Lecture 5
Modern Symmetric-Key Encryption

Jason Lin

學習目標

- 介紹現代區塊加密法並討論其特性。
- 解釋為何要將現代區塊加密法設計成取代加密法。
- 介紹區塊加密法的組成元件，例如 P-box 和 S-box。
- 討論乘積加密法並區別兩類乘積加密法：Feistel 加密法和非 Feistel 加密法。
- 討論特別為現代區塊加密法而設計的兩種攻擊：差異破密分析和線性破密分析。

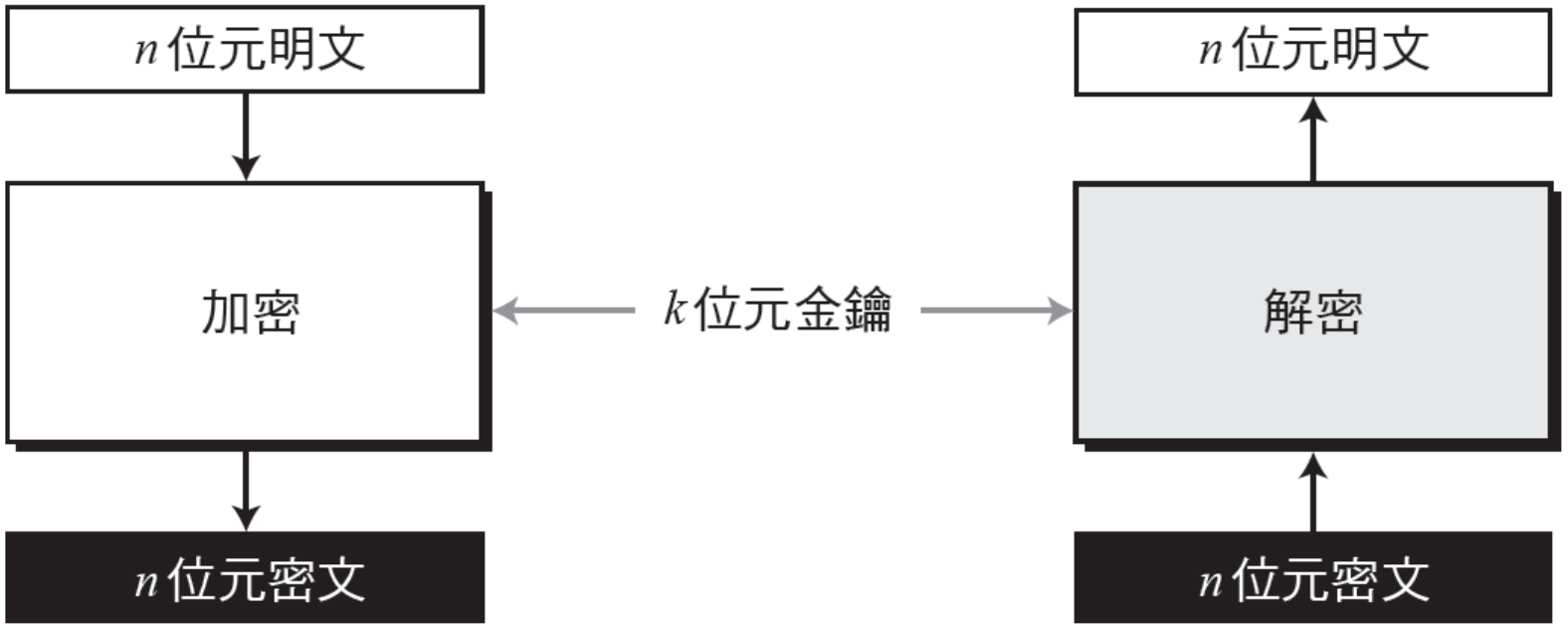
5.1 現代區塊加密法

- 一個對稱式金鑰現代區塊加密法（Modern Block Cipher）是加密一個 n 位元的明文區塊或解密一個 n 位元的密文區塊。
- 加密演算法或解密演算法使用一把 k 位元的金鑰。
- 適用於區塊加密的資料（一整段）
  - 範例：電子郵件、檔案傳輸

5.1 現代區塊加密法 (續)

- 本節討論的主題
  - 取代或換位
  - 區塊加密法形成排列群
  - 現代區塊加密法的組成要素
  - 乘積加密法
  - 兩種乘積加密法的類型
  - 區塊加密法的攻擊

圖 5.1 一個現代區塊加密法



範例 5.1

- 如果一個字元使用 8 位元的 ASCII 編碼，而區塊加密法接受 64 位元的區塊，則 100 個字元的訊息必須加入多少填塞位元呢？
- 解法：使用 8 位元的 ASCII 對 100 個字元編碼，將產生一個 800 位元的訊息，明文必須可以被 64 整除，假設 |M| 和 |Pad| 分別表示訊息的長度和填塞位元的長度

$$|M| + |Pad| = 0 \bmod 64 \quad \rightarrow \quad |Pad| = -800 \bmod 64 \quad \rightarrow \quad 32 \bmod 64$$

5.1.1 取代或換位

- 現代區塊加密法可以設計成取代（Substitution）加密法或者換位加密法（Transposition）。

注意

為了抵抗徹底搜尋攻擊（Exhaustive Search Attack），現代區塊加密法必須設計成取代加密法。

範例 5.2

- 假設有一個區塊加密法，其中 $n = 64$。如果密文中有 10 個位元 1，在下列情況下，Eve 需要做多少次的嘗試錯誤測試，才能從竊聽的密文取得明文？
  a. 加密法被設計成取代加密法。
  b. 加密法被設計成換位加密法。

範例 5.2 (續)

- 解法：
  a. 在第一種情況中（取代），因為 Eve 不知道明文中有多少個位元 1，所以她必須嘗試全部 $2^{64} - 1$ 個可能的 64 位元區塊，以找到其中有意義的一個。
  b. 在第二種情況中（換位），若 Eve 知道明文中正好有 10 個位元 1，因為換位不會改變密文裡 1 的數目，我們可以知道原本有 $C(64,10) - 1$ 種可能的明文。Eve 也可以只使用那些正好有 10 個位元 1 的 64 位元區塊，來進行徹底搜尋攻擊。

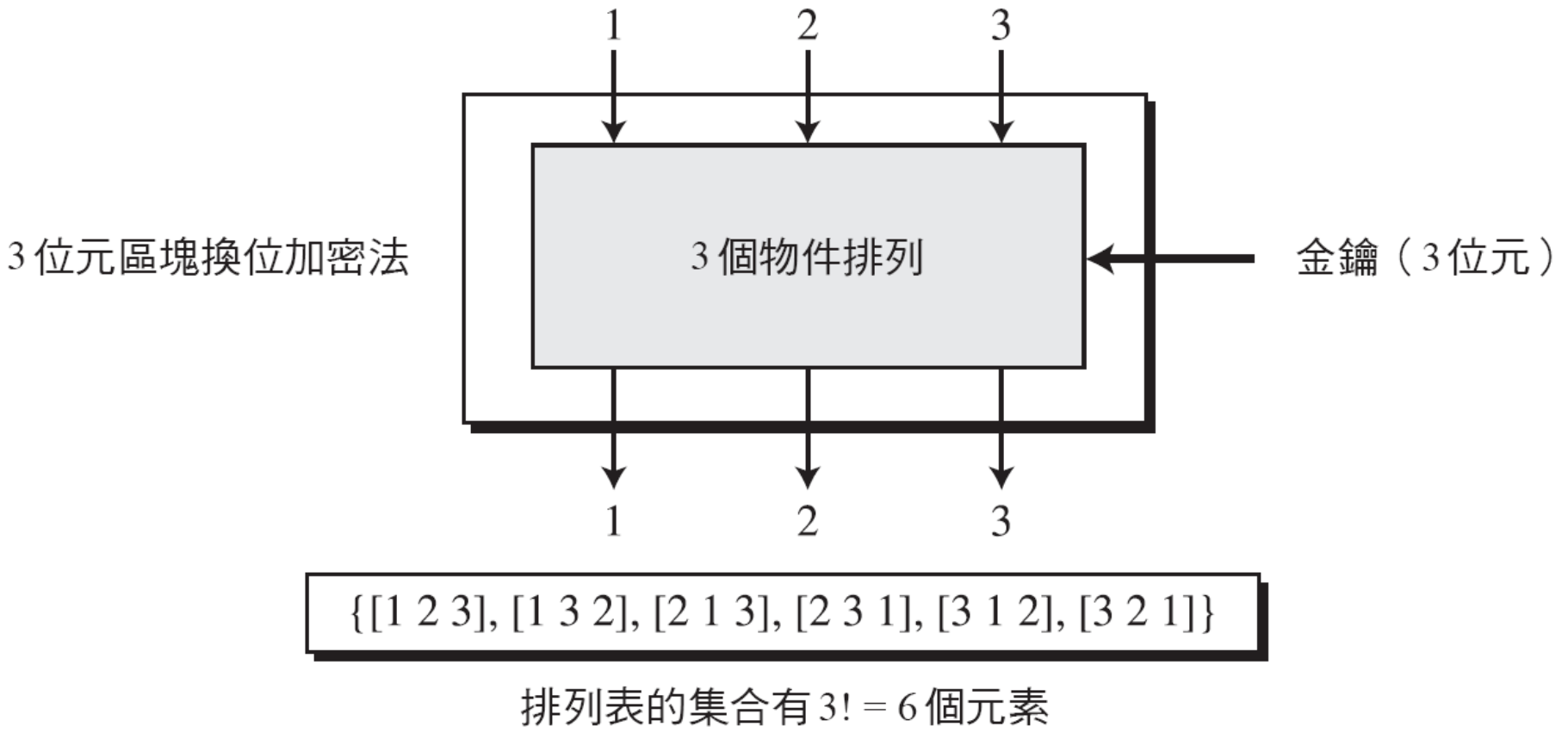
全大小金鑰換位區塊加密法

- 在一個全大小金鑰換位加密法，我們需要有 $n!$ 個可能的金鑰，因此金鑰應該有 $\lceil \log_2 n! \rceil$ 個位元。

範例 5.3

- 顯示一個 3 位元區塊換位加密法的模型和其排列表集合，其中區塊大小是 3 位元。
- 解法：排列表的集合共有 3! = 6 個元素，如圖 5.2 所示。

圖 5.2


1
2
3
3 位元區塊換位加密法
3 個物件排列
金鑰（3 位元）
1
2
3
{[1 2 3], [1 3 2], [2 1 3], [2 3 1], [3 1 2], [3 2 1]}
排列表的集合有 3! = 6 個元素

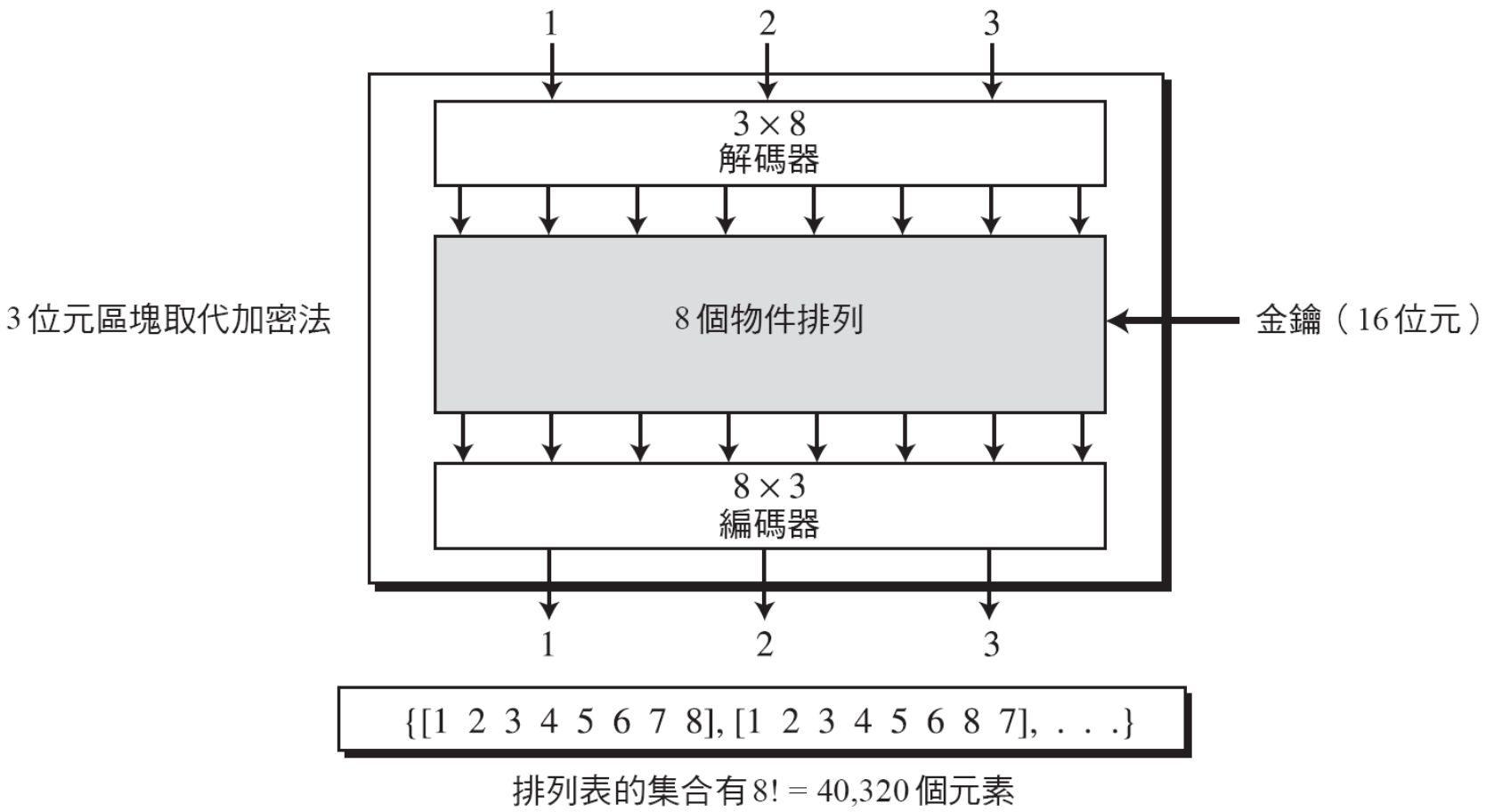
全大小金鑰取代區塊加密法

- 一個全大小金鑰取代加密法不調換位元而是取代位元，但如果能對輸入解碼並對輸出編碼，就能將取代加密法模型化成一種排列。

範例 5.4

- 顯示一個 3 位元區塊取代加密法的模型和其排列表集合。
- 解法：圖 5.3 顯示其模型和排列表集合，金鑰長度 $\lceil \log_2 40{,}320 \rceil = 16$ 位元也長很多。

圖 5.3 區塊取代加密法模型化成一種排列



全大小金鑰加密法

注意

一個全大小金鑰的 n 位元區塊換位或取代加密法可以被模型化成一種排列，但是它們的金鑰大小是不相同的：

對換位加密法而言，金鑰的長度是 $\lceil \log_2 n! \rceil$ 位元。

對取代加密法而言，金鑰的長度是 $\lceil \log_2 (2^n)! \rceil$ 位元。

部分大小金鑰加密法

注意

如果一個部分金鑰加密法是其相對應全大小金鑰加密法的一個子群，那麼這個部分金鑰加密法在組合運算下是一個群。

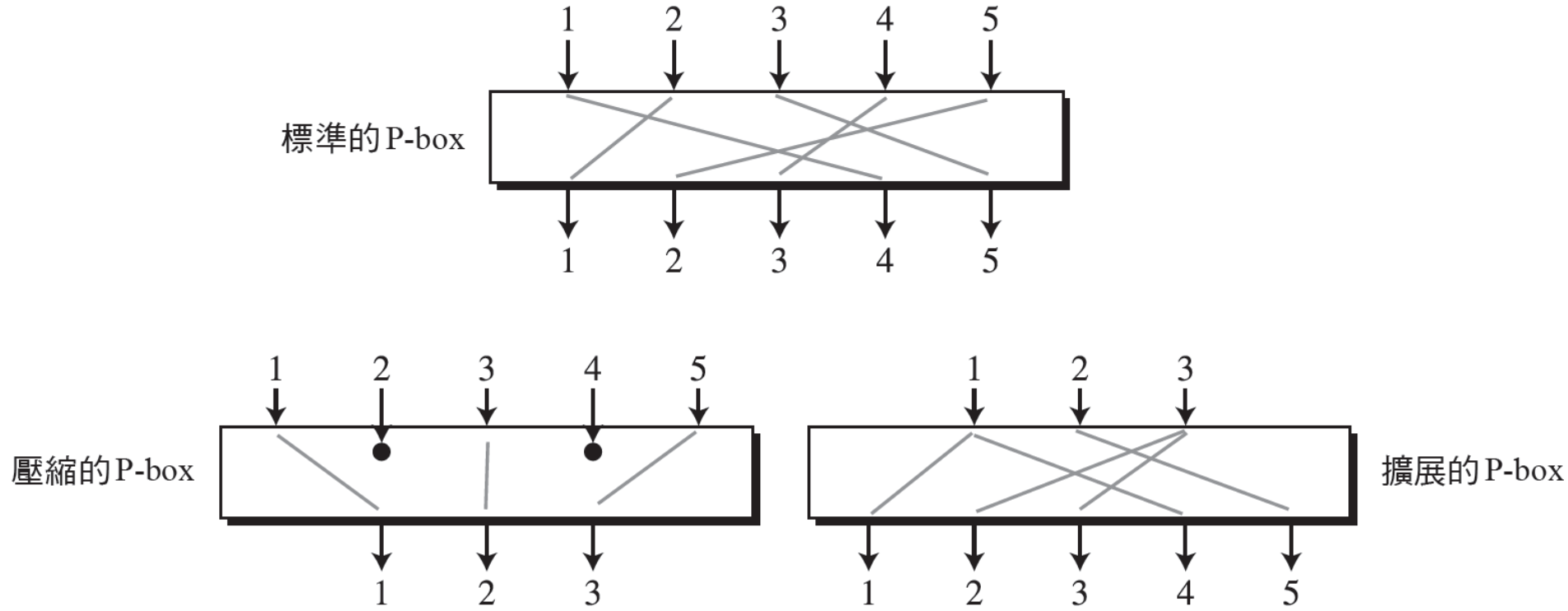
現代區塊加密法的組成要素

- 現代區塊加密法通常是有金鑰的取代加密法，其中金鑰只允許從可能的輸入到可能的輸出之部分對應。

P-box

- P-box（排列 box）類似於傳統的字元換位加密法，不同之處在於 P-box 的換位單元是位元。

圖 5.4 三種不同型態的 P-box



範例 5.5

- 圖 5.5 顯示一個 3 × 3 的標準的 P-box 之全部六種的可能對應。

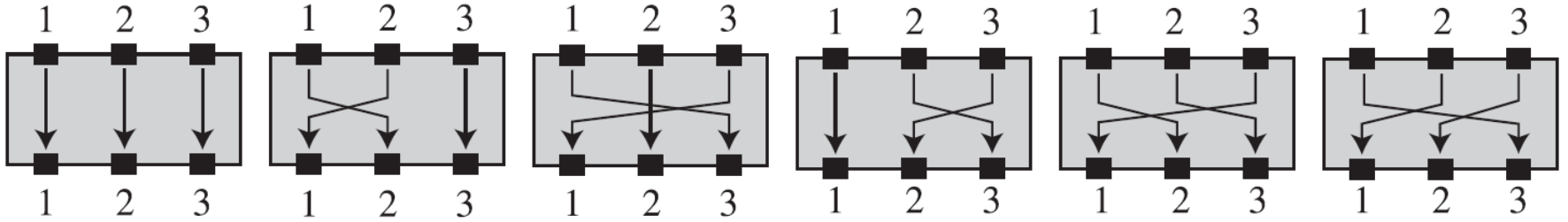


表 5.1 一個標準的 P-box 的排列表範例

58	50	42	34	26	18	10	02	60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06	64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01	59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05	63	55	47	39	31	23	15	07

範例 5.6

- 為一個標準的 P-box 來設計一個 8×8 排列表，將輸入字組的兩個中間位元（位元 4 和 5）移動至輸出字組的兩端兩個位元（位元 1 和 8），而其他位元彼此間的相對位置不變。
- 解法：此標準的 P-box 排列表是 [4 1 2 3 6 7 8 5]，輸入位元 1、2、3、6、7、8 彼此間的相對位置沒有改變，但是第 1 個輸出來自第 4 個輸入，且第 8 個輸出來自第 5 個輸入。

壓縮的 P-box

- 一個壓縮的 P-box（Compression P-box）是一個輸入為 n 且輸出為 m 的 P-box，其中 $n > m$ 。

01	02	03	21	22	26	27	28	29	13	14	17
18	19	20	04	05	06	10	11	12	30	31	32

擴展的 P-box

- 一個擴展的 P-box（Expansion P-box）是一個輸入為 n 且輸出為 m 的 P-box，其中 $n < m$。

01	09	10	11	12	01	02	03	03	04	05	06	07	08	09	12

P-box：可逆性

注意

一個標準的 P-box 是可逆的，但壓縮的 P-box 和擴展的 P-box 是不可逆的。

範例 5.7

- 圖 5.6 以一個一維的表格來顯示如何反轉一個排列表。

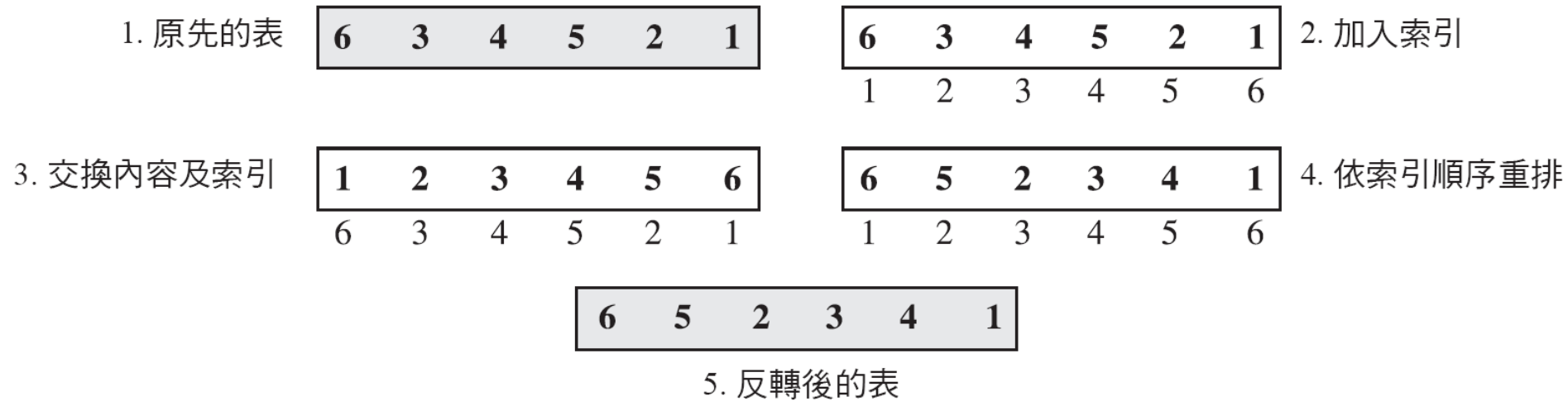
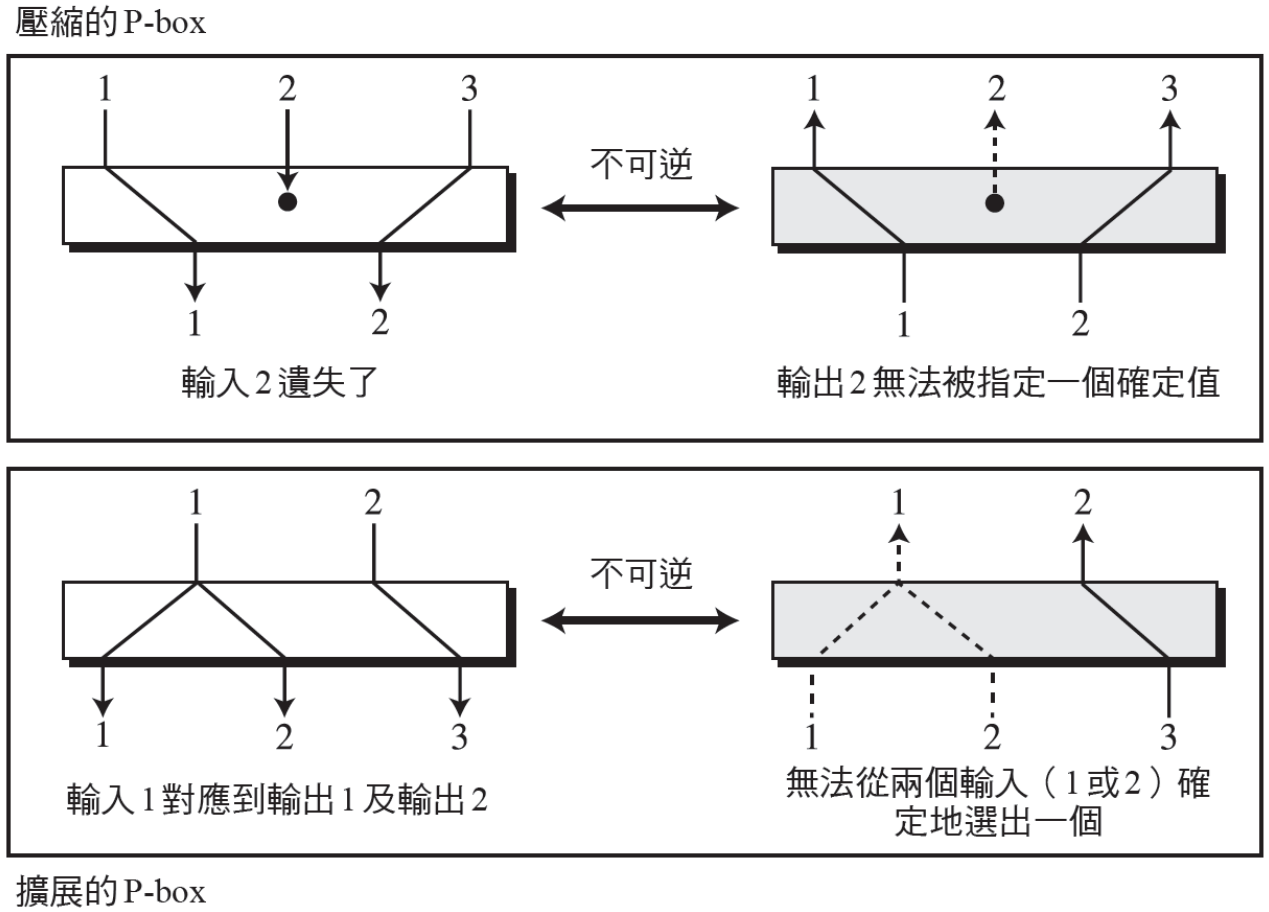


圖 5.7 壓縮和擴展的 P-box 是不可逆



S-box

- **S-box**（取代 box）可以視為一種小型的取代加密法。

注意

一個 S-box 是一個 $m \times n$ 的取代裝置，其中 m 和 n 不一定要相同。

範例 5.8

- 在一個具有 3 個輸入和 2 個輸出的 S-box 中，若輸入和輸出之間的關係是

$$y_1 = x_1 + x_2 + x_3 \qquad y_2 = x_1$$

則此 S-box 是線性的，因為 $a_{11} = a_{12} = a_{13} = a_{21} = 1$ 且 $a_{22} = a_{23} = 0$。此關係可用以下的矩陣來表示：

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

範例 5.9

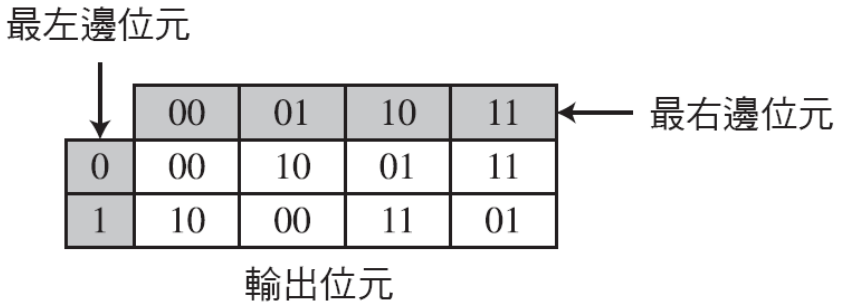
- 在一個具有 3 個輸入和 2 個輸出的 S-box 中，若輸入和輸出之間的關係是

$$y_1 = (x_1)^3 + x_2 \qquad y_2 = (x_1)^2 + x_1x_2 + x_3$$

其中，乘法和加法是在 **GF**(2) 裡的運算。此 S-box 是非線性的，因為輸入和輸出之間沒有線性關係。

範例 5.10

- 下表定義一個 3 × 2 S-box 的輸入／輸出關係，其中列表示輸入的最左邊位元，欄表 示輸入的最右邊兩個位元，而列與欄交叉處的值則是兩個位元的輸出值。



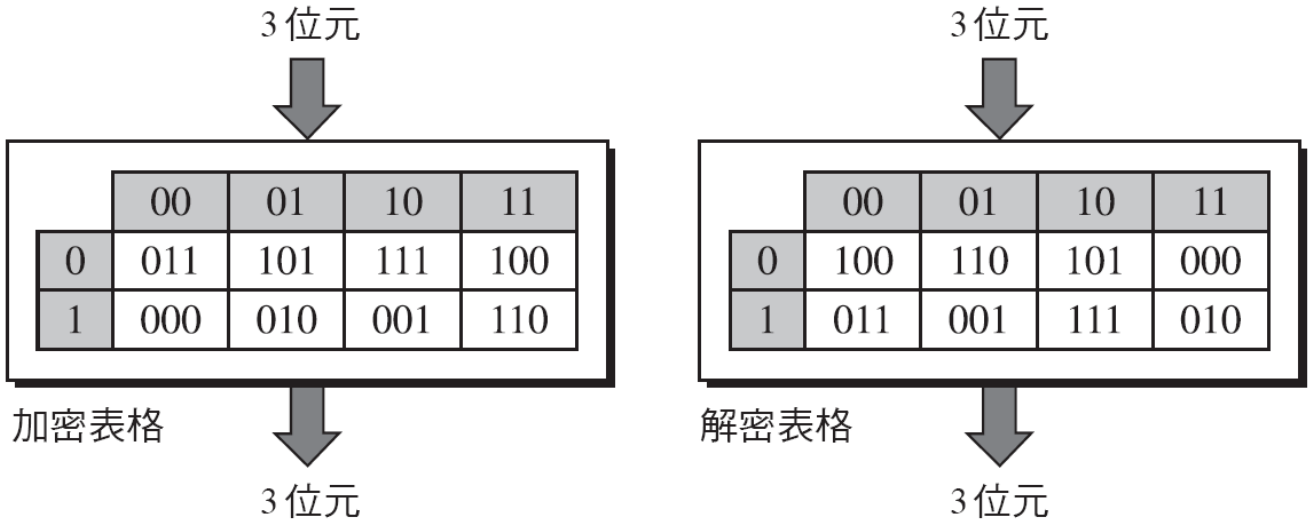
依此表格，010 的輸入會產生 01 的輸出，101 的輸入會產生 00 的輸出。

S-box：可逆性

- S-box 可以是可逆的或是不可逆的，可逆的 S-box 其輸入位元的數量必須與輸出位元的數量相同。

範例 5.11

- 圖 5.8 是一個可逆的 S-box 例子，如果左邊 S-box 的輸入是 001，則輸出是 101；而右邊表格的輸入是 101 時，將產生 001的輸出。這顯示了這兩個表格彼此互為反向。



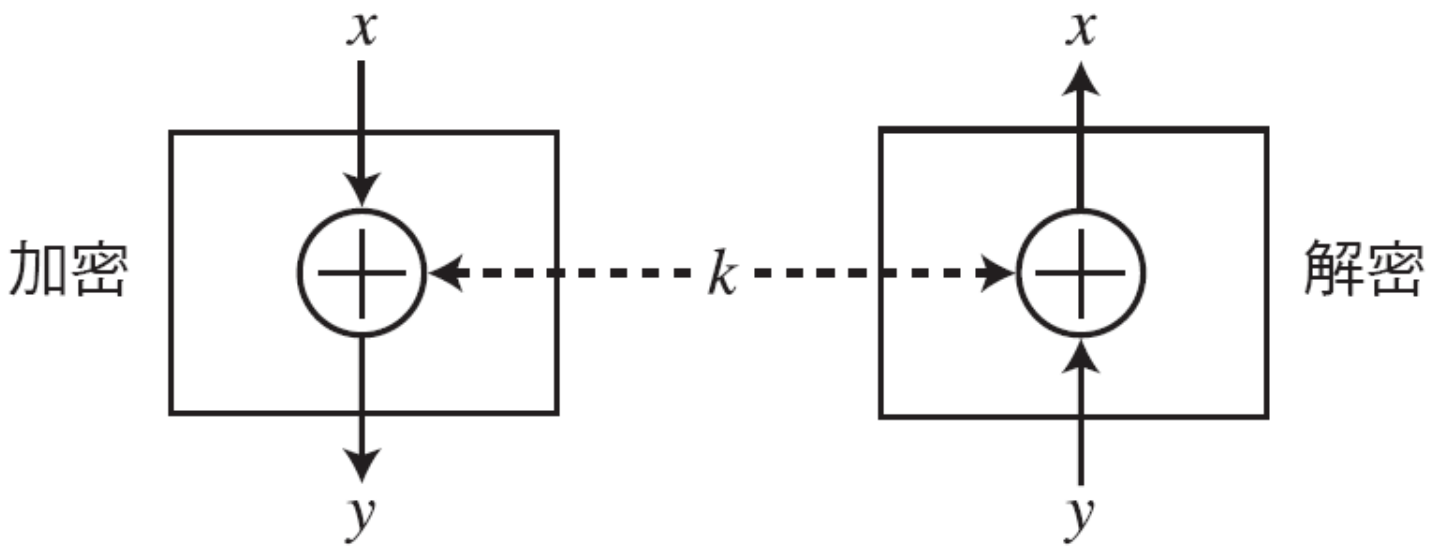
Exclusive OR

- 在大多數區塊加密法中，Exclusive OR（XOR）運算是非常重要的組成元件。我們在第四章曾討論過，在 $\boldsymbol{GF}(2^n)$ 中的加法和減法運算是透過稱為 XOR 的單一運算來執行的。
- XOR 運算在 $GF(2^n)$ 中的五種特性使得此運算是區塊加密法中非常有趣的組成元件：封閉性、結合性、交換性、存在單位元素、存在反元素。

XOR：反向

- 在一個加密法中，如果某個組成元件是代表單元運算（一個輸入和一個輸出），那麼此組成元件的反向是有意義的。例如，一個無金鑰的 P-box 或者一個無金鑰的 S-box 可以是可逆的，因為有一個輸入和一個輸出。一個 XOR 運算是一個單元運算，XOR 運算只有在輸入之一是固定的情況下（加密和解密時都一樣），其反向才會有意義。例如，如果輸入之一是金鑰，此金鑰通常在加密和解密時是一樣的，那麼 XOR 運算是自我可逆的，如下頁圖 5.9 所示。

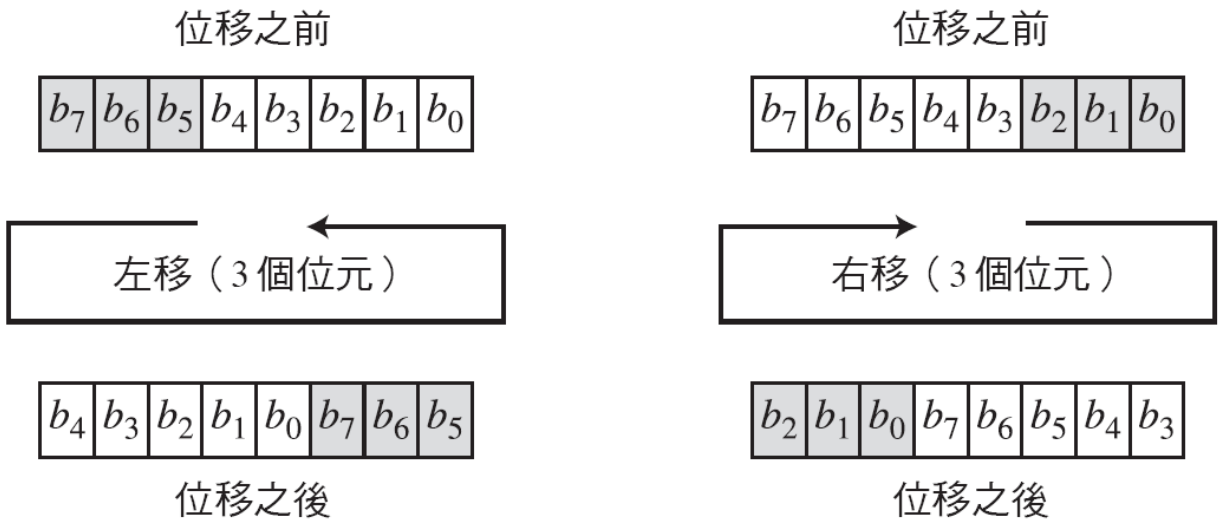
圖 5.9 Exclusive OR運算的可逆性



循環位移

- 一些現代區塊加密法也常使用循環位移運算（Circular Shift Operation）。

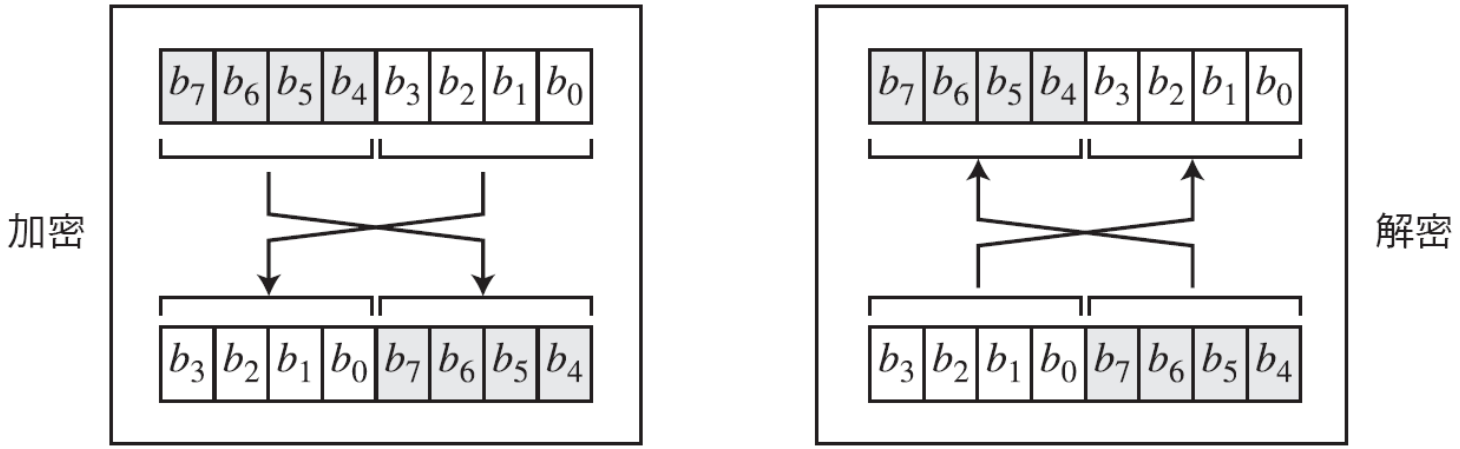
圖 5.10　向左或向右循環位移一個 8 位元字組



交換

- 交換運算（**Swap Operation**）是循環位移運算的特殊情況，其中 $k = n/2$。

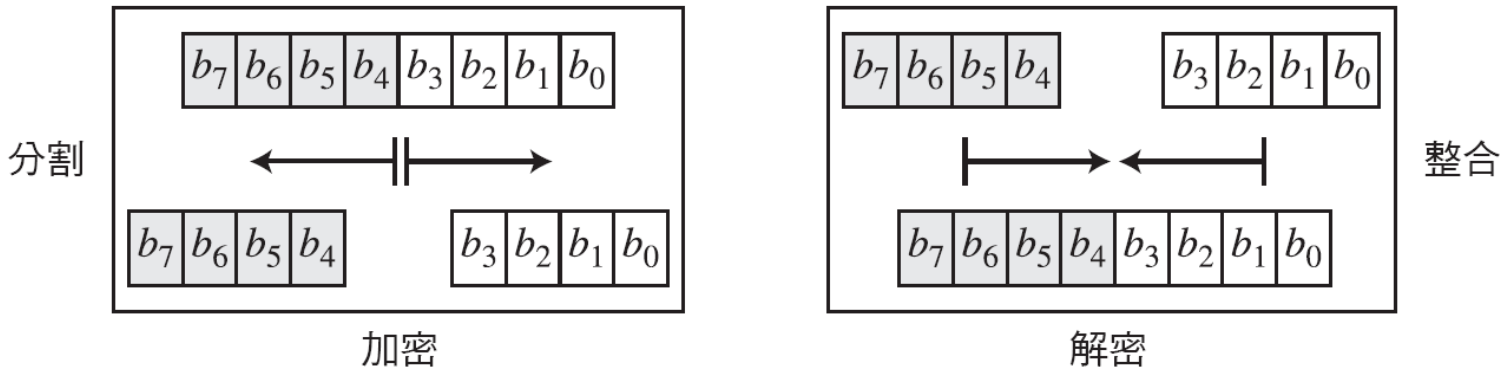
圖 5.11 一個 8 位元字組的交換運算



分割與整合

- 一些現代區塊加密法中其他兩個常見的運算是分割與整合。

圖 5.12 一個 8 位元字組的分割與整合運算



5.1.4 乘積加密法

- 乘積加密法（**Product Cipher**）的概念是由 Shannon 提出。乘積加密法是一種結合了取代、排列和前幾節所討論其他運算的複雜加密法。

擴散與混淆

- 擴散（Diffusion）

擴散隱藏密文和明文之間的關係。

- 混淆（Confusion）

混淆隱藏密文和金鑰之間的關係。

回合

- 擴散和混淆可以使用迭代的乘積加密法來達到，其中每一次的迭代是漂白（Whitening）、S-box、P-box 和其他組成元件的結合。

圖 5.13 一個兩回合的乘積加密法

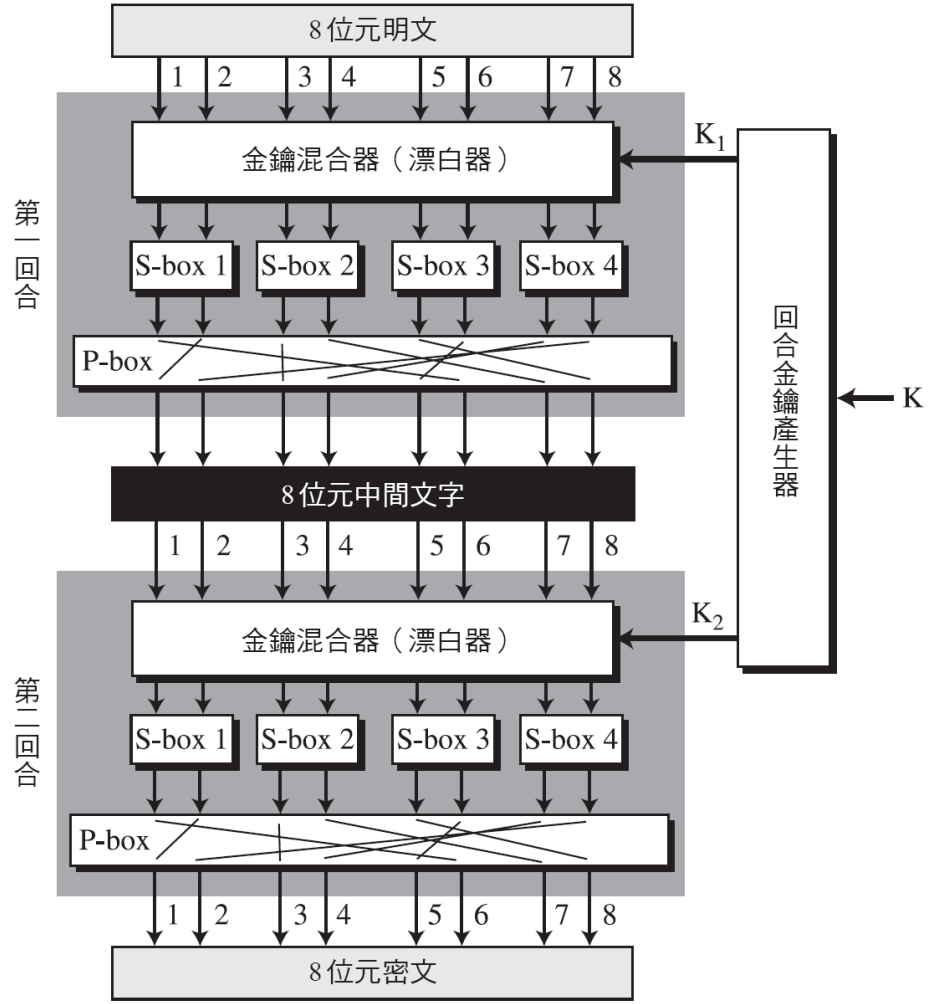
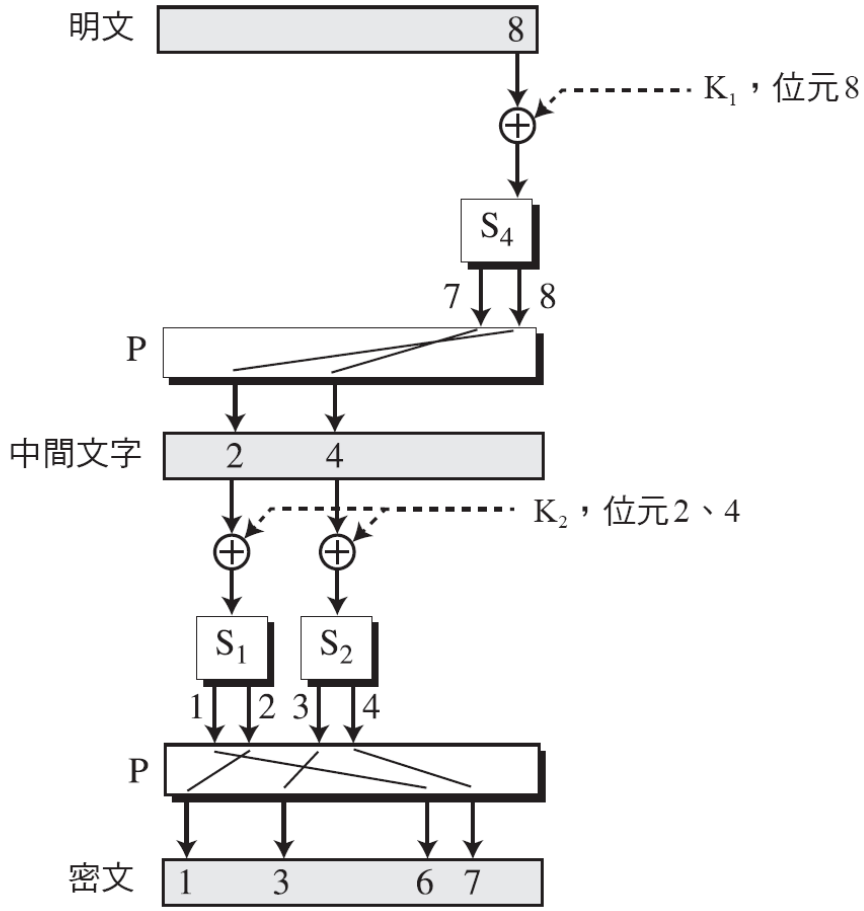
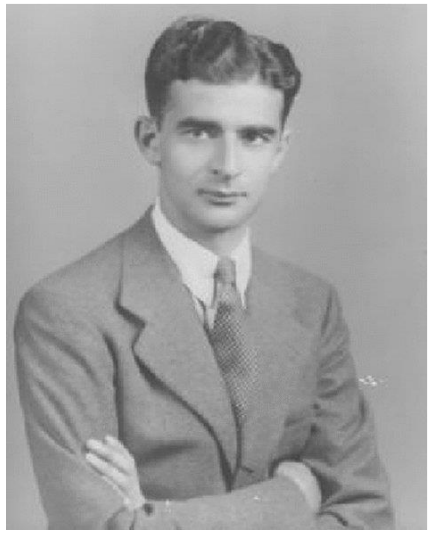


圖 5.14 區塊加密法的擴散和混淆



5.1.5 兩種乘積加密法的類型

- 現代區塊加密法都是乘積加密法，但是可分成兩種：
  - Feistel 加密法
  - 非 Feistel 加密法



Horst Feistel (1915 – 1990)

Feistel 加密法

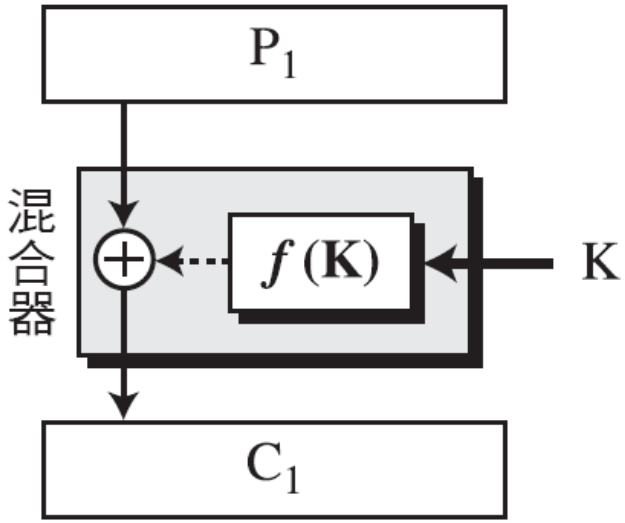
- Feistel 設計了一種非常聰明和有趣的加密法，用於構造區塊加密法的對稱結構，已經使用數十年了。
- **Feistel** 加密法（**Feistel Cipher**）有三種組成元件：自我可逆、可逆和不可逆的。

注意

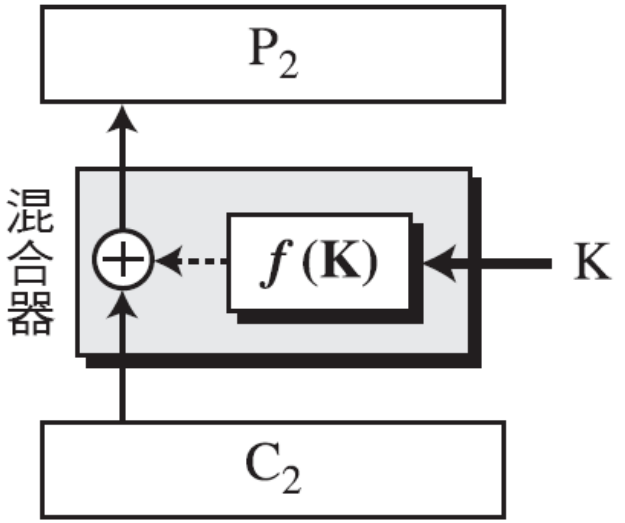
在 Feistel 設計裡的混合器是自我可逆的。

圖 5.15 Feistel 加密法設計的最初想法

- 混合器 (Mixer)：將不可逆元件結合在函數裡，依然可以反向。



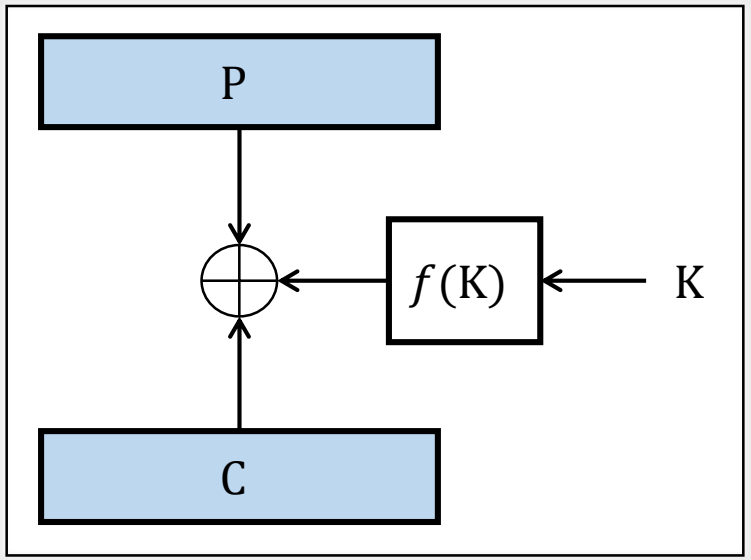
加密



解密

證明混合器的可逆性

- 設一個不可逆函數 $f(\mathrm{K})$
  - 加密：$\mathrm{C}_1 = \mathrm{P}_1 \oplus f(\mathrm{K})$
  - 解密：$\mathrm{C}_1 \oplus f(\mathrm{K}) = \mathrm{P}_1 \oplus f(\mathrm{K}) \oplus f(\mathrm{K}) = \mathrm{P}_1$



範例 5.12

- 這是一個簡單的範例。明文和密文的長度各是 4 位元，金鑰長度是 3 位元，假設此函數取金鑰的第一和第三位元，將此二位元解釋成十進位的數字，再將該數字平方，以二進位的 4 位元表示結果。如果原始的明文是 0111，而金鑰是 101，請顯示加密和解密的結果。

範例 5.12 (續)

- 解法：此函數取出第一和第三位元，得到二進位的 11 或十進位的 3，平方的結果是 9，以二進位表示則是 1001。

加密： $C = P \oplus f(K) = 0111 \oplus 1001 = 1110$

解密： $P = C \oplus f(K) = 1110 \oplus 1001 = 0111$　　與原始的 P 相同

圖 5.16 先前 Feistel 設計的改進

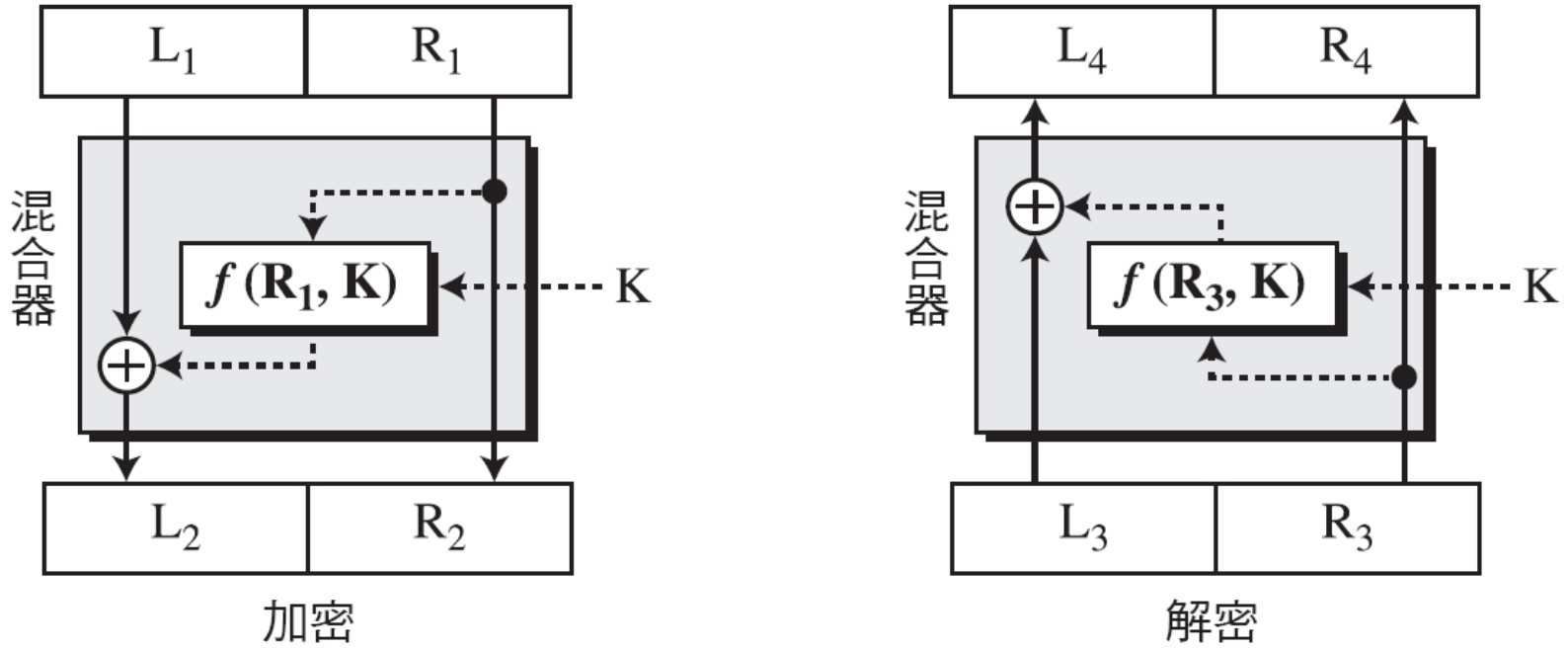
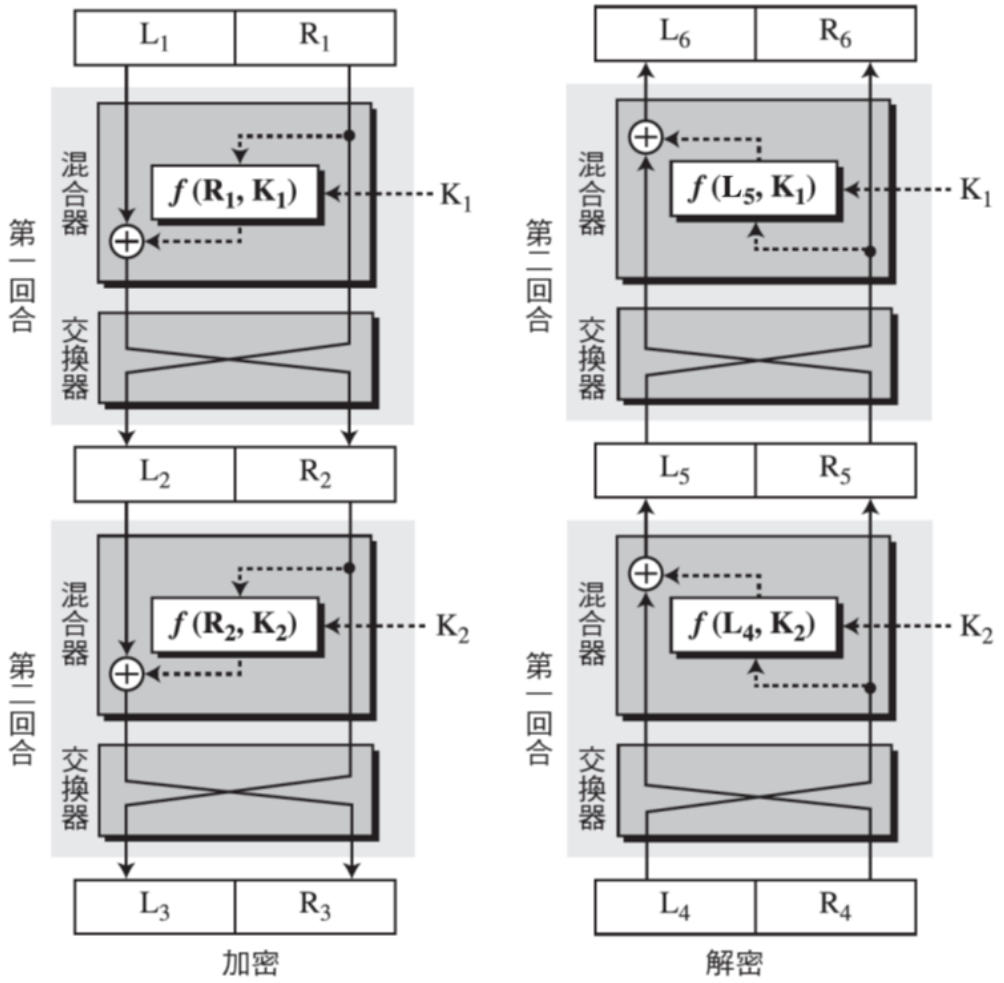


圖 5.17 兩回合的 Feistel 加密法最後設計



非 Feistel 加密

- 非 Feistel 加密法（Non-Feistel Cipher）只使用可逆的組成元件，明文的某個成分在密文中有相對應的成分。

5.1.6 區塊加密法的攻擊

- 對傳統加密法的攻擊也能用在現代區塊加密法上，但是第三章討論過現今的區塊加密法能抵抗大多數這類攻擊。
- 本節討論主題
  - 差異破密分析 (Differential Cryptanalysis)
  - 線性破密分析 (Linear Cryptanalysis)

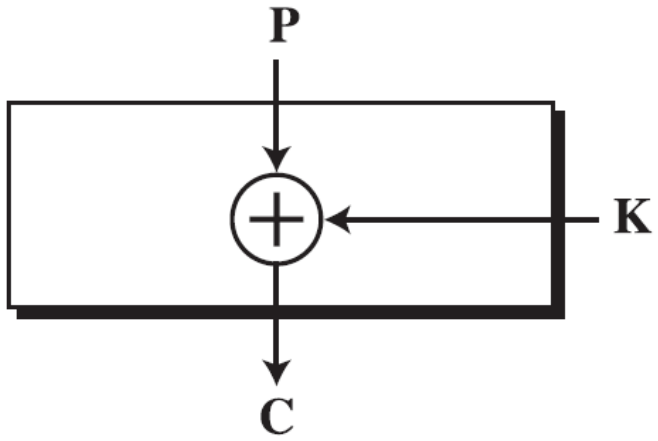
差異破密分析

- Eli Biham 和 Adi Shamir 提出差異破密分析（Differential Cryptanalysis）的想法，這是一種選擇明文攻擊。

範例 5.13

- 假設有一加密法只由一個 XOR 運算組成，如下圖所示，在不知道金鑰的情況之下，Eve 可以容易地找出明文差異與密文差異之間的關係，其中明文差異是指 $P_1 \oplus P_2$，而密文差異是指 $C_1 \oplus C_2$。以下證明 $C_1 \oplus C_2 = P_1 \oplus P_2$ ：

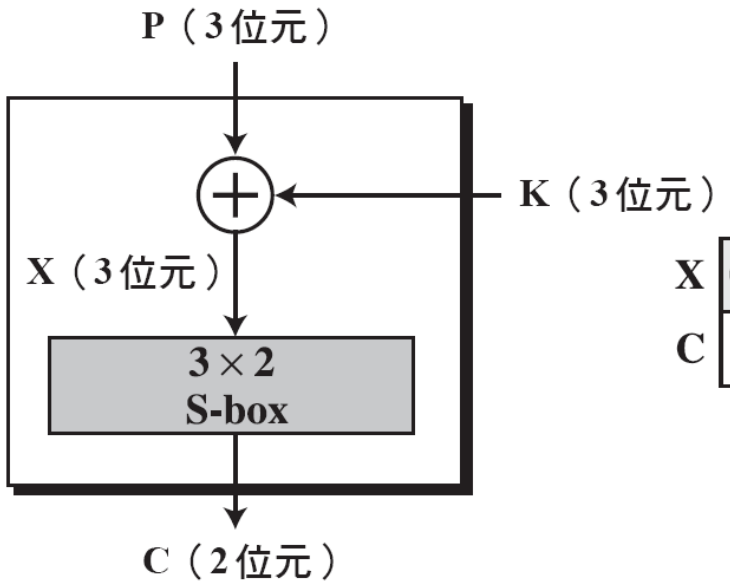
$$C_1 = P_1 \oplus K \quad C_2 = P_2 \oplus K \quad \rightarrow \quad C_1 \oplus C_2 = P_1 \oplus K \oplus P_2 \oplus K = P_1 \oplus P_2$$



範例 5.14

- 我們在範例 5.13 中新增一個 S-box，如圖 5.19 所示。

圖 5.19



X	000	001	010	011	100	101	110	111
C	11	00	10	10	01	00	11	00

S-box 表格

範例 5.14 (續)

- 此時 Eve 能建立一個機率的關係，如表 5.4 所示。

表5.4

$P_1 \oplus P_2$ \ $C_1 \oplus C_2$	*00*	*01*	*10*	*11*
000	**8**			
001	**2**	**2**		**4**
010	**2**	**2**	**4**	
011		**4**	**2**	**2**
100	**2**	**2**	**4**	
101		**4**	**2**	**2**
110	**4**		**2**	**2**
111			**2**	**6**

範例 5.15

- Eve 能從範例 5.14 的結果建立機率的資訊，如表 5.5 所示。

表5.5

$P_1 \oplus P_2$ \ $C_1 \oplus C_2$	*00*	*01*	*10*	*11*
000	**1**	**0**	**0**	**0**
001	**0.25**	**0.25**	**0**	**0.50**
010	**0.25**	**0.25**	**0.50**	**0**
011	**0**	**0.50**	**0.25**	**0.25**
100	**0.25**	**0.25**	**0.50**	**0**
101	**0**	**0.50**	**0.25**	**0.25**
110	**0.50**	**0**	**0.25**	**0.25**
111	**0**	**0**	**0.25**	**0.75**

範例 5.16

- 查看表 5.5，Eve 知道若 $P_1 \oplus P_2 = 001$，則 $C_1 \oplus C_2 = 11$ 的機率是 0.50（50%），她嘗試 $C_1 = 00$ 並且得到 $P_1 = 010$（選擇密文攻擊），她也嘗試 $C_2 = 11$ 並且得到 $P_2 = 011$（另一次選擇密文攻擊）。現在，根據這兩對 P 和 C，她試著往回推，

$$C_1 = 00 \rightarrow X_1 = 001 \text{ 或 } X_1 = 101 \text{ 或 } X_1 = 111$$

$$\begin{cases} \text{若 } X_1 = 001 \rightarrow K = X_1 \oplus P_1 = 011 \\ \text{若 } X_1 = 101 \rightarrow K = X_1 \oplus P_1 = 111 \\ \text{若 } X_1 = 111 \rightarrow K = X_1 \oplus P_1 = 101 \end{cases}$$

$$C_2 = 11 \rightarrow X_2 = 000 \text{ 或 } X_2 = 110$$

$$\begin{cases} \text{若 } X_2 = 000 \rightarrow K = X_2 \oplus P_2 = 011 \\ \text{若 } X_2 = 110 \rightarrow K = X_2 \oplus P_2 = 101 \end{cases}$$

這兩次的試驗確認 $K = 011$ 或 $K = 101$。

差異破密分析 (續)

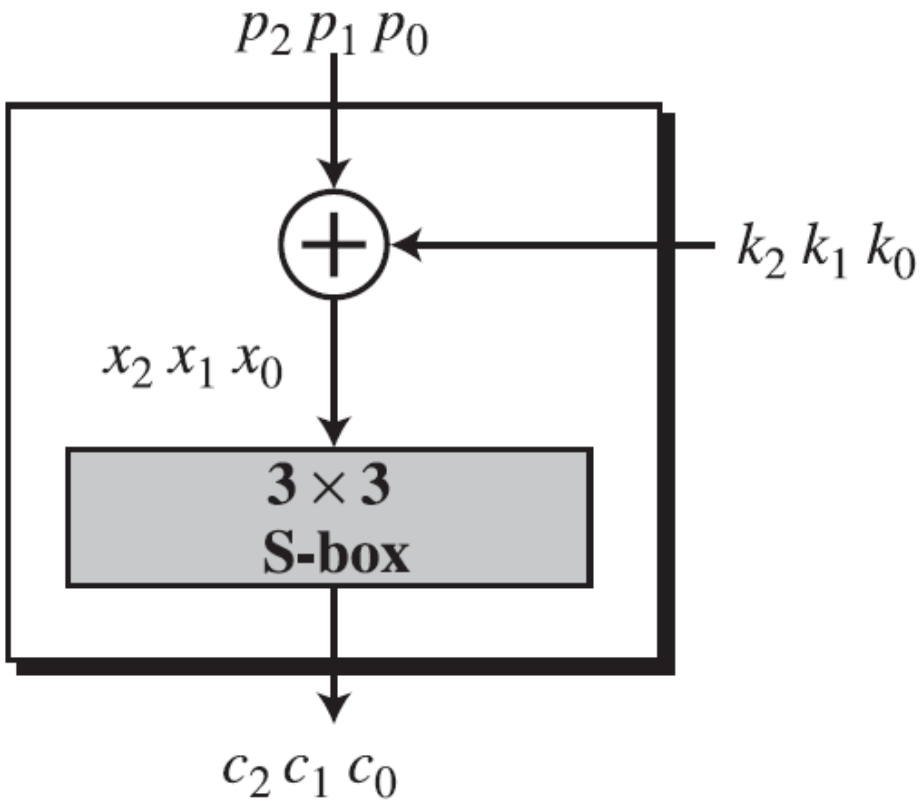
注意

差異破密分析是基於區塊加密法中 S-box 的不平均差異分布表。

線性破密分析

- 線性破密分析（Linear Cryptanalysis）是 1993 年由日本密碼學家 Mitsuru Matsui 所提出，此分析使用已知明文攻擊
- Mitsuru Matsui 也是第一位公開報告對加密標準 DES 進行實驗性破密分析的人，他利用 12 台工作站在 50 天的時間裡進行計算

圖 5.20 具一個線性 S-box 的簡單加密法



$$c_0 = x_0 \oplus x_1$$
$$c_1 = x_0 \oplus x_1 \oplus x_2$$
$$c_2 = x_1 \oplus x_2$$

S-box

線性破密分析 (續)

- 求解三個未知數，我們得到

$$c_0 = p_0 \oplus k_0 \oplus p_1 \oplus k_1$$
$$c_1 = p_0 \oplus k_0 \oplus p_1 \oplus k_1 \oplus p_2 \oplus k_2$$
$$c_2 = p_1 \oplus k_1 \oplus p_2 \oplus k_2$$

- 這意味著三個已知明文攻擊能找出 k_0、k_1 和 k_2。

$$k_1 = (p_1) \oplus (c_0 \oplus c_1 \oplus c_2)$$
$$k_2 = (p_2) \oplus (c_0 \oplus c_1)$$
$$k_0 = (p_0) \oplus (c_1 \oplus c_2)$$

線性相近

- 在一些現代區塊加密法中，可能發生的情況是某些 S-box 並不是完全非線性，而是藉由一些線性函數機率式地近似於非線性

$$(k_0 \oplus k_1 \oplus \cdots \oplus k_x) = (p_0 \oplus p_1 \oplus \cdots \oplus p_y) \oplus (c_0 \oplus c_1 \oplus \cdots \oplus c_z)$$

其中，$1 \leq x \leq m$、$1 \leq y \leq n$ 且 $1 \leq z \leq n$