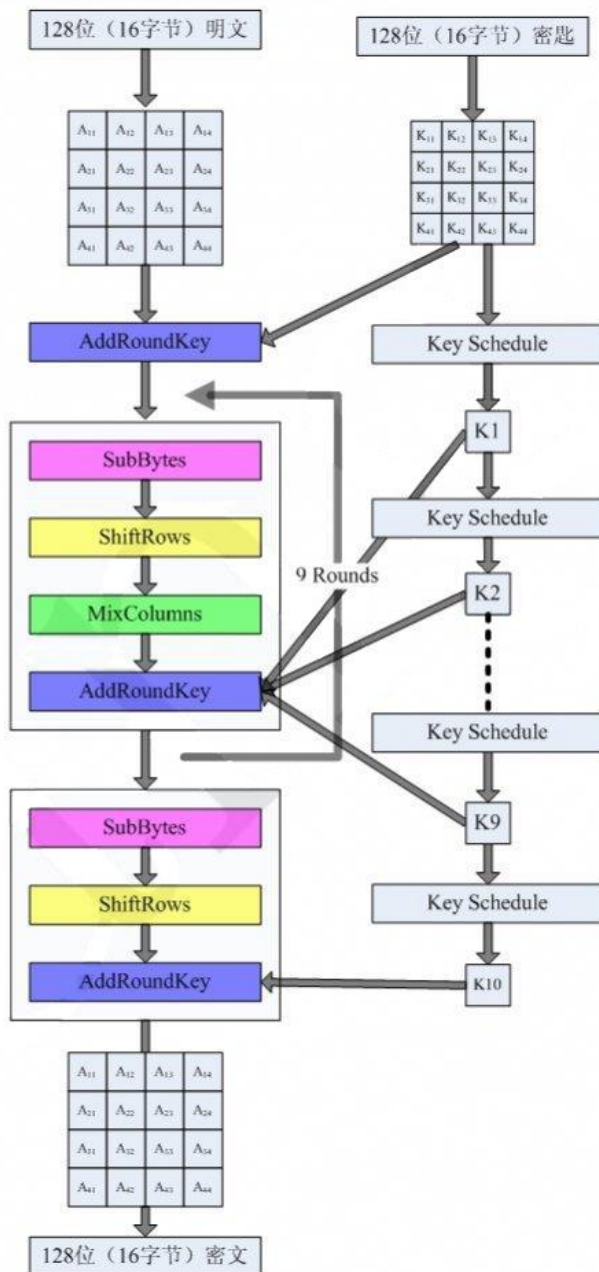
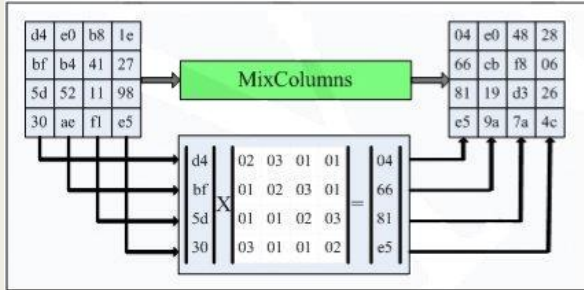
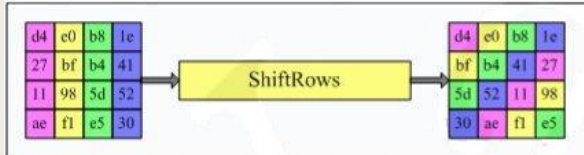
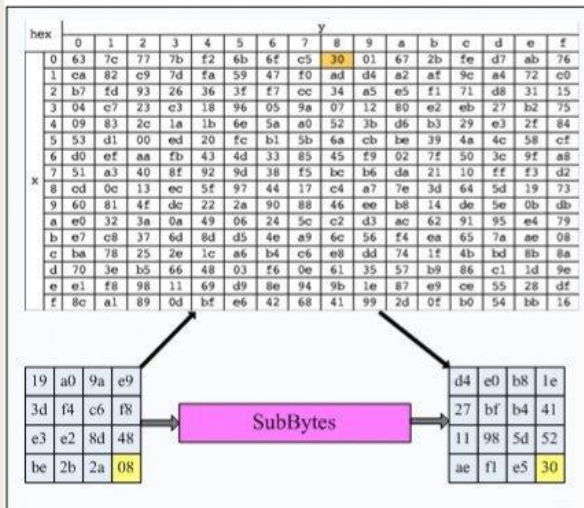
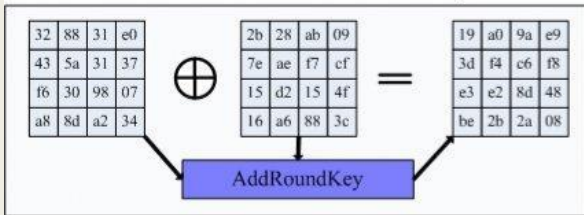


图2—DES的費斯妥函数（F函数）



Many block ciphers, such as DES and Blowfish utilize structures known as *Feistel ciphers*

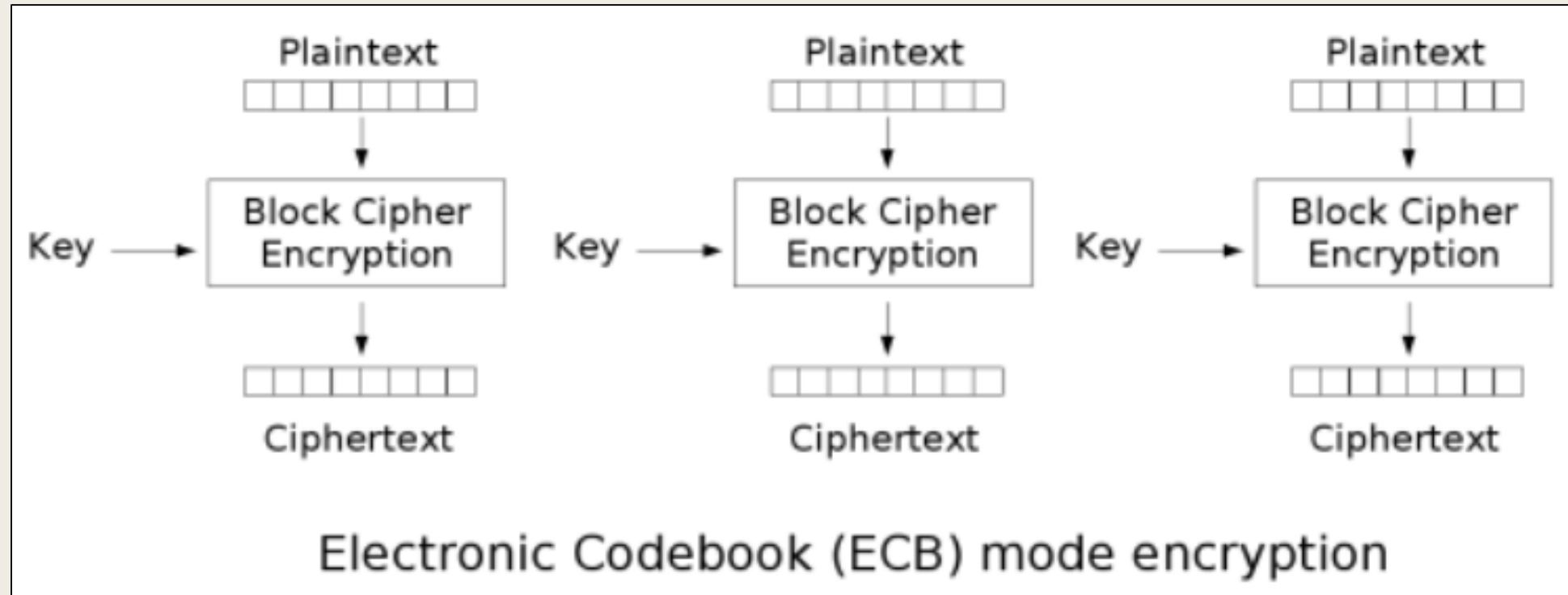
AES加密算法图解



Mode of Operation

- ECB mode (Electronic Codebook)
- CBC mode (Cipher Block Chaining)
- CFB mode (Cipher Feedback)
- OFB mode (Output Feedback)
- CTR mode (Counter)

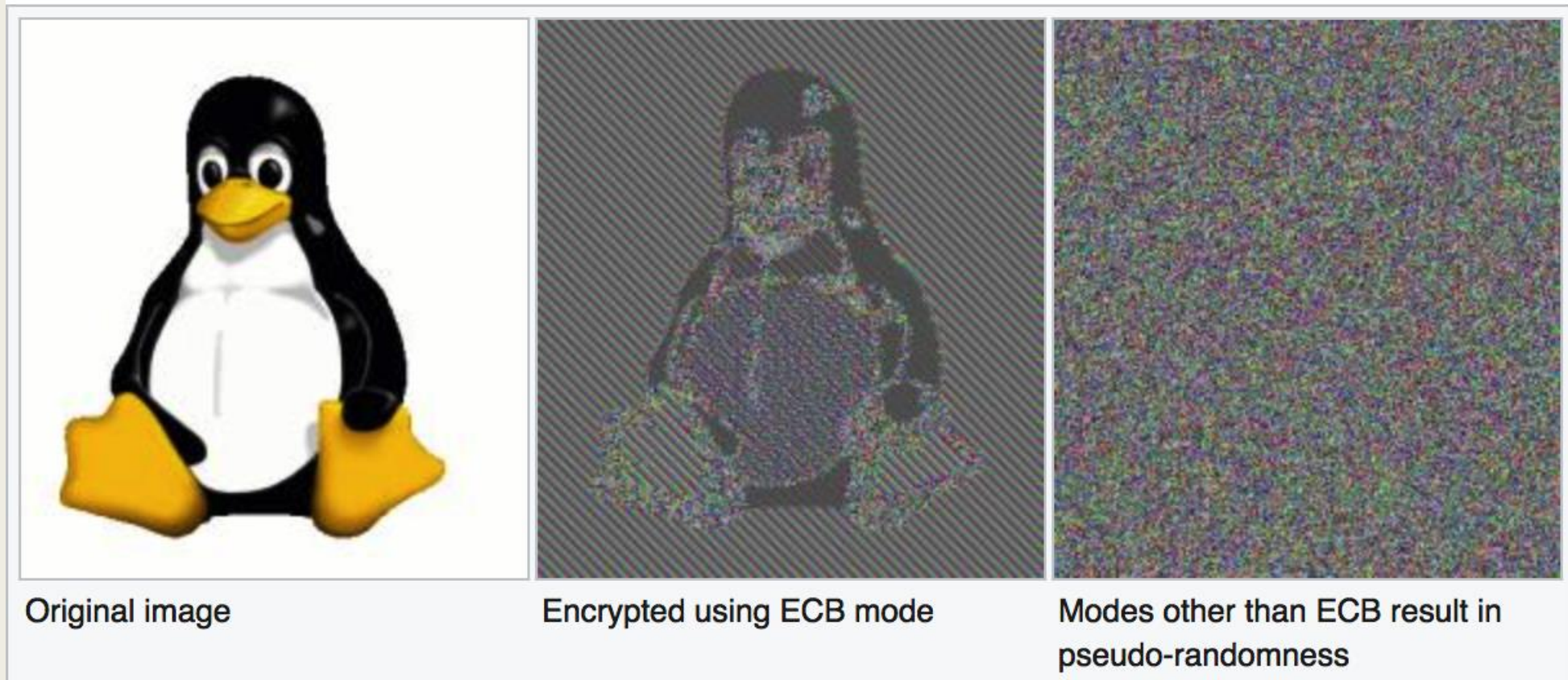
ECB mode (Electronic Codebook)



■ 優點

- 實現簡單
- 不同明文分組的加密可以並行計算，速度很快

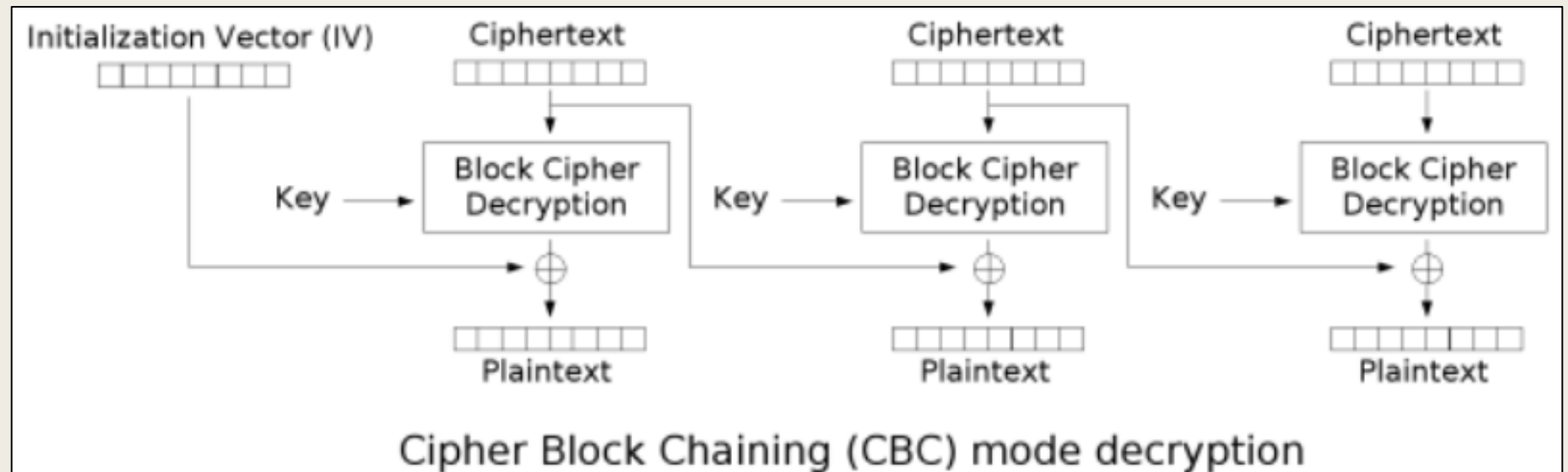
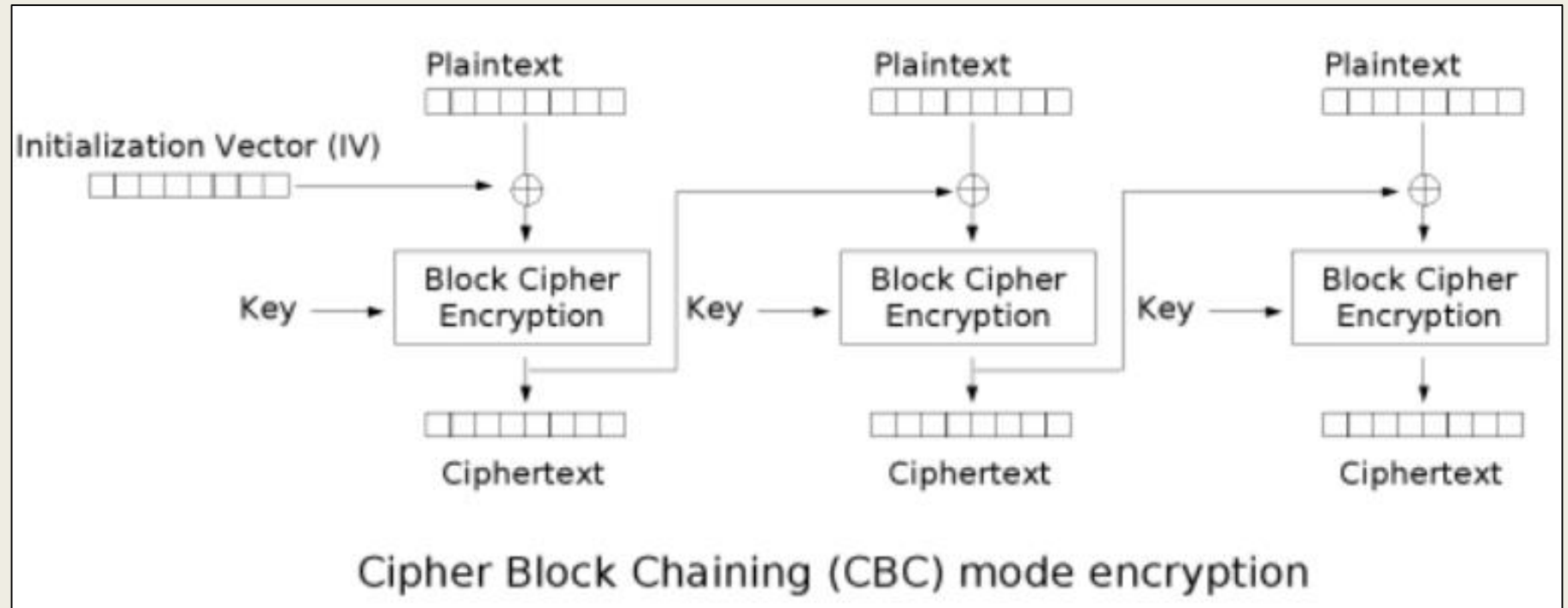
ECB mode (Electronic Codebook)



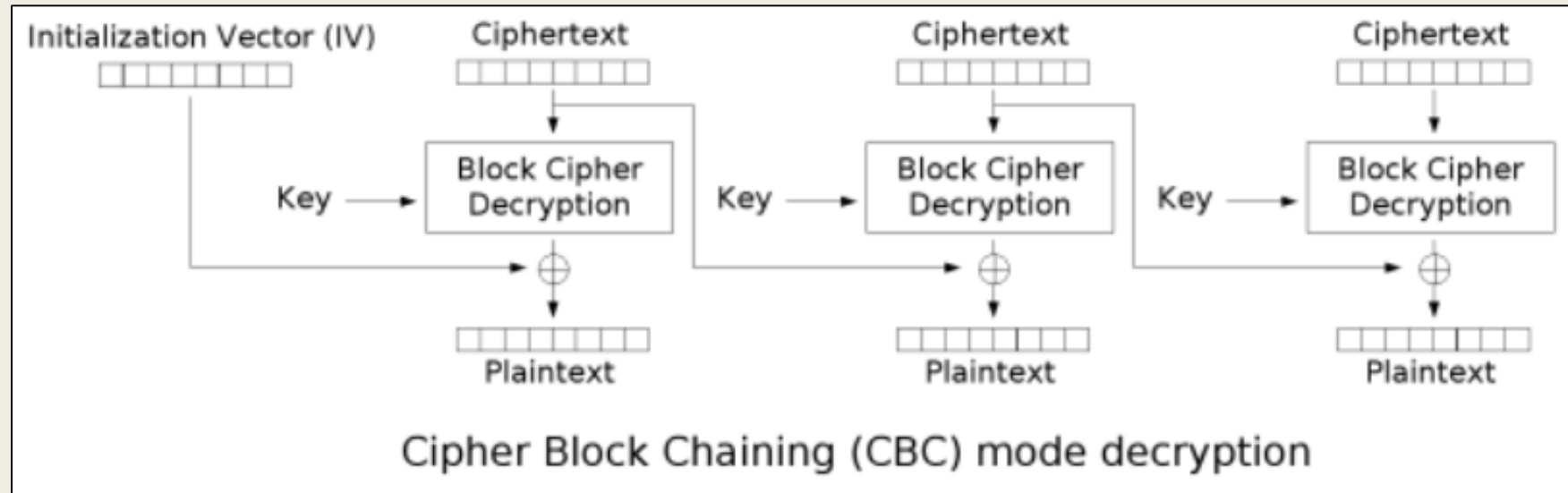
- 缺點
 - 同樣的明文塊會被加密成相同的密文塊，不會隱藏明文分組的統計規律

CBC mode (Cipher Block Chaining)

IV 不要求保密
IV 必須是不可預測的



CBC mode (Cipher Block Chaining)



■ 優點

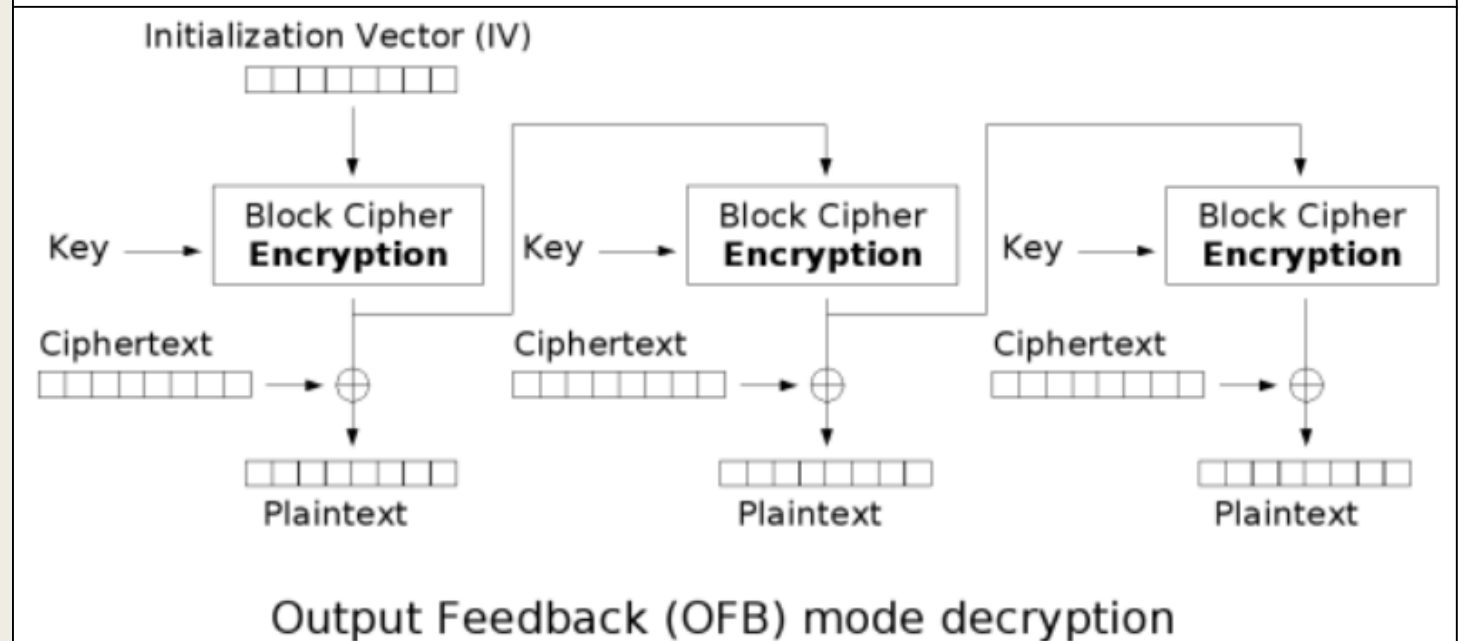
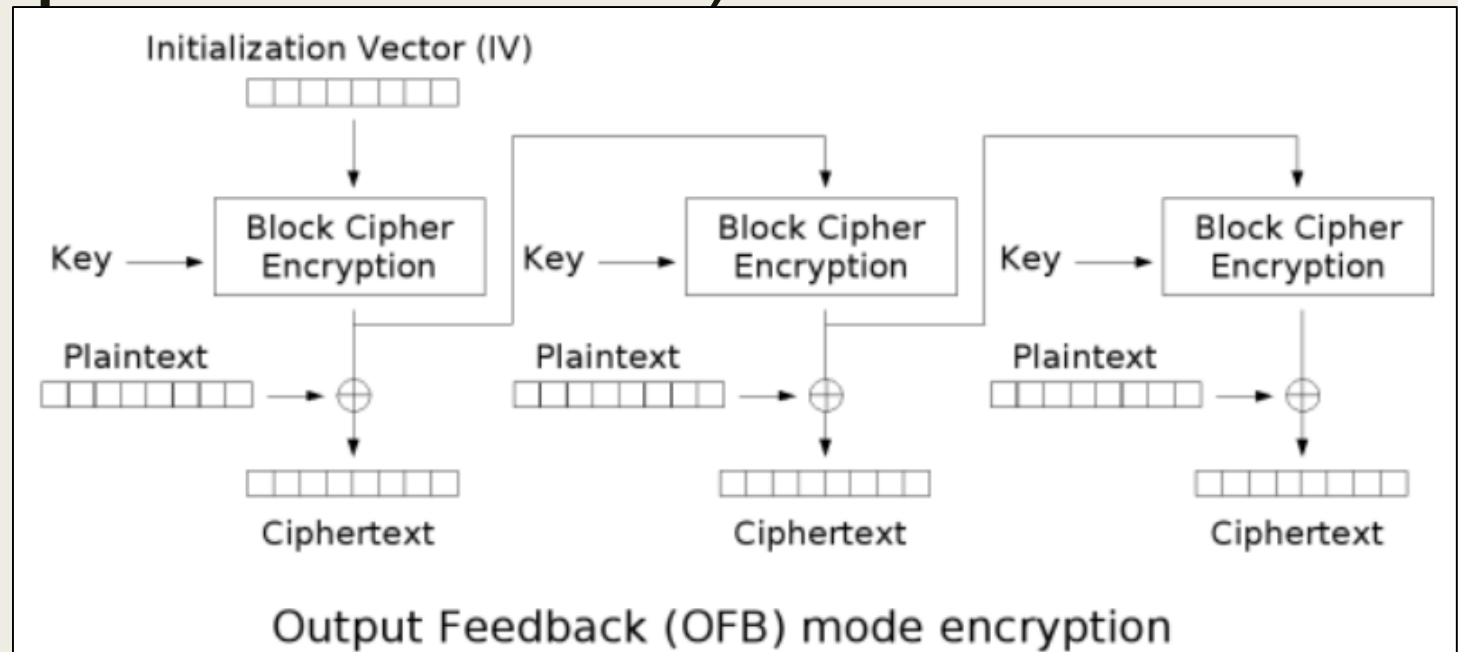
- 密文塊不僅和當前密文塊相關，而且和前一個密文塊或 IV 相關，隱藏了明文的統計特性。
- 具有有限的兩步錯誤傳播特性，即密文塊中的一位變化只會影響當前密文塊和下一密文塊，即第 k 塊起密文正確，則第 $k+1$ 塊就能正常解密。

■ 缺點

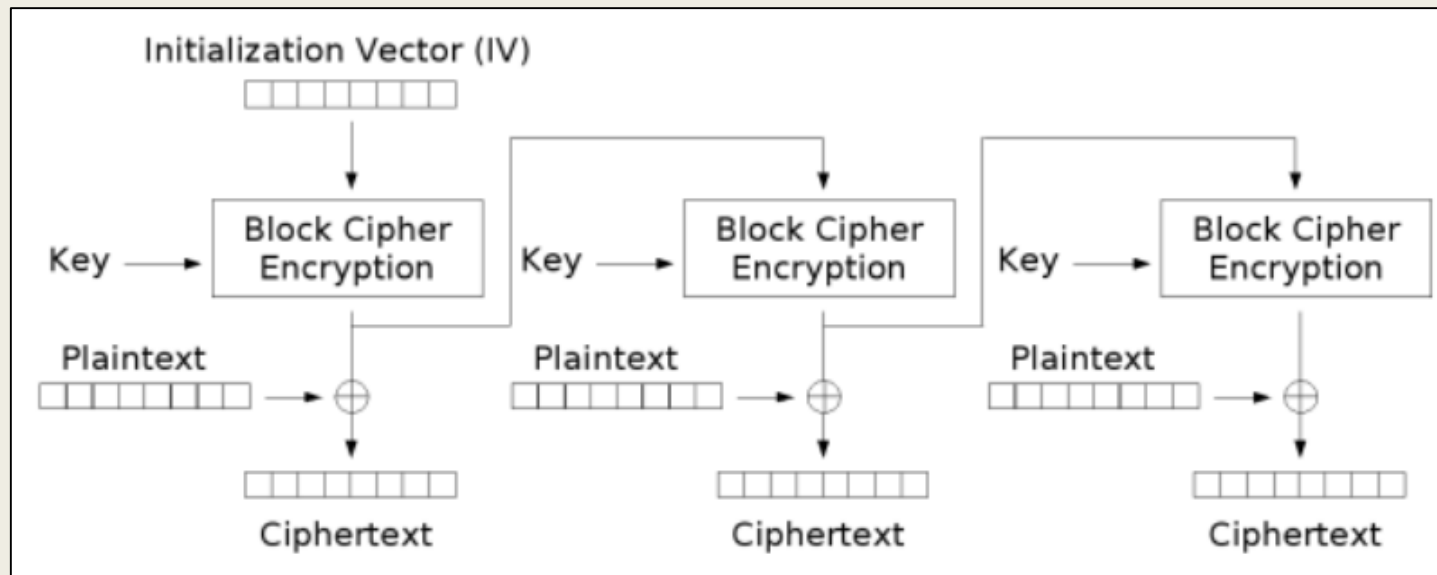
- 加密不能並行，解密可以並行

OFB mode (Output Feedback)

可以將塊密碼變成串流加密法



OFB mode (Output Feedback)



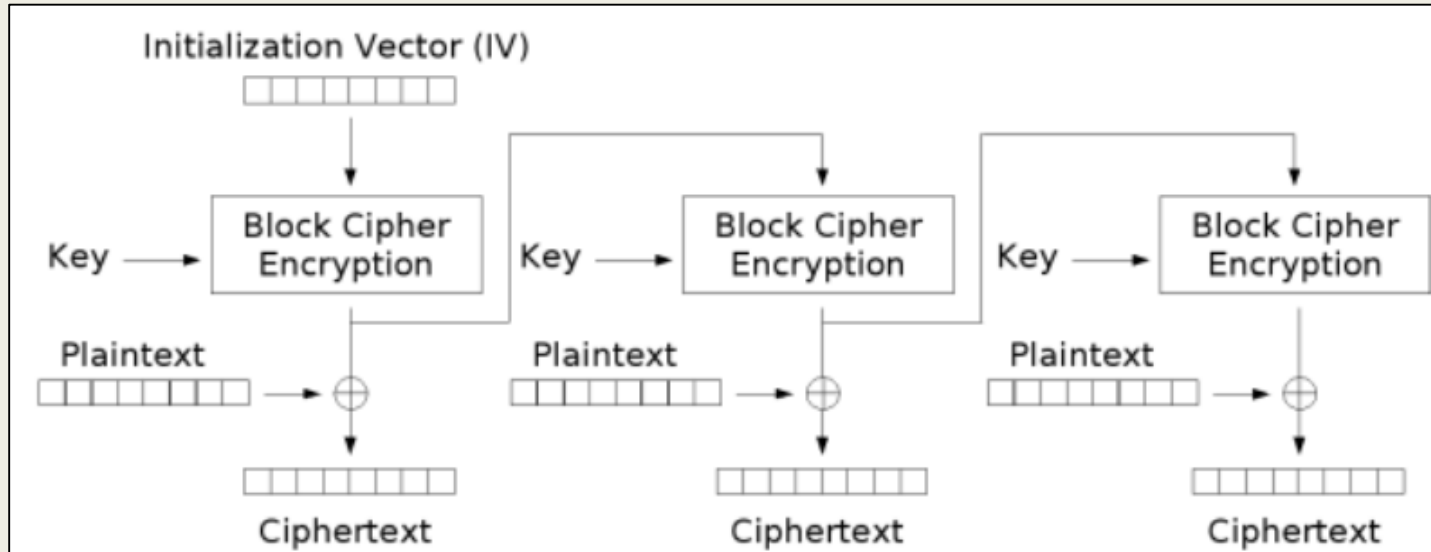
■ 優點：

- 只需要加密器，加密做兩次相當於解密
- 依序使用的 key，可以事先算出來，然後依次使用
- 雜訊下支持的能力好 (沒有錯誤傳播)

■ 缺點：

- 訊息被修改時，不易被發現，只單純影響單一明文 (沒有錯誤傳播)
- 起始狀態的 Initial Vector，不能重複使用，否則很容易被攻擊者抓到

OFB mode (Output Feedback)



- Plaintext "I w i l l s e n d y o u \$ 0 0 0 0 0 1 0 d o l l a r s"
- Ciphertext "1119a25cde3b97de48cb26e1ada3f12c09482fd4b69178a57007dede79d1358f"

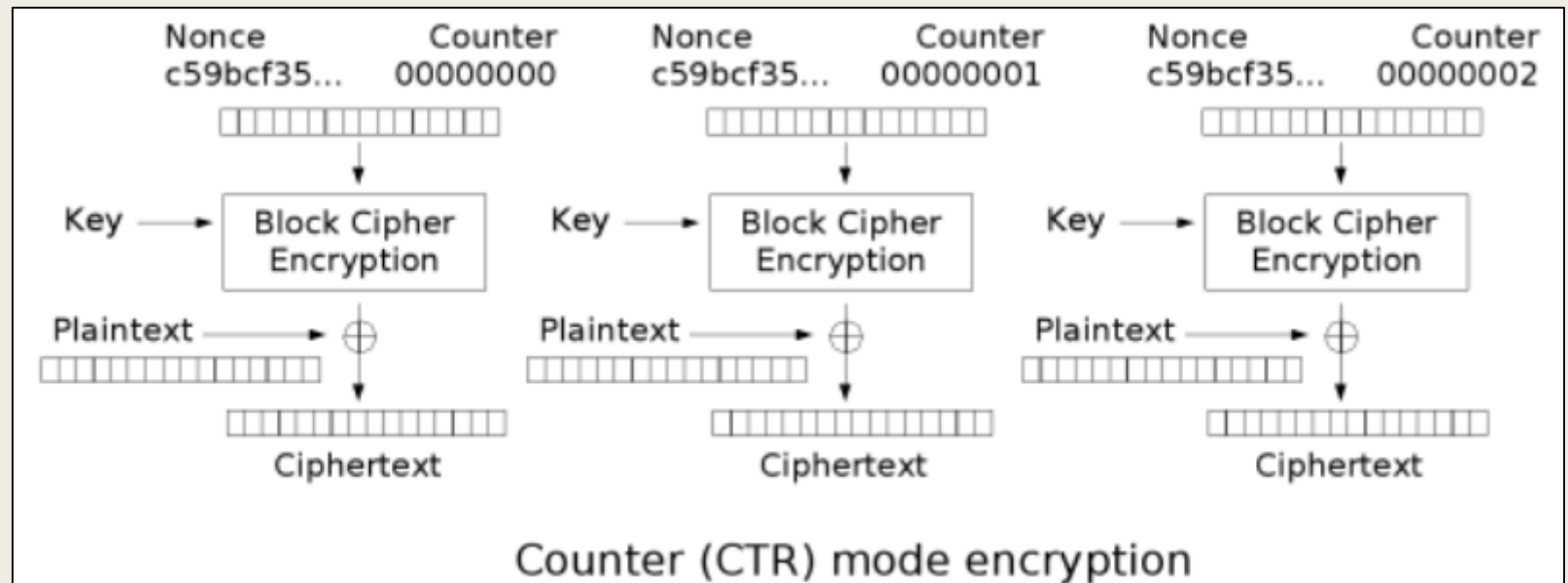
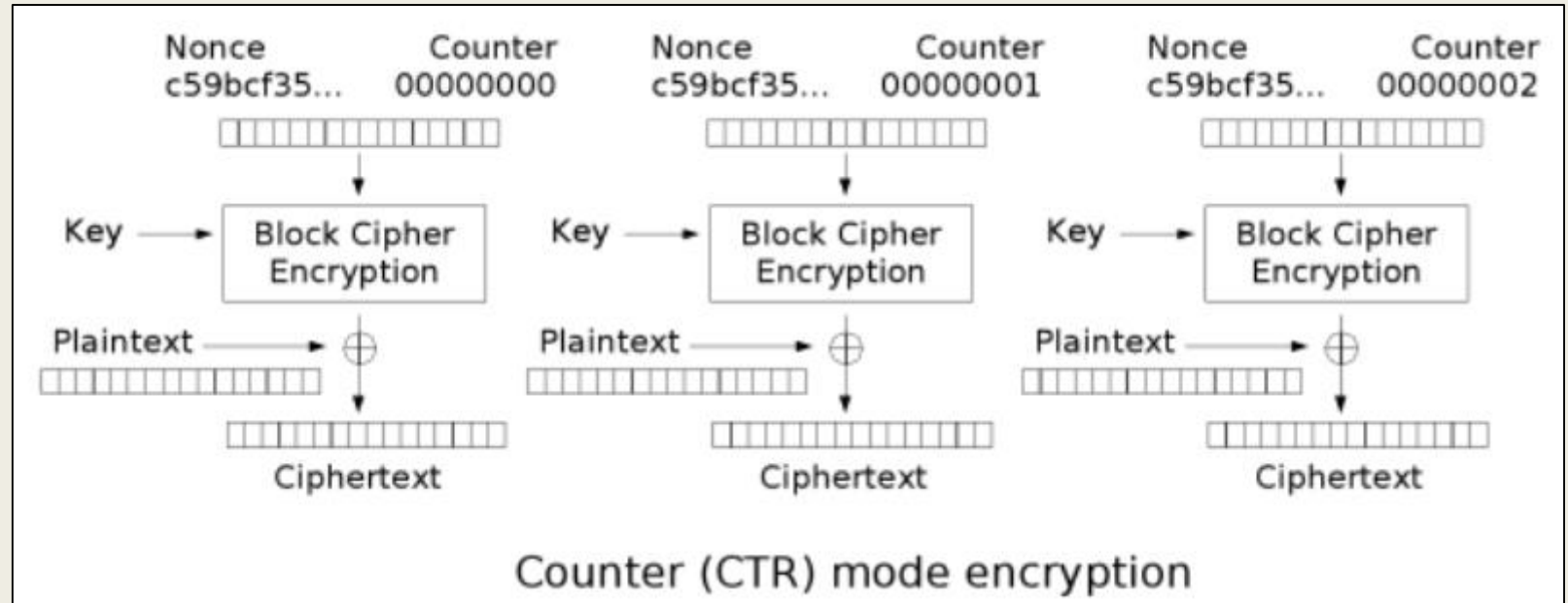
482fd4b69178a5 $\xrightarrow{\text{XOR with 0x07}}$ 4f28d3b1967fa2

- New Ciphertext "1119a25cde3b97de48cb26e1ada3f12c094f28d3b1967fa27007dede79d1358f"
- Decrypted "I w i l l s e n d y o u \$ 7 7 7 7 7 6 7 d o l l a r s"

CTR mode (Counter)

類似於 OFB，直接利用
計數器產生金鑰流

圖中的「Nonce」與其
它圖中的IV（初始化向
量）相同



常見非對稱式加密

■ RSA

- 1978 年，Rivest、Shamir 及 Adleman 三位學者利用**分解大質數**的困難度所提出的非對稱性金鑰演算法，是目前最普遍的公開金鑰加密法

■ 橢圓曲線密碼學 (Elliptic Curve ; ECC)

- 新一代的公開金鑰演算法，由於ECC只需使用**較短的金鑰**長度就可達到與較長金鑰的RSA演算法強度一般，所以非常適合在例如智慧卡等的資源有限環境下使用

■ ElGamal 加密算法

- 基於 Diffie-Hellman 密鑰交換的非對稱加密算法
- ElGamal加密算法可以定義在任何循環群G上。它的安全性取決於**循環群G上的離散對數難題**

RSA


- 計算 $N = pq$, p 和 q 為隨意兩個大的質數 , p 不等於 q
- 計算 $r = \varphi(N) = \varphi(p)\varphi(q) = (p-1)(q-1)$
- 選擇一個小於 r 的整數 e , e 與 r 互質
- 計算 $d = e^{-1} \pmod{r}$

- Encryption: $c = m^e \pmod{N}$
- Decryption: $m = c^d \pmod{N}$

- $c^d = m^{ed} = m^{1 + k \cdot \varphi(N)} = m \cdot 1^k = m \pmod{N}$, 歐拉定理

Hash Function

- 雜湊函數將任何長度的輸入訊息，轉換成一個長度較短且固定的輸出，此輸出訊息為雜湊值 (Hash Value)或訊息摘要(Message Digest)

Hash() = feaf57dab24b1da14425c098f37805e1

- 一個理想的密碼雜湊函式應該有四個主要的特性：
 - 對於任何一個給定的訊息，它都很容易就能運算出雜湊數值
 - 難以由一個已知的雜湊數值，去推算出原始的訊息
 - 在不更動雜湊數值的前提下，修改訊息內容是不可行的
 - 對於兩個不同的訊息，只有極低的機率會產生相同的雜湊數值

常見的雜湊演算法

- ~~MD2、MD4、MD5~~
- Secure Hash Algorithm (SHA)
 - ~~— SHA-1~~
 - SHA-2 family (SHA-224, SHA-256, ...)
 - SHA-3
- RIPEMD-160

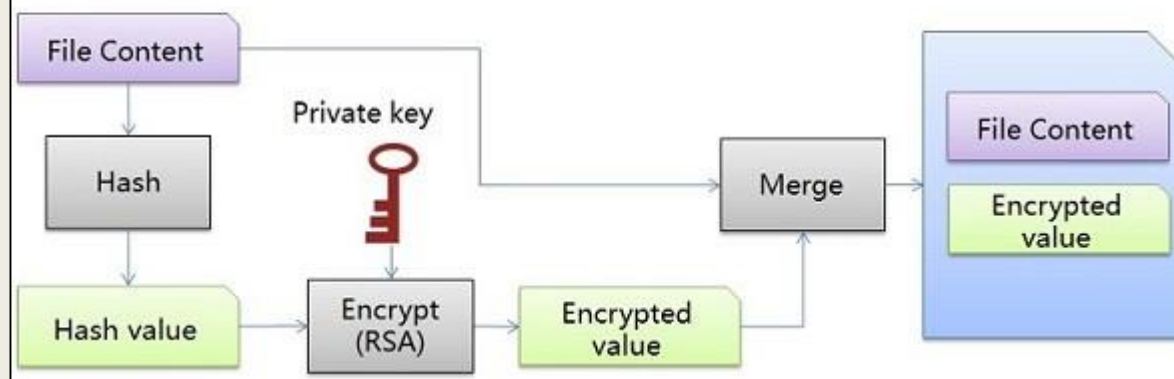
數位簽章

- 在數位簽章中，使用私鑰加密（相當於生成簽名），公鑰解密（相當於驗證簽名）
- 可以直接對訊息進行簽名（即使用私鑰加密，此時加密的目的是為了簽名，而不是保密），驗證者用公鑰正確解密訊息，如果和原訊息一致，則驗證簽名成功
- 但通常我們會對訊息的摘要 (Message Digest) 簽名，因為通常訊息摘要的長度遠小於訊息原文，使得簽名（非對稱加密）的效率大大提高
- 在實際使用中，我們既想加密訊息，又想簽名，所以要對加密和簽名組合使用，比如TLS就組合了加密和簽名

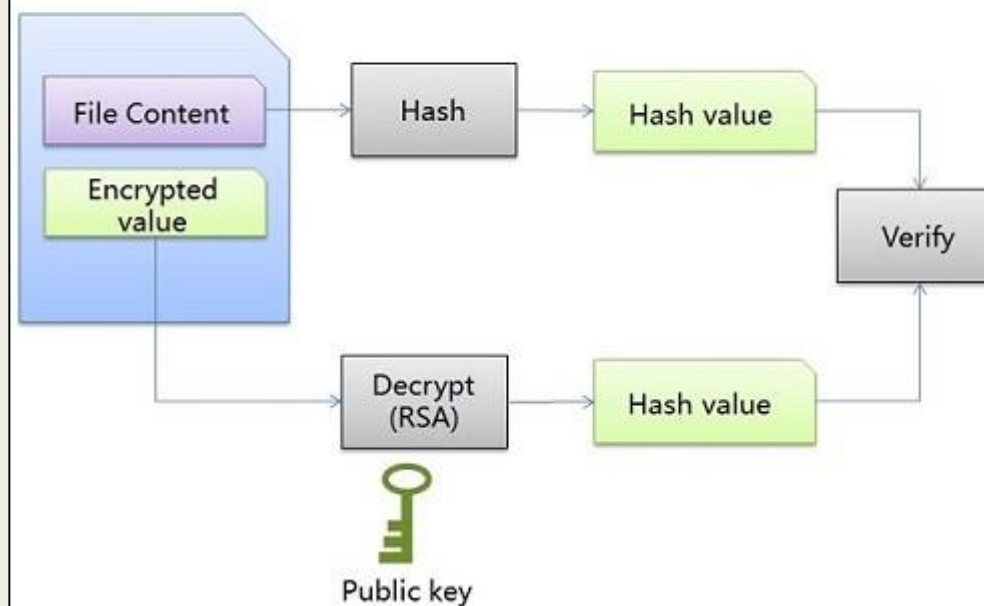
數位簽章

- 確保檔案來源
- 檔案未遭竄改

數位簽章Encryption flow



數位簽章Decryption flow



Message Authentication Code (MAC，訊息鑑別碼)

- 類似數位簽章，但是是基於金鑰
- $H(\text{金鑰} \parallel \text{訊息})$
 - 長度擴充攻擊 (Length extension attacks)
- 雜湊訊息鑑別碼 (Hash-based message authentication code，縮寫為HMAC)

根據RFC 2104，HMAC的數學公式為：

$$HMAC(K, m) = H\left((K' \oplus opad) \parallel H((K' \oplus ipad) \parallel m)\right)$$

認證加密

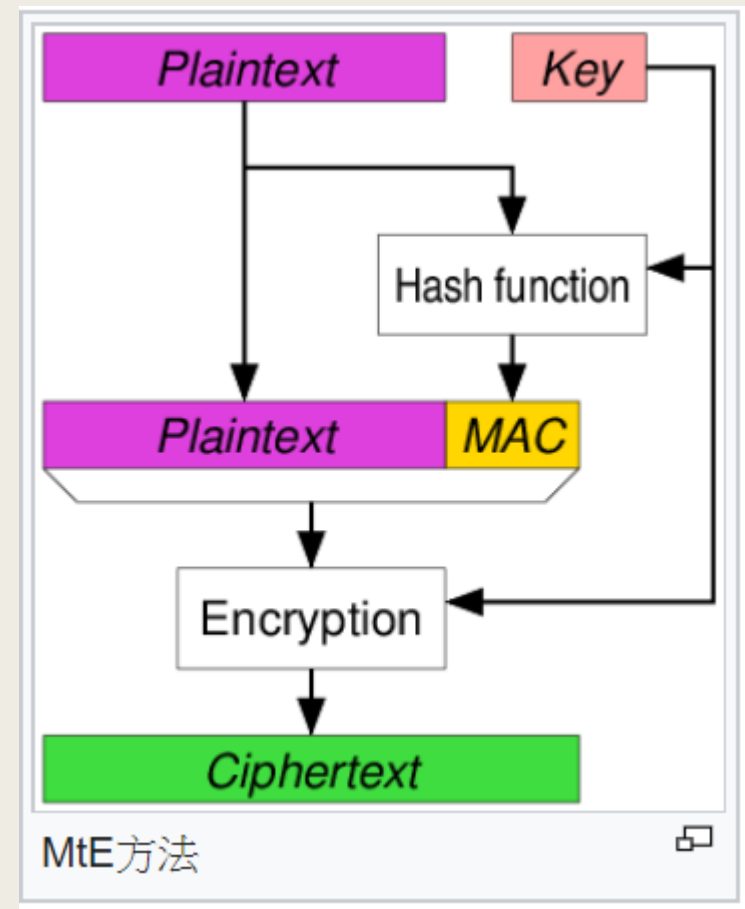
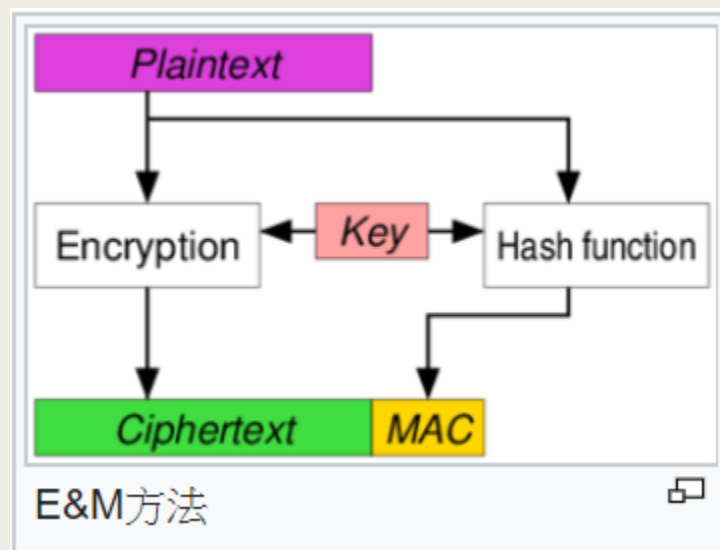
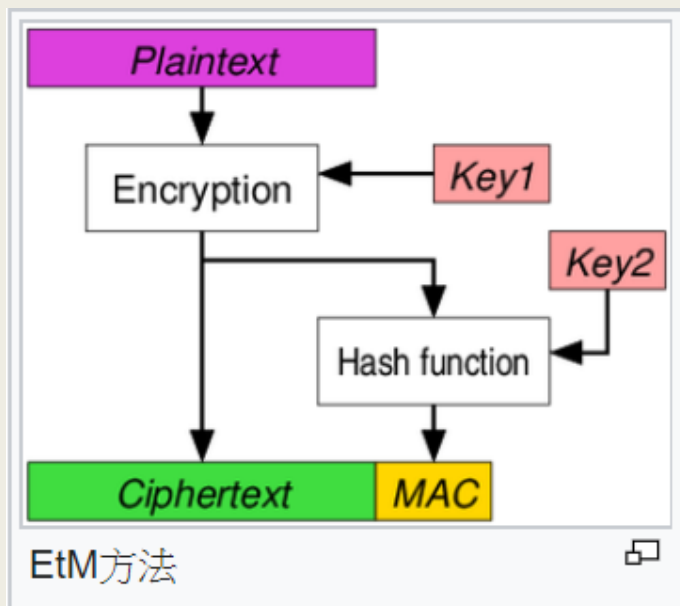
(Authenticated encryption , AE)

- 一種能夠同時保證資料的保密性、完整性和真實性的一種加密模式
- 人們觀察發現安全地將**保密模式**與**認證模式**組合可能是容易出錯和困難的，於是認證加密應運而生
- 認證加密的方法
 - *Encrypt-then-MAC (EtM)*
 - *Encrypt-and-MAC (E&M)*
 - *MAC-then-Encrypt (MtE)*

認證加密

(Authenticated encryption , AE)

- Encrypt-then-MAC (EtM)
- Encrypt-and-MAC (E&M)
- MAC-then-Encrypt (MtE)

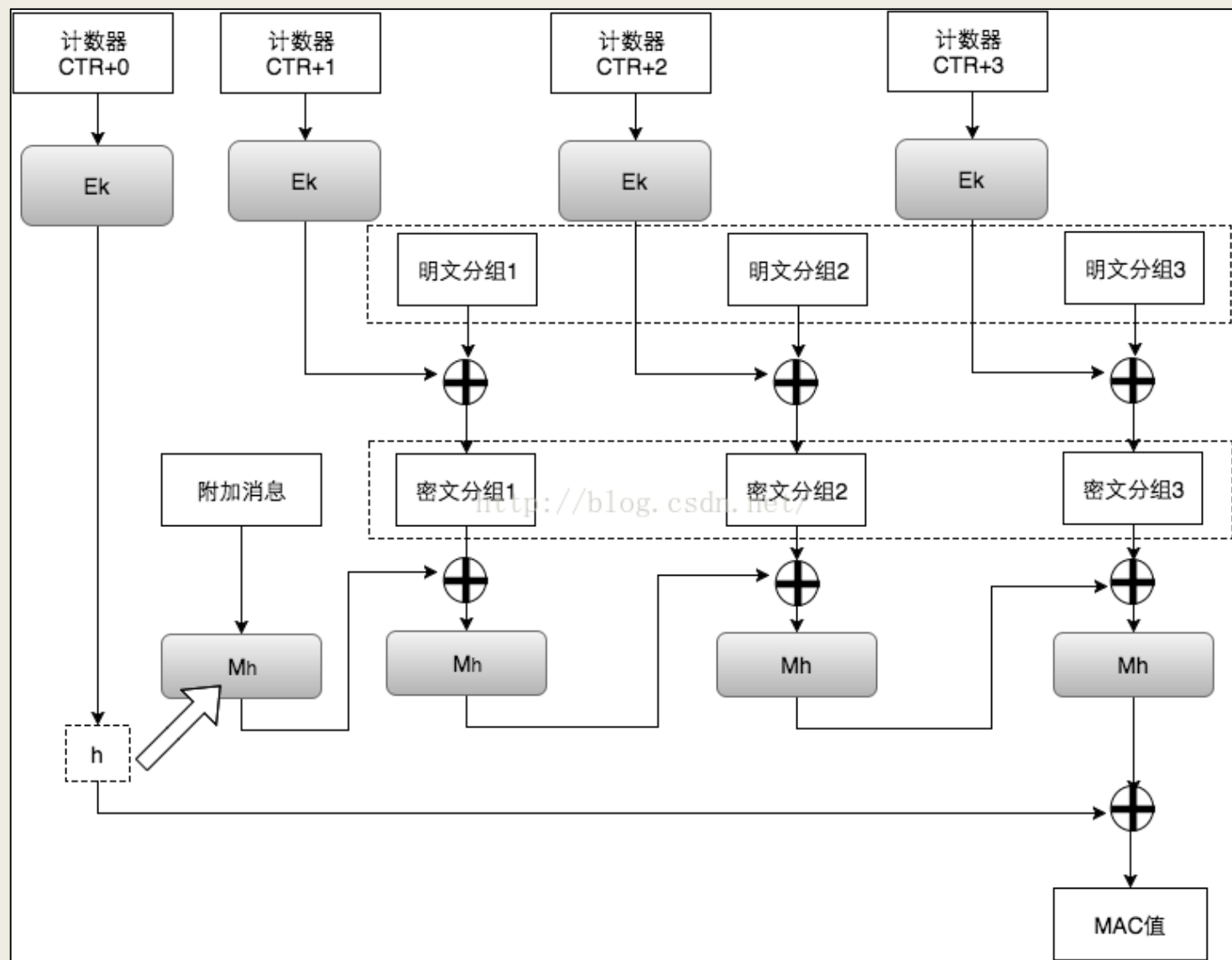


帶有關聯資料的認證加密 (authenticated encryption with associated data , AEAD)

- 實際上傳統的EtM , MtE , E&M都是有一些問題的 , 設計時要用真正的AEAD算法
- 常見的AEAD 算法
 - AES-128-GCM
 - AES-192-GCM
 - AES-256-GCM
 - ChaCha20-IETF-Poly1305
 - XChaCha20-IETF-Poly1305

Galois/Counter Mode (GCM)

- GCM可以提供對訊息的加密和完整性校驗，另外，它還可以提供附加訊息的完整性校驗。
- 在實際應用場景中，有些訊息是我們不需要保密，但訊息的接收者需要確認它的真實性的，例如源IP，源端口，目的IP，IV，等等。因此，我們可以將這一部分作為附加訊息加入到MAC值的計算當中



HW

- <https://cryptopals.com/>
- 寫第 4、8 題和簡單以外 1 題，共 3 題
- 簡單: 1 ~ 9、15、21、28、33
- 用 python
- 上傳 ZIP 檔，結構:
- [小寫學號]/
[題號]/
code/
writeup.pdf
- writeup 包含
 - 解題方法、思路
 - code做了什麼、怎麼操作