A Novel Untraceable Authentication Scheme for Mobile Roaming in GLOMONET

Hai-Duong Le[1], Chin-Chen Chang[1,2], and Yeh-Chieh Chou[1]
(Corresponding author: Chin-Chen Chang)

Department of Information Engineering and Computer Science, Feng Chia University[1]
Taichung, 40724, Taiwan, R.O.C.
Department of Computer Science and Information Engineering, Asia University[2]
Taichung, 41354, Taiwan, R.O.C.
(Email: alan3c@gmail.com) *(Received Dec. 22, 2014; revised and accepted Feb. 22 & Mar. 6, 2015)*

Abstract

In global mobile network, it is required to authenticate mobile users, provide secure communication channel between a user and a foreign agent using session key, and guarantee users' anonymity and untraceability. In order to improve the security of mobile roaming service, two-factor authentication which employs smart card and password was introduced to global mobile network. In 2014, Kuo et al. [5] proposed an anonymous two-factor authentication scheme for mobile roaming service. However, we found that this scheme is vulnerable to four kinds of man-in-the-middle attacks and denial-of-service attack. In this paper, we first review Kuo et al.'s scheme and analyze its weaknesses. Then, we propose an efficient anonymous two-factor authentication protocol that overcomes those vulnerabilities in Kuo et al.'s.

Keywords: Anonymity, authentication, GLOMONET, mobile roaming, untraceability

1 Introduction

Mobile telecommunications technology has developed at a rapid pace. The 3G and 4G networks have been deployed all over the world providing mobile users with broadband Internet access. When a user travels from one place to another, the continuity of the mobile service is enforced by the mobile roaming service, which is also known as the global mobility network (GLOMONET) [8, 11]. Along with the great advantages that the mobile networks provide, there are security challenges that we need to overcome. Because of its nature, data transmission in mobile network is susceptible to eavesdropping and interception. Therefore, establishing a secure channel between a user's device and a foreign agent is a must in GLOMONET. Moreover, the personal information of a user, such as his/her identity, location, travelling, Internet accessing habits, etc., should be kept confidential. Thus, both anonymity and untraceability are the characteristics that the global mobile networks must ensure.

In 2004, Zhu and Ma [13] first proposed an anonymous authentication scheme based on hash function for global mobile network. In 2006, Lee et al. [6] pointed out that Zhu-Ma's scheme cannot achieve mutual authentication and perfect backward secrecy, and it is vulnerable to forgery attack. Lee et al. then proposed a scheme to overcome these weaknesses. However, Wu et al. [9] and Chang et al. [2] showed that both Lee et al.'s and Zhu-Ma's schemes failed to ensure anonymity. Later, Youn at al. [12] demonstrated that Chang et al.'s protocol does not provide secure key establishment and anonymous authentication. In 2012, Mun et al. [7] illustrated that Wu et al.'s scheme cannot achieve anonymity and perfect forward secrecy. Recently, Kim and Kwak [4] showed that Mun et al.'s scheme is susceptible to replay attacks and man-in-the-middle attacks.

In 2013, Xie et al. [10] proposed a new anonymous two-factor authentication scheme for roaming service in GLOMONET. In this scheme, a mobile user must possess a smart card and memorize a password in order to authenticate with the mobile network agents. The computation cost of Xie et al.'s scheme is high due to employing both modular exponentiation and symmetric encryption/decryption operations. Furthermore, He et al. [3] found that Xie et al.'s protocol fails to prevent two types of impersonation attack; and they proposed another scheme that resolves these weaknesses.

In 2014, in line with Xie et al.'s two-factor authentication scheme, Kuo et al. [5] proposed an efficient anonymous authentication protocol for mobile roaming service. Kuo et al.'s protocol is more efficient in computing than Xie et al.'s. However, we found that Kuo et al.'s scheme is prone to four kinds of man-in-the-middle attacks and denial-of-service attack. In this paper, we first demonstrate how an attacker can exploit the weaknesses in Kuo et al.'s. Then, we propose a novel and secure anonymous two-factor authentication scheme for global mobile network.

The rest of the paper is organized as follows. Section 2 reviews Kuo et al.'s scheme and illustrates its weaknesses. Our proposed protocol is introduced in Section 3. The security analysis is provided in Section 4. After that, we evaluate the security and performance of our scheme in Section 5. Finally, we conclude the paper in Section 6.

2 Related Work

In this section, we briefly review Kuo et al.'s scheme and demonstrate its weaknesses. At first, we list all the notations used in this paper in Table 1.

Table 1: Notations

MU	The mobile user
ID_{MU}	The identity of MU
FA	The foreign agent
ID_{FA}	The identity of FA
HA	The home agent
ID_{HA}	The identity of HA
pw_{MU}	The password of MU
$\mathcal{A}$	The adversary
SC	The smart card
$h(\cdot)$	The hash operation
P	A point on the elliptic curve
$P.x$	The value of P on x-axis
s	HA's long-term secret
r, r_1, r_2, N_{MU}	Random numbers

2.1 Review of Kuo et al.'s Scheme

There are four phases in Kuo et al.'s scheme: registration phase, authentication and establishment of the session key phase, update session key phase, and password change phase.

2.1.1 Registration Phase

In this phase, MU registers with HA in order to use roaming service. MU and HA execute the following steps:

Step 1. MU chooses an identity ID_{MU} and a password pw_{MU}, and then computes $PW_{MU} = h(ID_{MU}\|pw_{MU})$. It sends $\{ID_{MU}, PW_{MU}\}$ to HA via a secure channel.

$$MU \rightarrow HA : m_{reg} = \{ID_{MU}, PW_{MU}\}.$$

Step 2. HA checks whether ID_{MU} is available for use. If it is, HA chooses a random nonce N_{MU_i} and p_{HA-MU_i}, and computes $U = h(p_{HA-MU_i}\|N_{MU_i})$, $W_i = PW_{MU} \oplus N_{MU_i}$, and $V_i = N_{MU_i} \oplus p_{HA-MU_i}$. It then writes $\{ID_{HA}, W_i, V_i, h(\cdot)\}$ to a smart card and issues it to MU. Finally, HA stores the values U, PW_{MU}, and p_{HA-MU_i} in its database.

$$\begin{aligned} HA &\rightarrow MU : SC = \{ID_{HA}.W_i, V_i, h(\cdot)\}, \\ HA &\rightarrow DB : \{U, PW_{MU}, p_{HA-MU_i}\}. \end{aligned}$$

2.1.2 Authentication and Establishment of the Session Key Phase

In this phase, HA helps FA and MU authenticating each other as follows:

Step 1. MU inserts the smart card into the reader and provides ID_{MU} and pw_{MU}. The smart card chooses a random nonce $N_{MU_{i+1}}$, and derives $PW_{MU} = h(ID_{MU}\|p_{MU})$, $N_{MU_i} = PW_{MU} \oplus W_i$, $p_{HA-MU_i} = N_{MU_i} \oplus V_i$. Then, it computes $S_1 = h(p_{HA-MU_i}\|N_{MU_i})$, $S_2 = PW_{MU} \oplus N_{MU_{i+1}}$, $S_3 = h(N_{MU_{i+1}}\|ID_{FA})$, $S_4 = h(PW_{MU} \oplus h(p_{HA-MU_i}\|N_{MU_{i+1}}))$, and sends $m_1 = \{ID_{HA}, S_1, S_2, S_3, S_4\}$ to FA after saving $N_{MU_{i+1}}$.

$$MU \rightarrow FA : m_1 = \{ID_{HA}, S_1, S_2, S_3, S_4\}.$$

Step 2. FA chooses a random nonce a and computes aP. It sends $\{ID_{FA}, S_1, S_2, S_3, S_4, aP\}$ to HA, and stores ID_{HA}, a, aP.

$$FA \rightarrow HA : m_2 = \{ID_{FA}, S_1, S_2, S_3, S_4, aP\}.$$

Step 3. Upon receiving m_2, HA uses S_1 to search the database and retrieves PW_{MU}, p_{HA-MU_i}. It derives $N_{MU_{i+1}} = S_2 \oplus PW_{MU}$, and computes $S_3' = h(N_{MU_{i+1}}\|ID_{FA})$, and $S_4' = h(PW_{MU} \oplus h(p_{HA-MU_i}\|N_{MU_{i+1}}))$. HA checks whether $S_3' \stackrel{?}{=} S_3$ and $S_4' \stackrel{?}{=} S_4$. If they both hold, HA successfully authenticates MU, and FA; otherwise, it informs FA to terminate the session. Next, it computes $S_5 = h(PW_{MU}\|N_{MU_{i+1}})$, $S_6 = h(ID_{FA}\|ID_{HA}\|S_5)$, and $S_7 = h(aP.x\|S_5)$, and replaces S_1 in the database with $h(p_{HA-MU_i}\|N_{MU_{i+1}})$. It then sends $\{ID_{HA}, S_6, S_7\}$ to FA.

$$HA \rightarrow FA : m_3 = \{ID_{HA}, S_6, S_7\}.$$

Step 4. FA authenticates HA by checking ID_{HA} in its database. If ID_{HA} exists, FA trusts that HA is legitimate and transmits $\{ID_{FA}, S_6, S_7, aP\}$ to MU.

$$FA \rightarrow MU : m_4 = \{ID_{FA}, S_6, S_7, aP\}.$$

Step 5. MU verifies whether $S_6 \stackrel{?}{=} h(ID_{FA}\|ID_{HA}\|S_5)$ and $S_7 \stackrel{?}{=} h(p_{HA-MU_i}\|N_{MU_{i+1}})$. If both equations hold, it chooses a random nonce b, and computes bP, $K_{MF} = h(abP.x)$, and $C_{MF} = h(K_{MF}\|bP.x)$. The smart card updates W_i, V_i with $W_{i+1} = PW_{MU} \oplus N_{MU_{i+1}}$, $V_{i+1} = N_{MU_{i+1}} \oplus p_{HA-MU_i}$, respectively, and stores aP. Then, it sends $\{bP, C_{MF}\}$ to FA.

$$MU \rightarrow FA : m_5 = \{bP, C_{MF}\}.$$

Step 6. FA computes $K_{MF} = h(abP.x)$, and $C'_{MF} = h(K_{MF}\|bP.x)$. It verifies whether $C_{MF} \stackrel{?}{=} C'_{MF}$. If they are equal, FA successfully authenticates MU, and writes C_{MF}, aP into its database.

Eventually, FA and MU mutually authenticate each other and share the session key K_{MF}.

2.1.3 Update Session Key Phase

To update the session key, MU and FA perform the following steps:

Step 1. MU chooses a new random nonce b_i and sends b_iP, C_{MF_i} to FA.

$$MU \rightarrow FA : m_6 = \{b_iP, C_{MF}\}.$$

Step 2. FA checks the existence of C_{MF} in the database. If there is a record of it, FA trusts MU, and retrieves $a_{i-1}P$ from the record. Next, it selects a new random nonce a_i, and computes $K_{MF_{i+1}} = h(a_ib_iP.x)$, $C_{MF_{i+1}} = h(K_{MF_{i+1}}\|b_iP.x)$, and $h_1 = h(C_{MF_{i+1}}\|a_{i-1}P.x)$. It then replaces C_{MF_i} by $C_{MF_{i+1}}$ and $a_{i-1}P$ by a_iP in the database before delivering $\{a_iP, h_1\}$ to MU.

$$FA \rightarrow MU : m_7 = \{a_iP, h_1\}.$$

Step 3. MU computes the new session key $K_{MF_{i+1}} = h(a_ib_iP.x)$, and $C'_{MF_{i+1}} = h(K_{MF_{i+1}}\|b_iP.x)$. It then checks whether $h_1 \stackrel{?}{=} h(C'_{MF_{i+1}}\|a_{i-1}P.x)$. If it holds, MU authenticates FA; otherwise, it terminates the session. At last, it replaces C_{MF_i} with $C_{MF_{i+1}}$, $a_{i-1}P$ with a_iP in the smart card's memory.

2.1.4 Password Change Phase

In this phase, MU changes its password as follows:

Step 1. MU chooses a new password $pw_{MU_{new}}$, and computes $PW_{MU_{new}} = h(ID_{MU}\|pw_{MU_{new}})$, $U = h(p_{HA-MU_i}\|N_{MU_i})$, $h_1 = PW_{MU} \oplus PW_{MU_{new}}$, and $h_2 = h(PW_{MU_{new}}\|p_{HA-MU_i})$. It then transmits U, h_1, h_2 to HA.

$$MU \rightarrow HA : m_8 = \{U, h_1, h_2\}.$$

Step 2. HA uses U to search and retrieves PW_{MU}, p_{HA-MU_i} from the database. Next, it computes $PW'_{MU_{new}} = PW_{MU} \oplus h_1$, $h'_2 = h(PW'_{MU_{new}}\|p_{HA-MU_i})$. It checks whether $h'_2 \stackrel{?}{=} h_2$. If the equation holds, HA replaces PW_{MU} with $PW_{MU_{new}}$. It then computes $h_3 = h(PW_{MU}\|p_{HA-MU_i})$, and sends h_3 to MU.

$$HA \rightarrow MU : m_9 = \{h_3\}.$$

Step 3. MU verifies whether $h_3 \stackrel{?}{=} h(PW_{MU}\|p_{HA-MU_i})$. If it holds, MU updates W_i with $PW_{MU_{new}} \oplus N_{MU_i}$.

2.2 The Weakness of Kou et al.'s Scheme

2.2.1 Man-in-the-middle Attack in Authentication Phase ($MU - FA$)

In the authentication phase, an attacker $\mathcal{A}$ intercepts the messages sending between MU and FA. It forwards the message m_1 from MU to FA. Upon intercepting m_4 from FA to MU, it generates a random nonce c, and computes $K^*_{MF} = h(caP.x)$, $C^*_{MF} = h(K^*_{MF}\|cP.x)$, where aP is in m_4. Then, it sends the message $m^*_5 = \{cP, C^*_{MF}\}$ to FA.

When receiving m^*_5, FA computes $K^*_{MF} = h(caP.x)$, and $C^*_{MF} = h(K_{MF}\|cP.x)$. Since C^*_{MF} equals to C_{MF}, FA trusts that it is in communication with a legitimate mobile user and the shared session key is K_{MF}. At this stage, $\mathcal{A}$ has successfully deceived FA, and it impersonates a valid user to use FA's services.

In this attack, the values K_{MF} and C_{MF} are computed from cP and aP, where aP is sent in plaintext. $\mathcal{A}$ exploits the fact that FA does not verify whether cP really comes from MU or not.

2.2.2 Man-in-the-middle Attack in Update Session Key Phase ($MU - FA$)

In this attack, $\mathcal{A}$ eavesdrops all the messages between MU and FA. When MU sends $m_6 = \{b_iP, C_{MF_i}\}$ to FA, the attacker intercepts this message and generates a new random nonce c. Then, it transmits $m^*_6 = \{cP, C_{MF}\}$ to FA.

Once receiving m^*_6, FA only searches for the record of C_{MF_i} in its database, but it does not verify whether cP comes from MU or not. Therefore, FA will accept $\mathcal{A}$ as a legitimate mobile user, if it finds a match for C_{MF}. It then computes the session key $K_{MF_{i+1}} = h(ca_iP.x)$. At this point, it has succeeded in forging a legitimate MU.

2.2.3 Man-in-the-middle Attack in Authentication Phase ($MU - FA, FA - HA$)

Since the communication channel between FA and HA is not secure, the attacker $\mathcal{A}$ can intercept the message m_2 from FA to HA. It generates a new random nonce c and replaces aP in m_2 by cP. Then, it forwards the modified m_2 to HA. Because HA does not verify whether

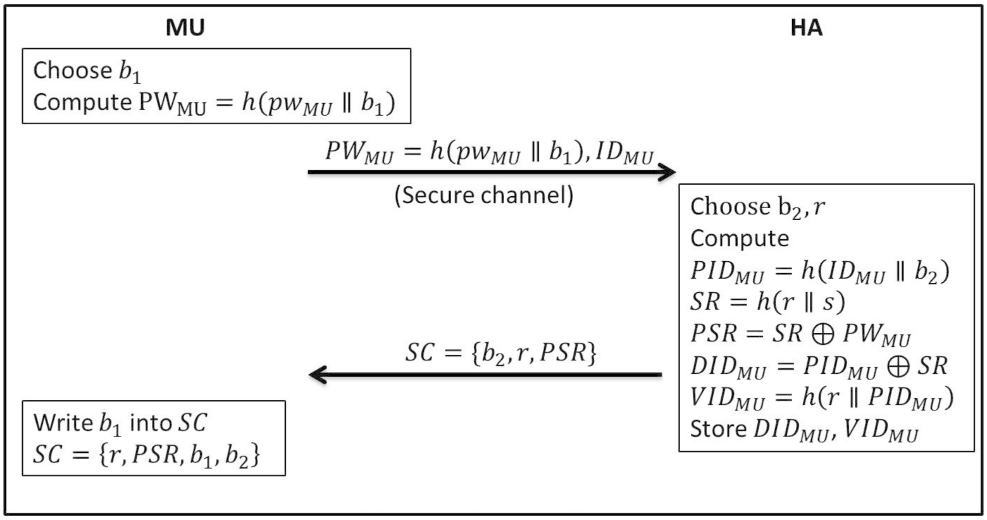

Figure 1: Registration phase

cP comes from FA or not, it will accept MU and compute the value $S_7 = h(cP.x \| S_5)$ based on cP. That means HA has accidentally authenticated cP. $\mathcal{A}$ forwards m_3 from HA to FA. Then, it intercepts m_4 from FA to MU, and replaces aP with cP.

Upon receiving the modified m_4, MU verifies S_6 and S_7; it trusts that FA is valid, and cP comes from FA. Then, MU selects a random nonce b and sends bP in m_5. Now, the attacker intercepts m_5, and computes the session key $K_{MF} = h(caP.x)$. Finally, $\mathcal{A}$ has successfully masqueraded as FA.

2.2.4 Man-in-the-middle Attack in Authentication Phase using Stolen Verifiers

Suppose that the attacker can either steal HA's database or access it. Using $U = h(p_{HA-MU_i} \| N_{MU_i})$, PW_{MU}, and pw_{HA-MU_i}, $\mathcal{A}$ can derives $N_{MU_i} = S_2 \oplus PW_{MU}$, and computes $S_5 = h(PW_{MU} \| N_{MU_{i+1}})$, $S_6 = h(ID_{FA} \| ID_{HA} \| S_5)$, and $S_7 = h(aP.x \| S_5)$, where S_2 is in m_1 sent from MU, and a is generated by $\mathcal{A}$. The attacker then can send $m_4 = \{ID_{FA}, S_6, S_7, aP\}$ to MU.

Upon receiving m_4 from $\mathcal{A}$, MU verifies S_6 and S_7, and trusts that it is communicating with a legitimate FA. Then, it sends bP to $\mathcal{A}$. After this, the attacker computes the session key $K_{MF} = h(abP.x)$. In the end, $\mathcal{A}$ has successfully deceived MU into thinking of it as a valid FA.

2.2.5 Denial-of-Service Attack

HA might face unsynchronization problem when it updates the database without knowing whether MU has completed the authentication phase or not. If MU terminates the session before updating W_i and V_i in the smart card, then it will not be able to authenticate with HA next time.

This may lead to a bigger problem in which an attacker $\mathcal{A}$ can mount DoS attack (Denial-of-Service) to any mobile user. $\mathcal{A}$ can seize any message m_3 or m_4, and cause the corresponding MU unable to authenticate with FA in the future unless re-registering with HA.

3 The Proposed Scheme

The proposed scheme shows how a foreign agent (FA) authenticates a mobile user (MU) with the help of MU's home agent (HA). As a result, FA and MU will be mutually authenticated and share a session key. The scheme consists of four phases: registration phase, authentication phase, session key update phase, and password changing phase.

Assumption. In this scheme, we assume that FA and HA are mutually authenticated and they communicate via a secure channel.

3.1 Registration Phase

In a mobile network, it is required that a mobile user MU registers with its home agent HA. The registration procedure is shown in Figure 1 and has the following steps:

Step 1. First, the mobile user ID_{MU} chooses a password pw_{MU} and a random number $b_1 \in \mathbb{Z}_p^*$. It computes the has $PW_{MU} = h(pw_{MU} \| b_1)$. Then, it submits the registration request to the home agent HA via a secure channel.

$$MU \rightarrow HA : m_{reg} = \{ID_{MU}, PW_{MU}\}.$$

Step 2. Upon receiving the request, the home agent identifies MU and verifies the identity ID_{MU}. HA chooses two random numbers b_2, $r \in \mathbb{Z}_p^*$

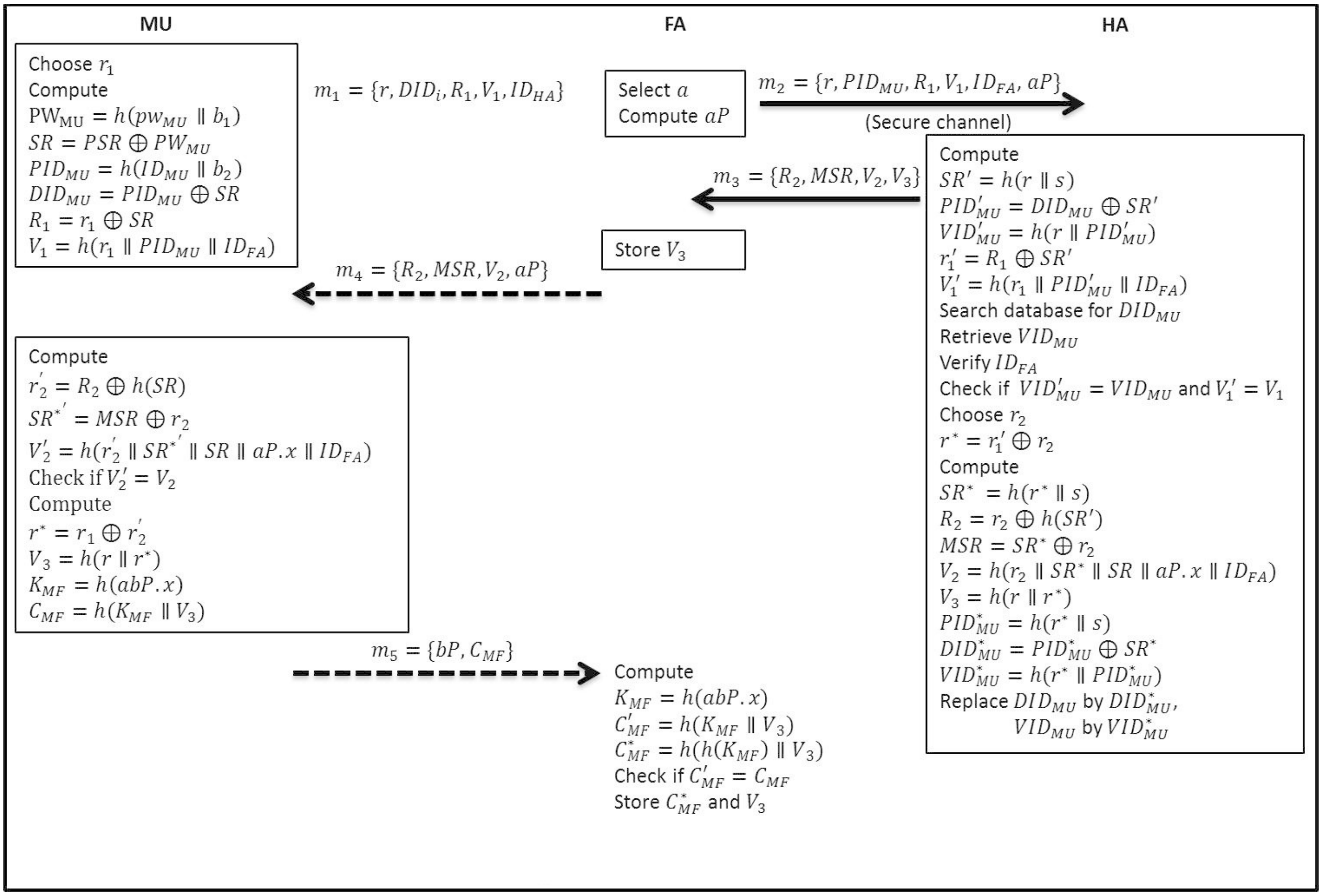

Figure 2: Authentication Phase

and computes the corresponding pseudo-identity $PID_{MU} = h(ID_{MU} \| b_2)$ for MU, and the value $SR = h(r\|s)$, where s is the long-term secret of HA. It uses the hash of the user's password to compute $PSR = SR \oplus PW_{MU}$. Then, HA computes and stores $DID_{MU} = PID_{MU} \oplus SR$, $VID_{MU} = h(r\|PID_{MU})$ in its database as shown in the Table 2. In the end, the home agent writes r, b_2, PSR into a smart card and issues it to MU.

$$HA \to MU : SC = \{r, b_2, PSR\}.$$

Step 3. After receiving the smart card, the mobile user ID_{MU} writes b_1 into it.

3.2 Authentication Phase

In this phase, MU moves into a region handled by a foreign agent FA whose identity is ID_{FA}. The mobile user ID_{MU} will be authenticated anonymously by the home agent HA as shown in the Figure 2 and the following steps:

Step 1. MU inserts the smart card into the reader, and inputs its username ID_{MU} and password pw_{MU}. The smart card uses the provided information to compute $SR = PSR \oplus PW_{MU} = h(r\|s)$, where $PW_{MU} = h(pw_{MU}\|b_1)$. Using the identity of the mobile user, the smart card computes the pseudo-ID $PID_{MU} = h(ID_{MU}\|b_2)$, and the dynamic identity $DID_{MU} = PID_{MU} \oplus SR$. It then chooses a random number $r_1 \in \mathbb{Z}_p^*$, and computes $R_1 = r_1 \oplus SR$, and $V_1 = h(r_1\|PID_{MU}\|ID_{FA})$, where ID_{FA} is the identity of the current foreign agent. The smart card forms a message $m_1 = \{r, DID_{MU}, R_1, V_1, ID_{HA}\}$ and sends it to FA.

$$MU \to FA : m_1 = \{r, DID_{MU}, R_1, V_1, ID_{HA}\}.$$

Step 2. Upon receiving m_1, FA chooses a random number $a \in \mathbb{Z}_p^*$ and computes aP. Then, it sends $m_2 = \{r, DID_{MU}, R_1, V_1, aP\}$ to HA.

$$FA \to HA : m_2 = \{r, DID_{MU}, R_1, V_1, aP\}.$$

Step 3. HA uses its long-term secret s to compute $SR' = h(r\|s)$, and then compute $PID'_{MU} = DID_{MU} \oplus SR'$, and $r' = R_1 \oplus SR'$. It looks up DID_{MU}

Table 2: HA's database layout

$Current - DID$	$Current - VID$	$Previous - DID$	$Previous - VID$
DID_{MU}	VID_{MU}	-	-

in the database and retrieves the corresponding VID_{MU}. HA checks whether $VID_{MU} = h(r\|PID'_{MU})$ and $V_1 = h(r'_1\|PID'_{MU}\|ID_{FA})$. If both equations hold, it trusts that MU is valid. HA then chooses a random number $r_2 \in \mathbb{Z}_p^*$ and computes $r* = r'_1 \oplus r_2$, $SR^* = h(r\|s)$, $R_2 = r_2 \oplus h(SR')$, $MSR = SR^* \oplus r_2$, $V_2 = h(r_2\|SR^*\|SR\|aP.x\|ID_{FA})$, and $V_3 = h(r\|r^*)$, where $aP.x$ is the x-axis value of the point aP.

HA computes $PID^*_{MU} = h(r^*\|s)$, $DID^*_{MU} = PID^*_{MU} \oplus SR^*$, and $VID^*_{MU} = h(r^*\|PID^*_{MU})$. It saves the old values DID_{MU} and VID_{MU} to the $Previous - DID$ and $Previous - VID$ columns in the database, then writes DID^*_{MU} and VID^*_{MU} to the $Current - DID$ and $Current - VID$ columns, respectively. Finally, it transmits the message $m_3 = \{R_2, MSR, V_2, V_3\}$ to FA.

$$HA \rightarrow FA : m_3 = \{R_2, MSR, V_2, V_3\}.$$

Step 4. Ater receiving the confirmation from HA that MU is legitimate, FA stores V_3 and forwards R_2, MSR, V_2, and aP to MU.

$$FA \rightarrow MU : m_4 = \{R_2, MSR, V_2, aP\}.$$

Step 5. From m_4, MU's smart card computes $r'_2 = R_2 \oplus h(SR)$, $SR^{*\prime} = MSR \oplus r'_2$. It verifies whether $V_2 \stackrel{?}{=} h(r'_2\|SR^{*\prime}\|SR\|aP.x\|ID_{FA})$. If it holds, the smart card believes that it is talking to a valid FA. It then computes $r^* = r_1 \oplus r'_2$, $V_3 = h(r\|r^*)$. It chooses a random number $b \in \mathbb{Z}_p^*$, and compute the session key $K_{MF} = h(abP.x)$, and $C_{MF} = h(K_{MF}\|V_3)$. It then sends bP, C_{MF}, to FA.

$$MU \rightarrow FA : m_5 = \{bP, C_{MF}\}.$$

After that, the smart card replaces the current PSR in memory by $PSR^* = PW_{MU} \oplus SR^{*\prime}$, and r by r'.

Step 6. Upon receiving m_5, FA computes the session key $K_{MF} = h(abP.x)$, and verifies if $C_{MF} \stackrel{?}{=} h(K_{MF}\|V_3)$. If they are equal, FA and MU are mutually authenticated and share the session key K_{MF}. The foreign agent then computes $C^*_{MF} = h(h(K_{MF})\|V_3)$, and saves C^*_{MF} and V_3 for this MU in the database for roaming users.

At the end, the mobile user and the foreign agent are mutually authenticated and have a shared session key K_{MF}.

3.3 Session Key Update Phase

MU and FA can renew their session key so that MU can extend its stay in the FA's region or rejoin it. This phase commences with MU sending FA a session key update message as shown in Figure 3.

Step 1. MU's smart card chooses a random number $c \in \mathbb{Z}_p^*$, and computes $C^*_{MF} = h(h(K_{MF})\|V_3)$, and $V_{MF1} = h(cP.x\|V_3)$, where K_{MF} is the current session key. Then, it transmits $\{C^*_{MF}, cP, V_{MF1}\}$ to FA.

$$MU \rightarrow FA : m_6 = \{C^*_{MF}, cP, V_{MF1}\}.$$

Step 2. FA searches for C^*_{MF} in its database. If it is found, FA trusts that MU has been authenticated previously and it retrieves the corresponding V_3 from the database. After that, FA verifies whether $V_{MF1} = h(cP.x\|V_3)$. If it holds, FA chooses a random number $d \in \mathbb{Z}_p^*$ and computes the new session key $K^*_{MF} = h(cdP.x)$, and $V_{MF2} = h(V_3\|K^*_{MF})$. It then sends $\{dP, V_{MF2}\}$ to MU, and updates the current C^*_{MF} with $C^*_{MF} = h(h(K^*_{MF})\|V_3)$.

$$FA \rightarrow MU : \{dP, V_{MF2}\}.$$

Step 3. MU computes the new session key $K^*_{MF} = h(cdP.x)$ and checks if $V_{MF2} = h(V_3\|K^*_{MF})$. If the equation holds, MU updates the session key to K^*_{MF}.

3.4 Password Changing Phase

In this phase, we suppose that MU is already authenticated/ In order to update the password, MU inputs the old password pw_{MU} and the new password pw^*_{MU}. The smart card computes $PW_{MU} = h(b_1\|pw_{MU})$, and $PW^*_{MU} = h(b_1\|pw^*_{MU})$. At last, it replaces the old PSR with $PSR^* = PSR \oplus PW_{MU} \oplus PW^*_{MU}$.

4 Security Analysis

Our scheme has the following security properties.

4.1 Mutual Authentication

In our scheme, we assume that HA and FA are mutually authenticated. In the authentication phase, MU first authenticates with HA; then, HA helps FA and MU to to authenticate each other. Our reasoning is based on BAN

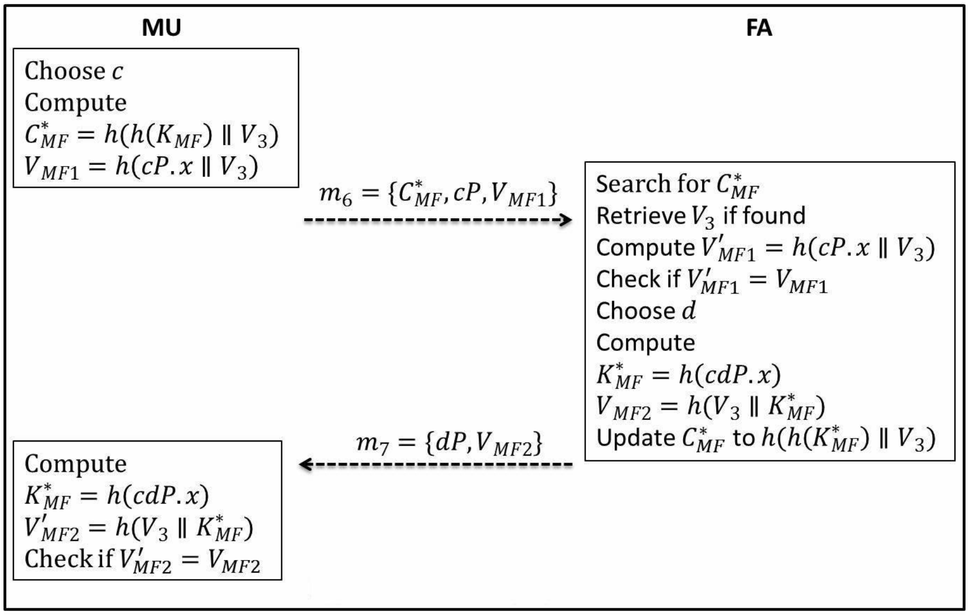

Figure 3: Session key update phase

login [1] to prove that the proposed scheme provides mutual authentication for $MU - HA$ and $MU - FA$.

The value r that is kept by MU is *fresh* for each session, and HA has a verifier of r stored in its database under the tuple $Current - VID_{MU}$, where $VID_{MU} = h(r \| PID_{MU})$. When receiving m_2 from FA, HA uses its long-term secret s to derive PID_{MU}. Then, it verifies the received r by competing VID'_{MU} and comparing it with VID_{MU}. Since HA trusts that VID_{MU} is *fresh*, it believes that m_2 is also *fresh*. Therefore, it trusts that the parameters r, DID_{MU}, R_1, and V_1 come from the legitimate mobile user whose pseudo-ID is PID_{MU}.

At the other end, MU knows that its r is *fresh*. Upon receiving m_4, it verifies $V_2 = h(r'_2 \| SR^{*\prime} \| SR \| aP.x \| ID_{FA})$. If V_2 is valid, MU trusts that V_2 comes from HA since only HA can compute $SR = h(r \| s)$ using its long-term secret s. Because ID_{FA} is in V_2, MU also trusts that FA is genuine. At this point, MU and HA believe that their counterpart is authentic. Since HA helps FA to authenticate MU, both FA and MU are also mutually authenticated.

4.2 Perfect Forward Secrecy

Our scheme uses ECDH (Elliptic Curve Diffie-Hellman) to provide perfect forward secrecy. In the computational ECDH problem, given two points xP and yP, where x, $y \in \mathbb{Z}^*_p$, computing the point xyP is infeasible. Therefore, ECDH is commonly used in establishing session key between two entities. Since x and y are selected at random for each session, there is no feasible way to compute the past session key xyP without knowing either x or y.

In the proposed scheme, HA and FA compute the session key $K_{MF} = h(abP.x)$, where a, b are generated *freshly* for each session. It is infeasible to derive either a or b from aP and bP, respectively. Based on ECDH, it is a computationally difficult problem to guess abP provided aP and bP. Therefore, no adversary can guess the session key K_{MF} for any session even after compromising long-term secrets of MU and HA.

4.3 Achieve Anonymity

In our protocol, HA authenticates MU by verifying its $PID_{MU} = h(ID_{MU} \| b_2)$. The parameters in the message m_2 from FA to HA do not contain ID_{MU}, but only carrying its hash value PID_{MU} in DID_{MU}. Since a secure hash function is used, HA cannot deduce ID_{MU} from PID_{MU}.

Moreover, ID_{MU} is never sent in plaintext. Only PID_{MU} is sent over insecure channel in $DID_{MU} = PID_{MU} \oplus h(r \| s)$. FA or an eavesdropping adversary $\mathcal{A}$ will not be able to derive ID_{MU} from DID_{MU}. Therefore, the mobile user is anonymous to HA, FA, and $\mathcal{A}$.

4.4 Achieve Untraceability

At the end of each authentication phase, r is assigned a new value $r^* = r_1 \oplus r_2$, where r_1, r_2 are *freshly* generated by MU and HA, respectively; thus, it leads to the charges of $h(r \| s)$ and $DID_{MU} = PID_{MU} \oplus h(r \| s)$. Consequently, the parameters in the messages originated from MU do not retain the same values for different sessions. Therefore, FA and $\mathcal{A}$ cannot know whether the messages in two different sessions come from the same MU or not.

Table 3: Comparison regarding security properties

	Mun et al. [7]	Xie et al. [10]	Kuo et al. [5]	Ours
Achieve anonymity	Yes	Yes	Yes	Yes
Achieve untraceability	Yes	Yes	Yes	Yes
Provide perfect forward secrecy	Yes	Yes	Yes	Yes
Prevent disclosure of user's password	No	No	Yes	Yes
Prevent replay attack	No	Yes	Yes	Yes
Provide mutual authentication ($MU - HA$)	Yes	Yes	Yes	Yes
Provide mutual authentication ($MU - FA$)	Yes	Yes	Yes	Yes
Prevent man-in-the-middle attack	Yes	No	No	Yes
Session key security	Yes	Yes	Yes	Yes
Smart card lost attack	No	Yes	No	No
Stolen verifier attack	No	No	Yes	No

4.5 Prevent Disclosure of User's Password

In the proposed scheme, the mobile user's password is only used to compute $PW_{MU} = h(pw_{MU} \| b_1)$, and there is no information related to PW_{MU} in any message sent from MU. Therefore, no adversary can obtain pw_{MU} by eavesdropping on the communication channel.

4.6 Prevent Man-in-the-middle Attack

Kuo et al.'s suffers four kinds of man-in-the-middle attacks because it does not verify aP and bP properly. In our scheme, MU makes sure that aP comes from FA, and FA can verify that bP comes from MU.

HA provides FA with $V_3 = h(r \| r^*)$ so that it can authenticate bP. When receiving m_4 from MU, FA computes the session key K_{MF} and uses V_3 to verify it by comparing $h(K_{MF} \| V_3)$ against the received value C_{MF}.

At the mobile user side, the smart card can verify aP since $aP.x$ is contained in $V_2 = h(r_2 \| SR^* \| SR \| aP.x \| ID_{FA})$ which is sent from HA. If MU and HA are already mutually authenticated, MU will trust that aP indeed comes from FA.

4.7 Prevent Replay Attack

The values r, r_1, a, and b are generated for each session, and the parameters in all the messages are all related to them, Those values are verified by MU, FA, and HA; therefore, an adversary cannot deceive any legitimate entity by replaying old messages.

4.8 Prevent Stolen Verifier Attack

If an attacker $\mathcal{A}$ compromises HA's database, it can obtain $DID_{MU} = PID_{MU} \oplus h(r \| s)$, and $VID_{MU} = h(r \| PID_{MU})$. However, without HA's long-term secret s, it cannot compete $h(r \| s)$ which is used in computing R_2, V_2. Therefore, it will not be able to masquerade as FA like it could do in Kuo et al.'s scheme.

4.9 Prevent Smart Card Lost Attack

If $\mathcal{A}$ obtains a valid smart card of ID_{MU}, it can retrieve b_1, b_2, r and $PSR = h(r \| s) \oplus PW_{MU}$, where $PW_MU = h(pw_{MU} \| b_1)$. The attacker does not know pw_{MU} to compute PW_{MU}. Without the password, $\mathcal{A}$ cannot derive $SR = h(r \| s)$ which plays essential role in authentication. Therefore, loosing smart card will not compromise the security of the system.

5 Functionality and Performance Analysis

In this section, we evaluate our proposed scheme in terms of security properties and computation costs. We compare these features in our scheme with their counterparts in Mun et al.'s [7], Xie et al's [10], and Kuo et al's [5].

The comparison for security features is shown in the Table 3. All the schemes can achieve mutual authentication ($MU - HA$, $MU - FA$), perfect forward secrecy, session key security, anonymity, and untraceability. However, Mun et al.'s and Xie et al.'s schemes are vulnerable to disclose the mobile user's password. Both Xie et al.'s and Kuo et al.'s cannot prevent man-in-the-middle attacks. Since there is no verifier database in Mun et al.'s and Xie et al.'s, stolen verifier attack is not a threat to them, but Kuo et al.' scheme is susceptible to this kind of attack. And lastly, all the schemes except Xie et al.'s are immune to smart card lost attack.

The schemes' performances in term of computation costs in the authentication phase are shown in Table 4. Mum et al.'s and Xie et al.'s employ both symmetric and asymmetric cryptography in their schemes. Therefore, their computing workloads are higher than Kuo et al.'s and our scheme. Like in Kuo et al.'s, we use only elliptic curve point multiplications in establishing session key. MU in our scheme performs less computing than Kuo et al.'s one, whereas computing workloads on our FA and HA are slightly higher than Kuo et al.'s. However, our scheme is more sufficient than Kuo et al.'s in the pass-

Table 4: Comparison regarding computation costs

	Mun et al. [7]	Xie et al. [10]	Kuo et al. [5]	Ours
MU	$2t_p + t_s + 5t_h + 2t_{XOR}$	$3t_e + 4t_h + 2t_s + t_{XOR}$	$2t_p + 9t_h + 6t_{XOR}$	$2t_p + 7t_h + 7t_{XOR}$
FA	$2t_p + t_s + 4t_h + 2t_{XOR}$	$3t_e + 2t_h + 3t_s$	$2t_p + 2t_h$	$2t_p + 3t_h$
HA	$5t_h + 3t_{XOR}$	$2t_e + t_h + 4t_s + t_{XOR}$	$6t_h + 2t_{XOR}$	$8t_h + 6t_{XOR}$
$Total$	$4t_p + 2t_s + 14t_h + 7t_{XOR}$	$8t_e + 7t_h + 9t_s + 2t_{XOR}$	$4t_p + 11t_h + 8t_{XOR}$	$4t_p + 18t_h + 13t_{XOR}$

t_e : time for performing modular exponentiation
t_p : time for performing elliptic curve point multiplication
t_s : time for performing symmetric encryption/decryption
t_h : time for performing hash operation
t_{XOR} : time for performing XOR operation

word changing phase as shown in Table 5 since HA does not have to involve in the process.

Table 5: Computing workloads in Password Changing Phase

	Kuo et al. [5]	Ours
MU	$4t_h + 2t_{XOR}$	$2t_h + 2t_{XOR}$
FA	0	0
HA	$2t_h + t_{XOR}$	0

6 Conclusions

In this paper, we proposed a novel and secure authentication and key agreement scheme for roaming service in global mobile network. Our scheme achieves mutual authentication for $MU - HA$, and $MU - FA$. To ensure mobile user's anonymity, pseudo-identity is used in place of the actual identity. All the parameters in the messages exchanged are *fresh* and not repeated so that mobile user's activities are not traceable. Perfect forward secrecy is preserved even in the extreme case where the long-term secret of HA is compromised. Furthermore, the proposed scheme uses mostly hash functions and XOR operations, and very few elliptic curve point multiplication; as a result, it is very efficient and suitable for use in mobile networks.

Acknowledgments

The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 426, pp. 233–271. The Royal Society, 1989.

[2] C. C. Chang, C. Y. Lee, and Y. C. Chiu, "Enhanced authentication scheme with anonymity for roaming service in global mobility networks," *Computer Communications*, vol. 32, no. 4, pp. 611–618, 2009.

[3] D. He, N. Kumar, M. Khan, and J. H. Lee, "Anonymous two-factor authentication for consumer roaming service in global mobility networks," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 4, pp. 811–817, 2013.

[4] J. S. Kim and J. Kwak, "Improved secure anonymous authentication scheme for roaming service in global mobility networks," *International Journal of Security and Its Applications*, vol. 6, no. 3, pp. 45–54, 2012.

[5] 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, no. 1, pp. 18–24, 2014.

[6] C. C. Lee, M. S. Hwang, and I-E. Liao, "Security enhancement on a new authentication scheme with anonymity for wireless environments," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 5, pp. 1683–1687, 2006.

[7] H. Mun, K. Han, Y. S. Lee, C. Y. Yeun, and H. H. Choi, "Enhanced secure anonymous authentication scheme for roaming service in global mobility networks," *Mathematical and Computer Modelling*, vol. 55, no. 1, pp. 214–222, 2012.

[8] S. Suzuki and K. Nakada, "An authentication technique based on distributed security management for the global mobility network," *IEEE Journal of Selected Areas in Communications*, vol. 15, no. 8, pp. 1608–1617, 1997.

[9] C. C. Wu, W. B. Lee, and W. J. Tsaur, "A secure authentication scheme with anonymity for wireless communications," *IEEE Communications Letters*, vol. 12, no. 10, pp. 722–723, 2008.

[10] Qi Xie, B. Hu, X. Tan, M. Bao, and X. Yu, "Robust anonymous two-factor authentication scheme for roaming service in global mobility network,"

Wireless personal communications, vol. 74, no. 2, pp. 601–614, 2014.

[11] C. K. Yeh and W. B. Lee, "An overall cost-effective authentication technique for the global mobility network," *International Journal of Network Security*, vol. 9, no. 3, pp. 227–232, 2009.

[12] T. Y. Youn, Y. H. Park, and J. Lim, "Weaknesses in an anonymous authentication scheme for roaming service in global mobility networks," *IEEE Communications Letters*, vol. 13, no. 7, pp. 471–473, 2009.

[13] J. Zhu and J. Ma, "A new authentication scheme with anonymity for wireless environments," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 231–235, 2004.

Hai-Duong Le received his B.E. degree in 2004 at University of Tasmania, Australia, and his M.I.T degree in 2006 at James Cook University, Australia. He is currently pursuing his Ph.D. degree in Computer Science and Information Engineering from Feng Chia University, Taichung, Taiwan. His current research interests include electronic commerce, information security, computer cryptography, and mobile communications.

Chin-Chen Chang received his Ph.D. degree in computer engineer- ing from National Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in computer and decision sciences. Both were awarded in National Tsing Hua University. Dr. Chang served in National Chung Cheng University from 1989 to 2005. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. Prior to joining Feng Chia University, Professor Chang was an associate professor in Chiao Tung University, professor in National Chung Hsing University, chair professor in National Chung Cheng University. He had also been Visit- ing Researcher and Visiting Scientist to Tokyo University and Kyoto University, Japan. During his service in Chung Cheng, Professor Chang served as Chairman of the Institute of Computer Science and Information Engineering, Dean of College of Engineering, Provost and then Acting President of Chung Cheng University and Director of Advisory Office in Ministry of Education, Taiwan. Professor Chang has won many research awards and honorary positions by and in prestigious organisations both nationally and internationally. He is currently a Fellow of IEEE and a Fellow of IEE, UK. And since his early years of career development, he consecutively won Outstanding Talent in Information Sciences of the R. O. C., AceR Dragon Award of the Ten Most Outstanding Talents, Outstanding Scholar Award of the R. O. C., Outstanding Engineering Professor Award of the R. O. C., Distinguished Research Awards of National Science Council of the R. O. C., Top Fifteen Scholars in Systems and Software Engineering of the Journal of Systems and Software, and so on. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression and data structures.

Yeh-Chieh Chou was born in Taichung, Taiwan, in 1990. He received his Bachelor?s Degree in Information Engineering form Chang Jung Christian University, Tainan. Currently, he is the second grade student for master?s program of Department of Information Engineering in Feng Chia University. His research interests include computer cryptography and information security.