以隨機亂數為基礎之多伺服器無驗證表驗證機制的安全問題

向漢城
國立清華大學資訊工程學系
shc@rtlab.cs.nthu.edu.tw

萬能科技大學資訊管理系
shc@msa.vnu.edu.tw

摘要

自從學者 Sun[14]提出了一種以單向雜湊函數為基礎的單伺服器無認證表驗證機制，此後，許多學者陸續提出利用雜湊函數的單伺服器無驗證表驗證機制（scheme）。但是，這些機制中有個嚴重的問題，任一個使用者想要使用多個伺服器的網路服務時，必須要在這些伺服器上面逐一註冊。對於需要多伺服器服務的使用者而言，使用上相當的麻煩。為了改善這個缺點，學者蔡・李提出了一個多伺服器的無驗證表驗證機制[1]。此驗證機制係使用隨機亂數(nonce)與單向雜湊函數(hash function)來保障資訊傳輸的安全性。並宣稱其設計可以提供雙向身份認證與交談金鑰協議服務。本文將指出蔡・李的設計未能提供完美前推安全(perfect forward secrecy) 且無法抵擋內部特權者攻擊。

關鍵詞：金鑰協議、驗證、多伺服器、智慧卡、通行碼。

1. 前言

隨著網際網路的發展，越來越多的服務架構在網路系統上。透過網際網路的方便性，遠端的使用者除了可以互相分享資訊，更能享受到網路服務所帶來的便利性。例如：電子商務、數位學習、e 化政府、線上繳款、線上娛樂等。但是如何驗證遠端使用者的身分，以提供授權的使用者能存取相關的網路服務就變得十分的重要。西元 1981 年學者 Lamport [9] 提出了一個在不安全的網路環境中驗證使用者的身分的安全機制後，許多的使用者身份驗證機制被提出來，但是這些驗證機制在伺服器上都必需存放一個驗證表來驗證使用者，一旦伺服器遭受攻擊之後，容易發生驗證表外洩的問題。直到西元 1990 年，學者 Hwang 等人[3]提出了一個使用智慧卡的單伺服器無認證表機制後，便開始有越來越多的學者提出各種不同的單伺服器無認證表驗證機制。這些無驗證表驗證機制大多是利用非對稱性加密的方式或是利用解離散對數的困難來保護資料傳輸的安全性，所以在計算的效能上是較耗時。西元 2000 年學者 Sun[14]提出了一種以單向雜湊函數為基礎的單伺服器無認證表驗證機制，這個機制僅僅使用了時間戳記(timestamp)以及單向雜湊函數來驗證遠端使用者的身分，大大降低了執行運算成本(cost)，不過此驗證機制有許多的安全性問題以及使用上的缺點存在。因此，最近幾年中，不斷地有學者提出利用雜湊函數的單伺服器無驗證表驗證機制來增強與改善安全性及效率。

然而，這些單伺服器無驗證表驗證機制有個嚴重的缺失，如果有使用者想要使用多個伺服器的網路服務時，此使用者必須在這些伺服器上個別註冊。因此對於一個需要多個伺服器服務的使用者而言，使用上就變得相當的麻煩。為了改善這個缺點，西元 2001 年學者 Li 等人[11]提出了一個使用類神經網路的多伺服器驗證機制，這個多伺服器驗證機制可以讓使用者僅需要註冊一次，便能使用多台伺服器所提供的網路服務。其後有相當多的研究分別使用不同的方式來提供多伺服器環境的應用，例如：學者 Lin 等人[10]在西元 2003 年提出一種以離散對數為基礎的多伺服器驗證機制，隨後學者 Tsaur 等人[16]也在西元 2004 年提出以 RSA 非對稱性加密法以及 Lagrange interpolating polynomial 為基礎的多伺服器驗證機制的研究。可惜的是這些驗證機制在運算的成本(cost)上都十分的高。直到西元 2004 年學者 Juang[5]提出了一個以對稱性加密演算法為基礎的多伺服器驗證機制，在此多伺服器驗證機制裡同時提供了需要存放驗證表以及不需要存放驗證表兩種不同的驗證方式。此機制使用的方法是透過一台額外的註冊中心(Register center，RC)來幫助伺服器進行驗證，使用者只需要在註冊中心(Register center，RC)上註冊一次，便可以在多台伺服器上驗證使用者的身分。這樣的驗證方式不但解決了使用者需要多次註冊的問題，同時也改善了計算效能過於龐大的問題。西元 2004 年學者 Chang 等人[2]也提出了一個改善學者 Juang 的驗證機制，這個驗證機制也是以對稱性加密為基礎。

最近，學者蔡・李提出了一個創新的多伺服器驗證機制[1]，這個多伺服器驗證機制跟其他的多伺服器驗證機制相比較下，主要的特色在於這個多伺服器驗證機制是以雜湊函數為基礎，而且使用者僅需在註冊中心(Registration Center，RC)上註冊過一次，便可以在多伺服器的環境中驗證使用者的身

分，而不需要進行多次的註冊。而且此多伺服器驗證機制在加密上只使用單向雜湊函數，因此在運算的效能上較其他利用對稱性加密演算法或非對稱性加密演算法的多伺服器驗證機制要來的好，運算成本也較低。此外，這個多伺服器驗證機制使用了隨機亂數(nonce)，所以並沒有時間同步的問題存在，非常適合分散式的網路環境中使用。

可惜的是，本研究發現學者蔡・李所提出的多伺服器驗證機制設計未能提供完美前推安全(perfect forward secrecy) 且無法抵擋內部特權者攻擊。這二個安全問題顯示學者蔡・李所提出的多伺服器驗證機制的安全性是不足的。下面各節將會先簡單的介紹學者蔡・李所提的多伺服器無驗證表驗證機制，然後再提出本研究所發現的安全問題及做一結論。

2. 以隨機亂數為基礎之多伺服器無驗證表驗證機制的回顧

在這一節中，將針對一個隨機亂數為基礎之多伺服器無驗證表驗證機制 [1]（本文中將簡稱為 TL 驗證機制）做一回顧，該驗證機制可以分為使用者註冊期(User Registration phase)、登入期(Login phase)、驗證伺服器與註冊中心期(Authenticate Server and Register Center phase)、驗證伺服器與使用者期(Authenticate Server and user phase)等四個部分。整個驗證機制中包含了使用者 U、遠端伺服器 S 及註冊中心 RC 三個角色。當某台伺服器提出加入時，一旦註冊中心同意該伺服器要求，註冊中心會以該伺服器所提供的 SIDj 來計算出伺服器所私有的 R_s= h(SIDj||y)，R_s 是註冊中心用來確認伺服器的身份是否合法的一個私密鑰匙。註冊中心(Registration Center，RC)會以安全通道的方式將 R_s 轉交給該伺服器，讓該伺服器可以利用 R_s 來進行相關的驗證程序。

在進行此多伺服器無驗證表驗證機制的回顧之前，在表 1 中定義了本研究所使用的相關符號。

表 1 本研究所需之相關符號

符號	說 明
h()	單向雜湊函數
x	使用者私密參數，為註冊中心所擁有
y	伺服器私密參數，為註冊中心所擁有
⊕	互斥或
ID	使用者帳號
PW	使用者密碼
‖	參數連結
N_c	使用者產生的隨機亂數
N_s	伺服器產生的隨機亂數
X⟶Y:M	表示從 X 經由一般通道傳送一個 M 訊息到 Y
X ⇒: Y:M	表示從 X 經由安全通道傳送一個 M 訊息到 Y
U，S	分別代表使用者及伺服器
RC	註冊中心
SID	伺服器的身分識別碼

2.1 使用者註冊期

在此驗證機制中，驗證中心主要任務是驗證遠端的使用者及伺服器。因此，一個使用者如果想要成為一位合法的使用者時，此使用者必須先進行註冊。首先，使用者 U_u 必須以安全通道將自己本身的帳號(ID_u)和密碼(PW_u)傳送到註冊中心進行註冊，當註冊中心接受此使用者時，它便會將使用者登入時所需的資訊存於智慧卡中，再透過安全通道的方式將智慧卡交給使用者。整個註冊期的執行步驟如下所示：

Step1：U_u⇒RC：ID_u、PW_u

遠端的使用者輸入註冊用的帳號 ID_u 及密碼 PW_u，並將此帳號和密碼以安全通道的方式傳送給註冊中心。

Step2: 註冊中心在收到使用者的申請之後，決定是否要讓該使用者成為合法使用者。如果願意接受該使用者的申請，註冊中心使用使用者帳號 ID_u 與註冊中心的私祕參數 x，去計算出 R_u= h(ID_u‖x)，並且使用 R_u 和 PW_u 去計算出 C=R_u⊕h(PW_u)。

Step3： RC ⇒ U_u：h()、C

註冊中心將 h()和 C 存放於智慧卡中，並且將智慧卡以安全通道的方式交給使用者使用。

2.2 登入期

當使用者想要登入系統的時候，他首先必須將自己的智慧卡卡片插入讀卡機中，並輸入自己的帳號(ID_u)以及密碼(PW_u)，登入系統。在登入時使用者會產生一個隨機的隨機亂數(nonce)來改變傳送的資訊，避免攻擊者藉由竊聽的方式取得卡片中重要的資訊，整個登入期(Login Phase)可以分為三個步驟：

Step1：R_u= C⊕h(PW_u)

當使用者輸入他的使用者密碼(PW_u)之後，讀卡機利用 PW_u 從智慧卡中計算出 Ru= C⊕h(PW_u)=h(ID_u‖x)。

Step2：C_1=h(ID_u ‖x) ⊕N_c

使用者產生一個隨機亂數 N_c，利用此隨機亂數 N_c 和 h(ID_u ‖x)去計算 C_1= h(ID_u ‖x) ⊕N_c。

Step3：U_u → S_j：ID_u、C_1

使用者將 ID_u 和 C_1 送往伺服器 S_j，讓伺服器 S_j 進行使用者身份的驗證工作。

2.3 伺服器與註冊中心驗證期

當伺服器收到使用者登入的申請之後，伺服器和註冊中心各自產生一個隨機亂數(nonce)來保護傳送資訊的安全。為了避免不斷的重覆驗證伺服器與註冊中心，在每一次驗證成功之後，註冊中心與伺服器皆存放保護使用者驗證分解參數鑰匙，此時便可以省略通訊的第一步驟 C_2 的計算及傳送、第二步驟、第三步驟、第四步驟以及第五步驟的 C_5 計算及傳送(只需執行步驟一的 ID、SID_j、C_1 之傳送以及步驟五的 C_6 之傳送)。

為了避免攻擊者因為竊聽過幾次伺服器與註冊中心之間的通訊，進而推敲出保護使用者驗證分解參數鑰匙，系統設定註冊中心及伺服器在驗證使用者數次之後，必須重新溝通出新的保護使用者驗證分解參數之鑰匙，以保護使用者分解參數的安全性。整個流程的步驟可以分為下面幾個部份：

Step1：$S_j \rightarrow RC$：ID_u、SID_j、C_1、C_2
當伺服器 S_j 收到使用者傳來的 ID_u 以及 C_1，伺服器產生一個隨機亂數 N_s，並且利用它和 $h(SID_j||y)$去計算出 $C_2=h(SID_j||y) \oplus N_s$，之後伺服器將 ID_u、SID_j、C_1、C_2 傳送到註冊中心。

Step2：$RC \rightarrow S_j$：C_3
註冊中心產生一個隨機亂數 N_{RC} 並計算出 $C_3=N_{RC} \oplus h(SID_j||y)$，將 C_3 送給伺服器 S_j.

Step3：$S_j \rightarrow RC$：C_4
當註冊中心收到伺服器 S_j 傳送過來的 C_3，伺服器 S_j 使用 C_3 和 $h(SID_j||y)$去計算出 N_{RC}'。之後伺服器 S_j 使用 N_{RC}'、N_S、$h(SID_j||y)$計算出 $C_4=h(h(SID_j||y)||N_S) \oplus N_{RC}'$，並且將 C_4 傳送給註冊中心。

Step4：$C_4' ?= C_4$
當註冊中心收到伺服器 S_j 傳來的 C_4，註冊中心計算 $C_4'= h(h(SID_j||y)||N_S') \oplus N_{RC}$ 並且比對 C_4'和 C_4 是否一樣。如果它們相等，代表伺服器 S_j 為合法的伺服器 S_j 無誤，如果不相等，代表伺服器有問題，停止下面的步驟。

Step5：$RC \rightarrow S_j$：C_5, C_6
註冊中心使用 $h(SID_j||y)$與 C_2 去計算出 $N_s'=C_2 \oplus h(SID_j||y)$，再利用 $h(SID_j||y)$、$N_S'+1$、N_{RC}、$h(ID_u||x)$ 以及 N_c 去計算 $C_5=h(h(SID_j||y)||N_s'||N_{RC})$、$C_6=h(h(SID_j||y)||N_S'+1||N_{RC}+2) \oplus h(h(ID_u||x)||Nc)$，並且將 C_5、C_6 傳送給伺服器 S_j 進行驗證工作。請注意 $h(h(SID_j||y)||N_S'+1||NR_C+2)$是雙方用來保護使用者驗證分解參數之鑰匙，讓使用者驗證分解參數能安全的送達伺服器的手中。

Step6：當伺服器 S_j 收到 C_5、C_6 時，S_j 先計算出 C_5'，比對 C_5'跟 C_5 是否一樣，如果一樣代表正確無誤，進行下面的步驟。否則，代表註冊中心有問題，不繼續下面的步驟。

2.4 伺服器與使用者驗證期

為了不讓伺服器知道使用者的 $h(ID_u||x)$，系統將以動態的方式讓註冊中心提供另外一個使用者驗證分解參數 $h(h(ID_u||x)||N_c)$，這個使用者驗證分解參數根據使用者的隨機亂數 N_u 而不斷的改變。當伺服器拿到這一個使用者驗證分解參數後，便可以拿來與使用者互相驗證對方身分：

Step1：$S_j \rightarrow U_u$：V_2、C_9
當伺服器 S_j 確認註冊中心合法後，S_j 首先必須先計算出使用者驗證參數 $C_7=C_6 \oplus h(h(SID_j||y)||N_S+1||N_{RC}+2')=h(h(ID_u||x)||N_c)$，$C_7$ 是使用者驗證參數。然後利用 C_7 去計算 $C_8=C_7 \oplus h(ID_u||x) \oplus N_c$。伺服器 Sj 接著利用 N_s、C_7 以及 C_8 計算出 $V_2=C_7 \oplus N_s$ 及 $C_9=h(C_7||N_s) \oplus C_8$。並將 V_2 及 C_9 傳送給使用者 U_u。

Step2：$C_9' ?= C_9$
當使用者收到 V_2、C_9 時，智慧卡利用 $h(ID_u||x)$ 以及 N_c 計算出使用者驗證分解參數 $C_7'=h(h(ID_u||x)||N_c)$，然後計算出伺服器 S_j 的隨機亂 $N_s'=C_7' \oplus V_2$ 及 $C_8'=C_7' \oplus h(ID_u||x) \oplus N_c$，再利用 C_7'與 C_8'去計算 $C_9'=h(C_7'||N_s') \oplus C_8'$，比對 C_9 與 C_9'，如果相等，代表伺服器 S_j 無誤，進行下面的步驟。否則停止下面的步驟。

Step3：$U_u \rightarrow S_j$：C_{10}
計算出 $C_{10}= h(C_7'||C_8'||N_s')$，並將它送到伺服器 S_j 以進行使用者身份驗證工作。

Step4：$C_{10}' ?= C_{10}$
伺服器 S_j 計算出 $C_{10}'= h(C_7||C_8||N_s)$，並比較 C_{10} 跟 C_{10}'是否相等？如果不相等，便停止下面步驟。如果相等，代表使用者身份無誤，雙方定義出一把階段鑰匙(session key) $SK= h(C_7+1||C_8+2||N_s+3)$，後續資料利用這一把階段鑰匙(Session key)加密，以保護資料的安全性。整個多伺服驗證機制流程如圖 1 所示。

3. 相關的安全問題

本節將指出TL驗證機制設計不具有完美前推安全(perfect forward secrecy) [13, 15]及無法抵擋內部特權者攻擊（Privileged insider’s attack）[6, 7]。

3.1 不具有完美前推安全(perfect forward secrecy)

完美前推安全(perfect forward secrecy) [13, 15]是用來評估一個遠端驗證機制的安全強度是否足夠的重要考量因素。所謂完美前推安全（Perfect Forward Secrecy, PFS）指即使遠端驗證機制任一角色的生命週期較長的秘密金鑰（long-term private key）或使用者密碼不小心洩漏了，在此秘密金鑰洩漏之前所產生的會議金鑰也不會因此而連帶洩漏。

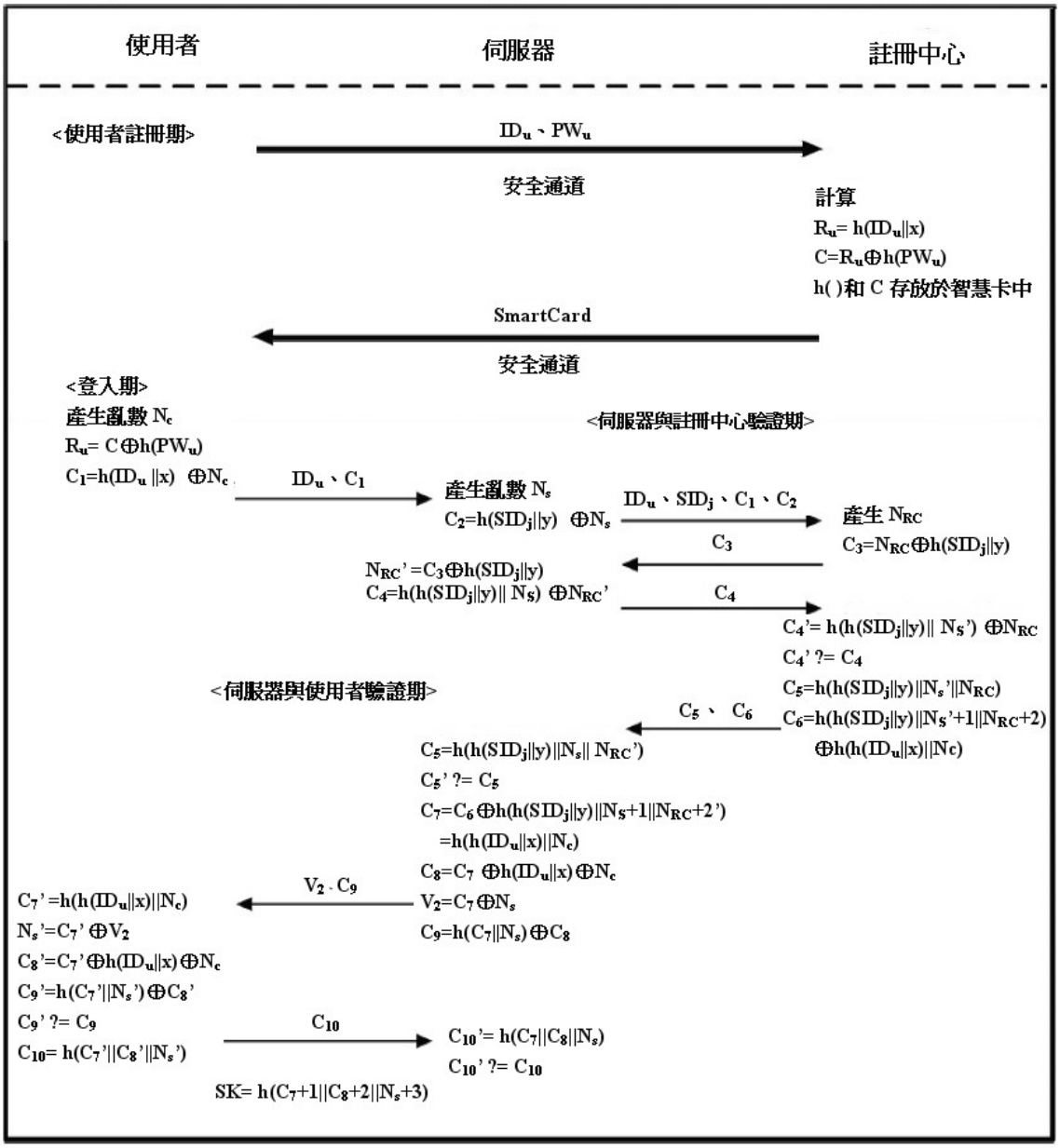

圖 1 以隨機亂數為基礎之多伺服器無驗證表驗證機制

其主要作用是對於過去被加密的資料提供完善的機密性保護。就算所有的使用者的秘密金鑰全部洩漏，系統還能保護所有過去已被加密的資料。例如，知名的 Diffie-Hellman 金鑰協議（key agreement）就具有完美前推安全。但在 TL 驗證機制中，卻未提供完美前推安全。因為一旦秘密金鑰 x 及 y 洩露，階段鑰匙(session key) SK 將被解開，所有先前的通訊資料將被攻擊者取得。詳細過程如下：

在 TL 驗證機制中，假設攻擊者 Eve 從破解註

冊中心或其安全漏洞中或因某些原因獲得秘密金鑰 x 及 y 並在開放網路上攔截到由使用者傳送給伺服器 S_j 的登入訊息{ID_u、C_1}及由伺服器 S_j 傳送到註冊中心的訊息{ID_u、SID_j、C_1、C_2}。要獲得這些訊息是相當容易，因為這些訊息是被暴露在開放的網路中。接下來，Eve 使用所攔截到的 ID_u、SID_j、C_1、C_2、x 和 y 分別計算出 $N_s = h(SID_j||y) \oplus C_2$ 及 $N_c = h(ID_u||x) \oplus C_1$。當 N_s 及 N_c 被計算出來後，Eve 再利用 N_s 及 N_c 導出 C_7 及 C_8。其導出過程為 $C_7 = h(h(ID_u||x)||N_c)$，$C_8 = C_7 \oplus h(ID_u||x) \oplus N_c$。取得 N_s、N_c、C_7 及 C_8 後，就可輕易導出共用的階段鑰匙(session key) $SK = h(C_7+1||C_8+2||N_s+3)$。Eve 使用階段鑰匙 SK 就能容易地解出先前通訊的資料。所以 TL 驗證機制設計很顯然地不具有完美前推安全(perfect forward secrecy)。

3.2 內部特權者攻擊（Privileged insider's attack）

TL 驗證機制易遭受一個內部特權者攻擊[10, 11]詳如下列所述：

在 TL 驗證機制中，使用者只需要在註冊中心上註冊過一次，便可以在多伺服器的環境中驗證使用者的身分，而不需要再向個別的伺服器進行註冊。在 TL 驗證機制的註冊期中， 使用者 U_i 使用帳號 ID_i 及密碼 PW_i 向註冊中心進行註冊。由於 ID_i 及 PW_i 是直接傳送到註冊中心，並未做任何加密或資訊隱藏，所以註冊中心的內部特權者可以很輕易取得 ID_i 及 PW_i。 然後，註冊中心的內部特權者可以試著使用 PW_i 假冒是 U_i 登入本系統之外 U_i 已註冊的其他伺服器，如果此外面的伺服器是採用正常的密碼識別系統，則註冊中心的內部特權者是很有可能使用 PW_i 成功地假冒 U_i 登錄此伺服器。雖然也有可能，伺服器的所有有特權內部人員是值得信賴，而且 U_i 並未使用相同的密碼去存取不同的伺服器。但驗證機制的制定者及使用者應該意識到有這樣一個潛在的安全弱點。因為這樣的原因，在許多密碼認證機制裡[4, 8, 12]，使用者密碼是不會暴露給註冊中心及伺服器知道。

4. 結論

如何正確且有效率的驗證使用者的身分一直是網路安全中一個重要的議題。在 TL 驗證機制中，提出了一種創新的多伺服器驗證機制。此驗證機制在伺服器與註冊中心都不需要存放任何的使用者驗證表，並以隨機亂數(nonce)為基礎。因此，可以廣泛的應用在分散式的網路環境之中，且在加密的方法上僅僅使用單向雜湊函數，所以可以有效避免伺服器執行效率的問題。在本文裡，我們已指出 TL 驗證機制設計不具有完美前推安全(perfect forward secrecy)及無法抵擋內部特權者攻擊（Privileged insider's attack)。因此，TL 驗證機制安全強度顯然是相當不足的。

參考文獻

[1] 蔡佳倫、李榮耀。2007。一個以隨機亂數為基礎之多伺服器無驗證表驗證機制。TANET 2007。

[2] C. C. Chang and J. S. Lee, "An efficient and secure multi-server password authentication scheme using smart cards," International Conference on Cyberworlds, pp. 417-422, Nov. 2004.

[3] T. Hwang, Y. Chen, and C. S. Laih, "Non-interactive password authentication without password tables," IEEE Region 10 Conference on Computer and Communication System, Vol. 1, pp. 429-431, Sept. 1990.

[4] M. S. Hwang, C. C. Lee, and Y. L. Tang, "A simple remote user authentication scheme," Mathematical and Comput. Modelling, vol. 36, no. 1-2, pp. 103-107, July 2002.

[5] W. S. Juang, "Efficient multi-server password authenticated key agreement using smart cards," IEEE Transactions on Consumer Electronics, Vol. 50, No. 1, pp. 251-255, Nov. 2004.

[6] Ku W. and S. Chen, "Weaknesses and Improvements of an Efficient Password based Remote User Authentication Scheme using Smart Cards," IEEE Transactions on Consumer Electronics, Vol. 50, No.1, pp.204-207, 2004.

[7] Ku W.C., H.M. Chuang, and M.J. Tsaur, "Vulnerabilities of Wu-Chieu's Improved Password Authentication Scheme Using Smart Cards," IEICE Trans. Fundamentals, Vol. E88-A, No.11, pp.3241-3243, 2005.

[8] W. C. Ku and S. M. Chen, "Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards," IEEE Transactions on Consumer Electronics, Vol. 50, No. 1, pp. 204-207, Feb. 2004.

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

[10] I. C. Lin, M. S. Hwang, and L. H. Li, "A new remote user authentication scheme for multi-server architecture," Future Generation Computer System, Vol. 19, No. 1, pp. 13-22, Jan. 2003.

[11] L. H. Li, I. C. Lin, and M. S. Hwang, "A remote password authentication scheme for multi-server architecture using neural networks," IEEE Transactions on Neural Network, Vol. 12, No. 6, pp. 1498-1504, Nov. 2001.

[12] C. C. Lee, M. S. Hwang, and W. P. Yang, "A

flexible remote user authentication scheme using smart cards," ACM Operat. Syst. Rev., vol. 36, no. 3, pp. 46-52, July 2002.

[13] A.J. Menezes, P.C. Oorschot, and S.A. Vanstone Handbook of Applied Cryptograph, CRC Press, New York, 1997.

[14] H. M. Sun, "An efficient remote use authentication scheme using smart cards," IEEE Transactions on Consumer Electronics, Vol. 46, No. 4, pp. 958-961, Nov. 2000.

[15] H.M. Sun and H.T. Yeh, Password-based authentication and key distribution protocols with perfect forward secrecy, Journal of. Computer System Science, Vol. 72, pp.1002–1011, 2006.

[16] W. J. Tsaur, C.C. Wu, and W.B. Lee, "A smart card-based remote scheme for password authentication in multi-server Internet services," Computer Standard & Interfaces, Vol. 27, No. 1, pp. 39-51, 2004.