東海大學資訊工程研究所

碩士論文

指導教授：廖啟賢

Wrapping Feedback Encryption Based on Password

包覆式回授加密法

研究生:廖偉信

中華民國 一百零四 年　一月

東海大學碩士學位論文考試審定書

東海大學資訊工程學系 研究所

研究生 廖 偉 信 所提之論文

包 覆 式 回 授 加 密 法

經本委員會審查，符合碩士學位論文標準。

學位考試委員會

召 集 人 簽志宏 簽章

委 員 林正堯

指 導 教 授 [illegible] 簽章

中華民國 104 年 1 月 12 日

摘要

在網路和雲端的快速蓬勃發展下，資料的傳輸也愈來愈頻繁，資料的安全保護機制因而顯得更為重要。目前 DES 和 AES 為最廣泛使用的區塊加密機制，因 DES 和 AES 為組合邏輯式的加密方法，一直飽受於各種暴力法的攻擊；幾年前 DES 已被公開破解了，在計算機快速處理速度和強大計算能力的持續提升下，AES 其安全性正岌岌可危。針對於前述的威脅，在本文中，我們提出了一個以密碼為基礎的包覆式回授加密法，它採用輸入的密碼或輸入的通道金鑰作為加密的啟動金鑰，並使用動態旋轉置換盒(Dynamic Rotating S-Box, DRS-Box)加密技術來產生系統加密所需之子金鑰群，接著擷取當前的時間參數與隨機亂數金鑰為動態參數進行包覆式回授加密，由此產生的包覆式密文檔案資料會因密碼、當前時間的不同而有不同的密文內容與檔案長度，如此破密者無法有效收集明文與密文對，大大提升破密的困難度，而且每個明文區塊都受到內部回授串流碼與二維運算的加密保護，使得密文區塊的安全度得以確保，此外，也同時提升了效能。經由理論與分析顯示，WFBPW 與 AES 應用於雲端通訊，兩者皆可達實際安全，而 WFBPW 之效能遠優於 AES，WFBPW 更適合為雲端通訊檔案傳輸之保護。

關鍵字: 資料加密標準、進階加密標準、區塊加密、系統安全碼、動態旋轉置換盒(DRS-Box)加密技術、包覆式回授加密

Abstract

The transmission of information is getting more frequently because of the rapid development of the Internet and cloud. Thus information security protection mechanisms are important. DES and AES are currently the most widely used encryption mechanism. The two encryption methods for logical formula have been suffering from a variety of attacks. A few years ago DES was cracked open. Computer processing speed and computing capability are improving, and AES may not be safe in the future.

A wrapping feedback encryption method based on password is proposed. We use a password or input key channel as an encrypted key, and dynamic rotary displacement box (Dynamic Rotating S-Box, DRS-Box) encryption technology to produce the required subsystem encryption key group. Current time parameters and random number keys are wrapped feedback encryption as dynamic parameters. This password and ciphertext length of the current content and file of time keep changing, so intruder cannot effectively collect the secret plaintext and ciphertext files. Each plaintext block are subject to internal feedback streaming and two-dimensional code encryption operations, which ensures the security of the ciphertext block and enhances the performance. Through theory and analysis, the WFBPW and AES are up to the actual security to cloud communications. The WFBPW has better performance than AES; thus WFBPW is more suitable for the protection of cloud file transmission.

關鍵字: DES、AES、block encryption、system security codes、dynamic rotary displacement box (Dynamic Rotating S-Box, DRS-Box) encryption technology、wrapping feedback encryption

目錄

圖目錄

表目錄

第一章 緒論

1.1 研究動機與背景

在網際網路和計算機的快速蓬勃發展下，不僅是豐富了我們的日常生活，也提供了更便利的生活環境。而現今 4G 網路的快速發展，是一種用來替代 3G 蜂窩的第四代無線蜂窩系統，能夠以 100 Mbps 的速度下載資料，而 4G 的速度比 3G 快 7 到 10 倍[1]。由於 4G 具有高速傳輸的優點，把資料上傳於雲端變成更快速與簡單，且因雲端的快速發展，使得大眾人們紛紛把資料傳輸至雲端儲存或進行運算處理，然而在資料的傳輸過程中，尤其是經由無線傳輸的資料很容易被破密者取得而造成安全隱憂，所以資料的安全保護機制顯得更為重要；另外，傳輸至雲端的資料檔案經常是較大的檔案資料，其效能的問題也就相對的重要了。

目前，DES 和 AES 為最廣泛使用的區塊加密機制，然而因 DES 和 AES 為組合邏輯式的加密方法，所以一直飽受於各種暴力法的攻擊[2-3]。在 1999 年 1 月時，由 distributed.net 與電子前哨基金會合作公開破解 DES[4]，而 AES 其安全性亦岌岌可危[5]，這意味著，我們需要更安全的區塊資料加密方法，同時，對於較大型的資料檔案，如大於 MB 的資料檔案，因 4G 的高速傳輸速率，導致資料在運輸方面所需的時間將會大大的減少，所以加密時間會佔總時間很大的一部分，DES 和 AES 的效能也顯得不足。

針對於前述問題，在本文中，我們提出一種以密碼為基礎的包覆式回授加密方法(A Wrapping Feedback Encryption Based on PassWord，簡稱 WFBPW)，來增強密文的安全度並提升其效能。本法採用輸入的密碼或輸入的通道金鑰作為加密的啟動金鑰，並使用動態旋轉置換盒加密技術與二維運算加密技術來產生系統加密所需之子金鑰群，接著擷取當前的時間參數與隨機亂數金鑰[6]為動態參數群進行包覆式回授加密[7]，使用本方法進行加密，即使在同一明文與相同密碼的條件下進行加密，其產生的包覆式密文檔案資料，會因時間參數與隨機亂碼的不同而有不同的密文內容與不同的檔案長度，如此破密者將無法知道密文的正確位置，當然也就無法得到正確的明文與密文對，如此大大增加了破密的困難度；除此之外，每個明文區塊都受到內部回授串流碼與二維運算的加密保護，使密文區塊的安全度得以提升。在效能方面，由於每個明文區塊只需使用 3 次的 $+_2$ 運算與⊕運算，就可產生密文區塊，讓檔案在較安全的狀態。

1.2 研究目的

因網際網路的快速發展跟雲端計算的進步，現今 4G 網路又即將快速的進入我們生活中，所以有愈來愈多人紛紛把資料送至雲端空間去進行儲存跟計算的工作，所以資料檔案的加密安全就顯得更為重要的，在此，我們針對 DES 跟 AES 的共同特點而導致的缺點

去進行分析和改善，以減少中大型檔案的加密時間和提高它的安全性，讓使用者把資料檔案放置雲端空間時，不會輕易的被攻擊，以保證檔案的安全性。在 4G 中傳輸速度會大大的提升，對於下載資料的時間會減少很多，所以加密的時間與安全就顯得更為重要。

1.3 論節架構

本文的其他部份安排如下，第 2 章簡要介紹本文的相關研究，包括 AES 的內涵與缺點、其他的加密方法，如 SNOW 3G 和 3DES 和雲端安全與資料傳輸安全，第 3 章則是闡述植基於 PW 的包覆式回授加密法，第 4 章則是針對本法在訊息資料加/解密實際使用的運作模式中的運作模式一與運作模式二的工作環境中進行的安全性分析，在第五章則是把本法跟 AES 在個人電腦中進行模擬並進行效能的分析比較，最後則是總結全文和概述了我們未來的研究方向。

第二章相關研究

本章我們描述了(1)AES 的內涵與其面臨的威脅與缺點，(2)SNOW 3G 的內涵和 3DES 加密方式(3) 雲端安全的挑戰與資料傳輸安全的探討。

2.1 進階加密標準(Advanced Encryption Standard, AES)

AES[8-10]是具有 128 位元長度的密鑰分組密碼技術。實際上，長度可以是 128、192 或 256 位元。重複回合數可為 10、12 和 14 次，重複回合長度由密鑰長度而定。AES 加密過程是在一個 4x4 的位元組矩陣上運作，這個矩陣又稱為「體（state)」，其初值就是一個明文區塊。

2.1.1 子金鑰產生

AES 加密過程是在一個 4x4 的位元組矩陣上運作，其初值就是一個明文區塊（矩陣中一個元素大小就是明文區塊中的一個 Byte，分為 W_{i-4} 到 W_{i-1} 列)。第 W_i 列所產生的子金鑰則是由初值的明文區塊所產生的，由 W_{i-1} 列的元素先往上移一位產生新的列，此列的元素再經由對應 S-box 的所對應值來產生一列新的元素，然後由 W_{i-4} 與新的 W_{i-1} 列和 Rcon(i)來進行 XOR 運算，所產生的值為第 W_i 的元素，之後第 W_{i+1} 列的值則由 W_{i-2} 與第 W_i 列的元素進行 XOR 運算所產生，由此列推，當第 i 列所需產生的子金鑰為前一個區塊的第 i/4 列的值和第 i-1 列的值進行 XOR 運算所產生的值為第 i 列的元素，用此方法擴充金鑰來產生 AES 的子金鑰，之後所產生的子金鑰將參與進行後續 AES 的加密動作。

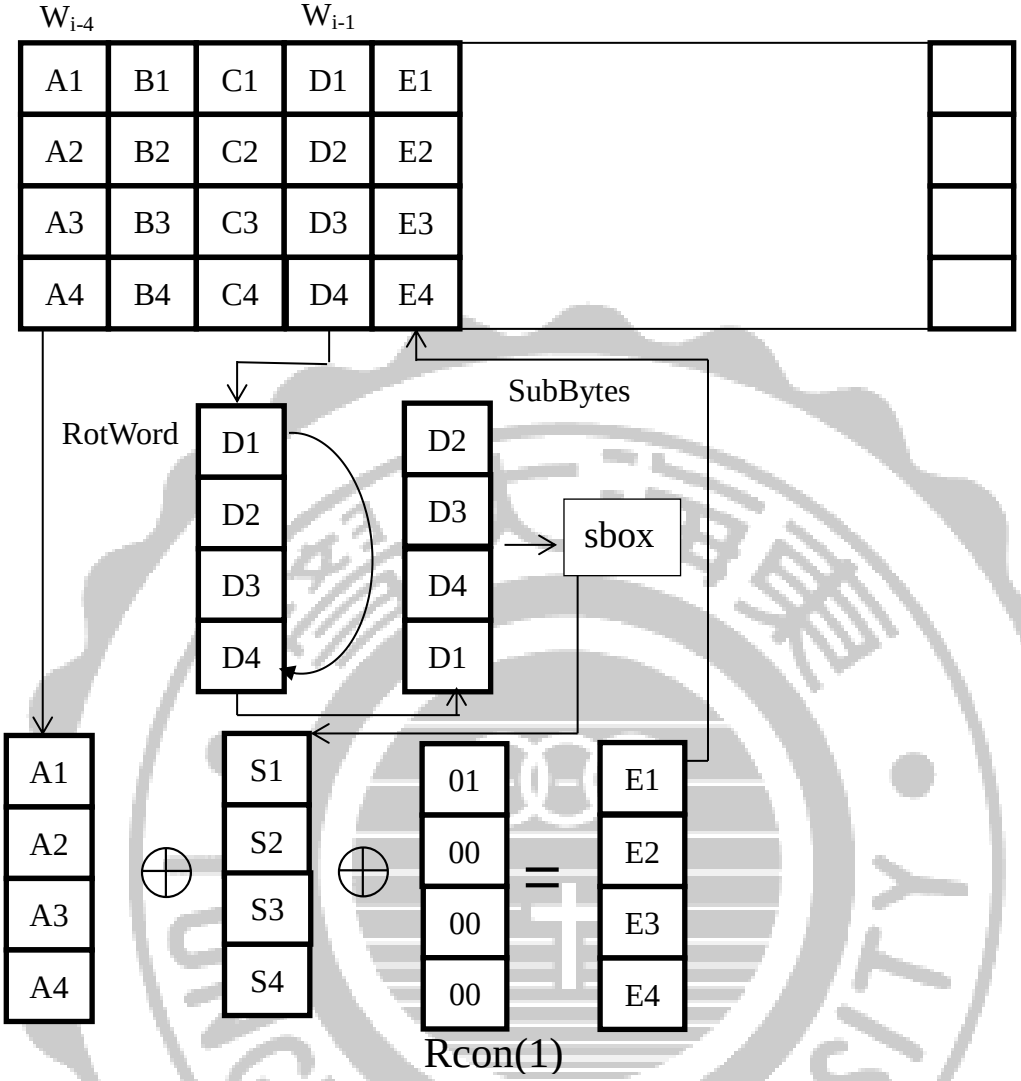

圖 1 AES 子金鑰產生過程圖

2.1.2 AES 加/解密過程

初次使用 AddRoundKey 運算對明文加密，之後是 9 次回合的加密運算，每一回合加密包含下列四種運算，即 SubBytes、ShiftRows、MixColumns 與 AddRoundKey 運算，最後一回運算省略 MixColumns 運算,即最後一輪只有 SubBytes、ShiftRows 和 AddRoundKey。此四種運算內涵如下：(1) AddRoundKey 運算，是指將一把回合子金鑰與「體」矩陣作⊕運算;(2) SubBytes 運算是指「體」矩陣中的每個位元組透過 S-box 查表來進行替換;(3) ShiftRows 運算是指是針對「體」矩陣的每一個橫列位元組，進行向左循環位移某個偏移量;(4) MixColumns[11-13]運算是指對於「體」矩陣的每一直行，與一多項式 $c(x)=3X^3+X^2+X+2$ 在 modulo X^4+1 下進行多項式乘法運算。

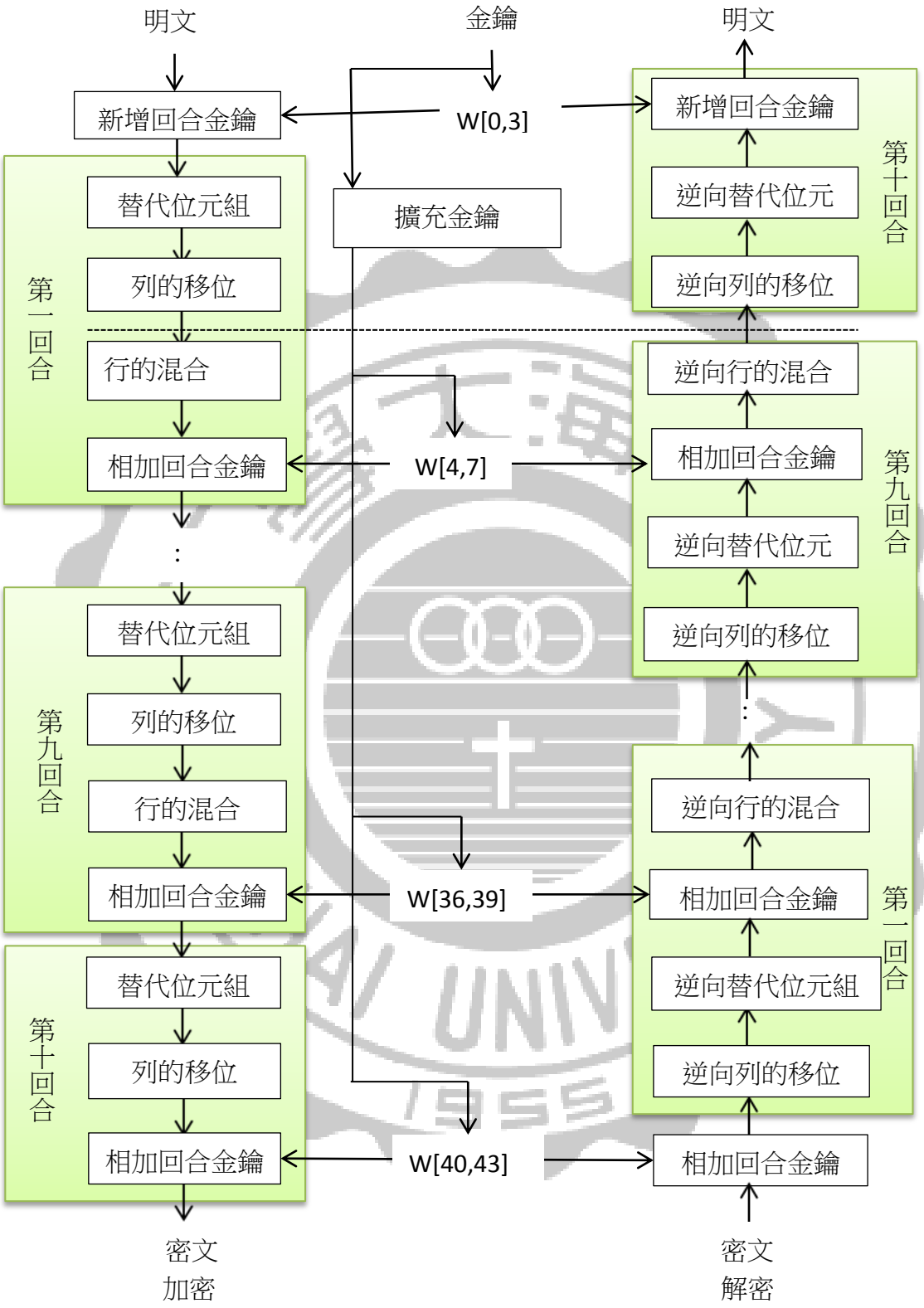
明文
金鑰
明文
新增回合金鑰
W[0,3]
新增回合金鑰
逆向替代位元
逆向列的移位
第十回合
擴充金鑰
替代位元組
列的移位
行的混合
相加回合金鑰
第一回合
W[4,7]
逆向行的混合
相加回合金鑰
逆向替代位元
逆向列的移位
第九回合
:
:
替代位元組
列的移位
行的混合
相加回合金鑰
第九回合
W[36,39]
逆向行的混合
相加回合金鑰
逆向替代位元組
逆向列的移位
第一回合
替代位元組
列的移位
相加回合金鑰
第十回合
W[40,43]
相加回合金鑰
密文
加密
密文
解密

圖 2 AES 加密圖[14]

2.1.3 AES 面臨的威脅與缺點

AES 自從由美國國家標準與技術研究院(NIST)於 2001 年 11 月 26 日發佈於 FIPS PUB 197,並在 2002 年開始被廣泛應用於資料加密上以來,一直都是被視為最安全的加密法,直到 2009 年 Bogdanov 等人[15-16]提出的分組密碼算法的密碼分析技術與 bicliques 攻擊。此 biclique 分析適用於 AES 的所有完整版本,並根據密鑰長度來破解 AES 密鑰所需的時間大約只需五分之一到三分之一的暴力法解密時間。此外,AES 的發明人 Daemen 和 Rijmen 也證明了上述攻擊的有效性[15-17],即使尚未完全破解,但有破解 AES 的可能性,所以藉由更快的計算機和新的算法,AES 將岌岌可危。

2.1.3 AES 的缺點

雖然 AES 為目前大眾廣泛使用的區塊加密方式之一,但以下為我們針對 AES 所提出的幾項缺點:

(1) AES 之組合邏輯加密方式,其輸出之密文完全由當時輸入之明文決定,此種加密方式無法有效抵禦暴力法攻擊,例如,已知明文/密文攻擊及差異攻擊等。由於 DES 為 64 位元之資料區塊加密法,目前已被電子前線基金會(Electronic Frontier Foundation, EFF) 的「DES Cracker」專門機器所破解,由此觀之,具有較長 128 位元的 AES 未來也將要面臨被破解的危機。

(2) AES 是固定大小的資料區塊加密,其缺點是限制了加密系統的彈性。若資料區塊加密的大小得以彈性變化,則可使加密系統能更靈活地依實際需要對資料加密,而有效抵禦暴力法或其他方法的攻擊。

(3) 由於 AES 加密方法都是重複多次特定核心運算,如 AES 重複 10 次,雖然每次重複計算皆有新的子金鑰加入,但因重複計算相同的公式,難免會有較低安全度之慮,又多次的重複計算使得效能大幅降低。

(4) AES 均採用固定的 S-Box,若能設計成動態的 S-Box,讓不同的資料加密時,均能面對不同的 S-Box 內涵值,而做不同的非線性置換,將會增大其安全度。

2.2 常見加密方法簡介

除了一開始介紹的 AES 加密方法以外,在此小節也介紹了 SNOW 3G 跟 3DES 的加密方法。

2.2.1 SNOW 3G

SNOW 3G[18]密碼算法是 3GPP 中實現數據保密性和數據完整性的算法,由 UEA2 和

UIA2 為核心所組成的，是一個為 32 字元所實現的密碼算法，密鑰為 128 字元的密碼算法。SNOW 3G 算法的結構如圖 3 所示，它主要是由三個部分所組成，有線性反饋移位寄存器（LFSR）、有限狀態機（FSM）和反饋過程(Feedback)。此線性反饋移位寄存器由 16 次的轉換使用在 32 位元的長度上（S_0到 S_{15}）。在 FSM 的數據映射是使用替換盒 S1 和 S2 進行。訊息是 32 位元的轉換這由 Rijndael 算法 S 盒（SR）和另一個指定的 S 盒（SQ），分別為此算法的反饋部分，並使用 MULa 函數和 DIVa 函數，把原本為 8 位元的密鑰映射成為 32 位元的密鑰檔案輸出。

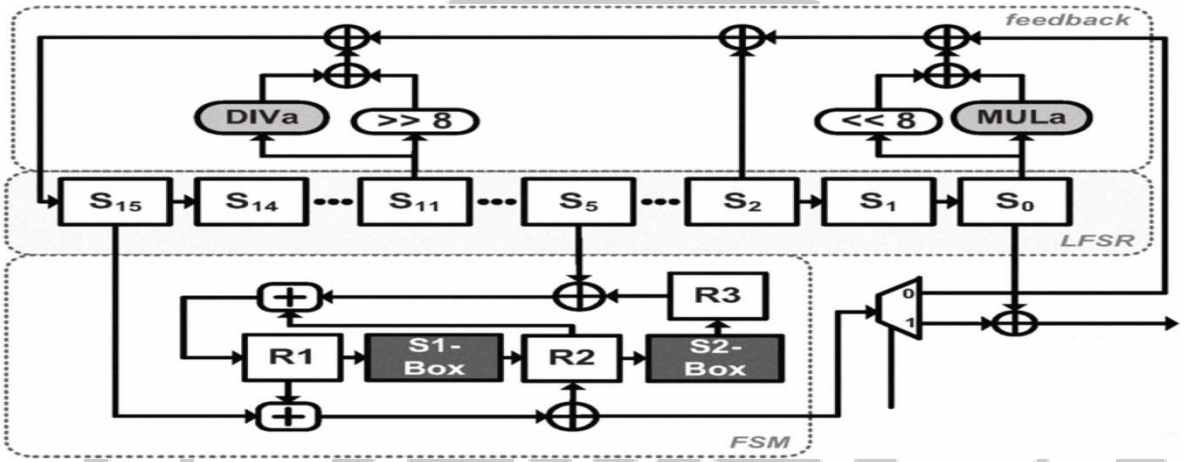

圖 3 SNOW 3G 結構圖

SNOW 3G 具有兩種操作模式：初始化過程和密鑰生成過程。在初始化期間，LFSR 階段被重置為加密密鑰，並通過無線電資源控制（RRC）子層信號的通信參數的 XOR 所組合。之後再密鑰生成過程中 FSM 的輸出將不再進入入 LFSR，而是作為整體的輸出。輸出就作為密鑰產生明文的加密過程，而這些參數是採用訊息長度、數據包數、無線承載和傳播的方向（上行/下行），而密鑰區塊由 FSM 的輸出和 S_0之間執行 XOR 所組成密鑰。

2.2.2 三重資料加密演算法(TDEA, Triple Data Encryption Algorithm)

3DES[19-20]（或稱為 Triple DES）是三重資料加密演算法（TDEA, Triple Data Encryption Algorithm）塊密碼的總稱，它相當於對每個資料區塊進行三次 DES 加密演算法。由於計算機運算能力的增強，3DES 是設計提供一種相對簡單的方法，即增加 DES 的金鑰長度來避免暴力法的攻擊。3DES 定義了三種金鑰選項：選項一:此三個金鑰是獨立的，安全強度是最高的，有 168 個獨立金鑰位元。選項二：K_1和 K_2是獨立的，且 $K_3=K_1$，此選項擁有 112 個獨立金鑰位元,也可防禦中途相遇攻擊。選項三則是三個金鑰均相等，即 $K_1=K_2=K_3$，雖只有 56 個獨立金鑰位元，但提供了與 DES 的相容性。

3DES 的加密方式是先將明文用第一把金鑰加密後再經第二把金鑰來解密，最後用第三把金鑰執行加密的動作，此加密過程為密文 $=E_{K3}(D_{K2}(E_{K1}(\text{明文})))$，而解密則是反之，先解密，再加密後解密，其解密過程為明文 $=D_{K1}(E_{K2}(D_{K3}(\text{密文})))$，而每次加密動作只處

理 64 位元資料檔案。無論是執行加密還是解密時，中間值行的動作皆是前後兩者的逆動作。

2.3 雲端安全

雲端安全[21]是指一套廣泛的技術和被佈署控制的方法，它是用來保護資料、應用程式和雲端運算的基礎設施。當我們傳輸檔案資料到雲端時，因為雲端上的檔案資料會受到來自許多不同用戶端設備的存取，所以在安全風險上面臨的問題包括：(1) 網路資料傳輸，在上傳至雲端中，將面臨資料傳輸的安全威脅，若資料缺少安全加密機制保護，在傳輸中可能被非法的第三方監控並擷取，或是被更改破壞。(2) 雲端資料保密，當資料在系統上是否能受到足夠的保密與保護，以致於系統資料能不被外人竊取或外洩。(3) 資料儲存穩定，儲存於雲端系統上的資料，是否能穩定地被保存，不會受到外在因素而消失。(4) 個人帳號管理，存取雲端儲存的資料必須透過個人的帳號才能登入並存取，若個人的帳號密碼管理不當或遭他人盜用，則個人帳號、隱私及其儲存之資料將受到嚴重的威脅。還要考慮可能會有哪些人員可以存取這些資料檔案，例如是作業人員、執行的應用程式或者是系統本身，而又可以透過什麼樣的方式來進行資料檔案的存取。

2.3.1 雲端安全的挑戰與威脅

目前，雲端安全聯盟 CSA (Cloud Security Alliance) 認為和雲端資料有關的安全挑戰，大致可分為以下幾項：(1) 檔案資料本身的安全；(2) 檔案資料存放的位置；(3) 檔案資料的刪除與銷毀；(4) 不同用戶資料的混合；(5) 資料的備份與重建；(6) 法律所需的資料發現和資料的聚合與推理等。雲端運算可能遭遇的七大安全威脅分述如下：(1) 濫用或惡意使用雲端運算的行為；(2)不安全的使用者介面與 API;(3) 惡意的內部人員;(4) 共享環境所造成的議題;(5)資料遺失或外洩;(6) 帳號或服務被竊取;(7) 未知的風險模型。本文方法將針對檔案資料本身的安全這一項挑戰來進行改進。

2.3.2 雲端之資料傳輸安全探討

如果系統對資料檔案的傳輸有高度的安全性要求，可以用以下這三者方法來進行傳輸，第一種是可針對用戶端和應用程式來進行加密，如果資料是存放在用戶端設備或透過應用程式傳輸之前，就能夠實施加密，檔案資料安全將會獲得較多的保障[22-23]；其次是針對網路連線的加密，可以透過硬體設備或軟體方式，建立所需的加密連線通道，常見的作法包括 SSL、VPN、SSH 等，確保在點對點之間已實施了安全防護[24-26]； 第三種是採用代理伺服器(Proxy-based)的加密方法，這對於一些較為老舊或缺少加密功能的應用程式來說，最大的好處是在傳輸資料之前，能夠將資料先傳輸到代理伺服器，再透過它來實施資料加密功能。本文的方法可以提升第一種和第二種資料傳輸的安全與效能，對於中大型的檔案其效能更加顯著。

第三章 以密碼為基礎的包覆式回授加密法

以密碼為基礎的包覆式回授加密法是由模擬隨機數列 1(PRNS1)、已加密過的隨機金鑰(CRK)、密文檔案和模擬隨機數列 2(PRNS2)所組成的一個包覆式密文檔案，在這當中，在每個明文區塊都會受到內部回授串流碼(b_{i-1})與二維運算的加密保護，來使得密文區塊的安全度得以確保，也利用隨機亂碼金鑰與當前時間金鑰動態的來對明文進行加密以產生隨機金鑰，使得同一明文資料在不同時間點加密，會有不同的密文檔案長度與內涵，且並放置在密文檔案的前面，並採用密碼、檔名、附檔名、系統安全碼、二維運算與 mod 運算來產生 PRNS1 和 PRNS2 的長度和產生隨機亂數碼，在隨機金鑰前面加上一段 PRNS1，在檔案密文的後面也加上一段 PRNS2，來產生由 PRNS1、CRK、密文檔案和 PRNS2 所組成的一個包覆式密文檔案結構，也因包覆式的密文檔案結構，使得破密者不知密文的真正起始點，因而無法取得明文／密文對，而提升了破密的難度。

3.1 定義系統參數

我們首先定義本法所使用到的參數與運算子。

3.1.1 定義參數

下面為此包覆式回授加密法所使用到的參數。

PW：密碼，由使用者輸入為 8 到 16 字元

K_{PW}：密碼金鑰，將輸入的 *PW* 透過一定的演算法則而產生

ct：累積位移的變數值

SS、*SB*：字元變數值

SSC：系統安全碼 (System Security Codes)

ssc(i)：第 i 個系統安全碼，1≦i≦8，長度為 128 位元

FN：檔名||副檔名

K_{FN}：檔名金鑰，由重複串接 *FN* 而擴充至 128 位元，i.e.，K_{FN}=*FN*||*FN*||*FN*||……至 128bit

PRNS1：模擬隨機數列 1

PRNS2：模擬隨機數列 2

Δh：*PRNS1* 的長度

Δt：*PRNS2* 的長度

K_{CT}：由當時的 CPU TIME 產生之時間金鑰，長度為 128 位元，其內容為奈秒/日/時/分/秒/奈秒/時/分/秒

K_{RCT}：K_{CT}之倒置金鑰，長度為 128 位元，其內容為秒/分/時/奈秒/秒/分/時/日/奈秒

RK：隨機金鑰

CRK：加密過的隨機金鑰

b_0 ~ b_n：內部回授串流碼

明文：$P_1P_2\cdots\cdots P_n$, P_j, $1 \leqq j \leqq n$, 大小為 128 bits

密文：$C_1C_2\cdots\cdots C_n$, P_j, $1 \leqq j \leqq n$, 大小為 128 bits

3.1.2 定義運算子

以下為包覆式加密回授加密法所使用到的運算子，有互斥或算子、二進制加法算子、模算子和把密碼轉換成密碼金鑰所使用的動態移位置換技術的定義。

(1) 互斥或算子：$\oplus$

加密：$c=p\oplus k$，其中 p 為明文，k 為通道金鑰，c 為密文

解密：$p=c\oplus k$

(2) 二進制加法算子：$+_2$

加密：$c=p+_2k$，為二進制加法運算，此加法運算捨棄了溢位位元

解密：$p=c-_2k=\begin{cases} c-k & , if c \geq k \\ c+\bar{k}+1, & if c < k \end{cases}$，其中 $-_2$ 為 $+_2$ 的逆運算

(3) 模算子：mod

$c=p \bmod n$，n 為一大整數

(4) 動態移位置換技術：

Input: *SS*(字元), *ct*(位移量)

Output: *SB*(字元)

以 S-Box 為置換盒，先找出對應於 *SS* 字元的置換字元，再以此字元為起點，位移 *ct* 個字元後之 S-Box 目標字元即為 *SB* 字元

(5) *Fct*(*SS*)： 一個計值函數，計算字元 *SS* 其 ASCII 二進制表示式中 '1' 的個數。

(6) *Mid*(*PW*, *i*, *n*)：由 *PW* 第 *i* 個字元開始取出 *n* 個字元函數

(7) *Right*(*PW*, *n*)：取出 *PW* 右邊的 *n* 個字元函數

(8) *trunc*(*RN(j)*, *t*)：捨去隨機亂數金鑰 *RN(j)* 最右邊 t 個位元

3.2 加密機制

以密碼為基礎的包覆式回授加密法的加密機制主要是由前面的前置處理工作產生所需要的金鑰與內部回授串流碼和二維運算所構成，藉由使用內部回授串流碼與二維運算使效能提高又能保有其密文安全度，以下將闡述包覆式回授加密法的加密流程、密碼金鑰的產生與加密步驟。

3.2.1 加密流程

以密碼為基礎的包覆式回授加密法的加密流程可以從圖 4 看出密文經由明文、內部回授串流碼(b_{i-1})、隨機金鑰 *RK* 與系統安全碼經由二維運算所保護，內部回授串流碼也經由明文、內部回授串流碼(b_{i-1})、隨機金鑰 *RK* 與系統安全碼經由二維運算所保護。

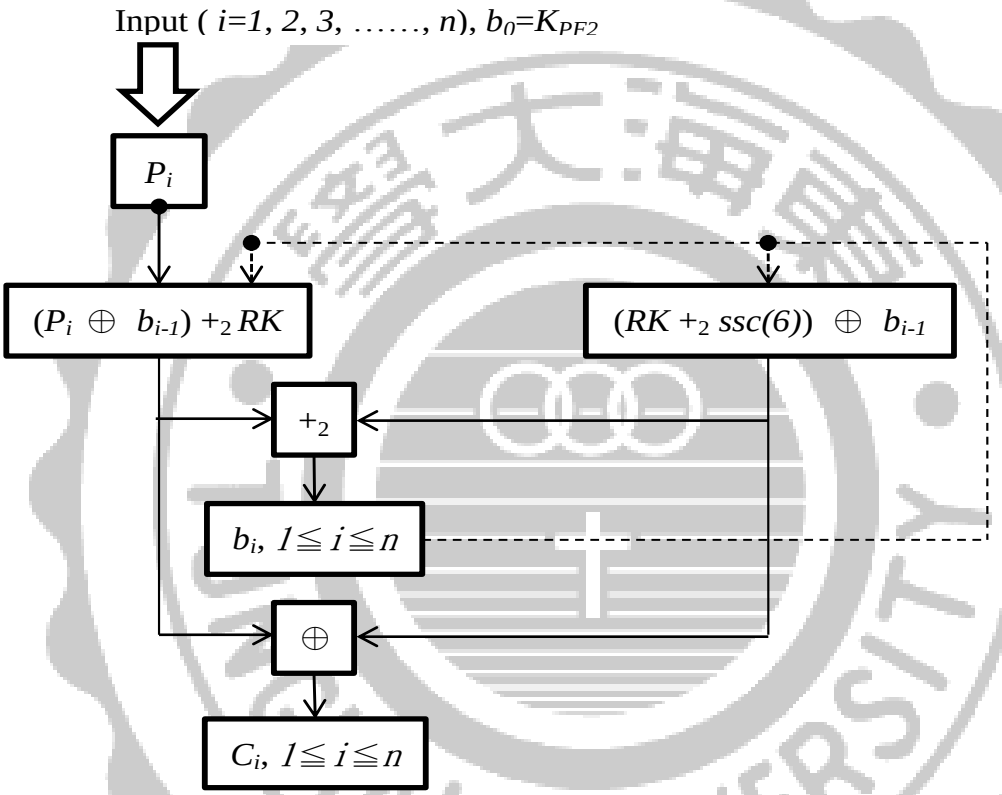

圖 4 WFBPW 加密流程圖

3.2.2 密碼金鑰的產生

由使用者輸入的密碼(*PW*)來產生密碼金鑰(K_{PW})是件重要的工作，因 K_{PW} 就是系統起始金鑰 K_0，其內容品質會深深影響本加密系統的安全度，基於此，由 *PW* 擴充產生 K_{PW} 我們遵守下列三原則：(1) 保有原來 *PW* 的內容 (2) 以原來 *PW* 的內容為依據進行擴充碼的產生 (3) 在原來 *PW* 的內容中有相同字元重複出現時，則對應相同字元的擴充碼會不相同。以下演算法表示當使用者輸入密碼 (*PW*) 到最後產生密碼金鍵 (K_{PW}) 的過程。

Algorithm 1: 由使用者輸入密碼(*PW*)並藉由累計轉移的 S-Box 來產生密碼金鑰(K_{PW})，

一開始由使用者輸入 8 到 16 位字元的密碼，假設如果輸入值為 16 位字元，則密碼金鑰元原始的 16 位字元，若使用者輸入值在 8 到 16 位字元之間，則需要經過動態移位置換技術技術，經由 *S-box* 的轉換擴充碼和字元位移的累加計算方法，來達至擴充密碼金鑰到 16 位字元

(1) Input *PW*.
(2) Find *l*; *l* is the length of *PW* with byte unit.
(3) If *l* < 8 or *l* >16 then call re-input.
(4) If *l* = 16 then K_{PW} = *PW*, END.
(5) *n* = 16 - *l*.
    ct = 1.
    K_{PW} = Null.
    SS = Null.
(6) For *i* = 1 to *n*
        SS = *Mid(PW, i,* 1)
        ct = *ct* + *Fct*(*SS*)
        Generate *SB*; (依據 S-Box, 找出對應於 *SS* 字元的置換字元，再以此字元為起點，位移 *ct* 字元後之 S-Box 目標字元即為 *SB* 字元)
        K_{PW} = K_{PW} //*SS*//*SB*
    Next *i*
(7) K_{PW} = K_{PW} //Right(*PW*, *l* - *n*)
(8) END.

3.2.3 加密步驟

Step1:
(1)由輸入的 *PW*(password)透過演算法一而產生密碼金鑰 K_{PW}
(2)由檔名，副檔名產生檔名金鑰 K_{FN}
(3)$K_{PF1} = K_{PW} \oplus K_{FN}$，$K_{PF2} = K_{PW} +_2 K_{FN}$ (1)
(4)由 *SSC*, K_{PF1}, 與 K_{PF2} 產生 Δh, $3 \le \Delta h \le 1024$

$$\Delta h = [[(ssc(1) +_2 K_{PF1}) \oplus (ssc(2) +_2 K_{PF2})] +_2 (ssc(3) \oplus ssc(4))] \bmod 1022 + 3 \quad (2)$$

Step2:
(1)產生一初始隨機金鑰 R_{K0}
(2)讀取 CPU time, 產生目前時間金鑰 K_{CT} 與倒置時間金鑰 K_{RCT}
(3)產生隨機金鑰，$RK = (R_{K0} +_2 K_{CT}) \oplus (R_{K0} \oplus K_{RCT})$ (3)

$$CRK = [(RK \oplus ssc(5)) +_2 K_{PF1}] +_2 [(K_{PF2} \oplus ssc(6)) +_2 ssc(7)] \quad (4)$$

Step3:設明文(Plaintext)=$P_1P_2\cdots P_n$，密文(Ciphertext)=$C_1C_2\cdots C_n$

$b_0 = K_{PF2}$

For i = 1 to n

$$b_i=[(P_i \oplus b_{i-1}) +_2 RK]+_2[(RK+_2 ssc(6)) \oplus b_{i-1}] \quad (5)$$

$$C_i=[(P_i \oplus b_{i-1}) +_2 RK] \oplus [(RK+_2 ssc(6)) \oplus b_{i-1}] \quad (6)$$

Next i

Step4:

(1) $\Delta t = [(ssc(1)+_2 RK) \oplus ssc(8)+_2 (ssc(2) \oplus RK)] \bmod 1022+3$ (7)

(2)將 $ssc(5), K_{PF1}, b_n, \Delta h, \Delta t$ 輸入 PRNG 產生 PRNS1||PRNS2，如圖 5 所示

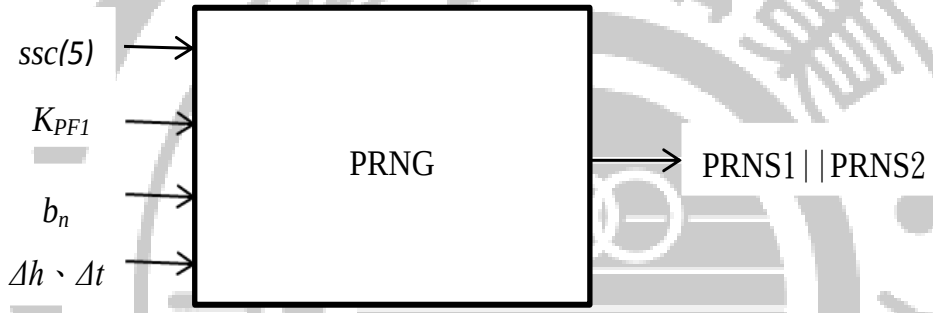

圖 5 PRNS1、PRNS2 產生之示意圖

(3) 產生包覆式密文檔案，其檔案結構如下

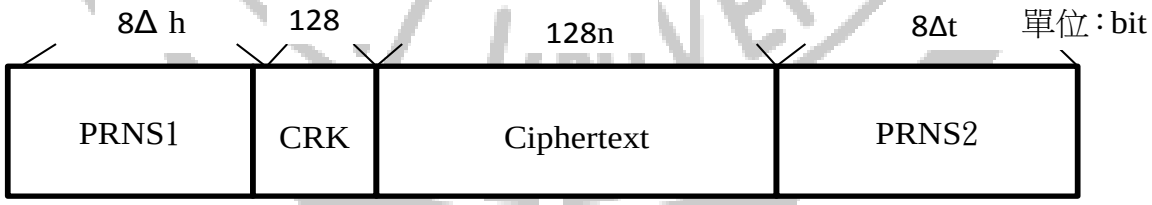

圖 6 包覆式密文檔案示意圖

3.3 解密機制

以密碼為基礎的包覆式回授加密法的解密機制，步驟一為求出 Δh 以移除 *PRNS1*，進而取出剩餘密文檔案的第一筆資料(*CRK*)，依據方程求出 *RK*，再來求出 Δt 以移除 *PRNS2*，最後根據方程式運算，解密密文，求得出明文，下面為闡述包覆式回授加密法的解密步驟與其解密流程圖。

3.3.1 解密步驟

Step1：求出Δh 以移除 PRNS1
(1)輸入 PW，透過一定的演算法則而產生密碼金鑰 K_{PW}
(2)讀取檔名、附檔名，產生 K_{FN}
(3)根據方程式(1)，計算 K_{PF1}和 K_{PF2}
(4)根據方程式(2)，計算Δh
(5)從包覆式密文檔案中移除 PRNS1

Step2：取出剩餘密文檔案的第一筆資料(*CRK*)，根據下列方程求出 RK

$$RK=[CRK-_2[(K_{PF2}\oplus ssc(6))+_2 ssc(7)]-_2 K_{PF1}]\oplus ssc(5) \quad (8)$$

Step3：求出Δt 以移除 PRNS2
(1) 根據方程式(7)，計算Δt
(2) 從剩餘的密文檔案中移除 *PRNS2*

Step4：根據下列運算，解密密文，求得明文

$b_0= K_{PF2}$

For i = 1 to n

$$P_i=[C_i\oplus((RK+_2 ssc(6))\oplus b_{i-1})-_2 RK]\oplus b_{i-1} \quad (9)$$

$$b_i=[(P_i\oplus b_{i-1})+_2 RK]+_2[(RK+_2 ssc(6))\oplus b_{i-1}] \quad (10)$$

Next i

3.3.2 解密流程圖

以密碼為基礎的包覆式回授加密法的解密流程，從下面的圖可得知一開始為求出Δh以移除 *PRNS1*，進而取出剩餘密文檔案的第一筆資料(*CRK*)，並還原 *RK*，利用 *RK* 來求得Δt以移除 *PRNS2*，最後根據方程式運算，解密密文，求得出明文。

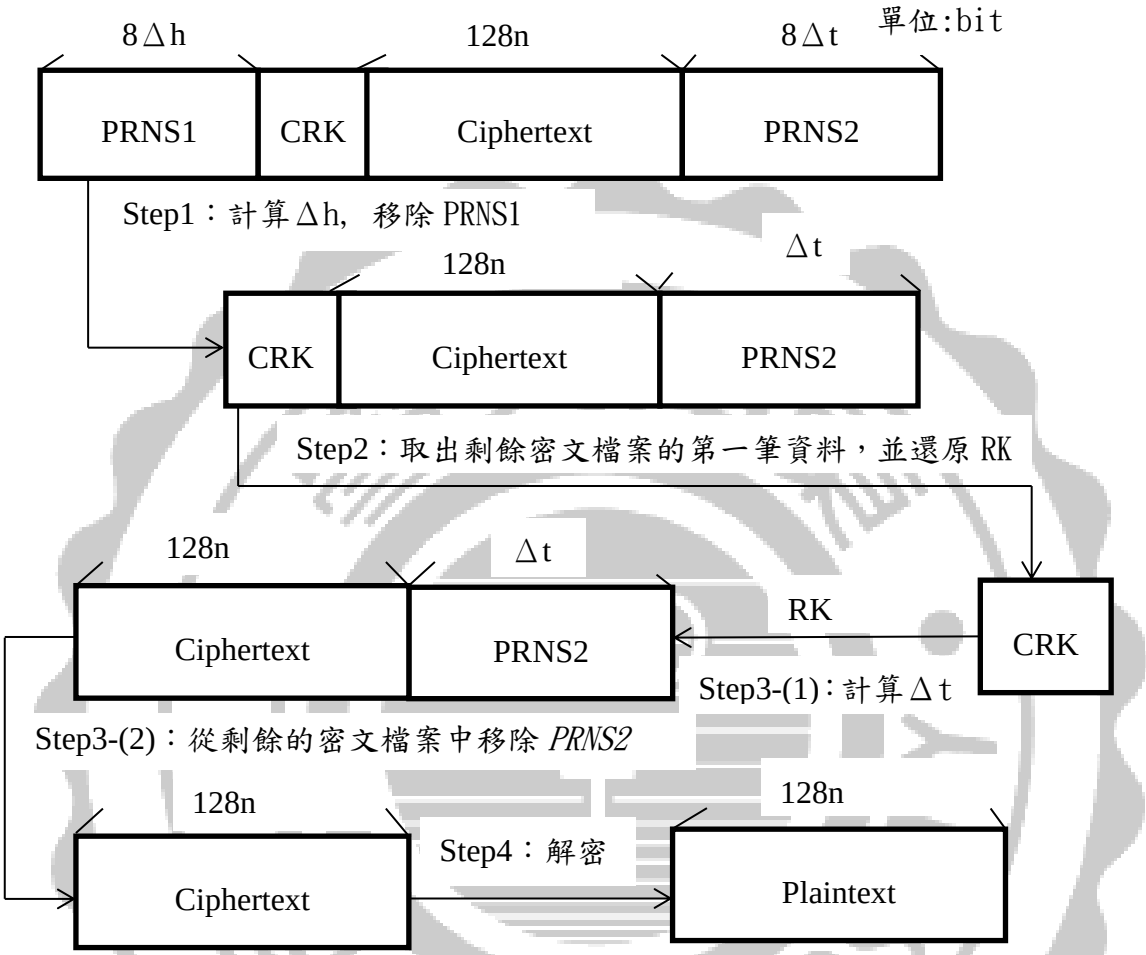

圖 7 包覆式密文檔案之解密流程圖

第四章 安全度分析與比較

本研究方法有下列特色，使本方法能夠同時保有高安全與高效能，包括：(1) 使用循序邏輯方式加/解密，如此可有效抵禦暴力法攻擊。(2) 有效整合密碼、檔名、附檔名、系統安全碼、二維運算與 mod 運算來產生密文檔的核心參數 Δh，以確保 Δh 的安全性。(3) 引進隨機亂碼金鑰與當前時間金鑰，動態的對明文加密，使得同一明文資料在不同時間點加密，會有不同的密文檔案長度與內涵，安全性因而大為提升。(4)包覆式的密文檔案結構，使得破密者不知密文的真正起始點，因而無法取得明文／密文對，大大增加了破密的困難度。(5)每個明文區塊都受到內部回授串流碼與二維運算的加密保護，使得密文區塊的安全度得以確保，也同時提升了效能。

而在訊息資料加/解密實際使用的運作模式有三種，運作模式一：為使用於無線通訊時，用戶端與基地台端間或雲端之間的通訊資料加密保護之用，此時，用戶端與基地台端之間或雲端之間的鏈結通訊金鑰 K_{CH} 被用來對當時的通訊資料進行加/解密。運作模式二：為使用者對其私有的資料檔案進行加密保護，此時，使用者輸入 PassWord(PW)，此輸入的 PW 經過一定的演算法則而產生之密碼金鑰 K_{PW} 被用來對當時的資料檔案進行加/解密。運作模式三：為使用於多筆可攜式資料檔案加/解密系統，此時，本系統使用一把固定的初始加密金鑰 K_{PW}，對許多不同資料檔案進行加/解密。

基於實際有效的考量，在本章節中，將只對本研究的方法在運作模式一與運作模式二的工作環境中進行 WFBPW 的安全性分析，因在工作模式三的環境下，因使用一把固定的初始加密金鑰 K_{PW}，對許多不同資料檔案進行加/解密，會有密碼可攜性的問題所產生，對本此方法來說密碼金鑰不能因不同人而有不同的密碼金鑰產生，會造成密碼固定，而檔名跟副檔名也為固定的參數，這樣易造成此包覆式回授加密方法的安全度降低，易受破密者以暴力法攻擊而破解，所以在此只針對對本研究的方法在運作模式一與運作模式二的工作環境中進行 WFBPW 的安全性分析，包括(1)Δh 的安全度分析，(2)動態隨機金鑰 RK 的安全度分析， (3)包覆式密文檔案的安全度分析，(4)密文資料的安全度分析，(5)本研究方法(WFBPW)與 AES 安全度比較等。

4.1 Δh 的安全度分析

本方法所產生的包覆式密文檔案結構，其破密的關鍵是在於如何求得 Δh，因只有求出正確的 Δh 才可以有效破解包覆式密文檔案結構，而取得正確的 CRK 與密文檔案資料來進行解密，所以對 Δh 有較高安全度的要求，當 WFBPW 在運作模式一或運作模式二時，所有可能破密 Δh 的方法皆相同於盲猜法，使得 Δh 達到實際安全程度，此推論結果之分析證明詳見於 Theorem 1 .

Theorem 1 ：

WFBPW 在運作模式一或運作模式二的工作環境下，由 4 個系統安全碼(*SSC(1)-SSC(4)*)與密碼金鑰 K_{PW} 產生之長度參數 Δh，可達實際安全程度。

〈*Proof*〉

破密者欲破密來求得 Δh 之值，除了盲猜法以外，破密者有二個方法來破密 Δh，首先是使用 Eq.(2)進行破密，再來是結合分析包覆式密文檔案結構與暴力法攻擊進行破密。

當 WFBPW 在運作模式一時，環境為用戶端與基地台間或環境為雲端間的通訊資料保護，此時， $K_0=K_{PW}$，或 WFBPW 在運作模式二時，環境為使用者對其私有的資料檔案之加密保護，此時， $K_0=K_{PW}$，然而，K_{PW} 為外部輸入的資料，此種資料對加密系統而言，是擁有近乎理想的高安全度，更重要的是，在此環境下，他們都是用過即丟、只用一次，所以不會有同一把 K_0 多次使用的問題，亦即由此產生的 K_{DW} 可達實際安全，在此環境下，破密者在不知道 4 個系統安全金鑰 *SSC(1)-SSC(4)* 下使用 Eq.(2)， $\Delta h=[[(ssc(1)+_2K_{PF1}) \oplus (ssc(2)+_2K_{PF2})]+_2(ssc(3)\oplus ssc(4))]$ mod 1022+3，來進行破密取得 Δh 是不可能的，因為要正確求得 Δh，需要知道密碼金鑰 K_{PW} 與 4 個系統安全碼 *SSC(1)-SSC(4)*，然而，在每個金鑰的長度為 128bit，其值的範圍為 0 ~ 2^{128}-1，此範圍遠遠大於 1022，在沒有正確的密碼金鑰 K_{PW} 和系統安全碼 *SSC(1)-SSC(4)* 情況下，經過二維運算後的金鑰值再 mod 1022 + 3 所得的 Δh 之值是很難正確猜測到的，所以與盲猜法無異。

再者破密者可以結合分析包覆式密文檔案結構與暴力法攻擊進行破密，然而因在運作模式一或運作模式二的工作環境下，破密者無法進行已知明文攻擊或選擇明文攻擊等暴力攻擊，破密者只能從其收集到的包覆式密文檔案資料進行分析破密，又因密文檔案資料是包夾於 *PRNS1* 與 *PRNS2* 之間，且 *CRK*、密文資料、*PRNS1* 與 *PRNS2* 都是亂碼，無法由外表進行分析，加上整個包覆式密文檔案長度為 ($8\Delta h$+ 128 + $128n$ + $8\Delta t$) bits，在其中 128bits 為 *CRK* 長度，而 $128n$ bits 為明文長度，Δt 為 *PRNS2* 的長度，對破密者而言，Δh、n 與 Δt 等三個變數都不知道，因此，要由分析整個包覆式密文檔案結構求出 Δh 之值亦是與盲猜法相同，所以 Δh 可達實際安全。

又或是破密者可以將同一已知明文檔案資料，給予不同的檔名、副檔名進行破密攻擊，然而由於包覆式密文檔案結構的保護，使得破密者只能得到 $\Delta h+\Delta t$ 的值，然而 Δh 之值會因不同的檔名、副檔名而改變，Δt 之值又會因動態隨機金鑰 *RK* 的作用而隨機改變，如此 Δh 之值仍然受到良好的保護，也就是說，Δh 具有相當高的安全度。

4.2 動態隨機金鑰 *RK* 的安全度分析

隨機金鑰 *RK* 是明文加密的主要加密種子參數，是由 Eq.(3)， $RK=(R_{K0}+_2K_{CT})\oplus(R_{K0}\oplus K_{RCT})$ 中顯示 *RK* 的產生是由隨機初始金鑰 RK_0 與當前時間金鑰 K_{CT} 與 K_{RCT} 經由二維運算

所產生。在 WFBPW 中，即使對不同的明文輸入相同初始加密金鑰 K_{PW} 下，每筆明文檔案資料將會擁有不同的隨機金鑰 RK 對其加密，而每次將會有效地破壞前後筆密文間因相同的 K_{PW}，而使安全度大增，且因 RK 一直持續在改變，為一動態隨機金鑰，將擁有極高的安全度，以下 Theorem 2 證明在 WFBPW 中隨機金鑰 RK 是具有高度的安全性。

Theorem2:

WFBPW 在運作模式一或運作模式二的工作環境下，若系統產生之動態隨機金鑰 RK 長度為 n bit，則要從包覆式密文檔案中求出正確 RK 之值的機率為$\frac{1}{2^n}$.

〈*Proof*〉

若破密者想破密來求得正確的 RK，除了盲猜法以外，破密者還有三個方法來破密 RK，第一是單獨使用 Eq.(4)進行破密，再來是合併使用 Eq.(5)與 Eq.(6)進行破密，後者則是統合使用 Eq.(7)與分析包覆式密文檔案結構以求得 RK。

本方法用於運作模式一或運作模式二時，由前述的分析結果顯示，Δh 可達實際安全，在不知道 Δh 條件下，破密者將無法正確地由包覆式密文檔案取出 CRK，而在不知道正確 CRK 條件下，Eq.(4)將失去其解密功能，即代表，若要由 Eq.(4)來反求 RK 將相同於盲猜法，即使，破密者猜到 CRK，接著分析 Eq.(4)， $CRK=[(RK\oplus ssc(5))+_2K_{PF1}]+_2[(K_{PF2}\oplus ssc(6))+_2ssc(7)]$，以求出動態隨機金鑰 RK，然而，實際的情況是，在每次攻擊時取得的 CRK 之值，都會有一個隨意變化的動態隨機金鑰 RK 的加密貢獻，使得每一筆 CRK 資料均只對應一把隨機金鑰 RK，且前後筆(CRK,RK)對之間是沒有關聯的，在不知道 $SSC(5)$-$SSC(7)$之值的情況下，要由一筆 CRK 資料求出其對應之 RK 與盲猜法無異，故其破密機率為$\frac{1}{2^n}$。

再者，因在運作模式一或運作模式二的工作環境下，破密者無法進行已知明文攻擊或選擇明文攻擊等暴力攻擊，破密者只能從其收集到的包覆式密文檔案資料進行分析破密，在此情況下，破密者不知道 Δh、密文長度與 Δt 之值，所以也就不知道密文在包覆式密文檔案中的正確位置，將無法正確地由包覆式密文檔案取出密文資料(C_i，$1\leqq i\leqq n$)以進行破密，Eq.(5)與 Eq.(6)將失去其解密功能，亦即，要由 Eq.(5)與 Eq.(6)反求 RK 將相同於盲猜法，即使，破密者猜到密文資料(C_i，$1\leqq i\leqq n$)，接著分析 Eq.(5)與 Eq.(6)，然因 Eq.(6)， $C_i=[(P_i\oplus b_{i-1})+_2RK]\oplus[(RK+_2ssc(6))\oplus b_{i-1}]$，中之內部回授串流金鑰 b_{i-1}，$1\leqq i\leqq n$,是由 Eq.(5)， $b_i=[(P_i\oplus b_{i-1})+_2RK]+_2[(RK+_2ssc(6))\oplus b_{i-1}]$產生，在破密者不知道 P_1、b_0、C_0、RK 與 $SSC(6)$之值情況下，就無法求得 b_1，在不知道 P_2、b_1、RK 與 $SSC(6)$之值情況下，就無法正確求得 b_2，不知道 P_3、b_2、RK 與 $SSC(6)$之值情況下，就無法求得 b_3，…， 依此類推可得，此 b_i，$0\leqq i\leqq n$，是破密者無法求得且安全的內部回授串流碼，將此結果帶入 Eq.(6)中可清楚表示，在對每一個 i 而言，雖然破密者知道 C_i 與 C_{i-1}, 但在不知道 P_i、b_{i-1}、RK 與 $SSC(6)$之值情況下，破密者將不知道$[(P_i\oplus$

b_{i-1}) +$_2$ RK]與[(RK+$_2$$ssc(6)$)⊕ b_{i-1}]之值，故要由 C_i之值求得正確之[(P_i⊕ b_{i-1}) +$_2$ RK]與[(RK+$_2$$ssc(6)$)⊕ b_{i-1}]之值與盲猜法同[25]，亦即，隱藏於其中動態隨機金鑰 RK的是安全的，所以由此方向欲求出正確 RK之機率為 $\frac{1}{2^n}$與盲猜法相同。

最後，在不知道 $SSC(1)$、$SCC(2)$、$SSC(8)$與 RK的情況下，破密者將無法由 Eq.(7)中來產生 Δt，又在不知道 Δh與密文長度條件下，破密者將無法由包覆式密文檔案長度來反求其 Δt之值，亦即表示 Δt是安全的。即使破密者猜到 Δt之值，並結合使用 Eq.(7) ， Δt =[($ssc(1)$+$_2$$RK$)⊕ $ssc(8)$+$_2$($ssc(2)$⊕ RK)]mod 1022+3 來進行破密分析，以便求出動態隨機金鑰 RK，然而，實際上，每次攻擊時 Δt之值的產生，都會有一個隨意變化的動態隨機金鑰 RK的貢獻，使得每一筆 Δt資料均對應一把隨機金鑰 RK，且因 3≦Δt≦1025，Δt之值變動範圍僅 1K 是遠遠小於動態隨機金鑰 RK變動範圍(0≦RK≦2^{128}-1)，而且破密者亦不知道($ssc(1)$+$_2$$RK$)，$ssc(8)$與($ssc(2)$⊕ RK)之值，所以在此情況下要由 Δt資料求出其對應 RK之機率為 $\frac{1}{2^n}$與盲猜法相同[25]。

4.3 包覆式密文檔案的安全度分析

本研究方法的密文檔案示意圖如圖 6 所示，在其中密文檔案資料被包裹在長度Δh的 $PRNS1$和長度Δt的 $PRNS2$之間，由於密文檔案資料、$PRNS1$與 $PRNS2$均是亂碼，所以無法從其內容來分辨何者是密文檔案資料，跟哪裡是 $PRNS1$與 $PRNS2$，所以破密者必須求出正確的Δh或Δt，才可有效的拆解此包覆式密文檔案，以得到正確的密文檔案資料，然而如前面所述，Δh與Δt都受到良好的保護，皆是破密者非常不易取得的參數資料，所以密文檔案資料受到良好的保護，使得破密者無法有效取得明文/密文對，大大增加了破密的困難度。

4.4 密文資料的安全度分析

本研究方法的密文檔案資料有下列兩個安全機制所保護著，首先是密文檔案資料被安全地包裹在 $PRNS1$和 $PRNS2$之間，使得破密者無法有效取得明文/密文對以進行破密的工作，即使破密者收集到了明文/密文對，然而依據 Eq.(9)要由(P_i, C_i)反求出三把解密金鑰 RK、ssc(6) 與 b_{i-1}，幾乎是不可能達到的，這是因為動態隨機金鑰 RK在每次明文檔案加密時都會隨機改變，而回授串流碼金鑰 b_{i-1} 會因 P_i與 RK的不同而逐次改變，由此可知本研究方法產出的密文資料具有相當高的安全度。

在運作模式一或運作模式二的工作環境下，破密者雖有包覆式密文檔案資料卻無法有效取得正確的密文檔案資料，即使破密者可以猜到密文檔案資料，而明文檔案資料確仍是安全的，Theorem 3 將證明 WFBPW 中明文檔案資料的安全度。

Theorem 3 :

令明文檔案資料為 $P_1P_2P_3\cdots P_m$，密文檔案資料為 $C_1C_2C_3\cdots C_m$，每個明文區塊 P_i 與密文區塊 C_i 的長度為 n 位元，$1\leqq i\leqq m$，當 WFBPW 在運作模式一或運作模式二的工作環境下，則由密文檔案資料 $C_1C_2C_3\cdots C_m$ 破密求得正確明文檔案資料為 $P_1P_2P_3\cdots P_m$ 的機率為 $\left(\frac{1}{2^n}\right)^m$.

<Proof>:

在 Eq.(9)中， $P_i=[C_i\oplus((RK+_2ssc(6))\oplus b_{i-1})-_2RK]\oplus b_{i-1}$ 提供由密文區塊 C_i 求得明文區塊 P_i 的有效方法，由此方程式指出要得到 P_i 需要有 C_i、$SSC(6)$、b_{i-1} 和 RK 等資料，由 Theorem 1 證明 K_5 是被安全保護的，由 Theorem 3 證明 RK 是被安全保護的，而要取得 b_{i-1} 的資料則需要透過 Eq.(10)， $b_i=[(P_i\oplus b_{i-1})+_2RK]+_2[(RK+_2ssc(6))\oplus b_{i-1}]$，亦即須要 P_{i-1}、b_{i-2}、$SSC(6)$與 RK 等資料，明文區塊 P_{i-1} 破密者是不知道的，而 $b_0=K_{PF2}$，被安全保護的，所以 $b_1=[(P_1\oplus b_0)+_2RK]+_2[(RK+_2ssc(6))\oplus b_0]$ 是被安全保護，同樣地，在破密者無法有效取的 P_2、b_1 與 RK 的情況下， $b_2=[(P_2\oplus b_1)+_2RK]+_2[(RK+_2ssc(6))\oplus b_1]$ 仍是被安全保護著，依此類推，內部回授串流碼(b_{i-1}，$1\leqq i\leqq n$)，是被安全保護的，將此結果帶入 Eq.(9)我們可得， $P_1=[C_1\oplus((RK+_2ssc(6))\oplus b_0)-_2RK]\oplus b_0$ 是安全的，因在不知道 C_1、$SSC(6)$、b_0 與 RK 的條件下，經二維運算保護的 P_1 其破密機率為 $\frac{1}{2^n}$[25]與盲猜法同，又 $P_2=[C_2\oplus((RK+_2ssc(6))\oplus b_1)-_2RK]\oplus b_1$ 是安全的，因在不知道 $SSC(6)$，b_1 與 RK 的條件下，經二維運算保護的 P_2 其破密機率為 $\frac{1}{2^n}$[25]與盲猜法同，依此類推，每個明文區塊 P_i，$1\leqq i\leqq$m 都是被安全保護的，其個別破密機率均為 $\frac{1}{2^n}$，依據 Rule of Product，整筆明文資料 $P_1P_2P_3\cdots P_m$ 的破密機率為 $(\frac{1}{2^n})^m$.

4.5 本研究方法(WFBPW)與 AES 安全度比較

以下表一中詳列了不同的已知破密攻擊法，並比較本法 WFBPW 與 AES 之安全度與其說明。

表一 AES 與 WFBPW 針對已知攻擊之安全度比較

方法 / 攻擊方式	AES	WFBPW	說明
Parallel decryption	Yes	No	WFBPW 為動態循序邏輯加密方式，AES 為組合邏輯加密方式。

Brute force attack	Middle	High	因 WFBPW 為動態循序邏輯加密方式，所在暴力法攻擊下，組合邏輯加密方式的 AES 較容易被破解。
Known plaintext/ ciphertext attack	Middle	High	因 WFBPW 之明文/密文對會隨著動態金鑰的變動，而有不同的明文/密文，所以導致破密者不易收集 WFBPW 之明文/密文對，而破密者可以有效收集 AES 之明文/密文對。
Linear attack	Middle	High	WFBPW 動態的對明文/密文加密，使得同一明文資料在不同時間點加密，會有不同的密文檔案長度與內涵，大大增加了破密的困難度，相較於 AES 來說，安全度較高。
Security	Middle	High	由以上幾種的攻擊法比較來看，由於 WFBPW 是動態循序邏輯加密方式，且明文/密文加密回隨著時間做動態的變動而有不同的明文/密文，使得同一明文資料在不同時間點加密，會有不同的密文檔案長度與內涵，相較於 AES 為組合邏輯加密方式，且明文/密文皆由同樣的方式所產生，所以 WFBPW 較安全。

第五章 效能分析與比較

5.1 AES 之效能分析

AES 在執行中主要是作了一回子金鑰的產生跟 10 回的工作模式，在初次使用 AddRoundKey 運算對明文加密之後是 9 次回合的加密運算，每一回合加密包含下列四種運算，即 SubBytes、ShiftRows、MixColumns 與 AddRoundKey 運算，最後一回運算省略 MixColumns 運算，即只有 SubBytes、ShiftRows 和 AddRoundKey。

5.1.1 加密效能之分析

檔案資料加/解密方法的效能，主要決定於加/解密過程中計算量的多寡，表二中詳列了 AES 對於 128 位元明文區塊，在加/解密過程中的各種運算數量。

表二 AES 對於 128 位元明文區塊，在加/解密過程中的各種運算數量

方 法	運算數量(加 密)	運算數量(解 密)
AES (128-bit，10 回合工作模式)	(AddRoundKey) 176 ⊕s (8 bits)	ddRoundKey、SubBytes、ShiftRows 三個階段相同於加密時所需的數量一樣。 (MixColumns) 36 Rijndael columns mixing[15](128 bits)。(通常，解密過程的操作往往比相應的加密過程複雜些)
	(SubBytes) 160 Substitutions (8 bits)	
	(ShiftRows) 30 ShiftRows(128 bits)[27]	
	(MixColumns) 36 Rijndael columns mixing[15](128 bits)	

5.1.2 子金鑰產生效能之分析

子金鑰產生效能效能，主要決定於子金鑰產生過程中計算量的多寡，表三中詳列了 AES 對於 128 位元明文區塊，在子金鑰產生過程中的各種運算數量。

表三 AES 子金鑰產生工作的各種運算次數

方法	運算次數	說明
AES	(50 ⊕s) + (10 SubWords) + (10 RotWords)	子金鑰產生工作

5.1.3 AES 加/解密效能分析

AES 加/解密效能分析包含一回子金鑰的產生跟 9 回的工作模式，為更清楚呈現兩者之間得效能，我們以個人電腦為測試平台，其規格為 CPU：Intel i7-3770 3.40GHz，RAM：16GB，Platform：Windows 7, 64-bit，進行模擬測試。

表四 AES 對一個 128 位元明文區塊加/解密時間

Operation / Method	Time consumed (μs)	
	Encryption	Decryption
AES	10.492	10.667
AES(Key-Expansion)	3.211	3.211

5.2 WFBPW 之效能分析

WFBPW 之效能分析主要是包含加/解密的過程跟前置處理(包含 3.3.2 Initial process, 3.3.3 Encryption process-Step 2 and the generation of Δt)與後段處理(包含 PRNS1 與 PRNS2 的產生)。

5.2.1 加密效能之分解

檔案資料加/解密方法的效能，主要決定於加/解密過程中計算量的多寡，表五中詳列了 WFBPW 對於 128 位元明文區塊，在加/解密過程中的各種運算數量。

表五 WFBPW 對於 128 位元明文區塊，在加/解密過程中的各種運算次數

方 法	運算次數(加 密)	運算次數(解 密)
WFBPW (128-bit)	(3 ⊕s) + (3 +2s)	(4 ⊕s) + (3 +2s) + (1 −2)

5.2.2 前置處理效能之分析

前置處理效能效能，主要決定於前置處理過程中計算量的多寡，表六中詳列了 WFBPW 對於 128 位元明文區塊，在前置處理過程中的各種運算數量。

表六 WFBPW 對於 128 位元明文區塊，在前置處理工作各種運算次數

方法	運算次數	說明
WFBPW	產生金鑰: KPW, KFN, RK0, KCT, KRCT (10 ⊕s) + (10 +2s) + (2 mods) + (2 +s) + (1 PRNG)	前置處理工作

5.2.3 WFBPW 加/解密效能分析

WFBPW 之效能分析主要是包含加/解密的過程跟前置處理(包含 3.3.2 Initial process, 3.3.3 Encryption process-Step 2 and the generation of Δt)與後段處理(包含 PRNS1 與 PRNS2 的產生)，為更清楚呈現兩者之間的效能，我們以個人電腦為測試平台，其規格為 CPU:Intel i7-3770 3.40GHz，RAM:16GB，Platform:Windows 7, 64-bit，進行模擬測試。

表七 WFBPW 對於 128 位元明文區塊，在加/解密過程中的各種運算時間

Operation / Method	Time consumed (μs)	
	Encryption	Decryption
WFBPW	0.336	0.447
WFBPW(前置處理與後段處理)	25.774	17.472

5.3 WFBPW 與 AES 之效能比較

WFBPW 與 AES 之效能分析比較由 5.1AES 之效能分析與 5.2WFBPW 之效能分析的內容可知，檔案資料加/解密方法的效能，主要決定於加/解密過程中計算量的多寡，表八中詳列了 WFBPW 與 AES 對於 128 位元明文區塊，在加/解密過程中的各種運算數量。

表八 WFBPW 與 AES 對於 128 位元明文區塊，在加/解密過程中的各種運算次數

方 法	運算數量(加 密)	運算數量(解 密)
AES (128-bit,10回	(AddRoundKey) 176 ⊕s (8 bits)	AddRoundKey、SubBytes、 ShiftRows 三個階段相同於加密

合工作模式)	(SubBytes) 160 Substitutions (8 bits)	時所需的數量一樣。
	(ShiftRows) 30 ShiftRows(128 bits)[27]	(MixColumns) 36 Rijndael columns mixing[15](128 bits)。(通常，解密過程的操作往往比相應的加密過程複雜些)
	(MixColumns) 36 Rijndael columns mixing[15](128 bits)	
WFBPW (128-bit)	(3 $\oplus$s) + (3 $+_2$s)	(4 $\oplus$s) + (3 $+_2$s) + (1 $-_2$)

由表八資料顯示，WFBPW 對每一明文區塊(128 bits)的加/解密運算量遠少於 AES 的加/解密運算，其效能遠優於 AES，為更清楚呈現兩者之間得效能比，我們把上述的時間整理成表九。

表九 AES 與 WFBPW 的加/解密時間

Operation / Method	Time consumed (μs)	
	Encryption	Decryption
AES	10.492	10.667
WFBPW	0.336	0.447

表九清楚指出對一個明文區塊加密時間,WFBPW 是 AES 的 31.26 倍,而解密時間 WFBPW 是 AES 的 23.88 倍，然而，AES 在進行區塊加密前，需執行一次 Key-Expansion，即由 Parent-Key 來產生加密時的子金鑰(Sub-Keys)，同樣地，WFBPW 在進行區塊加密前，需執行一次，前置處理(包含 3.3.2 Initial process, 3.3.3Encryption process-Step 2 and the generation of Δt)與後段處理(包含 PRNS1 與 PRNS2 的產生)，這兩項資料的時間測試如表十所示。

表十：AES(Key-Expansion)與 WFBPW(前置處理與後段處理)的加/解密時間

Operation / Method	Time consumed (μs)	
	Encryption	Decryption
AES(Key-Expansion)	3.211	3.211
WFBPW(前置處理與後段處理)	25.774	17.472

表十中,WFBPW 解密時,其前/後段處理時間小於 WFBPW 加密時其前/後段處理時間，主要是因解密運算時不需要產生 PRNS1 與 PRNS2 之故。

假若目前有 n 個 128 bits 的明文區塊資料要加/解密，則綜合了表九與表十資料，我們可以得到 WFBPW 與 AES 的模擬費時如下：

AES 加密的模擬費時

=（子金鑰產生的模擬費時）+（每個明文區塊加密模擬費時）* n

=（表十中 AES 的模擬費時）+（表九中 AES 加密的模擬費時）* n

= 3.211 + 10.492 * n (μs) (11)

AES 解密的模擬費時= 3.211 + 10.667 * n (μs) (12)

WFBPW 加密的模擬費時

=（前置處理與後段處理的模擬費時）+（每個明文區塊加密模擬費時）* n

=（表十中 WFBPW 的模擬費時）+（表九中 WFBPW 加密的時模擬費時）* n

= 25.774 + 0.336* n (μs) (13)

WFBPW 解密的模擬費時= 17.472 + 0.447 * n (μs) (14)

當明文資料為 1024 bits 時，則 n = 8，由 Eq(11)與(13)可得：

$$\frac{\text{AES 加密費時}}{\text{WFBPW 加密費時}} = \frac{87.147}{28.462} = 3.062 \approx 3$$

由 Eq(12)與(14)可得：

$$\frac{\text{AES 解密費時}}{\text{WFBPW 解密費時}} = \frac{88.574}{21.048} = 4.208 \approx 4$$

當明文資料為 128Kb 時，則 n = 1000，由 Eq(11)與(13)可得：

$$\frac{\text{AES 加密費時}}{\text{WFBPW 加密費時}} = \frac{10495.211}{361.774} = 29.010 \approx 29$$

由 Eq(12)與(14)可得：

$$\frac{\text{AES 解密費時}}{\text{WFBPW 解密費時}} = \frac{10670.211}{464.472} = 22.973 \approx 23$$

所以，加密效能 WFBPW 是 AES 的 3 ~ 29 倍，而解密效能 WFBPW 是 AES 的 4 ~ 23 倍，對於大於 128Kb 的明文檔案資料，WFBPW 大約 23 倍 AES 的效能，是明顯優於 AES 的。

第六章 結論與未來展望

本研究方法以使用者密碼為基礎，用包覆式回授加密法建構出一具有高安全、高效能的包覆式密文檔案，採用輸入的密碼或輸入的通道金鑰作為加密的啟動金鑰，並使用動態旋轉置換盒加密技術與二維運算加密技術來產生系統加密所需之子金鑰群，也引進了隨機亂碼金鑰與當前時間金鑰，動態的對明文加密，使得同一明文資料在不同時間點加密，會有不同的密文檔案長度與內涵，安全性因而大為提升；又密文資料被包裹在長度Δh 的 PRNS1 和長度Δt 的 PRNS2 之間，且密文資料、PRNS1 與 PRNS2 均是亂碼，破密者不知密文資料的起始位置，因而無法有效取得明文/密文對，大大增加了破密的困難度；再者，每個明文區塊都受到內部回授串流碼(b_{i-1})與二維運算的加密保護，密文區塊的安全度得以確保，且因所有加密運算都只執行一次，使得效能得以大大提升，理論定性分析指出，本方法在安全性與效能皆優於 AES，對於較大型的明文資料檔案，本法效能約 23 倍於 AES 的效能，是而本法比 AES 更適合用於雲端與用戶間之資料傳輸保護。

現今科技的科技計算的能力愈來愈強，有朝一日此系統可能遭受到破解，在未來我們希望朝著完全隨機的路線來改良，因在此包覆式回授法當中並非完全隨機的狀態，而這當中使用了密碼、檔名和附檔名來做為密碼金鑰跟檔名金鑰為固定參數來進行加密的過程中，在未來希望可以不要使用到任何的固定參數來進行加密，以達到完全隨機的狀態，使得安全度更為提升，在密碼的方面，目前此方法在忘記密碼的情況下就會發生無法順利加解密資料檔案的不便之處，在未來希望可以讓使用者即使是忘記密碼的狀態下，也可以利用系統本身的啟動碼用來取代原本密碼來執行系統，以便讓使用者在忘記密碼的情況下，也能順利把加解密的資料檔案解密來執行使用，而不會因忘記密碼，就無法順利的把檔案系統作加解密的工作，造成執行上的不便。

參考文獻

[1] COMPUTERWORLD, 3G vs. 4G: Real-world speed tests. http://www.computerworld.com/article/2511923/wireless-networking/3g-vs-4g-real-world-speed-tests.html

[2] Eli Biham and Adi Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," Journal of Cryptology, Vol. 4, No.1, pp. 3-72, 1991.

[3] Eli Biham and Adi Shamir, "Differential Cryptanalysis of the Data Encryption Standard," Springer Verlag, ISBN 0-387-97930-1, pp. 1-188, 1993.

[4] Wiki, the EFF DES cracker. http://en.wikipedia.org/wiki/EFF_DES_cracker.

[5] Andrey Bogdanov, Dmitry Khovratovich and Christian Rechberger, "Biclique Cryptanalysis of the Full AES," Springer Berlin Heidelberg , ISBN 978-3-642-25384-3, pp. 344-371, 2011.

[6] Yi-Li Huang, Fang-Yie Leu, Jian-Hong Chen, Cheng-Chung Chu and Chao-Tung Yang, " A True Random-Number Encryption Method," IEEE International Conf. Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 654-659, 2013.

[7] Yi-Li Huang, Fang-Yie Leu and Ko-Chung Wei, "A secure communication over wireless environments by using a data connection core," Mathematical and Computer Modelling, Vol. 58, No. 5-6, pp. 1459-1474, 2013.

[8] National Institute of Standards and Technology, "Advanced Encryption Standard," National Institute of Standards and Technolog, 2001.

[9] JeongSoo Park, KiSeok Bae, YongJe Choi, DooHo Chio and JaeCheol Ha, "A Fault-Resistant AES Implementation Using Differential Characteristic of Input and Output," Journal of Internet Services and Information Security, Vol. 2,No.3-4, pp.93-109, 2012.

[10]Federal Information Processing Standards Publication 197, "Announcing the Advanced Encryption Standard (AES)," 2001.

[11]Elad Barkan and Eli Biham, "In How Many Ways Can You Write Rijndael?," International Conf. on the Theory and Application of Cryptology and Information Security , Vol. 2501,pp. 160-175,2002.

[12]Joan Daemen and Vincent Rijmen, "AES Proposal: Rijndael," The Frist Advanced Encryption Standard Candidate Conf., pp.1-45,2000.

[13]Joan Daemen and Vincent Rijmen, "The Design of Rijndael: AES - The Advanced Encryption Standard," Information Security and

Cryptography, ISBN 3-540-42580-2, 2002.
[14] William Stallings, “Cryptography and Network Security Principles and Practice,” ISBN 978-986-154-481-6, 2013.
[15]Daniel.J. Bernstein, “Cache-timing attacks on AES,” 2005.
[16]Eran Tromer, Dag Arne Osvik and Adi Shamir, “Efficient Cache Attacks on AES, and Countermeasures,” Springer Verlag, Vol. 23, pp 37-71, 2010.
[17]Joan Daemen and Vincent Rijmen, “Understanding two-round differentials in AES,” International Conf. Security and Cryptography for Networks, pp. 78-94, 2006.
[18]Biryukov Alex, Priemuth-Schmid Deike and Zhang Bin, “Analysis of SNOW 3G resynchronization mechanism,” IEEE International Conf. Security and Cryptography, pp. 1-7, 2010.
[19]National institute of Standards and Technology, “Data Encryption Standard,” Federal Information Processing Standards,pp. 7-46, 1999.
[20]Amit Dhir, “Data Encryption using DES/Triple-DES Functionality in Spartan-II FPGAs,” XILINX, 2000.
[21]Nelson Gonzalez, Charles Miers, Fernando Red´ıgolo, Marcos Simpl´ıcio, Tereza Carvalhol,Mats Naslund ” and Makan Pourzandi, “A quantitative analysis of current securityconcerns and solutions for cloud computing,” Journal of Cloud Computing, 2012.
[22]Yi-Li Huang, Fang-Yie Leu and Cheng-Ru Dai, “A secure data encryption method employing a sequential-logic style mechanism for a cloud system,” International Journal of Web and Grid Services, Vol. X, No. X, pp. XX-XX, 2014.
[23]Yi-Li Huang, Fang-Yie Leu, Jung-Chun Liu and Jing-Hao Yang, “A Block Cipher Mode of Operation with Two Keys” International Conf. ICT-EurAsia, pp. 392-398, 2013.
[24]Chen Fei, Wu Kehe, Chen Wei and Zhang Qianyuan, “The Research and Implementation of the VPN Gateway Based on SSL” IEEE International Conf. Computational and Information Sciences, pp. 1376-1379, 2013.
[25]Yi-Li Huang, Fang-Yie Leu, Ilsun You, Yao-Kuo Sun and Cheng-Chung Chu, “A secure wireless communication system integrating RSA, Diffie-Hellman PKDS, intelligent protection-key chains and a Data Connection Core in a 4G environment,” Journal of Supercomputing, Vol. 67, No. 3,pp. 635-652, 2014.
[26]Yi-Li Huang and Fang-Yie Leu, “Constructing a Secure Point-to-Point Wireless Environment by Integrating Diffie-Hellman PKDS RSA and

Stream Ciphering for Users Known to Each Other," Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, Vol. 2, No. 3, pp. 96-107, 2011.

[27]Lingguo Cui and Yuands Cao, "A New S-Box Structure Named Affine-Power-Affine," International Journal of Innovative Computing、Information and Control, Vol. 3, No. 3, pp. 751-759, 2007.