Secure and Efficient Anonymous Authentication Scheme in Global Mobility Networks

Jun-Sub Kim and Jin Kwak

1 ISAA Lab, Department of Information Security Engineering, Soonchunhyang University, Asan, Chungchungnam-do 336-745, Republic of Korea
2 Department of Information Security Engineering, Soonchunhyang University, Asan, Chungchungnam-do 336-745, Republic of Korea

Correspondence should be addressed to Jin Kwak; jkwak@sch.ac.kr

Received 28 August 2013; Accepted 16 September 2013

Abstract

In 2012, Mun et al. pointed out that Wu et al.’s scheme failed to achieve user anonymity and perfect forward secrecy and disclosed the passwords of legitimate users. And they proposed a new enhancement for anonymous authentication scheme. However, their proposed scheme has vulnerabilities that are susceptible to replay attack and man-in-the-middle attack. It also incurs a high overhead in the database. In this paper, we examine the vulnerabilities in the existing schemes and the computational overhead incurred in the database. We then propose a secure and efficient anonymous authentication scheme for roaming service in global mobility network. Our proposed scheme is secure against various attacks, provides mutual authentication and session key establishment, and incurs less computational overhead in the database than Mun et al.’s scheme.

1. Introduction

Global mobility network (GLOMONET) provides global roaming services for mobile user between the home agent and the foreign agent. The GLOMONET must have a user authentication scheme in which the mobile user has secure access to the foreign agent. A strong user authentication scheme in GLOMONET should satisfy the following requirements: (1) user anonymity, (2) low communication cost and computation complexity, (3) single registration, (4) update session key periodically, (5) user friendly, (6) password/verifier table, (7) update password securely and freely, (8) prevention of fraud, (9) prevention of replay attack, (10) security, and (11) providing the authentication scheme when a user is located in the home network [1, 2].

Many user authentication schemes for use in GLOMONET have been proposed [1–18]. In 2004, Zhu and Ma [4] proposed a simple, efficient wireless authentication scheme that provides user anonymity for wireless environments. However, Lee et al. [5] subsequently pointed out that Zhu et al.’s scheme does not achieve mutual authentication and perfect backward secrecy, and therefore cannot protect against forgery attacks. They then proposed a slight modification of Zhu et al.’s scheme. Unfortunately, Wu et al. [6] demonstrated that Lee et al.’s proposed scheme still failed to provide anonymity and perfect backward secrecy. Consequently, they proposed an improvement to overcome the weakness identified in Lee et al.’s scheme. In 2009, Zeng et al. [7] showed that Wu et al.’s scheme also fails to provide anonymity. In 2012, Mun et al. [12] showed that Wu et al.’s scheme discloses the password of legitimate users and does not achieve perfect forward secrecy. They subsequently proposed a new enhancement for anonymous authentication to overcome these security weaknesses. However, their scheme is vulnerable to replay attack and man-in-the-middle attack, and incurs a high overhead in the database of the home agent.

Therefore, in this paper, we analyze the existing schemes [5, 6, 12] and show that it is vulnerable to security requirement. And we propose a secure and efficient anonymous authentication scheme that is resistant to replay attack and man-in-the-middle attack. Our proposed scheme also incurs less computational overhead in the database than Mun et al.’s scheme.

The remainder of this paper is organized as follows. In Section 2, we review the existing schemes, while in Section 3,
we investigate the security vulnerabilities mentioned above. In Section 4, we present our proposed secure and efficient anonymous authentication scheme. This scheme is analyzed and compared with other schemes in Section 5. Finally, Section 6 presents our conclusions.

2. Review of the Previous Schemes

In this section, we examine variety of authentication schemes with anonymity proposed by Lee et al. [5], Wu et al. [6], and Mun et al. [12].

2.1. Lee et al’s Scheme. Figure 1 shows the procedure of Lee et al’s scheme. Their scheme comprises three phases: an initial phase, a first phase, and a second phase.

2.1.1. Initial Phase. When a new mobile user MU wants to register with a home agent HA, he/she performs the following steps.

Step 1. Consider MU → HA : \{IDMU\}.
MU sends his/her identifier IDMU to HA for registration.

Step 2. HA computes PW_{MU} = h(N \| IDMU) and r = h(N \| ID_{HA} \oplus h(N \| IDMU) \oplus ID_{HA}) \oplus IDMU, where N is a long random number kept by HA.

Step 3. Consider HA → MU : \{PW_{MU}, smart card \[ID_{HA}, r, h(\cdot)\}\}.
HA delivers PW_{MU} and a smart card containing \[ID_{HA}, r, h(\cdot)\] to MU through a secure channel.
2.1.2. First Phase. In this phase, FA authenticates MU and issues a temporary certificate to MU, which will be used in the second phase when MU always communicates this FA within this area. MU performs the following steps.

Step 1. Consider MU → FA : \{n, c_1, ID_HA, T_MU\}.
MU computes \( n = r \oplus h(N \parallel ID_MU) \) and temporary key \( L = h(T_MU \parallel PW_MU) \), and encrypts \( c_1 = (h(ID_MU) \parallel [x_0|x])_L \) using symmetric key \( L \), where \( x_0 \) and \( x \) are secret random numbers. And, MU sends \( n, c_1, ID_HA, \) and \( T_MU \) to FA.

Step 2. Consider FA → HA : \{a, n, c_1, T_MU, c_2, Cert_{FA}, T_{FA}\}.

If certificate and timestamp are valid, FA generates a secret random number \( a \) and computes signature \( c_2 = E_{K_{FA}}(h(a, n, c_1, T_MU, Cert_{FA})) \) using private key \( K_{FA} \). And, FA sends \( a, n, c_1, T_MU, c_2, Cert_{FA}, \) and \( T_{FA} \) to HA.

Step 3. Consider HA → FA : \{b, c_3, Cert_{HA}, T_{HA}\}.
If certificate and timestamp are valid, HA computes \( L = h(T_MU \parallel PW_MU) \) and \( H = h(ID_HA) \), and decrypts \( (h(ID_MU) \parallel [x_0|x])_L \) using symmetric key \( L \). If \( h(ID_HA) \) is identical to \( H \), HA authenticates MU. And, HA encrypts \( c_3 = E_{K_{FA}}(h(ID_MU) \parallel [x_0|x]) \) using public key \( K_{FA} \) and computes signature \( c_4 = E_{K_{HA}}(h(a, b, c_3, Cert_{HA})) \) using private key \( K_{HA} \). HA then sends \( b, c_3, c_4, Cert_{HA}, \) and \( T_{HA} \) to FA.

Step 4. Consider FA → MU : \{c_5\}.
If certificate and timestamp are valid, FA issues the temporary certificate \( T_{Cert_{MU}} \) and decrypts \( E_{K_{MU}}(h(ID_MU) \parallel [x_0|x]) \) using private key \( K_{FA} \). And, FA computes \( h(x_0 \parallel x) \) and session key \( k = h(ID_MU \parallel x) \oplus x_0 \) and encrypts \( c_6 = (T_{Cert_{MU}} \parallel h(x_0 \parallel x))_k \) using symmetric key \( k \). FA then sends \( c_6 \) to MU.

Step 5. MU computes \( M = h(x_0 \parallel x) \) and session key \( k = h(h(h(N \parallel ID_MU)) \parallel [x_0|x]) \parallel [x_0|x] \) using symmetric key \( k \). If \( h(x_0 \parallel x) \) is identical to \( M \), MU authenticates FA.

2.1.3. Second Phase. In this phase, MU visits FA at \( i \)th session when he/she is always within this FA. MU performs the following steps.

Step 1. Consider MU → FA : \{T_{Cert_{MU}}, c_6\}.
MU encrypts \( c_6 = (x_i \parallel T_{Cert_{MU}} \parallel OtherInfomation)_k \) using symmetric key \( k \), where \( k_i = h(ID_MU \parallel x) \oplus x_{i-1} \), for \( i = 1, 2, \ldots, n \). And, MU sends \( T_{Cert_{MU}} \) and \( c_6 \) to FA.

Step 2. If \( T_{Cert_{MU}} \) is valid, FA decrypts \( (x_i \parallel T_{Cert_{MU}} \parallel OtherInfomation)_k \) using symmetric key \( k \). If received \( T_{Cert_{MU}} \) if identical to obtained \( T_{Cert_{MU}} \), FA authenticates MU.

2.2. Wu et al’s Scheme. Figure 2 shows the first and second phase of Wu et al’s scheme. Their scheme comprises three phases: an initial phase, a first phase, and a second phase. The initial phase is the same as the initial phase of Lee et al’s scheme.

2.2.1. First Phase. In this phase, FA authenticates MU and issues a temporary certificate to MU, which will be used in the second phase when MU always communicates this FA within this area. MU performs the following steps.

Step 1. Consider MU → FA : \{n, c_1, ID_HA, T_MU\}.
MU computes \( n = r \oplus h(N \parallel ID_MU) \) and temporary key \( L = h(T_MU \parallel PW_MU) \), and encrypts \( c_1 = (h(ID_MU) \parallel [x_0|x])_L \) using symmetric key \( L \), where \( x_0 \) and \( x \) are secret random numbers. And, MU sends \( n, c_1, ID_HA, \) and \( T_MU \) to FA.

Step 2. Consider FA → HA : \{a, n, c_1, T_MU, c_2, Cert_{FA}, T_{FA}\}.
If certificate and timestamp are valid, FA generates a secret random number \( a \) and computes signature \( c_2 = E_{K_{FA}}(h(a, n, c_1, T_MU, Cert_{FA})) \) using private key \( K_{FA} \). And, FA sends \( a, n, c_1, T_MU, c_2, Cert_{FA}, \) and \( T_{FA} \) to HA.

Step 3. Consider HA → FA : \{b, c_3, Cert_{HA}, T_{HA}\}.
If certificate and timestamp are valid, HA computes \( L = h(T_MU \parallel PW_MU) \) and \( H = h(ID_HA) \), and decrypts \( (h(ID_MU) \parallel [x_0|x])_L \) using symmetric key \( L \). If \( h(ID_HA) \) is identical to \( H \), HA authenticates MU. And, HA encrypts \( c_3 = E_{K_{FA}}(h(h(h(N \parallel ID_MU)) \parallel [x_0|x]) \parallel [x_0|x]) \) using public key \( K_{FA} \) and computes signature \( c_4 = E_{K_{HA}}(h(a, b, c_3, Cert_{HA})) \) using private key \( K_{HA} \). HA then sends \( b, c_3, c_4, Cert_{HA}, \) and \( T_{HA} \) to FA.

Step 4. Consider FA → MU : \{c_5\}.
If certificate and timestamp are valid, FA issues the temporary certificate \( T_{Cert_{MU}} \) and decrypts \( E_{K_{MU}}(h(h(h(N \parallel ID_MU)) \parallel [x_0|x]) \parallel [x_0|x]) \) using private key \( K_{FA} \). And, FA computes \( h(x_0 \parallel x) \) and session key \( k = h(h(h(N \parallel ID_MU)) \parallel [x_0|x]) \parallel [x_0|x] \) using symmetric key \( k \). FA then sends \( c_5 \) to MU.

Step 5. MU computes \( M = h(x_0 \parallel x) \) and session key \( k = h(h(h(N \parallel ID_MU)) \parallel [x_0|x]) \parallel [x_0|x] \) using symmetric key \( k \). If \( h(x_0 \parallel x) \) is identical to \( M \), MU authenticates FA.

2.2.2. Second Phase. In this phase, MU visits FA at \( i \)th session when he/she is always within this FA. MU performs the following steps.

Step 1. Consider MU → FA : \{T_{Cert_{MU}}, c_6\}.
MU encrypts \( c_6 = (x_i \parallel T_{Cert_{MU}} \parallel OtherInfomation)_k \) using symmetric key \( k \), where \( k_i = h(h(h(N \parallel ID_MU)) \parallel [x_0|x]) \parallel [x_0|x] \) for \( i = 1, 2, \ldots, n \). And, MU sends \( T_{Cert_{MU}} \) and \( c_6 \) to FA.

Step 2. If \( c_6 \) is valid, FA decrypts \( (x_i \parallel T_{Cert_{MU}} \parallel OtherInfomation)_k \) using symmetric key \( k \). If received \( T_{Cert_{MU}} \) if identical to obtained \( T_{Cert_{MU}} \), FA authenticates MU.

2.3. Mun et al’s Scheme. Their scheme comprises three phases: a registration phase, an authentication phase, and an update phase.

2.3.1. First Phase. Figure 3 shows the procedure of the first phase. When a new MU, wants to register with HA, he/she performs the following steps.
**First and second phase**

**MU (mobile user)**
Compute: \( L = h(T_{\text{MU}} \oplus \text{PWMU}) \)
\( n = r \oplus \text{PWMU} \)
Encrypt: \( c_1 = (h(ID_{\text{MU}}) || x_0 || x) L \)
\( \{ n, c_1, ID_{\text{MU}}, T_{\text{MU}} \} \)
Select: \( a \)
Compute: \( c_2 = E_{KR_{\text{FA}}} (h(a, n, c_1, T_{\text{MU}}, Cert_{\text{FA}})) \)
\( \{ a, n, c_1, T_{\text{MU}}, c_2, Cert_{\text{FA}}, T_{\text{FA}} \} \)

**HA (home agent)**
Compute: \( ID_{\text{MU}} = n \oplus h(N || ID_{\text{HA}}) \oplus ID_{\text{HA}} \)
Encrypt: \( c_3 = E_{KR_{\text{FA}}} (h(h(N || ID_{\text{MU}})) || x_0 || x) \)
\( \{ b, c_3, c_4, Cert_{\text{HA}}, T_{\text{HA}} \} \)
 Decrypt: \( E_{KU_{\text{FA}}} (h(h(N || ID_{\text{MU}})) || x_0 || x) \)
Compute: \( k = h(h(h(N || ID_{\text{MU}})) || x || x_0) \)
Encrypt: \( c_5 = (T_{\text{Cert}_{\text{MU}}} || h(x_0 || x)) k \)
\( \{ c_5 \} \)

**FA (foreign agent)**
Decrypt: \( E_{KU_{\text{FA}}} (h(h(N || ID_{\text{MU}})) || x_0 || x) \)
Compute: \( k = h(h(N || ID_{\text{MU}})) || x || x_0) \)
Encrypt: \( c_6 = xi || T_{\text{Cert}_{\text{MU}}} || OtherInformation) k_i \)
\( \{ T_{\text{Cert}_{\text{MU}}}, c_6 \} \)

**Step 1.** Consider MU → HA: \( \{ ID_{\text{MU}}, N_{\text{MU}} \} \).
MU sends his/her identifier \( ID_{\text{MU}} \) and nonce \( N_{\text{MU}} \) to HA for registration.

**Step 2.** HA generates nonce \( N_{\text{HA}} \) and computes \( \text{PWMU} = h(N_{\text{MU}} \parallel N_{\text{HA}}) \) and \( r_{\text{MU}} = h(ID_{\text{MU}} \parallel \text{PWMU}) \oplus ID_{\text{HA}} \).

**Step 3.** Consider HA → MU: \( \{ r_{\text{MU}}, ID_{\text{HA}}, N_{\text{HA}}, \text{PWMU}, h(\cdot) \} \).
HA sends \( r_{\text{MU}}, ID_{\text{HA}}, N_{\text{HA}}, \text{PWMU} \), and \( h(\cdot) \) to MU through a secure channel.

2.3.2. **Second Phase.** Figure 4 shows the procedure of the second phase. In this phase, for mutual authentication between MU and HA and between MU and a foreign agent FA, the following steps are performed.

**Step 1.** Consider MU → FA: \( \{ ID_{\text{HA}}, N_{\text{HA}}, r_{\text{MU}} \} \).
MU accesses the new FA and sends \( ID_{\text{HA}}, N_{\text{HA}}, \) and \( r_{\text{MU}} \) to it.

**Step 2.** Consider FA → HA: \( \{ ID_{\text{FA}}, N_{\text{FA}}, r_{\text{MU}} \} \).
FA stores the message received from MU for further communication and generates nonce \( N_{\text{FA}} \). FA then sends \( ID_{\text{FA}}, N_{\text{FA}}, \) and \( r_{\text{MU}} \) to HA.

**Step 3.** Consider HA → FA: \( \{ S_{\text{FA}}, P_{\text{FA}} \} \).
HA computes \( r'_{\text{MU}} = h(ID_{\text{MU}} \parallel \text{PWMU}) \oplus ID_{\text{HA}} \) and checks whether \( r'_{\text{MU}} \) is identical to the received \( r_{\text{MU}} \). If they are identical, HA authenticates MU. Next, HA computes \( P_{\text{HA}} = h(\text{PWMU} \parallel N_{\text{FA}}) \) and\( S_{\text{HA}} = h(ID_{\text{FA}} \parallel N_{\text{FA}}) \oplus r_{\text{MU}} \oplus P_{\text{FA}} \), and sends the computed \( S_{\text{HA}} \) and \( P_{\text{HA}} \) to FA.

**Step 4.** Consider FA → MU: \( \{ S_{\text{FA}}, aP, P_{\text{FA}} \} \).
FA computes \( S'_{\text{HA}} = h(ID_{\text{FA}} \parallel N_{\text{FA}}) \oplus r_{\text{MU}} \oplus P_{\text{FA}} \) and checks whether \( S'_{\text{HA}} \) is identical to the received \( S_{\text{HA}} \). If they are identical, FA authenticates HA. Then, FA computes \( S_{\text{FA}} = h(S_{\text{HA}} \parallel N_{\text{FA}} \parallel N_{\text{HA}}) \), selects a random number \( a \), and then computes \( aP \) on \( E \) using the elliptic curve Diffie-Hellman (ECDH) protocol. Next, FA sends \( S_{\text{FA}}, aP, \) and \( P_{\text{FA}} = (S_{\text{HA}} \parallel ID_{\text{FA}} \parallel N_{\text{FA}}) \) to MU.

**Figure 2:** First and second phase of Wu et al.’s scheme.
Step 5. Consider MU → FA: \{bP, S_{MF}\}.

MU computes \(S'_{FA} = h(ID_{FA} || N_{FA}) \oplus r_{MU} \oplus PW_{MU} || N_{FA}\) and \(S'_{FA} = h(S'_{FA} || N_{FA} || N_{HA})\), and checks whether \(S'_{FA}\) is identical to the received \(S_{FA}\). If they are identical, MU authenticates HA and FA. After checking \(S_{FA}\), MU selects a random number \(b\) and computes \(bP\), a session key \(K_{MF} = h(abP)\) using the received \(aP\) and the computed \(bP\), and \(S_{MF} = f_{K_{MF}}(N_{FA} || bP)\). Next, MU sends the computed \(bP\) and \(S_{MF}\) to FA.

Step 6. FA computes \(K_{MF} = h(abP)\) using private and public values, and \(S'_{MF} = f_{K_{MF}}(N_{FA} || bP)\). FA then checks whether \(S'_{MF}\) is identical to the received \(S_{MF}\). If they are identical, FA authenticates MU.

2.3.3. Third Phase. The procedure followed in the third phase is depicted in Figure 5. The steps are as follows.

Step 1. Consider MU → FA: \{bP\}.
Figure 5: Third phase of Mun et al.’s scheme.

3. Vulnerabilities in the Previous Schemes

3.1. Vulnerability of Lee et al’s and Wu et al’s Scheme. Lee et al’s and Wu et al’s scheme are almost the same. Therefore, their schemes are also the same vulnerabilities. Their scheme is vulnerable replay attack, is disclosed password, and cannot achieve anonymity and perfect forward secrecy.

3.1.1. Anonymity. An adversary A can eavesdrop on and record the message \(\{n, c_1, ID_{HA}, T_{MU}\}\) transmitted from MU to FA, and can obtain MUs ID_{MU} as follows.

Step 1. A register as legitimate user to HA and obtain own PW_A and r. And, A compute \(h(N \ || \ ID_{HA})\) using PW_A, r, ID_{HA}, and ID_A.

Step 2. A eavesdrops on and records messages \(\{n, c_1, ID_{HA}, T_{MU}\}\) transmitted from FA to MU.

Step 3. A compute ID_{MU} using \(n, h(N \ || \ ID_{HA})\) and ID_{HA}.

Therefore, Lee et al’s and Wu et al’s scheme cannot achieve anonymity [7].

3.1.2. Replay Attack. Legitimate FA_i can record the message \(\{n, c_1, ID_{HA}, T_{MU}\}\) transmitted from MU, and can then impersonate MU by using the recorded message \(\{n, c_1, ID_{HA}, T_{MU}\}\) to another FA_j as follows.

Step 1. FA_i accesses another FA_j and sends recorded message \(\{n, c_1, ID_{HA}, T_{MU}\}\) to this FA_j. FA_i can replay this message within the lifetime of T_{MU}. After receiving this message, FA_j sends the message \(\{a, n, c_1, T_{MU}, c_2, Cert_{FA}, T_{FA}\}\) to HA.

Step 2. HA compute \(h(ID_{MU})\) and checks whether the computed \(h(ID_{MU})\) is identical to the received \(h(ID_{MU})\). If they are identical, HA authenticate FA_i, then sends the message \(\{b, c_3, c_4, Cert_{HA}, T_{HA}\}\) to FA_j.

Step 3. FA_j computes session key k and sends the message \(\{c_3\}\) to FA_i. FA_i computes the session key k between FA_i and MU, which is the same as the session key between FA_j and FA_i. And, FA_i decrypts \(c_3\) and authenticates FA_j.

Therefore, Lee et al’s and Wu et al’s scheme is vulnerable to replay attack [11].

3.1.3. Disclosure Password. If an adversary A can steal MU’s smart card, A can obtain MU’s password PW_{MU} as follows.

Step 1. A can record the message \(\{n, c_1, ID_{HA}, T_{MU}\}\) transmitted from MU to FA. And, as described in Section 3.1.1, A can obtain the message \(h(N \ || \ ID_{HA})\) of ID_{HA}.

Step 2. A stole MU’s smart card, inserts MU’s smart card into the device, and enters the fake password PW* = 0. The smart card computes \(n^* = r @ PW* = h(N \ || \ ID_{HA})@h(N \ || \ ID_{MU})@ID_{HA} \oplus ID_{MU}\) and A obtains \(n^*\) by eavesdropping.

Step 3. A computes PW_{MU} using \(n^*, h(N \ || \ ID_{HA}), ID_{HA},\) and ID_{MU}.

Therefore, Lee et al’s and Wu et al’s scheme are disclosed password [11].

3.1.4. Perfect Forward Secrecy. Assume that an adversary A obtain MU’s password PW_{MU}. Failing to provide perfect forward secrecy is as follows.

Step 1. A computes L using T_{MU} and PW_{MU} and decrypts \(h(ID_{MU}) \oplus x_0 \ || \ x\|L\) using L. Thus, A obtains \(x_0, x,\) and \(h(ID_{MU})\).
Step 2. A computes session key $k_1$ using $x_0, x$, and $PW_{MU}$ and decrypts $(x_1 || T_{Cert_{MU}} || OtherInformation)$ using $k_1$. Thus, $A$ obtains $x_1$.

Step 3. A computes session key $k_2$ using $x_1, x$, and $PW_{MU}$. Therefore, Lee et al.'s and Wu et al.'s scheme cannot achieve perfect forward secrecy [11].

3.2. Vulnerability of Mun et al.'s Scheme. Mun et al. claimed that their scheme can thwart a variety of known attacks. Unfortunately, we found that their scheme is vulnerable to replay attack and man-in-the-middle attack. In addition, their scheme incurs a high overhead in the database of the home agent.

3.2.1. Replay Attack. In Mun et al.'s scheme, an adversary $A$ can eavesdrop on and record the message $(ID_{HA}, N_{HA}, r_{MU})$ transmitted from MU to FA; and can then impersonate MU by using the recorded message $(ID_{HA}, N_{HA}, r_{MU})$ as follows.

Step 1. $A$ accesses a new FA and sends the recorded message $(ID_{HA}, N_{HA}, r_{MU})$ to this FA. After receiving this message, the FA sends the message $(ID_{FA}, N_{FA}, r_{MU})$ to HA.

Step 2. HA computes $r'_{MU}$ and checks whether $r'_{MU}$ is identical to the received $r_{MU}$. If they are identical, HA authenticates $A$, then computes $P_{HA}$ and $S_{HA}$ and sends the message $(S_{HA}, P_{HA})$ to FA. On receiving this message, FA computes $S_{HA}$ and checks whether $S_{HA}$ is identical to the received $S_{HA}$. Next, FA sends the message $(S_{FA}, aP, P_{FA})$ to $A$.

Step 3. $A$ computes $S'_{FA}$ and checks whether $S'_{FA}$ is identical to the received $S_{FA}$. If they are identical, $A$ authenticates HA and FA, then computes $bP$ and $S_{MF}$, and sends the message $(bP, S_{MF})$ to FA. On receiving this message, FA computes $S'_{MF}$ and checks whether $S'_{MF}$ is identical to the received $S_{MF}$. If they are identical, FA authenticates $A$.

Therefore, Mun et al.'s scheme is vulnerable to replay attack [18].

3.2.2. Man-in-the-Middle Attack. In Mun et al.'s scheme, an adversary $A$ can eavesdrop on messages transmitted between FA and MU. Consequently, $A$ can also successfully mount a man-in-the-middle attack as follows.

Step 1. $A$ blocks and copies the message $(S_{FA}, aP, P_{FA})$ transmitted from FA to MU. It then selects a new random number $a'$, computes $a'P$, replaces message $(S_{FA}, aP, P_{FA})$ with $(S_{FA}, a'P, P_{FA})$, and sends this to MU.

Step 2. MU computes $S'_{HA}$ and $S'_{FA}$, and checks whether $S'_{FA}$ is identical to the received $S_{FA}$. After checking $S_{FA}$, MU selects a random number $b$ and computes $bP$, a session key $k_{MF} = h(a'bP)$ using the received $a'P$, the computed $bP$, and $S_{MF} = f_{K_{MF}}(N_{FA} || bP)$. Next, MU sends the message $(bP, S_{MF})$ to FA.

Step 3. $A$ blocks and copies the message $(bP, S_{MF})$ transmitted from MU to FA. It then selects a new random number $b'$ and computes $b'P$, a session key $K_{MF} = h(ab'bP)$ using the copied $aP$ and the computed $b'P$, and $S'_{MF} = f_{K_{MF}}(N_{FA} || b'P)$. Next, $A$ replaces message $(bP, S_{MF})$ with $(b'P, S'_{MF})$ and sends this to FA.

Step 4. FA computes $K_{MF} = h(ab'bP)$ using private and public values and $S''_{MF} = f_{K_{MF}}(N_{FA} || b'P)$. It then checks whether $S''_{MF}$ is identical to the value received for $S_{MF}$. If they are identical, FA authenticates MU. However, the session key between FA and MU is different.

Therefore, Mun et al.'s scheme is vulnerable to man-in-the-middle attack [18].

3.2.3. High Overhead. For authentication, MU sends message $(ID_{HA}, N_{HA}, r_{MU})$ to FA. After receiving this message, FA sends message $(ID_{FA}, N_{FA}, r_{MU})$ to HA. In order to authenticate MU, HA computes $r_{MU} = h(ID_{MU} || PW_{MU}) || ID_{HA}$. To compute $r_{MU}$ for MU, HA must find $ID_{MU}$ and $PW_{MU}$ in its own database to compute the authentication message. However, HA incurs a high overhead because of the difficulty of finding $ID_{MU}$ and $PW_{MU}$ in the authentication message. In addition, HA incurs computational cost because of the one-way hash function and exclusive OR operation used to compute the authentication message. In other words, HA computes the authentication message using $ID_{MU}$ and $PW_{MU}$ in its own database, and incurs a high overhead because it has to compare it with the received authentication message.

4. Our Proposed Scheme

In this section, we propose a secure and efficient anonymous authentication scheme for roaming services in GLOMON-ETs. This scheme consists of three phases: a registration phase, an authentication and key establishment phase, and an update session key phase.
4.1. Notation. Table 1 shows the notation used to describe our proposed scheme.

4.2. Registration Phase. Figure 6 illustrates the procedure of the registration phase. When a new MU wants to register with HA, he/she performs the following steps.

Step R1. Consider MU → HA : {IDMU, NMU}.

MU selects the identity IDMU and a random nonce NMU, and sends IDMU and NMU to HA for registration.

Step R2. Consider HA → MU : {Smart card [IDMU, IDHA, K, N, h(x), h(·)]}.

After receiving the registration message from MU, HA selects a random nonce NHA and computes the following:

\[
A = h(x \parallel y) \oplus h(x \parallel IDMU),
\]
\[
B = h(h(x \parallel IDMU) \parallel h(y)),
\]
\[
PW_{MU} = h(B \parallel NHA),
\]
\[
K = h(x \parallel IDMU) \oplus h(PW_{MU} \parallel NMU).
\]

HA then issues a smart card containing [IDMU, IDHA, A, K, N, h(·)] and delivers it to MU through a secure channel.

4.3. Authentication and Key Establishment Phase. The procedure followed in the authentication and key establishment phase is illustrated in Figure 7. In this phase, to attain mutual authentication between MU and HA, and between MU and FA, the following actions are performed.

Step A1. Consider MU → FA : {IDFA, A, c1, c2, aP, NFMU}.

For authentication, MU selects a random nonce NFMU and a random number a, and computes aP value on E using ECDH. MU then computes the following:

\[
c_1 = K \oplus h(PW_{MU} \parallel NFMU),
\]
\[
c_2 = h(aP \parallel h(PW_{MU} \parallel NFMU)) \parallel h(PW_{MU} \parallel NMU).
\]

Next, MU sends IDFA, A, c1, c2, aP, and NFMU to FA.

Step A2. Consider FA → HA : {IDFA, A, c1, c2, aP, bP, NFMU}.

FA stores the IDHA and aP received from MU for further communication, selects a random number b, and computes the bP value on E using ECDH. FA then sends IDFA, A, c1, c2, aP, bP, and NFMU to HA.

Step A3. Consider HA → FA : {IDFA, A, c1, aP, bP}.

On receiving the authentication message from FA, HA computes the following:

\[
h(x \parallel IDMU) = A \oplus h(x \parallel y),
\]
\[
B' = h(h(x \parallel IDMU) \parallel h(y)),
\]
\[
PW_{MU} = h(B' \parallel NHA),
\]
\[
K = h(x \parallel IDMU) \oplus h(PW_{MU} \parallel NMU),
\]
\[
K' = h(PW_{MU} \parallel NFMU) = c_1 \oplus K
\]
\[
c_2' = h(aP \parallel h(PW_{MU} \parallel NFMU)) \parallel h(PW_{MU} \parallel NMU).
\]

HA then checks whether c2' is identical to c2. If they are identical, HA authenticates MU. HA then computes c3 = h(IDFA \parallel aP \parallel bP \parallel K \parallel h(PW_{MU} \parallel NFMU) \parallel h(PW_{MU} \parallel NMU)) and sends IDFA, IDFA, aP, and bP to FA.

Step A4. FA → MU : {IDFA, IDFA, aP, bP}.

FA checks IDFA, IDFA, and aP, and sends IDFA, IDFA, c3, aP, and bP to MU.

Step A5. MU → FA : {SMFA}.

MU checks IDFA and aP, and computes c3' = h(IDFA \parallel aP \parallel bP \parallel K \parallel h(PW_{MU} \parallel NFMU) \parallel h(PW_{MU} \parallel NMU)). MU checks whether c3' is identical to c3. If they are identical, MU authenticates HA and FA. MU then computes S_{MF} = h(abP) using private and public keys and S_{MF} = \text{f}_{K_{MF}}(IDFA \parallel aP \parallel bP). Next, MU sends S_{MF} to FA.

Step A6. FA computes K_{MF} = h(abP) using private and public keys and S_{MF} = \text{f}_{K_{MF}}(IDFA \parallel aP \parallel bP). FA then checks whether S_{MF} = S_{MF}. If they are identical, FA authenticates MU. Otherwise, the procedure is terminated.

4.4. Update Session Key Phase. The update session key phase is the same as the third phase of Mun et al.'s scheme, as shown in Figure 5.
5. Analyses

5.1. Security Analysis. Table 2 compares the security of existing schemes with that of our proposed scheme. Our scheme has the following security properties.

Anonymity. Assume that an adversary $A$ intercepts the message $\{c_1, c_2, c_3, A\}$ over a public network. An adversary cannot derive the identifier $ID_{MU}$ of the mobile user from $c_1, c_2, c_3,$ and $A$. This is because an adversary does not know $x, y,$ and $PW_{MU}$.

**Perfect Forward Secrecy.** The authentication and key establishment and update session key phases of our scheme use ECDH to provide perfect forward secrecy. To establish a session key, MU and FA use different $a_i P$ and $b_i P$ for each session, and

![Diagram](https://via.placeholder.com/150)

FIGURE 7: Authentication and key establishment phase of our proposed scheme.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Anonymity</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Perfect forward secrecy</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Mutual authentication (MU-HA)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Mutual authentication (MU-FA)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Replay attack</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Impersonation attack</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Disclosure of password</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Man-in-the-middle attack (MU-HA)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Man-in-the-middle attack (MU-FA)</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 2: Security analysis of the compared schemes.
thus they are not related to previous values $a_i \cdot P$ and $b_i \cdot P$. Thus, if the previous session key $K_{MF_i} = h(a_i \cdot b_i \cdot P)$, is disclosed, an adversary $A$ cannot guess $K_{MF} = h(a \cdot P)$. In other words, guessing $K_{MF_i}$ is a computationally difficult problem.

**Mutual Authentication.** HA can authenticate MU by checking $c_2$ in Step A3 of the authentication and key establishment phase, and MU can authenticate HA and FA by checking $c_4$ in Step A5 of the authentication and key establishment phase. And, FA can authenticate MU by checking $S_{MF}$ in Step A6 of the authentication and key establishment phase.

**Impersonation Attack.** An adversary $A$ cannot compute the authentication message $[ID_{HA}, A, c_1, c_2, aP, N_{MU}']$ because he/she cannot know $ID_{MU}, x, y, PW_{MU}$, and $N_{HA}$. Even if $A$ is a legitimate user of HA, he/she cannot compute the authentication message $[ID_{HA}, A, c_1, c_2, aP, N_{MU}']$.

**Disclosure of Password.** We assume that an adversary $A$ eavesdrops on MU's authentication message $[ID_{HA}, A, c_1, c_2, aP, N_{MU}']$ in the authentication and key establishment phase. However, $A$ cannot know MU’s $PW_{MU}$ from the authentication message $[ID_{HA}, A, c_1, c_2, aP, N_{MU}']$ by the nature of a one-way hash function.

**Replay Attacks.** MU uses a random nonce $N_{MU}$ and checks $c_2$ to resist replay attacks in each authentication session. If an adversary $A$ is replaying the previous authentication message, but he/she cannot authenticate from HA because $c_2$ fail to check.

**Man-in-the-Middle Attacks.** Man-in-the-middle attacks are thwarted because of the authentication between MU and HA. Similarly, man-in-the-middle attacks can be thwarted by the establishment of a session key between MU and FA.

### 5.2. Performance Analysis

Table 3 compares the performance of existing schemes with that of our proposed scheme. Our scheme incurs less communication cost than conventional schemes [4–6]. Although our scheme incurs a little more communication cost than Mun et al’s scheme, it incurs less computational overhead in the database than Mun et al’s scheme [12].

**No Need for Time Synchronization.** Conventional schemes use timestamps to resist replay attacks. Thus, time synchronization takes place when each entity is located in a different time zone. However, our scheme does not use timestamps, so there is no need to synchronize time between different entities.

**Use of ECDH.** Conventional schemes use certificates. However, mobile devices have power limitations; low-level computation based on certificates incurs a significant overhead. Our scheme uses ECDH instead of a public key cryptosystem with certificates in order to reduce the communication overhead. ECDH provides the same security properties and uses fewer resources than a public key cryptosystem with certificates. The performance advantage of ECDH is improved further as security needs increase.

**Overhead Analysis.** Our proposed authentication scheme can be compared with Mun et al’s scheme in terms of the database overhead incurred by HA as the number of devices increase. In order to compare the overhead, the following terms are defined: the number of devices is $d$ ($d = 1, 10, 20, \ldots, 100$), the identifier stored in the database of the home agent is $i$, the computational cost for a one-way hash function and exclusive OR operation is $c$ (it is assumed that the computational cost for a one-way hash function and exclusive OR operation is 2, thus, $c = 2$), and, finally, the overhead in the database of the home agent is $O$. Thus, the overhead can be expressed as $O = d \times i \times c$, that is, $O = 10 \times 10 \times 2 = 200$. Mun et al’s scheme must obtain identifier and password information from its own database in order to compute the authentication message. However, their scheme compares the authentication message to compute the identifier and password of all the mobile users stored in its own database because of the difficulty of finding identifier and password information in the authentication message. For example, in Mun et al’s scheme, if the number of devices to be authenticated by HA is 30, the number

### Table 3: Performance analysis of the compared schemes.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$5T(h) + 2T(\oplus)$</td>
<td>$2T(h) + 3T(\oplus)$</td>
<td>$2T(h) + 3T(\oplus)$</td>
<td>$2T(h) + 3T(\oplus)$</td>
<td>$2T(h) + 17T(\oplus)$</td>
</tr>
<tr>
<td>MU</td>
<td>$4T(h) + 1T(\oplus) + 1\text{Asym}$</td>
<td>$2T(h) + 3T(\oplus) + 2\text{Sym}$</td>
<td>$2T(h) + 3T(\oplus) + 2\text{Sym}$</td>
<td>$2T(h) + 3T(\oplus) + 1\text{Sym}$</td>
<td>$5T(h) + 2T(\oplus) + 1\text{Asym}$</td>
</tr>
<tr>
<td>FA</td>
<td>$1T(h) + 1\text{Asym}$</td>
<td>$2T(h) + 3T(\oplus) + 1\text{Sym} + 2\text{Asym}$</td>
<td>$4T(h) + 1T(\oplus) + 2\text{Sym} + 2\text{Asym}$</td>
<td>$5T(h) + 3\text{Asym}$</td>
<td>$4T(h) + 2T(\oplus) + 1\text{Asym}$</td>
</tr>
<tr>
<td>HA</td>
<td>$6T(h) + 3T(\oplus)$</td>
<td>$3T(h) + 1\text{Sym} + 2\text{Asym}$</td>
<td>$3T(h) + 1\text{Sym} + 3\text{Asym}$</td>
<td>$2T(h) + 2\text{Sym}$</td>
<td>$3T(h) + 3T(\oplus)$</td>
</tr>
<tr>
<td>Total</td>
<td>$16T(h) + 6T(\oplus)$</td>
<td>$9T(h) + 7T(\oplus)$</td>
<td>$13T(h) + 7T(\oplus)$</td>
<td>$12T(h) + 4T(\oplus)$</td>
<td>$14T(h) + 8T(\oplus)$</td>
</tr>
</tbody>
</table>

$T(h)$: number of hash operation, $T(\oplus)$: number of XOR operation, Sym: number of symmetric key operation, Asym: number of asymmetric key operation.
of identifiers stored in the database of the home agent is also 30, the computational cost for a one-way hash function and exclusive OR operation is 2 (according to Mun et al’s scheme, $c = 2$ because of the computational cost incurred); therefore, the overhead incurred in the database of HA is $O = 30 \times 30 \times 2 = 1800$. Our proposed scheme can compute the authentication message in its own database because the identifier information can be found in the authentication message. For example, in our proposed scheme, if the number of devices to be authenticated by the home agent is 30, the number of identifiers stored in the database of the home agent is also 30, the computational cost for a one-way hash function and exclusive OR operation is 1 (our proposed scheme does not incur computational cost; thus, $c = 1$), and thus, the overhead incurred in the database of HA is $O = 30 \times 30 \times 1 = 900$. Just like our proposed scheme, Lee et al’s and Wu et al’s scheme are the same overhead analysis. Compared to the existing scheme, our proposed scheme incurs less computational overhead in the database (Figure 8).

6. Conclusion

In this paper, we examined the previous schemes and security vulnerabilities of the previous schemes. Lee et al’s and Wu et al’s scheme was vulnerable to replay attack, cannot achieved perfect forward secrecy, cannot provided anonymity. And Mun et al’s scheme was vulnerable to replay attack and man-in-the-middle attack, and incurred a high overhead in the database. Therefore, we proposed a secure and efficient anonymous authentication scheme for roaming service in GLOMONET. Our scheme was developed using ECDH instead of the authentication mechanism used by Mun et al’s scheme. Consequently, unlike Mun et al’s scheme, our scheme achieves anonymity, provides perfect forward secrecy and mutual authentication, and is resistant to replay attack and man-in-the-middle attack. And our scheme incurs less overhead in the database than Mun et al’s scheme does. In addition, our scheme does not use timestamps, and as a result, it does not need to synchronize time between different entities.

Acknowledgments

This research was funded by the MSIP (Ministry of Science, ICT & Future Planning), Korea in the ICT R&D Program 2013. This work was supported by the Soonchunhyang University Research Fund. The authors declare that there is no conflict of interests regarding the publication of this article.

References


Submit your manuscripts at http://www.hindawi.com