南　華　大　學

資訊管理學系

碩士論文

改善一個遠端使用者認證方案與金鑰協商

Improvement on a remote user authentication scheme with key agreement

研 究 生：廖一清

指導教授：周志賢 博士

中華民國　105　年　6　月　27　日

南 華 大 學

資 訊 管 理 學 系

碩 士 學 位 論 文

論文題目(中文)：改善一個遠端使用者認證方案與金鑰協商

論文題目(英文)：Improvement on a remote user authentication scheme with key agreement

研究生：廖一清

經考試合格特此證明

口試委員：

周志賢

許乙清

指導教授：周志賢

系主任(所長)：

口試日期：中華民國 105 年 6 月 27 日

南 華 大 學 碩 士 班 研 究 生

論 文 指 導 教 授 推 薦 函

資訊管理系碩士班 廖一清 君所提之論文

改善一個遠端使用者認證方案與金鑰協商

(Improvement on a remote user authentication scheme with key agreement)

係由本人指導撰述，同意提付審查。

指導教授 周志賢

105 年 5 月 15 日

南華大學資訊管理學系碩士論文著作財產權同意書

立書人：廖一清 之碩士畢業論文

中文題目：**改善一個遠端使用者認證方案與金鑰協商**

英文題目：**Improvement on a remote user authentication scheme with key agreement**

指導教授：周志賢 博士

學生與指導老師就本篇論文內容及資料其著作財產權歸屬如下：

☑ 共同享有著作權

☐ 共同享有著作權，學生願「拋棄」著作財產權

☐ 學生獨自享有著作財產權

學　生：廖一清（請親自簽名）

指導老師：周志賢（請親自簽名）

中　華　民　國　105　年　6　月　20　日

誌　　謝

感恩南華大學老師增益學生慧命的無私奉獻，時節交替，歲月如流水，碩士在職專班二年的日子即將隨著這篇誌謝的完成進入尾聲。

現代科技方面首先感恩 微軟、Google、Adobe、Nuance、...、.. 、. 等這些軟件公司開發了很好的產品，讓學生學習到新的技能與未來的美麗憧景。其次在硬體方面要感恩 Asus 公司提供了很穩定的產品，讓學生得以不中斷的學習。最後在人文方面要感恩三好校園這二年來所提供的優質服務，讓學生有一種賓至如歸，家的感覺。

啟蒙恩師方面:
感恩第一堂課_雲端計算與應用_曾俊雄老師，電子商務理論_邱英華老師，電子交易安全技術_周志賢老師暨指導教授，研究方法論_洪銘建老師。

一下課程_德育原理專題_社科院_陳宏模老師，語意網技術_邱英華老師，資料分析_吳梅君老師，高等資訊管理_王佳文老師，旁聽_RFID_張介耀老師，旁聽_Photoshop & Illustrator _蔡德謙老師。

二上課程_供應鏈管理_張介耀老師，資訊保密方法_周志賢老師暨指導教授，網路技術與管理_蔡德謙老師，影音多媒體設計與研究_謝定助老師。旁聽_財務管理_鍾國貴老師 (您不理財、財不理您)。

二下課程_畢業論文_改善一個遠端使用者認證方案與金鑰協商 (Improvement on a remote user authentication scheme with key agreement)，旁聽_資訊保密_周志賢老師暨指導教授，每個星期四_國科會計劃_Meeting_周志賢老師暨指導教授。

另外感恩資管 2A 同學這二年的陪伴，讓學習的過程格外有趣以下是同學名單:林宜申、孫愛盈、蔡瑞銘、王瑞男、吳長翰、劉昇益、林秋芬、王思梅、王秀芬、蔡旻家、陳惠美、姚博文、蘇珍婷、凌于雯、翁卓偉。

感恩口試老師:
南華大學資管系指導教授:周志賢 博士
虎尾科技大學資工系教授:許乙清 博士
南華大學資管系助理教授:尤國任 博士

一清 2016/06/20 南華大學

改善一個遠端使用者認證方案與金鑰協商

學生：廖一清　　　　　　　　　　　指導教授：周志賢 博士

南　華　大　學　資訊管理學系碩士班

摘　　要

最近，韋瑪黎(Kumari)等學者指出張(Chang)等學者的智慧卡驗証身份協定“以動態身份為基礎之不可追蹤的遠端使用者可驗證密碼更新認證方案”不僅有幾個缺點，而且也沒有提供任何會議金鑰的協商機制。因此，他們提出了一個具有金鑰協商的改進方案。經過密碼分析後，他們確認了他們方法的安全性。然而，經過我們進一步檢視該改進方案後，發現他們的方法仍然遭受到匿名的揭露和智慧卡丟失時的密碼猜測攻擊。上述二者是廖(Liao)等學者所主張一個安全智慧卡身份驗証協定中十個基本規範中的兩個，是一般在研究使用智慧卡作安全身份認證協定所必需尊循的規則。基於此，我們修改了他們所提的改進方案，以包含這些被遺漏的安全性，這在一個使用智慧卡來做使用者身份認證協定的系統內是相當重要的。

關鍵字:使用者認證，金鑰協商，密碼分析，智慧卡，更換密碼，不可追蹤，動態身份，匿名，遠端使用者認證

Improvement on a remote user authentication scheme with key agreement

Student：Liao, Yi-Ching Advisors：Dr. Chou, Jue-Sam.

Department of Information Management
The Graduated Program
Nan-Hua University

ABSTRACT

Recently, Kumari *et al*. pointed out that Chang *et al*.'s "Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update" not only has several drawbacks, but also does not provide any session key agreement. Hence, they proposed an improvement with key agreement on the scheme. After cryptanalysis, they confirmed its security properties. However, we determined that the improved scheme still suffers from both anonymity breach and the smart card loss password guessing attack, which are two of the ten basic requirements in a secure identity authentication protocal using smart card, insisted by Liao *et al*. Therefore, we modified their improvement to include those desired security functionalities, which are significantly important in a user authentication smart card system.

Keywords: user authentication, key agreement, cryptanalysis, smart card, password change, untraceable, dynamic identity, anonymity, remote user authentication

目錄

論文指導教授推薦函 i
碩士論文著作財產權同意書 ii
誌謝 iii
中文摘要 iv
英文摘要 v
目錄 vi
表目錄 (List of Tables) viii
圖目錄 (List of Figures) ix
符號說明 (Notation table) x
Abstract 1
1. Introduction 1
2. Review of Kumari *et al.*'s scheme 2
  2.1 Registration Phase 2
  2.2 Login phase 3
  2.3. Authentication phase 3
3. Weakness of the scheme 3
  3.1 The insider attack on the protocol's anonymity property 4
  3.2 The smart card loss password guessing attack 4
4. Modification 4
  4.1 Registration phase 5
  4.2 Login and authentication phase 6
  4.3 Password change phase 8
5. Security analysis 8
  5.1 The user password is not stored on the server. 9
  5.2 The user can freely choose / change the password. 9
  5.3 The password cannot be revealed by the administrator of the server. 9
  5.4 The user password is not transmitted in plain form over the internet. 9
  5.5 The scheme can resist the insider attacks. 9
  5.6 The scheme can resist the replay, modification-verifier-table, and stolen-verifier attacks. 9
  5.7 The length of a password is appropriate for memorization. 9
  5.8 The scheme can be efficient and practical. 10
  5.9 The scheme can achieve mutual authentication. 10
    (1) Mutual authentication: 10

(2) Man-In-the Middle attack: 10
5.10 Even if the smart card is lost, it can resist the password guessing attack........ 11
(1) Supposing the user’s smard card is obtained by AE after registration. 11
(2) The card is obtained by **AE** after the login and authentication phase. 11
6. Conclusion 12
References 12
附錄一 14

List of Tables

Table 1. notations definitions 5

List of Figures

Fig. 1. The Rigistration phase... 5
Fig. 2. The Login and the Authentication... 7
Fig. 3. The Password Change Phase.. 8
Figure 4. MIMA on our scheme as shown in Figure 2.. 11

符 號 說 明

Table 1. notations definitions

Notation table
Pw_i : user i's password.
RPw_i : user i's randomized password.
b : a random number.
$\|\|$: concatenation operation.
$\oplus$: bitwise *Xor* operation.
$h(.)$: a collision free one-way hash function.
ID_i : user i's identity.
r_i, y_i : user i's two nonces.
S_i : the ith server.
U_i : the ith user.
AE : an attacker.
T_i : user i's current timestamp.
T_s, T_{ss} : *server*'s two current timestamps.
x, y : *server*'s two secret numbers.
SC_i : user i's smart card.

Improvement on a remote user authentication scheme with key agreement

Yalin Chen[1] and Jue-Sam Chou*[2] and Yi - Ching Liao[3]

[1] Institute of information systems and applications, National Tsing Hua University

[2] Department of Information Management, Nanhua University, Taiwan
*: corresponding

[3] Department of Information Management, Nanhua University, Taiwan

Abstract

Recently, Kumari *et al*. pointed out that Chang *et al*.'s "Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update" not only has several drawbacks, but also does not provide any session key agreement. Hence, they proposed an improvement with key agreement on the scheme. After cryptanalysis, they confirmed its security properties. However, we determined that the improved scheme still suffers from both anonymity breach and the smart card loss password guessing attack, which are two of the ten basic requirements in a secure identity authentication protocal using smart card, insisted by Liao *et al*. Therefore, we modified their improvement to include those desired security functionalities, which are significantly important in a user authentication smart card system.

Keywords: user authentication, key agreement, cryptanalysis, smart card, password change, untraceable, dynamic identity, anonymity, remote user authentication

1. Introduction

There have been many cryptographic scientists working in the system design of remote user authentication using smart card (*SC*) [1-19]. A user authentication using smart card system typically contains two roles: the user and the server; and three protocols: registration, login and authentication, and password change. In the design principle, the user's identity should not be revealed to the outside world to ensure his login privacy.

In 2014, Kumari *et al*. [13] pointed out that Chang *et al*.'s scheme [14] has some shortcomings. It suffers: (1). offline password guessing attack, (2). impersonation attack, (3). insider attack, (4). anonymity breach when the smart card is obtained by a legal user, (5). denial of service attack, and (6). lacking session key agreement. Hence, they overcome the security weaknesses by proposing a new one with key agreement. It provides with user anonymity, establishes proper mutual authentication, and offers a secure password change phase, without maintaining any database record at the server side. They claimed that the proposed scheme could resist various attacks, including those existed in Chang *et al*.s' and outperform six other related schemes in the aspect of security characteristics. However, upon a closer examination, we discovered that it suffers from two security weaknesses: (1). anonymity breach, and (2). the smart card loss password guessing attack. To enhance its security, we modified their scheme to include these features. We will demonstrate the enhancement in this paper.

The rest of this article is organized as follows. In Session 2, we briefly introduce Kumari *et al*.'s improvement on Chang *et al*.'s scheme. In Session 3, we analyze its weaknesses. The modifications and related security issue discussions are demonstrated in Session 4 and 5, respectively. Finally, a conclusion is given in Session 6

2. Review of Kumari *et al*.'s scheme

Kumari *et al*.'s "An improved remote user authentication with key agreement" is based on Chang *et al*.'s scheme [14]. It also consists of two roles: the user and the remote server; and three phases: registration, login and authentication, and password change. They claimed that their scheme not only eliminate all the security vulnerabilities existed in Chang *et al*.s' but also introduce the session key agreement function. In this article, we only review their registration phase, and the login and authentication phase, to illustrate its weaknesses. As for the definitions of the used notations, please refer to the original article.

2.1 Registration Phase

When user U_i registers to the service provider server S_i, this phase is performed as follows:

(1) The user U_i chooses his identity ID_i, password Pw_i, and selects a random nonce b. He then computes $RPw_i = h(b \| Pw_i)$ and sends the registration message $\{ID_i, RPw_i\}$ to S_i over a secure channel.

(2) After receiving the registration message from U_i, S_i then chooses a random number y_i, which is different from all of the other users'.

(3) S_i computes the values $N_i = h(ID_i \| x) \oplus RPw_i$, $Y_i = y_i \oplus h(ID_i \| x)$, $D_i = h(ID_i \| y_i \| RPw_i)$, and $E_i = y_i \oplus h(y \| x)$.

(4) S_i stores the values $\{Y_i, D_i, E_i, h(.)\}$ into U_i's smart card (SC_i) and delivers SC_i and N_i to U_i via a secure channel.
(5) After receiving SC_i, U_i computes $A_i = (ID_i \| Pw_i) \oplus b$ and $M_i = N_i \oplus b$, and inserts them into SC_i which thus now contains the parameters $\{Y_i, D_i, E_i, h(.), A_i$, and $M_i\}$. U_i hereafter needs not to remember the random number b anymore.

2.2 Login phase

This phase is for user U_i to access the needed resources from a server. U_i inserts his SC_i into a card reader and inputs his username ID_i and password Pw_i. SC_i then verifies its owner with the secret data stored by using the following steps.

(1) First, SC_i computes $b = A_i \oplus h(ID_i \| Pw_i)$, $RPw_i = h(b \| Pw_i)$, $h(ID_i \| x) = M_i \oplus RPw_i \oplus b$, and $y_i = Y_i \oplus h(ID_i \| x)$. It then computes $D_i{*} = h(ID_i \| y_i \| RPw_i)$.
(2) SC_i verifies whether the equation $D_i{*} = D_i$ holds, if it does not hold, SC_i drops the session. And U_i is required to enter PUK (*Private Unblock Key*) to re-activate his SC_i.
(3) Only if $D_i{*} = D_i$ holds, SC_i proceeds further. It computes the values $h(y \| x) = y_i \oplus E_i$, $N_i = M_i \oplus b$, $CID_i = ID_i \oplus h(N_i \| y_i \| T_i)$, $N_i' = N_i \oplus h(y_i \| T_i)$, $B_i = N_i \oplus RPw_i = h(ID_i \| x)$, $C_i = h(N_i \| y_i \| B_i \| T_i)$, and $F_i = y_i \oplus (h(y \| x) \| T_i)$, where T_i is the system's current timestamp.
(4) SC_i then transfers the login request $= \{CID_i, N_i', C_i, F_i, T_i\}$ to S_i.

2.3. Authentication phase

After receiving the login request, S_i and U_i together perform the following steps to authenticate each other:

(1) S_i verifies to see whether $(T_s - T_i) < \Delta T$ holds, where T_s is the current timestamp. If it does, S_i retrieves $y_i = F_i \oplus (h(y \| x) \| T_i)$, $N_i = N_i' \oplus h(y_i \| T_i)$, and $ID_i = CID_i \oplus h(N_i \| y_i \| T_i)$. It then computes $B_i{*} = h(ID_i \| x)$, $C_i{*} = h(N_i \| y_i \| B_i{*} \| T_i)$, and compares $C_i{*}$ with the received C_i.
(2) If $C_i{*} = C_i$ holds, S_i confirms the legality of U_i. It then computes $a = h(B_i{*} \| y_i \| T_{ss})$, and transmits $\{a, T_{ss}\}$ to SC_i, where T_{ss} is the server's current timestamp.
(3) On receiving $\{a, T_{ss}\}$, SC_i checks T_{ss} for freshness. If T_{ss} is fresh, SC_i computes $a{*} = h(B_i \| y_i \| T_{ss})$ and verifies to see whether $a{*} = a$ holds. If it does, SC_i confirms the legality of the server.
(4) After successful mutual authentication, U_i and S_i both compute the common session key as $sk = h(B_i \| y_i \| T_i \| T_{ss} \| h(y \| x))$ and $sk{*} = h(B_i{*} \| y_i \| T_i \| T_{ss} \| h(y \| x))$, respectively.

3. Weakness of the scheme

Due to the parameters $\{Y_i, D_i, E_i, h(.), A_i,$ and $M_i\}$ stored in the smart card and the user's ability in computing the values $b = A_i \oplus h(ID_i \| Pw_i)$, $RPw_i = h(b \| Pw_i)$, $h(ID_i \| x) = M_i \oplus RPw_i \oplus b$, and $y_i = Y_i \oplus h(ID_i \| x)$, an insider can compute $y_i \oplus E_i$. That is, each user can know the value $h(y \| x)$, because $E_i = y_i \oplus h(y \| x)$. Under this situation, we can see that their scheme suffers from: (1) anonymity breach, and (2) the smart card loss password guessing attack. We describe them both below.

3.1 The insider attack on the protocol's anonymity property

If a user Bob's login request $\{CID_i, N_i', C_i, F_i, T_i\}$ is intercepted by an insider attacker *Alice*, with the knowledge of $h(y \| x)$ *Alice* can know Bob's y_i by calculating $y_i = F_i \oplus (h(y \| x) \| T_i)$. She then computes $ID_i = CID_i \oplus h(N_i \| y_i \| T_i)$. That is, *Alice* knows the user's identity ID_i, which now is Bob. Therefore, the attack succeeds.

3.2 The smart card loss password guessing attack

From the collected login request messages $\{CID_i, N_i', C_i, F_i, T_i\}$, and from the knowledge of $h(y \| x)$ and the equations $y_i = F_i \oplus (h(y \| x) \| T_i)$, $h(y \| x) = y_i \oplus E_i$, the insider *Alice* can calculate the corresponding E_is of each U_i's login request by computing $E_i = y_i \oplus h(y \| x)$. Therefore, once Bob, who has ever loggined to the server, loses his smart card which was obtained by *Alice*, then from the equations, $N_i = N_i' \oplus h(y_i \| T_i)$ and $ID_i = CID_i \oplus h(N_i \| y_i \| T_i)$, and from comparing the calculated E_is with the E_i stored in the lost card, *Alice* can identify which intercepted login request is Bob's. After obtaining the knowledge of Bob's ID_i, and the stored values A_i, D_i, *Alice* can successfully launch a smart card loss password guessing attack as follows.

She first guesses the lost card owner's password as Pw_i', and then computes $b' = A_i \oplus h(ID_i \| Pw_i')$, $RPw_i' = h(b_i' \| Pw_i')$, and $D_i' = h(ID_i \| y_i \| RPw_i')$. Obviously, we can see that if $D_i' = D_i$, *Alice* can confirm that Pw_i is Bob's password. Therefore, the attack succeeds.

4. Modification

From the weaknesses found in Section 3, we note that the key point is that the insider can obtain the server's secret $h(y \| x)$. To further disguise it, we modify the messages, *e.g.*, replace the value $h(y \| x)$ with $h(y \| x \| y_i)$, where y_i is U_i's dedicated random number, in the registration phase and the login and authentication phase. We show the modifications as follows, and depict them in Fig.1 through Fig.3, respectively. As for the definitions of used notations, please refer to Table 1.

Table 1. notations definitions

Notation table
Pw_i : user i's password.
RPw_i : user i's randomized password.
b : a random number.
$\|\|$: concatenation operation.
$\oplus$: bitwise *Xor* operation.
$h(.)$: a collision free one-way hash function.
ID_i : user i's identity.
r_i, y_i : user i's two nonces.
S_i : the ith server.
U_i : the ith user.
AE : an attacker.
T_i : user i's current timestamp.
T_s, T_{ss} : *server*'s two current timestamps.
x, y : *server*'s two secret numbers.
SC_i : user i's smart card.

4.1 Registration phase

When user U_i registers to the service provider server S_i, they together perform the following steps which are also shown in Fig.1.

User (U_i)		Server (S_i)
Registration Phase		Choose two nonces r_i, y_i for
Chooses $ID_i, Pw_i, \& \ b$,		each user, then computes
Computes		$G_i = r_i \oplus h(x)$
$RPw_i = h(b \| Pw_i)$		$H_i = y_i \oplus h(y \| r_i)$
	$\{ID_i, RPw_i\}$ →	$N_i = h(ID_i \| x \| y_i) \oplus RPw_i$
		$Y_i = y_i \oplus h(ID_i \| x \| y_i)$,
		$D_i = h(ID_i \| y_i \| RPw_i)$ and
	← $[SC_i = \{G_i, H_i, Y_i, D_i, E_i, h(.)\}$, and $N_i]$	$E_i = y_i \oplus h(y \| x \| y_i)$
Computes		
$A_i = h(ID_i \| Pw_i) \oplus b$		
$W_i = N_i \oplus b$		
Inserts A_i and W_i into SC_i so that		
$SC_i = \{G_i, H_i, Y_i, D_i, E_i, h(.), A_i, W_i\}$		

Fig. 1. The Registration phase.

(1) U_i chooses his identity ID_i, password Pw_i, and selects a random nonce b. He then computes $RPw_i = h(b \| Pw_i)$ and sends $\{ID_i, RPw_i\}$ to S_i over a secure channel.

(2) After receiving the registration message from U_i, S_i chooses two random numbers r_i and y_i, which both are different from all the other users'.

(3) S_i then computes the values $G_i = r_i \oplus h(x)$, $H_i = y_i \oplus h(y \| r_i)$, $E_i = y_i \oplus h(y \| x \| y_i)$, $N_i = h(ID_i \| x \| y_i) \oplus RPw_i$, $Y_i = y_i \oplus h(ID_i \| x \| y_i)$, and $D_i = h(ID_i \| y_i \| RPw_i)$.

(4) S_i stores the values $\{G_i, H_i, Y_i, D_i, E_i, h(.)\}$ into U_i's smart card (SC_i), and then delivers $\{SC_i$ and $N_i\}$ to U_i via a secure channel.

(5) After receiving the message from S_i, U_i computes $A_i = (ID_i \| Pw_i) \oplus b$, $W_i = N_i \oplus b$, and inserts them into SC_i which now contains the parameters $\{G_i, H_i, Y_i, D_i, E_i, h(.), A_i$ and $W_i\}$. U_i hereafter needs not remember the random number b anymore.

From the above-mentioned, we know that we add only two values G_i, H_i and replace E_i with $y_i \oplus h(y \| x \| y_i)$, where $h(y \| x \| y_i)$ is also used in the session key generation. The others are the same as in the original scheme.

4.2 Login and authentication phase

This phase is to enable a user to access the needed resources from a server. First, U_i inserts his SC_i into a card reader and inputs his username ID_i and password Pw_i. SC_i then verifies its owner with the secret data stored by using the following steps which are also shown in Fig.2.

(1) First, SC_i computes $b = A_i \oplus (ID_i \| Pw_i)$, $RPw_i = h(b \| Pw_i)$, $h(ID_i \| x \| y_i) = W_i \oplus RPw_i \oplus b$, and $y_i = Y_i \oplus h(ID_i \| x \| y_i)$. It then computes $D_i^* = h(ID_i \| y_i \| RPw_i)$.

(2) SC_i verifies to see whether the equation $D_i^* = D_i$ holds, if it does not hold, SC_i drops the session and U_i is required to enter PUK (*Private Unblocking Key*) to re-activate his SC_i.

(3) Only if $D_i^* = D_i$ holds, SC_i authenticates its owner and proceeds further. It computes $h(y \| x \| y_i) = y_i \oplus E_i$, $N_i = W_i \oplus b$, $CID_i = ID_i \oplus h(N_i \| y_i \| T_i)$, $N_i' = N_i \oplus h(y_i \| T_i)$, $C_i = h(N_i \| y_i \| B_i \| T_i)$, and where T_i is SC_i's current timestamp.

(4) SC_i then transfers the login request = $\{G_i, H_i, CID_i, N_i', C_i, T_i\}$ to S_i. After receiving the login request, S_i and U_i together perform the following steps to authenticate each other.

(5) S_i verifies to see whether $(T_s - T_i) < \Delta T$ holds, where T_s is S_i's current timestamp. If it does, S_i computes $r_i = G_i \oplus h(x)$, $y_i = H_i \oplus h(y \| r_i)$, $N_i = N_i' \oplus h(y_i \| T_i)$, and $ID_i = CID_i \oplus h(N_i \| y_i \| T_i)$. It then computes $B_i^* = h(ID_i \| x \| y_i)$, $C_i^* = h(N_i \| y_i \| B_i^* \| T_i)$, and compares C_i^* with C_i.

(6) If $C_i^* = C_i$ holds, S_i confirms the legality of U_i. It then computes $\alpha = h(B_i^* \| y_i \| T_{ss})$, chooses a random r_i', computes $G_i = r_i' \oplus h(x)$, $H_i = y_i \oplus h(y \| r_i')$, $EGH = E_{sk^*}(G_i, H_i)$, $MAC = h(\alpha \| EGH \| y_i)$, and transmits $\{\alpha, T_{ss}, EGH, MAC\}$ to SC_i,

where T_{ss} is the server's current timestamp, $E_{sk^*}(G_i, H_i)$ denotes the encryption of (G_i, H_i) using session key $sk^* = h(B_i^* \| y_i \| T_i \| T_{ss} \| h(y \| x \| y_i))$.

(7) On receiving $\{\alpha, T_{ss}, EGH, MAC\}$, SC_i checks T_{ss}'s freshness. If T_{ss} is fresh, SC_i

User (U_i)		Server (S_i)
Login and Authentication Phase		**Authentication Phase**
U_i : Inserts ID_i, Pw_i		For $(T_s - T_i) < \Delta T$, then
SC : Computes		Computes
$b = A_i \oplus h(ID_i \| Pw_i)$,		
$RPw_i = h(b \| Pw_i)$,		$r_i = G_i \oplus h(x)$,
$h(ID_i \| x \| y_i) = W_i \oplus RPw_i \oplus b$,		$y_i = H_i \oplus h(y \| r_i)$
$y_i = Y_i \oplus h(ID_i \| x \| y_i)$,		$N_i = N_i' \oplus h(y_i \| T_i)$.
$D_i^* = h(ID_i \| y_i \| RPw_i)$		$ID_i = CID_i \oplus h(N_i \| y_i \| T_i)$.
If $D_i^* = D_i$, (otherwise to enter PUK)		$B_i^* = h(ID_i \| x \| y_i)$ and
Computes		$C_i^* = h(N_i \| y_i \| B_i^* \| T_i)$.
$h(y \| x \| y_i) = y_i \oplus E_i$		If $C_i^* = C_i$, Computes
$N_i = W_i \oplus b$,		$\alpha = h(B_i^* \| y_i \| T_{ss})$, and
$CID_i = ID_i \oplus h(N_i \| y_i \| T_i)$,		chooses a random r_i', then
$N_i' = N_i \oplus h(y_i \| T_i)$,		Computes
$B_i = N_i \oplus RPw_i = h(ID_i \| x \| y_i)$,		$G_i = r_i' \oplus h(x)$, $H_i = y_i \oplus h(y \| r_i')$;
$C_i = h(N_i \| y_i \| B_i \| T_i)$,	$\{G_i, H_i, CID_i, N_i', C_i, T_i\}$ →	$sk^* = h(B_i^* \| y_i \| T_i \| T_{ss} \| h(y \| x \| y_i))$
		$EGH = E_{sk^*}(G_i, H_i)$;
		$MAC = h(\alpha \| EGH \| y_i)$
	← $\{\alpha, Tss, EGH, MAC\}$	
For fresh T_{ss}, SC_i computes $\alpha^* = h(B_i \| y_i \| T_{ss})$.		
If $\alpha^* = \alpha$, U_i regards S_i as authentic.		
If $MAC = h(\alpha^* \| EGH \| y_i)$		
Computes $sk = h(B_i \| y_i \| T_i \| T_{ss} \| h(y \| x \| y_i))$		
Decrypts EGH, obtaining G_i, H_i		
Replaces the old G_i, H_i in SC_i.		

Fig. 2. The Login and the Authentication.

computes $\alpha^* = h(B_i \| y_i \| T_{ss})$ and verifies to see whether $\alpha^* = \alpha$ holds. If it holds, SC_i confirms the legality of the server. It then computes $MAC = h(\alpha^* \| EGH \| y_i)$ and compares it with the received one to see if they are equal.

(8) If they are, then SC_i computes the common session key sk as $h(B_i \| y_i \| T_i \| T_{ss} \| h(y \| x \| y_i))$.

(9) It decrypts EGH, obtaining the newer G_i, H_i and then uses these two items to replace the old two stored in the smard card.

4.3 Password change phase

In this phase, we only replace $h(ID_i \| x)$ with $h(ID_i \| x \| y_i)$ and refresh the parameters which are directly or indirectly related to Pw_i, *e.g.*, A_i, W_i, and D_i, as shown in Fig.3. The others are the same as in the original scheme.

User (U_i)		Smart Card (SC_i)
Password Change Phase		$b = A_i \oplus (ID_i \| Pw_i)$, $RPw_i = h(b \| Pw_i)$,
U_i : Inserts ID_i, Pw_i	$\{ID_i, Pw_i\}$ →	$h(ID_i \| x \| y_i) = W_i \oplus RPw_i \oplus b$,
		$y_i = Y_i \oplus h(ID_i \| x \| y_i)$,
		$D_i^* = h(ID_i \| y_i \| RPw_i)$. If $D_i^* = D_i$, allows
		U_i to enter new password
	$\{Pw_i\}_{new}$ →	Computes $(RPw_i)_{new} = h(b \| (Pw_i)_{new})$,
		$(A_i)_{new} = (ID_i \| (Pw_i)_{new}) \oplus b$,
		$(W_i)_{new} = W_i \oplus (Rpw_i) \oplus (RPw_i)_{new}$
		$(D_i)_{new} = h(ID_i \| y_i \| (RPw_i)_{new})$.
	refresh parameters →	$A_i = (A_i)_{new}$, $W_i = (W_i)_{new}$, and $D_i = (D_i)_{new}$

Fig. 3. The Password Change Phase.

5. Security analysis

Compared with the orginal scheme, we can see that without the knowledge of server's secrets x and y, an insider cannot compute the value of $h(y \| x \| y_i)$ to breach the anonymity property, due to the one-way hash function and the unknown value of y_i. Hence, the insider attack fails. As for the lost card password guessing attack, even if an insider obtains a lost card and knows all the parameters stored, however, without the knowledge of y, y_i, b and ID_i, from the descriptions of Session 3.2, we can easily see that he cannot launch a password guessing attack. Therefore, both attacks existed in the original scheme have been resolved. Moreover, the newly generated G_i, H_i by S_i can not be altered by any attacker, because they are protected via the parameter MAC which must pass U_i's verification by checking wether $MAC = h(\alpha^* \| EGH \| y_i)$ holds or not. Only if the equation holds, U_i can decrypt EGH to obtain the newly generated G_i, H_i for replacing the two old ones stored in the smart card.

After describing the reasons why our improvements can eliminate the weaknesses found in Kumari *et al*.'s scheme, in the following, we go a step further to demonstrate that why it can also satisfy the ten security requirements of a remote user authentication scheme, proposed by Liao *et al*. [12].

5.1 The user password is not stored on the server.

Our scheme requires no verifier tables stored on the server side. Hence, it meets the requirement.

5.2 The user can freely choose / change the password.

In our modification, we let the smart card authenticate the user by checking to see whether the equivalence $D_i^* = D_i$ holds before the password change. If it does, that means the smart card regards the user as authentic. This guarantees that only the real card holder can safely and freely choose / change the password.

5.3 The password cannot be revealed by the administrator of the server.

From Fig.1 and Fig.2, we can see that the user's password Pw_i has never been revealed to the server in the registration phase, and the login and authentication phase. Thus, this goal can be achieved.

5.4 The user password is not transmitted in plain form over the internet.

In the registration and password change phases, both pairs (U_i, S_i, and U_i, SCi) communicate over a secure channel. Therefore, we only need take login and authentication into consideration. From Fig.2, we can see that the user password Pw_i has never been transmitted in plaintext.

5.5 The scheme can resist the insider attacks.

In our modification, we have introduced a new random y_i for each user, to avoid the insider attack as occured in the original scheme. That is, each user cannot compute the other user's $h(y \| x \| y_i)$, because y_is are all different. Therefore, even if the attacker intercepted the transmitted message, however, without the *knowledy* of y_i, he cannot launch an insider attack. Not to mention, he doesn't know the values of x and y.

5.6 The scheme can resist the replay, modification-verifier-table, and stolen-verifier attacks.

Our scheme requires no verifier table on the server side, thus it can resist the modification-table attack and stolen-verifier attack. In addition, when server S_i receives the login request message $\{G_i, H_i, CID_i, N_i', C_i, T_i\}$ from U_i, it instantaneously checks whether the received T_i is a valid timestamp. Likewise, the freshness T_{ss} in the response message $\{\alpha, T_{ss}, EGH, MAC\}$ transmitted from S_i to U_i also undergoes U_i's verification. Thus, the replay attack on our scheme could not be fulfiled successfully.

5.7 The length of a password is appropriate for memorization.

In our scheme, Pw_i is embeded in $RPw_i = h(b \| Pw_i)$, and then used to generate parameters N_i, D_i, A_i, and D_i* in the registration phase. That is, Pw_i is protected by both b and the one-way hash function. Hence, our scheme's strength does not rely on the length of the password. The user, therefore, can choose a password of any length for easy memorization.

5.8 The scheme can be efficient and practical.

Our scheme has several advantages that it only demands two passes, requires no complex computations, and makes use of only hash functions and *Xor* operations. Therefore, our scheme is efficient and practical.

5.9 The scheme can achieve mutual authentication.

In our scheme, both the server and the user must confirm each other's identity before generating the common session key. This means that mutual authentication could be achieved. In the following, we first demonstrate that our scheme can achieve this goal and then show why it can resist the man-in-the middle attack (**MIMA**).

(1) Mutual authentication:

In the login and authentication phase, the server has to verify the validity of $C_i = h(N_i \| y_i \| B_i \| T_i)$ to validate the user, and the user must check the validity of $\alpha = h(B_i* \| y_i \| T_{ss})$ to authenticate the server. In other words, after both parties complete these validity checkings, they successfully authenticate each other.

(2) Man-In-the Middle attack:

In the man-in-the-middle attack, an active attacker might intercept a communication between a legal user and the server, and next use some means to successfully masquerade as both the server (to the user) and the user (to the server). The user will then believe that he is talking to the intended server, and vice versa. But indeed, this is not the case.

We now describe what happens when **MIMA** is launched on our login and authentication protocol, as shown in Figure 4. Assuming that after intercepting the communcation message $\{G_i, H_i, CID_i, N_i', C_i, T_i\}$ between the server and the user, the attacker AE then impersonates the user by sending $\{G_i', H_i', CID_i', N_i'', C_i', T_i'\}$ to the real server, and later after receiving $\{\alpha, T_{ss}, EGH, MAC\}$ from the real server, he masquerades as the server by sending $\{\alpha', T_{ss}', EGH', MAC'\}$ to the user. If the server can successfully verify C_i', and the user can succeed in confirming α', AE will then be regarded as authentic by them both, and will have the two common session keys shared by the user and the server, respectively.

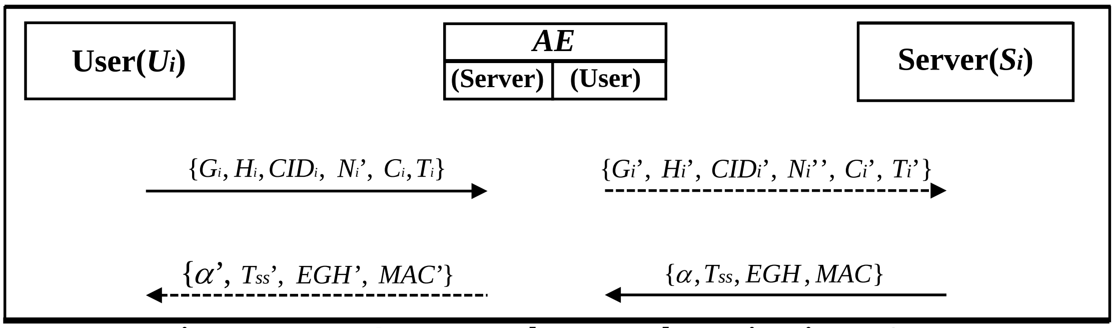

Figure 4. **MIMA on our scheme as shown in Figure 2**.

However, in order to verify C_i' the server should compute $C_i^* = h(N_i \,||\, y_i \,||\, B_i^* \,||\, T_i)$, where $N_i = N_i'' \oplus h(y_i \,||\, T_i')$, $B_i^* = h(ID_i \,||\, x \,||\, y_i)$. But without the knowledge of x, y_i, AE cannot compute N_i, ID_i, B_i^* to send valid C_i'. Similaly, to verify α' the user should compute $\alpha^* = h(B_i \,||\, y_i \,||T_{ss}')$, where $B_i = h(ID_i \,||\, x \,||\, y_i)$. Nevertheless, from the equations, $ID_i = CID_i \oplus h(N_i \,||\, y_i \,||\, T_i)$ and $N_i = N_i' \oplus h(y_i \,||\, T_i)$, we know that AE should know y_i to confirm ID_i. Yet, even if AE has the value of y_i, he cannot send a genuine α' without the knowledge of x. Hence, the **MIMA** fails.

5.10 Even if the smart card is lost, it can resist the password guessing attack.

An attacker AE might launch various attacks when he obtains a user's smart card. Under such a situation, we discuss the most common attack, the offline password guessing attack, to demonstrate why our scheme can eliminate such a defect. We show it in two cases: (1) the user's smart card is obtained by AE after registration, and (2) the card is obtained after the login and authentication phase.

(1) Supposing the user's smard card is obtained by AE after registration.

Although AE can read the stored values $\{G_i, H_i, Y_i, D_i(= h(ID_i \,||\, y_i \,||\, RPw_i)), E_i, h(.), A_i(= h(ID_i \,||\, Pw_i) \oplus b), W_i\}$, however, without the knowledge of y_i, ID_i, and b, he cannot confirm whether his guessed password is correct or not. Therefor, he cannot launch an offline password guessing attack on a lost card. For instance, AE might guess password Pw_i as Pw_{AE}; yet, without the knowledge of values ID_i and b, AE cannot to confirm the validity of his guessing.

(2) The card is obtained by AE after the login and authentication phase.

Even with the related parameters, $ID_i = CID_i \oplus h(N_i \,||\, y_i \,||\, T_i)$, $N_i = N_i' \oplus h(y_i \,||\, T_i)$, $W_i = h(ID_i \,||\, x \,||\, y_i) \oplus RPw_i \oplus b$, $D_i = h(ID_i \,||\, y_i \,||\, RPw_i)$, and $N_i' = (h(ID_i \,||\, x \,||\, y_i) \oplus RPw_i) \oplus h(y_i \,||\, T_i)$, where $RPw_i = h(b \,||\, Pw_i)$, AE has no advantage in deducing any helpful information about user's password to examine his guessing. Because he still needs to know x, y_i, b to confirm $W_i = h(ID_i \,||\, x \,||\, y_i) \oplus h(b \,||\, Pw_i) \oplus b$, $N_i' = W_i \oplus b \oplus h(y_i \,||\, T_i)$, and y_i, b to verify $D_i = h(ID_i \,||\, y_i \,||\, h(b \,||\, Pw_i))$. As a result, we conclude that AE cannot succeed.

6. Conclusion

In this paper, we showed that Kumari *et al.*'s scheme is flawed, because it suffers from (1). the smart card loss password guessing attack, and (2). anonymity breach. We, therefore, modified the scheme to avoid these weaknesses. From the analysis shown in Session 5, we can see that our method not only corrected the security issues of the original scheme but also satisfied the ten security requirements of a remote user authentication protocal using smart card which was insisted by Liao *et al*.

References

[1] C. T. Li and M. S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications,* vol. 33, pp. 1-5, 2010.

[2] W. C. Kuo, H. J. Wei, and J. C. Cheng, "An efficient and secure anonymous mobility network authentication scheme," *Journal of Information Security and Applications,* vol. 19, pp. 18-24, 2014.

[3] J. S. Chou and Y. Chen, "An Efficient Two-Pass Anonymous Identity Authentication Protocol Using a Smart Card," *Jökull Journal,* vol. 63, no. 8, 2013.

[4] D. Wang and P. Wang, "Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks," *Ad Hoc Networks,* vol. 20, pp. 1-15, 2014.

[5] D. Wang, N. Wang, P. Wang, and S. Qing, "Preserving privacy for free: Efficient and provably secure two-factor authentication scheme with user anonymity," *Information Sciences,* vol. 321, pp. 162-178, 2015.

[6] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Networks,* vol. 20, pp. 96-112, 2014.

[7] K. Xue, P. Hong, and C. Ma, "A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture," *Journal of Computer and System Sciences,* vol. 80, pp. 195-206, 2014.

[8] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions," *Computer Networks,* vol. 73, pp. 41-57, 2014.

[9] C. T. Li and C. C. Lee, "A novel user authentication and privacy preserving scheme with smart cards for wireless communications," *Mathematical and Computer Modelling,* vol. 55, pp. 35-44, 2012.

[10] M. S. Farash, M. Turkanović, S. Kumari, and M. Hölbl, "An efficient user

authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Networks,* vol. 36, Part 1, pp. 152-176, 2016.

[11] C. Li, U. T. Nguyen, H. L. Nguyen, and N. Huda, "Efficient authentication for fast handover in wireless mesh networks," *Computers & Security,* vol. 37, pp. 124-142, 2013.

[12] I. E. Liao, C. C. Lee, and M. S. Hwang, "A password authentication scheme over insecure networks," *Journal of Computer and System Sciences,* vol. 72, pp. 727-740, 2006.

[13] S. Kumari, M. K. Khan, and X. Li, "An improved remote user authentication scheme with key agreement," *Computers & Electrical Engineering,* vol. 40, pp. 1997-2012, 2014.

[14] Y. F. Chang, W. L. Tai, and C. Hung-Chin, "Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update," *International Journal of Communication Systems,* vol. 27, pp. 3430-3440, 2014.

[15] M. C. Chuang and M. C. Chen, "An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics," *Expert Systems with Applications,* vol. 41, pp. 1411-1418, 2014.

[16] M. Karuppiah and R. Saravanan, "A secure remote user mutual authentication scheme using smart cards," *Journal of Information Security and Applications,* vol. 19, pp. 282-294, 2014.

[17] D. Mishra, A. K. Das, and S. Mukhopadhyay, "A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards," *Expert Systems with Applications,* vol. 41, pp. 8129-8143, 2014.

[18] A. K. Das and A. Goswami, "A robust anonymous biometric-based remote user authentication scheme using smart cards," *Journal of King Saud University - Computer and Information Sciences,* vol. 27, pp. 193-210, 2015.

[19] V. Odelu, A. K. Das, and A. Goswami, "An efficient ECC-based privacy-preserving client authentication protocol with key agreement using smart card," Journal of Information Security and Applications, vol. 21, pp. 1-19, 2015.

附錄一

Improved on an improved remote user authentication scheme with key agreement

Yalin Chen[1] and Jue-Sam Chou*[2] and I - Chiung Liao[3]

[1] Institute of information systems and applications, National Tsing Hua University

[2] Department of Information Management, Nanhua University, Taiwan

[3] Department of Information Management, Nanhua University, Taiwan

Abstract

Recently, Kumari et al. pointed out that Chang et al.'s scheme "Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update" not only has several drawbacks, but also does not provide any session key agreement. Hence, they proposed an improved remote user authentication Scheme with key agreement on Chang et al.'s Scheme. After cryptanalysis, they confirm the security properties of the improved scheme. However, we determine that the scheme suffers from both anonymity breach and the smart card loss password guessing attack, which are in the ten basic requirements in a secure identity authentication using smart card, assisted by Liao et al. Therefore, we modify the method to include the desired security functionality, which is significantly important in a user authentication system using smart card.

Keywords: user authentication, key agreement, cryptanalysis, smart card, password change, untraceable, dynamic identity, anonymity, remote user authentication

1. Introduction

There have been many cryptographic scientists working within the field of remote user authentication using smart card system design [1-21]. A user authentication using smart card system typically contains two roles: the user and the server; and three protocols: registration, login and authentication, and password change. In the protocol design principle, to ensure the login privacy, it cannot reveal the user's identity. In

2014, Kumari et al. [14] pointed out that Chang et al.'s scheme [15] has some shortcomings: (1). offline password guessing attack, (2). impersonation attacks, (3). insider attack, (4). anonymity breach when the smart card is obtained by a legal user, (5). It sufferers from the denial of service attack, and (6). It doesn't provide session key agreement. Hence, they overcome the security weaknesses by proposing a new one with key agreement. It provides user anonymity, establishes proper mutual authentication, and offers a secure password change phase, without maintaining any database record at the server side. They claimed that the proposed scheme resists various attacks, including those existing in Chang et al.s', and outperforms six other related schemes in the aspect of security characteristics. However, upon a closer examination, we discovered that it suffers from the security weaknesses of (1) anonymity breach, and (2) the smart card loss password guessing attack. To enhance its security, we modified their scheme to include these features. We will demonstrate the enhancement in this article.

The rest of this article is organized as follows. In Section 2, we briefly introduce Kumari et al.'s Scheme. In Section 3, we analyze the weaknesses of the scheme. The modifications and the security issues are demonstrated and discussed in Section 4 and 5, respectively. Finally, a conclusion is given in Section 6.

2. Review of Kumari et al.'s scheme

Kumari et al.'s improved remote user authentication Scheme with key agreement is based on Chang et al.'s Scheme [15]. It also consists of two roles: user and the remote server; and the phases: registration, login, authentication, and password change phase. They claimed that their scheme not only tackles and eliminates all security shortcomings and vulnerabilities of Chang et al.'s Scheme, but also introduces the session key agreement. In this article, we only review the registration phase, and login and authentication phase to illustrate its weaknesses. As for the definitions of the used notations, please refer to the original article.

2.1 Registration Phase

When a user Ui registers to the service provider server Si, this phase is performed as follows:

(1) The user U_i chooses its identity ID_i, password PW_i, and selects a random nonce b. He then computes $RPW_i= h(b \| PW_i)$ and sends $\{ID_i, RPW_i\}$ to Si over a secure channel.

(2) After receiving the registration message from U_i, Si chooses a random number yi, which is different for each user.

(3) Si computes the value $N_i = h(ID_i \| x) \oplus RPW_i$, $Y_i = y_i \oplus h(ID_i \| x)$, $D_i =$

$h(ID_i||y_i||RPw_i)$ and $E_i = y_i \oplus h(y||x)$

(4) Si stores the values $\{Y_i, D_i, E_i, h(.)\}$ into U_i's smart card SC_i for and delivers $\{SC_i$ and $N_i\}$to U_i via a secure channel.

(5)After receiving the message from SC_i, U_i computes $A_i =(ID_i||Pw_i)\oplus b$ and $M_i = N_i \oplus b$, inserts A_i and M_i into SC_i which now contains the parameters $\{Y_i, D_i, E_i, h(.), A_i$ and $M_i\}$. U_i needs not remember the random number b anymore.

2.2 Login phase

This phase is to enable a user to access the needed resources from a server. U_i inserts his SC_i into a card reader and inputs its username ID_i and password PW_i. The SCi then verifies the owner of the SC_i with the secret data stored in it.

(1) First, the SC_i computes $b = A_i \oplus (ID_i||Pw_i)$, $RPw_i = h(b||Pw_i)$, $h(ID_i||x)= M_i \oplus RPw_i \oplus b$, and $y_i = Y_i \oplus h(ID_i||x)$. He then computes $D_i^*= h(ID_i||y_i||RPw_i)$

(2) SC_i verifies whether the equation $D_i^*= D_i$ holds, if it does not hold, SC_i drops the session. And U_i is required to enter PUK (Private Unblocking Key) to re-activate his SC_i

(3) Only if $D_i^*= D_i$ holds, SCi proceeds further. it computes $h(y||x)= y_i \oplus E_i$, $N_i = M_i \oplus b$, $CID_i = ID_i \oplus h(N_i||y_i||T_i)$, $N_i' = N_i \oplus h(y_i||T_i)$, $B_i = N_i \oplus RPw_i =h(ID_i||x)$, $C_i = h(N_i||y_i||B_i||T_i)$ and $F_i = y_i \oplus (h(y||x)||T_i)$, where T_i is the system's current timestamp T_i.

(4) SC_i transfers the login request = $\{CID_i, N_i', C_i, F_i, T_i\}$ to S_i.

2.3. Authentication phase

After receiving the login request, S_i and U_i together perform the following steps to authenticate each other:

(1) S_i verifies to see whether $(T_s - T_i) < \triangle T$ holds, where Ts is the current timestamp. If it does, S_i retrieves $y_i = F_i \oplus (h(y||x)||T_i)$, $N_i = N_i' \oplus h(y_i||T_i)$ and $ID_i = CID_i \oplus h(N_i||y_i||T_i)$. It then computes $B_i^*= h(ID_i||x)$, $C_i^*= h(N_i||y_i||B_i^*||T_i)$ and compares C_i^* with C_i.

(2) If $C_i^*=C_i$ holds, S_i confirms the legality of U_i. It then computes $a = h(B_i^*||y_i||Tss)$ and transmits $\{a, T_{ss}\}$ to SC_i, where T_{ss} is the server's current timestamp.

(3) On receiving $\{a, T_{ss}\}$, SC_i checks T_{ss} for freshness. If T_{ss} is fresh, SC_i computes $a^*= h(B_i||y_i||T_{ss})$ and verifies to see whether $a^*= a$ holds. If it holds, SC_i confirms the legality of the server.

(4) After successful mutual authentication, U_i and S_i both compute the common session key as $Sessk = h(B_i||y_i||T_i||T_{ss}||h(y||x))$ and $(Sessk)= h(B_i^*||y_i||T_i||T_{ss}||h(y||x))$ respectively.

3. Weakness of the scheme

Due to the parameters {Y_i, D_i, E_i, h(.), A_i and M_i} stored in the smart card and the user himself can compute the $b = A_i \oplus (ID_i||Pw_i)$, $RPw_i = h(b||Pw_i)$, $h(ID_i||x)= M_i \oplus RPw_i \oplus b$, and $y_i = Y_i \oplus h(ID_i||x)$, an insider can compute his own $h(y||x)= y_i \oplus E_i$. That is, each user can know the value $h(y||x)$. Under this situation, we can see that their scheme suffers from: (1) Anonymity breach, (2) The smart card loss password guessing attack. We describe them below.

(1) The insider attacks on the protocol's anonymity property

If a user Bob's login request {CID_i, N_i', C_i, F_i, T_i}, transferred to S_i, is intercepted by an insider attacker Alice, Alice can know Bob's y_i by calculating $y_i=F_i\oplus (h(y||x)||T_i)$. He then computes $ID_i = CID_i \oplus h(N_i||y_i||T_i)$. That is, Alice obtains the user's ID_i, which now is Bob. Therefore, the attack succeeds.

(2) The smart card loss password guessing attack

From the collected login request messages {CID_i, N_i', C_i, F_i, T_i} and from the equations $y_i=F_i\oplus(h(y||x)||T_i)$ and $h(y||x)= y_i\oplus E_i$, the insider Alice can calculate the corresponding E_is of each login request by computing $E_i = y_i \oplus h(y||x)$. Therefore, once Bob, who has ever loggined to the server, loses his smart card and obtained by Alice, then from comparing the value E_i stored in the lost card with the calculated corresponding E_is. Alice can identify which intercepted login request is Bob's own. After obtaining the knowledge of Bob's ID_i, and the stored values A_i, D_i, Alice can successfully launch a smart card loss password guessing attack as follows.

The insider first guesses the lost card owner's password as pw_i'. He then computes $b'= A_i\oplus(ID_i||pw_i')$, $RPw_i'= h(b'||pw_i')$, and $D_i'= h(ID_i||y_i||RPw_i')$. Obviously, we can see that if $D_i'= D_i$, then pw_i' is Bob 's password. Therefore, the attack succeeds.

4. Modification

From the weaknesses found in Section 3, we note that the key point is the insider can obtain the value $h(y||x)$. To disguise it, we modify the messages in the registration phase and the login and authentication phases as follows.

4.1 Registration phase

When a user Ui registers to the service provider server Si, they perform the following steps:

(1) The user U_i chooses its identity ID_i, password PW_i, and selects a random nonce b. He then computes $RPW_i= h(b || PW_i)$ and sends {ID_i, RPW_i} to S_i over a secure channel.

(2) After receiving the registration message from U_i, S_i chooses two random number r_i,

y_i, which are different for each user.

(3) S_i computes the values $G_i=r_i \oplus h(x)$, $H_i = y_i \oplus h(y|| r_i)$, $E_i = y_i \oplus h(y||x||y_i)$, $W_i = y_i \oplus RPW_i$, $N_i = h(ID_i || x) \oplus RPW_i$, $Y_i = y_i \oplus h(ID_i||x)$, and $D_i = h(ID_i||y_i||RPw_i)$

(4) Si stores the values { G_i, H_i, W_i, Y_i, D_i, E_i, h(.)} into U_i's smart card SC_i for and delivers {SC_i and N_i}to U_i via a secure channel.

(5)After receiving the message from SC_i, U_i computes $A_i =(ID_i||Pw_i) \oplus b$ and $M_i = N_i \oplus b$, inserts Ai and Mi into SC_i which now contains the parameters { G_i, H_i, W_i, Y_i, D_i, E_i, h(.), A_i and M_i}. U_i needs not remember the random number b anymore.

From the above-mentioned, we know that we add three values G_i, H_i, W_i and replace E_i with $y_i \oplus h(y||x|| y_i)$. The others are the same to the original scheme.

4.2 Login and authentication phase

This phase is to enable a user to access the needed resources from a server. U_i inserts his SC_i into a card reader and inputs its username ID_i and password PW_i. The SCi then verifies the owner of the SC_i with the secret data stored in it.

(1) First, the SC_i computes $b = A_i \oplus (ID_i||Pw_i)$, $RPw_i = h(b||Pw_i)$, $h(ID_i||x)= M_i \oplus RPw_i \oplus b$, and $y_i = Y_i \oplus h(ID_i||x)$. He then computes $D_i{*}= h(ID_i||y_i||RPw_i)$

(2) SC_i verifies whether the equation $D_i{*}= D_i$ holds, if it does not hold, SC_i drops the session. In addition, U_i is required to enter PUK (Private Unblocking Key) to re-activate his SC_i

(3) Only if $D_i{*}= D_i$ holds, SC_i proceeds further. it computes $y_i = W_i \oplus RPW_i$, $h(y||x||y_i)= y_i \oplus E_i$, $N_i = M_i \oplus b$, $CID_i = ID_i \oplus h(N_i||y_i||T_i)$, $N_i' = N_i \oplus h(y_i||T_i)$, $B_i = N_i \oplus RPw_i =h(ID_i||x)$, $C_i = h(N_i||y_i||B_i||T_i)$ and $F_i = y_i \oplus (h(y||x||y_i)||T_i)$, where T_i is the system's current timestamp T_i.

(4) SC_i transfers the login request = { G_i, H_i, CID_i, N_i', C_i, F_i, T_i} to S_i.

4.3. Authentication phase

After receiving the login request, S_i and U_i together perform the following steps to authenticate each other:

(1) S_i verifies to see whether $(T_s - T_i) < \triangle T$ holds, where Ts is the current timestamp. If it does, S_i computes $r_i = G_i \oplus h(x)$, $y_i = H_i \oplus h(y|| r_i)$. Then, calculates $h(y||x||y_i)$ to retrieve $y_i = F_i \oplus (h(y||x||y_i)||T_i)$, $N_i = N_i' \oplus h(y_i||T_i)$ and $ID_i = CID_i \oplus h(N_i||y_i||T_i)$. It then computes $B_i{*}= h(ID_i||x)$, $C_i{*}= h(N_i||y_i||B_i{*}||T_i)$ and compares $C_i{*}$ with C_i.

(2) If $C_i{*}=C_i$ holds, S_i confirms the legality of U_i. It then computes $a = h(B_i{*}||y_i||Tss)$ and transmits {a, T_{ss}} to SC_i, where T_{ss} is the server's current timestamp.

(3) On receiving {a, T_{ss}}, SC_i checks T_{ss} for freshness. If T_{ss} is fresh, SC_i computes

$a^* = h(B_i||y_i||T_{ss})$ and verifies to see whether $a^* = a$ holds. If it holds, SC_i confirms the legality of the server.

(4) After successful mutual authentication, U_i and S_i both compute the common session key as $Sessk = h(B_i||y_i||T_i||T_{ss}||h(y||x))$ and $(Sessk) = h(B_i^*||y_i||T_i||T_{ss}||h(y||x))$ respectively.

5. Security analysis

After the above modification, we can see that without the knowledge of server's secrets x and y, an insider cannot compute the value of $h(y||x||y_i)$ due to the one-way hash and the unknown value of y_i. Hence, the insider attack fails. About the lost card password guessing attack, even if an insider obtains a lost card and knows all the parameters stored, however, without the knowledge of y, y_i, b and ID_i, he cannot launch a password guessing attack. Therefore, both attacks in the original article have been resolved.

6. Conclusion

In this paper, we showed that Kumari et al.'s Scheme's Scheme is flawed, because it suffers from (1). The smart card loss password guessing attack, and (2). Anonymity breach. We, therefore, modify the Scheme to avoid these weaknesses. From the analysis shown in Section 5, we see that we have corrected the security issues.

References

[1] Chun-Ta Li, Min-Shiang Hwang , "An efficient biometrics-based remote user authentication Scheme using smart cards", Journal of Network and Computer Applications, Volume 33, Issue 1, January 2010, Pages 1–5

[2] Wen-Chung Kuo, Hong-Ji Wei, Jiin-Chiou Cheng, "An efficient and secure anonymous mobility network authentication Scheme", journal of information security and applications 19 (2014) 18-24

[3] Jue-Sam Chou, Yalin Chen, "An Efficient Two-Pass Anonymous Identity Authentication Protocol Using a Smart Card", Vol 63, No. 8;Aug 2013

[4] Ding Wang, Ping Wang, "Understanding security failures of two-factor authentication Schemes for real-time applications in hierarchical wireless sensor networks", Ad Hoc Networks 20 (2014) 1–15

[5] "Preserving privacy for free: Efficient and provably secure two-factor authentication Scheme with user anonymity", Ding Wang, Nan Wang b, Ping Wang, Sihan Qing, Information SCiences 321 (2015) 162–178

[6] Muhamed Turkanovic´, Boštjan Brumen, Marko Hölbl, "A novel user

authentication and key agreement Scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion”, Ad Hoc Networks 20 (2014) 96–112
[7] Kaiping Xue, Peilin Hong, Changsha Ma, “A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture”, Journal of Computer and System SCiences 80 (2014) 195–206
[8] Ding Wang, Ping Wang, “On the anonymity of two-factor authentication Schemes for wireless sensor networks: Attacks, principle and solutions” Computer Networks 73 (2014) 41–57
[9] Chun-Ta Li, Cheng-Chi Lee , “A novel user authentication and privacy preserving Scheme with smart cards for wireless communications”, Mathematical and Computer Modelling 55 (2012) 35–44
[10] Ding Wang, Ping Wang,“Understanding security failures of two-factor authentication Schemes for real-time applications in hierarchical wireless sensor networks”, Ad Hoc Networks 20 (2014) 1–15
[11] Mohammad Sabzinejad Farasha, Muhamed Turkanovic, Saru Kumaric,Marko Hölblb,“An efficient user authentication and key agreement Scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment” Ad Hoc Networks 36 (2016) 152–176
[12] Celia Li, Uyen Trang Nguyen, Hoang Lan Nguyen, Nurul Huda, “Efficient authentication for fast handover in wireless mesh networks”, computers & securit y 37(2013) I 24 -I 42
[13] I-En Liao, Cheng-Chi Lee, Min-Shiang Hwang, “A password authentication Scheme over insecure networks”, Journal of Computer and System SCiences, Vol. 72, No. 4, pp. 727-740, 2006.
[14] Kumari, Saru, Muhammad Khurram Khan, and Xiong Li. "An improved remote user authentication Scheme with key agreement." Computers & Electrical Engineering 40.6 (2014): 1997-2012.
[15] Chang, Ya-Fen, Wei-Liang Tai, and Hung-Chin Chang. "Untraceable dynamic-identity-based remote user authentication Scheme with verifiable password update." International Journal of Communication Systems 27.11 (2014): 3430-3440.
[16] M.-C. Chuang and M. C. Chen, "An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics," Expert Systems with Applications, vol. 41, pp. 1411-1418, 2014.
[17] M. Karuppiah and R. Saravanan, "A secure remote user mutual authentication scheme using smart cards," Journal of Information Security and Applications, vol.

19, pp. 282-294, 2014.

[18] D. Mishra, A. K. Das, and S. Mukhopadhyay, "A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards," Expert Systems with Applications, vol. 41, pp. 8129-8143, 2014.

[19] A. K. Das and A. Goswami, "A robust anonymous biometric-based remote user authentication scheme using smart cards," Journal of King Saud University - Computer and Information Sciences, vol. 27, pp. 193-210, 2015.

[20] V. Odelu, A. K. Das, and A. Goswami, "An efficient ECC-based privacy-preserving client authentication protocol with key agreement using smart card," Journal of Information Security and Applications, vol. 21, pp. 1-19, 2015.

[21] D. Wang, N. Wang, P. Wang, and S. Qing, "Preserving privacy for free: Efficient and provably secure two-factor authentication scheme with user anonymity," Information Sciences, 2015.