進階加密標準加密強度之研究

A study on encryption strength of advanced encryption standard

林聰武
東吳大學資訊管理學系
twlin@csim.scu.edu.tw

郇智喬
東吳大學資訊管理學系
johnn0071983@gmail.com

摘要—隨著網際網路的普及，資訊的傳遞速度與便利性也跟著大幅提昇，因此在傳遞敏感或機密資訊時，為了防止資訊被攔截外洩，可以對檔案進行加密，現在主流的加密方法分為許多種類，本研究使用了進階加密標準分別在改變金鑰與明文的狀態下，使用不同的加密模式加密各種常見類型檔案，觀察其密文的變化量，研究結果顯示，在 ECB 區塊加密模式下，因為檔案格式的關係，在變化量上會有較為明顯的差異，因此在加密上建議採用 CBC、CFB、OFB 這三種區塊加密模式進行加密。

關鍵詞—進階加密標準、加密強度

Abstract—With the popularity of the Internet, information transmission speed and convenience has been a substantial increase. In the transmission of sensitive or confidential information, you can encrypt them in order to prevent the leakage or intercept of the information. There are many kind of mainstream encryption functions. In our study, the Advanced Encryption Standard(AES) is used to encrypt several common file types by changing the encryption key or plaintext in different encryption modes to examine its encryption strength. The comparisons of ciphertexts have significant differences in the ECB encryption mode for some file types. Therefore, we suggest that the CBC, CFB, OFB encryption modes are suitable for the AES method.

***Ketword*—Advanced Encryption Standard、Encryption Strength**

一、前言

最近幾年來網際網路的迅速普及，使得資訊傳遞的速度與便利性也跟著大幅度的上升，然而隨著這些科技的進步，透過網際網路傳遞的個人資料安全性與隱私性就更值得被注意。

在這個議題上，如果我們要主動的對資料進行保護的話，可以透過密碼學對資料檔案進行加密。所謂的加密是指把可以理解的明文（plaintext）運用一連串的規則改變成無法理解的密文（ciphertext），其中置換（transposition）[17]與代換（substitution）[17]這兩種方法是較為簡單的轉換方法，置換是把字母的順序照一定的規則重新排列，舉例來說把字母兩兩對換位置就是一種方法，而代換則是把字母用別的字母替換掉，例如每個字母都往後位移一個字，即所有的 a 都用 b 來表示，其他字母依此類推。

但傳統的這兩種加密方法非常容易被破解，最先被提出來的方法是運用統計資料對每個字母的出現頻率來分析密文，進而破解密文得到隱藏的明文，這個就是被稱做頻率分析[8, 17]的方法，簡單的代換加密方式都可以使用頻率分析來破解。也因此義大利學者Leon Battista Alberti 發明了一種多字元加密法[5, 17]，這種加密的原理是對不同的訊息片段使用不同的代換方式，來避免被頻率分析成功解碼，但這個方法後來在十

九世紀時被英國學者Charles Baggage發現還是無法完全避開流量分析的破解[4]。

在近代由於電腦與電子學的發展使得密碼學進入另一個階段，不只是文字可以進行加密，只要是電子化的檔案不管是文字或圖片等資料都可以進行加密，不過電腦的發展也同樣的讓解密速度可以更上一層樓，所以一個良好的加密方法除了要有效率外，還必須要能有效的延長被破解的時間。

西元1976年時，美國國家標準局（NIST）制訂出了資料加密標準（Data Encryption Standard）[6]，從此之後密碼學開始快速的發展，而RSA加密方法也是在這個時期被發表出來，RSA目前是一種只要金鑰長度足夠就無法被有效破解的加密方法（表2）[2,17]，同時RSA也是一種非對稱金鑰加密的方式，相對於傳統對稱金鑰加密使用同一把金鑰加密與解密，RSA的加密方與解密方使用了不同的金鑰。

現在的密碼學另一個要考慮的問題是，當電腦計算速度越來越快時，在短時間內用暴力演算法[3,9,10]來解開加密的可能性就會增加，像是目前相當熱門的量子電腦，未來要是順利研發出來運算速度將可以大幅度提升，屆時勢必要有新的加密演算法來對抗。

檔案加密的保護性是會受到很多因素的影響，例如加密方法的選擇、金鑰的長短、電腦的計算速度、密文與金鑰的傳遞方式等原因都可能造成破解加密速度的增快。

本研究是以美國國家標準局在 2002 年決定用來取代資料加密標準的進階加密標準[7]（Advanced Encryption Standard，也稱做 Rijndael 加密法）為加密函式，並使用五種不同類型加密模式（ECB、CBC、CFB、OFB、CTR）進行加密，以常見的六種檔案類型（txt、doc、pdf、jpg、zip、exe）來進行進階加密標準加密強度的檢驗。

本文在架構上分為五個章節，第一章是描述本研究的研究背景與研究動機，並說明研究目的、文章架構與研究流程。第二章一開始會先簡單介紹一些加密的方法，之後會再介紹加密所使用的各種加密模式。第三章說明了研究的方法以及研究假設。第四章主要為透過各種檔案的實驗數據，使用單因子變異數分析來檢定研究假設正確與否。第五章主要是提出研究結論與後續研究建議。

二、相關文獻

現代的密碼學可分為對稱金鑰以及非對稱金鑰這兩個部分，其中對稱金鑰加密又可以分為區塊加密（block cipher）與串流加密（stream cipher）這兩種，這些加密方法各有其特性。除此之外加密的模式也各有不同，以下將稍微介紹一下這些加密方法

2.1 資料加密標準

1976年時美國國家標準局決定的資料加密標準[6]是使用對稱金鑰加密法，對稱金鑰加密法還可以分為區塊加密（block cipher）與串流加密（stream cipher）等幾種，資料加密標準就是區塊加密中的一種，加密時會先把輸入的資料切成相同大小，如果最後一個區塊長度不夠則補足，輸出的密文則是與輸入的區塊大小相同，以下解說一下加解密流程。

首先先將輸入資料從頭切成64bits的區塊大小，經過初始排列後切成左右各32bits兩部分（L0,R0），之後把R0與子鑰匙（subkey）丟進F函數（F function）可以得到下一回合運算所需的L1，另外把L0與L1進行XOR運算可以得到R1，如此進行16回合的運算後，把得到的R16,L16進行反轉初始排列後即可得到密文。

其中的 F 函數包含擴充函式、S-box 與 P-box 這幾個部分，擴充函式是用來把輸入的 Rn 從 32bits 轉換成 48bits 以便與子鑰匙進行 XOR 運算，之後每 6bits 一組使用 S1 到 S8 共八個 S-box 轉換回 32bits，最後經過 P-box 的重新排序後就可以

得到要與 Ln 進行 XOR 運算的 32bits 輸出。

在金鑰的部分雖然有 64bits 的輸入，但每個 byte 的同位檢查碼（parity bit）並不會在加密時所用到，也就是說其實金鑰的長度只有 56bits，子鑰匙的產生是由原始金鑰重新排列後分成兩部分、向左位移與排列選擇所產生的，而每一回合的向左位移量則都有規定[6]。

2.2 三重資料加密標準

三重資料加密標準[6]的出現是由於資料加密標準中的金鑰長度太短（56bits），如果使用暴力破解法可以輕易的在短時間內破解加密，三重資料加密原理與原始的加密方式相同，只是對加密後的密文用不同的金鑰再進行加密（或解密）如此重複三次，如此一來金鑰的長度就等於加長了，三重加密時可以使用兩或三把金鑰，使用兩把金鑰表示第一與第三把金鑰相同，如果全部使用相同金鑰的話就相當於資料加密標準，只是徒增運算時間。

2.3 進階加密標準

進階加密標準[7]是美國國家標準局用來取代資料加密標準[6]的一個新加密方法，同樣的也是區塊加密的演算法，在區塊長度的部分則是增加為 128bits，金鑰長度有三種長度可以選擇使用，分別為 128bits、192bits 和 256bits，進階加密標準是使用置換組合方式進行加密，可以分為四個函式 SubBytes()、ShiftRows()、MixColumns()及 AddRoundKey()，加密時是在一個 4x4 大小的位元組矩陣上運算，而加密的回合數則按照金鑰的長度有所不同，以下是加密流程與簡單的函式功能介紹。

SubBytes：使用非線性的代換方式（S-box）代換矩陣內容。

ShiftRows：將矩陣中每一列進行向左的循環位移，由上至下分別位移0、1、2、3個位元組。

MixColumns：使用線性的方式轉換每一行的元素。

AddRoundKey：與子密鑰進行 XOR 運算，子密鑰則是經由 Key Schedule 所產生。

2.3 區塊加密模式

在區塊加密法中要加密的檔案通常不會只有一個區塊，於是區塊與區塊之間就可以與加密產生一些關連，以下介紹一些比較常見的區塊加密模式[12,17]。

2.3.1 Electronic CodeBook（ECB）

最簡單的一種加密模式，區塊與下一個區塊間並無關連性，加密後的密文也都互相獨立。（圖 1）

2.3.2 Cipher-Block Chaining（CBC）

第二種加密模式是 cipher-block chaining，這種加密模式會把先把明文區塊與上一個區塊加密後產生的密文做 XOR 運算，之後再丟進加密函式進行加密，第一個區塊因為沒有前一個區塊密文可以使用，所以預設了一個初始向量（initialization vector,IV）與其做 XOR 運算。（圖 2）

2.3.3 Cipher Feedback（CFB）

在 cipher feedback 模式中，同樣也需要一個初始向量（n bits）來做運算，一開始時先由初始向量與金鑰進行加密演算法，之後把得到的輸出值取前 s 個 bits 與相同大小的明文做 XRO 運算的到密文，之後把初始向量向左位移 s bits 並且把得到的密文接在後面成為另一個區塊加密的輸入值，因此 s 的大小為 $1 \leqq s \leqq n$，較常用的是 s=8 也就是每次加密一個位元組，如此可以避免 s 太大時浪費區塊空間，太小時浪費計算時間。（圖 3）

2.3.4 Output Feedback（OFB）

Output Feedback 與上一種 Cipher Feedback 非常相似，這兩種模式唯一差別是在輸出給下一個輸入值的那個 s bits，Output Feedback 模式中是在與明文進行 XOR 之前的值，並不是之後得

到的密文。（圖 4）

2.3.5 Counter（CTR）

最後一種模式是與 ECB 模式較為類似，每個區塊的加密都互相獨立，不同的是這種模式是透過一個計數器，每次加密完一個區塊就增加一，而與金鑰進行加密的是計數器的數值並不是明文本身，得到的輸出值與明文區塊作 XOR 後得到的才是密文。（圖 5）

三、研究方法

在本研究中為了公平的觀察密文變化量，因此在比較差異的部分是把原始的密文（金鑰、明文無改變）與改變過後的密文（金鑰或明文改變）以區塊大小（AES：128 位元）為單位計算區塊中的位元變化量，以每個檔案為一組，進行單因子變異數分析以瞭解各組間是否有顯著的差異，以下敘述本研究之假設。

本研究將各別在改變金鑰（圖 6）和改變明文（圖 7）下，觀察各種加密模式加密後的密文變化量有無顯著差異，由於 CTR 區塊加密模式的特性關係，在改變明文時只使用其餘四種加密模式進行比較。

H_1: 使用AES加密，在相同的金鑰改變下，五種加密模式的密文變化量無明顯差異。

H_2: 使用AES加密，在相同的明文改變下，四種加密模式的密文變化量無明顯差異。

接著，我們想知道在使用三重資料加密標準加密檔案的情況下，是否與進階加密標準的結果相似，因此採用與 H_3、H_4 相同的研究假設，只是加密函式改成使用三重資料加密標準加密。

H_3: 相同的金鑰改變下，使用三重DES加密，五種加密模式的密文變化量無明顯差異。

H_4: 相同的明文改變下，使用三重DES加密，四種加密模式的密文變化量無明顯差異。

最後我們在 ECB 與 CBC 區塊加密模式下，觀察使用進階加密標準與三重資料加密標準加密後密文的變化量是否有所差異，由於兩者區塊大小與金鑰長度皆不同，為了方便觀察以三重加密標準為準，分別觀察至 168 位元（金鑰）與 64 位元（明文）的改變量，而由於進階加密標準區塊長度較大，故每區塊所測得的密文變化量皆縮小為一半後進行單因子變異數分析以茲平衡，研究假設如下。

H_5: 使用*ECB*模式在相同的金鑰改變下，AES與3-DES的密文變化量無明顯不同。

H_6: 使用*ECB*模式在相同的明文改變下，AES與3-DES的密文變化量無明顯不同。

H_7: 使用*CBC*模式在相同的金鑰改變下，AES與3-DES的密文變化量無明顯不同。

H_8: 使用*CBC*模式在相同的明文改變下，AES與3-DES的密文變化量無明顯不同。

四、研究結果

4.1 實驗環境與架構

本研究使用 C 語言撰寫程式，並使用以 C 語言所寫成的公開原始碼的 Rijndeal（AES）演算法與資料加密標準（DES）演算法 ，以 Microsoft Excel 內提供的單因子變異數分析，在實驗方面選用六種常見的檔案格式，所使用的金鑰的長度為最高等級的 256 位元與 168 位元，分別對 H_1 到 H_8 的研究假設進行驗證（在 α=0.05 之下）。

透過以下的實驗數據我們可以知道對 jpg 類型的檔案而言，這五種區塊加密模式在改變金鑰中，對密文的位元改變量並無顯著的影響，因此無法拒絕 H_1 的研究假設。

表 33. H_1 研究假設的變異數分析

改變位元數	1	2	3	4
F 值	0.937	2.233	0.796	1.057
H_1	接受	接受	接受	接受
…				
改變位元數	253	254	255	256
F 值	1.303	1.794	0.882	0.645
H_1	接受	接受	接受	接受

在改變明文的部分，因為 CTR 加密模式較為特殊，密文會與明文變化量成正向關係，若參

與比較必定會拒絕H_2的研究假設，只有在明文改變量為 64 位元（接近每區塊密文改變的平均值）時為非顯著，在剔除 CTR 加密模式後的結果如下表，皆為接受H_2的研究假設。

表 33. H_2研究假設的變異數分析

改變位元數	1	2	3	4
F 值	0.534	0.656	1.675	0.107
H_2	接受	接受	接受	接受
…				
改變位元數	125	126	127	128
F 值	2.169	2.366	2.107	2.447
H_2	接受	接受	接受	接受

同樣的在資料加密標準加密檔案可以觀察到（表 35、36），對於H_3、H_4的研究假設同樣為接受。

表 33. H_3研究假設的變異數分析

改變位元數	1	2	3	4
F 值	0.607	0.319	0.213	0.636
H_3	接受	接受	接受	接受
…				
改變位元數	165	166	167	168
F 值	0.451	0.284	0.828	1.533
H_3	接受	接受	接受	接受

表 33. H_4研究假設的變異數分析

改變位元數	1	2	3	4
F 值	1.133	2.337	0.441	0.495
H_4	接受	接受	接受	接受
…				
改變位元數	61	62	63	64
F 值	0.812	0.520	0.183	0.607
H_4	接受	接受	接受	接受

最後是相同模式下的兩種加密演算法進行比較，實驗結果如表 37、38，可以看出 jpg 的檔案不管是改變金鑰或改變明文，在 ECB 區塊加密模式下使用進階加密標準與三重資料加密標準兩者的密文變化量並無明顯的差異。

表 33. H_5研究假設的變異數分析

改變位元數	1	2	3	4
F 值	1.262	0.423	0.024	0.017
H_5	接受	接受	接受	接受
…				
改變位元數	165	166	167	168
F 值	1.393	0.179	0.312	0.554
H_5	接受	接受	接受	接受

表 33. H_6研究假設的變異數分析

改變位元數	1	2	3	4
F 值	0.002	1.479	0.854	0.027
H_6	接受	接受	接受	接受
…				
改變位元數	61	62	63	64
F 值	0.321	0.225	0.016	0.162
H_6	接受	接受	接受	接受

同樣的在 CBC 區塊加密模式下，進階加密標準與三重資料加密標準的實驗數據，皆為接受H_7、H_8之研究假設。

表 33. H_7研究假設的變異數分析

改變位元數	1	2	3	4
F 值	0.471	0.269	1.634	0.592
H_7	接受	接受	接受	接受
…				
改變位元數	165	166	167	168
F 值	0.379	0.737	0.985	0.946
H_7	接受	接受	接受	接受

表 33. H_8研究假設的變異數分析

改變位元數	1	2	3	4
F 值	0.506	2.320	0.928	0.021
H_8	接受	接受	接受	接受
…				
改變位元數	61	62	63	64
F 值	0.680	0.039	1.238	1.011
H_8	接受	接受	接受	接受

對不同的檔案類型整理如下，可以看到在 doc 與 exe 類型的檔案會拒絕 H_1、H_2 研究假設，推測是由於檔案內有不少相同的區塊，導致在 ECB 區塊加密模式下出現大量重複的密文，進而影響平均與標準差，因此在進行各種區塊加密模式的比較時會有明顯的不同。

表 46. 各檔案類型對 H_1、H_2 研究假設之關係

	Jpg	Txt	Doc	Pdf	Zip	Exe
H_1	接受	接受	拒絕	接受	接受	拒絕
H_2	接受	接受	拒絕	接受	接受	拒絕

從下表可知使用三重加密標準與進階加密標準在各種區塊加密模式的比較下並無差異，同樣是 doc 與 exe 下拒絕研究假設。

表 46. 各檔案類型對 H_3、H_4 研究假設之關係

	Jpg	Txt	Doc	Pdf	Zip	Exe
H_3	接受	接受	拒絕	接受	接受	拒絕
H_4	接受	接受	拒絕	接受	接受	拒絕

在 ECB 區塊加密模式下進階加密標準與三重加密標準的實驗結果如下表，可以看到兩者的密文變化量與檔案類型有關，其中較為特別的是 pdf 檔案類型在改變明文時的變化量，大約為一半接受一半拒絕。

表 46. 各檔案類型對 H_5、H_6 研究假設之關係

	Jpg	Txt	Doc	Pdf	Zip	Exe
H_5	接受	接受	拒絕	接受	接受	拒絕
H_6	接受	接受	拒絕	接受/拒絕	接受	拒絕

而 CBC 區塊加密模式中不管任何類型檔案皆為接受 H_7、H_8 研究假設。

表 46. 各檔案類型對 H_7、H_8 研究假設之關係

	Jpg	Txt	Doc	Pdf	Zip	Exe
H_7	接受	接受	拒絕	接受	接受	拒絕
H_8	接受	接受	拒絕	接受	接受	拒絕

五、結論

在本實驗中，分別透過了改變金鑰與改變明文來觀察加密後的密文之變化量，而由實驗數據可以看出，不同的檔案類型在透過進階加密標準加密後，由於檔案格式的關係，在 ECB 區塊加密模式中可能會有較為顯著的變化，但在其他加密模式中，改變金鑰所產生的變化並不是那麼的明顯。但是在改變明文的部分，由於 CTR 區塊加密模式的特性關係，會與明文變化量成正比，是與其他模式較為不同的地方。

在這五種加密模式中，除了改變明文的 CTR 區塊加密模式外，不論何種類型的檔案，每個區塊平均的改變量大約落在 64 位元左右，剛好佔了全部區塊（128 位元）的一半。如果使用三重資料加密標準進行加密的話，則每個區塊平均的變化量大約為 32 位元左右，同樣是區塊大小的一半。

依照實驗結果看來，使用何種區塊加密法與密文變化量較無直接關係，而是與採用的區塊加密模式有關，因此不論使用哪一種區塊加密法都比較建議使用 CBC、CFB、OFB 這三種區塊加密模式。

在 ECB 區塊加密模式下，進階加密標準與

三重資料加密標準兩者的密文變化程度與檔案類型有關，所以在使用 ECB 區塊加密模式進行加密時建議使用強韌度較高的檔案類型（jpg、txt、zip 等）。

而 CBC 區塊加密模式下則與 ECB 不同，不管任何類型的檔案在加密後其密文變化量並無明顯差異。

本文只選出較為常見的六種檔案格式測試密文的變化量，但除了這六種外還有其他也較常用於網際網路傳輸的檔案格式，後續研究可以同樣觀察這些格式密文之變化量，進一步驗證進階加密標準之加密強度。

六、參考文獻

1. 曹志良，「橢圓曲線密碼系統於GSM網路上的應用」，元智大學，碩士論文，2000。
2. A. Jurisic, A. J. Menezes; “Elliptic Curves and Cryptography”
3. Daniel J. Bernstein; “Understanding brute force”
4. David Kahn; “The Codebreakers”
5. David Salomon; “Coding for Data and Computer Communications”
6. FIPS 46-3; “Data Encryption Standard(DES)”
7. FIPS 197; “Advanced Encryption Standard（AES）”
8. Ibrahhim A. Al-Kaid; “The origins of cryptology: The Arab contributions”; Cryptologic 16(2), pp.97-126
9. Jim Owens, Jeanna Matthews; “A Study of Passwords and Methods Used in Brute-Force SSH Attacks”
10. Michael J. Wiener; “The full cost of cryptanalytic attacks”, Journal of Cryptology ;17 (2004), pp.105-124
11. Neil A. Weiss; Introductory Statistics.
12. NIST special publication 800-38A; “Recommendation for Block Cipher Modes of Operation”
13. N. Koblitz; “Elliptic curve cryptosystems”; in Mathematics of Computation 48, 1987, pp. 203-209
14. Paul C. Kocher; “Timing Attacks on Implementations of Diffie-Hellman,RSA,DSS and Other Systems”
15. R.L. Rivest, A. Shamir, L. Adleman; “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”; Communications of the ACM 21 1978 pp: 120-126
16. V. Miller, “Use of elliptic curves in cryptography”,Advance in Cryptology-CRYPTO 85, 1985, pp.417-426
17. William Stallings; Cryptography and Network Security; 2005.

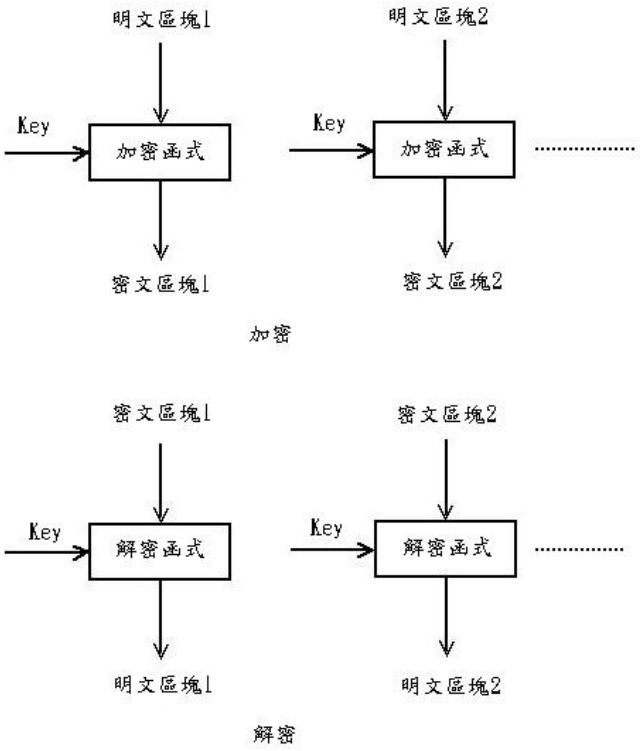

圖1. Electronic CodeBook Mode Encryption[17]

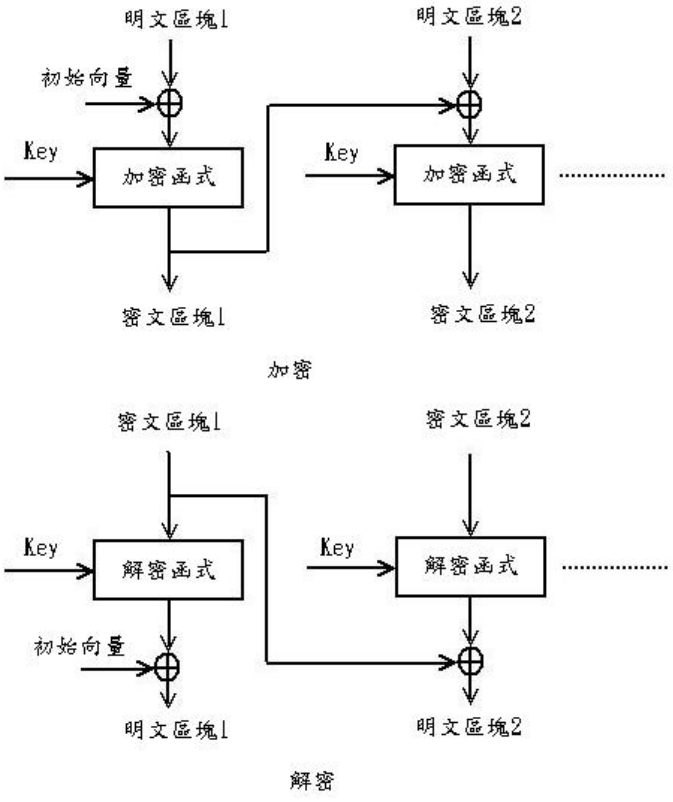

圖 2. Cipher-Block Chaining Mode Encryption[17]

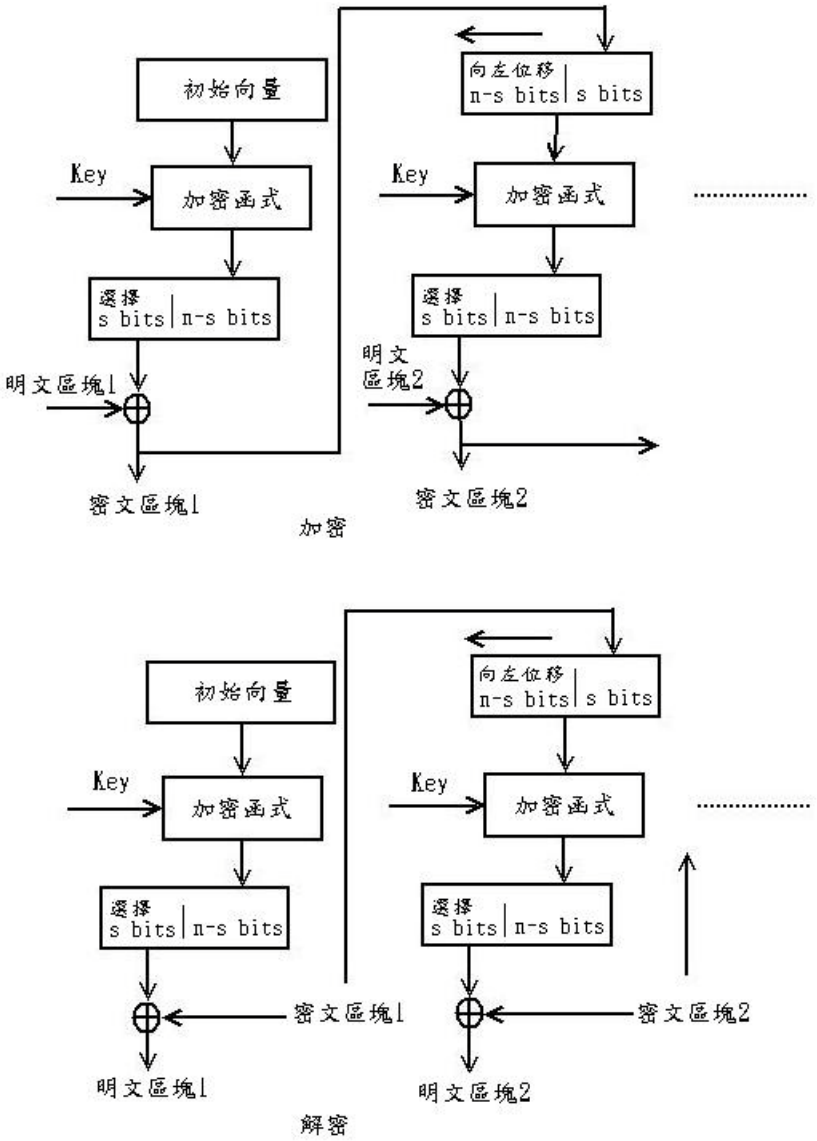

圖 3. Cipher Feedback Mode Encryption[17]

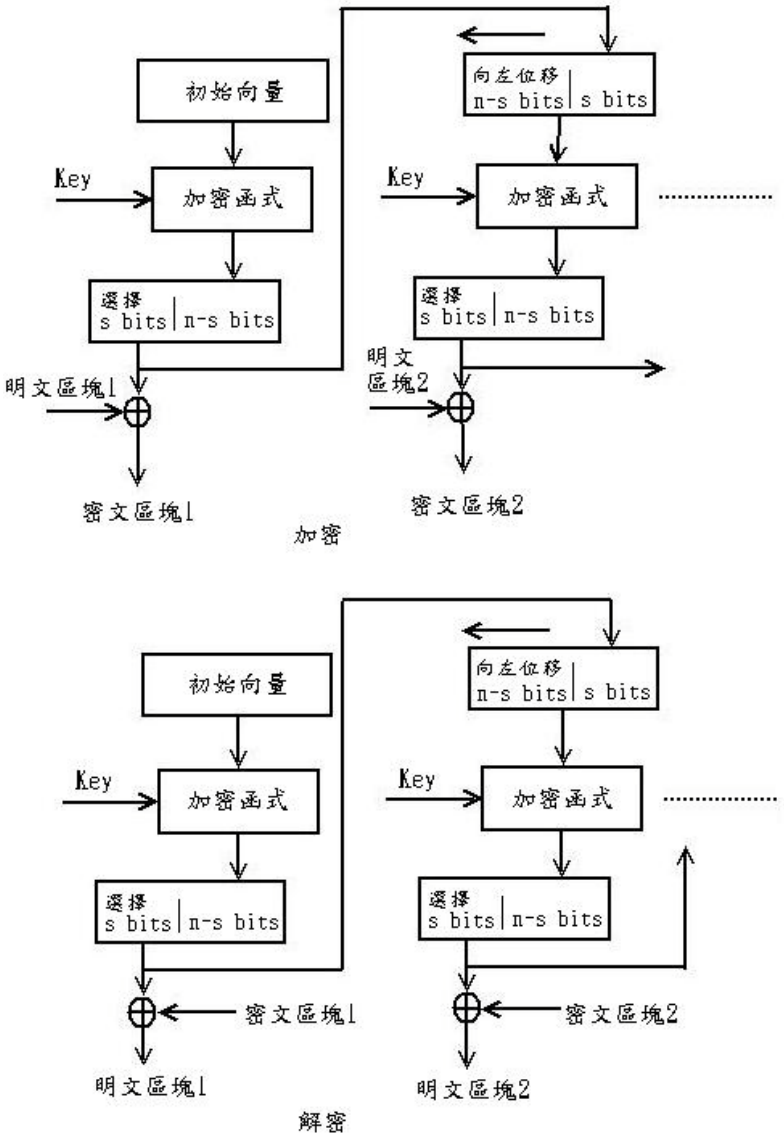

圖 4. Output Feedback Mode Encryption[17]

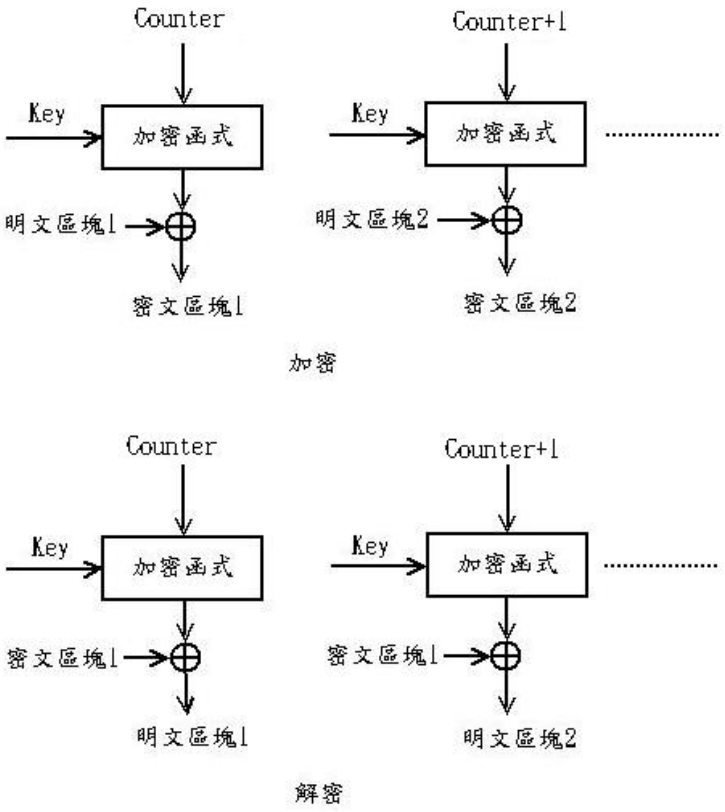

圖 5. Counter Mode Encryption[17]

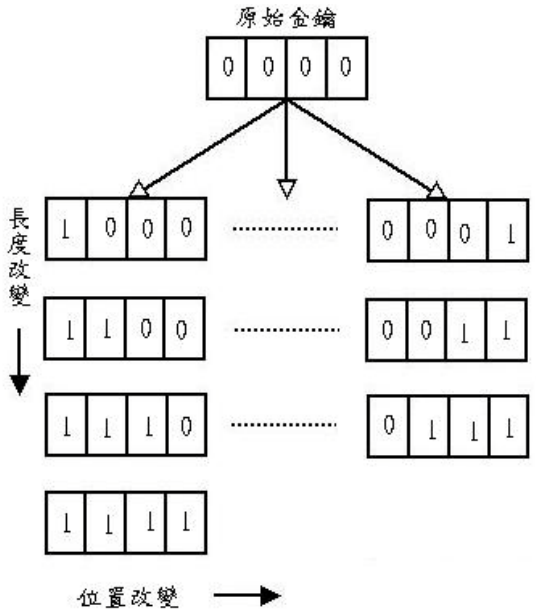

圖 6. 金鑰改變示意圖

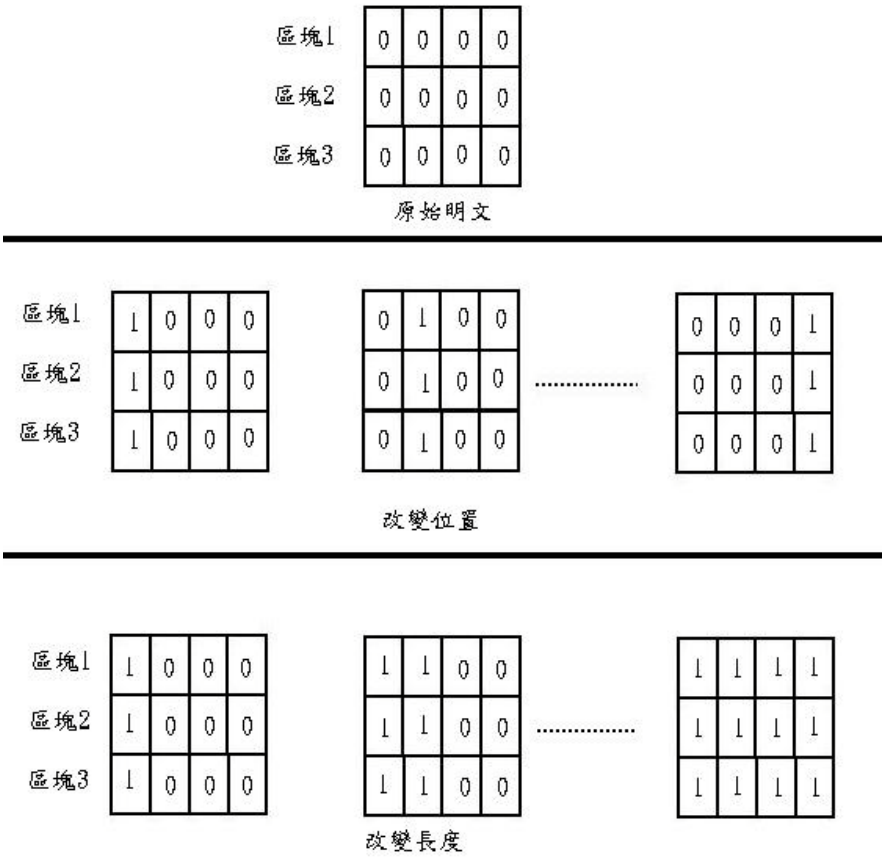
區塊1
區塊2
區塊3
0 0 0 0
0 0 0 0
0 0 0 0
原始明文
區塊1
區塊2
區塊3
1 0 0 0
1 0 0 0
1 0 0 0
0 1 0 0
0 1 0 0
0 1 0 0
.....
0 0 0 1
0 0 0 1
0 0 0 1
改變位置
區塊1
區塊2
區塊3
1 0 0 0
1 0 0 0
1 0 0 0
1 1 0 0
1 1 0 0
1 1 0 0
.....
1 1 1 1
1 1 1 1
1 1 1 1
改變長度

圖7. 明文改變示意圖