國立交通大學

電機資訊學院資訊學程

碩士論文

嵌入式 Word 編輯器之數位簽章系統開發

Digital Signature Plug-in for Microsoft Word

研 究 生：趙聰龍

指導教授：劉振漢 教授

中華民國九十三年六月

嵌入式 Word 編輯器之數位簽章系統開發
Digital Signature Plug-in for Microsoft Word

研 究 生：趙聰龍　　Student：Tsung-Lung Chao

指導教授：劉振漢　　Advisor：Jenn-Hann Liou

國 立 交 通 大 學

電機資訊學院 資訊學程

碩 士 論 文

A Thesis

Submitted to Degree Program of Electrical Engineering

Computer Science

College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in

Computer Science

June 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

嵌入式 Word 編輯器之數位簽章系統開發

學生：趙聰龍　　指導教授：劉振漢

國立交通大學電機資訊學院資訊學程

摘　　要

隨著個人電腦及網際網路的普及應用，促使電子商務的應用範疇的突飛猛進,現在人們的買賣方式可以不再受到時間、空間及對象的限制，大部分的交易均可以隨時隨地在個人電腦前，以數位方式經由網際網路的傳輸，與不特定的對象溝通，談生意並完成交易。但是由於網際網路是一個開放的網路，線上進行的交易資料有被冒名傳送及被竄改之風險，而且進行交易的雙方，有可能彼此並不認識對方，所以亦無法辨識對方身分的真實性。

本論文的研究動機，就是基於上述所提的問題，希望能夠開發一套數位簽章(Digit signature)系統，使它可以直接嵌入(Plug-in)現已被廣被使用的 Word 文字編輯器，如此一來，使用者不但可以直接使用他原已熟悉的文字編輯器，來填寫他的訂單，並可透過他現有的 e-mail 系統，將訂單傳送到客戶的電子信箱，而收信的對方，更可以根據他收到訂單，來進行身分及內容的查核，以確保該訂單沒有被冒名傳送及竄改，並且將來如果雙方對於該訂單的真實性發生爭議時，可以直接根據訂單上的數位簽章，來進行公正的裁奪。

Digital Signature Plug-in for Microsoft Word

student：Tsung-Lung Chao　　Advisors：Dr. Jenn-Hann Liou

Degree Program of Electrical Engineering Computer Science

College of Electrical Engineering and Computer Science

National Chiao Tung University

ABSTRACT

Along with the widespread use and application of personal computer and internet , e commerce is also booming in its application fields ,people nowadays are no longer be limited by time , space and target in the transaction method , most of the business nowadays can be completed any time and anywhere just in front of a personal computer and digitally through the transmission by internet , the transaction targets are no longer limited too . But since internet is an open network , online transaction data could possibly be transmitted by impostor or have the risk of being modified . Furthermore , both sides on transaction could possibly be not familiar with each other , it is also difficult to distinguish the reality of identity of the opposite side .

The motive of this study is try to solve the above-mentioned problem , we try to develop a Digital Signature system , the currently wide-spread used Microsoft Word editor can be embedded directly in the system, therefore , user can use directly the word editor that he/she is familiar with to fill the purchase order , it can also transmit the purchase order to customer’s email box through its existed e-mail system ,moreover , email receiver could perform ID check and content examination to make sure the purchase order has not been transmitted by impostor or modified , in the future , if there is any controversy of the reality of the purchase order between both sides , the digital signature on the purchase order can be used to judge which side is right.

誌　　謝

本論文承蒙恩師劉振漢教授的細心指導及教悔才得以完成，老師不僅在學術上給予我多方面的指導，甚至在為人處世及做研究的態度上，多讓我受益良多，僅此，獻上本人無限的感謝。

此外，家人的鼓勵是我得以無後顧之憂完成學業的最大支持，在此，表達我無限的感恩，謝謝。

目錄

中文摘要 III

英文摘要 IV

誌謝 V

目錄 VI

圖目錄 VIII

表目錄 IX

一、 緒論 1

1.1 研究動機 1

1.2 研究方法 1

二、 背景知識 4

2.1 數位簽章的架構 4

2.2 數位簽章的安全防護 6

2.3 數位簽章的資料驗證 8

2.4 數位簽章的密鑰保護 9

三、 相關研究 12

3.1 數位簽章系統的比較 12

3.1.1 漢龍資訊:機關表單簽核自動化系統 12

3.1.2 漢龍資訊:教務行政管理系統 13

3.2 數位簽章系統的相關需求 14

3.2.1 數位簽章的功能需求 14

3.2.2 數位簽章的安全需求 15

3.3 數位簽章系統的特性 15

四、 嵌入式 WORD 文件簽章系統 16

4.1 系統分析 16

4.2 操作介面 20

4.3 程式流程 24

4.4 數位簽章 PLUG-IN 到 WORD 編輯器 26

4.5 系統評估 27

五、 結論 29

5.1 總結 29

5.2 未來工作 29

參考文獻 30

圖目錄

圖 1. 原始的簽章圖檔的部分二進位資料 2
圖 2. 被 WORD 應用程式修改過的部分二進位資料 3
圖 3. 雜湊函數的功能示意圖 4
圖 4. 文件加密的功能示意圖 5
圖 5. 文件的訊息摘要比對的功能示意圖 6
圖 6. 漢龍資訊 : 表單簽核系統 12
圖 7. 漢龍資訊 : 教務行政管理系統方塊圖 13
圖 8. 金鑰對產生的功能示意圖 17
圖 9. 數位簽章的加簽功能示意圖 18
圖 10. 數位簽章的簽章查核功能示意圖 19
圖 11. 數位簽章系統的功能表 20
圖 12. 數位簽章簽核的對話盒 21
圖 13. 數位簽章查核的對話盒 21
圖 14. 金鑰對產生器 22
圖 15. 公鑰伺服器 23
圖 16. SIGN IN 程式流程圖 24
圖 17. SIGN OUT 程式流程圖 25
圖 18. WORD 巨集指令呼叫 26
圖 19. 系統修正前加解密耗費的時間 28
圖 20. 系統修正後加解密耗費的時間 28

表目錄

表 1. MD5 的非線性函數表.....9

表 2. RC5 的參數表 10

一、 緒論

1.1 研究動機

一般傳統的交易方式乃是買賣雙方，依據雙方談好的貨品規格、數量、價格及個別對應的權利義務記載於買賣契約上，並經簽名蓋章後，由雙方各持一份留存，以保障買賣雙方均按約定條件完成交。

但是隨著個人電腦及網際網路的普及應用，促使電子商務的應用範疇的突飛猛進,現在人們的買賣方式可以不再受到時間、空間及對象的限制，大部分的交易均可以隨時隨地在個人電腦前，以數位方式經由網際網路的傳輸，與不特定的對象溝通，談生意並完成交易。

但是由於網際網路是一個開放的網路，線上進行的交易資料有被冒名傳送及被竄改之風險，而且進行交易的雙方，有可能彼此並不認識對方，所以亦無法辨識對方身分的真實性。

本論文的研究動機，就是基於上述所提的問題，希望能夠開發一套數位簽章(Digit signature)系統，使它可以直接嵌入(Embed)現已被廣被使用的 Word 文字編輯器，如此一來，使用者不但可以直接使用他原已熟悉的文字編輯器，來填寫他的訂單，並可透過他現有的 e-mail 系統，將訂單傳送到客戶的電子信箱，而收信的對方，更可以根據他收到訂單，來進行身分及內容的查核，以確保該訂單沒有被冒名傳送及竄改，並且將來如果雙方對於該訂單的真實性發生爭議時，可以直接根據訂單上的數位簽章，來進行公正的裁奪。

1.2 研究方法

原始構想

本系統初期的想法，是在 Word 軟體裡，加入我們的 plug-in 程式。

- 使用者先準備好他的印章的 bmp 影像檔，並且準備好他的私密金鑰。
- 當他編輯好一個 word 文件，他可以用 plug-in 增加到 Word 的功能，在這文件上簽章。
- 所謂簽章，是
  - 將文件的文字產生文摘（MD5 的 digest），用私鑰簽章，
  - 將簽章（128 Bytes）藏入印章的 bmp 檔案裡，
  - 將藏有簽章的影像貼到 Word 文件的首頁後，存檔

這樣的好處是：

1. 瀏覽文件的人，一眼就可看到這文件有簽章。
2. 如果瀏覽文件的人的 Word 也安裝有我們的 plug-in，可以用簽章人的公鑰驗證一下簽章的真實性。如果不符，可能是文件內容被竄改了，也可能是別人僞簽。

實際遇到的問題

我們採用 24bits 的影像檔，每個 pixel 有 RGB（Red/Green/Blue）值各佔一個 byte(8bits)。將每個 byte 最不重要的低位的 2bits 用來存放 128 bytes 的 MD5 的簽章。這是浮水印常用的方法。結果，卻很意外的發現，當我們把影像貼到 word 文章裡，再取出來時，pixel 的值會被改變。這個改變雖然很輕微，以肉眼看不出影像有被改變。但對要求完全不可有任何 bits 改變的簽章來說，卻完全不可行。

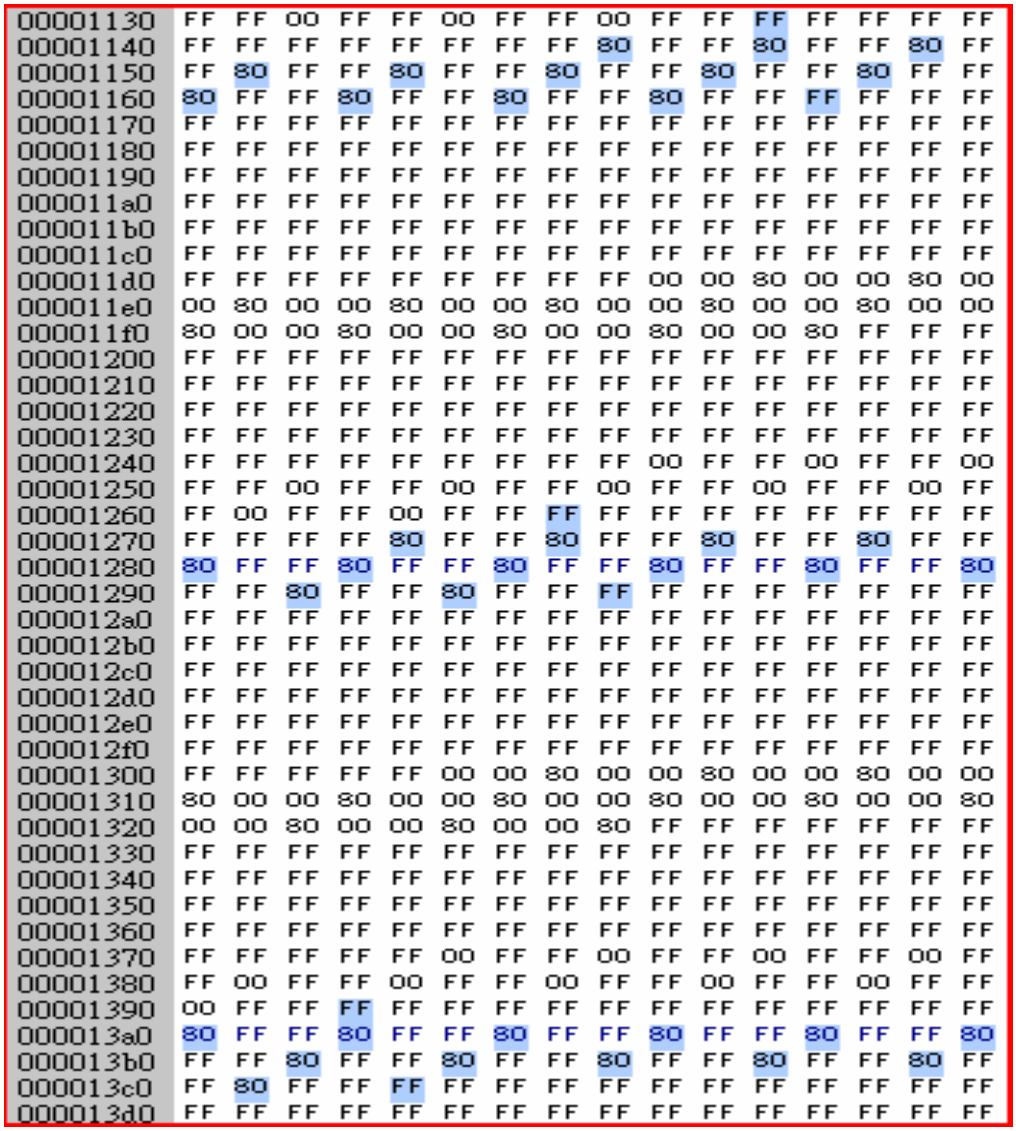
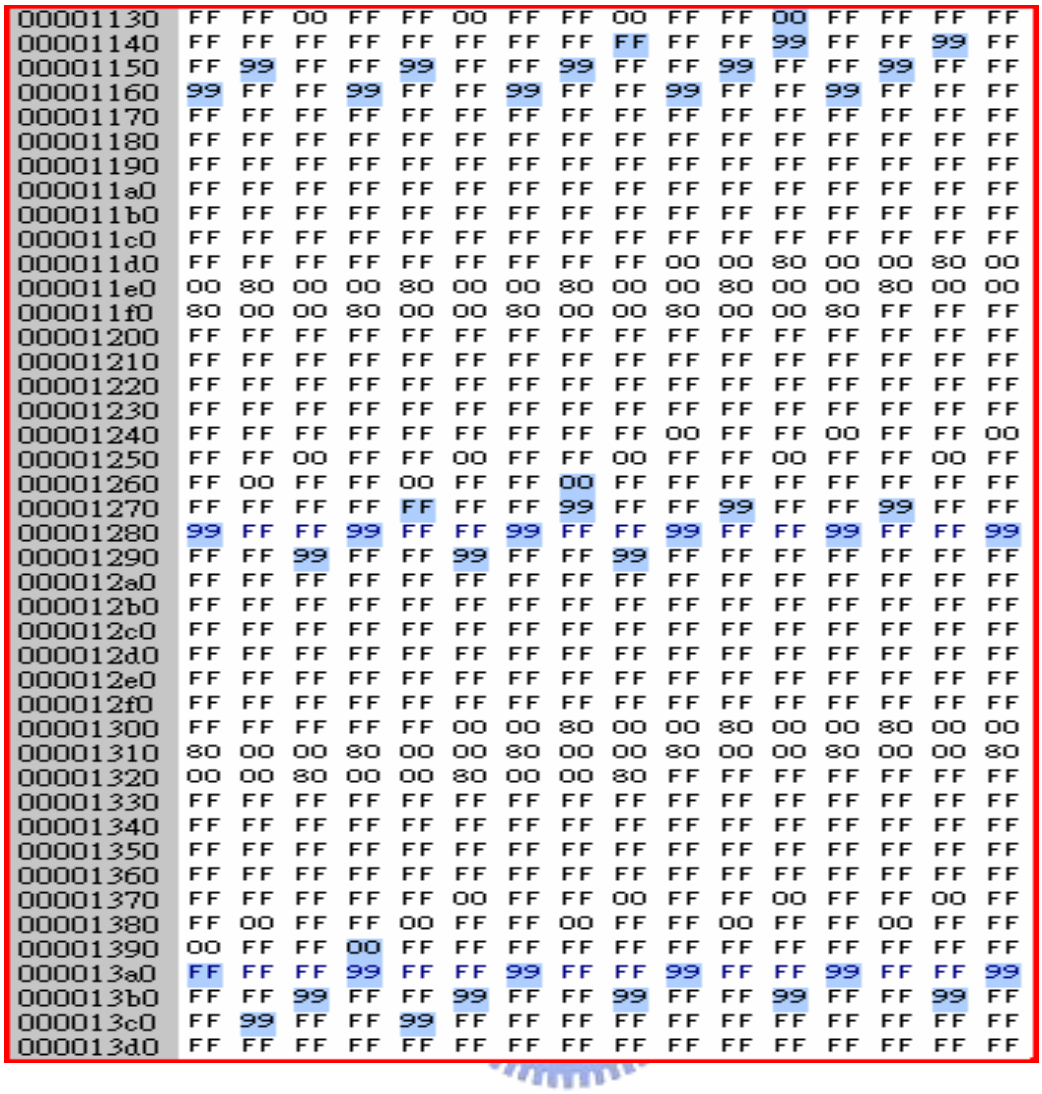


圖 1. 原始的簽章圖檔的部分二進位資料



```
00001130 FF FF 00 FF FF 00 FF FF 00 FF FF 00 FF FF FF FF
00001140 FF FF FF FF FF FF FF FF FF FF FF 99 FF FF 99 FF
00001150 FF 99 FF FF 99 FF FF 99 FF FF 99 FF FF 99 FF FF
00001160 99 FF FF 99 FF FF 99 FF FF 99 FF FF 99 FF FF FF
00001170 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00001180 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00001190 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000011a0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000011b0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000011c0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000011d0 FF FF FF FF FF FF FF FF FF 00 00 80 00 00 80 00
000011e0 00 80 00 00 80 00 00 80 00 00 80 00 00 80 00 00
000011f0 80 00 00 80 00 00 80 00 00 80 00 00 80 FF FF FF
00001200 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00001210 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00001220 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00001230 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00001240 FF FF FF FF FF FF FF FF FF 00 FF FF 00 FF FF 00
00001250 FF FF 00 FF FF 00 FF FF 00 FF FF 00 FF FF 00 FF
00001260 FF 00 FF FF 00 FF FF 00 FF FF FF FF FF FF FF FF
00001270 FF FF FF FF FF FF FF 99 FF FF 99 FF FF 99 FF FF
00001280 99 FF FF 99 FF FF 99 FF FF 99 FF FF 99 FF FF 99
00001290 FF FF 99 FF FF 99 FF FF 99 FF FF FF FF FF FF FF
000012a0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000012b0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000012c0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000012d0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000012e0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000012f0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00001300 FF FF FF FF FF 00 00 80 00 00 80 00 00 80 00 00
00001310 80 00 00 80 00 00 80 00 00 80 00 00 80 00 00 80
00001320 00 00 80 00 00 80 00 00 80 FF FF FF FF FF FF FF
00001330 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00001340 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00001350 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00001360 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00001370 FF FF FF FF FF 00 FF FF 00 FF FF 00 FF FF 00 FF
00001380 FF 00 FF FF 00 FF FF 00 FF FF 00 FF FF 00 FF FF
00001390 00 FF FF 00 FF FF FF FF FF FF FF FF FF FF FF FF
000013a0 FF FF FF 99 FF FF 99 FF FF 99 FF FF 99 FF FF 99
000013b0 FF FF 99 FF FF 99 FF FF 99 FF FF 99 FF FF 99 FF
000013c0 FF 99 FF FF 99 FF FF FF FF FF FF FF FF FF FF FF
000013d0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

圖 2. 被 Word 應用程式修改過的部分二進位資料

圖 1. 是原始的簽章圖檔的二進位資料，圖 2. 被 Word 應用程式修改過的二進位資料，有陰影標記的地方，表示被 Word 應用程式修改過的地方。

至於要將外部的系統，直接嵌入到的 Word 文字編輯器，就只能利用 Word 所提供的後門，也就是 Macro 巨集指令，並且以 Macro 指令來跟外部的系統溝通，其主要的原因，分述如下:

1. Macro 指令執行效率較差，如果全部以巨集指令來編寫程式，程式的執行速度恐怕會太慢。
2. Macro 指令的功能有限，不能完全達到本系統所要達成的目標。
3. 無法直接利用現有的軟體模組，以軟體重複使用的方式，來縮短系統開發的時間。
4. Macro 指令無法達到保密性及安全性的要求，由於指令是在執行的階段，才一行一行經過解譯器轉成機器碼之後才能執行，所以程式的內容無法保密，容易遭到他人的拷貝及竄改。

二、 背景知識

本論文是探討如何將數位簽章的理論，運用到 Word 編輯器的實作系統，所以針對數位簽章之原理及相關理論，作一概要的說明。

2.1 數位簽章的架構

假設簽章者爲 A，對一明文 m 做數位簽章之後，傳送給接收方 B，則此數位簽章必須滿足下列條件：

1. B 必須能驗證 A 對 m 簽章之合法性。
2. 任何人抱括 B，均無法僞造 A 的簽章。
3. 當 A 否認其對 m 的簽章時，可由公正的第三者解決 A 與 B 之間的爭執。

數位簽章是以雜湊函數(hash function)的數學函數來處理欲簽名的文件，雜湊函數會把文件內的所有位元組經過運算處理之後，產生一段稱之爲訊息摘要的數值(MD5 128 bits, SHA-1 160 bits)，而此一數值無法再透過此雜湊函數反向推導出文件的原始內容，圖 3 爲雜湊函數的功能示意圖。(Reference:11)

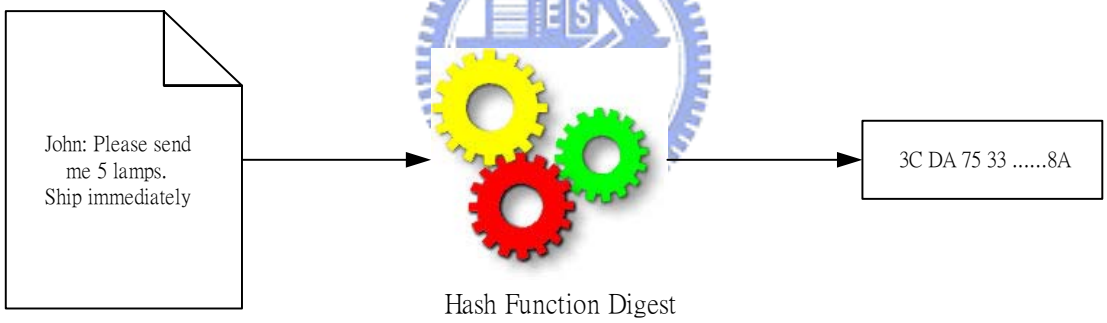


圖 3. 雜湊函數的功能示意圖

一般而言，使用於數位簽章的單向雜湊函數必需滿足下列條件：

1. 對於任意長度的輸入明文，產生固定長度的輸出雜湊值。
2. 對任意的輸入明文 m，雜湊值 h(m)可以由軟體或硬體很輕易且快速的產生。
3. 對任意的雜湊值 X，利用 X=h(m) 欲求得一明文 m，在計算上是不可行的。
4. 對一個固定的明文 m1，要找到另一個不同的明文 m2，使 h(m1) = h(m2)，在計算上是不行的。

當文件透過 Hash 函數，將文件的內容轉換成一段訊息摘要，再將此訊息摘要和文件一齊送出。如果中途有人修改過此文件，那麼接收者將被修改過的文件，經過 Hash 函數重新計算過後，將會發現算出來的摘要和傳送者送來的摘要不相吻

合，這就表示文件在傳送的過程中被人動過手腳，如此我們就能順利偵測出文件的真僞。

不過如果僞造者知道文件使用的 Hash 函數，他同時修改文件的內容及文件的訊息摘要，我們將無法透過上述的方法證明文件的真僞，也就是說我們必須先驗證傳送者的身份，才能驗證文件的正確性。

至於如何驗證傳送者的身份呢？傳送者只要在訊息摘要送出前用自己的密鑰加密，也就是說簽章，接收方在得到加密過的簽章後如果能用傳送者的公鑰正確解密，也就是說簽章查核，那麼接收方就可以確信這個文件的確來自傳送方，而且文件完整無誤沒有被修改過。圖 4 爲文件加密的功能示意圖。

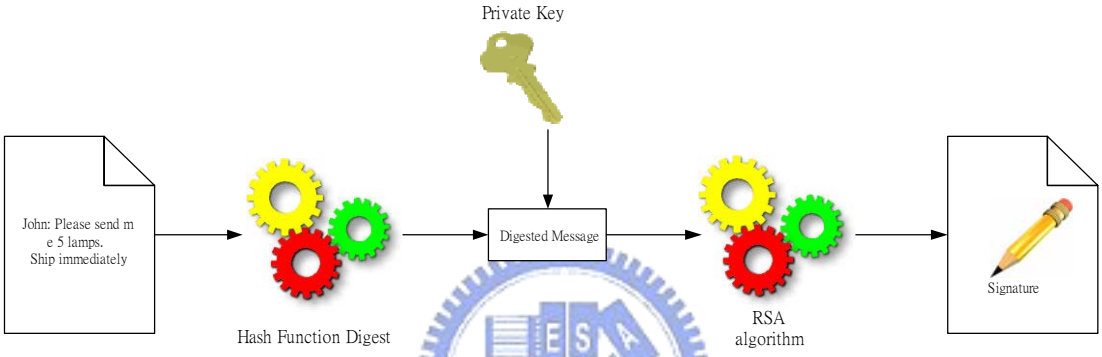


圖 4. 文件加密的功能示意圖

爲了確認簽章的正確性，接收端首先需透過網路將傳送端的帳號及公司名稱送給 DocClient.exe，並且透過 DocClient.exe 向公鑰伺服器取得傳送端的公鑰來解密。然後把訊息摘要的雜湊值存放在暫存區。接著接收端的巨集指令把文件裏的文字資料以同樣的雜湊函數來產生雜湊值。然後，再以這個新的雜湊值與解密過的雜湊值相比對。如果相同，則此簽名是正確的。圖 5 爲文件的訊息摘要比對的功能示意圖。

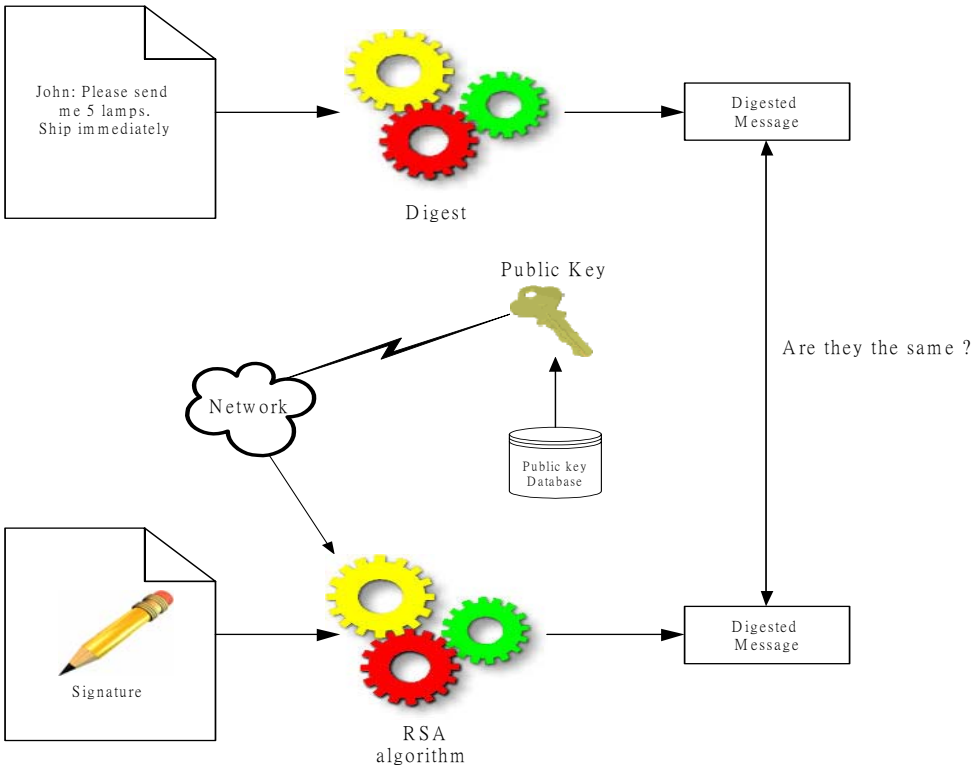


圖 5. 文件的訊息摘要比對的功能示意圖

2.2 數位簽章的安全防護

加密系統是數位簽章的安全防護系統，所以必須要有足夠的強度，才能避免資料被破解。從密碼學的演進，我們發現傳統的密碼系統往往只注重資訊的安全性及隱密性，然而近代密碼學廣泛的應用在商業上，使得資訊的鑑定性、完整性及不可否認性，益加顯得重要。(Reference:10)

RSA 是目前使用最爲廣泛的公開金鑰密碼系統，它是基於因數分解的指數函數爲基礎的一套密碼系統，其功能說明如下：

1. 金匙的產生：在 RSA 密碼系統中，每位使用者的金匙，可用下列方式取得：

   1.1 隨機找出兩個夠大的質數 p、q。
   1.2 計算 n，n 爲 p、q 之乘積。
   1.3 隨機找出一個滿足 gcd (e , Φ(n)) = 1 之整數 e，在此 Φ(n)爲尤拉商數，表示比 n 小且和 n 互質之整數個數，當 n = pq 時，Φ(n)之值爲(p - 1)(q - 1)。
   1.4 計算 d，使得 d 滿足 ed ≡ 1 (mod Φ(n))。
   1.5 以 e、n 爲公開金匙，d、n 爲祕密金匙。

2. 加密與解密：

2.1 欲將一明文 M 加密成爲密文 C 須取得其公開金匙 e、n，其步驟下：
C = E (M) ≡ M^e (mod n)，0 ≦ M < n。
2.2 欲將一密文 C 解密成爲明文 M 須取得其祕密金匙 d、n，其步驟如下：
M = D (C) ≡ C^d (mod n)，0 ≦ C < n。

由於 RSA 演算法需要質數 p 和 q 相乘得到的 N，作爲系統運作時的參數，假使(p-1) 的質因數太小，就很容易遭受攻擊，所以對 RSA 演算法來說，其強度決定於大數分解的困難度。至於如何尋找大質數，在實際應用上我們以 Miller-Rabin 的檢驗步驟，來找尋我們所需要的大質數，其方法如下 :

1. 根據所需要的位數，隨機挑選一個整數 n，當作候選的質數。
2. 對質數 n 作驗證。
3. 如果 n 被驗證不是質數，則回到步驟 1，否則輸出質數 n。

我們將整個 Miller-Rabin 的演算法，以程式語法表示如下:

WITNESS(a, n)

1. Let $b_k b_{k-1}.....b_0$ be the binary representation of (n - 1)
2. d ← 1
3. for i ← k downto 0
4. do x ← d
5. d ← (d * d) mod n
6. if d = 1 and x ≠ 1 and x ≠ n - 1
7. then return TRUE
8. if b_i = 1
9. then d ← (d * a) mod n
10. if d ≠ 1
11. then return TRUE
12. return FALSE

根據費馬公式(Fermat’ s theorem) $a^{n-1} \equiv 1 \pmod n$, n 乃是基於 a 的僞質數(base-a pseudoprime), 所以最後結果要是 d 不爲 1,則表示 n 不是質數。至於第 6 行主要是檢查是否有 $x^2 \equiv 1 \pmod n$，如果有 x 不等於 ± 1 (mod n) 的情形發生，則表示 n 不是質數。

2.3 數位簽章的資料驗證

訊息摘要是由一組輸入的資料，透過函數計算所得到的一組特別數字，其函數由於具有單向的特性，一般稱爲雜湊函數。在密碼學的運用裡，一般是用來驗證資料是否被竄改。(Reference:10)

MD5 訊息摘要演算法可以將任意長度的訊息內容，轉換成 128 bits 的訊息摘要，並且由於它具有單向的特性，因此無法將計算得出的訊息摘要，再透過此雜湊函數反向推導出文件的原始內容，因此廣泛的被運用在數位簽章系統，其執行的步驟如下：

1. 在 MD5 演算法中，首先需要對訊息進行填充，使其位元長度除於 512 的餘數等於 448 (length ≡ 448 mode 512) ，即訊息的位元長度等於 N*512+448 (bits) 或者是 N*64+56 (bytes) ，N 爲一正整數。
2. 填充的方法爲，在信息的後面填充一個 1 和無數個 0，直到滿足上述的條件時才停止用 0 對信息的填充。然後，在這個結果後面附加一個以 64 位元二進位表示的填充前信息長度。經過上述兩個步驟的處理，訊息內容的位元長度等於 N*512+448+64=(N+1)*512，即長度剛好是 512 的倍數。
3. 設定 MD5 的四個 32 位元的整數參數初始值，分別爲：A=0x01234567，B=0x89abcdef，C=0xfedcba98，D=0x76543210。

4. 將上面四個參數初始值，複製到另外四個變數 a、b、c、d。接著，進行四輪的循環。每一輪進行 16 次的操作。每次操作對 b、c 和 d 作一次非線性函數運算，然後將結果加上變數 a、原文的某一 32 位元區塊 X 和陣列 T 的某一 32 位元整數值。最後再將所得結果向左環移(Rotate)一個 s 的數值，並加上 b。以該結果取代 b。以下是 MD5 的壓縮方程式：

   $$a \leftarrow b + ((a + g(b,c,d) + X[k] + T[i]) <<< s)$$

   其中

   a, b, c, d : 32 位元的四個變數。

   g : 表 1. MD5 的四個非線性函數 F、G、H、I。

   <<< s : 向左環移(Rotate) s 個位元數。

   $X[k]$: 原文的某一 512 位元區塊的 32 位元子區塊。

   $T[i]$: 陣列 T 的第 i 個整數值。

   + : 加法模數 2^{32} 。

以下是每次操作中用到的四個非線性函數：

Round	Primitive function	g(b, c, d)	$X[k]$
1	F(b, c, d)	(b ^ c) v (~b ^ d)	$i \mod 16,\ 0 \le i \le 15$
2	G(b, c, d)	(b ^ d) v (c ^ ~d)	$1+5i \mod 16,\ 0 \le i \le 15$
3	H(b, c, d)	b ⊕ c ⊕ d	$5+3i \mod 16,\ 0 \le i \le 15$
4	I(b, c, d)	c ⊕ (b v ~d)	$7i \mod 16,\ 0 \le i \le 15$

表 1. MD5 的非線性函數表 (^ : AND, v : OR, ~ : NOT, ⊕ : XOR)

MD5 是以 512 位元爲單位，進行四輪的操作，第一輪對應的的非線性函數爲 F，第二輪對應的的非線性函數爲 G，第三輪對應的的非線性函數爲 H，第四輪對應的的非線性函數爲 I。而每一輪進行 16 次的操作。每次操作對 b、c 和 d 作一次非線性函數運算，然後將結果加上變數 a、原文的某一 32 位元區塊 X 和陣列 T 的某一 32 位元整數值。最後再將所得結果向左環移(Rotate)一個 s 的數值，並加上 b。以該結果取代 b、d 取代原來的變數 a、c 取代原來的變數 d，而 b 取代原來的變數 c。

2.4 數位簽章的密鑰保護

一般而言，數位簽章的密鑰是擺在個人的目錄或者是可攜式的儲存裝置，所以必須考慮到密鑰有可能被他人盜用的情形發生，而最常用的方法就是用密碼加以保護。(Reference:10)

RC5 加密演算法是一種對稱式的加解密方法，也就是加密跟解密都使用同一把密鑰，其特點如下:

1. 快速,簡單,適合以硬體或軟體的方式開發。
2. 運算的循環數(Round)可以變換。
3. 密碼的長度可以變換。
4. 不需要很大的記憶體運算空間。
5. 輪迴數(Rotation)隨資料(Plain Text)長度不同而變換。

RC5 加解密演算法事實上是由三個參數所決定的，這三個參數會影響整個演算法運算所需要的空間及時間，這三個參數列表如下:

參數	定義	允許值
w	Word 的長度，RC5 加解密一次使用兩個 Word 的空間	16，32，64
r	運算的循環數(Round)	0，1……255
b	密鑰 K 的長度(單位:Byte)	0，1……255

表 2. RC5 的參數表

RC5 加解密的運算需將密鑰展開，子密鑰的單位爲 Word 的長度，子密鑰陣列的大小 $t = 2r + 2$ 。

RC5 子密鑰展開的演算法，以程式語法表示如下:

1. $i = j = X = Y = 0$
2. $do 3 \times \max(t, c) times$
3. $S[i] = (S[i] + X + Y) <<< 3; X = S[i]; i = (i+1) \bmod (t);$
4. $L[j] = (L[j] + X + Y) <<< (X + Y); Y = L[j]; j = (j+1) \bmod (c);$

RC5 加密的演算法，以程式語法表示如下:

1. $LE_0 = A + S[0];$
2. $RE_0 = B + S[1];$
3. $for \;\; i = 1 \;\; to \;\; r \;\; do$
4. $LE_i = ((LE_{i-1} \oplus RE_{i-1}) <<< RE_{i-1}) + S[2 \times i];$
5. $RE_i = ((RE_{i-1} \oplus LE_{i-1}) <<< LE_i) + S[2 \times i + 1];$

RC5 解密的演算法，以程式語法表示如下:

1. $for \;\; i = r \;\; down \;\; to \; 1 \;\; do$
2. $RD_{i-1} = ((RD_i - S[2 \times i + 1] >>> LD_i) \oplus LD_i);$
3. $LD_{i-1} = ((LD_i - S[2 \times i] >>> RD_{i-1}) \oplus RD_{i-1});$
4. $A = LD_0 - S[0];$
5. $B = RD_0 - S[1];$

+ : 加法模數 2^w 。

⊕ : XOR 。

x<<< y : x 向左環移(Rotate) y 個位元數。

x>>> y : x 向右環移(Rotate) y 個位元數。

RC5 執行加解密時，原文會先擺放在變數 A 及 B，接著利用展開的子密鑰，進行加解密的運算，由於 RC5 演算法的運算大多是位元的操作，所以擁有很高的運算速度，並且由於本身具有可變的 Word 長度，可變的運算循環數，及可變的密鑰長度，使得它非常適合以硬體或軟體的方式來實現。

這一章節我們討論到數位簽章的架構，一個好的數位簽章系統必須滿足三個條件一、接收方 B 必須能驗證簽章者 A 對 m 簽章之合法性。二、任何人抱括 B，均無法僞造 A 的簽章。三、當 A 否認其對 m 的簽章時，可由公正的第三者解決 A 與 B 之間的爭執。

數位簽章必須要有足夠的的安全防護，才能避免資料被破解。而提供安全防護的基礎是建立在加密系統的強度上，所以必須要有安全可靠的密碼系統，才有可信賴的數位簽章系統。一個具有商業用途的密碼系統，不僅要提供資訊的安全性及隱密性，並且還要使得資訊具有鑑定性、完整性及不可否認性，而 RSA 公開金匙密碼系統的特性，正是符合當今商業應用的要求而被廣泛的使用。

數位簽章的資料驗證是透過訊息摘要的比對驗證，防止文件被他人竄改，而訊息摘要是經有單向雜湊函數的計算得出，因此無法利用已知的訊息摘要，反向推導出文件的原始內容。MD5 訊息摘要演算法可以將任意長度的訊息內容，轉換成 128 bits 的訊息摘要，並且由於它具有單向的特性，因此被運用在數位簽章系統的訊息摘要處理。

數位簽章的密鑰保護主要是防止 RSA 密鑰被他人盜用，所以透過對稱式加密系統的加密，提供數位簽章系統更進一步的安全保護。由於 RC5 對稱式加密系統具有簡單快速的特性, 而且又具有相當程度的安全強度，因此常被運用在數位簽章系統的密鑰保護。

三、相關研究

本章節希望能夠從現有的已經商業化的簽章系統，去研究個別系統的優缺點，並且經由深入的研究，去了解一個有價值的數位簽章系統，該具有的相關功能和基本需求，並且經由這樣的學習，開發一套符合商業需求的數位簽章系統。

3.1 數位簽章系統的比較

爲了更深入的了解，我們從不同的案例，挑出具代表性的兩個實例加以分析和探討。

3.1.1 漢龍資訊 : 機關表單簽核自動化系統 (Reference:7)

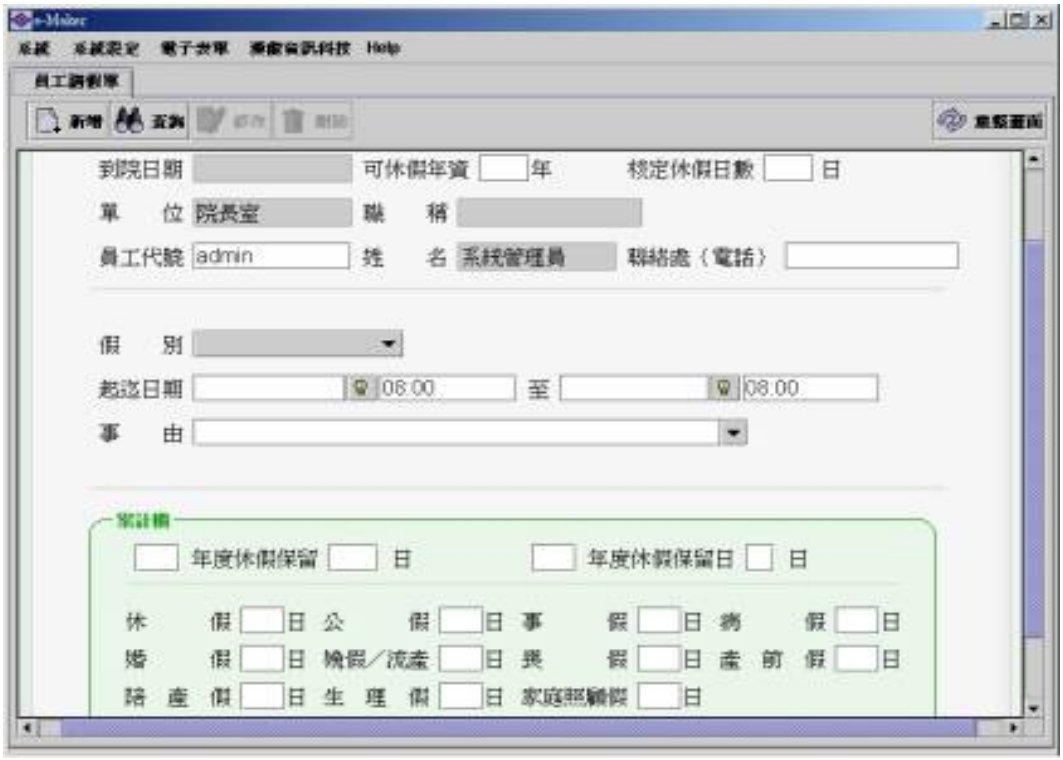


圖 6. 漢龍資訊 : 表單簽核系統

其系統的優點描述如下:

- 全方位的擴充與整合：完全符合 PEMIS 2000。
- 建置彈性且客製化：具有完善的延展性與擴充性，可依使用者的需求，建置所需之功能，而減少購置不必要系統的經費。

- 跨平台的優勢：全 web 化環境，採用 JAVA 開發，沒有平台相容性的問題，可在 Windows、Linux 平台運作。
- 系統安全控制：權限管理容易，可整合 CA 與 PKI 技術。

其系統的缺點描述如下:

- 所有的流程都必須經過簽章的驗證，如果簽章系統出現問題時，會影響整體工作的進行。
- 當資料修改或異動時，所有的會簽程序都必須重新執行，相對地會提高相關人員的工作量。
- 由於系統複雜度的提高，使得整體的系統維護費用增加。
- 由於作業跟操作習慣的改變，必須要有配套的人員訓練。

3.1.2 漢龍資訊 : 教務行政管理系統 (Reference:7)

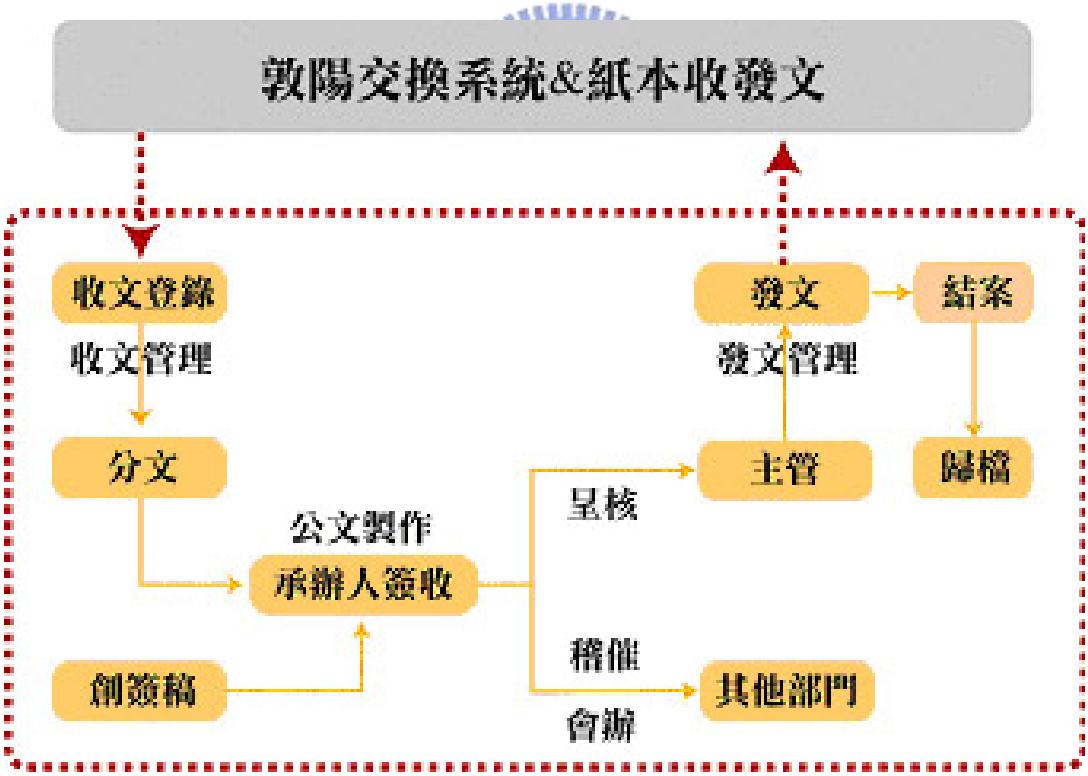


圖 7. 漢龍資訊 : 教務行政管理系統方塊圖

其系統的優點描述如下:

- 結合公文交換機制、公文製作、公文管理、流程簽核、電子簽章、檔案管理，具有相當完整的 e 化解決方案。

- 提供電子附件功能，將電子文件當成附件傳送給使用者。
- 流程具有相當的彈性，各關卡可自行加簽、退簽等與自訂個人流程。
- 使用 Word 作爲列印介面，或 web Base 公文製作。

其系統的缺點描述如下:

- 由於各個子系統都有對應的檢查點及簽章驗證，使得整體的複雜度提高，當系統出錯時，相對地要找出真正的問題點，困難度也跟著提高。
- 由於系統必須要有完整的追蹤紀錄，才可以保障整體系統的安全度，所以相對地必須提高系統的備分容量，方可以達到系統的安全需求。
- 整個工作流程的複雜度相對提高，必須要有完整的教育訓練，才可以讓系統運作順利。
- 當有新的需求功能加到系統時，必須要經過長時間的驗證及測試，方可以整合到系統裡面，相對地會拖延系統的開發時間。

3.2 數位簽章系統的相關需求

從上面的案例，我們可以歸納出數位簽章有兩個基本需求，即功能需求及安全需求，如下所述。

3.2.1 數位簽章的功能需求

一個電子簽章系統應提供下列功能：

- 簽章流程、各簽章人及其權限設定 : 對於簽章的詳細流程，個別文件需要哪些人的簽章核可，及其相關的權限，都需要有明確的規範。
- 密碼修改 : 簽章必須提供密碼的修改，並且必須明確的規定密碼的有效日期及其有效的格式，強迫使用者定期的更改密碼，並且必須合乎安全的規範。
- 資料查詢 : 簽章必須提供相關的資料查詢，以便提供必要的管理及考核。
- 資料修改 : 簽章必須提供相關的資料修改，以便提供必要的管理及異動的修改。
- 簽章活動歷史、紀錄及追蹤 : 簽章必須提供相關的歷史資料，以便將來的追蹤及必要證明資料。

3.2.2 數位簽章的安全需求

- 身份確認 : 確認使用者簽章的身份正確無誤，才允許進行簽章認可。
- 存取控制 : 使用者之權限規定，以免資料遭到非授權者的任意竄改或自行盜用。
- 資料傳送之完整性 : 爲了防止資料在傳輸過程中產生錯誤或被有意竄改，當錯誤發生或不正常竄改時，要能及時偵測出來，通知對方更正或通知具有權限者立刻出面處理。
- 建立防火牆 : 以保護電子文件、電子簽章作業系統。
- 強制性的使用者認證機制及警示機制 : 採取更強制性的使用者認證機制及警示機制，並且收集記錄所有的網路活動歷史。

3.3 數位簽章系統的特性

電子簽章除了基本的功能需求和安全需求外，一個成功的數位簽章系統需具有下列之特性：

- 電子文件要具有良好的操作介面，使所有的資料能夠迅速輸入、檢查，並且主動告知錯誤。
- 以簽章者的角度來規劃電子簽章作業系統，滿足簽章之真正功能需求。
- 數位簽章作業之規則設計應讓使用者、簽章者共同參與問題，否則問題百出的數位簽章系統恐怕難以被使用者、簽章者所樂意接受。

經由上述兩個案例的探討，我們可以歸納出一套符合商業需求的數位簽章系統，必須提供基本的功能需求，如簽章流程的權限設定、密碼修改、資料查詢、資料修改及簽章活動的歷史紀錄。以及必要的安全需求，如身份確認、存取控制之權限設定、資料傳送完整性之偵測、電子簽章系統及文件的保護以及強制性的使用者認證機制。同時必須滿足使用者的特性要求，如具有良好的操作介面、完善的規劃以及讓使用者能夠充足的參與，如此方能讓使用者及簽章者所樂意使用。

四、嵌入式 **Word** 文件簽章系統

經由上述章節的研究探討，我們對於數位簽章的理論原理，已經有深刻的了解，並且藉由商業化簽章系統的實例探討，我們對於要設計的數位簽章系統有更明確的目標和方向。一、系統要能夠提供完整的驗證，也就是必須具備完整的公開金鑰密碼系統，以及提供必要的公鑰管理。二、系統的操作要盡量的簡化，也就是要降低系統操作的困難度，使系統能夠自動連結到使用者的公鑰及私鑰，以便執行加密及解密的功能。三、爲了方便系統的維護管理，系統的設計必須盡量朝向模組化。四、使用者的操作介面要盡量符合使用者的操作習慣，以降低使用者的學習時間。

4.1 系統分析

本系統可分爲四大部分，分述如下:

1. Macro 巨集指令，Word 本身有提供一套類似 VB 語法的巨集指令，使得程式設計師可以藉由客製化應用系統的開發，來增強 Word 編輯器的功能。Word 巨集指令主要的任務，大都在處理跟 Word 本身有關的工作，例如從現在已開啓的文件中，取出所有的字元存放到暫存檔，將數位簽章擺放到第一頁的頁首，或者將數位簽章二進位轉成文字及文字轉成二進位數位簽章等等工作。所有子系統的呼叫也都是有巨集指令控制，負責所有的協調工作。
2. 加解密引擎，這部分爲 dll 動態聯結程式庫，所有與加解密相關的程式，以及訊息摘要的處理，都被擺在此一 dll 檔，並且經由動態聯結的方式，呼叫所需要的功能程式。加解密引擎負責處理所有的計算工作，例如加密、解密的工作，訊息摘要的處理，金鑰對的產生，幾乎所有子系統都會呼叫到它，這也是考慮將它作成動態聯結程式庫的目的，如此可以節省相當多的記憶體空間，由於加解密的運算工作，需要耗費相當多 CPU 資源，所以程式有相當大的比例都是用組合語言完成，以縮短運算的時間，在下面的章節我們會做系統評估，從中可以發現整個系統的處理時間，事實上是非常快速的。
3. 金鑰對產生器，這部分爲執行應用程式，使用者利用它來產生公私鑰。產生的公鑰，會被送到公鑰伺服器儲存，產生的私鑰，會以使用者的密碼做爲 KEY，用 RC5 加密保護。當使用者要執行數位簽章查核時，也是透過本應用程式，將使用者的公鑰，從公鑰伺服器取回。金鑰對產生器由於需透過網路跟公鑰伺服器存取公鑰，所以必須包含有網路的處理程序，而它跟公鑰伺服器之間也必須有互相溝通的 Protocol，因爲必須考慮非同步處理的情形。
4. 公鑰伺服器，這部分爲執行應用程式，負責使用者公鑰的存取，它跟金鑰對產生器之間有一套溝通的 Protocol，並且透過網路來傳遞公鑰。公鑰伺服器資料庫是採用 Borland 資料庫引擎，而公鑰是放在專門儲存二進位大資料的 BLOB 欄位。圖 8、圖 9 及圖 10 爲本系統的數位簽章功能示意圖。

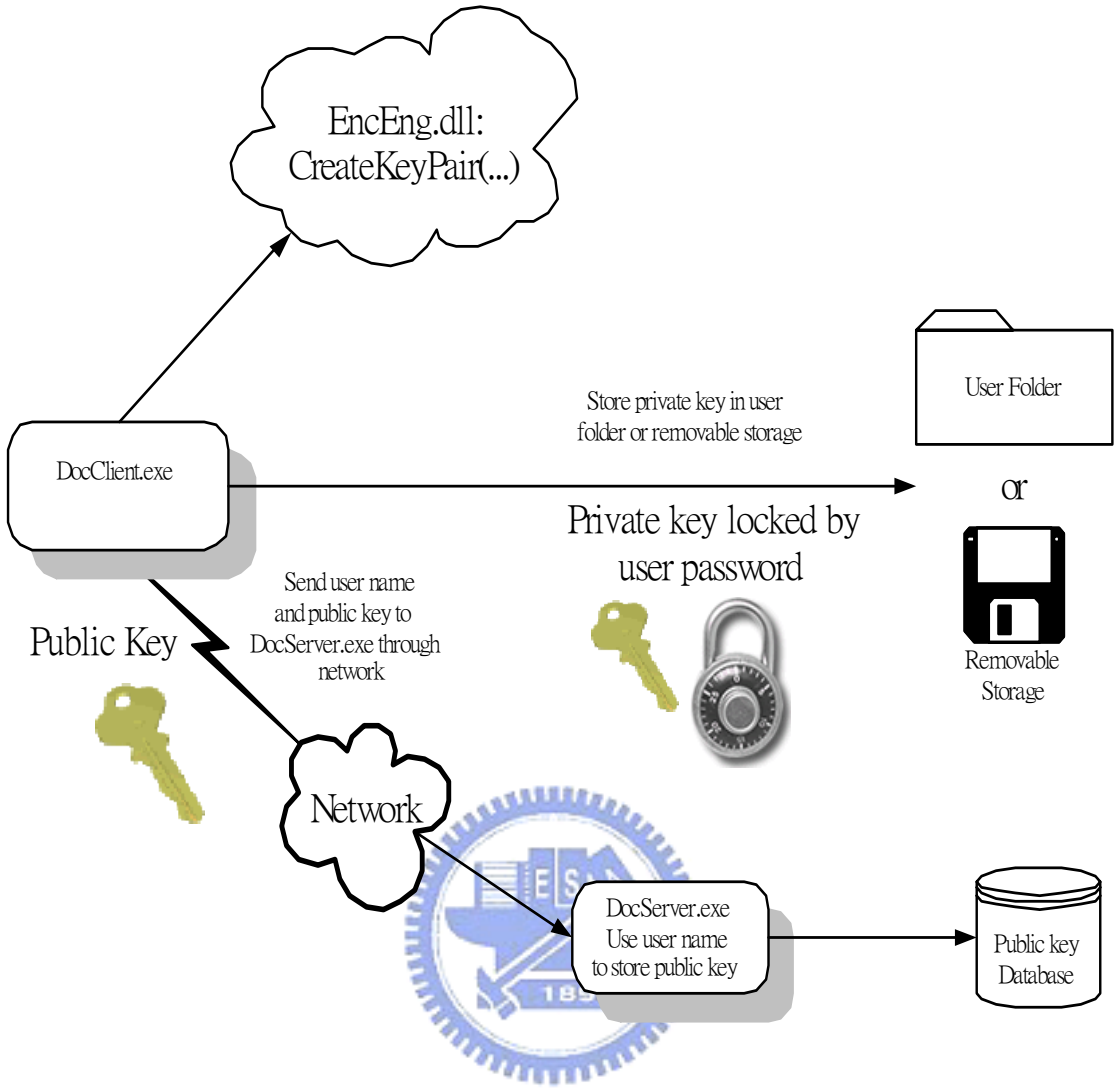


圖 8. 金鑰對產生的功能示意圖

金鑰對產生器執行應用程式，主要功能是產生公鑰和私鑰。產生的公鑰會被送到公鑰伺服器儲存，產生的私鑰會以使用者的密碼做爲 KEY，用 RC5 加密保護。當使用者要執行數位簽章查核時，也是透過本應用程式，將使用者的公鑰，從公鑰伺服器取回。

金鑰對的產生分爲下列三個步驟 :

1. 呼叫加解密引擎的金鑰對產生方程式，由它負責產生一對 RSA 的公私鑰。

2. 私鑰先以 RC5 加密後，送到使用者的個人目錄。

3. 公鑰會透過網路傳送到公鑰伺服器，由公鑰伺服器保存到資料庫裏面。

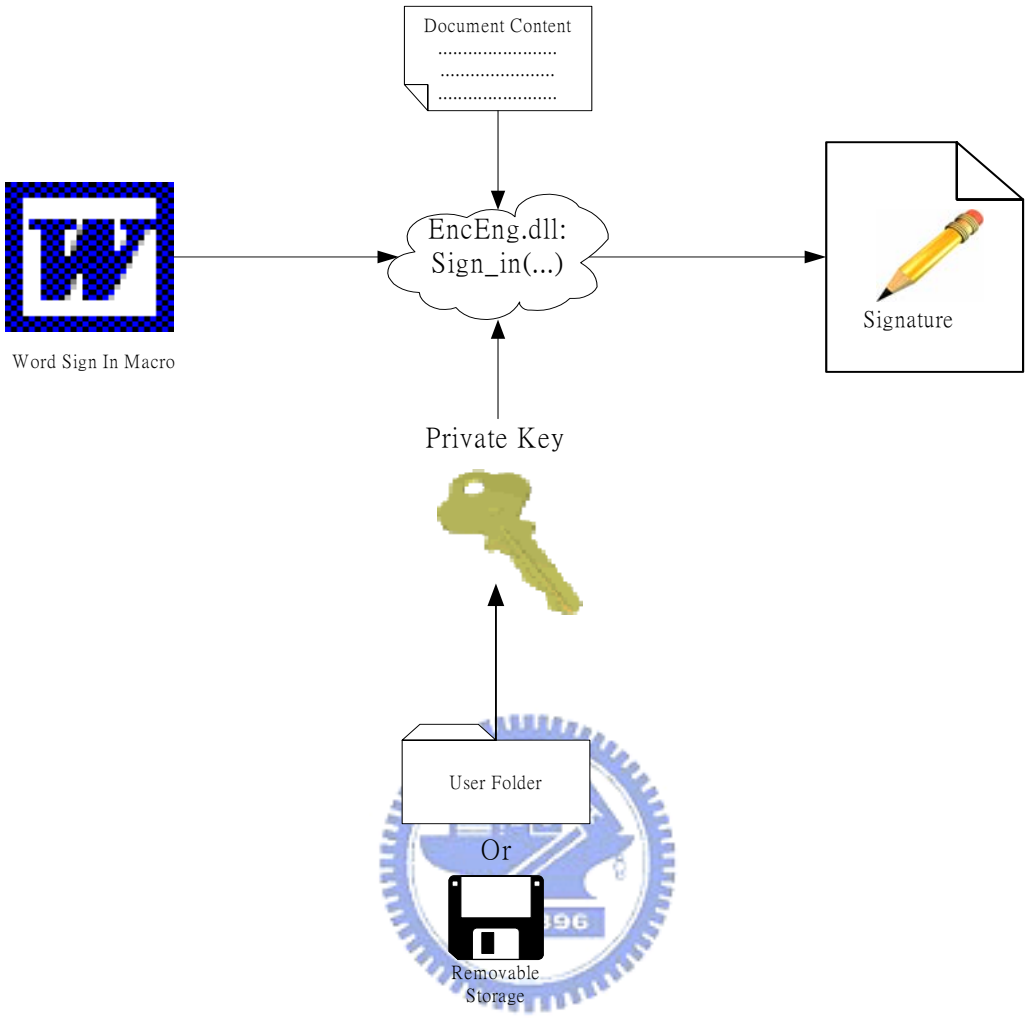


圖 9. 數位簽章的加簽功能示意圖

數位簽章的加簽功能，會呼叫加解密引擎的 Sign_In 程式，並且將使用者的存放私鑰的路徑，存放字元暫存檔的路徑及 Password 傳遞給 Sign_In 程式。數位簽章的加簽功能的執行步驟如下：

1. 取出現在被開啓的 Word 檔的所有字元，存到字元暫存檔。
2. 存放字元的暫存檔，執行一遍 MD5 的雜湊函數，以得到訊息摘要。
3. 以 Password 作爲 RC5 的 Key，將私鑰解密得到 RSA 的私鑰。
4. 用 RSA 的私鑰對訊息摘要加密，以獲得該文件的數位簽章。

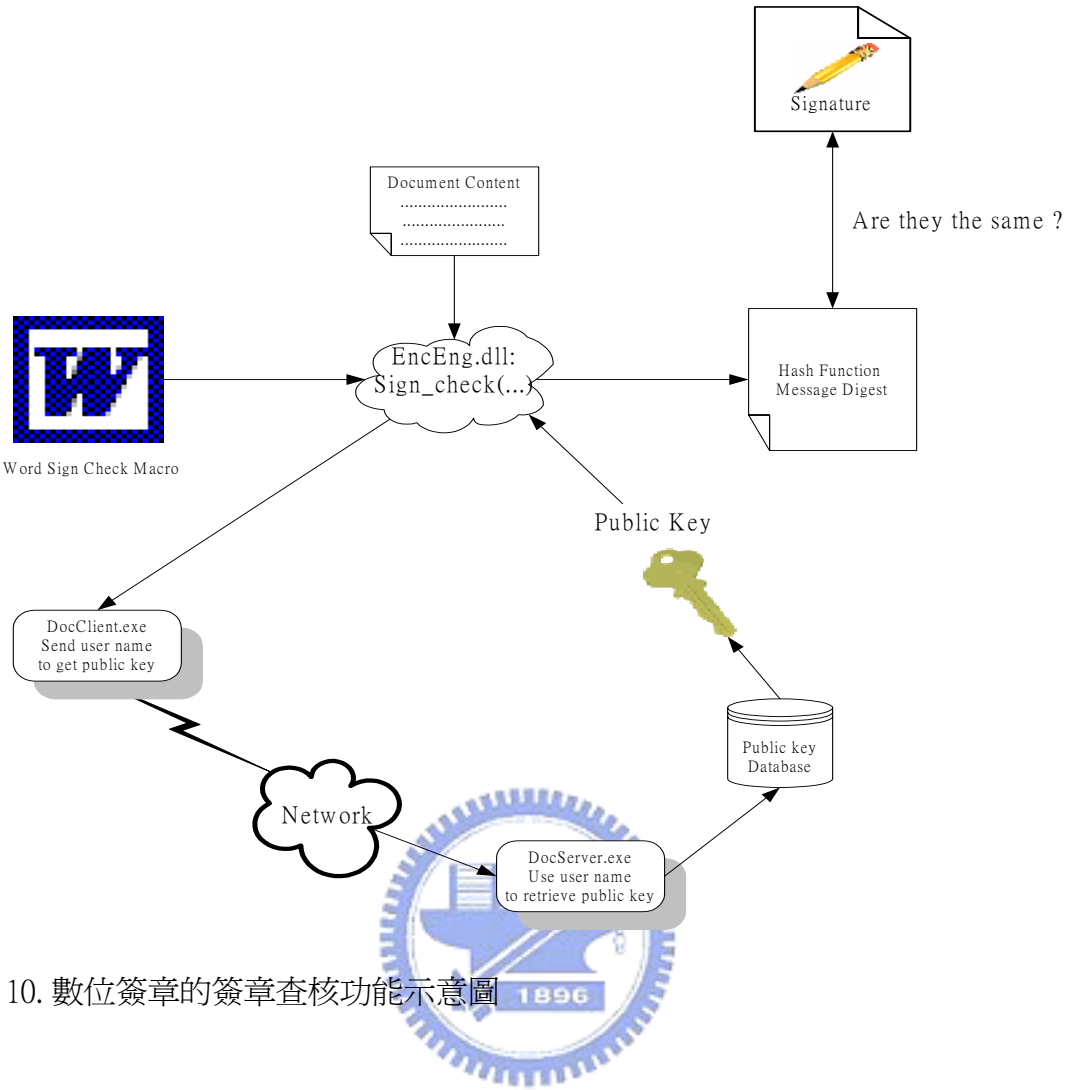


圖 10. 數位簽章的簽章查核功能示意圖

數位簽章的簽章查核功能，會呼叫加解密引擎的 Sign_Check 程式，並且將 DocClient.exe 傳回的使用者的公鑰路徑，存放字元暫存檔的路徑及存放簽章暫存檔的路徑傳遞給 Sign_Check 程式。數位簽章的簽章查核功能的執行步驟如下：

1. 取出現在被開啓的 Word 檔的所有字元，存到字元暫存檔。
2. 執行 DocClient，由它透過網路，取出文件擁有者的公鑰，傳回到使用者的目錄。
3. 呼叫加解密引擎的 Sign_Check 方程式，它會執行下列三個動作。

    - 針對文字暫存檔的資料，同樣執行一遍 MD5 的雜湊函數。
    - 利用傳回的 RSA 公鑰，對文件裏的數位簽章解密，以取出舊有的訊息摘要。
    - 比對這兩個訊息摘要是否完全一樣，如果一樣表示文件沒有被竄改。

4.2 操作介面

本系統的操作介面是直接嵌入到 Word 的功能表，使用者可以直接由 Word 的功能表，呼叫數位簽章的加簽及查核的功能，如圖 11 所示。

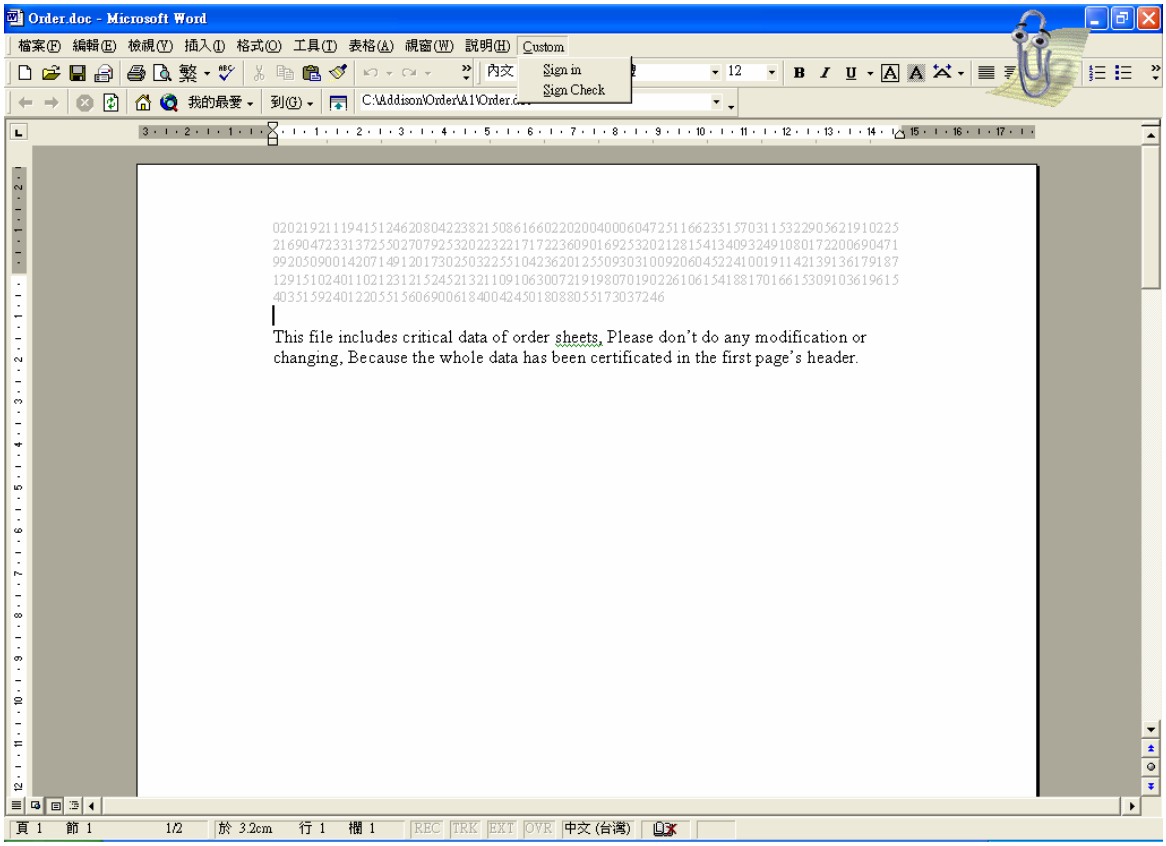


圖 11. 數位簽章系統的功能表

當呼叫數位簽章的加簽(Sign In)功能時，會跳出加簽(Sign In)的對話盒，如圖 12 所示，使用者需要輸入密碼，主要是私鑰(Private Key)被加密保護，使用者必須有正確的密碼，才能取出私鑰執行加密的功能，其目的是爲了防止私鑰被盜用。當輸入的密碼錯誤時，系統會偵測出私鑰解密的錯誤，而告知使用者密碼錯誤，無法執行數位簽章的加簽功能。

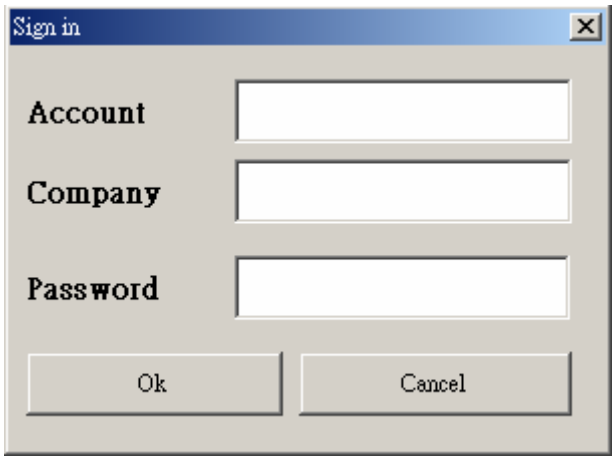
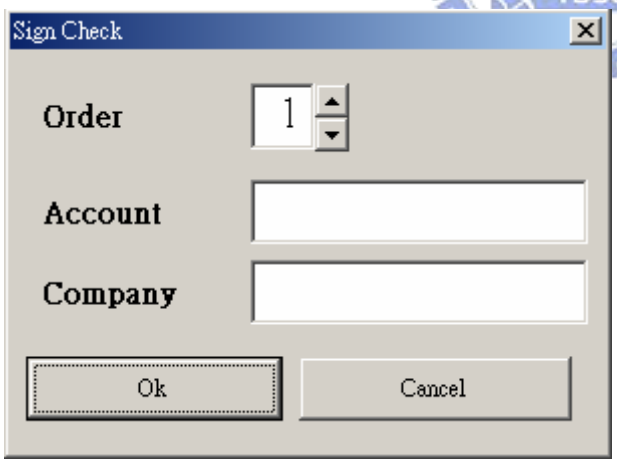


圖 12. 數位簽章簽核的對話盒

當呼叫數位簽章的查核(Sign Check)功能時，會跳出查核(Sign Check)的對話盒，如圖 13 所示，使用者需要輸入順序(Order) 、帳號及公司名稱，由於本系統可以執行多重加簽的功能，所以當執行查核時，系統需要知道是要對那個簽章做查核。至於輸入帳號及公司名稱的目的，主要是公鑰(Public Key)儲存在公鑰資料庫裡面，當要取出來使用時，需要用帳號及公司名稱當索引，才能找到對應的公鑰(Public Key) ，並且透過網路傳回，以執行簽章的解密。

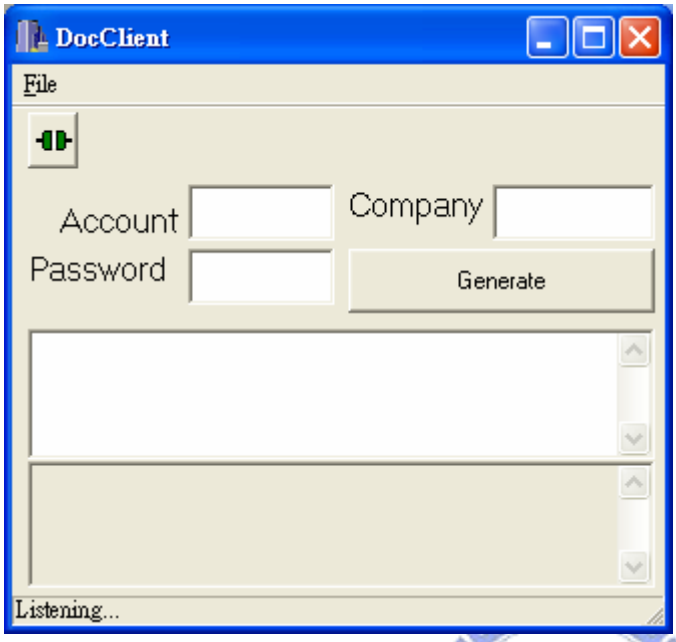


Order 選項的目的，是因爲本數位簽章系統允許多重簽章，所以執行查核時，必須告訴系統，是要針對哪一位簽章的擁有者執行查核。

圖 13. 數位簽章查核的對話盒

當使用者要產生自己的 RSA 公鑰(Public Key)及私鑰(Private Key)時，執行 DocClient.exe 執行檔，會有一個操作視窗，如圖 14 所示，使用者先連線到公鑰伺服器，而後輸入帳號、公司名稱及密碼，接著按 Generate 按鍵，程式會產生一組

公私鑰，並且將私鑰用輸入的密碼加密，存放在使用者的私人目錄，而公鑰會透過網路，傳送到公鑰伺服器儲存。



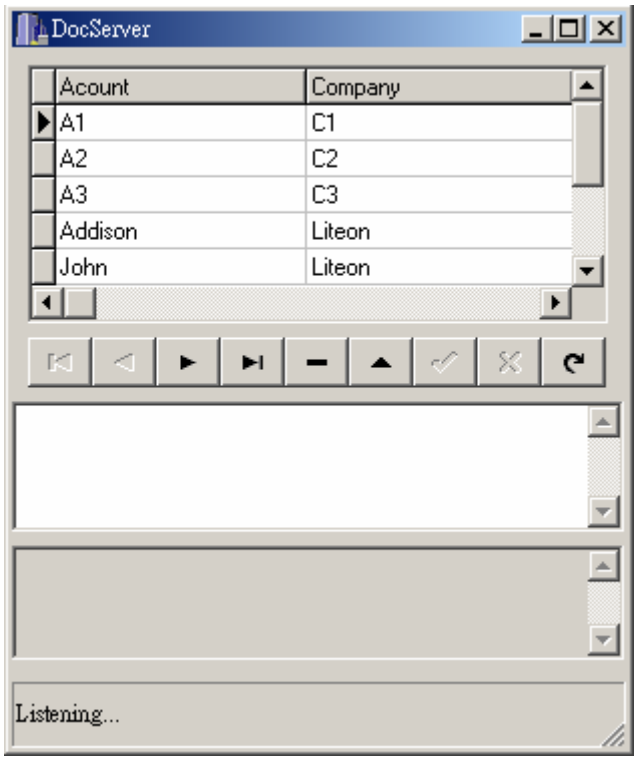
金鑰對產生器擔任兩種角色:

1. 產生 RSA 金鑰對。
2. 負責公鑰的取得，它是透過網路，將使用者的 Account 及 Company Name 傳給公鑰伺服器，以取得該使用者的公鑰。

圖 14. 金鑰對產生器

公鑰伺服器主要是擺放 RSA 公鑰(Public Key)的資料庫，當 DocClient.exe 傳送公鑰給公鑰伺服器時，會同時傳送公鑰擁有者的帳號及公司名稱，公鑰伺服器利用帳號及公司名稱建立索引，將公鑰儲存在資料庫。當使用者執行簽章查核(Sign Check)功能時，巨集指令會呼叫 DocClient.exe，並且透過 DocClient.exe 將使用者的公鑰，由公鑰伺服器取出，經由網路傳回到使用者的目錄。

公鑰伺服器由於需同時服務多個使用者，所以在設計階段就必須考慮以一對多的方式來設計程式，主要採取的方法乃是以 Multi-Thread 的方式，來滿足同一時間有多個連線需求的情形發生。每當有一連線需求產生時，就建立一個 Thread 來服務該連線的使用者，所以有可能同時有多條的 Thread 被開啓，當某一連線被關閉時，就將建立的 Thread 刪除，以便將該 Thread 所佔用的資源還給系統。



公鑰伺服器擔任兩種角色:

1. 負責公鑰的保存。
2. 依據金鑰對產生器要求的 Account 及 Company 名稱，將對應的公鑰找出來，傳回給金鑰對產生器。

圖 15. 公鑰伺服器

我們在系統分析的階段，將系統分爲四個部分，一、Macro 巨集指令，主要是負責 Word 文件的文字處理，從現在已開啓的文件中，取出所有的字元存放到暫存檔，或者將數位簽章二進位轉成文字及文字轉成二進位的數位簽章。二、加解密引擎，負責處理所有與加解密相關的計算工作。三、金鑰對產生器，產生加解密所需要的公私鑰及負責將擺放在公鑰伺服器的公鑰，透過網路取回，儲存到使用者的目錄。四、公鑰伺服器，負責使用者公鑰的管理及存取。這四個部分看似各司其職，互不相關，但實際上是由 Macro 巨集指令掌控，負責所有子系統的呼叫及聯繫，使整個系統運作達到完全自動化的執行。在 4.3 節我們將以程式流程圖說明，讓整個系統的輪廓及運作方式，有更清楚的表明。

4.3 程式流程

1. Sign In 程式流程圖

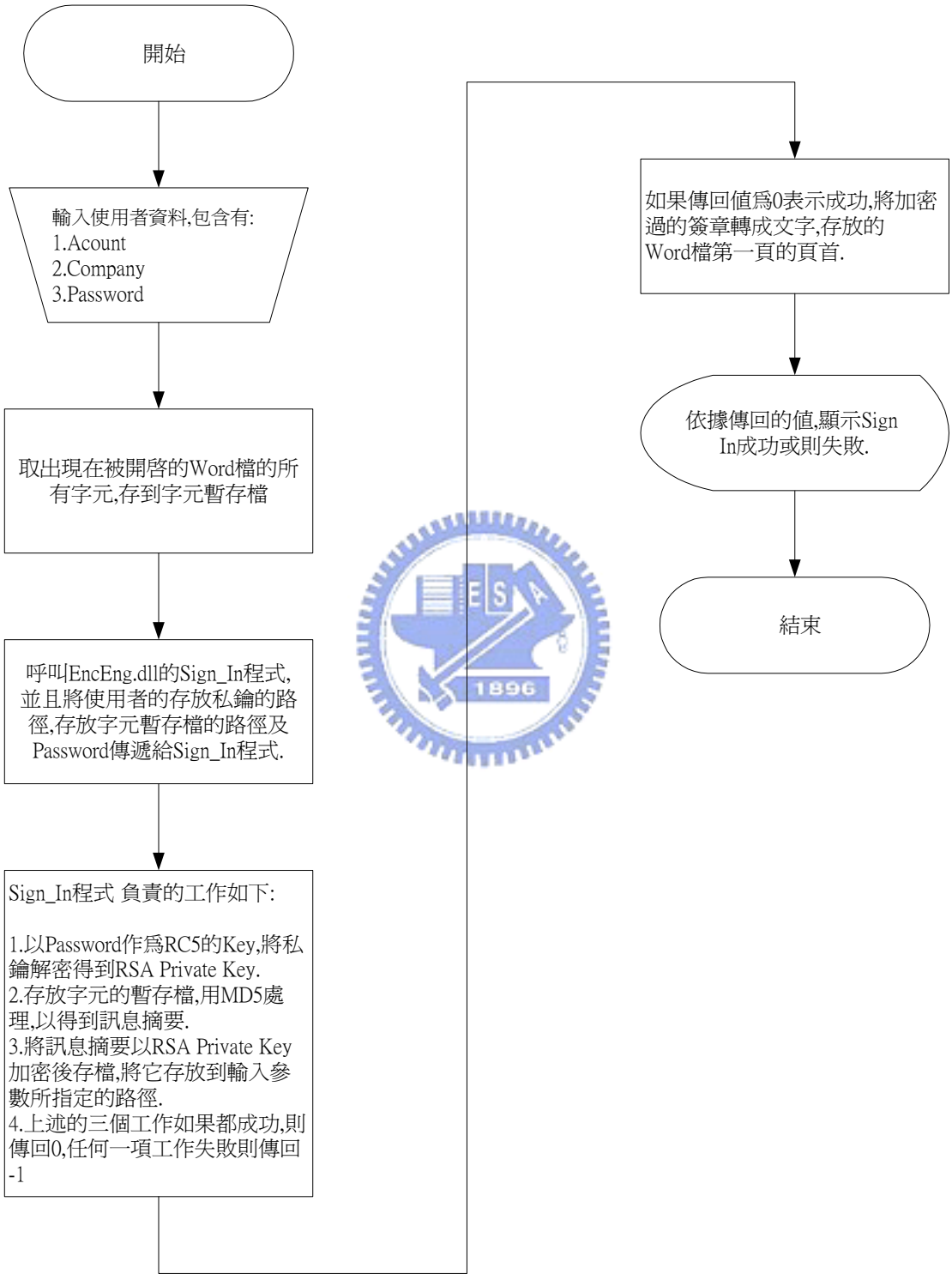


圖 16. SIGN IN 程式流程圖

2. Sign Out 程式流程圖

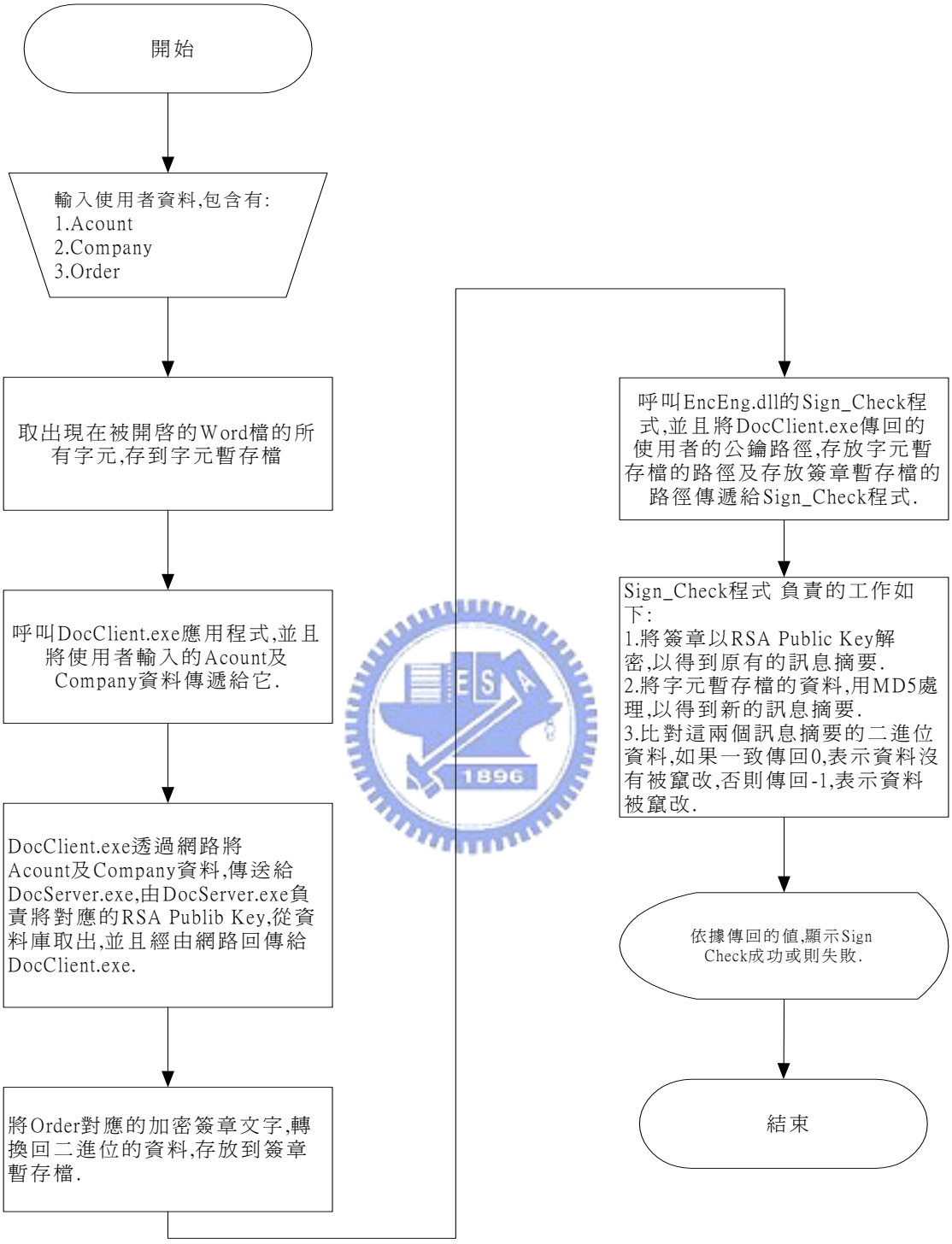


圖 17. SIGN OUT 程式流程圖

4.4 數位簽章 Plug-in 到 Word 編輯器

當要將數位簽章 Plug-in 到 Word 編輯器時，需要呼叫 Word 功能表上工具選項的巨集功能，挑選巨集名稱 AddCustomMenu，該巨集指令會建立一組客製化的表單，裡面包含有 Sing in 及 Sing Check 的功能選項。

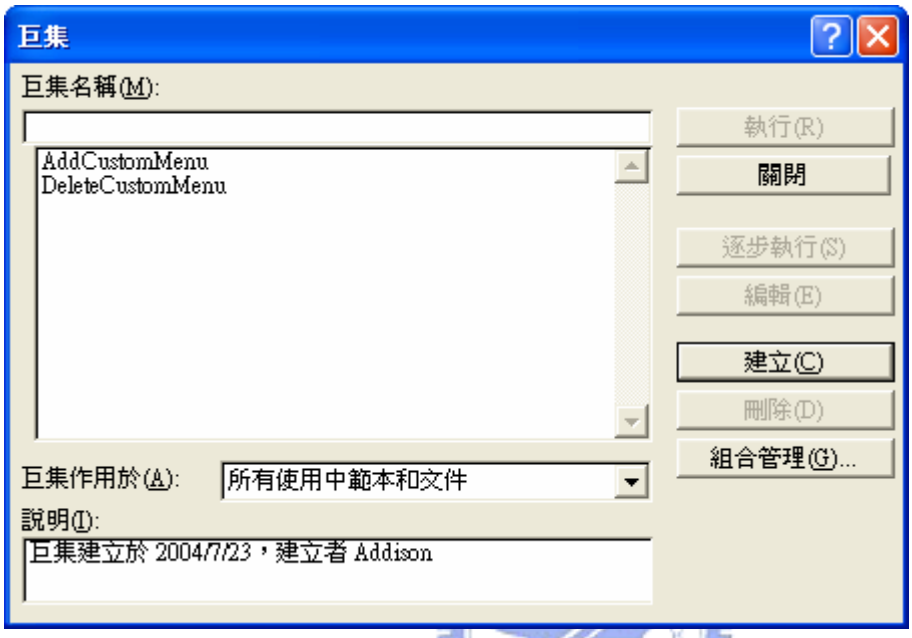


圖 18. Word 巨集指令呼叫

要建立一組客製化的表單，其語法如下:

```
' Create a popup control on the main menu bar
Set cbpop = CommandBars("Menu bar"). Controls.Add(Type:=msoControlPopup)
cbpop.Caption = "&Custom"
cbpop.Visible = True

' Add a menu item of Sign in
Set cbctl = cbpop.Controls.Add(Type:=msoControlButton)
cbctl.Visible = True
cbctl.Style = msoButtonCaption
cbctl.Caption = "&Sign in"
cbctl.OnAction = "SignInMacro"

' Add a menu item of Sing Check
Set cbctl = cbpop.Controls.Add(Type:=msoControlButton)
cbctl.Visible = True
cbctl.Style = msoButtonCaption
```

```
cbctl.Caption = "&Sign Check"
cbctl.OnAction = "SignCheckMacro"
```

4.5 系統評估

本系統經過系統分析、系統設計與系統實作後，完成整個系統的開發。但爲了驗證整個系統運作的成效，需要有個完整的實機測試，以作爲系統進一步修改的參考，其測試數據如圖 19 所示。

硬體配備 : Processor Pentium M 1.4 GHz
RAM 128M
作業系統 : Windows XP
應用程式 : Word Editor
資料庫 : Borland BDE

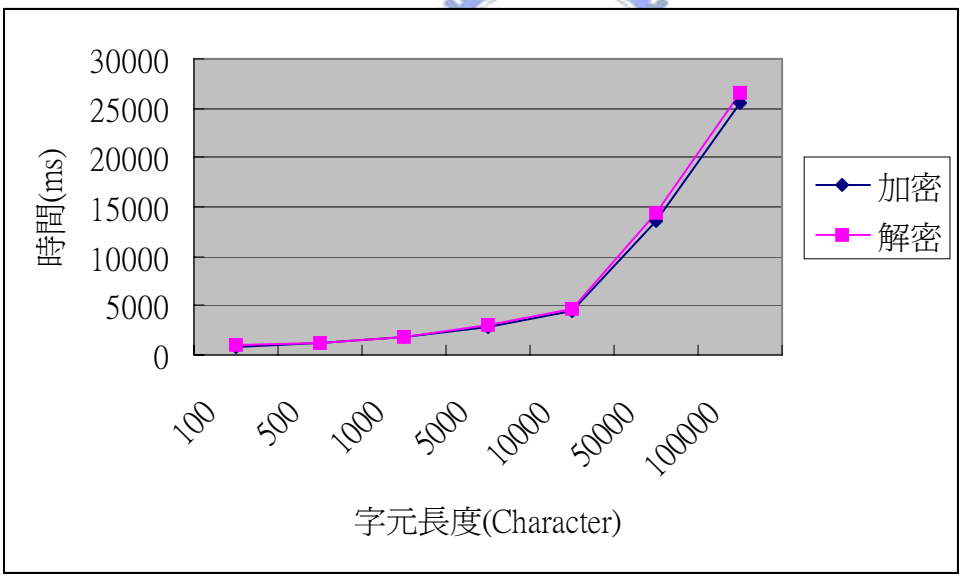


圖 19. 系統修正前加解密耗費的時間

從圖 19 上的測試數據，我們可以發現整個系統的運算速度確實不夠理想，因爲太長的等待時間，會降抵使用者使用的意願，經過仔細的分析之後，我們確實找到影響系統運算的瓶頸點在於跟指數運算相關的 Function 上，我們特別針對這幾個 Function 做最佳化的處理之後，重新完成實機測試，其測試數據如圖 20 所示。

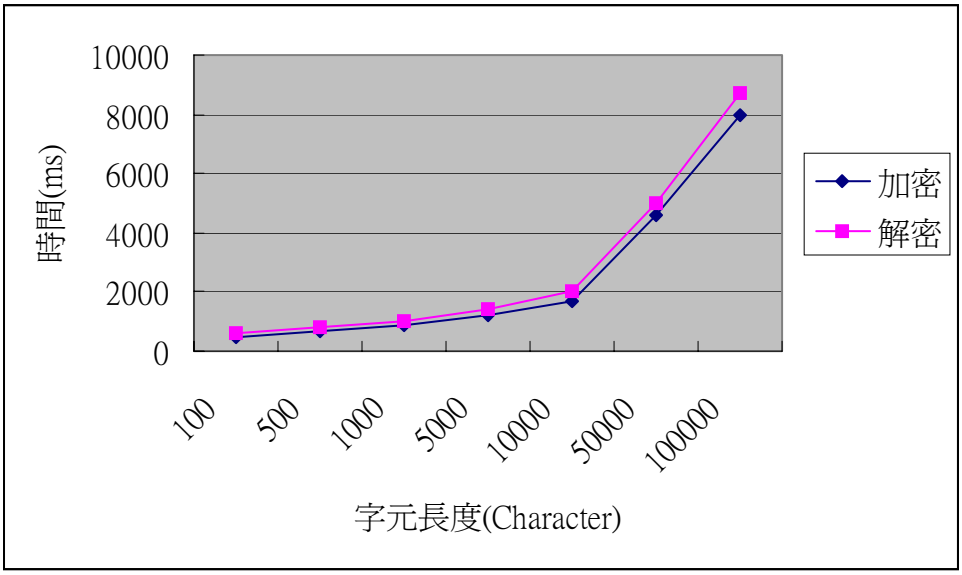


圖 20. 系統修正後加解密耗費的時間

本章節主要是討論數位簽章系統的實作，一個完整的系統開發，必須要有系統分析、系統的流程設計、使用者的操作介面設計、完整的系統測試及系統評估。當然還必須蒐集使用者的使用意見，系統上線的測試報告，做爲系統修正改版的參考依據。

五、結論

5.1 總結

今日，隨著網際網路交易的日益普及，人們開始追求以電子文件取代傳統的書面文件，由於數位簽章的發明及應用，可以有效防止電子文件被竄改及僞造，我們可以確定電子文件將逐步取代書面文件，而電子簽章也將逐步取代書面簽章。

嵌入式 Word 編輯器之數位簽章系統的開發目的，就是希望能在這股電子商務的潮流中，開發一套可以直接嵌入已被廣被使用的 Word 文字編輯器的數位簽章系統，以幫助使用者可以非常順利的進入數位簽章的時代。

爲了達到上述目的，本系統在開發時的系統設計方向如下：

- 操作簡便：本系統的操作介面是直接嵌入到 Word 的功能表，使用者可以直接由 Word 的功能表，呼叫數位簽章的加簽及查核的功能。
- 易於維護：本系統的開發方法是採用模組化設計，系統人員可以根據個別的需求，針對個別的模組維護修改。
- 易於推廣： 由於 Word 文字編輯器已被廣泛的使用在各行各業，基本上大家對於 Word 編輯器的使用不會有恐懼感，但是對於電子簽章的使用，卻是懷有相當的不安與恐懼，其中一部分的原因就是對它的不熟悉，如果能夠將電子簽章直接結合到 Word 編輯器，應該可以幫助電子簽章的推廣。

5.2 未來工作

本系統基本上是個封閉系統，所有的公私鑰都是系統自己產生的，但是對於一套可以被接受的商業系統它必須達到下列的需求:

- 開放的系統：要能跟外部的系統溝通，也就是說可以透過已認證過的網站，直接使用網站核發的憑證，作爲簽章的公私鑰，如此，才能廣泛的被使用。
- 整體的美觀性：由於本系統的數位簽章是放在第一頁的頁首，雖然不在文字的編輯範圍內，但是還是會影響整體的外觀。
- 操作的自動化：對於系統的安裝及操作的介面，可以更進一步的自動化，以方便使用者的使用。

參考文獻

1. 張真誠(1990)：電腦密碼學與資訊安全，松崗電腦圖書資料有限公司，1990 年十月第二版。
2. 賴溪松、韓亮與張真誠(1995)：近代密碼學及其應用，松崗電腦圖書資料有限公司， 1997 年五月第二版。
3. 楊宗誌(2001)：C++ Builder 資料庫程式設計，文魁資訊股份有限公司， 2001 年四月初版。
4. 經濟部網際網路商業應用網站，URL： http://www.ec.org.tw/ 。
5. OpenSSL Web Site，URL： http://www.openssl.org/。
6. 科技資訊網 Web Site，URL： http://taiwan.cnet.com/。
7. 漢龍資訊科技網 Web Site，URL： http://www.hanglong.com.tw/mds.htm
8. 財團法人資訊工業策進會 Web Site，URL： http://www.iii.org.tw。
9. 林峻立，”使用者通行碼之身份驗證與金鑰交換協定”，中華民國資訊安全學會，資訊安全通訊第九卷第三期，民國 92 年 6 月。
10. Stallings, William, Cryptography and network security : principles and practice, New Jersey, Prentice-Hall, Inc, 1998 2nd edition.
11. Steve Burnett and Stephen Paine, RSA Security's Official Guide to CRYPTOGRAPHY, Berkeley,California ,McGraw-Hill Company, 2001.
12. L. Harn, and H.Y. Lin, “An authenticated key agreement protocol without using one-way function”, Proceeding of 8th Information Security Conference, Taiwan, May 1998, pp. 155-160.
13. J.C Cha, and J.H. Cheon, “An identity-based signature from gap Diffie- Hellman groups”, Public Key Cryptography-PKC 2003, Springer-Verlag LNCS 2139, 2003, pp. 18-30.
14. W.C. Ku, and S.D. Wang, “Cryptanalysis of modified authenticated key agreement protocol”, Electronics Letters, Vol. 36, No. 21, 2000, pp. 1770-1771.
15. Y. S. Chang, T. C. Wu, and S. C. Huang, "ElGamal-like digital signature and multisignature schemes using self-certified public keys," The Journal of Systems and Software, vol. 50, pp. 99-105, Feb. 2000.
16. S. W. Changchien, M. S. Hwang, and K. F. Hwang, "A batch verifying and detecting multiple RSA digital signatures," International Journal of Computational and Numerical Analysis and Applications, vol. 2, no. 3, pp. 303-307, 2002.
17. T. S. Chen, T. P. Liu, G. S. Hwang, and Y. F. Chung, "An improvement of proxy-protected proxy multi-signature scheme," Proceeding of 13th International Conference on Information Management, R.O.C., pp. 33-40, 2002.

18. I. B. Damgard, "A design principle for hash functions," in Advances in Cryptology, CRYPTO'89, pp. 416-427, 1989.
19. D. E. R. Denning, Cryptography and Data Security. Massachusetts: Addison- Wesley, 1982.
20. W. Diffie and M. E. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, vol. IT-22, pp. 644-654, Nov. 1976.
21. W. H. He, "Digital signature scheme based on factoring and discrete logarithms," Electronics Letters, vol. 37, no. 4, pp. 220-222, 2001.
22. M. S. Hwang, E. J. L. Lu, and I. C. Lin, "A practical (t, n) threshold proxy signature scheme based on the RSA cryptosystem," IEEE Transactions on Knowledge and Data Engineering, vol. 15, no. 5, pp. 1-9, 2003.
23. M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," Technical Report, MIT/LCS/TR212, MIT Lab., Computer Science Cambridge, MA, USA, January 1979.