實作一套適用於 WWW 的無驗證表驗證機制

蔡佳倫
交通大學
crousekimo@yahoo.com.tw

李榮耀
交通大學
jlee@math.nctu.edu.tw

摘要

使用者身份驗證(Authentication)一直是網路安全中一個十分重要而且基本的安全機制，透過這個機制，我們可以保護系統內部的資料不被攻擊者所竊取或盜用，西元 2000 年學者 sun[1]提出了一套以單向雜湊函數及時間戳記為基礎的無驗證表驗證機制，不過，這套驗證機制有許多安全性的問題及時間同步的問題，之後陸陸續續有學者紛紛提出不同的無驗證表驗證機制來改進它的缺失。本研究主要在 WWW 之下實作出一套無驗證表驗證機制，這個驗證機制是以隨機亂數(nonce)以及雜湊函數為基礎，因此，沒有時間同步的問題存在，非常適合目前分散式的網路環境。

關鍵詞：Authentication、Http、smartcard、

1. 前言

隨著網路科技的日益發展，越來越多的系統利用網際網路的便利性來提供使用者進行資訊的交流以及互通，但是在這樣的開放環境下，連線的伺服器及使用者很難去知道連線的對方是否為正確無誤，因此如何驗證通訊雙方的身分變的十分的重要。目前的網路系統大多是利用所謂傳統的帳號密碼認證方式，在系統中先建立一個所謂的認證表(verification table)，並將使用者的帳號密碼存放在這個使用者認證表中，當使用者要登入時，系統會查詢一下這個認證表(veritication table)，是否有這一個使用者的存在，如果存在便讓這個使用者使用這個系統內部的資訊。這樣的實作方式雖然很簡單，但是卻也無形中產生了許多安全性的問題，例如：驗證表的安全性、傳輸帳號密碼的安全性等等問題。為了避免因為驗證表遭到攻擊者盜竊，而導致使用者的帳號密碼被攻擊者所知悉，西元1981年學者Lamport[3]提出一種改善方法，這個方法是先將密碼以雜湊函數將其加密後，再存放到伺服器上，這樣的改善方式雖然可以讓攻擊者不能夠在第一時間就得之密碼值為何，但是攻擊者還是可以透過一些解密工具來破解出該密碼值，只是所需花費的時間很大。

不過，隨著電腦運算速度越來越快的影響，攻擊者破解密碼的速度也越來越快，單純的利用單向雜湊函數加密密碼已經不敷使用，為了要有效解決驗證表被竊的安全性問題，開始有學者提出以無認證表的方式來驗證遠端的使用者身分。西元1990年，Hwang、Chen和Lai[5]提出了一個需要智慧卡的無認證表機制，這個認證機制的特色是伺服器不需要存放任何的驗證表，讓認證表被竊的安全性問題得到了一個解決方案。從此以後，許多以智慧卡為基礎且不需要驗證表的驗證機制紛紛出爐，這些出爐的驗證機制有許多都是以離散對數的困難度或是以非對稱性加密的方式來保護傳輸資訊的安全性，因此這些認證機制在執行的效能上十分的大而且困難，為了避免效能上的問題，學者Sun[1]於西元2000年提出了一種以單向雜湊函數及時間戳記(timestamp)為基礎之無認證表驗證機制，有別於其他無認證表驗證機制，這個驗證機制的特點在於僅僅利用單向雜湊函數與Xor運算，大大降低伺服器與使用者的運算量，不過它的驗證方法還是有著安全性的問題，因此，後續陸陸續續有學者以此為基礎，發表了許許多多相關的研究。

可惜的是，這些研究因為都使用了時間戳記(Timestamp)來保障資訊的傳輸，因此，都有著時間同步的問題，所以開始有學者朝向以隨機亂數(Nonce)的方式來建立驗證機制，因為唯有這樣，才能夠適用於目前分散式的網路環境之中。西元2005年學者Lee、Kim、Yoo[4]提出了一個以隨機亂數(Nonce)為基礎的無驗證表驗證機制，在驗證的過程中不需要有驗證表的存在，而且僅僅利用了單向雜湊函數來保護驗證資料的安全性，不過，有許多專家學者提出這個驗證機制有著許多的安全性問題存在[2]。湊巧的是西元2005年學者Chen、Yeh[7]也提出了另一個以隨機亂數(nonce)為基礎的無認證表驗證機制，在這一個認證機制中不但僅僅只使用單向雜湊函數來驗證使用者外，更在驗證後加入了階段鑰匙(Session Key)的訂定，以加解密後續資料、保護後續資料傳輸的安全性，不過，它的加密次數比起學者Lee、學者Kim和學者Yoo要來的多次。

在本研究提出了一個以隨機亂數(nonce)為基礎的驗證機制並且將它實作在HTTP通訊協定之中，以證明本驗證機制是可行的。不過，瀏覽器執行的單位是”網頁”，因此，為了要能夠實作這一套無驗證表驗證機制，在使用者端我們利用ActiveX技術來製作一個加密型入口介面，這個入口介面會與伺服器後端的ASP網頁進行一連串的溝通，如此才能擴充瀏覽器的功用。

2. 以隨機亂數(nonce)為基礎的驗證機制

這個章節將會介紹本研究所提出的驗證機制，本研究所提出的以隨機亂數為基礎的驗證機制可以分為註冊期、登入期、驗證期三個部份，在介紹前我們的驗證機制之前，首先要先定義一些本章節所需要使用的運算符號，如下表所示：

表 1. 符號說明表

符號	說　　明
h()	單向雜湊函數
x	遠端伺服器的秘密參數，為伺服器所有
⊕	互斥或
ID	使用者帳號
PW	使用者密碼
‖	參數連結
N	使用者及伺服器產生的隨機亂數
X->Y:M	表示從X傳送一個M訊息到Y
U，S	分別代表使用者及伺服器

2.1 註冊期(Registration Phase)

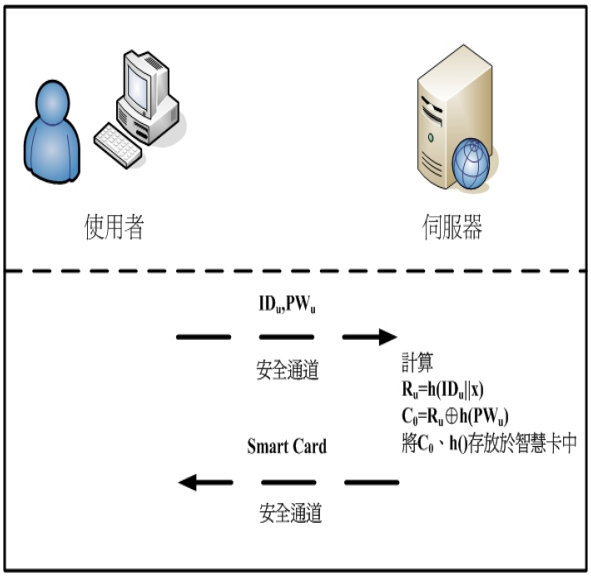

圖 1 註冊期(Registration Phase)

整個註冊期主要是在伺服器上執行，當一個使用者想要使用註冊為一個新的合法使用者時，它首先必須要將自身的ID_u以及PW_u以安全的通道傳送到伺服器，x為伺服器的私密變數，為伺服器所擁有，整個註冊期執行步驟如下所示：

Step1：U_u->S：ID_u

使用者將自己的ID_u、PW_u以安全通道的方式傳送到註冊中心。

Step2：$R_u= h(ID_u||x)$

伺服器利用使用者的ID_u與伺服器的私密參數x去計算出$R_u= h(ID_u||x)$。

Step3：$C_0=R_u\oplus h(PW_u)$

伺服器利用R_u與PW_u計算出$C_0=R_u\oplus h(PW_u)$，並寫入到智慧卡中。

Step4：S->U_u：Smart Card

伺服器將智慧卡以安全通道的方式交給使用者使用。

2.2 登入期(Login Phase)

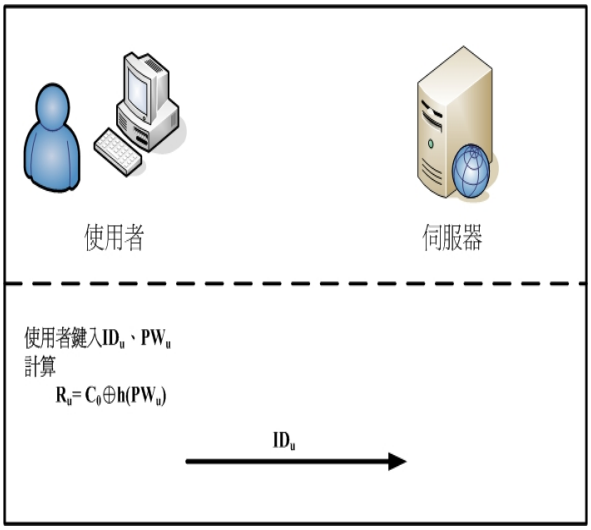

圖 2 登入期(Login Phase)

當一個合法的使用者想要登入系統的時候，這個使用者必須先將個人所擁有的智慧卡插入讀卡機中，讀取智慧卡中所存放的C_0，並且要求使用者輸入他的帳號(ID_u)以及密碼(PW_u)，整個登入期間進行的步驟如下所示：

Step1：$R_u= C_0\oplus h(PW_u)$

讀卡機利用PW_u以及智慧卡中的C_0計算出$R_u=h(ID_u||x)$。

Step2：U_u-> S：ID_u

使用者將自身的ID_u傳送給伺服器，以登入到遠端的伺服器。

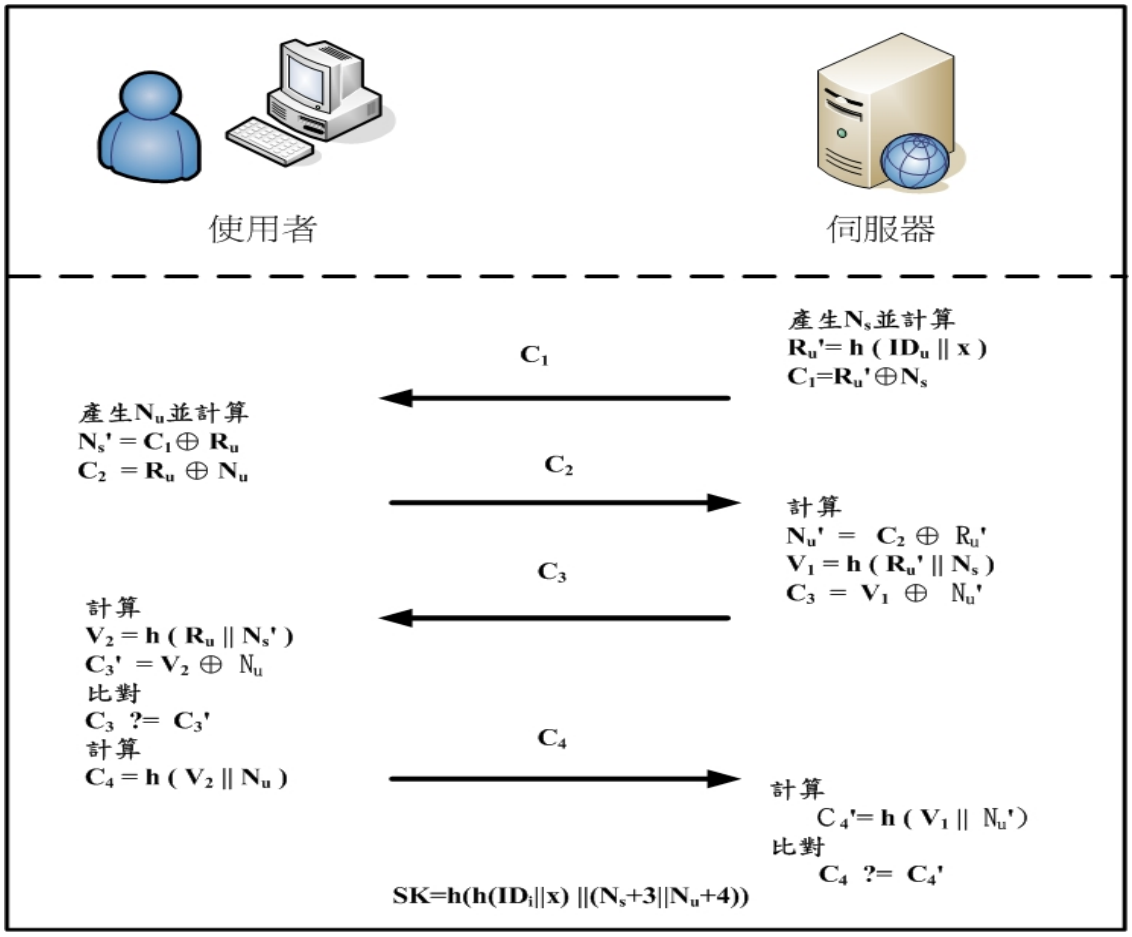

圖 3 驗證期(Authentication Phase)

2.3 驗證期(Authentication phase)

當使用者傳送了ID_u給伺服器之後，伺服器產生隨機亂數N_s與使用者產生的隨機亂數N_c來互相傳遞交換，但是還先不用先驗證這兩個隨機亂數的正確性，只有在後面的驗證流程才會驗證這兩個隨機亂數是否正確，用以驗證使用者以及伺服器的身分。因為這兩個隨機亂數都是在每次登入時隨機產生的，所以每次登入時的資訊都不一樣，在驗證確認使用者以及伺服器的身分之後，最後會產生一個階段鑰匙(session key)，用以加解密驗證時期以後資訊，保護資訊傳輸的安全性。以下是整個驗證時期的步驟：

Step1：$R_u'=h(ID_u||x)$

當伺服器收到使用者傳送過來的ID_u之後，用ID_u及伺服器的秘密參數x計算出R_u'。

Step2：$C_1 = R_u' \oplus N_s$

伺服器產生一個伺服器的隨機亂數Ns，利用R_u'及N_s計算出C_1。

Step3：S->U_u：C_1

伺服器將C_1傳送給使用者。

Step4：$N_s'=C_1 \oplus R_u$

當使用者收到C_1之後，先利用自己本身的R_u與C_1來計算出N_s'。

Step5：$C_2=R_u \oplus N_u$

使用者產生一個隨機亂數N_u，並將N_u與R_u計算出C_2。

Step6：U_u->S：C_2

使用者將C_2傳送給伺服器。

Step7：$N_u'=C_2 \oplus R_u'$

當伺服器收到C_2時，他會利用R_u'與C_2先算出N_u'。

Step8：$V_1=h(R_u'||N_s)$

伺服器利用R_u'與N_s計算出V_1。

Step9：$C_3 = V_1 \oplus N_u'$

伺服器利用V_1與N_u'計算出C_3。

Step10：S -> U_u：C_3

伺服器將C_3傳送給使用者。

Step11：$V_2=h(R_u||N_s')$

當使用者收到C_3之後，他首先利用R_u與N_s'計算出V_2。

Step12：$C_3'=V_2 \oplus N_u$

使用者利用V_2與N_u計算出C_3'。

Step13：C_3 ?= C_3'

使用者比對C_3與C_3'，如果成功，伺服器正確無誤，如果錯誤，便可以知道不是該伺服器並停止下面步驟。

Step14：$C_4=h(V_2||N_u)$

使用者利用C_3'與N_u計算出C_4。

Step15：U_u->S：C_4

使用者將C_4傳送給伺服器。

Step16：$C_4'=h(V_1||N_u')$

當伺服器收到使用者傳來的C_4之後，利用之前的V_1以及N_u'計算出C_4'。

Step17：C_4 ?= C_4'

伺服器比對C_4與C_4'，如果成功，允許使用者進入，如果失敗，不允許使用者進入。

Step18：$SK=h(h(ID_i||x)\ ||(N_s+3||N_u+4))$

當雙方驗證完畢之後，雙方會訂定出階段鑰匙(session key)，階段鑰匙是一把用來加密後續通訊資料的鑰匙。

3. 安全性分析

以下我們用幾個已知的網路攻擊方法來檢視一下我們的通訊協定是否具有相同的網路安全問題，以證明我們所提出的驗證機制是安全無虞的。

3.1 重送攻擊(Replay Attack)

由於在每次的驗證過程中雙方都會自行隨機產生使用者的N_c以及以及伺服器的N_s來改變通訊的內容，而且在後續的驗證過程的資訊都有使用者的N_C以及伺服器的N_s來保護資訊，攻擊者無法在得知幾次的通訊之後，便將該訊息重新送入系統之中，以進入系統，因為只要在驗證的過程中伺服器發現到使用者的無法傳送正確的認證訊息，便會判斷該使用者是有問題的，這時系統便會終止使用者的登入要求，所以我們的驗證機制可以有效的避免重送攻擊(Replay Attack)。

3.2 驗證表被竊攻擊(Stolen-Vertifier Attack)

在我們的驗證機制中，伺服器並沒有存放任何的使用者驗證表，所以攻擊者無法從伺服器上竊取到任何的使用者驗證表，因此，我們的驗證方法可以有效的避免驗證表被竊攻擊(Stolen-Verifier Attack)的攻擊。

3.3 伺服器偽裝攻擊(Server Spoofing Attack)

在我們的驗證方式中，很明顯要求在伺服器驗證使用者之前，使用者要先知道該伺服器是否正確無誤，因此，如果攻擊者想要假冒伺服器以騙取使用者，那他就必須在通訊的過程中事先知道伺服器私密參數x為何，才能夠假冒伺服器來騙取使用者進入系統。如果發現到有人假冒伺服器騙取使用者的話，經過求證該伺服器為假冒之後，使用者端的程式便會中斷後續的資訊傳輸，所以我們的驗證機制可以有效的避免伺服器偽裝攻擊(Server Spoofing Attack)。

3.4 使用者假冒攻擊(Impersonation Attack)

在我們的驗證方法中，一個攻擊者無法利用使用者假冒攻擊(Impersonation Attack)來假冒成其他的合法使用者入侵系統，因為為了要成功利用使用者假冒攻擊去攻擊系統，這個攻擊者必須要知道使用者智慧卡中所存放的 h(IDu||x)，這樣他才能夠順利的製造出合法的驗證訊息，進而假冒使用者入侵系統，不過這是不可能的，因為再登入的訊息中，h(IDu||x)是被伺服器所產生的隨機亂數 Ns 和使用者所產生的隨機亂數 Nu 所保護，攻擊者無法從中解出正確的 h(IDu||x)，而智慧卡中的 h(IDu||x)則是被使用者記憶的密碼所保護，因此，我們的驗證機制可以有效的避免使用者假冒攻擊。

3.5 階段鑰匙的安全性

一把階段鑰匙(session key)的產生來自於$h(ID_u||x)$、N_s以及N_u，這些參數值只有伺服器以及使用者知道，每當使用者與伺服器在通訊結束之後，這把鑰匙便會立即銷毀不會留到下次繼續使用，當使用者重新進入系統時，整個驗證機制再驗證成功之後會另外產生一把新的階段鑰匙(session

key)來加密通訊過程中的資訊，所以假設一個攻擊者在利用某種安全性問題而得知一把階段鑰匙(Session Key)時，該攻擊者也無法再利用這一把階段鑰匙(Session Key)來解密其他通訊過程中的資訊，更因為N_s以及N_u都是隨機產生的，而且在我們的驗證機制中都是被h(ID||x)所隱藏，因此攻擊者無法利用一把已知的階段鑰匙(Session key)來推算出下一把階段鑰匙(Session key)的值為何。而且N_s以及N_u是分別由伺服器以及使用者所產生，而且這個數字非常的大，攻擊者很難以猜測的方式猜測出這把鑰匙的值為何，所以階段鑰匙(Session Key)是非常安全的。

4. 系統實作

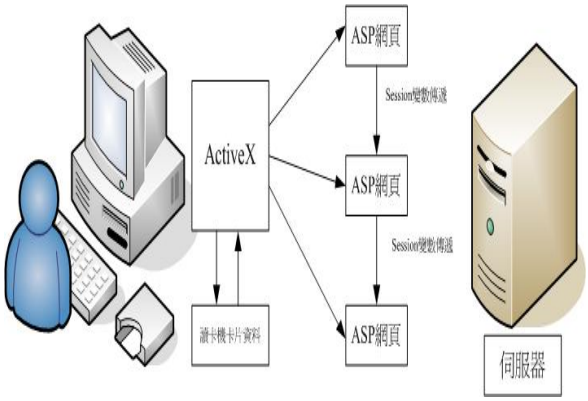

圖 4 系統實作概念圖

本研究主要是在 HTTP 通訊協定之下，實作出一套無驗證表驗證機制，這是因為 HTTP 是目前使用者最常使用的網路通訊協定，保護這個通訊協定便變得十分的重要。為了要能夠在瀏覽器上面執行以達到使用者可以很方便使用，不過，瀏覽器執行的單位是以網頁為主，因此，無法保存使用者所需的驗證參數以及讀取智慧卡中的資料，所以，本研究採用了 ActiveX 技術來發展本研究的使用者的前端程式，這個 ActiveX 會與多個伺服器後端 ASP 網頁進行一連串的驗證流程，而伺服器後端的 ASP 網頁程式會用 session 機制來有效的保存每次溝通的驗證參數，以進行整個完整的驗證機制。

4.1 系統軟硬體介紹

本系統之硬體主要是以虹堡科技的 EZ-100PU 讀卡機以及低價位的 SLE442 卡片為主，EZ-100PU 為目前自然人憑証專用之讀卡機，因此，市面上便可以很輕鬆的買到而且便宜，而 SLE442 卡片是一種低廉的記憶 IC 卡，它的記憶空間為一連串的記憶空間。而在程式的撰寫上，本研究主要是使用 Visual Basic 6.0 來研發相關的程式，選用 Visual Basic6.0 的原因是該程式語言可以很輕鬆的製作出系統所需的 ActiveX 程式。

4.2 使用說明

本研究之實作主要是按照上面所提出的驗證機制，當使用者想要註冊成一個合法使用者時，這個使用者必須先將自身的帳號密碼以安全通道的方式交給網頁伺服器，此時系統管理人員會審查該使用者，當審查通過之後，系統管理人員會將登入必須之相關驗證資訊存放到 IC 卡中，並將 IC 卡轉交給使用者使用。

4.3 選用雜湊函數

雜湊函數加密法的選取上我們選用了 SHA-256 加密法，選用這個加密法的原因是因為中國數學家專家在科學會議上已經提出有效破解的 MD5、SHA-1 雜湊函數的方法，也因為中國數學家的發現，美國國家標準與技術研究院表示，為配合先進的計算機技術，美國政府 5 年內將不再使用 SHA-1，並計劃在 2010 年改用先進的 SHA-224、SHA-256、SHA-384 及 SHA-512 的密碼系統。因此，在這邊我們提供了 SHA-256 的加密技術，以因應未來的時代潮流[8]。

4.4 系統軟硬體介紹

為了達成上面的使用，在我們的工具提供上可以分為三個主要的使用工具，分別加密型帳號密碼寫入工具、加密型登入入口系統介面以及伺服器後端 ASP 子系統等三種，其分別敘述如下：

(一)加密型帳號密碼寫入工具：

為了讓系統管理人員在註冊期時可以新增使用者，本研究提供了系統帳號密碼寫入工具，當使用者以安全通道將使用者的帳號密碼傳送給系統管理人員時，系統管理人員便可以利用這個工具來提供使用者驗證時所需的智慧卡。

圖 5 加密型登入系統帳號密碼寫入工具

(二)加密型登入入口系統介面：

加密型登入入口系統介面主要是提供使用者一個登入的畫面，這個介面在使用者輸入卡片的解密密碼之後，會讀取卡片中的資料並取出網路驗證用的密碼來與遠端的 ASP 網頁進行一連串的驗證流程，讓使用者與伺服器可以互相驗證對方的身份是否合法。

圖 6 加密型登入入口介面

(三)伺服器後端 ASP 子系統：

這一部分的子系統主要是提供伺服器可以和使用者前端的加密型登入入口介面進行一連串的相互驗證工作，為了保存在驗證過程中的驗證參數，本研究使用 Session 機制來進行存放工作。

5. 結論與後續分析

本研究主要是以隨機亂數(nonce)為基礎，並利用單向雜湊函數以及 Xor 運算來建立以及實作出一套以智慧卡為儲存工具的無認證表驗證機制。為了驗證這個驗證機制安全而且可行，本研究也實際將這一個機制實作於 HTTP 通訊協定之中，除了可以了解它是否安全外，更可以有效地提升網站系統的安全性，並去除掉部份不合理的地方。不過，以單向雜湊函數為基礎的驗證機制其安全性皆仰賴於單向雜湊函數的特性-抗碰撞性、不論明文大小皆能輸出固定長度、執行的速度快等等，不過，西元 2004 年 MD5 單向雜湊函數就曾經被中國山東大學的王小雲教授等學者[6]所破解，因此對於雜湊函數的選取上必須盡量選用安全度較高的單向雜湊函數，以避免驗證機制被攻擊者所破解。

參考文獻

[1] H. M. Sun, “An Efficient Remote Use Authentication Scheme Using Smart Cards,” *IEEE Transactions on Consumer Electronics*, Vol. 46, pp. 958-961, 2000.

[2] J. NAM, S. KIM, S. PARK, and D. WON, "Security Analysis of a Nonce-Based User Authentication Scheme Using Smart Cards," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E90-A, pp. 299 - 302, January 2007.

[3] L. Lamport, “Password authentication with insecure communication,” Communications of the ACM, Vol. 24, No. 11, pp. 770-772, Nov. 1981.

[4] S. W. Lee, H. S. Kim and K. Y. Yoo, “Efficient Nonce-based Remote User Authentication Scheme Using Smart Cards,” Mathematics and Computation, Vol. 167, No. 1, pp. 355-361, 2005.

[5] T. Hwang, Y. Chen and C. S. Laih, “Non-interactive Password Authentication without Password Tables,” *IEEE Region 10 Conference on Computer and Communication*, Vol. 1, pp. 429 431, 1990.

[6] X. Y. Wang and H. G. Yu, “How to Break MD5 and Other Hash Functions,” *Eurocrypt*, pp. 19-35, 2005.

[7] Y. C. Chen and L. Y. Yeh, “An Efficient Nonce-based Authentication Scheme with Key Agreement,” *Mathematics and Computation*, pp. 982-993, 2005.

[8] 北方網(2005),世界震驚 美國擔心 王小雲破全球兩大密碼算法，存取於 2006 年五月，http://news.big5.enorth.com.cn/system/2005/03/25/000991494.shtml。