Cryptanalysis of Mun et al.'s anonymous authentication scheme for roaming service in global mobility networks

Hongbin Tang, Xinsong Liu

School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

Corresponding author: Hongbin Tang, Tel: 86-13540872147, Email: tanghongbin@uestc.edu.cn

Abstract: An anonymous user authentication scheme allows the user and the remote server to authenticate each other, and should preserve user anonymity. In 2011, Mun et al. proposed an enhanced secure anonymous user authentication scheme for roaming service in global mobility networks. They claimed that their scheme was more secure and efficient than others. However, we demonstrate that their scheme is vulnerable to the insider, impersonation, server spoofing, and denial of service attacks along with the efficiency and password issues. Meanwhile, it cannot provide any user anonymity. Thus it is not feasible for the real-life implementation.

Keywords: Password; Anonymity; Authentication; Protocol; Cryptography

1. Introduction

The global mobility network (GLOMONET) provides global roaming service for mobile users. It permits users to enjoy the services provided by the home agent in a foreign region. Authentication with anonymity in GLOMONET is an important security issue. In order to achieve user privacy, several authentication schemes with anonymity have been considered for the roaming service in GLOMONET [1-8].

In 2004, Zhu and Ma [1] proposed a new authentication scheme with anonymity for roaming service according to the hash function, smart cards and that mobile users only performed symmetric encryption and decryption. This scheme was efficient due to the hash function operation and symmetric cryptographic algorithm. In 2006, Lee et al. [2] pointed out that Zhu et al.'s scheme could neither provide mutual authentication and perfect forward secrecy property, nor protect against forgery attack. And they suggested an alternative enhanced scheme. In 2008, Wu et al. [3] demonstrated that Lee et al.'s scheme still failed to provide the anonymity and the perfect backward secrecy. Moreover, they introduced an improved scheme by a simple fix. Recently, Zeng et al. [5], Xu et al. [4], and Mun et al. [8] showed that Wu et al.'s scheme failed to achieve the anonymity and other security problems independently.

In 2011, Mun et al. [8] pointed out the security vulnerabilities of Wu et al.'s scheme. They put forward an enhanced anonymous authentication scheme basing on Elliptic Curve Discrete Logarithm Problem (ECDLP) and claimed that their scheme was more efficient than the previous schemes. Unfortunately, we find that Mun et al.'s scheme still suffers to the insider, impersonation, server spoofing, and denial of service (DoS) attacks along with the efficiency and password issues. At the same time, it cannot provide any user anonymity property.

The remainder of this letter is organized as follows. In section 2, we give a brief review of Mun et al.'s scheme. In section 3, we discuss the security flaws in Mun et al.'s scheme. And section 4 is the conclusions.

2. Review of Mun et al.'s anonymous authentication scheme

Before reviewing Mun et al.'s scheme, we introduce some notations defined here in this letter.

2.1. Notations

MU, FA, HA: mobile user, foreign agent, home agent
PW_X: password of an entity X
ID_X: identity of an entity X
D: a uniformly distributed dictionary of size |D|
N_X: number used only once (Nonce) generated by an entity X
$h(\cdot)$: a secure one-way hash function
f_K: MAC generation function by using the key K
K_{XY}: session key between entity X and Y
$A \oplus B$: XOR operation of A and B
$A||B$: data A concatenates with data B

2.2. Mun et al.'s scheme

Mun et al.'s scheme consists of three phases: registration, authentication and establishment of session key, and update session key. These phases work as follows:

2.2.1. Registration phase

When a new MU wishes to register at his/her HA, first phase occurs.

Step 1: MU *sends* a nonce N_{MU} and his/her identity ID_{MU} to HA.

Step 2: First, HA generates a nonce N_{HA}. Then, HA computes PW_{MU} = h($N_{MU} \| N_{HA}$) and r_{MU} = h($ID_{MU} \| PW_{MU}$) $\oplus$ ID_{HA}.

Step 3: HA sends computed r_{MU} and PW_{MU}, N_{HA}, ID_{HA}, and h(.) to MU via secure channel.

2.2.2. Authentication and establishment of session key phase

The procedure of the authentication and establishment of session key phase is as follows.

Step 1: MU sends the request message ID_{HA}, N_{HA} and r_{MU} to FA.

Step 2: FA stores received parameters from MU for further communications and generates a nonce N_{FA}.

Step 3: FA sends ID_{FA}, N_{FA} and r_{MU} to HA.

Step 4: On receiving of the message above, HA computes r'_{MU}=h($ID_{MU} \| PW_{MU}$) $\oplus$ ID_{HA}. Then, HA verifies whether r'_{MU} is equal to r_{MU}. If they are equal, HA authenticates MU. Otherwise, HA considers MU as illegal user. Refer to the HA's security policy, HA cannot provide services to MU and terminate this procedure. Next, HA computes P_{HA} = h($PW_{MU} \| N_{FA}$) and S_{HA} = h($ID_{FA} \| N_{FA}$) $\oplus$ $r_{MU} \oplus P_{HA}$.

Step 5: HA sends S_{HA} and P_{HA} to FA.

Step 6: After FA receives S_{HA} and P_{HA} from HA, FA verifies S_{HA}. First, FA computes S'_{HA}= h($ID_{FA} \| N_{FA}$) $\oplus$ $r_{MU} \oplus P_{HA}$. Then, FA verifies whether S'_{HA} equals the received S_{HA}. If the result is not equal, the procedure is terminated. Then FA computes S_{FA}=h($S_{HA} \| N_{FA} \| N_{HA}$), and selects a random number a, then computes aP on E.

Step 7: FA sends S_{FA}, aP and P_{FA} = ($S_{HA} \| ID_{FA} \| N_{FA}$) to MU.

Step 8: MU checks S_{FA}. As a result, MU can authenticate HA and FA. The procedure of checking S_{FA} is as follows:

– *MU* computes $S'_{HA} = h(ID_{FA} \| N_{FA}) \oplus r_{MU} \oplus h(PW_{MU} \| N_{FA})$.

– *MU* verifies whether S'_{FA} equals S_{FA}. If the result is correct, *MU* can authenticate *FA* and *HA*. Otherwise, the procedure is terminated. After checking S_{FA}, *MU* selects a random number *b* and computes *bP*. Then *MU* computes the session key $K_{MF} = h(abP)$. Moreover, *MU* computes $S_{MF} = f_{K_{MF}}(N_{FA} \| bP)$.

Step 9: *MU* sends *bP* and S_{MF} to *FA*.

Step 10: *FA* computes $K_{FA} = h(abP)$. After computing K_{FA}, *FA* verifies S_{MF}. The procedure of verifying S_{MF} is as follows:

– *FA* computes $S'_{FA} = f_{K_{FA}}(N_{FA} \| bP)$.

– *FA* verifies whether S'_{FA} equals S_{MF}. If the result is not correct, the session key $K_{MF} = K_{FA} = h(abP)$ between *MU* and *FA* is not valid and *FA* terminates the procedure. Otherwise, *FA* can authenticate *MU*.

2.2.3. Update session key phase

MU and *FA* wish to renew the session key K_{MF_i} (i=1, …, n) when the user is always within this *FA*. Thus, *MU* and *FA* can update session key at *i*th session. The procedure of the third phase is described as follows:

Step 1: *MU* selects a new random number b_i and computes b_iP (i = 1, …, n). Then, b_iP is sent to *FA*.

Step 2: *FA* selects a new random number a_i and computes a_iP (i = 1, …, n). *FA* generates a new session key $K_{MF_i} = h(a_ib_iP)$, and then computes $S_{MF_i} = f_{K_{MF_i}}(a_ib_iP \| a_{i-1}b_{i-1}P)$.

Step 3: *FA* sends a_iP and S_{MFi} to *MU*.

Step 4: *MU* computes the session key $K_{MF_i} = h(a_ib_iP)$ by using received a_iP. *MU* computes $S'_{MF_i} = f_{K_{MF_i}}(a_ib_iP \| a_{i-1}b_{i-1}P)$. Then, *MU* checks whether S'_{MF_i} equals the received S_{MF_i}. If the result is correct, *MU* uses the new session key K_{MF_i}. Otherwise, *MU* cannot use it, and inform the result to *FA* and *HA*.

3. Comments on the security pitfalls of Mun et al.'s scheme

In this section, we demonstrate the vulnerability of Mun et al.'s scheme to the insider attack, impersonation attack, server spoofing attack, and DoS attack along with the efficiency and password issues.

3.1. Insider attack

If the password of a user can be derived by the server in the registration protocol, it is called the insider attack. In the registration phase of the Mun et al.'s scheme, the insider of the home agent can get the user's password easily by recording the output values of *HA*. The insider attacker can use this password to access other servers if *MU* has registered at different agents with the same password.

3.2. Impersonation attack

The attacker Eve could launch the impersonation attack against Mun et al.'s scheme. The details are described as follows:

Step 1: Assumed that Eve has recorded the message ID_{HA}, N_{HA}, r_{MU} in some prior transactions.

Step 2: Eve resends ID_{HA}, N_{HA}, r_{MU} to a foreign agent *FA*.

Step 3: Once receiving this message, the agent generates a nonce N_{FA} and sends ID_{FA}, N_{FA}, r_{MU} to

the home agent *HA*. The message is correct and passes the validity check successfully.

Step 4: *HA* sends the response message S_{HA}, P_{HA} to *FA*, where $P_{HA}=h(PW_{MU} \| N_{FA})$ and $S_{HA}=h(ID_{FA} \| N_{FA}) \oplus r_{MU} \oplus P_{HA}$. This message also passes the check successfully.

Step 5: FA computes aP and $S_{FA}=h(S_{HA}||N_{FA}||N_{HA})$, and sends S_{FA}, aP, $P_{FA}=(S_{HA}||ID_{FA}||N_{FA})$ to Eve.

Step 6: After getting the message, Eve selects a random number b, extracts N_{FA} from P_{FA} and computes the following values: $K_E=h(b \cdot (aP))=h(abP)$ and $S_E = f_{K_E}(N_{FA} || bP)$. After that, Eve sends bP and S_E to *FA*.

Step 7: On the receipt of the message, *FA* computes $K_{FA}=h(a \cdot (bP))$ and $S_{MF} = f_{K_{FA}}(N_{FA} || bP)$, and compares S_{MF} with S_E. Finally, *FA* accepts *MU* as the valid user since S_{MF} is equal to S_E.

From the aforementioned description, we know that Mun et al.'s scheme suffers from the impersonation attack.

3.3. Server spoofing attack

In this section we point out that Mun et al.'s scheme is vulnerable to the server spoofing attack. The details are described as follows:

Step 1: Eve selects two random numbers N_1, N_2 and sends the request message ID_{HA}, N_1, N_2 to a foreign agent *FA*.

Step 2: After receiving this message, the foreign agent *FA* sends ID_{FA}, N_{FA}, N_2 to the home agent *HA*. However, this message is intercepted by Eve or her confederate.

Step 3: Eve or the confederate selects a random number N_3, computes $S_{HA}=h(ID_{FA}||N_{FA}) \oplus N_2 \oplus N_3$ and sends S_{HA} and N_3 back to *FA*.

Step 4: The foreign agent *FA* computes $S'_{HA}=h(ID_{FA}||N_{FA}) \oplus N_2 \oplus N_3$ and compares S'_{HA} with S_{HA}. *FA* accepts this message as a valid one since S'_{HA} is equal to S_{HA}. *FA* then computes $S_{FA}=h(S_{HA}||N_{FA}||N_1)$ and sends the message S_{FA}, aP, and $P_{FA}=(S_{HA}||ID_{FA}||N_{FA})$ to Eve.

Step 5: Eve selects a random number b and computes bP, $K_E=h(b \cdot (aP))$, and $S_E = f_{K_E}(N_{FA} || bP)$. Finally, Eve sends the message bP, S_E to *FA*.

Step 6: Upon receiving the message, *FA* computes $K_{FA}=h(a \cdot (bP))$ and $S_{MF} = f_{K_{FA}}(N_{FA} || bP)$, and compares S_{MF} with S_E. Because they are equal, *FA* accepts Eve as a valid user.

From the description above, we know that the foreign agent cannot distinguish the home agent from the fake agent. Thus, Mun et al.'s scheme is susceptible to the server spoofing attack.

3.4. No user anonymity

Consider that a mobile user *MU* roams into the foreign network and sends the login message ID_{HA}, N_{HA} and r_{MU} to the *FA* to access service, the contents of N_{HA} and r_{MU} are for *MU*'s exclusive use and these two values always unchanging in step 1 of the authentication and establishment of session key phase. Thus, user anonymity will not be achieved even though the identity ID_{MU} is still protected. An attacker could easily trace down the relation between the mobile user and the message by comparing (N_{HA}, r_{MU}) with all of the eavesdropped messages in the wireless network.

3.5. DoS attack

In this section, we show that Mun et al.'s scheme suffers from DoS attack in the update session key phase. The details are as follows:

Step 1: Supposed *MU* has just finished the establishment of session key phase or update session key phase. *MU* obtains b_{i-1}, $a_{i-1}b_{i-1}P$, and the session key $K_{MF_{i-1}} = h(a_{i-1}b_{i-1}P)$; *FA* maintains the state with *MU* and also has a_{i-1}, $a_{i-1}b_{i-1}P$, and the session key $K_{FA_{i-1}} = h(a_{i-1}b_{i-1}P)$.

Step 2: Eve masquerades as *MU* to send c_iP to *FA*.

Step 3: Refer to *FA*'s security policy, *FA* selects a random number a_i and computes $K_{FA_i} = h(a_ic_iP)$ and $S_{MF_i} = f_{K_{FA_i}}(a_ic_iP \,||\, a_{i-1}b_{i-1}P)$. *FA* sends a_iP and S_{MF_i} to *MU* and replaces a_{i-1}, $a_{i-1}b_{i-1}P$, and $K_{FA_{i-1}}$ with a_i, a_ic_iP, and K_{FA_i}, respectively.

Step 4: *MU* will discard this message, because it does not ask for updating the session key before.

Step 5: After that, *MU* cannot communicate with *FA* anymore. The procedure is as follows:

- *MU* sends b_iP to *FA*.
- *FA* selects a random number a_{i+1} and computes $K_{FA_{i+1}} = h(a_{i+1}b_iP)$ and $S_{MF_{i+1}} = f_{K_{FA_{i+1}}}(a_{i+1}b_iP \,||\, a_ic_iP)$. *FA* then sends $a_{i+1}P$, $S_{MF_{i+1}}$ to *MU*.
- After receiving the message, *MU* computes $K_{MF_i} = h(a_{i+1}b_iP)$ and $S'_{MF_i} = f_{K_{MF_i}}(a_{i+1}b_iP \,||\, a_{i-1}b_{i-1}P)$, and compares S'_{MF_i} with $S_{MF_{i+1}}$. Because they are not equal, *MU* fails to update his or her session key and informs *FA* and *HA* about it.

From the description mentioned above, we know that Mun et al.'s scheme suffers from DoS attack.

3.6. Pitfalls of Mun et al.'s scheme

3.6.1. Efficiency issue

In the second phase of Mun et al.'s scheme, *MU* sends ID_{HA}, N_{HA}, r_{MU} to *FA* and *FA* sends ID_{FA}, N_{FA}, r_{MU} to *HA* at the beginning. After receiving the request message, *HA* computes $r'_{MU}=h(ID_{MU}||PW_{MU}) \oplus ID_{HA}$ and compares it with the received r_{MU}. However, we are wondering how *HA* can compute r'_{MU} because it is an anonymous authentication scheme and *HA* cannot know to whom r_{MU} belongs from the incoming messages. The only way that *HA* could compute the value is as follows:

HA gets a record (ID_i, PW_i) from its database, computes $r_i=h(ID_i||PW_i) \oplus ID_{HA}$ and compares it with r_{MU}. If $r_i = r_{MU}$, *HA* assures that this message is sent from ID_i. Otherwise, *HA* gets another record from the database and repeats this procedure until it finds the correct identity.

From the description above, we know that Mun et al.'s scheme is inefficient.

3.6.2. Password issue

In Mun et al.'s scheme, the password of the user *MU* is difficult to be remembered since the password is equal to $h(N_{MU}||N_{HA})$ which is a high entropy number. Meanwhile, the user cannot freely change his password since there is no password change mechanism in their scheme. On the other hand, if the user changes his password to an easy-to-remember one, the scheme will be vulnerable to the dictionary attack and cannot provide the user anonymity property. The procedure is as follows:

Step 1: Supposed the message ID_{HA}, N_{HA}, r_{MU} has been recorded by Eve in a previous transaction.

Step 2: Eve computes $r_{MU} \oplus ID_{HA}$ and gets $t=h(ID_{MU}||PW_{MU})$. She then starts a dictionary attack

as follows:

- Eve selects a password candidate PW^* and identity candidate ID^* from two dictionary independently.
- Eve computes $t^*=h(ID^*||PW^*)$ and compares t^* with t. If they are equal, Eve finds the correct password and the identity of *MU*. Otherwise she selects another combination of password and identity and repeats the guessing procedure until she finds the correct one. It is feasible because both the identity and the password of *MU* are short and easy-to-remember words.

From the description above, we know that the password of MU must be written down or stored. It is very inconvenient for users to use such a system.

4. Conclusions

In 2011, Mun et al.'s proposed an anonymous authentication protocol for the roaming service in global mobility networks and demonstrated its immunity against various attacks. Nevertheless, after reviewing their scheme and analyzing its security, we point out that Mun et al.'s scheme suffers from the insider, impersonation, server spoofing, and DoS attacks. Meanwhile, it cannot provide user anonymity. Moreover, the password is difficult to be remembered and the scheme is inefficient. The analyses show that the scheme is insecure and inconvenient for the practical application.

References

1. J. Zhu, J. Ma, A new authentication scheme with anonymity for wireless environments. IEEE Trans. Consum. Electron. 50 (1) (2004) 230-234.
2. C. C. Lee, M. S. Hwang, I.E. Liao, Security enhancement on a new authentication scheme with anonymity for wireless environments, IEEE Trans. Ind. Electron. 53 (5) (2006) 1683-1687.
3. C. C. Wu, W. B. Lee, W. J. Tsaur, A secure authentication scheme with anonymity for wireless communications, IEEE Commun. Lett. 12(10) (2008) 722-723.
4. J. Xu, D. Feng, Security flaws in authentication protocols with anonymity for wireless environments, ETRI J. 31 (4) (2009) 460-462.
5. P. Zeng, Z. Cao, K. K. R. Choo, S. Wang, On the anonymity of some authentication schemes for wireless communications, IEEE Commun. Lett. 13 (3) (2009) 170-171.
6. C. C. Chang, C. Y. Lee, W. B. Lee, Cryptanalysis and improvement of a secure authentication scheme with anonymity for wireless communications, in Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2009, September 2009 (2009) 902 - 904
7. D. J. He, S. Chan, A Secure and Lightweight User Authentication Scheme with Anonymity for the Global Mobility Network, in 2010 13th International Conference on Network-Based Information Systems (NBiS) (2010) 305-312.
8. H. Mun, K. Hanb, Y. S. Lee, C.Y. Yeund, H.H. Choi, Enhanced secure anonymous authentication scheme for roaming service in global mobility networks. Mathematical and Computer Modelling 55 (1-2) (2012) 214-222.