行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※
※ ※
※ **設計及實現非決定性的 MAC** ※
※ ※
※ **Design and Implementation of Non-deterministic MAC** ※
※ ※
※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：■個別型計畫 □整合型計畫
計畫編號：NSC－89－2213－E－009－156
執行期間：89 年 08 月 01 日至 90 年 07 月 31 日

計畫主持人：葉義雄 副教授
共同主持人：林祝興 副教授

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

執行單位：國立交通大學資訊工程系

中 華 民 國 90 年 7 月 31 日

行政院國家科學委員會專題研究計畫成果報告
設計及實現非決定性的 MAC
Design and Implementation of Non-deterministic MAC

計畫編號：NSC 89-2213-E-009-156
執行期限：89 年 08 月 01 日至 90 年 07 月 31 日
主持人：葉義雄　　執行機構及單位名稱 :國立交通大學資訊工程系

一、中文摘要

在本篇論文中，我們提出利用 SHA-1 及 AES(Rijndael) 建立訊息認證碼的方法，其中 AES 可以接受 128, 192， 或是 256 位元長度的加密金鑰。在產生訊息認證碼的過程中，並藉由加入一個隨機變數，以防止中間雜湊值的洩漏。此方法對於雜湊函數的要求並不需要很高，而整個方法的安全性是建立於我們採用的區塊加密法 AES。在處理很長的訊息時，本方法提供了有效率的軟體實作方式，並且擁有清楚而明確的安全性質。

關鍵詞：AES, SHA-1, MAC, NMAC

Abstract

In this thesis, we suggest a new method to construct Message Authentication Code with SHA-1 and AES(Rijndael), with key of 128, 192, 256 bits. In the process of MAC construction, we add a random number to prevent the problem of leaking the intermediate hash value. The security of the whole scheme is based on the block cipher, AES. That is even if the adopted hash function is not collision free or one-way, the scheme is still secure. This method provides an efficient software implementation to process long messages and has clear security properties.

Keywords: AES, SHA-1, MAC, NMAC

二、緣由與目的

近年來電腦在我們的每日生活方面已經變得非常重要，因為它的快速處理速度和大資料儲存體容量。在許多領域中，它扮演一個重要的角色，例如商務、教育和科學。在一部多使用者的電腦系統中，管理人的責任就是避免系統資訊和私人資料遭受未經許可的公開、修正和破壞。當網際網路逐漸成長的時候，人們利用它交換資料，因此該如何維持機密資料的機密變成一個必要的議題。

資料安全是一門在電腦和溝通系統中如何保護私人和秘密的資料之技術。基本上我們可將密碼系統分類如下：

1. 單向雜湊函數：對於一單向雜湊函數 H(M)，對於給定任意長度的訊息M。它會傳回固定長度的雜湊值，h。
   h=H(M)

另外單向雜湊函數必須滿足下列條件：

   a. 給定訊息M，很容易計算出h。
   b. 給定h，很難計算出訊息M，使得 H(M)=h。
   c. 給定訊息M，很難找到另一個訊息M’，使得H(M)=H(M’)。
   d. Collision-resistance：很難找到任易兩個訊息 M 與 M’，使得 H(M)=H(M’)。

2. 對稱型加密系統：假若一個加密系統有兩個轉換函數encryption（加密函數）與decryption（解密函數），若這兩個函數所需使用的金鑰為K_e和K_d，若很容易從K_e去推算出K_d，反之亦然，則我們稱為對稱型加密系統。
3. 非對稱型加密系統[1,3]：對稱型加密系統：假若一個加密系統有兩個轉換函數encryption（加密函數）與decryption（解密函數），若這兩個函數所需使

用的金鑰為K_e和K_d，若很困難從K_e去推算出K_d，反之亦然，則我們稱為非對稱型加密系統。

密碼系統所欲達成的目標[1,2]：

1. 機密(confidentiality)：它是不可能的讓違法的接收者讀明文。
2. 認證(authentication)：它可讓接收者確定他所接收的訊息的來源。不可讓侵入者假裝。
3. 完整性(data integrity)：它可讓接收者證實訊息在傳輸的過程中沒被修改過。

無可否認性(non-repudiation)：一個發送人不能否認他之前所送的訊息。執行行政院國家科學委員會(以下簡稱國科會)專題研究計畫，在結案時主持人均需繳交完整的研究成果報告，國科會並訂有一統一格式以供撰寫的參考[1]。除此以外，有些學門也訂有適當格式，要求主持人據此繕打增送精簡報告，並將之編訂成冊，分送同學門其他研究人員參考與保存，發揮了很大的學術交流效果，普獲學界好評。

三、研究報告應含的內容

我們會給兩種類型的 NMAC，一種是使用 SHA-1，一種是使用 AES，其 key 的長度為 128, 192, 或256 bits。MAC會被計算成如下的一個訊息 X $MAC = E_k(R\|PART\text{-}SHA\text{-}1(R\|X))$, 這裏的 R 是一個隨機選取的數字，Ek 是 AES 的加密函數，而 X 的長度是有限的(非無窮盡的)，但是並沒有限制多長，而 k 是key 的長度可為 128, 192, 或256 bits。

3.1 SHA-1-AES-MAC(類型 I)

3.1.1 MAC 的產生

我們提出的第一種類型有三種輸入， 訊息 X , key K , 還有 user 自己選定的區塊的長度 (block size) T, 可為128, 192, 或256 bits，有三個部份: 一個隨機位元產生器(random bits generator)，一個 one-way hash 函數SHA-1，一個區塊密文 AES。首先，隨機位元產生器產生一個 *(T+1)*32* 位元的隨機數 R，然後我們將 X 附加在 R 的後面，變成 "R‖X"，然後 "R‖X" 變成 SHA-1 的輸入，產生一個 hash 數，因為 SHA-1 可以妥善的處理任何有限的數，所以我們可以不用擔心 " R‖X" 的長度有多長，之後將 hash 出來的值的前面 (T+1)*32 bits 截取出來，然後附加到 R的後面，更具體的來說，當 T 為 1(2,3)，也就是區塊的長度為 128(192,256)，SHA-1所產生的 hash 值的前面 64(86,128) bits 會被附加到 64(96,128) bits R 的後面，而這輸入的 key 被用來當作是 Rijndael 的 key，最後，Rijndael 的輸出就是 MAC。

當給定一個訊息 X，它的 MAC 將會被計算成如下，MAC= $E_k(R\|PART\text{-}SHA\text{-}1(R\|X))$, SHA-1 一定會產生一個長 160 bits的hash 數值, 但是在這裏只有 (T+1)*32 bits 會被用到，下圖描述出這整個架構:

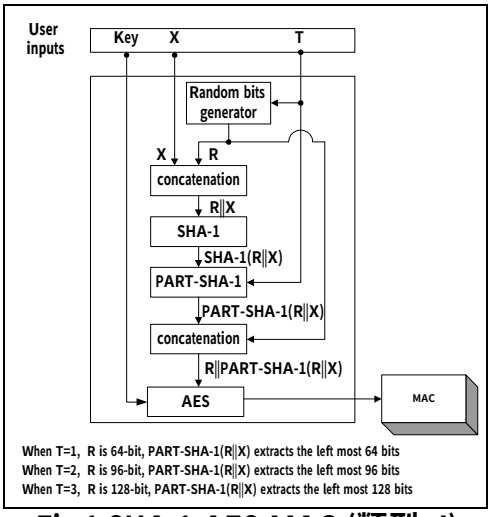

Fig 1 SHA-1-AES-MAC (類型 I)

如何產生 MAC 的區塊長度 (block size) 的演算法如下 :

```
Input: Message X, Cipher Key K and
User-specified Block Length T
Output: MAC
Process:
Plaintext P
    // the plaintext for the block cipher AES
    (Rijndael) Generate a [(T+1)*32]-bit
    random number R (by random bit
    generator) M=R||X, || mean
    concatenation//
    H=SHA-1(M)
    switch (T){
        case 1:
            for(i=0; i<128; i++){
                if(i<64) P[i]=R[i];
                else P[i]=H[i-64]; // get
```

```
                    the first 64 bits of H
                } break;
        case 2:
                for(i=0; i<192; i++){
                    if(i<96) P[i]=R[i];
                    else  P[i]=H[i-96];  //  get
                    the first 96 bits of H
                    } break;
        case 3:
                    for(i=0; i<256; i++){
                    if(i<128) P[i]=R[i];
                    else  P[i]=H[i-128];  //  get
                    the first 128 bits of H
                    } break;
    }
MAC= E_K(P)  // the encryption of AES
```

3.1.2 MAC 的驗証

因為傳送者和接收者已經協調過AES 的 key 了，接下來，當接收者接收到訊息X 和 MAC 的時候，就可以以下列的步驟來驗証是否所接收到的訊息是正確的。

步驟1 ：將MAC 解密成一個中間的hash 數值 $R \| PART-SHA-1(R \| X)$ ，如 $D_k(MAC)$ = $D_k(E_k(R \| PART-SHA-1(R \| X)))$ = $R \| PART-SHA-1(R \| X)$，其中的Dk () 是AES 的解密。

步驟2: 將 *R* 和 *h*= $PART-SHA-1(R \| X)$ 從步驟1的中間的hash 數值取出來，在我們的NMAC 中有一點需要注意的是 *R* 和 *h* 的長度是相等的，所以說在從步驟1所得到的中間hash 數值的前面的一平是 *R* ，後面的一半是 *h*，事實上我們也必須要知道h 的長度，因為這個訊息在步驟3 將要用到。

步驟 3: 接收者藉著算出 $PART-SHA-1(R \| X)$ ，然後來檢查是否和 *h* 一樣來驗証訊息的正確與否。

SHA-1-AES-MAC(類型1) 驗証的演算法:

```
Input: Message X, MAC, Cipher Key K
Output: YES/NO
Process:
    I[0…N-1]  = D_K(MAC) ;      //D is
    decryption of AES, N is length o
    intermediate hash value//
    for (i=0; i<N/2; i++)     R[i] = I[i];
    for (i=N/2; i<N; i++)   H_1[i] = I[i];
    H=SHA-1(R||X);
    for (i=0; i<N/2; i++)   H_2[i] =H[i];
    if( H_1 == H_2 ) output YES;
    else output NO;
```

3.2 SHA-1-AES-MAC (類型II)

3.2.1 MAC 的產生

此處的MAC 的不特定的區塊長度和之前所提到的MAC 非常的相似，這裏的區塊長度T 也不是由使用者選定的，而是由一個隨機變數而決定，它有兩個輸入：訊息 X，和 key K，還有三個部份: 一個隨機位元產生器(random bits generator)，一個 one-way hash 函數SHA-1，一個區塊密文 AES。首先隨機位元產生器產生一個隨機變數 Dice，而 T 的值就是(Dice mod 3)+1，然後隨機位元產生器又產生一個 (T+1) *32 bits 的數值 R，我們把 X 附加在 R 的後面變成 "R||X"，接下來 "R||X" 就被拿來當作 SHA-1 的輸入，如此產生一個 hash 數，然後這個 hash 數的前面 (T+1)*32 bits 被抽取出來然後附加到 R 的後面，而輸入的 key 被用來當作 Rijndael 的 key，最後， Rijndael 的輸出就是 MAC。

和在 6.1 節所描述的一樣，當給定一個訊息 X，它的 MAC 將會被計算成如下， MAC= E_k *(R‖PART-SHA-1(R‖X))*，SHA-1 一定會產生一個長 160 bits的hash 數值，但是在這裏只有 (T+1)*32 bits 會被用到，下圖描述出這整個架構 :

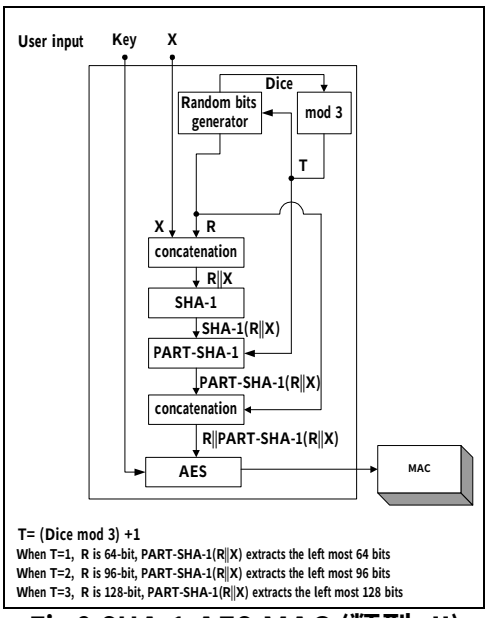

Fig 2 SHA-1-AES-MAC (類型 II)

如何產生 MAC 的 區塊長度 (block size) 的演算法如下 ：

Input: Message *X*, Cipher Key *K* and User-specified Block Length *T*
Output: *MAC*
Process:
Plaintext P // the plaintext for the bloc cipher AES Generate a random number *Dice* (by random bits generator) //
T = (*Dice* mod 3) +1

3.2.2 MAC 的驗証

和 3.1.2 節中所講的 MAC 驗証是一樣的 。

四、研究成果與自評：

相關研究發表如下：

1. Construct Message Authentication Code with AEA and SHA1: Submitted to Information Processing Letter。
2. A new scheme of constructing message authentication code with SHA-1 and AES: Submitted to 第十屆國防科技學術研討會。
3. Construct MAC with SHA-2 and RSA: Submitted to Journal of Information & Optimization Sciences。

五、參考文獻

[1] Bruce Schneier, "Applied Cryptography", John Wiley & Sons, Inc., 1996.

[2] 賴溪松、韓亮、張真誠, "近代密碼學及其應用", 松岡電腦圖書資料股份有限公司, 臺北, 1995.

[3] D. E. Denning, "Cryptography and Data Security", Addison-Wesley Publishing Company, Inc, U.S.A., 1982.

[4] National Institute of Standards and Technology, NIST FIPS RFC RIN 0693-ZA-42, "Announcing Development of a Federal Information Processing Standard for Advanced Encryption Standard," U.S. Department of Commerce, Jan. 1997.

[5] National Institute of Standards and Technology, NIST FIPS RFC RIN 0693-ZA-42, "Announcing Draft Federal Information Processing Standard (FIPS) for the Advanced Encryption Standard (AES) and Request for Comments", U.S. Department of Commerce, Feb. 2001,http://csrc.nist.gov/encryption/aes/draftfips/fr-AES-200102.html.

[6] National Institute of Standards and Technology, NIST FIPS PUB 180-1, "Secure Hash Standard," U.S. Department of Commerce, Apr. 1995.

[7] "The MD4 message Digest Algorithm," Advances in Cryptology – CRYPTO '90 Proceedings, Springer-Verlag, 1991, pp. 030-311.

[8] National Institute of Standards and Technology, NIST FIPS PUB 81, "DES Modes of Operation," U.S. Department of Commerce, Dec. 1980.

[9] National Institute of Standards and Technology, NIST FIPS PUB 113, "Computer Data Authentication," U.S. Department of Commerce, May 1985.

[10] I. Verbauwhede, F. Hoornaert, J. Vander-walle, H. De Man, and R. Govaerts, "Security Considerations in the Design and Implementation of a New DES Chip," Advances in Cryptography-EUROCRYPT '87 Proceedings, Springer-Verlag. 1988, pp.287-300.

[11] M. Bellare, R. Canetti, and H. Krawczyk, "Keying Hash Functions for Message authentication," Advances in Cryptology-CRYPTO '96, pp.1-15, Springer-Verlag, 1996.

[12] A. J. Menezes, P.C.V. Oorschot, and S.A. Vanstone, "Handbook of Applied Cryptography," CRC Press, 1997.

[13] Y.S. Yeh and C.C. Wang, "Construct Message Authentication Code with One-Way Hash Functions and Block Ciphers," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Feb. 1999

[14] R. Michael, "Performance of Symmetric Ciphers and One-way Hash Functions", Fast Software Encryption, Cambridge Security Workshop, Proceedings, p.83-89, Dec. 1993.