Improved Secure Anonymous Authentication Scheme for Roaming Service in Global Mobility Networks

Jun-Sub Kim and Jin Kwak*

ISAA Lab, Department of Information Security Engineering,
Soonchunhyang University, Korea
**Department of Information Security Engineering Soonchunhyang University, Korea*
jskim0911@sch.ac.kr, jkwak@sch.ac.kr

Abstract

The user authentication scheme in a global mobility network is an important security issue that allows users of mobile devices to access a secure roaming service through wireless networks. Over recent years, many anonymous authentication schemes have been proposed for roaming services in global mobility networks. In 2012, Mun et al. proposed a new enhancement for scheme that uses the Elliptic Curve Diffie–Hellman protocol to overcome security weaknesses and improve performance. However, this has some vulnerabilities, such as in the case of replay attacks and man-in-the-middle attacks. In this paper, we demonstrate the weaknesses of Mun et al.'s scheme to the attacks mentioned above. We also propose an improved secure anonymous authentication scheme for roaming services.

Keywords: *Authentication, Anonymity, Key establishment, Roaming service, Wireless security*

1. Introduction

Wireless communication is the transfer of information between two or more mobile devices (e.g., notebook, cellular phone, PDA, smart phone) that are not physically connected. A special network environment that provides a personal communication user with a global roaming service is referred to as a global mobility network (GLOMONET). Through universal roaming technology, when mobile users attempt to connect to a foreign network, they can access the services provided by their home agent through the foreign network. Before providing services, the foreign agent needs to authenticate the user through the user's home agent [1, 10].

Many user authentication schemes have been proposed for the GLOMONET [1-10]. Recently, Mun et al. showed that the scheme of Wu et al. disclosed the passwords of legitimate users and failed to achieve perfect forward secrecy [9], before proposing a new enhancement for anonymous authentication to overcome these security weaknesses.

In this paper, we analyze Mun et al.'s scheme and find that it is vulnerable to replay attacks and man-in-the-middle attacks. In order to overcome these security weaknesses, we propose an improved secure anonymous authentication scheme that is also resistant to replay attacks and man-in-the-middle attacks.

This study is organized as follows: Section 2 reviews the scheme of Mun et al., and Section 3 demonstrates the security weaknesses mentioned above. In Section 4, we propose an improved secure anonymous authentication scheme, which is analyzed with other schemes in Section 5. Finally, Section 6 presents our conclusions.

2. Review of Mun et al.'s scheme

In this section, we examine the anonymous authentication scheme proposed by Mun et al. [9]. Their scheme consists of three phases: a registration phase, an authentication phase, and an update phase.

2.1. First phase: registration

When a new *MU* wants to register with *HA*, he/she perform the following steps:

Step 1. $MU \rightarrow HA: \{ID_{MU}, N_{MU}\}$

MU sends his/her identity ID_{MU} and nonce N_{MU} to *HA* for registration.

Step 2. *HA* generates nonce N_{HA} and computes $PW_{MU} = h(N_{MU} || N_{HA})$ and $r_{MU} = h(ID_{MU} || PW_{MU}) \oplus ID_{HA}$.

Step 3. $HA \rightarrow MU: \{r_{MU}, ID_{HA}, N_{HA}, PW_{MU}, h(\cdot)\}$

HA sends r_{MU}, ID_{HA}, N_{HA}, PW_{MU}, and $h(\cdot)$ to *MU* through a secure channel.

2.2. Second phase: authentication and establishment of session key

In this phase, for mutual authentication between *MU* and *HA* and between *MU* and *FA*, *MU* performs the following steps:

Step 1. $MU \rightarrow FA: \{ID_{HA}, N_{HA}, r_{MU}\}$

When *MU* accesses the new *FA*, *MU* sends ID_{HA}, N_{HA}, and r_{MU} to *FA*.

Step 2. $FA \rightarrow HA: \{ID_{FA}, N_{FA}, r_{MU}\}$

FA stores the received message from *MU* for further communication and generates nonce N_{FA}. *FA* then sends ID_{FA}, N_{FA}, and r_{MU}.

Step 3. $HA \rightarrow FA: \{S_{HA}, P_{HA}\}$

HA computes $r'_{MU} = h(ID_{MU} || PW_{MU}) \oplus ID_{HA}$ and checks whether r'_{MU} equals the received r_{MU}. If they are equal, *HA* can authenticate *MU*. Next, *HA* computes $P_{HA} = h(PW_{MU} || N_{FA})$ and $S_{HA} = h(ID_{FA} || N_{FA}) \oplus r_{MU} \oplus P_{HA}$ and sends the computed S_{HA} and P_{HA} to *FA*.

Step 4. $FA \rightarrow MU: \{S_{FA}, aP, P_{FA}\}$

FA computes $S'_{HA} = h(ID_{FA} || N_{FA}) \oplus r_{MU} \oplus P_{HA}$ and checks whether S'_{HA} equals the received S_{HA}. *FA* computes $S_{FA} = h(S_{HA} || N_{FA} || N_{HA})$, selects a random number a, and then computes aP on E using the Elliptic Curve Diffie–Hellman (ECDH) protocol. Next, *FA* sends S_{FA}, aP, and $P_{FA} = (S_{HA} || ID_{FA} || N_{FA})$ to *MU*.

Step 5. $MU \rightarrow FA: \{bP, S_{MF}\}$

MU computes $S'_{HA} = h(ID_{FA} || N_{FA}) \oplus r_{MU} \oplus h(PW_{MU} || N_{FA})$ and $S'_{FA} = h(S'_{HA} || N_{FA} || N_{HA})$ and checks whether S'_{FA} equals the received S_{FA}. If they are equal, *MU* can authenticate *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} equals the received S_{MF}. If they are equal, *FA* can authenticate *MU*.

2.3. Third phase: update session key

Step 1. $MU \rightarrow FA: \{b_i P\}$

MU selects a new random number b_i and computes $b_i P (i = 1, 2, ..., n)$. *MU* then sends b_i and $b_i P$ to *FA*.

Step 2. $FA \rightarrow MU: \{a_i P, S_{MF_i}\}$

FA selects a new random number a_i and computes $a_i P (i = 1, 2, ..., n)$. *FA* then computes a new session key $K_{MF_i} = h(a_i b_i P)$ and $S_{MF_i} = f_{K_{MF_i}}(a_i b_i P || a_{i-1} b_{i-1} P)$. Next, *FA* sends $a_i P$ and S_{MF_i} to *MU*.

Step 3. *MU* computes a session key $K_{MF_i} = h(a_i b_i P)$ using the received $a_i P$ and the computed $b_i P$ and $S'_{MF_i} = f_{K_{MF_i}}(a_i b_i P || a_{i-1} b_{i-1} P)$. *MU* then checks whether S'_{MF_i} equals the received S_{MF_i}. If they are equal, *MU* and *FA* use the new session key K_{MF_i}.

3. Weaknesses of Mun et al.'s scheme

Mun et al. claimed that their scheme could resist various known attacks. Unfortunately, we find that their scheme is flawed against replay attacks and man-in-the-middle attacks.

3.1. Replay attack

An attacker *A* can eavesdrop on and record the message $\{ID_{HA}, N_{HA}, r_{MU}\}$ transmitted from *MU* to *FA*. *A* can impersonate *MU* by using the recorded message $\{ID_{HA}, N_{HA}, r_{MU}\}$ as follows:

Step 1. When *A* accesses another new *FA*, *A* sends the recorded message $\{ID_{HA}, N_{HA}, r_{MU}\}$ to *FA*. After receiving this message, *FA* sends the message $\{ID_{FA}, N_{FA}, r_{MU}\}$ to *HA*.

Step 2. *HA* computes r'_{MU} and checks whether r'_{MU} equals the received r_{MU}. If they are equal, *HA* can authenticate *A*. *HA* then computes P_{HA} and S_{HA} and sends the message $\{S_{HA}, P_{HA}\}$ to *FA*. After receiving this message, *FA* computes S'_{HA} and checks whether S'_{HA} equals the received S_{HA}. Next, *FA* sends the message $\{S_{FA}, aP, P_{FA}\}$ to *A*.

Step 3. *A* can compute S'_{FA} and can check whether S'_{FA} equals the received S_{FA}. If they are equal, *A* can authenticate *HA* and *FA*. *A* then computes *bP* and S_{MF} and sends the message $\{bP, S_{MF}\}$ to *FA*. After receiving this message, *FA* computes S'_{MF} and checks whether S'_{MF} equals the received S_{MF}. If they are equal, *FA* can authenticate *A*.

3.2. Man-in-the-middle attack

An attacker *A* can eavesdrop on the message transmitted between *FA* and *MU*. As a result, *A* can successfully mount a man-in-the-middle attack as follows:

Step 1. *A* can block and copy the message $\{S_{FA}, aP, P_{FA}\}$ transmitted from *FA* to *MU*. *A* selects a new random number a' and computes $a'P$, then replaces the 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} equals 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 $a'P$ and 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* then blocks and copies the message $\{bP, S_{MF}\}$ transmitted from *MU* to *FA*. *A* selects a new random number b' and computes $b'P$, a session key $K_{MF} = h(ab'P)$ using the copied aP and the computed $b'P$, and $S'_{MF} = f_{K_{MF}}(N_{FA} || b'P)$. Next, *A* replaces the message $\{bP, S_{MF}\}$ with $\{b'P, S'_{MF}\}$ and sends this to *FA*.

Step 4. *FA* computes $K_{MF} = h(ab'P)$ using private and public values and $S''_{MF} = f_{K_{MF}}(N_{FA} || b'P)$. *FA* then checks whether S''_{MF} equals the received value of S'_{MF}. If they are equal, *FA* can authenticate *MU*. However, the session key between *FA* and *MU* is different.

4. Proposed improved secure anonymous authentication scheme

In this section, we propose an improved, secure, and anonymous authentication scheme for a roaming service on GLOMONET. 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.

Table 1. Notations of our scheme

Notation	Description
MU , FA, HA	Mobile User, Foreign Agent, Home Agent
ID_X	Identity of an entity X
P	Password of mobile user
N / N'	Random nonce for current session / Random nonce for next session
x	Secret key of home agent
y	Random nonce for generates each mobile user
$h(\cdot)$	A one-way hash function
$PRNG(\cdot)$	Pseudo Random Number Generator
$\oplus$	Exclusive OR operation
$\|\|$	Concatenation operation
E_K / D_K	Encryption/Decryption function of symmetric key cryptosystem using key K
f_K	MAC generation function by using the key K
K_{XY}	Session key between entity X and Y
$A \rightarrow B: X$	X is transmitted from A to B

4.2. Registration phase

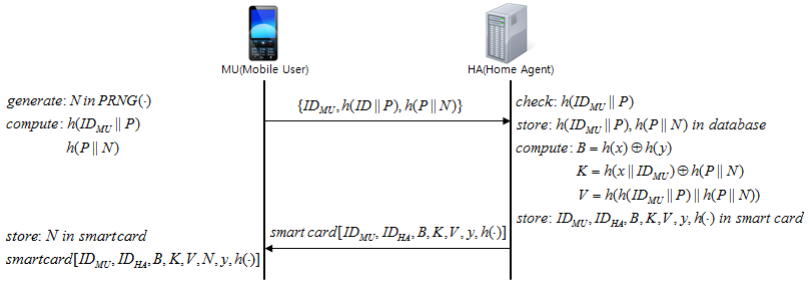

Figure 1. Registration phase in our scheme

Figure 1 illustrates the procedure of the registration phase. When a new *MU* wants to register with the Home Agent *HA*, he/she performs the following steps:

Step 1. $MU \rightarrow HA: \{ID_{MU}, h(ID_{MU}||P), h(P||N)\}$

MU generates a random nonce N using $PRNG(\cdot)$, and computes $h(ID_{MU}||P)$ and the password verifier $h(P||N)$ using ID_{MU}, P, and N. *MU* sends ID_{MU}, $h(ID_{MU}||P)$, and $h(P||N)$ to *HA* for registration.

Step 2. $HA \rightarrow MU: \{Smart\ card[ID_{MU}, ID_{HA}, B, K, V, y, h(\cdot)]\}$

HA stores $h(ID_{MU}||P)$ and $h(P||N)$ in its database after the received $h(ID_{MU}||P)$ is identified. *HA* computes $B = h(x) \oplus h(y)$, $K = h(x||ID_{MU}) \oplus h(P||N)$, and $V = h(h(ID_{MU}||P)||h(P||N))$ using x, y, and ID_{MU}. *HA* then issues a smart card containing $[ID_{MU}, ID_{HA}, B, K, V, y, h(\cdot)]$ and delivers it to *MU* through a secure channel.

Step 3. *MU* stores the random nonce N within the smart card.

4.3. Authentication and key establishment phase

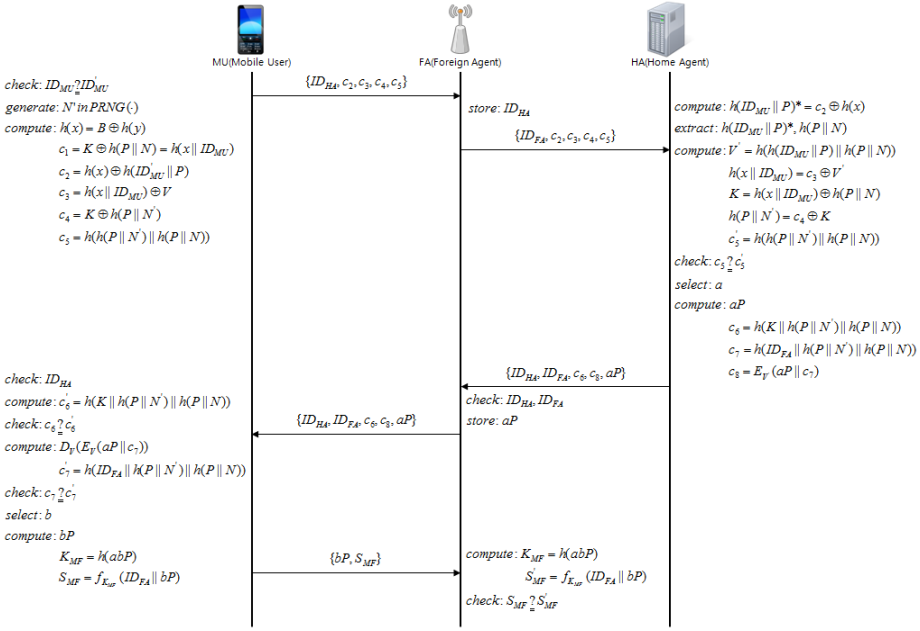

Figure 2. Authentication and key establishment phase in our scheme

The procedure of the authentication and key establishment phase is shown in Figure 2. In this phase, to attain mutual authentication between *MU* and *HA*, and between *MU* and *FA*, *MU* performs the following steps:

Step 1. For authentication, *MU* inserts his/her smart card into the device and inputs identity ID'_{MU} and password P. Then, *MU* checks whether ID'_{MU} equals ID_{MU}. If they are not equal, the procedure is terminated. Next, *MU* generates a random nonce N' using $PRNG(\cdot)$ and computes the following:

$h(x) = B \oplus h(y)$

$c_1 = K \oplus h(P||N) = h(x||ID_{MU})$

$c_2 = h(x) \oplus h(ID_{MU}||P)$

$c_3 = h(x||ID_{MU}) \oplus V$

$c_4 = K \oplus h(P||N')$

$c_5 = h(h(P||N')||h(P||N))$

Step 2. $MU \rightarrow FA: \{ID_{HA}, c_2, c_3, c_4, c_5\}$

MU sends ID_{HA}, c_2, c_3, c_4, and c_5 to *FA*.

Step 3. $FA \rightarrow HA: \{ID_{FA}, c_2, c_3, c_4, c_5\}$

FA stores the received ID_{HA} from *MU* for further communication and sends ID_{FA}, c_2, c_3, c_4, and c_5 to *HA*.

Step 4. After receiving the authentication message from *FA*, *HA* computes $h(ID_{MU}||P)^* = c_2 \oplus h(x)$ and finds $h(ID_{MU}||P)$ corresponding to $h(ID_{MU}||P)^*$ in its database. *HA* then extracts $h(P||N)$ corresponding to $h(ID_{MU}||P)^*$ from the database and computes the following:

$V' = h(h(ID_{MU}||P)||h(P||N))$

$h(x||ID_{MU}) = c_3 \oplus V'$

$K = h(x||ID_{MU}) \oplus h(P||N)$

$h(P||N') = c_4 \oplus K$

$c'_5 = h(h(P||N')||h(P||N))$

HA checks whether c'_5 equals c_5. If they are equal, *HA* can authenticate *MU*. *HA* then selects a random number a and computes a value on *E* using ECDH. Next, *HA* computes the following:

$c_6 = h(K||h(P||N')||h(P||N))$

$c_7 = h(ID_{FA}||h(P||N')||h(P||N))$

$c_8 = E_V(aP||c_7)$

Step 5. $HA \rightarrow FA: \{ID_{HA}, ID_{FA}, c_6, c_8, aP\}$

HA sends ID_{HA}, ID_{FA}, c_6, c_8, and aP to *FA*.

Step 6. $FA \rightarrow MU: \{ID_{HA}, ID_{FA}, c_6, c_8, aP\}$

FA checks the format of ID_{HA} and ID_{FA} and stores aP. Then, *FA* sends ID_{HA}, ID_{FA}, c_6, c_8, and aP to *MU*.

Step 7. *MU* checks the format of ID_{HA} and computes $c_6^{'} = h(K||h(P||N')||h(P||N))$. *MU* then checks whether $c_6^{'}$ equals c_6. If they are equal, *MU* can authenticate *HA*. Next, MU computes $D_V(E_V(aP||c_7))$ and $c_7^{'} = h(ID_{FA}||h(P||N')||h(P||N))$ and checks whether $c_7^{'}$ equals c_7. If they are equal, *MU* can authenticate *FA*. *MU* then selects a random number b and computes bP, $K_{MF} = h(abP)$, and $S_{MF} = f_{K_{MF}}(ID_{FA}||bP)$.

Step 8. $MU \rightarrow FA: \{bP, S_{MF}\}$
MU sends the computed bP and S_{MF} to *FA*.

Step 9. *FA* computes $K_{MF} = h(abP)$ using private and public values and $S_{MF}^{'} = f_{K_{MF}}(ID_{FA}||bP)$, and checks whether $S_{MF}^{'}$ equals S_{MF}. If they are not equal, the procedure is terminated. Otherwise, *FA* can authenticate *MU*.

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 3.

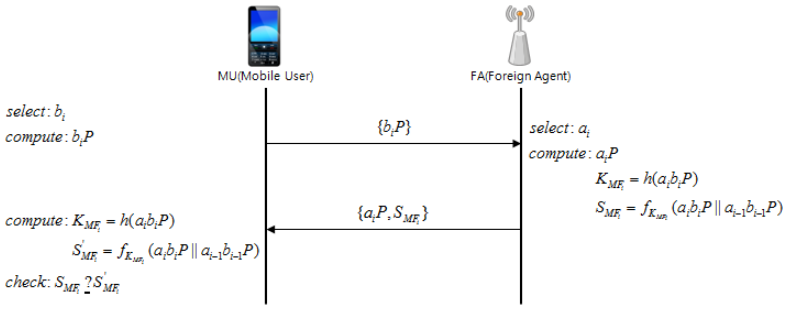

Figure 3. Update session key in our scheme

5. Security analysis

Table 2 compares the security of existing schemes with that of our proposed scheme. Our scheme has several security properties, as follows:

Anonymity: Assume that an attacker *A* intercepts the message $\{c_2, c_3, c_4, c_5\}$ over a public network. *A* cannot derive the identity ID_{MU} of the mobile user from c_2, c_3, c_4, and c_5. This is because *A* does not know x, P, N, and N'.

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_iP and b_iP for each session, and thus they are not related to previous values $a_{i-1}P$ and $b_{i-1}P$. Thus, if the previous session key $K_{MF_{i-1}} = h(a_{i-1}b_{i-1}P)$ is disclosed, an attacker cannot guess $K_{MF_i} = h(a_ib_iP)$. In other words, guessing K_{MF_i} is a computationally difficult problem.

Mutual authentication: *HA* can authenticate *MU* by checking c_5 in **Step 4** of the authentication and key establishment phase, and *MU* can authenticate *HA* by checking c_6 in

Step 7 of the same phase. Similarly, *MU* can authenticate *FA* by checking c_7 in **Step 7** of the authentication and key establishment phase, and *FA* can authenticate *MU* by checking S_{MF} in **Step 9** of the authentication and key establishment phase.

Replay attack: *MU* updates the password verifier $h(P||N)$ to resist replay attacks in each authentication session. The next password verifier is hidden in the previous session, such that is an implicit next password verifier $h(P||N')$. That is, an attacker *A* cannot authenticate the home agent by replaying the previous authentication message.

Man-in-the-middle attack: Man-in-the-middle attacks can be prevented because of the authentication between *MU* and *HA*. Similarly, a man-in-the-middle attack can be prevented by the establishment of a session key between *MU* and *FA*.

Table 2. Analysis of securities

Scheme	Proposed scheme	Mun et al. [9]	Zhu-Ma [3]	Lee et al. [4]	Wu et al. [5]
Anonymity	Yes	Yes	No	No	No
Perfect forward secrecy	Yes	Yes	No	No	No
Mutual authentication(MU-HA)	Yes	Yes	No	No	No
Mutual authentication(MU-FA)	Yes	Yes	No	Yes	Yes
Replay attack	Yes	No	Yes	Yes	Yes
Man-in-the-middle attack(MU-HA)	Yes	Yes	No	No	No
Man-in-the-middle attack(MU-FA)	Yes	No	No	Yes	Yes

6. Conclusion

In this paper, we discussed the security weaknesses in Mun et al.'s scheme, such as a vulnerability to replay attacks and man-in-the-middle attacks. In order to overcome these security weaknesses, we proposed an improved secure anonymous authentication scheme. Our scheme was developed to apply ECDH to Mun et al.'s scheme. Moreover, unlike Mun et al.'s scheme, our scheme achieves anonymity, provides perfect forward secrecy and mutual authentication, and is resistant to replay attacks and man-in-the-middle attacks.

References

[1] S. Suzuki and K. Nakada, IEEE Journal on Selected Areas in Communication, vol. 15, no. 8, **(1997)**, pp. 1608.
[2] L. Buttyán, C, Gbaguidi, S. Staamann and U. Wilhelm, IEEE Transactions on Communications, vol. 48, no. 3, **(2000)**, pp. 373.
[3] J. Zhu and J. Ma, IEEE Transactions on Consumer Electronics, vol. 50, no. 1, **(2004)**, pp. 231.
[4] C.-C. Lee, M.-S. Hwang and I.-E. Liao, IEEE Transactions on Industrial Electronics, vol. 53, no. 5, **(2006)**, pp. 1683.
[5] C.-C. Wu, W.-B. Lee and W.-J. Tsaur, IEEE Communications Letters, vol. 12, no. 10, **(2008)**, pp. 722.
[6] P. Zeng, Z. Cao, K.-K. R. Choo and S. Wang, IEEE Communications Letters, vol. 13, no. 3, **(2009)**, pp. 170.
[7] J.-S. Lee, J. H. Chang and D. H. Lee, IEEE Communications Letters, vol. 13, no. 5, **(2009)**, pp. 292.
[8] C.-C. Chang, C.-Y. Lee and Y.-C. Chiu, Computer Communications, vol. 32, no. 4, **(2009)**, pp. 611.
[9] H. Mun, K. Han, Y. S. Lee, C. Y. Yeun and H. H. Choi, Mathematical and Computer Modelling, vol. 55, no. 1-2, **(2012)**, pp. 214.
[10] D. He and S. Chan, 13th International Conference on Network-Based Information Systems, **(2010)** September 14-16, pp.305-312.

Authors

Jun-Sub Kim

Jun-Sub Kim received his BS and MS degree in Information Security from Soonchunhyang University, South Korea, in 2010 and 2012, respectively. And now he is working as a Ph.D candidate in the Information Security Application and Assurance Lab in the Soochunhyang University. His research interests include cryptographic protocol and cloud computing security.

Jin Kwak

Jin Kwak received his B.S. (2000), M.S. (2003), and Ph.D. (2006) from Sungkyunkwan University (SKKU) in Korea. Prior to joining the faculty at Soonchunhyang University (SCH) in 2007, He joined Kyushu University in Japan as a visiting scholar. After that, he served MIC (Ministry of Information and Communication, Korea) as a Deputy Director. Also, he have served as a Dean of DISE(2009-2010) and Vice-Dean of College of Engineering (2009) in SCH. Now he is a Professor of Department of Information Security Engineering (DISE) at SCH. Also, now he is a Director of SCH BIT Business Incubation Center and a Director of Industry-University & Institute Partnership Division center at SCH. His main research areas are cryptology, information security applications and information assurance.