致理技術學院

資 訊 網 路 技 術 系

實 務 專 題 報 告

具 SHA-256 功能之資料確認平台

指導教師：蕭勝華

學生：張修維(29534514)

馮星豪(29534521)

許鈞皓(29534525)

許哲源(29534543)

中華民國 96 年 12 月

致理技術學院

資 訊 網 路 技 術 系

實 務 專 題 報 告

具 SHA-256 功能之資料確認平台

學生：張修維(29534514)

馮星豪(29534521)

許鈞皓(29534525)

許哲源(29534543)

本成果報告書經審查及口試合格特此證明。

指導教師：_____

中華民國　96 年　12 月

專題研究授權書

本授權書所授權之專題研究為 張修維、馮星豪、許鈞皓、許哲原

共 4 人，在致理技術學院資訊網路技術系 九十六 學年度第 一 學期完成資網實務專題。

專題名稱：具 SHA-256 功能之資料確認平台

■同意　□不同意

本組同學共 4 人，皆同意著作財產權之論文全文資料，授予教育部指定送繳之圖書館及本人畢業學校圖書館，為學術研究之目的以各種方法重製，或為上述目的再授權他人以各種方法重製，

不限地域與時間，惟每人以一份為限。

上述授權內容均無須訂立讓與及授權契約書。依本授權之發行權為非專屬性發行權利。依本授權所為之收錄、重製、發行及學術研發利用均為無償。上述同意與不同意之欄位若未勾選，該組同學皆同意視同授權。

指導教師姓名:

專題學生簽名:　　學號:

中華民國　　年　　月　　日

誌　　謝

從專題製作的開始到結束以經過了一年下來，在這一年中經歷了許許多多的難關以及困境，但是我們的成功，不是只因為自我的努力和奮鬥，是因為有著更多付出關心和心力來幫助我們的人，除了自己班上的同學，他們發揮出同學之間應有的同袍之愛跟一起奮鬥過來的精神，在精神跟實質上給了我們許多的幫助，沒有了他們，我們也不會在專題成發的時候有如此傑出的表現。

接下來，我們所要感謝的是，在這個專題上面給我們最大的幫助的人，那就是我們的指導老師，可是我們這組的經過跟其他組有所不同，在暑假的期間，因前任的指導老師有事務纏身，因此在依依不捨的情況之下，離開了我們，但是我們還是打從心底，很感謝這位指導老師，那就是我們的王勝石老師，三年級下學期在他的指導之下，我們從中學習到了很多，除了知識上實際收穫以外，心理面也成長了許多，雖然無法在您的指導之下完成此專題，但是我們不會忘記您對我們的恩惠跟指教。

接下來我們所要感謝的是，接任王勝石老師繼續指導我們的老師，就是蕭勝華老師，雖然他中途接手了我們，但是他保持著不放棄我們的

精神，對我們不斷的打氣，讓我們不會在挫折中失敗或是放棄，也因為這樣我們才會有現在的成就，謝謝大家，因為你們，我們才會有今天的成就。

摘　　要

在網路傳輸檔案已經是日常生活中不可或缺的事實，本組對於此構想，利用 C#程式語言設計對檔案資料確認的正確性，我們採用 SHA-256 演算法來製作對檔案的資料演算，使用 ASP.NET 網頁程式語言來搭配製作檔案交換的分享，兩者結合則構成為具有 SHA-256 功能之資料確認平台。

在製作的過程中，我們也有想過要將其他可以做完資料確認的演算法，由於 MD-5 與 SHA-1 已經被人發現出破解的方法，已經變為不具有安全性的演算法，於是我們採用 SHA 系列更高一層的 SHA-256 演算法。

除了 C#採用 SHA-256 的演算法來製作資料確認平台，ASP.NET 也成為我們的一把利刃，兩者程式語言均可使用 Visual Studio 2005 來製作。

關鍵詞：資料確認、SHA-256、演算法、C#、ASP.NET

目　　錄

圖　目　錄

表　目　錄

第一章 緒論

第一節　重要性與發展演進

人類之所以為萬靈之長，所依靠的就是經驗的傳承。這些寶貴的經驗傳承，讓我們擁有進步的能力，以及傲視萬物的自信，而經驗的傳承就是靠訊息的傳遞，在遠古時期，紙張尚未發明，就無所不用其極的想盡辦法留下訊息，甚至在石頭.木頭.竹片等，刻畫出各式各樣記號以求留下資訊，並希望後世能保存觀看。

到了紙張發明之後，資訊傳遞顯得更加確實方便，也造就我們炎黃子孫五千多年的輝煌歷史，有古人曾言：『烽火連三月，家書抵萬金』，由此可知人類對重要訊息渴求，紙的發明造就人類輝煌的歷史，而電腦網路就如同紙張一般的重要，它讓現今社會資訊流通的更加方便更加確實，甚至能在短短時間內獲得最新資訊，但是過多的資訊也造成使用者選擇上的困難，有人稱這電腦網路發達的現代，為資訊爆炸時代。

其實早期的電腦網路並不發達，檔案傳輸速率很差，使用花費又高，可以說是一個好看卻不實用的花瓶，更徨論使用電腦網路做為檔案交換的媒介，但是由於現今的社會，各式各樣技術正不斷的創新改變，在不斷提升技術同時也大大提升軟硬體能力，以至於現今新興產品的設計優良，打破以往檔案傳輸速率差，傳輸檔案小的窘態。

現今社會使用電腦網路的人變多，對於資料檔案交換的重要性，可說是與日俱增。其中有些地方是我們必須注意的，譬如檔案在交換傳輸過程中，檔案的完整性、檔案的安全性、檔案機密性..等等，這些問題也說明檔案交換上可能會有的風險，甚至會影響我們在使用上的不便。

第二節　研究動機與目的

電腦網路的興起，造就了網路技術突破性的發展，在使用上方便性與日俱增，其中一項應用就是檔案交換，檔案交換的便利性，造福了許多人，舉凡是電影下載、音樂分享，資料的流通，或是訊息的傳遞等等，都能以不需親自接觸下在短短的時間獲得。

想想看既然不是對方親手交給你，萬一你所獲得的檔案在傳輸過程中，被人動手腳或是被人放入後門程式，而造成機密資料外洩，或是因不可抗拒之因素造成毀損的不完全檔案，這些可都是非常嚴重的事，雖然說檔案傳輸的基本規範已有相當水準，但也免不了他人有心的破壞，更何況台灣有句厘語：『雞蛋殼再密也會有縫』，說明了世上無絕對。為防萬一，檔案交換的安全驗證就顯得特別重要，目前所運用的加解密安全驗證技術，是以數學的雜湊函數經過編碼而成，隨著不同方式的編碼，有著不同的結果，就算旁人從中攔截加以改變，也能很容易的找出被改變的地方，以保證檔案的資料完整性。這些重要性的技術，引起我們組員強烈的興趣，於是檔案交換的安全驗證就成為我們所欲研究的方向與目標，我們將使用C# 和ASP等語法結合網頁技術做出一連串的程式，來測試檔案在交換過程中的完整性和安全性。

第二章　理論與技術探討

第一節　資料確認

在我們的專題中資料確認就是使用單向雜湊演算法[1][2](one-way hash function)（又被稱為訊息指紋(message fingerprint)演算法或訊息摘要(message digest)演算法）。在此方法中，不一定使用到金鑰，但和許多重要的密碼演算法相關。任意長度的輸入訊息資料（通常是一整份文件），透過單向雜湊演算法的計算可得一個較短固定的長度的訊息雜湊值。這個過程是單向的，由訊息摘要逆向操作反推原輸入訊息，從計算理論上來說是難以完成，而且找出碰撞（兩個不同的輸入產生相同的雜湊值），從計算理論上來說也是很困難的。因為任何對輸入訊息的變動，都會使得湊雜演算法輸出的雜湊值受到變更，所以碰撞的機率非常小。

圖 2-1 雜湊演算法流程

關於碰撞的抵抗性(collision resistance)[1]，兩個不同的資料訊息 Ma 與 Mb 被計算出相同的輸出訊息指紋是相當困難的，若我們以符號 H 代表單向雜湊演算法，則要從單向雜湊演算法的輸出 Y = H(x)找出 x，在計算上是不太容易的，這也就是為何稱為單向的原因。我們希望在計

算上做到若 Ma ≠Mb，則 H(Ma) ≠H(Mb) 。如果訊息摘要長度為 n 位元，則有強碰撞抵抗性的單向雜湊演算法的安全性約為（利用所謂的生日攻擊法評估 $\sqrt{2}=2^{n/2}$ ）

而單向雜湊演算法用於訊息防偽的方式[1]，就是訊息發送者先將訊息與訊息認證用的金鑰作為單向雜湊演算法的輸入，計算訊息摘要，再取訊息摘要的某些位元附加在原有訊息當作所謂的訊息認證碼(message authentication code)。

訊息與訊息認證碼將傳送給接收者，而兩者皆可能在傳送過程中被竄改，然而接收者可以重新計算收到之訊息與秘密金鑰的雜湊演算法值，再取雜湊演算法值的某些位元與收到的訊息認證碼做比較，相同時才接受，不同時則代表資料或認證碼被竄改。

我們在專題中製作的資料確認，就是利用這種訊息認證方式來完成的。

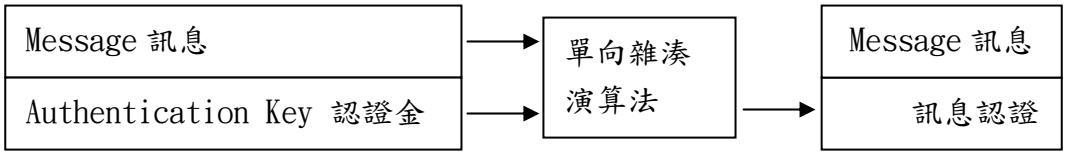

圖 2-2 產生訊息認證碼流程

第二節　網路安全常用演算法

(一)非對稱式加解密演算法[3][4]：

每個使用者都擁有一對公開金鑰(public key)和私密金鑰(private key)，公開金鑰就是可對外公佈的的金鑰，私密金鑰則是收件者所私人擁有的金鑰，而且幾乎不可能從公開的金鑰推算出私密金鑰。且此運算法還運用在數位簽章上面，使用私人擁有的金鑰甚至比當面筆跡鑑識更不容易遭到仿冒，因此運用此演算法的數位簽章具有不可否認性。

而非對稱式演算法最常使用的演算法是 RSA 密碼系統， RSA 密碼系統是在 1978 年由美國麻省理工學院三教授 Rivest、Shamir 及 Adleman 所共同研發 RSA 密碼系統是以因數分解為基礎。

非對稱式演算法。如圖 2-3 顯示基本架構。

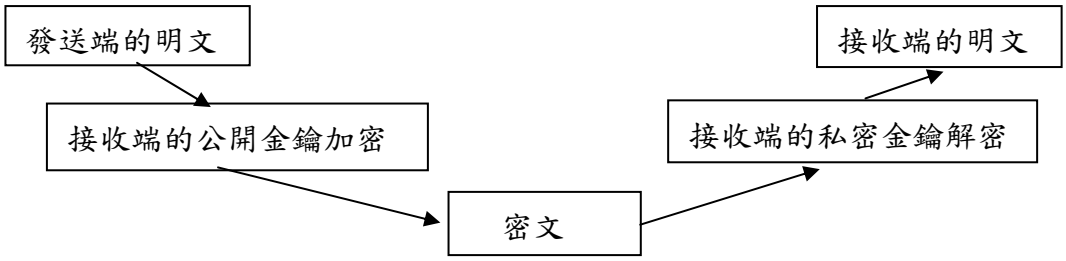

圖 2-3 非對稱式的發送端&接收端的金鑰運作方式

1、數位簽章:

數位簽章又稱為電子簽章，使用者每個人都擁有獨自的私密金鑰以及用來驗證簽章正確性的公開金鑰，在傳送資料時，先用獨自的私

密金鑰做簽證，當接收端收到資料時，會用公開金鑰來驗證資料的正確性。

(1)安全性

在發送端傳送資料時，只有自己擁有私密金鑰，其他接收端都只有公開金鑰　所以在安全性上多了一層保障，也較不易被破解。例:在一個客戶端在使用ATM存款的時候，所使用的密碼就是私密金鑰，而銀行在接收客戶資料的時候，只能用公開金鑰來解出客戶端所傳輸的資料，因此要在傳送中破解資料非常不易，才會有所謂的不可否認性。

(2)不可否認性

使用者端在文件的簽收是使用私密金鑰做簽證下，安全性提高許多，因此當 有第三者否認此資料的正確性及公信力時，接收端可提出此簽章來彰顯其文件的正確性。

(二)對稱式加解密演算法[3][4]：

使用者不管是接受還是發送端都使用相同的金鑰進行加解密的動作，因此無法應用在數位簽章上而減少了完整性以及不可否認性。

常見的對稱式演算法所使用的演算法種類如下：

1、Data Encryption Standard (DES):

它是 IBM 所研發而成的，在 1997 年被美國政府採用。尤其是在保護金融資料的安全中，最初開發的 DES 是嵌入硬體中的。通常

自動取款機（ATM）都使用 DES。

(1)早期之中廣泛使用的對稱金鑰的演算法。

(2)1977 年由美國國家標準與技術協會(NIST)採用為聯邦資訊處理標準。

(3)利用混淆(Confusion)與擴散(Diffusion)原理。

a.混淆就是將明文轉換成其它的檔案格式，讓金鑰和密文兩者之間複雜化。

b.擴散是指明文中的任何一個地方的增加或是減少其內容造成明文的全文有所改變。

c.DES 採用 56 位元的金鑰來對 64 位元的資料區段進行加密，需經 16 回合的運算。

d.主要缺點：56 位元的金鑰長度過短，以現在電腦的計算能力，不需要花費多少時間即可求出 DES 金鑰。

2、Triple DES (3DES)：

利用 DES 為基準的進階版加密方式，利用三把不同的金鑰，進行三次 DES 的運算，讓安全性提早了許多。

(1)1992 年期間，研發人員發現 DES 可以反覆計算來增加強度，因此 3DES 應運而生。

(2)3DES 可以使用二把至三把金鑰，如果是二把，則金鑰一和金鑰三是一樣的，金鑰二為不同，來進行加解密的動作。

(3)3DES 為 168 位元金鑰。

(4)比其它演算法較慢。

(5)類型：DES-EEE3、DES-EDE3、DES-EEE2、DES-EDE2。

其 3DES 的運作方式，如圖 2-4

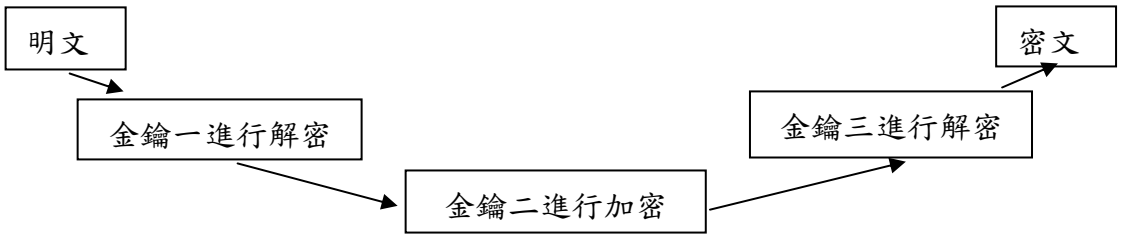

圖2-4 3DES的金鑰運作方式

3、Advanced Encryption Standard（AES）：

由 Joan Daemen 和 Vincent Rijmen 所設計，是新一代加密標準 AES(Advanced Encryption Standard) 的演算法

(1)NIST 為了取代 DES 於 1997 年公告徵求下一代的區塊加密碼器 AES，以保護 (sensitive)但非機密(unclassified)的聯邦資料。

(2)2000 年 10 月，NIST 宣佈來自比利時(Belgium)的兩位密碼學者 Joan Daemen、VincentRijmen 所提出的 Rijndael 演算法贏得這項徵選活動並作為新一代的加密標準。

(3)Rijdael 的強度高、適合用於高速網路並且容易在硬體上實作。

(4)AES為區段式加密技術，使用的區塊大小為128位元，而金鑰大小為128、192、256位元三種選擇。

(三)摻雜值運算法[3][5]：

在發送端時給予傳送的檔案，一組隨機的雜湊值，只要檔案不被竄改

其雜湊值就會永遠相同，因此在傳送時在接收端接收完畢後，會在運算一次，只要雜湊值跟發送端給予的一樣，就沒有被遭其竄改。

1、雜湊函數：

(1)雜湊函數是給予檔案一個隨機的亂數且長度固定跟固定的輸出，此輸出訊息為雜湊值(Hash Value)或訊息摘要(Message Digest)。

2、應用：

(1)數位簽章。

(2)確保資料的完整性。

(3)訊息確認。

(4)密碼儲存。

(5)雜湊函數的特性：

(6)單向的摻湊函數，無法反相推算出其函數值的內文。

(7)摻雜的函數，需從內文的改變而跟著改變。

(8)將僅無法從不同的文件，找出相同的摻雜值。

(9)外界稱為數位指紋。

(10)只要明文內容有一小部分更改，其會影響整個密文的摻雜值。

而常見的摻雜值演算法法種類如下。

1、MD-5：

在90年代初由MIT和RSA Data Security Inc發展出來的，經過

MD-2、MD-3和MD-4改良而出。

(1)為Ron Rivest 設計的MD-5是MD-4及MD-2的改良版本，較MD-4及MD-2複雜而安全，但稍慢。

(2)MD-5 會將512 位元區塊分成16 個32 位元的區塊，來處理輸入文字。

(3)輸入為一個 512 位元的區塊，輸出為一個 128 位元的訊息摘要。

(4)UNIX/Linux 的 shadow 密碼就是此種加密技術。

2、SHA：

安全雜湊演算法(Secure Hash Algorithm)為國家標準與技術協會(NIST)所發展出來，目的為支援數位簽章標準(DSS)所需要的雜湊演算法。

(1)輸入的訊息不能超過 264 個位元，會被分成多個 512 位元的區段來處理。

(2)SHA 產生 160 位元的雜湊值。

(3)比 MD-5 能夠預防駭客的入侵(因為多了 32 位元) 。

(4)而電子郵件安全性協定 PGP 就是使用此種演算法。

(5)SHA有出SHA-1及SHA-256這幾種的改良版，而本專題實驗是使用SHA-256演算法來執行網路安全的正確性。

(6)SHA-1是將任意字串轉換成160bits的整數，比MD-5足足多了32bits，破解的難度比MD-5更高，而SHA-1跟MD-5是個單向的摻湊函數，無法反推算出其函數值的內文，換句說字串有許多組但卻無法從這些摻雜函數推算回許多的字串。

關於對稱式演算法及非對稱式演算法個別的差異性如表 2-2-1

表 2-1 對稱式演算法及非對稱式演算法個別的差異性

	非對稱式演算法	對稱式演算法
其它名稱	公開金鑰加密法	秘密金鑰加密法
金鑰是否公開	公開金鑰可以公開 私密金鑰不可公開	不可公開
加解密金鑰是否相同	接受端跟發送端使用金鑰兩者不同	接受端跟發送端使用金鑰兩者相同
加解密金鑰保管問題	只需要保管自己的私密金鑰其公開金鑰多少人擁有都沒關係	如果個人交換訊息就需要保管 N 個加解密的金鑰
加解密的速度	慢（硬體執行非對稱式金鑰比對稱式 約快 1000 倍，軟體執行非對稱式金鑰比非對稱式 約快 100 倍）	快（硬體執行對稱式金鑰比非對稱式 約慢 1000 倍,軟體執行對稱式金鑰比對稱式約慢 100 倍）
運算法應用的地方	數位簽章	email

(7)SHA-1的運用地方:

a.確認檔案完整性

當檔案在上傳或是下載完成時，無法確認檔案是否遭到修改或是損壞的時候，驗證 SHA-1 所在此檔案設定的摻雜值是否完整相同即可確認資料的完整性。

b.密碼儲存

在網站上當使用會員註冊時，可使用 SHA-1 的演算將摻雜值匯入在資料庫內而不是將輸入的資料完整的彙入資料庫當中。如此可確保當駭客入侵所造成的會員資料外洩，另當登入時只要匯入與之前相同的密碼用 SHA-1 演算跟資料庫的所存摻雜值相同即可確認密碼的正確性。

其他 SHA-1 的運用地方跟 MD-5 大同小異而我們選擇使用 SHA-256 的主要原因是安全性較高。

第三節　MD-5 與 SHA-1、SHA-256 的比較

(一)Message-Digest Algorithm 5(MD-5) [3][6]:

在90年代初由MIT和RSA Data Security Inc發展出來的，經過MD-2、MD-3和MD-4改良而出。

MD-5是將任意字串轉換成128Bits的整數，而且MD-5是個單向的摻湊函數，無法反相推算出其函數值的內文，換句說字串有許多組但卻無法從這些摻雜函數推算回許多的字串。

而且當文件造到修改或是損壞時,MD-5再驗算的時候因摻雜值不同，介此發現檔案造到竄改，並且可以使用數位簽章來再次確認檔案的正確性及不可否認性。

1、MD-5的運用地方:

(1)確認檔案完整性

當檔案在上傳或是下載完成時，無法確認檔案是否遭到修改或是損壞的時候，驗證 MD-5 所在此檔案設定的摻雜值是否完整相同即可確認資料的完整性。

(2)密碼儲存

在網站上當使用會員註冊時，可使用MD-5的演算將摻雜值匯入在資料庫內而不是將輸入的資料完整的彙入資料庫當中，可確保當駭客入侵所造成的會員資料外洩，另當登入時只要匯入與之前相同的密碼用MD-5演算跟資料庫的所存摻雜值相同即可確認密碼的正確性。

2、MD-5的演算方式:

MD-5 是以16個32bits(4Byts)為一組的位元組合而成，即為512bits來提供摻雜值，經過排序完成後會產生4個32bits的數據，其最後組合成有128bits隨機的散亂數列，其數列的摻雜值將不會改變，當檔案遭到修改，數列的摻雜值也會跟著改變。

3、MD-5步驟如下：

(1)增加padding讓數據的長度能為448bits，如果不足448bits

的話，則增加padding讓其可以在有512bit內容值能放進448bits，其數據要為湊滿為1000000000數位數據。

(2)將數據的長度轉換成每個都為64bits,如超過64bits往後補齊直到剛512bits再將其分為16個32bits的整數，其32bits為一個word

(3)利用到4個變數，分別為A、B、C、D，均為32bit長。隨機亂數將其初始化為：

A: 65 78 4d ss

B: sl 14 56 ds

C: qw ac bv 80

D: sd fq uy wa

4、MD-5的計算方式如下。

$F(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z)$

$G(X,Y,Z) = (X \wedge Z) \vee (Y \wedge \neg Z)$

$H(X,Y,Z) = X \oplus Y \oplus Z$

$I(X,Y,Z) = Y \oplus (X \vee \neg Z)$

舉個簡單的例子，有一個txt再上傳時遭到竄改，最初時txt(“ 123456789s8f2s6q4e8”)，MD-5(“a55dfg1235d2d4d6f4s4f8s1vf65w4w5d5f”)遭到竄改時txt(“123456787s8f2s6q4e8”)，MD-5(“a55dfg12w5d2d4d6f4s4f8s1vf65w4w5d5f”)。

(只以當最後字串造到竄改為例)

(二)Secure-Hash Algorithm-1(SHA-1) [6][7]:

安全雜湊演算法(Secure Hash Algorithm)-1為國家標準與技術協會(NIST)所發展，其目的為支援數位簽章標準(DSS)所需要的雜湊演算法。

1、SHA-1的演算方式:

SHA-1是以16個32bits(4Byts)為一組的位元組合而成，即為512bits來提供摻雜值，經過排序完成後會產生4個32bits的數據每一個數據會執行20次的迴圈來運算出SHA-1的摻雜值，跟MD-5不同之處是，會產生32bits(4Byts)為一組的摻雜值，一共會有5組，而取得摻雜值方式也大為不同。

2、訊息擴充

(1)W[m], m=0, 1, ..., 79，W[0] 為一個 32bits 區塊 W[0] 至 W[1]為兩個 32bits 區塊以此類推，則 512bits 為 W[0] 至 W[15]

(2)W[16] 以後的填入方式：

W[t] = S1(W[t-16] ⊕ W[t-14] ⊕ W[t-8] ⊕ W[t-3]), t =16, 17, …, 79

譬如：

W[32] = S1(W[16] ⊕ W[18] ⊕ W[24] ⊕ W[29])

W[17] = S1(W[1] ⊕ W[3] ⊕ W[9] ⊕ W[14])

W[59] = S1(W[43] ⊕ W[45] ⊕ W[51] ⊕ W[56])

SHA-1的運算方式如圖2-5。

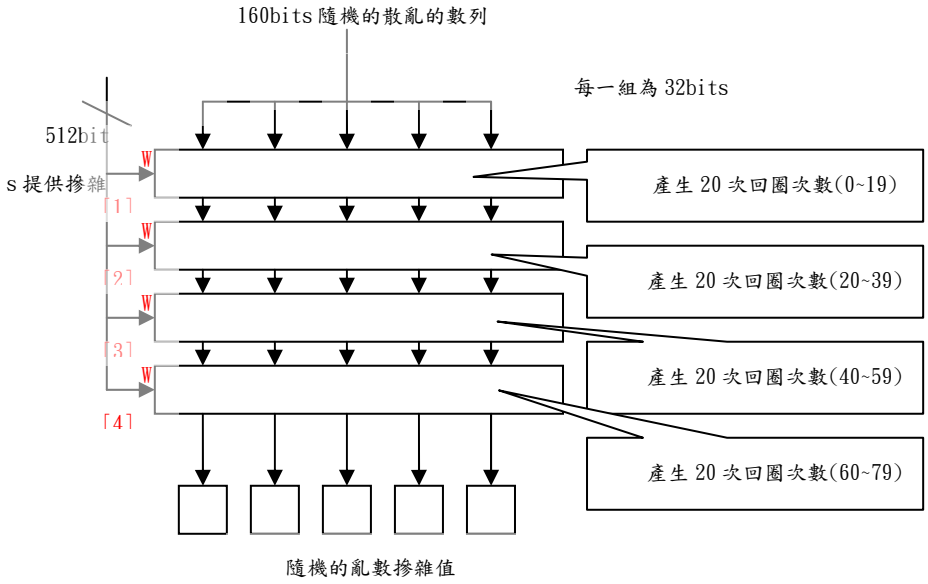

圖2-5 SHA-1取得摻雜值的運算方式

暫存器的初始值:

A: g5 q5 q1 c6

B: ds pk 13 d8

C: t2 j6 oh vj

D: xz nl 98 13

E: df y1 u6 7i

而我們這次專題所使用的就是SHA系列的SHA-256，為何不使用MD-5以及SHA-1我們在這向大家敘述我們選取SHA-256的原因。

3、SHA-1與MD-5慘遭破解[8]

(1)在2004年8月中國數學教授王小雲教授，在美國加州聖芭芭

拉召開的國際密碼大會上，發表MD-5、HAVAL－128、MD-4和RIPEMD四個密碼演算法，在他們工作團隊多年的努力下已經在理論在找出破解的方式以及問題所在，而在2005年2月的時候SHA-1也已經在理論下被破解了，而國際密碼學家Lenstra在王小雲教授所提供的MD-5破解理論下成功的破解MD-5(第2的63次方時破撞)，偽造了符合標準的數位證書，SHA-1也在理論上遭到破解，離實際被破解的日子應該不久遠。

(三)Secure-Hash Algorithm-1(SHA-256) [6][7][9]:

SHA-256是以16個32bits(4Byts)為一組的位元組合而成，即為512bits來提供摻雜值，經過排序完成後會產生4個32bits的數據會產生64次的迴圈，跟SHA-1不同之處是其最後組合成有256bits隨機的散亂的數列而取得亂數方式也大為不同。

1、訊息擴充

(1)W[m], m=0, 1, ..., 63，W[0] 為一個 32bits 區塊 W[0] 至 W[1]為兩個32bits 區塊以此類推，則512bits為W[0] 至 W[15]

(2)W[16] 以後的填入方式：

W[t] = S1(W[t-16] ⊕ W[t-14] ⊕ W[t-8] ⊕ W[t-3]), t =16, 17, …, 63

譬如：

W[32] = S1(W[16] ⊕ W[18] ⊕ W[24] ⊕ W[29])

W[17] = S1(W[1] ⊕ W[3] ⊕ W[9] ⊕ W[14])

W[59] = S1(W[43] ⊕ W[45] ⊕ W[51] ⊕ W[56])

SHA-256的W[16]~W[63]比SHA-1的W[16]~W[79]更為複雜。

SHA-256的運算方式如圖2-6。

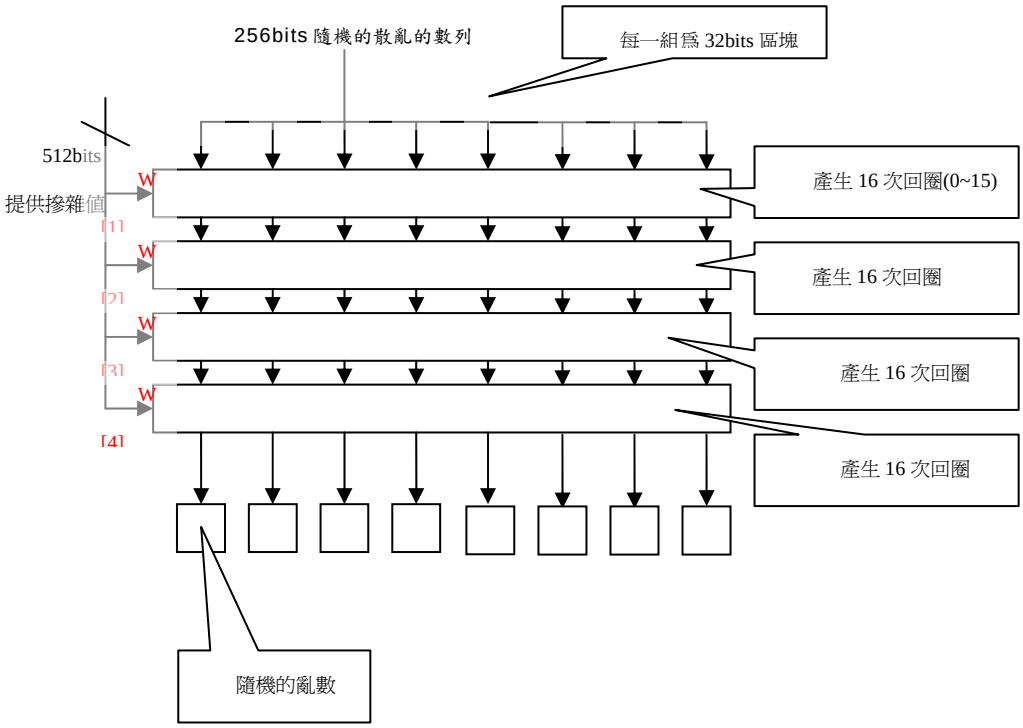

圖2-6 SHA-256取得掺雜值的運算方式

暫存器的初始值:

A: g5 q5 q1 c6

B: ds pk 13 d8

C: t2 j6 oh vj

D: xz nl 98 13

E: df y1 u6 7i

F:fd 56 qw 12

G:dc aq 5d io

H:ds 32 da sq

因此在往未來的方向進行時，我們選擇保密性以及破解困難度更高的SHA-256，藉此能在往後網站的經營發展中，暫時不會因為MD-5以及SHA-1演算法被破解的危機下，而使會員資料容易遭到破解外洩入不法份子的手中。

第四節　技術程式選擇

工欲善其事必先利其器，所以我們開始比較各設計程式軟體的優缺點，再選擇較適合的開發程式加以使用。以系統軟體介面來說有 C++、VB、C#、JAVA、等等相關軟體可以進行程式軟體介面開發設計。網頁的架構技術也有 ASP、ASP.NET、JSP、PHP、等等各式設計軟體可以使用。接下來將說明各設計軟體的優缺點以及我們所選擇的開發環境與原因。

(一)VB[15]

Visual Basic 的簡稱，是 Basic 進化改版而成，在早期 Basic 是微軟所推行的程式語言，為現任微軟總裁比爾蓋茲所發明的，應用於 DOS 環境下用來撰寫簡單的應用程式，而後為了應用於 Widows 環境下的開發與執行，就設計出一個全新的開發環境，其架構與功能與傳統的 Basic 差異非常大，VB 擁有圖形用戶界面（GUI）和快速應用程序開發（RAD）系統,可以輕易的使用 DAO、RDO、ADO 連接資料庫,或者輕鬆的創建 ActiveX 控制項。程式設計師可以輕鬆的使用 VB 提供的組件快速建立一個應用程序。

VB 中心思想就是要便於使用，無論是新手或者專家。因學習較容易上手，VB 是世界上使用人數最多的語言，而且 VB 的程序可以非常簡單的和資料庫連接。比如利用控制項可以綁定資料庫，這樣一來用 VB 寫出的程序就可以掌握資料庫的所有訊息而不用寫一行代碼。在使用上當然也有它的缺點， VB5 和 VB6 都是物件導向的程式語言，但是不包含繼承特性。VB 中提供了特殊的類的功能,但是還是不能滿足程式設計師的需求。VB.net 包含了所有物件導向的特性。

VB 對指針的支持非常有限。

VB 只能支持 8 到 32 位的整形，很多語言都有無限制的支持。

VB 不允許在任何數組內存在不變的變數。

VB 不支持以上特性，程式設計師需自己建構方法來實現相似功能。

2005 年，微軟宣佈將不會再對非.NET 版本的 VB 進行支持。也就是說這可能打擊到 VB 程式設計的未來發展性。

(二)C++[16]

C++是一種使用非常廣泛的電腦程式設計語言。C++是一種靜態資料類型檢查的，支援多範型的通用程式設計語言。它支援程序化程式設計、資料抽象化、物件導向程式設計、泛型程式設計等多種程式設計風格。

貝爾實驗室的比雅尼·斯特勞斯特魯普博士在20世紀80年代發明並實現了C++（最初這種語言被稱作「C with Classes」）。一開始C++是作為C語言的增強版出現的，從給C語言增加類別開始，不斷的增加新特性。虛擬函數（virtual function）、運算子重載（operator overloading）、多重繼承（multiple inheritance）、模板（template）、異常（exception）、RTTI、命名空間（name space）逐漸納入標準。1998年國際標準組織（ISO）頒布了 C++程式設計語言的國際標準 ISO/IEC 14882-1998。遺憾的是，由於C++語言過於複雜，以及他經歷了長年的演變，直到現在（2004年）只有少數幾個編譯器完全符合這個標準。

另外，就目前學習C++而言，可以認為它是一門獨立的語言；它並不依賴C語言，我們可以完全不學C語言，而直接學習C++。根據《C++編程思想》（Thinking in C++）一書所評述的，C++與C的效率往往相差在正負5%之間。所以有人認為在大多數場合中，C++完全可以取代C語言。

C++語言發展大概可以分為三個階段：第一階段從80年代到1995年。這一階段C++語言基本上是傳統類型上的物件導向語言，並且憑借著接近C語言的效率，在工業界使用的開發語言中佔據了相當大份額；第二階段從1995年到2000年，這一階段由於標準模板庫(STL)和後來的Boost等

程式庫的出現，泛型程式設計在C++中佔據了越來越多的比重性。當然，同時由於Java、C#等語言的出現和硬體價格的大規模下降，C++受到了一定的衝擊；第三階段從2000年至今，由於以Loki、MPL等程式庫為代表的產生式編程和模板元編程的出現，C++出現了發展歷史上又一個新的高峰，這些新技術的出現以及和原有技術的融合，使C++已經成為當今主流程式設計語言中最複雜的一員。

(三)JAVA[17]

Java，是一種可以撰寫跨平臺應用軟體的物件導向的程式語言，由昇陽（Sun Microsystems）公司的詹姆斯·高斯林（James Gosling）等人於1990年代初開發。Java 程式語言的風格十分接近C++語言。繼承了 C++語言物件導向技術的核心，Java 捨棄了 C++語言中容易引起錯誤的指標（以參照取代）、運算符重載（operator overloading）、多重繼承（以介面取代）等特性，增加了垃圾回收器功能用於回收不再被參照的對象所佔據的內存空間。在 Java SE 1.5 版本中 Java 又引入了泛型程式設計（Generic Programming）、類型安全的枚舉、不定長參數和自動裝/拆箱等語言特性。

Java 不同於一般的編譯執行電腦語言和解釋執行電腦語言。它首先將原始碼編譯成位元組碼(bytecode)，然後依賴各種不同平臺上的虛擬機來解釋執行位元組碼，從而實現了「一次編譯、到處執行」的跨平臺特性。不過，這同時也在一定程度上降低了 Java 程序的運行效率。但在J2SE1.4.2發佈後，Java的執行速度有了大幅提升。

與傳統程序不同,Sun 公司在推出 Java 之際就將其作為一種開放的技術。全球數以萬計的 Java 開發公司被要求所設計的 Java 軟體必須相互兼容。「Java 語言靠群體的力量而非公司的力量」是 Sun 公司的口號之一，並獲得了廣大軟體開發商的認同。這與微軟公司所倡導的注重精英和封閉式的模式完全不同。

Sun 公司對 Java 程式語言的解釋是：Java 程式語言是個簡單、物件導向、分散式、解釋性、健壯、安全與系統無關、可移植、高性能、多執行緒和動態的語言。

Java 平台是基於 Java 語言的平台。這樣的平台目前非常流行，因此微軟公司推出了與之競爭的.NET 平台以及模仿 Java 的 C#語言。

(四)C#[11]

續第二章第四節的部分介紹，C#到目前為止大約發展了六年左右，從一路 C#走到了 Visual C#2003，在 2005 年時跟著 Visual Studio 2005 (也就是 NET Framework 2.0)一起出的 Visual C#2005，C#語言的發展也漸趨成熟，只要.NET 的理念繼續茁壯，C#成為程式語言中的霸主亦不遠矣。

以下列出 Visual C#2005 與 Java 語言之重要特色比較，如表 3-2-1

表 2-2 Visual C#2005 與 Java 語言之重要特色比較

功能	Visual C#2005	Java
例外處裡(Exception Handing)	有	有
多重繼承(Multiple Inheritance)	可用 interface 實現	可用 interface 實現
平臺獨立性(Platform Independence)	有	有
支援的數值類別	較多	較少
Switch 語法	有	僅支援數值條件判斷
垃圾記憶體回收機制(Garbage Collection)	有	無
指標的使用(Pointer)	有支援但不建議使用	無
運算子多載(Operator Overloading)	有	無
委派(Delegate)	有	無
參考型別(Reference)	有	無
結構語法(Structure)	有	無
類別擁有屬性(Property)	有	無
Goto 語法	有，但有限制。	無
As 運算符號	有	無

(五)ASP 和 ASP.NET[10] [18]

ASP 是早期由 Microsoft 大力推行的網頁程式設計技術，而 ASP.NET 則是 ASP 的改版，與其說改版倒不如說 ASP.NET 是一個全新的產品，它把所有 Microsoft .NET 的應用程式，建構在 CLR 的基礎上而成。讓程式設計師可以使用它所支援的程式語言，如 VB、C#、C++、JAVA、等來撰寫 ASP.NET 程式，當然也包括 Open Source 領域的語言，像是 Perl、Python。

若拿 ASP.NET 與先前的 Scripting 技術比較，ASP.NET 速度快的原因在於，ASP.NET 平台會先把整個網站先編譯成一個（或數個）dll 檔案，然後讓網站伺服器執行，就是連上一個 WEP 伺服編譯過的網頁。

在視窗應用程式過渡到網站應用程式之間的開發工作中，ASP.NET 亦試圖讓開發人員利用一系列的控制項，來建立類似圖形用戶界面的操作環境。換句話說，ASP.NET 的開發環境，有兩個特點：擁有和視窗環境非常相似的 Web 控制項：像是 Button、Label 等等。這些控制項都有各自的事件，除此之外，也可以利用程式碼來設定這些控制項的屬性。ASP.NET 平台會自行處理這些控制項的所有細節：就好比我們丟一個控制項在視窗應用程式，然後在畫面上顯示的動作類似。差別在於，在 Web 的環境中，ASP.NET 平台會先處理控制項在畫面（Web Form）上產生的 HTML 標籤，然後再把處理的結果送到使用者的瀏覽器中。

與傳統的開發方式（Scripting Programming）相較，ASP.NET 也鼓勵程式設計師採用 事件驅動（Event-Driven Programming_ 或 使用者圖型介面（GUI)的方式進行開發工作，.NET 平台亦嘗試將內建元件（如 ViewState）與現有的網頁技術（如 Javascript）結合。.NET 平台是 ASP.NET 的基礎核心架構。在這個核心架構中，包括有 Runtime Environment（類似 Java 平台）、[[Virtual Machine|VM] 以及 JIT、Class Library。

在 ASP.NET 的環境裡，開發人員可以在撰寫程式碼時，把許多控制項、類別或工具直接剪下，然後貼在其它類似性質的開發作業中。資料

存取（Data Access）就是一個例子（把資料庫中的記錄顯示在畫面上）。

(六)JSP[19]

JSP 全名為 Java Server Pages，是由 Sun Microsystems 公司倡導和許多公司共同參與建立的一種網頁程式設計技術,目前最新版本為 2.0 版，是以 Java 語言作為腳本語言，開發者可以反應客戶端請求，而動態生成 HTML、XML 或其他格式的 Web 網頁的技術標準，程式是在 WEP 伺服器上執行，而不是在客戶端瀏覽程式。

Java 的 WEP 應用程式架構是一種結合 Servlet、JSP 技術和 JavaBeans 元件的三層架構如圖 2-7 所示：

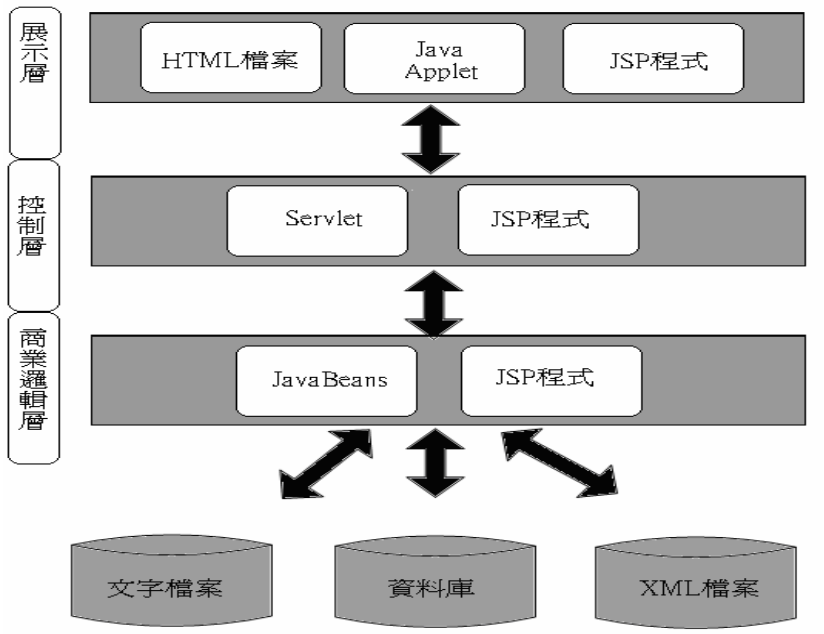

圖 2-7 Servlet、JSP 技術和 JavaBeans 元件的三層架構

1.展示層：是與使用者互動的使用者介面，這是在客戶端瀏覽程式實際所看到的顯示結果或表單介面。

2.控制層：主要用於連結展示層和商業邏輯層，及作為控制處理流程 的控制者，負責接收使用者的地 http 請求。

3.商業邏輯層：主要負責 WEP 應用程式的資料存取和處理。

JSP 頁面文件通常以.jsp 為擴展名，而且可以安裝到任何能夠存放普通 Web 頁面的地方。

看完上述程式大概的介紹之後，開始要來想我們開發的程式是需要怎樣取向的程式，我們的系統是一種在網路上應用的程式，在此開發系統上選擇 C# 最主要的原因是他是可以完全相容於 windows 作業系統的，在.NET 的架構下 C# 可以輕易的與其他語言做溝通，可以跨平台如 PDA 或是手機，正因為.NET 支援 C#的開發環境，也可以更輕易的與 ASP.NET 搭配開發程式，而這兩個我們所選的程式優勢如下:

(一) C#部分[11][12][13][14]

1.在 Visual Studio 2005 中 C#有著強大的完善的開發環境，有著功能完整與強大的控制項，編譯器智能感知功能，插入程式碼功能，程式碼重整功能，程式變更追蹤功能，完整的使用者文件，一次點選部署功能。

2.C#語言功能強化，提供了最新的語法功能，泛型，疊代器，匿名方法，部分類別，跨語言溝通能力，百分之百完全相容於 Microsoft

Windows 作業系統平台。

3. C#提供了資料庫連線的語法與管理，搭配 ASP.NET 可開發出網路服務的應用程式。

(二)ASP.NET 部分[18][20][21]][22]

1. ASP.NET 支援.net 開發環境，.net 開發環境能縮短網路建構應用程式上開發時間，而且是跨語言能用多種不同的語法編寫。
2. ASP.NET 為 Microsoft 大力推行，因此開發環境使用於 Microsoft Visual Studio 2005 這套強大個開發程式，在裡面提供了很多方便設計工具，使得開發過程中節省不少時間。
3. ASP.NET 學習較 JSP 學來更容易，而且功能較 PHP 來得強大，更重要的是使用的人最多，因此能參考的範例多。
4. VB.NET 也支援 ASP.NET 這點很重要，因為 VB 簡單易學的特性，使用者多如果有開發上的問題也較易解決。

以下是 Microsoft Visual Studio 2005 的介紹。

Microsoft Visual Studio 2005 [10]系列是微軟開發工具重要的里程碑！不僅有提升 200% 效能的 .NET Framework 2.0 應用程式平台、節省 50% 開發時程的 ASP.NET 2.0 技術，也進入了全新的 64 位元程式開發領域、在行動裝置上輕鬆實現創意！重要還有開發團隊的全新競爭力：Visual Studio Team System。這是微軟首度推出與開發工具完全整合的軟體開發生命週期管理平台，結合 Agile 方法論以及 CMMI 標

準，除了使軟體開發流程更加嚴謹之外，軟體開發的品質與時間也更能掌控，Visual Studio 具備易學易用、高生產力的優點，能確實提提升流程導向的開發團隊的效率。

而在 Microsoft Visual Studio 2005 我們使用的開發環境是使用 Visual C# 2005[11][12][13][14]，C#（發音為 C Sharp）是由微軟公司所開發的一種物件導向的、運行於.NET Framework 之上的高級程式語言。並且成為 ECMA 與 ISO 標準規範。C#在外觀上是基於 C++寫成，又融入其它語言如 Delphi, Java, VB 等。我們會使用 C# 最主要的原因是他是可以完全相容於 windows 作業系統的，在.NET 的架構下 C# 可以輕易的與其他語言做溝通，可以跨平台如 PDA 或是手機，用途廣泛所以我們才利用 C#來完成此次專題。

以下為安裝 Microsoft Visual Studio 2005 流程

1.首先放入光碟，跳出的畫面。(如圖 2-8)

圖 2-8 Visual Studio 2005 安裝畫面-1

2.按下安裝 Visual Studio 2005，之後出現的畫面，載入安裝元件。(如圖 2-9)

圖 2-9 Visual Studio 2005 安裝畫面-2

3.載入完成後，按下下一步。(如圖 2-10)

圖 2-10 Visual Studio 2005 安裝畫面-3

4.接受授權，輸入序號，輸入名稱後，按下下一步。(如圖 2-11)

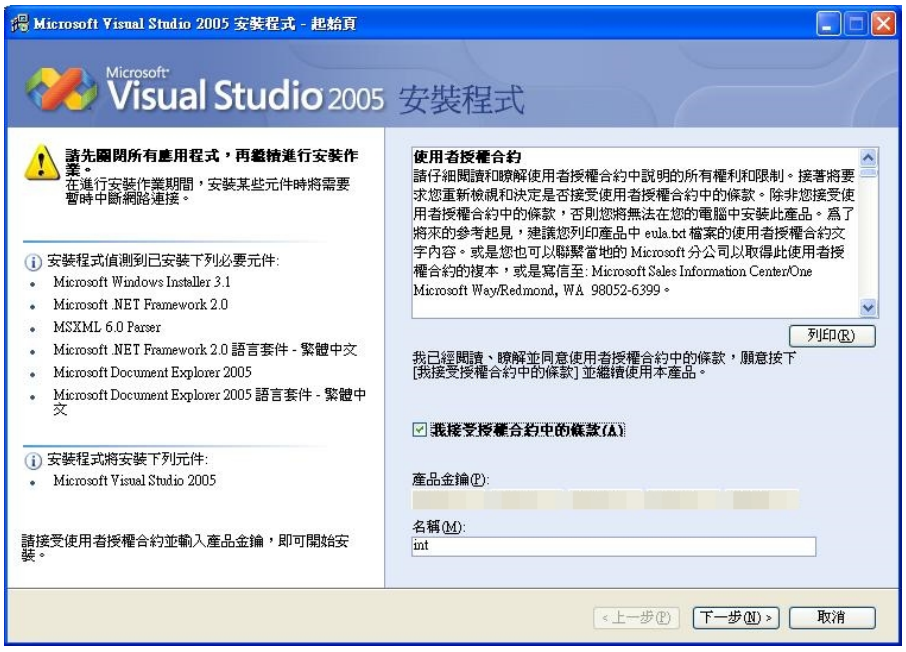

圖 2-11 Visual Studio 2005 安裝畫面-4

5.安裝功能類型，選擇安裝路徑，按下下一步。(如圖 2-12)

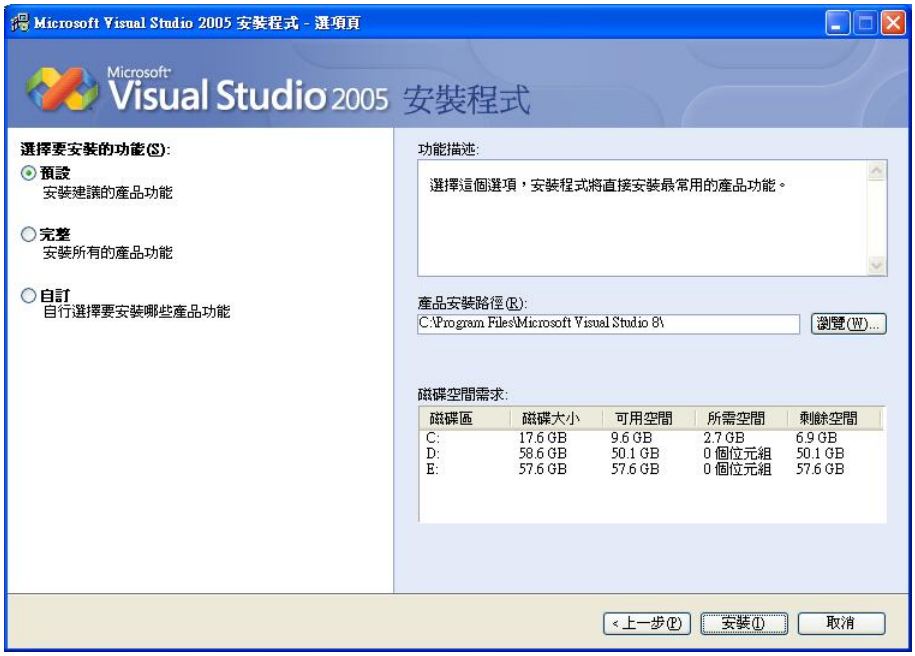

圖 2-12 Visual Studio 2005 安裝畫面-5

6. 開始正式安裝。(如圖 2-13)

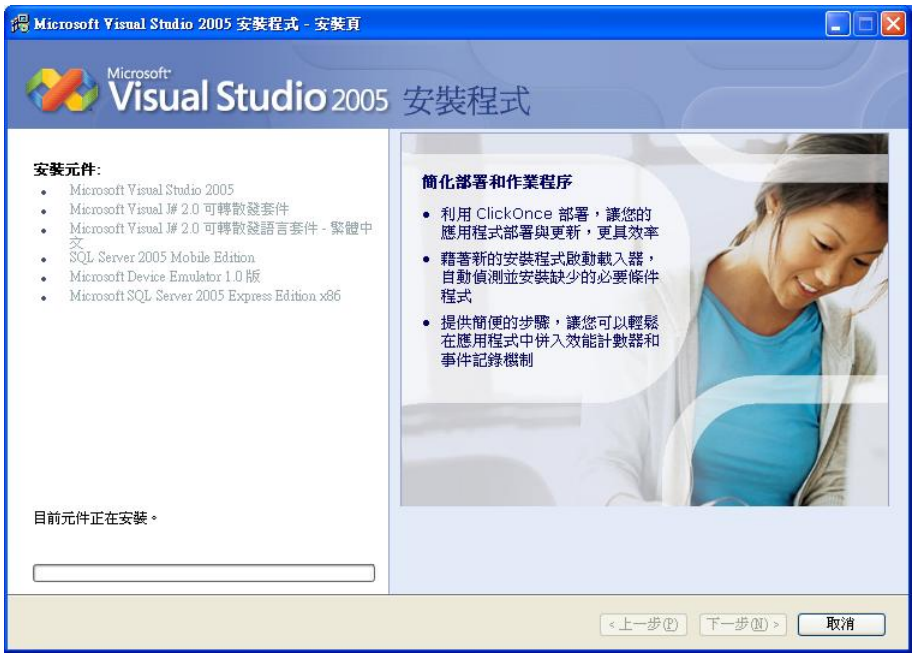

圖 2-13 Visual Studio 2005 安裝畫面-6

7. 安裝完成。(如圖 2-14)

圖 2-14 Visual Studio 2005 安裝畫面-7

8.程式圖示，點兩下執行。(如圖 2-15)

圖 2-15 Visual Studio 2005 執行畫面-1

9.預設環境設定，這裡我們選擇 C#開發設定，然後啟動。(如圖 2-16)

圖 2-16 Visual Studio 2005 執行畫面-2

10. 出現 Visual Studio 2005 的起始頁面。(如圖 2-17)

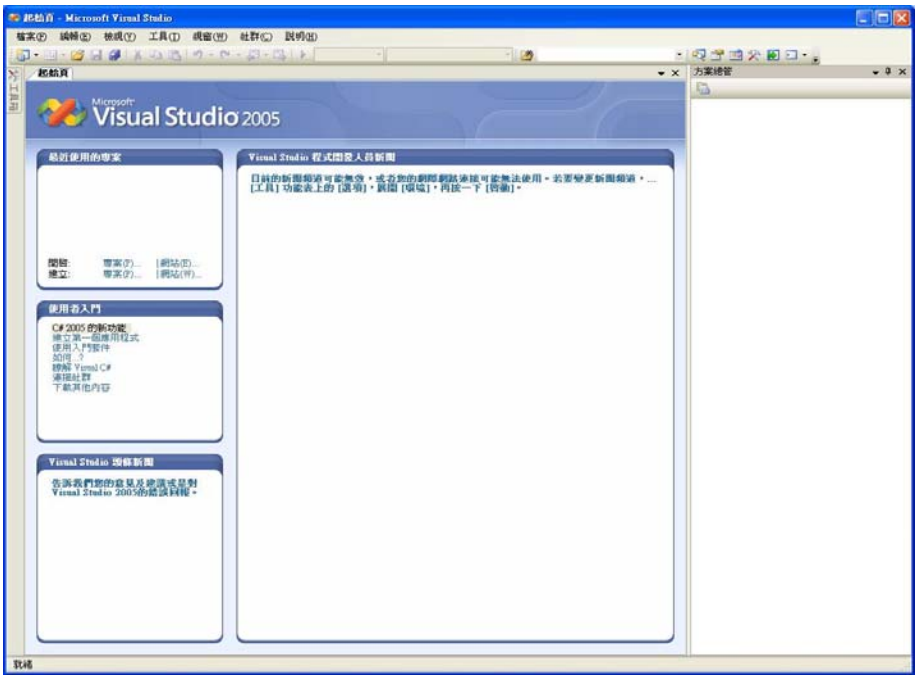

圖 2-17 Visual Studio 2005 執行畫面-3

11. 點選檔案→新增專案。(如圖 2-18)

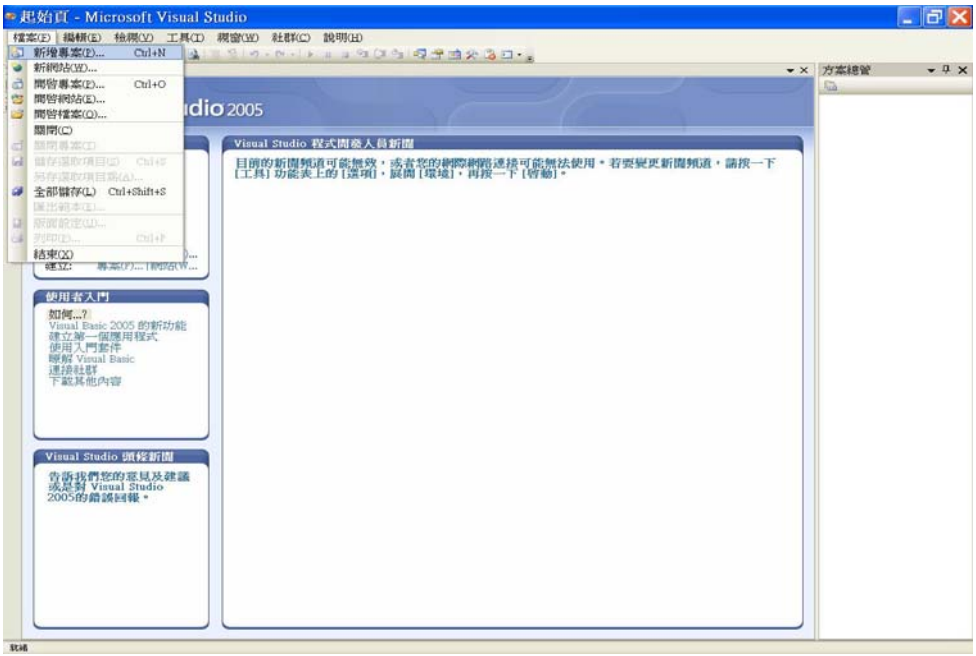

圖 2-18 Visual Studio 2005 執行畫面-4

12.選擇專案類型。(如圖 2-19)

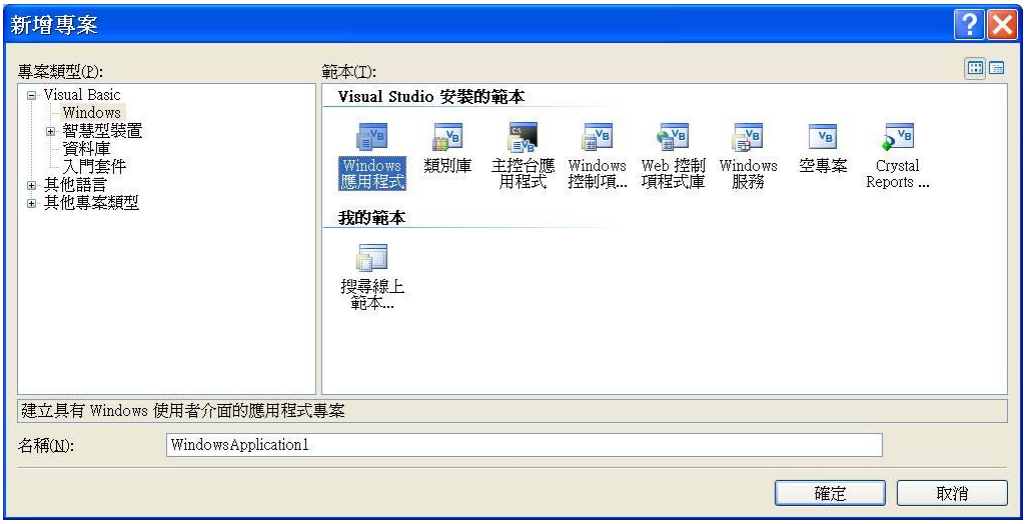

圖 2-19 Visual Studio 2005 執行畫面-5

13.執行畫面。(如圖 2-20)

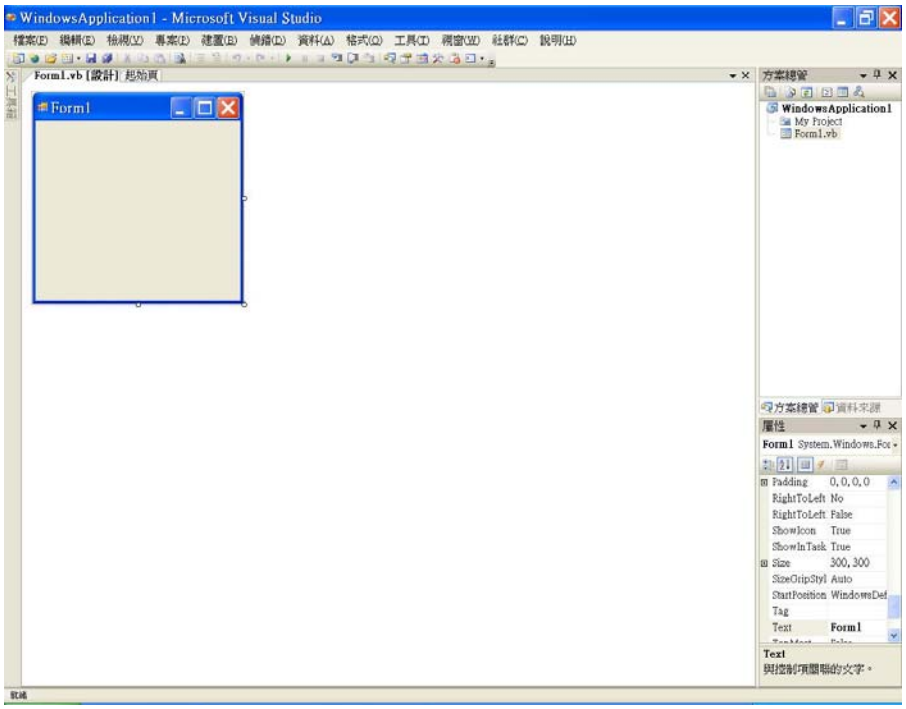

圖 2-20 Visual Studio 2005 執行畫面-6

14.利用工具箱設計程式外觀。(如圖 2-21)

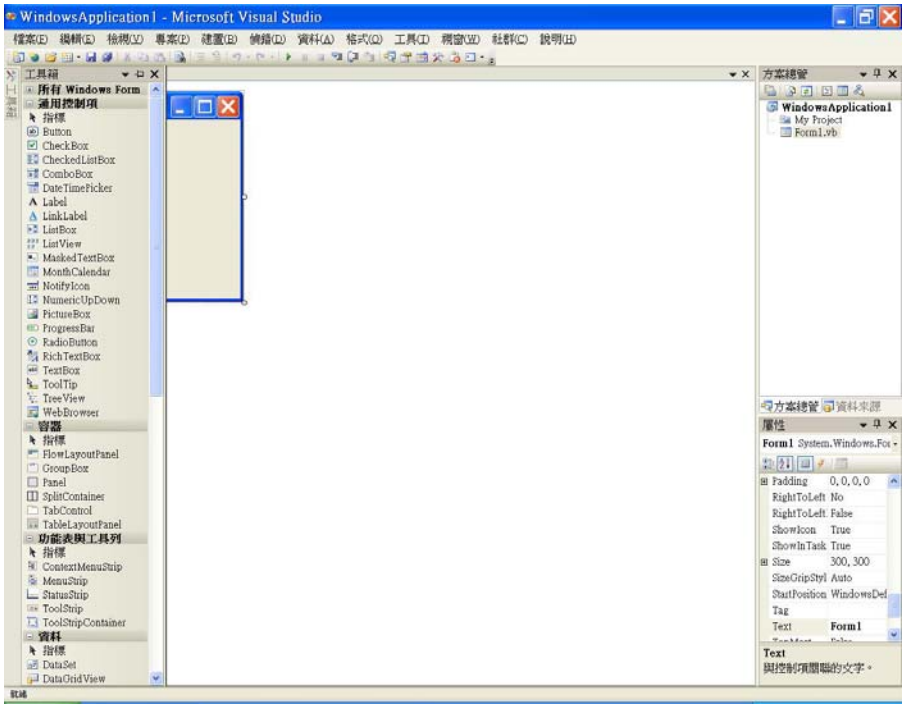

圖 2-21 Visual Studio 2005 執行畫面-7

第三章　具 SHA-256 功能之資料確認平台

第一節　系統架構與開發流程

對於整個系統,最初構想是以程式設計一個類似網頁介面的檔案交換平台，再進行檔案安全加解密驗證的動作，而且能在不同架構有網路的地方執行，然而本系統如果只是個檔案交換平台這樣一個單純的介面，其實用性及功能顯的十分薄弱，於是我們增加了一些功能使得這整個系統更加完整，如圖 3-1-1 系統架構圖所示，現在我們系統目前主要的架構有系統公告、留言板、上下傳檔案&驗證碼、驗證檔案、檔案列表、聊天室…等功能。

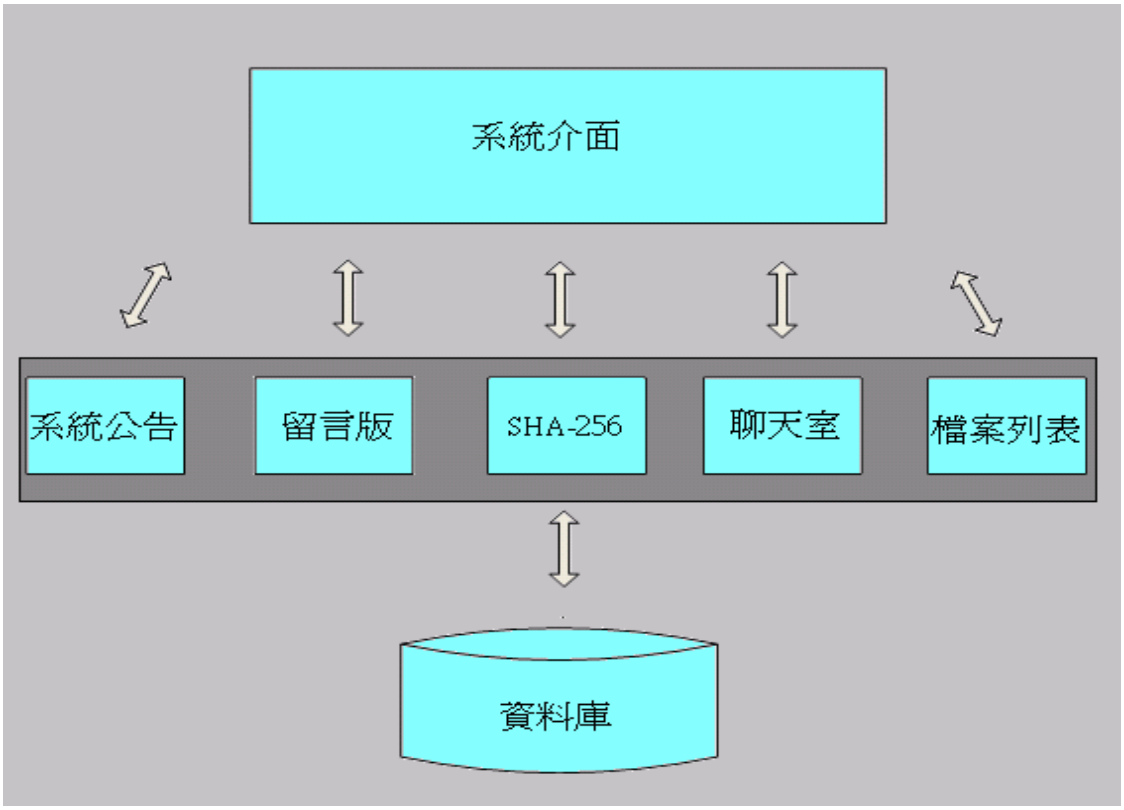

圖 3-1 系統架構圖

想做好任何事之前，事前的規劃是很重要，如圖 3-1-1 系統製作流程圖，我們將整個系統分成 5 大步驟來逐一完成，並在每個步驟仔細思考是否還有地方需要加強最後在逐一實行。

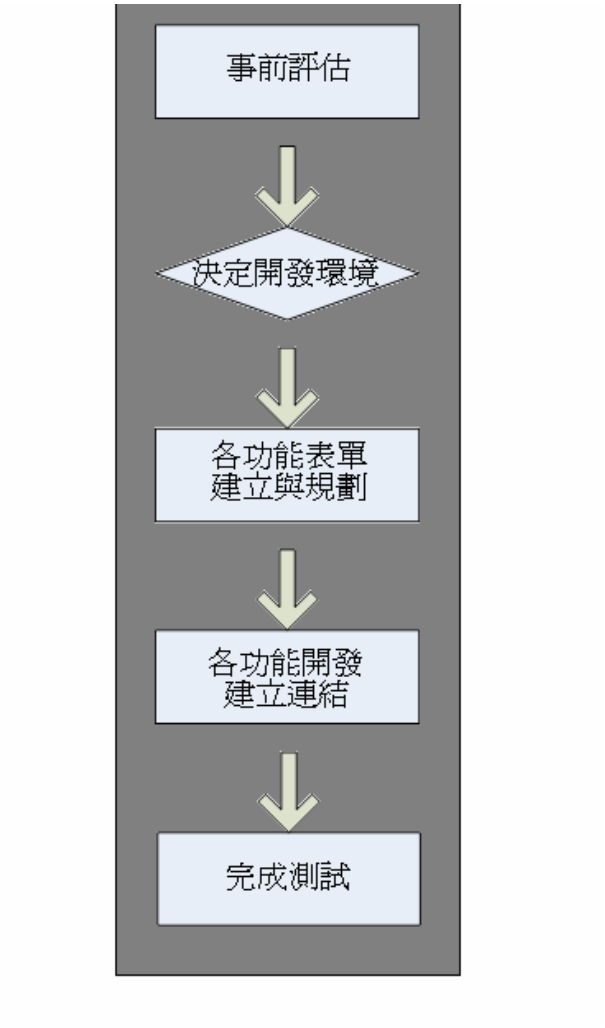

圖 3-2 系統製作流程圖

第二節　系統功能介紹

這是一個能提供使用者提升網路檔案資料安全性的程式平台，此平台具有 SHA-256 驗證功能、系統公告、上/下載檔案、檔案列表、留言板、聊天室等功能。

要使用 SHA-256 檔案驗證的方法，只要點選按下介面裡的「SHA-256」鍵，然後找出想要演算出雜湊值的檔案以進行編碼，這樣就可以獲得一組此檔案專屬的雜湊值。獲得雜湊值之後再把此檔案與他的雜湊值一起上傳，這樣就可以提供給別人下載使用。如果對上傳的檔案有任何疑問，也可以馬上利用留言功能告知給其他使用者。

雖然網路上有許多使用雜湊演算法的程式，可是它們並不能做到與原始檔案的使用者互動溝通，只能單純的做演算法算出雜湊值，如此可能會不知道真正原始檔案的正確性。而在我們的資料驗證平台中，提供了可以對話留言的空間，是一個可以進行互動的程式平台。

程式點開首先會看到我們的起始頁面，沒有按下「登入」之前功能按鍵是被鎖定的。(如圖 3-3)

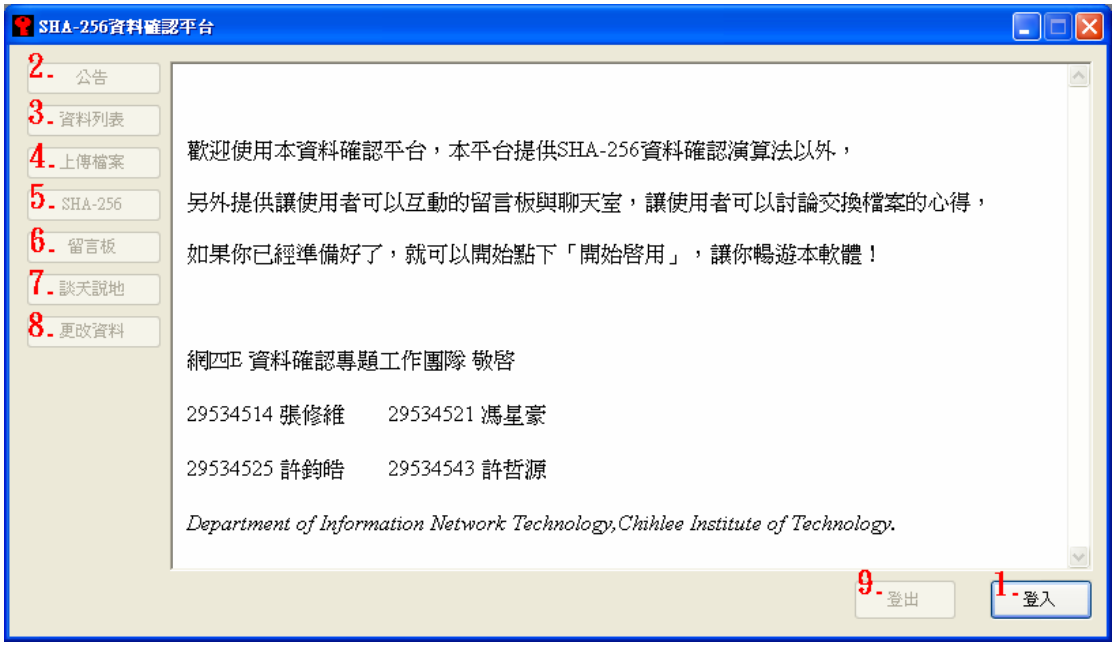

圖 3-3 平台功能圖起始頁

本資料確認平台的功能按鍵有：

1. 登入 （其內容請參閱（一）「**登入**」）
2. 公告 （其內容請參閱（二）「**公告**」）
3. 資料列表 （其內容請參閱（三）「**資料列表**」）
4. 上傳檔案 （其內容請參閱（四）「**上傳檔案**」與「SHA-256」）
5. SHA-256 （其內容請參閱（四）「**上傳檔案**」與「SHA-256」）
6. 留言板 （其內容請參閱（五）「**留言板**」）
7. 談天說地 （其內容請參閱（六）「**談天說地**」）
8. 更改資料 （其內容請參閱（七）「**更改資料**」）
9. 登出 （其內容請參閱（八）「**登出**」）

第三節　系統效能與畫面

下列為本具 SHA-256 功能之資料確認平台所擁有的功能展示。

（一）「登入」

1. 點選「登入」之後跳出登入的頁面。(如圖 3-4)

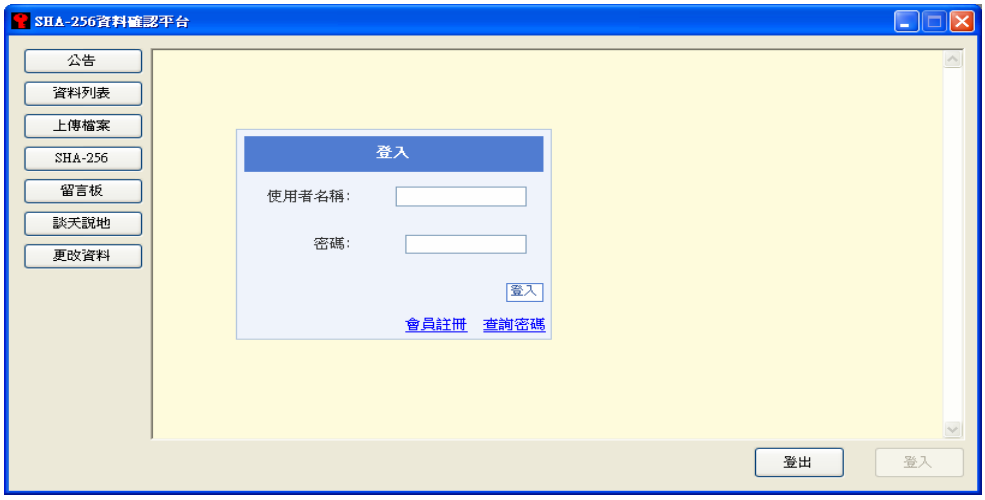

圖 3-4 平台功能圖登入頁

2. 點選會員註冊點選之後就會跳出註冊畫面。(如圖 3-5)

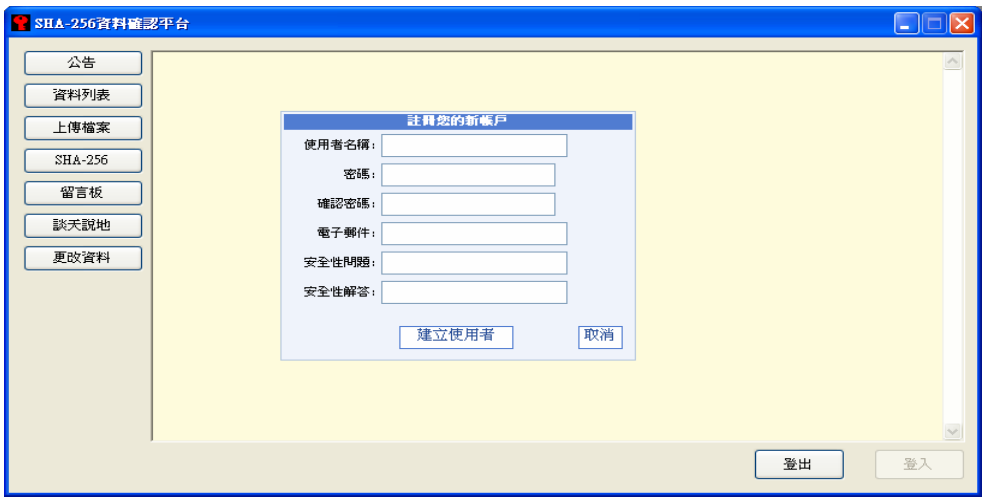

圖 3-5 平台功能圖註冊頁

3. 擁有帳號後開始登入。(如圖 3-6)

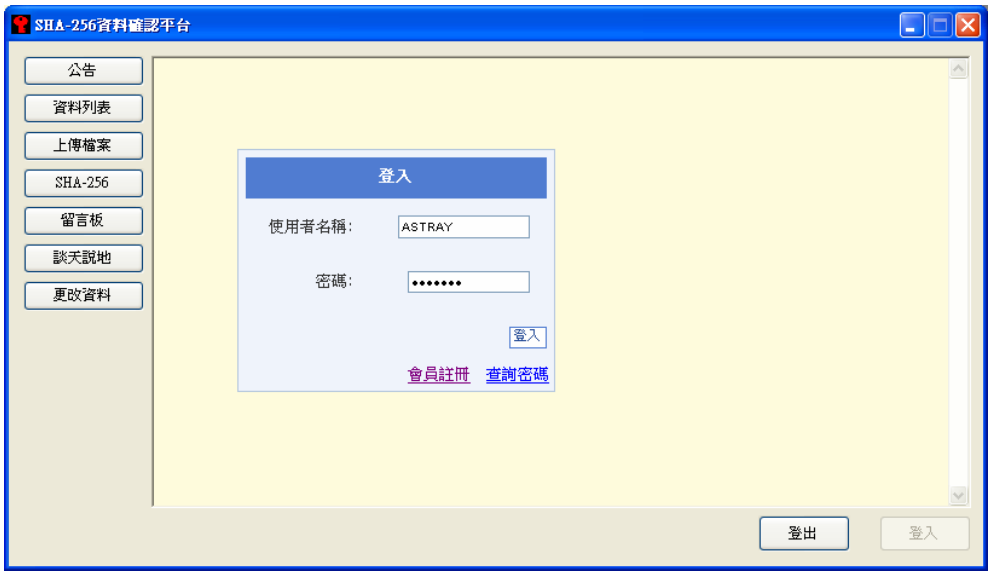

圖 3-6 平台功能圖登入頁

4. 如果忘了自己的密碼的話可以按下查詢密碼，送出之後會把相關資料寄到當初所註冊的信箱裡。(如圖 3-7)

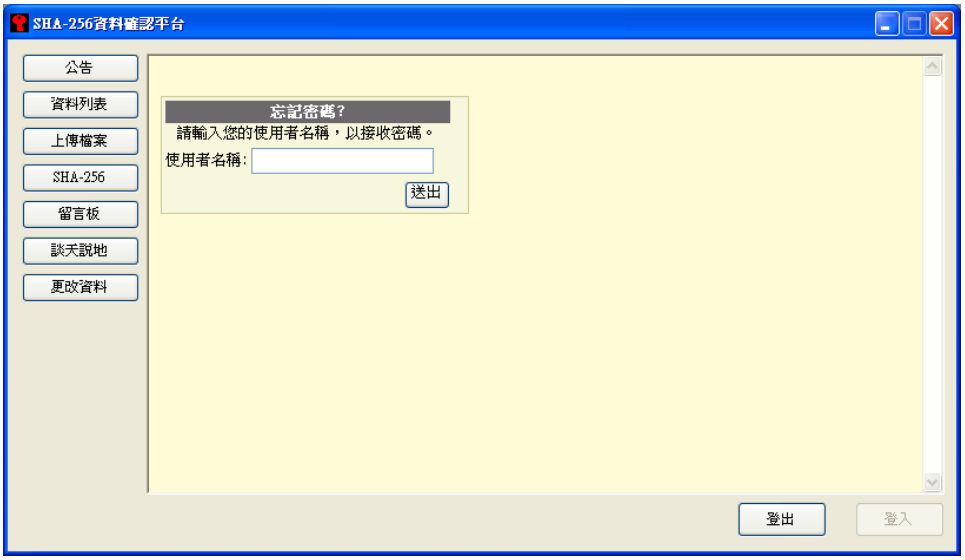

圖 3-7 平台功能圖查詢密碼

（二）「公告」

登入之後系統就會跳到公告頁，或者是點選「公告」鍵，有什麼最新消息都會在此公佈出來。(如圖 3-8)

圖 3-8 平台功能圖公告頁

（三）「資料列表」

點選「資料列表」鍵，會跳出上傳後的資料列表。(如圖 3-9)

圖 3-9 平台功能圖資料列表頁

（四）「上傳檔案」與「SHA-256」

1. 點選「上傳檔案」鍵開始我們準備要來上傳檔案了。(如圖 3-10)

圖 3-10 平台功能圖上傳檔案頁

2. 接著點選「SHA-256」鍵就會跳出編 SHA-256 碼的視窗。(如圖 3-11)

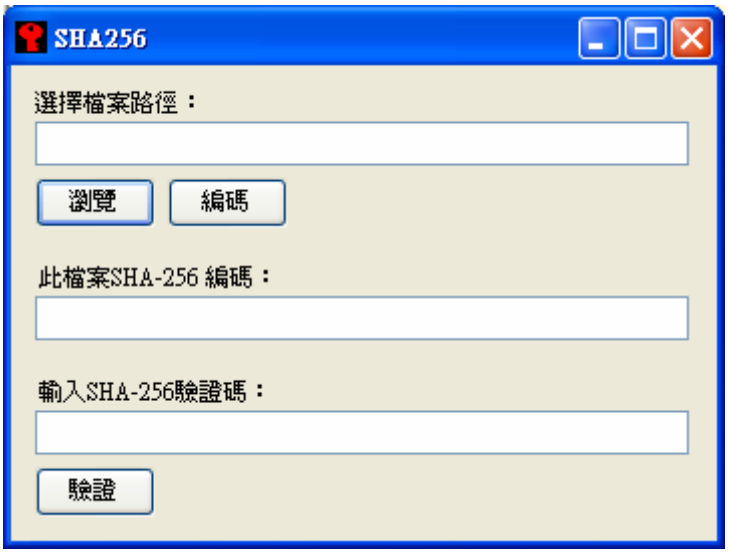

圖 3-11 平台功能圖 SHA-256 頁

3. 按下「瀏覽」鍵選取檔案做 SHA-256 資料確認演算。(如圖 3-12)

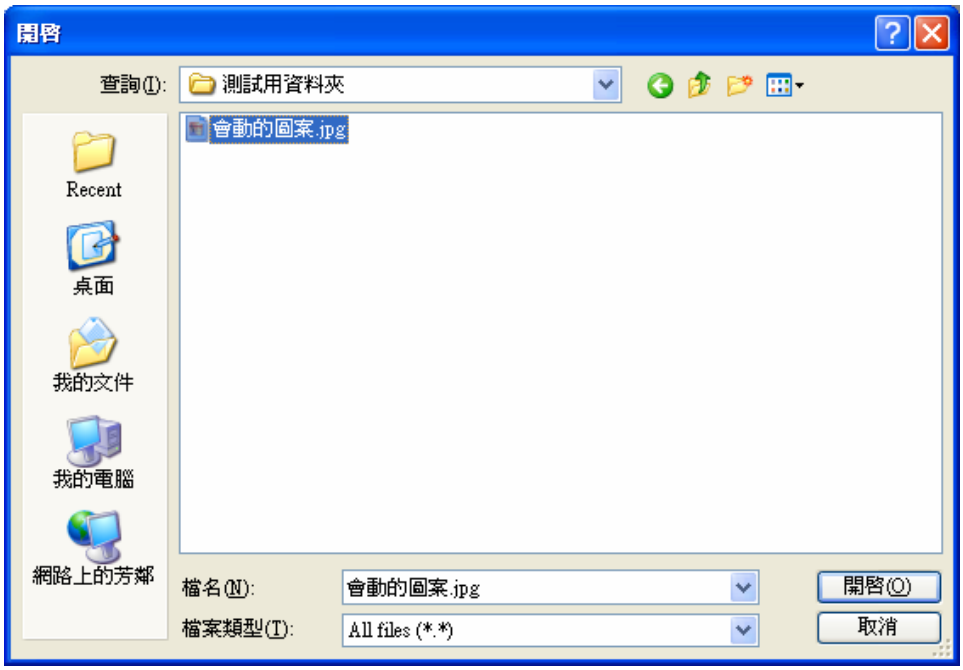

圖 3-12 平台功能圖 SHA-256 頁-瀏覽檔案

4. 選取之後按下「編碼」鍵，產生出 SHA-256 的編碼。(如圖 3-13)

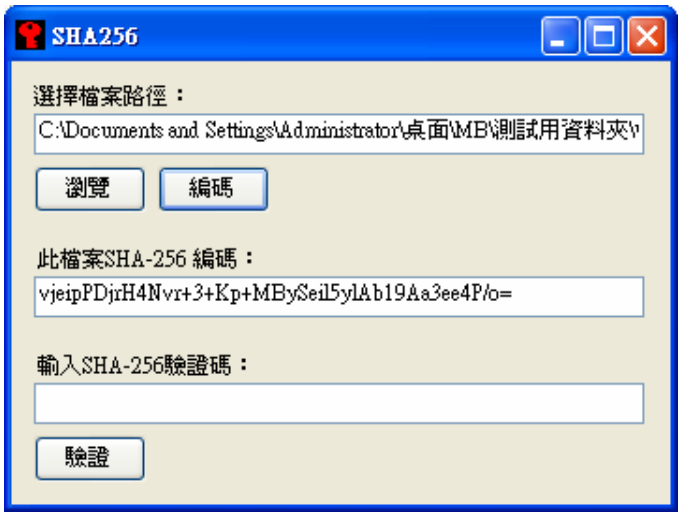

圖 3-13 平台功能圖 SHA-256 頁-檔案編碼

5. 獲得 SHA-256 編碼後把此編碼貼在 SHA-256 的格子裡，然後打好主題內容，再次瀏覽選擇想上傳之檔案。(如圖 3-14)

圖 3-14 平台功能圖上傳檔案頁-輸入內容

6. 然後檔案必須先按下「上傳」鍵，把檔案上傳，上傳完畢之後會跳出檔案名稱、檔案大小、檔案類型。(如圖 3-15)

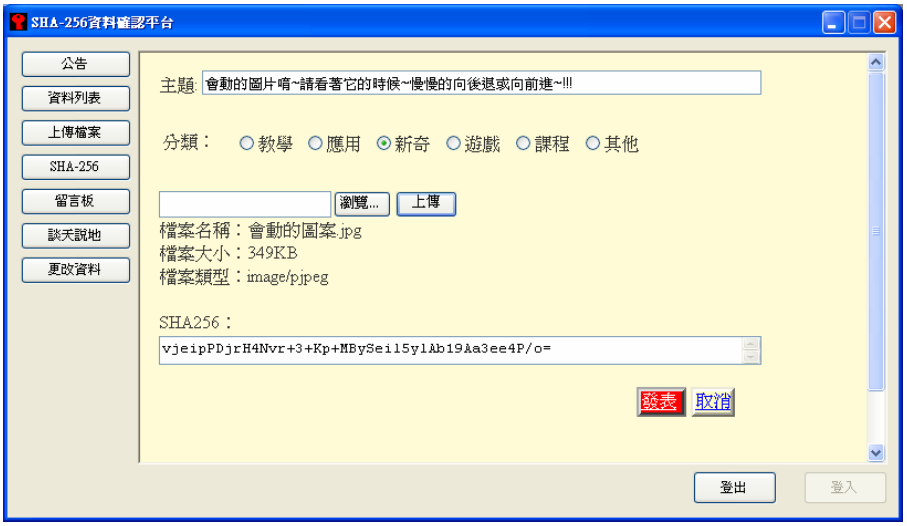

圖 3-15 平台功能圖上傳檔案頁-成功上傳

7. 上傳成功之後再按下「發表」，成功的話就會跳到資料列表頁。(如圖 3-16)

圖 3-16 平台功能圖資料列表頁-發表成功

8. 點選剛剛的發表標題，進去之後就會出現以下表格，裡面會出現發文者的帳號和檔案下載和 SHA 的編碼。(如圖 3-17)

圖 3-17 平台功能圖資料列表頁-檔案主題

9. 點選檔案下載會直接跳出連結頁，或者是點選右鍵另存目標就可以下載檔案，直接點選就會直接連結到檔案的位置。(如圖 3-18)

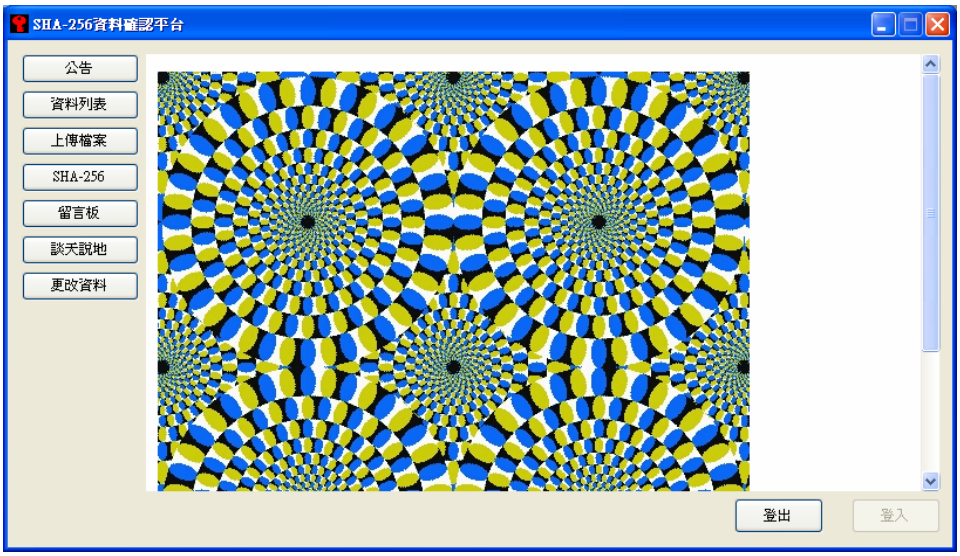

圖 3-18 平台功能圖資料列表頁-下載檔案

10. 除了下載檔案還有個「回應」的按鍵，就是可以回應這個上傳檔案發文的人事情，資料有問題的話也可以在此告知其他人注意小心。(如圖 3-19)

圖 3-19 平台功能圖資料列表頁-回應主題

11. 回應之後就會出現在主題下方的表格裡。(如圖 3-20)

圖 3-21 平台功能圖資料列表頁-回應成功

12. 再來要怎麼知道檔案被竄改過了呢？首先瀏覽找出下載的檔案之後，在按下編碼就會產生出 SHA-256 的編碼，此時再把主題上的 SHA 編碼複製，貼在輸入 SHA-256 驗證碼的格子裡。(如圖 3-21)

圖 3-21 平台功能圖 SHA-256 頁-驗證編碼

13. 按下驗證之後如果資料確認無誤就會出現。(如圖 3-22)

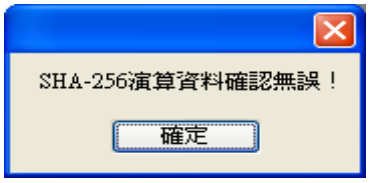

圖 3-22 平台功能圖 SHA-256 頁-資料無誤

14. 如果檔案有問題被竄改過就會出現。(如圖 3-23)

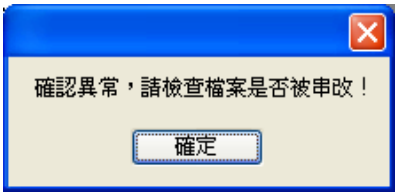

圖 3-23 平台功能圖 SHA-256 頁-資料有誤

(五)「留言板」

1. 如果有什麼請求 or 疑問 or 建議的話，就可以點下「留言板」鍵就可以把問題留言下來了。(如圖 3-24)

圖 3-24 平台功能圖留言板頁

2. 點下填寫留言之後就可以開始留言，這個地方可以使用匿名的方式留言，不會顯示出自己的帳號出來。(如圖 3-25)

圖 3-25 平台功能圖留言板頁-填寫留言

3. 送出留言之後，就會新增在最上方了。(如圖 3-26)

圖 3-26 平台功能圖留言板頁-留言成功

4. 而這個留言板管理修改的人只有這個留言版的管理人，這個就需要另外的帳號密碼登入了，點選管理登入後。(如圖 3-27)

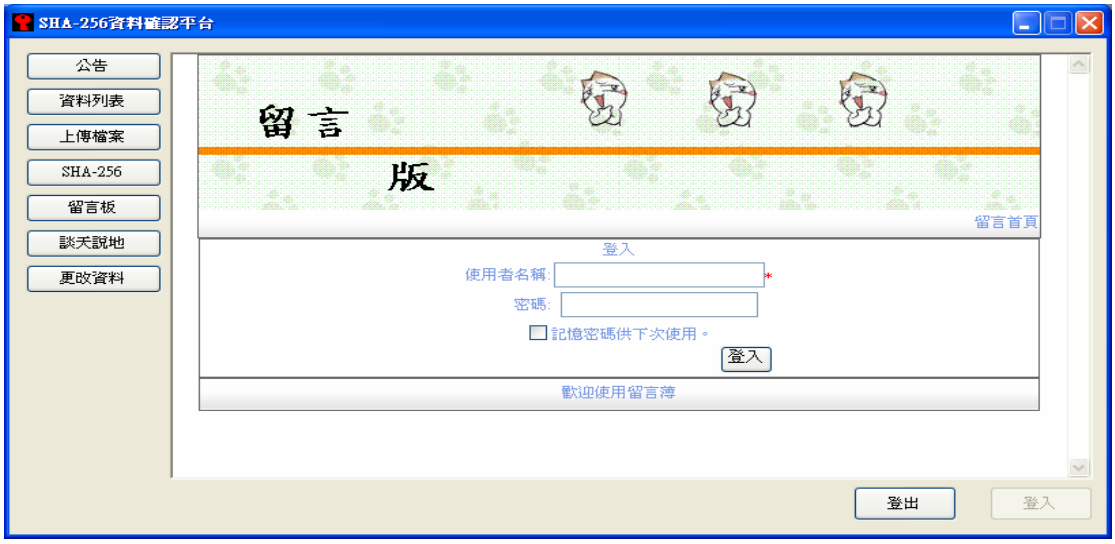

圖 3-27 平台功能圖留言板頁-管理登入

5. 管理者登入，就可以回應問題或者是刪除文章。(如圖 3-28)

圖 3-28 平台功能圖留言板頁-管理畫面

6. 管理者回應之後，相關內容就會出現在那篇文章的下方回應處。(如圖 3-29)

圖 3-29 平台功能圖留言板頁-管理者回應

(六)「談天說地」

1. 點選「談天說地」鍵之後跳出個聊天室的畫面。(如圖 3-30)

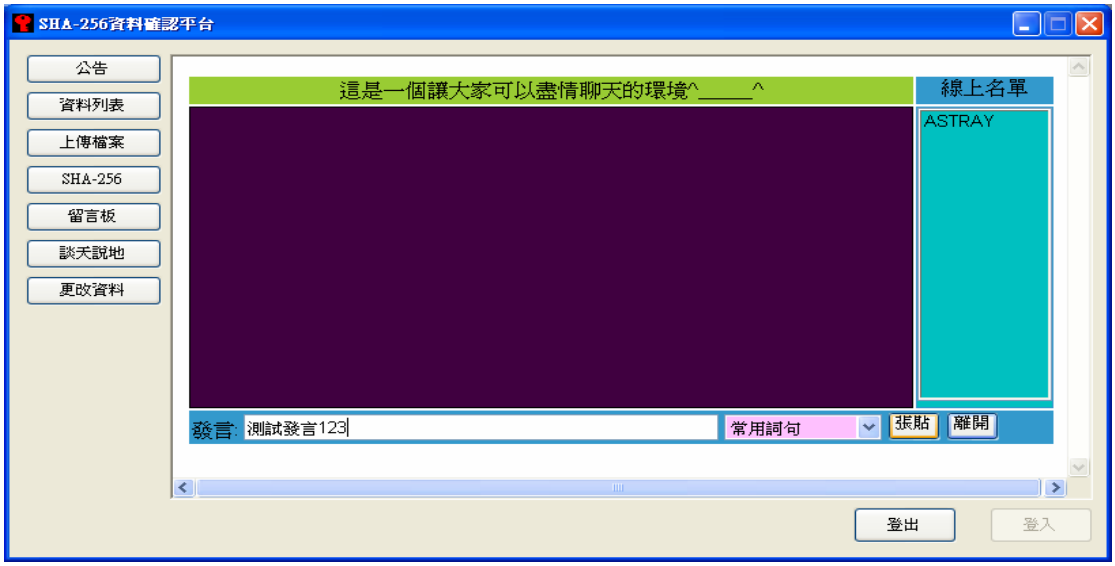

圖 3-30 平台功能圖談天說地頁

2. 此時如果有人一起在線上就可以進行像是聊天室一樣的即時談話的效果，問問題也會比較快獲得回應，要發言的話必須在發言欄裡打出想說的話然後按下「張貼」鍵這樣文字就會被送出了。(如圖 3-31)

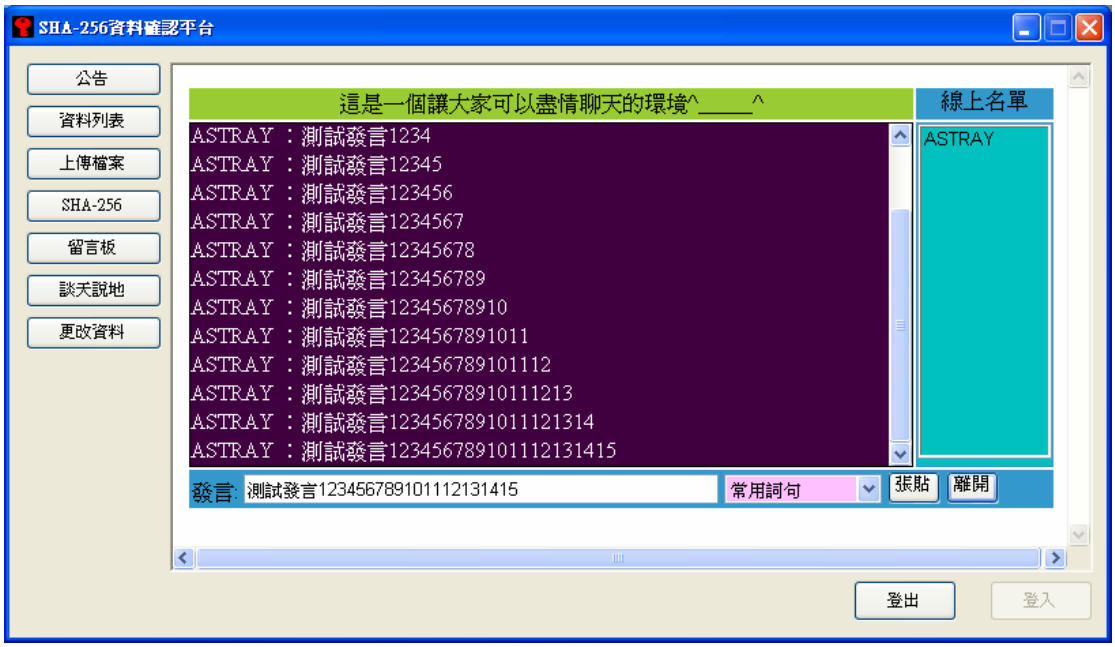

圖 3-31 平台功能圖談天說地頁-線上發言

(七)「更改資料」

1. 基本資料需要作更變得時候，點選「更改資料」鍵，就可以作修改了。(如圖 3-32)

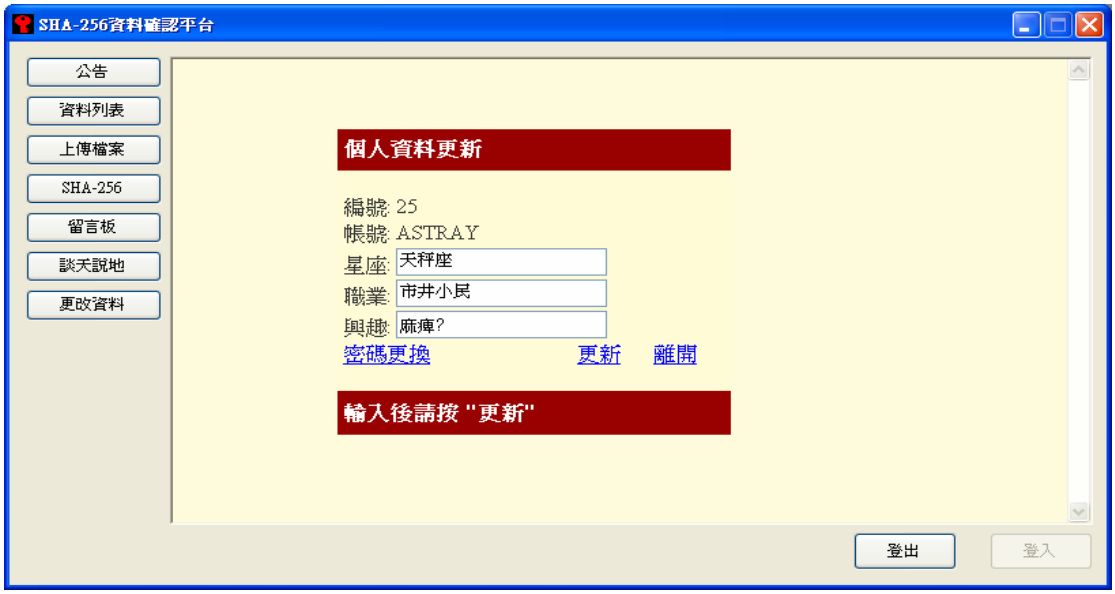

圖 3-32 平台功能圖更改資料頁

2. 在剛剛忘記密碼的部份(如圖 3-7)，把收到的密碼內容在這邊填上並重新修改自己的新密碼，然後按下變更密碼後就大功告成了。(如圖 3-33

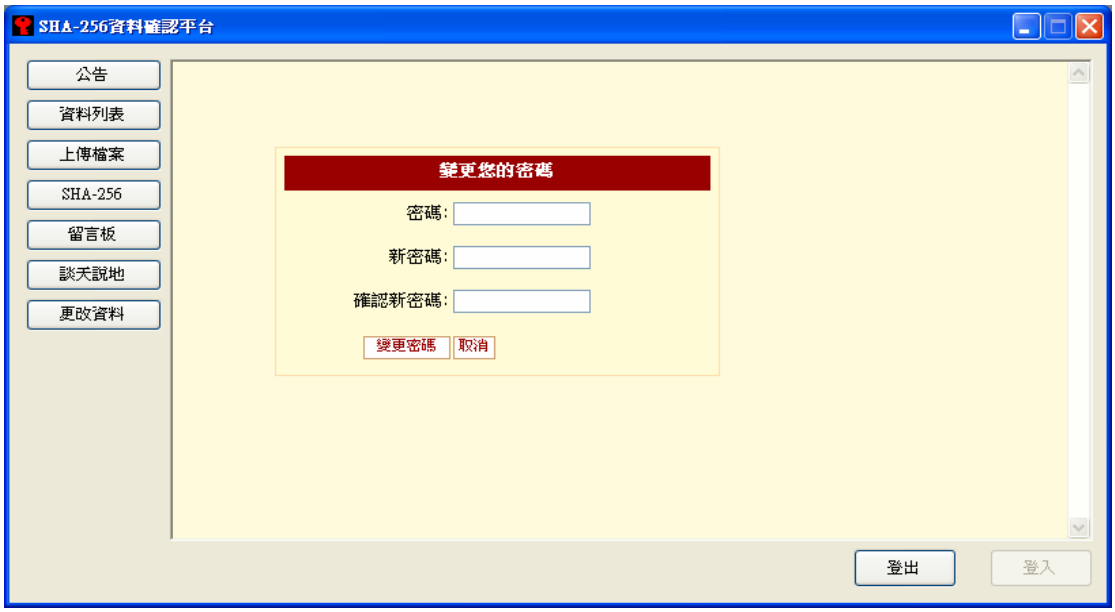

圖 3-33 平台功能圖更改資料頁-更改密碼

(八)「登出」

1. 按下「登出」鍵之後全部功能鍵就會被鎖起來，也就等於回到了本程式的起始頁了。(如圖 3-34)

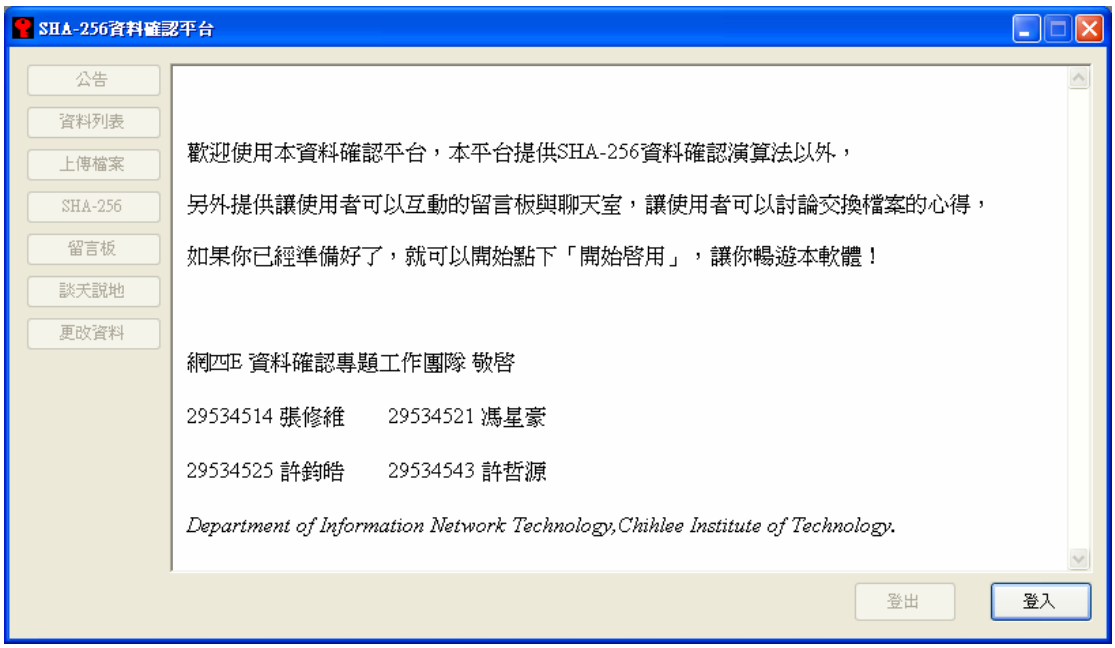

圖 3-34 平台功能圖登出頁

以上是目前本軟體的所有的功能圖解介紹。未來還有可能會做改版或者是更新追加一些功能。

第四章 系統比較

第一節 雜湊演算法與自創軟體比較

再來是與我們所製作的軟體相近的程式介紹，下面接著介紹的是「MD5summer」與「Karen's Hasher」這兩個有運用到雜湊演算法的程式。

(一)MD5summer[23]

1.點選 MD5Summer.exe 啟動程式。(如圖 4-1)

圖 4-1 MD5summer 執行畫面-1

2.點選軟體的存放資料夾，按下【Create sums】。(如圖 4-2)

(除了 MD-5 的演算法，也可以選擇 SHA-1 的演算法)

圖 4-2 MD5summer 執行畫面-2

3.加入所要驗證的軟體後，按下【Add】加入檔案後，再按下【OK】。（如圖 4-3）

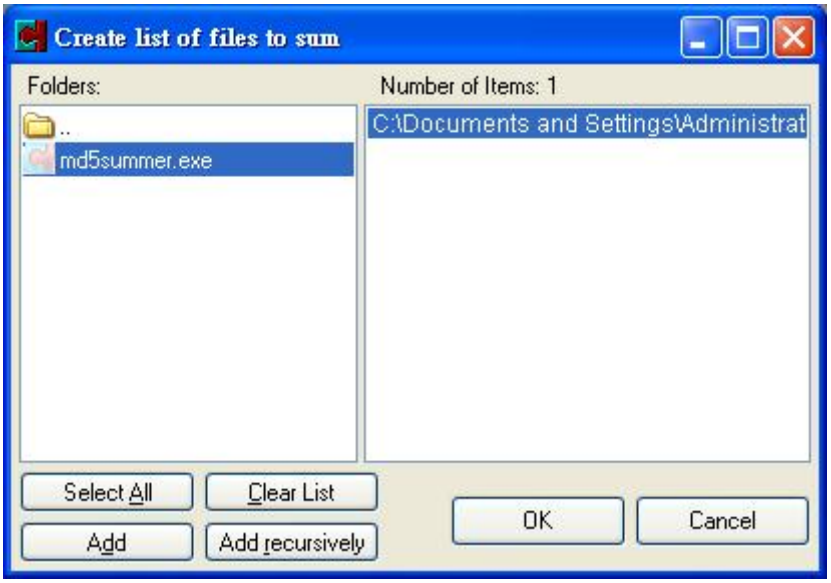

圖 4-3　MD5summer 執行畫面-3

4.按下 OK 之後軟體就開始演算出他的 Hash 值了。(如圖 4-4)

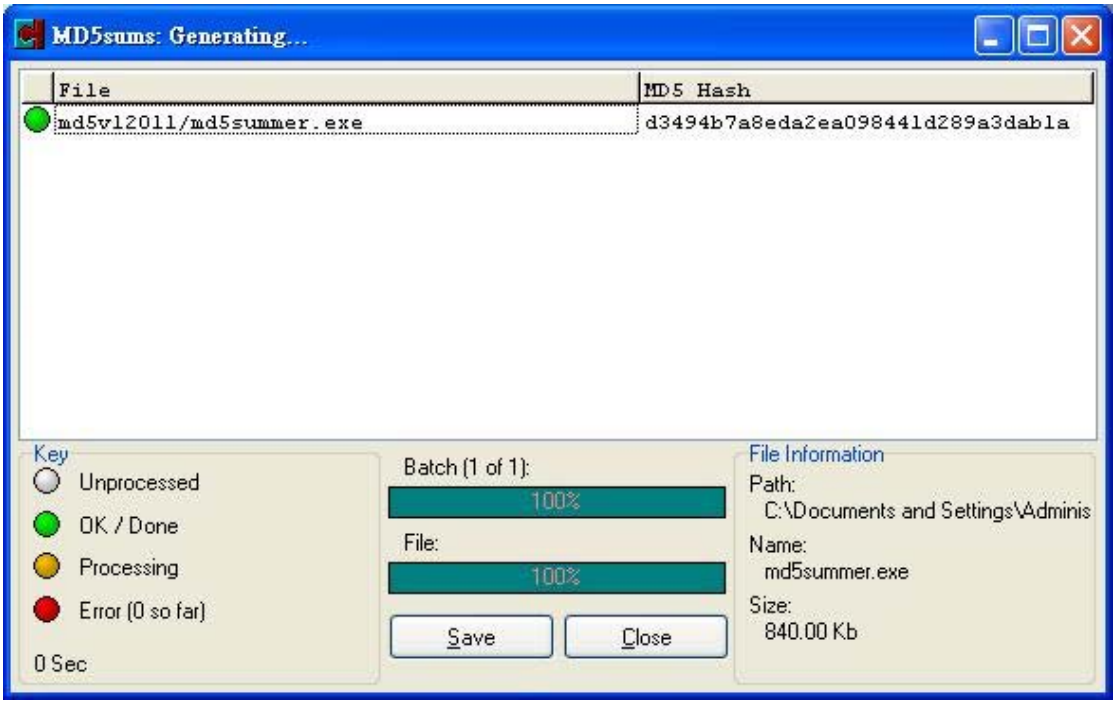

圖 4-4　MD5summer 執行畫面-4

最後只要比對上面的 Hash 值跟對方給的是不是一樣就好了，一樣的話代表檔案無誤，不同的話就代表檔案有問題。

(二)Karen's Hasher[24]

這套軟體可以對任意的檔案、字串或是資料夾進行計算或驗證經過 MD-5， SHA-1， SHA-224， SHA-256， SHA-384 或 SHA-512 所 hash 出來的值。一樣是比對 hash 值，如果是一樣的話就是同一字串或檔案。

使用的方式就是首先按下【Add File(s)…】加入你想演算的檔案。選擇分頁則是採想用哪種的演算法來演算其 Hash 值。按下 Compute 演算法(所選擇之演算法) Hash。這樣就會開始演算了，演算出來的 Hash 也會在上面的表格顯示出來。

Karen's Hasher 演算結果。(如圖 4-5)

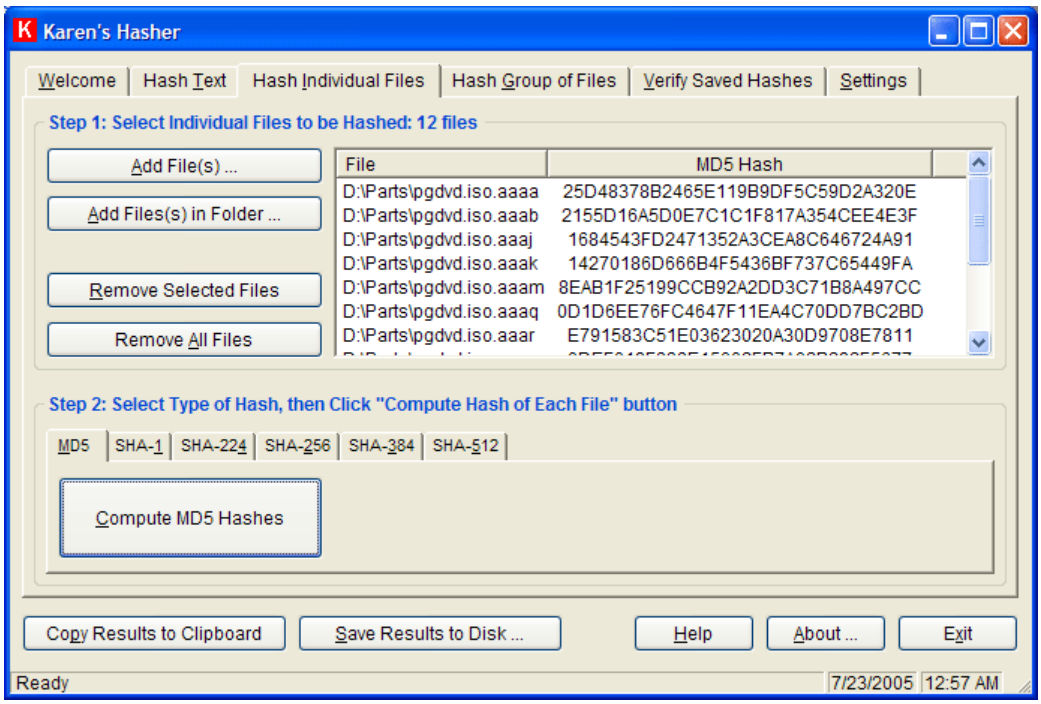

圖 4-5 Karen's Hasher 程式畫面

以上介紹的兩種程式，跟我們的軟體的差異為：

第一個軟體 MD5summer 他只有 MD-5 和 SHA-1 的演算法安全性不是說很高，因為這兩個演算法都被破解了，所以在安全上還是有瑕疵的，遇到有心人士的話就會比較危險了。

第二個軟體 Karen's Hasher 這個軟體他的演算法比較多樣安全性也就大大的提高了。

而我們使用的演算法是 SHA-256 在安全性上也高出 MD-5 和 SHA-1 很多很多，所以我們的東西在安全性上也是很 OK 的，不過我們和這兩個軟體最大的差異是，別的軟體只是單純的單純的做演算法而已，而我們提供了一個線上的平台可以自行上傳 Hash 值和檔案的空間，和一個可以下載檔案的空間，還有提供了留言板有任何的疑問都可以自由的留言發問，還有提供類似聊天室的功能可以讓大家互相交流溝通，總之是一個多功能的應用程式平台。

第五章　結論

因應時代的變遷，全球ｅ化的普及，所造成的網路購物與資料的傳輸越來越頻繁的情況下，在此流竄的金額每年鉅況，因此想要從中謀得不法暴利的人也每日劇增，所以網路所需要的資料安全性就讓人越來越注重，在本次專題我們所使用 SHA 就是網路安全的一種機制，但是網路安全也是人做的，就會有人為了金錢或是商業機密來破解網路的安全機制，其中目前全球所使用的網路安全機制大多是 MD-5 及 SHA-1，可是就在 2004 年 8 月中國王小雲教授，就宣佈理論上以經破解了 MD-5 跟 SHA-1 而 MD-5 更是在最近被國際密碼學家 Lenstra，他在王小雲教授所提供的 MD-5 破解理論下成功的破解 MD-5，偽造了符合標準的數位證書，SHA-1 也在理論上遭到破解，離實際被破解的日子應該不久遠，而全球的網路安全機制正慢慢向往高階的 SHA-256 更換，因此這次我們在設計「具 SHA-256 的資料確認平台」的時候，特地運用 SHA-256 來建立我們軟體的網路安全機制，讓使用者在使用此軟體傳輸功能的時候，能夠更放心的使用我們所設計的軟體，對於整個軟體，最初構想是以程式設計一個類似網頁介面的檔案交換平台，再進行檔案安全加解密驗證的動作，而且能在不同架構有網路的地方執行，然而本系統如果只是個檔案交換平台這樣一個單純的介面，其實用性及功能顯的十分薄弱，於是我們增加了

一些功能使得這整個系統更加完整，現在我們系統目前主要的架構有系統公告、留言板、上下傳檔案&驗證碼、驗證檔案、檔案列表、聊天室…等等初步基本功能。

而在軟體設計的方面，因為使用上的方便以及學習進度來看，我們使用了 C#。它除了強大的功能性以外，也支援跨平台系統，如：PDA、手機….等，還有與 VB 一樣簡單的語法宣告使得在操作和功能上遠遠大於 JAVA、C、C++，因此成為我們這次撰寫軟體的首選。

在網頁上面我們使用的是 ASP.NET 而不是單純的 ASP，因為支援 C# 與 VB 的語法，在整體運作效能比 ASP 大了許多，在網頁介面設計上，擁有和視窗環境非常相似的 Web 控制項，像是 Button、Label…等等。這些控制項都有各自的事件，除此之外，也可以利用程式碼來設定這些控制項的屬性，使得操作方面也簡易了許多卻可以讓整個網頁更加的美觀。

此專題整個設計的方向，為除了擁有傳輸功能跟多種人性化的設計，藉此在整個全球 e 化的情況下能夠為現在的使用者取出他們的所愛跟方便性，而達到設計此軟體的理念，讓人與人之間的互動不會因為工作的忙碌、鋼筋水泥的阻隔中因此平淡下來，有著熱絡 的互動可以讓整個區

域多了活力及向心力讓人們可以更加互助互愛的生存下去，在往後我們不短更新及改進下，也會一步一步的慢慢朝著我們的理念前進，期待那天的到來，也由衷的感謝為我們專題所評分的老師們，謝謝您的指導。

參考文獻

[1] 楊中皇，網路安全的理論與實務，第四章單向雜湊函數，金禾圖書，http://crypto.nknu.edu.tw/textbook/。

[2] 維基百科，http://zh.wikipedia.org/wiki/。

[3] 賴溪松、韓亮、張真誠，2003，近代密碼學及其應用，初版，旗標出版股份有限公司。

[4] Microsoft MSDN，2004，建置安全的 ASP.NET 應用程式: 驗證、授權和安全通訊。

[5] 娃娃，2004，MD5 的介紹、演算法和實現一，電腦世界開發者俱樂部。

[6] 粘添壽、吳順裕著，2004，資訊與網路安全技術，初版，旗標出版股份有限公司。

[7] 李家輝，2008，網路安全、MD4, MD5, SHA-1 破解評論，國立政治　大學，資訊科學研究所專班二。

[8] 翁木龍、楊中皇，2002，Linux環境下以AES及SHA-256強化VPN的設計與實現，第十三屆國際資訊管理學術研討會。

[9] 蔡哲民、http://gec.kmu.edu.tw/~tjm/security/sec4/img8.html 雜湊演算法、崑山科技大學資訊管理學系。

[10] 台灣微軟，Visual Studio，
http://www.microsoft.com/taiwan/vstudio/。
[11] 許清榮、買大誠，2006，Visual C# 2005 建構資訊系統實戰經典教本，初版，博碩文化。
[12] 張雨超，2006，Visual C# 2005 資訊安全程式設計，初版，文魁資訊。
[13] 洪國勝，2006，C# 2005 程式設計範例教本，初版，旗標出版股份有限公司。
[14] 章力民，2006，Visual C# 2005 檔案 IO 與資料存取秘訣，初版，碁峯資訊。
[15] 曹祖聖、蔡文龍，2005，Visual Basic 程式設計經典，第二版，碁峯資訊。
[16] 張耀仁，2004，C++程式設計，碁峯資訊。
[17] 李昇暾、詹智安，2007，Java 物件導向程式設計，初版，旗標出版股份有限公司。
[18] 楊居易，2006，ASP.NET 2.0 程式設計實務，初版，文魁股份。
[19] 陳會安 著、葉怡慧 編輯，2005，JSP2.0 網頁設計範例教本，初版，學貫行銷。

[20] 陳世訓，2006，ASP.NET 2.0 由初學邁向程式設計，初版，金禾資訊。

[21] 蔡俊平，2006 ，ASP.NET 2.0 網站開發實務，初版，加樺國際。

[22] 張瑞立，2007，ASP.NET 2.0 教戰手冊，初版，文魁資訊。

[23] MD5Summer程式首頁，http://www.md5summer.org/

[24] Karen's Hasher程式首頁，http://www.karenware.com/