防駭客之高安全伺服器設計及實現

指導老師：顏錦柱

詹哲瑜

競賽類別：資通類

摘要：

本作品主要解決資訊儲存及傳輸的安全問題。在目前網際網路發展的現代資料外洩事件已達 140 億筆每秒約有 76 筆的技術資產外洩，而造成這結果的最大原因是因為現今網路資料庫的資料保存全都是明文，駭客可以透過攻擊資料庫進行竊取資料的動作，為解決此問題，我們提出基於混沌理論之創新網路資料加密技術，結合本團隊發明專利中的混沌同步技術，以動態金鑰概念，配合傳統 AES 的加密方式加密資料庫內部的資料。此外，我們也將融合此加密技術設計出展示用的私密安全聊天室。

且本技術有別傳統使用固定金鑰的方式，將隨機動態金鑰的產生與還原都交由混沌系統處理，使其具有極高強度的安全性，同時利用混沌同步技術，不必預設且不會暴露於網路傳輸公共通道，可自動在傳送及接收端出現並使用，具有極高的安全性，相較傳統 AES ECB 和 AES CBC 金鑰被破解後整筆資料就被攻破，本系統以隨機動態金鑰加密，除保留原 AES 的高安全性外，也因動態金鑰，即使單筆金鑰被破解，破解者只能得到資料其中一區塊的明文，而不是整筆資料都被破解。再者，為確保本技術的安全性，本團隊將混沌產生之動態金鑰及密文以 NIST SP 800-22 測試，其數據結果顯示 15 項結果皆為通過且多數優於傳統之 AES ECB 和 AES CBC 之表現，確認及驗證本創新網路資料加密技術優越的安全性。

關鍵字：混沌系統、網路伺服器、AES 加密、動態金鑰、防駭客

一、前言(或其他相關標題)

根據「台灣 2017 企業雲端應用趨勢調查」，台灣已經有將近 85%的企業搬上雲端，對於那些不堪負荷或無法負荷的企業或新創公司而言，雲端平台應用對於降低實體與維護成本能夠提供有效的幫助，在安全性上也有一定層面上的保護，使得企業紛紛使用雲端平台，來減少 IT 管理資源的開支，也根據「IDC」在 2013 年的研究報告指出，全球雲端服務的市值已大幅度的達到一兆三千億的產值，並在未來會持續的向上攀升，但正如大家所知的，互聯網是一個毫無章法的世界，總是有許多有心人士瞄準那些具有高價值性的資訊資產，在高的防火牆總是有人能夠突破，在 2015 年四月「中華民國資料保護協會」發布了國內最常見的 10 大個資外洩的方式，因此在雲端平台的使用上固然方便，但企業也得對自家的資料做好保護與加密的責任。在把整個資安架構底下，一層一層撥開，最底層也是最重要的就是資料本身，而在此通常是使用加解密的方式來構築資安架構的最後一道防線。

本專題作品就是對此著墨，目前已知最佳的安全演算法有 AES、RSA、ECC 等等，但在未來的量子電腦時代，我們幾乎可以肯定這些加密方式將無法抵擋量子電腦的高運算速度，因此，在 2017 年 4 月，在台大電機系博理館舉行了一場「混沌理論的資訊安全應用」研討會，邀請了美國國家標準局 NIST 退休研究員 Richard Kautz、日本京都大學的梅野健教授與台科大劉聰德教授前來參與。在這場研討會中，學者們除了介紹混沌的研究歷史，也指出，混沌加密技術將使得量子電腦也難破解，其因為，混沌系統可以在短時間內產生大量的隨機訊號，又因為混沌系統的隨機不可預測性，這些大量的隨機訊號是動態的、更新時間又快，所以混沌加密技術可能為量子電腦時代下的解決方式之一。

二、研究目的(或其他相關標題)

混沌加密技術的原理為利用混沌系統的複雜行為來將原始訊號藏匿在其中。Rössler 在 1979 年提出了超混沌系統的概念[1]，超混沌系統有多個正 Lapunov 指數的奇異吸引子，其中錯綜複雜的特性應用在保密通訊領域上具有極優異的效果。因此，我們想到如果用來作為加密使用可以提升安全性，但如果只是單單擁有隨機狀態的加密，要達成解密，則仍必需仰賴混沌

同步控制技術。在 1990 年，O.G.Y（Ott, Grebogi and Yorke）[2]提出了”控制混沌”的相關研究，研究指出在混沌系統中，只要微小調整參數值，混沌行為就會產生巨大且不可預測的變化，因此，可藉由此適度的參數調整來控制或抑制混沌行為；隨後，Pecora 和 Carroll 兩位學者試著將兩個不同初始值的混沌系統進行同步，使其具有相同的隨機狀態響應[3]。自此之後，便有很多的研究，專注於混沌系統的同步的研究[4，5]，混沌系統的同步研究正好可以應用於本計劃規劃的同步動態金鑰上面，我們可以藉由同步控制達到還原當初動態金鑰的目的。

但在先前的技術中，混沌系統的設計，大多是透過固定之類比元件來實現，但類比元件容易因外在因素（溫度、濕度、老化）失真等問題，造成系統的不穩定甚或失效。為改善此限制並有效提昇系統之安全性，將數位化系統取代固定之類比元件來實現已成為未來趨勢。另一方面，在早期的混沌保密系統的設計技術中能產生的隨機亂數較少，然而在探討高安全性的要求下，當需要加密的資料數較為龐大時亂數的數量過少會無法提供加密機制有更多的隨機數可做選擇，這導致加密機制需花較多的時間與無法一次完成加密等問題。

為了改善上述問題，我們利用數位化超混沌 Henon map 來設計動態金鑰產生器，其中的特點除了使用超混沌系統外，還結合 SHA-256 利用其雪崩效應將原本 3 個狀態的 Henon map 透過 SHA-256 函式擴充成 256x3 個狀態，提供 AES 加密所需求的金鑰長度，以達到提升加解密系統的安全性與降低加密所需時間之目的。

三、原理與分析(或其他相關標題)

一、 傳輸加密端:

(一)、 HENON MAP 超混沌系統的動態分析及同步設計

該階段目標是完成動態金鑰產生器的動態分析與設計[6]，如圖 2 所示，超混沌 HENON MAP 系統的奇異吸子具有兩個向量的動態隨機分布，且具有既不收斂也不發散的特性，讓其無法預測的特性作為 AES 加密系統的動態金鑰，可有效提升安全等級。

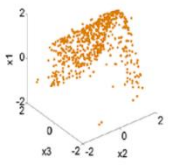

圖 2 HENON MAP 超混沌奇異吸子

超混沌 HENON MAP 系統描述如下:

$$x_1(k+1) = 1.76 - x_2{}^2(k)\text{-}0.1\, x_3(k)$$
$$x_2(k+1) = x_1(k)$$
$$x_3(k+1) = x_2(k)$$

為使改良式 AES 具有動態的金鑰，主僕混沌系統同步控制器的設計是其關鍵，主要技術說明如下：

我們將超混沌 HENON MAP 系統分為主端與僕端

我們將超混沌 HENON MAP 系統分為主端與僕端

主端的系統如下:

$$x_1(k+1) = 1.76 - x_2{}^2(k)\text{-}0.1\, x_3(k)$$
$$x_2(k+1) = x_1(k)$$
$$x_3(k+1) = x_2(k) \quad (1)$$

僕端的系統如下:

$$y_1(k+1) = 1.76 - y_2^2(k) - 0.1y_3(k) + u(k)$$
$$y_2(k+1) = y_1(k)$$
$$y_3(k+1) = y_2(k) \quad (2)$$

定義動態誤差:

$$e_i(k) = y_i(k) - x_i(k), i = 1,2,3 \quad (3)$$

經上述式子，我們可得到誤差動態方程式如下:

$$e_1(k+1) = x_2^2(k) - y_2^2(k) - 0.1e_3(k) + u(k)$$
$$e_2(k+1) = e_1(k)$$
$$e_3(k+1) = e_2(k) \quad (4)$$

欲使主僕混沌系統(1)(2)達到同步，必須要設計一個強健的同步控制器，本計畫使用的滑動模式轉換面設計如下：

$$s(k) = e_1(k) + c_1e_2(k) + c_2e_3(k) \quad (5)$$

假設$s(k) = 0$(進入轉換面)，便能得到：

$$e_1(k) = -c_1e_2(k) - c_2e_3(k) \quad (6)$$

將上述(10)式寫成矩陣形式：

$$E(k+1) = \begin{bmatrix} -c_1 & -c_2 \\ 1 & 0 \end{bmatrix} \bullet \begin{bmatrix} e_2(k) \\ e_3(k) \end{bmatrix} = AE(k) \quad (7)$$

由上式可知，若將c_1, c_2選定為特定值，A的特徵根將會侷限於單位元中，$i.e, |\lambda_i(A)| < 1$，則可以確保$e_2, e_3 = 0$，同時由$s(k) = 0$可知$e_1 = 0$，即可確保誤差收斂至零，轉換面的設計對於混沌同步來說相當重要，如果設計不好將會造成主僕兩者數值發散，在確認滑動模式下，誤差動態的穩定性後，接著我們討論如何設計控制器，使系統可進入滑動模式。

為了確保系統能夠達到轉換面，控制器的設計如下:

$$\Delta S_k = s(k+1) - s(k)$$
$$= x_2^2(k) - y_2^2(k) - 0.1e_3(k) + c_1e_1(k) + c_2e_2(k) - e_1(k) - c_1e_2(k) - c_2e_3(k) + u(k) \quad (8)$$

令控制器為:

$$u(k) = x_2^2(k) - y_2^2(k) - 0.1e_3(k) + c_1e_1(k) + c_2e_2(k) - e_1(k) - c_1e_2(k) - c_2e_3(k) + \alpha s(k) \quad (9)$$

代入後可得:

$$s(k+1) - s(k) = \alpha s(k) \quad (10)$$

由（10）式可得到 $s(k+1) = (\alpha+1)s(k)$，如果我們選擇適當的α，使 $|\alpha+1| < 1$，即可確保系統將順利進入滑動模式中而系統將順利進入$s(k) = 0$滑動模式中，並由上述步驟一，可知誤差收斂至零可完成同步。

由上述步驟，完成混沌同步設計。接下來我們進行模擬測試上述同步控制的正確性，模擬時，主僕系統的初始值為$x_1 = 0.5, x_2 = -0.3, x_3 = 0.4, y_1 = -0.3, y_2 - 0.1, y_3 = 0.8$，同步控制器的參數值為 $c_1 = -1.700, c_2 = 0.720, \alpha = -0.5, A$的特徵根為(0.9,0.8)。模擬結果如圖 3 所示。由上述模擬結果，我們可以發現，不管是主端與僕端的波形與動態誤差，還是轉換面、控制器，與主僕端奇異吸子，都可如上述推論，如預期的達到同步和收斂。

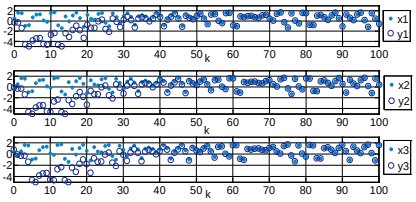

圖 3 同步誤差響應

動態金鑰產生器設計:

由上述可知，一組 HENON MAP 超混沌系統能產生三個不可預測的隨機亂數，在加解密系統中，金鑰的位元數越大，代表更加難以破解，AES 進階加密技術就有三種不同加密位元數的方式，分別是 128、192、256 三種，由此可知以位元數來說單一組超混沌系統在金鑰產出方面略顯不足，為解決此為題本計畫將使用將其中一組混沌亂數丟入 SHA256 雜湊函式，使得金鑰產出的長度符合本計畫所需共 256bits，其動態金鑰設計如圖 4 所示，因 SHA256 雪崩效應[7]的關係，每筆動態金鑰皆不會相同。

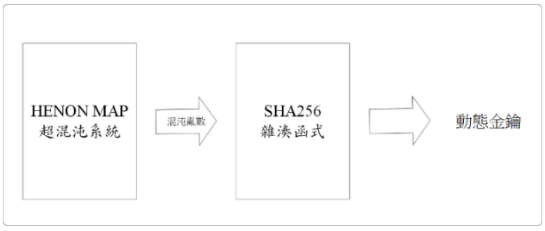

圖 4 動態金鑰產生器系統架構

而在同步方面，為考慮實際應用的安全性，我們分成主僕端，並將所設計之同步控制器分解為$u = F(f_m, f_s)$ 的型式，其中 f_m，f_s 分別表示由主端及僕端所提供之合成訊號，經函數 F 組合後可得到如(9)式所示的控制器，即可完成此階段主僕同步之設計目標。

接著將上述技術整合即可完成圖 1 情境架構圖中的傳輸加密端，而傳輸加密端的技術組成方式如圖 5。

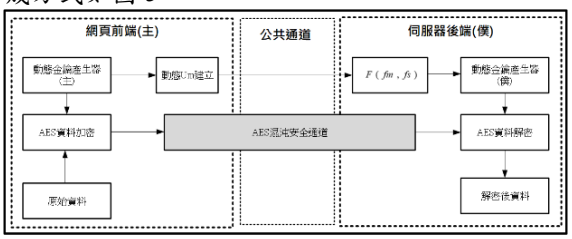

圖 5 傳輸加密端架構圖

二、 資料加密端混沌狀態與單邊同步控制器設計

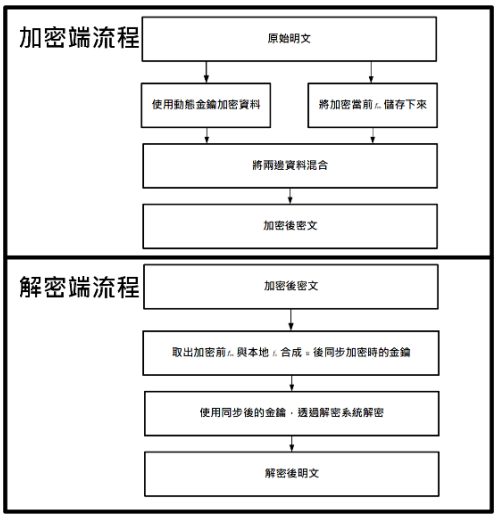

圖 6 資料加解密端流程圖

這裡與前面傳輸加密端的需求不太一樣，由於通訊時有兩端混沌系統同時配合所完成的，而在資料加密部分由於不能獲得加密時所持有的f_m(不可能邊加密邊解密)，不過由上敘述所知若要設計同步控制需要同時有f_m與f_s，而f_s是由本地端提供的，那只要能取得加密前的f_m，我們就能使用本地所產生的f_s使混沌系統同步回加密之前的狀態，進而取得動態金鑰。藉由上述的方法設

計，我將主端的動態產生器加密當下的f_m值與使用者的密碼做混合產出加密後的f_m值最後與需要加密資料一併整合，之後只要有需要解密就能從資料裡拿取之前的f_m值去做同步，就可以得到之前加密的金鑰了，具體加解密步驟如上圖 6 所示。

三、 將加密技術與伺服器整合

在網站架設中，為了方便**展示**及除錯我們將聊天網站分成兩個部分，聊天網站本體及加密用的後端伺服器，聊天網站本體只做資料顯示與儲存透過 call API 的方式與後端加密伺服器通訊

1.聊天伺服器本體

後端的部分我們將使用 Node.js[8]來架設 Web Server，並做為後端程式語言，其中使用 Express 框架協助後端的開發。在提供給使用者操作的前端則是使用 HTML5/CSS3 來建置版面，並使用 bootstrap3 來做版面美化，且運用 JavaScript 來做為前端的運算處理，在前端基本顯示版面撰寫完成後，將使用 JavaScript 來撰寫混沌系統於網站前端。

2.混沌加密伺服器

再加密伺服器方面我們選用現在比較熱門的科學計算用的 Python[9] 透過 Python Flask 架設輕量級的 API 伺服器[10] 將上述提到的傳輸加密端與資料加密端，構建在 API 伺服器上面，提供給聊天伺服器加密及通訊的需求。

四、軟硬體系統(或其他相關標題)

本計劃目的在於設計架構於混沌理論之創新網路資料加密方式，解決目前網路資料外洩問題已提升資訊安全的目標，並以此為方向實際應用此技術設計出安全伺服器及私密聊天系統作為展示。預計完成之技術及應用系統，如下圖系統架構圖所示:

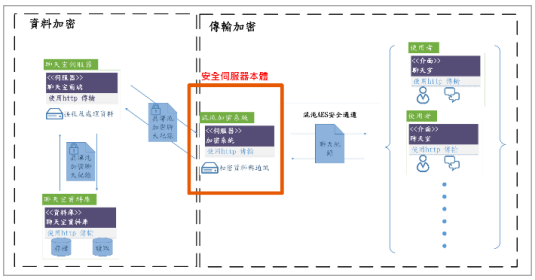

圖 7　系統架構圖

由上圖 7 可以看到架構圖本體分為兩大部分，**傳輸加密端**服務於使用者與混沌伺服器的溝通，而**資料加密部分**提供使用者加密資料，到此階段資料已經被混沌系統加密過，使用者可以安心放入資料庫或做其他操作，此時就算公司網路被駭客攻破盜走了資料庫的資料也無須擔心。

傳輸加密端架構解說:

由於在請求加密的途中會經過公共通道，為了防止駭客的盜取，我們在與使用者溝通前會經過混沌同步，建立起混沌安全通道，以確保資料安全性，我們採用傳統混沌同步的方式，而同步方面，為考慮實際應用的安全性，我們分成主僕端，並將所設計之同步控制器分解為$u = F(f_m, f_s)$ 的型式，其中 f_m，f_s 分別表示由主端及僕端所提供之合成訊號，經函數 F 組合後可得到如(9)式所示的控制器，即可完成此階段主僕同步之設計目標。

圖 8 傳輸加密端架構圖

資料加密端架構解說:

資料加密端主要提供資料庫內部資料的保護，有些駭客會透過防護能力較弱的資料庫做為攻擊，而我的安全伺服器能有效解決該問題，所有存放在資料庫中的資料接受道混沌動態金鑰的保護。

而資料加密端主要利用到單邊同步的與混沌系統動態金鑰產生器的技術，提供使用者資料加密的需求，該加密端的實際架構如下圖，將使用者資料透過動態金鑰產生器的金鑰，使用 AES256 進行加密後，與單邊同步裝置的混合 Um 進行封裝，接著便可在解密端同步出相同金鑰進行解密。

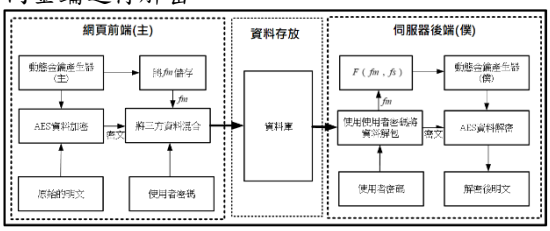

圖 9 資料加密端架構圖

本作品的架構能徹底解決掉現今網際網路上駭客入侵的可能性，我們將所有流通在網路上的資料流都進行了加密，從通訊安全到資料安全都獲得混沌加密的保障，而混沌理論又是現今公認安全度很高的一種理論。

四、實驗結果與比較(或其他相關標題)

作品展示:

本專題作品的作品展示圖如下圖 10 所示，照片中的左方是使用者 A 的電腦，透過網際網路存取聊天室與右邊使用者 B 的電腦聊天，其過程的所有通訊資料皆受到中間的混沌加密伺服器的保護。

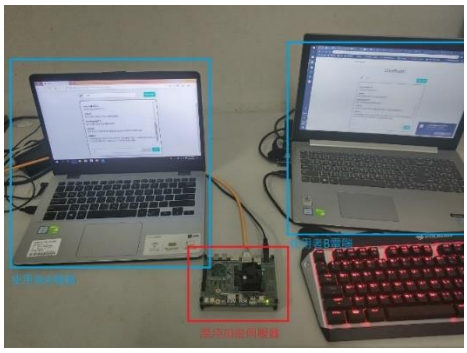

圖 10 成品實際展示

具體的保護方式呈現如下圖 11 所示，從圖 11 的左邊可以看到使用者 A 與使用者 B 的聊天室介面因有使用正確的密碼，可以發現聊天室內部的資料都是正確的聊天內容，而由圖的右邊可以發現因沒有打上正確的密碼而使聊天內容變得不可被閱讀。

圖 11 聊天室介面展示

有些駭客可能會繞過聊天室本身，轉而透過攻擊資料庫獲取未被加密的明文本體，在此方面我們的伺服器也可以保障存放資料庫內部的資料安全，由下圖 12，可以看到放在 firebase 資料庫內部的所有資料，都是經過混沌伺服器加密的密文，縱使駭客有幸攻破了 firebase 的驗證獲取了內部的資料，得到的也是一堆未經過解密的密文，並不影響資料本身的安全性。

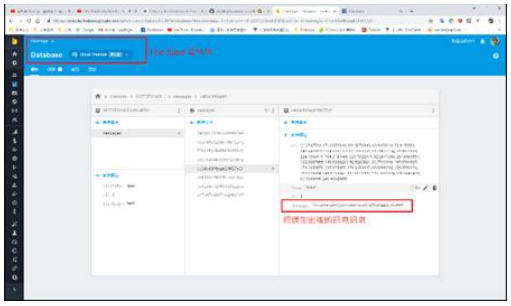

圖 12 firebase 資料庫內部展示

最後該聊天室是功能展示使用應用，實際使用可依照使用需求，透過 HTTP API 的方式訪問伺服器進行相關的加解密應用，不只能使用在聊天室上，像是近期阿里巴巴 0 元訂單的漏洞也可以靠安全伺服器解決。

混沌系統狀態同步驗證:

如前述的理論介紹，金鑰產生器是以超混沌 Henon map 系統為核心，加以改寫而來的設計，我們可以在式子(1)得知亂數產生器擁有三個隨機狀態，我們使用監視視窗來做顯示，同步驗證結果如下圖 13 所示。

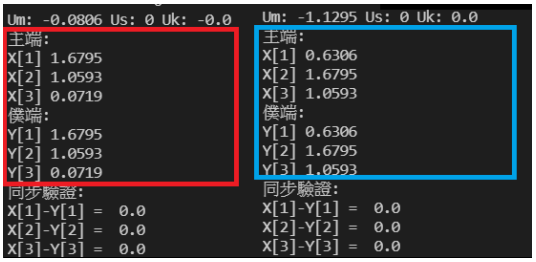

圖 13 亂數序列數與同步展示

安全性分析:

在做完了同步設計及驗證後，對於混沌動態金鑰產生器隨機性的亂度不能只由我們來保證，所以我們選擇使用公信力最高的美國國家標準暨技術研究院 NIST SP 800-22[11]來檢測及比較我們亂數產生的安全性分析結果如下表所示:

	P-VALUE		
	AES-ECB	AES-CBC	CHAOS-AES
1. Frequency	0.000000(F)	0.468595(P)	0.602458(P)
2.BlockFrequency	0.000954(F)	0.534146(P)	0.602458(P)
3. Cumulative Sums	0.000001(F)	0.949602(P)	0.602458(P)
4. Runs	0.000000(F)	0.407091(P)	0.991468(P)
5. Longest Run	0.000000(F)	0.534146(P)	0.739918(P)
6. Rank	0.000000(F)	0.534146(P)	0.350485(P)
7. FFT	0.000000(F)	0.468595(P)	0.407091(P)
8. Non Overlapping Template	0.000000(F)	0.998205(P)	0.991468(P)
9. Overlapping Template	0.000439(F)	0.000737(F)	0.407091(P)
10. Universal	0.000000(F)	0.804337(P)	0.862344(P)
11.Approximate Entropy	0.000000(F)	0.407091(P)	0.862344(P)
12. Random Excursions	0.122325(P)	0.484646(P)	0.350485(P)
13. Random Excursions Variant	0.025193(P)	0.980883(P)	0.637119(P)
14. Serial	0.000000(F)	0.350485(P)	0.911413(P)
15. Linear Complexity	0.000000(F)	0.299251(P)	0.299251(P)

表一 NIST 安全性測試

在表一中可以發現我們使用的 CHAOS-

AES 其數據結果顯示 15 項結果皆為通過且多數優於傳統之 AES ECB 和 AES CBC 之表現，確認及驗證本創新網路資料加密技術優越的安全性。

六、結論

在近幾年各大企業資安問題層出不窮的情況下，確保企業機密、客戶資訊不被洩漏是最根本的資安需求，而大量增設資安設備或相關軟體往往會對中小型企業造成資金上的困難，或是管理上不易等問題，本系統除了建置成本低廉以外，企業的 CMS 系統也只需透過請求 HTTP API 的方式就可以使資料加上一層加密保護，在安全性方面本系統使用 HENON MAP 超混沌系統結合 sliding mode 同步技術產生可還原狀態的動態金鑰，再將此動態隨機生成的金鑰帶入 AES256-ECB 加密系統中進行資料的加解密，使其安全性能夠提升至更高的層次，本系統的技術亮點如下:

(1). 系統功能全都透過 API 的方式使用：容易與其他相關系統結合使用，且可將系統放置於網際網路中，提供給對資料儲存安全有需求的使用者使用。

(2). 改良現有的進階加密標準 AES 加密系統：有別於傳統使用固定金鑰的方式，將隨機動態金鑰的產生與還原都交由混沌系統處理，使其具有極高強度的安全性。

(3). 不同於傳統固定金鑰概念，本技術利用混沌同步技術，不必預設且不會暴露於網路傳輸公共通道，可自動在傳送及接收端出現並使用，具有極高的安全性

(4). 系統金鑰產生處於一種混沌動態產生的狀態：系統可高速同時產生巨量的隨機且不重複之金鑰，當使用者請求時則會將當下連續數個狀態的金鑰進行加密處理。

(5). 混沌亂數產生通過 NIST SP 800-22 隨機亂數測試：由於辨識加密系統好壞的其中一項指標是其產生的隨機性，在此本團隊將混沌產生之動態金鑰加以做 NIST 測試，其數據結果顯示 15 項隨機測試結果皆為通過，得以驗證本系統亂數產生具有一定水準。

(6). 相較 AES ECB 和 AES CBC 金鑰被破解後整筆資料就被攻破，本系統改良 AES 由固定金鑰加密方式，改以隨機動態金鑰加密，除保留原 AES 的高安全性外，也因動態金鑰，所以無法進行暴力破解， 同時，即使單筆金鑰被破解也無傷大雅，不同於 AES，破解者只能得到資料其中一區塊的明文，而不是整筆資料都被破解。

七、參考文獻

[1] E. Rössler, An equation for hyperchaos, Physics Letters A, 1979; 71: pp.155-157.

[2] E. Ott, C. Grebogi, J. A. Yorke, Controlling Chaos, Phys Rev Lett ,1990;64: pp.77-80.

[3] LM. Pecora, TL. Carroll, Synchronization in chaotic systems. Phys Rev Lett ,1900; 64: pp.821-824.

[4] M.C. Pai, Global synchronization of uncertain chaotic systems via discrete-time sliding mode control, Applied Mathematics and Computation, 2014; 227: pp.663–671.

[5] J.J. Yan, C.Y. Chen, J.S.H. Tsai, Hybrid chaos control of continuous unified chaotic systems using discrete rippling sliding mode control, Nonlinear Analysis: Hybrid Systems, 2016; 22: pp. 276-283

[6] 萬培彥, 超混沌巨量亂數產生器設計與應用, 樹德科技大學，碩士論文,2018; pp.12-30.

[7] W. Penard, T. V. Werkhoven, On the Secure Hash Algorithm family (https://web.archive.org/web/20141014172403/http://www.staff.science.uu.nl/~werkh108/docs/study/Y5_07_08/infocry/project/Cryp08.pdf)

[8] JavaScript 大全, 6/e (JavaScript: The Definitive Guide: Activate Your Web Pages, 6/e） , David Flanagan 著、黃銘偉 譯 , 美商歐萊禮股份有限公司台灣分公司

[9] 從零開始學Python程式設計(適用Python 3.5以上）, 李馨 著, 博碩文化股份有限公司

[10] 精通 Python：運用簡單的套件進行現代運算, Bill Lubanovic 著, 賴屹民 譯, 歐萊禮

[11] A. Rukhin, et al. A statistical test suite for random and pseudorandom number generators for cryptographic applications. National Institute of Standards and Technology, Special Publication 800-22 (2001).