Research Article
Anonymous Authentication Scheme for Intercommunication in the Internet of Things Environments

Youngseok Chung,1,2 Seokjin Choi,1 and Dongho Won2

1Electronics and Telecommunications Research Institute, Daejeon 305-390, Republic of Korea
2Sungkyunkwan University, Suwon 440-746, Republic of Korea

Correspondence should be addressed to Dongho Won; dhwon@security.re.kr

Received 1 April 2015; Accepted 6 July 2015

Academic Editor: Hongxin Hu

Copyright © 2015 Youngseok Chung et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Authentication and privacy protection are important security mechanisms for keeping things safe in the Internet of Things environments. In particular, an anonymous authentication scheme is a privacy preserving authentication technique which provides both authentication and privacy preservation. An authentication scheme with anonymity in mobility networks was proposed recently. However, it was proven that it failed to provide anonymity against passive adversaries and malicious users and security against known session key attacks and side channel attacks. We propose an anonymous authentication scheme for intercommunication between the things in the Internet of Things environments. The proposed scheme provides not only anonymity and security, but also untraceability for the thing. Moreover, we only use low cost functions, such as hash functions and exclusive-OR operations in consideration of limited computing power of the thing.

1. Introduction

As the Internet of Things (IoT) market is expanding recently, the importance of trustworthy IoT products and services is increasing. Zhang et al. [1] discussed IoT security issues just in time. To preserve IoT security, authentication and privacy preservation are some of the major mechanisms and a few authentication schemes in IoT are proposed [2, 3]. And improvements and drawbacks of those schemes are presented [4–6]. Since a lot of systems and things in IoT environments are connected to the Internet for intercommunication, apparently they can be exposed to hacker’s attacks anytime and anywhere.

Systems of IoT infrastructure, such as a computer, a gateway, and a server system, have high computing power, enough memory, and wide communication bandwidth. On the contrary, things like a tag, a sensor, and a device deployed at the end of IoT environments have limited computing capacity and resource. Therefore, the convergence of symmetric and asymmetric cryptosystems and lightweight cryptographic protocols is necessary to guarantee the whole IoT security. Various things interconnected in IoT environments produce a lot of transmitting data. These data may contain user’s behavior, taste, and preference as well as private information. Hence, an adversary can abuse user’s information or do some damage through tapping, leaking, modifying, and destroying them. It is clear that privacy protection is one of the most important issues of IoT security. Because only human beings can have privacy, privacy preservation is not originally applied to things. However, things also need to have security measures to prevent privacy invasion in case they deal with human beings’ information. For such a reason, anonymity and anonymous authentication are essential to preserve privacy and security of all objects in IoT environments.

Many anonymous authentication schemes in wireless environments, proofs of weaknesses, and countermeasures have been proposed [7–14]. Particularly, Chang et al. [15] proposed an efficient authentication scheme with anonymity for roaming service in global mobility networks. Since their scheme only uses low cost functions, such as hash functions and exclusive-OR operations, it is suitable for battery-powered mobile devices. However, Youn et al. [16] showed
that Chang et al.'s scheme does not provide anonymity and security in certain situations. In other words, it fails to preserve anonymity against passive adversaries and malicious users as well as security against known session key attacks and side channel attacks. Since then, to make up the security faults proven by Youn et al., a few enhanced schemes have been proposed [17–19]. Also, anonymous authentication schemes withunlinkability that guarantee not to link one user and his/her messages have been followed [20–22].

In this paper, we suppose that IoT components are a thing, a gateway, and a registration server. The things, handles, and transmits various information. All things need to be registered for the registration server before being deployed in IoT environments. The registration server keeps and manages the thing's identity. Also, it maintains a trustworthy relationship with the gateway to respond to gateway’s request for authenticating the thing. No one but the thing and the registration server can know the thing’s identity because the thing certainly knows its identity and the registration server takes the thing’s identity when the thing registers for it. The gateway authenticates the thing in relation to the registration server without knowing the thing’s identity. And it makes the thing be able to communicate with other things through a secure way. The thing accomplishes mutual authentication and key agreement with the gateway through an anonymous authentication method when it connects to the gateway for the first time. At this time, the gateway which does not know the thing’s identity can authenticate the thing with the aid of the registration server. Centering around one gateway, a number of things consist of their own domain. Things in the same domain can do anonymous authentication and session key sharing through the same gateway.

We propose a novel authentication scheme that guarantees anonymity, security, and untraceability in IoT environments by remedying the defects of Chang et al.’s scheme which are shown by Youn et al. The proposed scheme makes the things in IoT environments communicate with each other safely. Also, it only uses low cost functions to reduce computing loads of the thing’s side as Chang et al.’s scheme does.

The remainder of this paper is organized as follows. We review the related works in Section 2 and present our scheme in Section 3. In Section 4, we analyze security of our scheme. Finally, a concluding remark is given in Section 5.

2. Related Works

In this section, we review Chang et al.'s scheme and Youn et al.’s analysis. Table 1 denotes notations in Chang et al.'s scheme.

### Table 1: Notations in Chang et al.'s scheme.

| Notations | Descriptions |
|-----------|--------------|
| MN        | Mobile user  |
| HA        | Home agent of a mobile user |
| FA        | Foreign agent of a foreign user |
| ID<sub>X</sub> | Identity of an entity X |
| PW<sub>X</sub> | Password of an entity X |
| SID       | Shadow identity |
| SK        | Session key |
| x         | Private key of HA |
| K<sub>XY</sub> | Common secret key shared between an entity X and an entity Y |
| h()       | Collision-free one-way hash function |
| ‖         | Concatenation |
| ⊕         | Exclusive-OR operation |

2.1. Chang et al.’s Scheme. Chang et al.’s scheme consists of three phases: registration, authentication, and session key establishment. And three entities are involved: the mobile user MN, the foreign agent FA, and the home agent HA in each phase. Their assumption is that each FA and each HA share and store a long term session key using any secure method, such as the Diffie-Hellman key agreement protocol.

#### 2.1.1. Registration Phase.

In the registration phase, MN submits his/her identity ID<sub>MN</sub> and the selected password PW<sub>MN</sub> to HA. HA uses its private key to compute the secret value R as follows:

\[
R = h(ID_{MN} \ || \ x) \oplus PW_{MN}. \tag{1}
\]

Then HA issues a smart card containing \{ID<sub>MN</sub>, ID<sub>HA</sub>, R, h(x), h(\cdot)\} and delivers it to MN in a secure way.

#### 2.1.2. Authentication Phase.

It is assumed that MN wants to take a roaming service from FA. Before providing services, FA tries to authenticate MN through HA. To achieve this, MN, FA, and HA perform the authentication phase as follows.

1. MN inserts his/her smart card into the device and enters a password PW<sub>MN</sub>.
2. Smart card generates a nonce \(n_{MN}\) and computes the parameter \(C\) as shown in

\[
C = (R \oplus PW_{MN}) \oplus n_{MN}. \tag{2}
\]

3. MN sends the login message \(m_1 = \{\text{Login request}, n_{MN}, ID_{HA}\}\) to FA. “Login request” is the header of the message to establish a new secure session between MN and FA.
4. Upon receiving \(m_1\), FA stores \(n_{MN}\) and generates a random nonce \(n_{FA}\) to send the authentication request message \(m_2 = \{\text{Authenticator request}, n_{FA}, ID_{FA}\}\) to HA. The message header “Authenticator request” notifies HA to authenticate MN.
5. After receiving \(m_2\), HA checks ID<sub>FA</sub> to see whether a trustworthy relationship between HA and FA is built or not. If ID<sub>FA</sub> is valid, HA generates a nonce \(n_{HA}\) randomly and sends the message \(m_3 = \{n_{HA}, ID_{HA}\}\) to FA.
6. Upon receiving \(m_3\), FA sends the message \(m_4 = \{n_{HA}, n_{FA}, ID_{FA}\}\) to MN.
After checking the identities and exchanging nonces between MN, FA, and HA, they accomplish the following steps for anonymous authentication.

(7) MN who receives \( m_4 \) computes the shadow identity SID, the parameter \( V_3 \), the session key SK, and the parameter \( V_2 \) sequentially such that

\[
\begin{align*}
\text{SID} &= ID_{MN} \oplus h (h(x) \parallel n_{HA}), \\
V_1 &= h(n_{HA} \parallel C), \\
SK &= h(h(x) \parallel ID_{MN} \parallel ID_{FA} \parallel n_{MN} \parallel n_{FA}), \\
V_2 &= SK \oplus h(n_{HA} \parallel ID_{MN}).
\end{align*}
\]

(8) MN computes the following hashing value \( S_1 \) and sends the message \( m_5 = \{ \text{SID}, V_1, V_2, n_{MN}, S_1, ID_{HA} \} \) to FA:

\[
S_1 = h(n_{FA} \parallel SID \parallel V_1 \parallel V_2 \parallel n_{FA}).
\]

(9) Upon receiving \( m_5 \), FA checks \( S_1 \) and computes the following hashing value \( S_2 \). And then it sends the message \( m_6 = \{ \text{SID}, V_1, V_2, n_{MN}, S_1, ID_{FA} \} \) to HA:

\[
S_2 = h(K_{FH} \parallel n_{HA} \parallel SID \parallel V_1 \parallel V_2 \parallel n_{MN}).
\]

(10) After receiving \( m_6 \), HA checks ID_{FA} and \( S_2 \) firstly. And it checks the format of MN's identity after computing ID_{MN} as follows:

\[
ID_{MN} = \text{SID} \oplus h (h(x) \parallel n_{HA}).
\]

(11) HA computes \( C^* \) and \( V_1^* \) as demonstrated in (7) to check whether \( V_1^* \) is equal to \( V_1 \). The equivalence between \( V_1^* \) and \( V_1 \) implies that the selected \( PW_{MN}^* \) in the registration phase is the same as the password \( PW_{MN} \) that MN enters in the authentication phase:

\[
\begin{align*}
V_1^* &= h(n_{FA} \parallel C^*), \\
C^* &= h(ID_{MN} \parallel x) \parallel n_{MN}, \\
K_1 &= SK \oplus h(K_{FH} \parallel n_{FA}), \\
V_2 &= h(ID_{FA} \parallel h(x) \parallel n_{MN}), \\
S_3 &= h(K_{FH} \parallel n_{FA} \parallel K_1 \parallel V_3).
\end{align*}
\]

2.1.3. Session Key Establishment Phase. After completing the authentication phase, MN and FA continue the following session key establishment phase.

(1) Upon receiving \( m_5 \), FA obtains the following session key SK from \( K_1 \) after checking \( S_3 \):

\[
SK = K_1 \oplus h(K_{FH} \parallel n_{FA}).
\]

And it computes the parameter \( K_2 \) as follows to send the message \( m_8 = \{ V_3, K_2 \} \) to MN:

\[
K_2 = SK \oplus h(SK \parallel n_{MN}).
\]

(2) After receiving \( m_8 \), MN computes the following \( V_3^* \) and checks whether \( V_3^* \) and \( V_3 \) are equal or not:

\[
V_3^* = h(ID_{FA} \parallel h(x) \parallel n_{MN}).
\]

The fact that two values are equal means MN confirms FA as a valid foreign agent.

(3) MN computes \( SK^* \) as follows: from \( K_2 \) to check the equivalence between \( SK^* \) and SK that MN computes in the authentication phase:

\[
SK^* = K_2 \oplus h(SK \parallel n_{MN}).
\]

According to the sameness of two values, MN confirms that he/she shares the authenticated session key with FA.

(4) Since then, MN and FA use \( SK \) when they want to communicate with one another through a secure channel.

Figure 1 describes the authentication and the session key establishment phases in Chang et al's scheme.

2.2. Youn et al’s Analysis. Youn et al. proved that Chang et al’s scheme suffers from some specific attacks. Their assumptions are as follows. Anyone can eavesdrop transmitting messages between MN, FA, and HA. And it is easy to perform a brute-force search of a valid identity since it is short and has a certain format. In addition, there exists an adversary who can execute side-channel attacks.

2.2.1. Anonymity against Passive Adversaries. A passive adversary can obtain \( V_2 \) and \( K_2 \) by eavesdropping messages between MN, FA, and HA. The adversary chooses a candidate identity \( ID' \) to compute \( SK' \) and \( K_2' \) as follows:

\[
\begin{align*}
SK' &= V_2 \oplus h(n_{HA} \parallel ID'), \\
K_2' &= SK' \oplus h(SK' \parallel n_{MN}).
\end{align*}
\]

If \( K_2' \) and \( K_2 \) are equal, the adversary can recognize that the candidate identity is the same as MN's real identity. Therefore, the adversary can know MN's identity.

2.2.2. Anonymity against Malicious Mobile User. It is supposed that MN' is a malicious mobile user who has a valid smart card issued by HA. MN' accomplishes the normal authentication phase with HA to get the nonce \( n'_{HA} \) generated by HA. And then he/she computes SID' using his/her identity \( ID'_{MN} \) as follows:

\[
\text{SID}' = ID'_{MN} \oplus h(h(x) \parallel n'_{HA}).
\]
MN' replaces $n_{HA}$ included in the message $m_4 = \{n_{HA}, n_{FA}, ID_{FA}\}$ between MN and FA with $n'_{HA}$. Then MN who receives $m_4$ computes the following SID to send it to FA:

$$SIF = ID_{MN} \oplus h(h(x) \parallel n'_{HA}).$$  \hspace{1cm} (15)

MN' eavesdrops SID once again to compute $ID_{MN}$ using SID and SID'. $ID_{MN}$ is calculated as follows:

$$ID_{MN} = SID \oplus SID' \oplus ID'_{MN}$$

$$= ID_{MN} \oplus h(h(x) \parallel n'_{HA}) \oplus ID'_{MN}$$  \hspace{1cm} (16)

Thus, a legal but malicious user can obtain other's identity.
2.2.3. Security against Known Session Key Attacks. An adversary can get a user’s identity when a former session key shared between MN and FA is revealed. The adversary obtains $V_2$ by eavesdropping and computes the verification value $Flg$ using $V_2$ and a revealed SK such that

$$Flg = V_2 \oplus SK.$$  \hfill (17)

The adversary also computes another verification value $Flg'$ after choosing a candidate identity $ID'$ in the following manner:

$$Flg' = h(n_{HA} \parallel ID').$$ \hfill (18)

If $Flg$ equals $Flg'$, the adversary can ensure that the guessed identity and the MN’s real identity are consistent.

2.2.4. Security against Side Channel Attacks. Let us suppose that an adversary who can execute side channel attacks obtains a valid smart card issued by HA. The adversary can acquire $h(x)$ from the smart card and take SID and $n_{HA}$ from messages in the authentication phase. Then it is possible that the adversary discovers MN’s identity by computing as follows:

$$ID_{MN} = SID \oplus h(n_{HA} \parallel n_{HA}).$$ \hfill (19)

Hence, Chang et al.’s scheme can be totally broken by side channel attacks.

3. Proposed Scheme

To provide anonymity and security, the proposed scheme that improves Chang et al.’s scheme makes up for the weak points shown by Youn et al. Also, it guarantees untraceability for things by using different anonymized identities all the time. Since it minimizes the number of transmitting messages and computations, it is suitable for IoT environments. Figure 2 presents IoT network environments applied to our scheme. A registration server and a gateway accomplish mutual communication through the Internet while they preserve a trustworthy relationship. The registration server maintains secret information of things. The gateway forms its own domain to combine the things that existed in its permissible range. And it makes the things link up with their registration server. The things collect and produce various information and exchange data one another if necessary. Depending on the number of things in a domain, several base stations are required to provide stable communication channel. Base stations do not participate in the join phase directly. They just act as mediators between the things and the gateway. If the stability of communications channel is guaranteed, a lot of things can work in one domain. It is because the thing only computes low cost functions, while relatively high cost computations are up to the gateway. In other words, a domain can be expanded regardless of the number of things.

The proposed scheme consists of three phases: registration, join, and communication. In each phase, the thing $T$, the gateway $G$, and the registration server $R$ are involved. We
In the registration phase, $T$ registers for $R$ in a safe way. $T$ generates a timestamp $t_T$ to compute the following shadow identity $\text{SID}_T$ using $\text{VID}_T$ and $h(x_R)$:

$$
\text{SID}_T = \text{VID}_T \oplus h(x_R) \oplus t_T.
$$

(21)

The virtual identity already concealed the thing's real identity. The virtual identity is essential element to make the proposed scheme provide anonymity.

3.2 Join Phase. Let us suppose $T$ wants to join $G$'s domain in order to take services from $G$. Then $G$ tries to authenticate $T$ through $R$ to provide services. We define $K_{RG}$ as the common secret key shared between $R$ and $G$ in advance. $T$, $G$, and $R$ perform the following authentication and key establishment steps. Figure 4 presents the join phase.

(1) $T$ generates a timestamp $t_T$ to compute the following shadow identity $\text{SID}_T$ using $\text{VID}_T$ and $h(x_R)$:

$$
\text{SID}_T = \text{VID}_T \oplus h(x_R) \oplus t_T.
$$

(21)

(2) $T$ sends the join request message $m_1 = \{\text{JoinReq}, \text{SID}_T, x_T, t_T\}$ to $G$, where the message header JoinReq means $T$ requests $R$ to authenticate $G$ and to provide a common secret key for joining $G$.

(3) After receiving $m_1$, $G$ checks $t_T$ and generates a timestamp $t_G$. And it computes the hashing value $H_1$ using the preshared common secret $K_{RG}$ as follows:

$$
H_1 = h(K_{RG} \parallel \text{SID}_T \parallel t_T \parallel t_G).
$$

(22)
And it sends the authentication request message \( m_2 = \{ \text{AuthReq, SID}_T, \text{ID}_G, H_1, t_T, t_G \} \) to \( R \). The message header AuthReq indicates \( G \) requests \( R \) to authenticate \( T \).

(4) Upon receiving \( m_2 \), \( R \) checks \( \text{ID}_G \) to determine whether it is an ally or not and checks \( t_G \) to verify the validity of time period. And it computes the following \( H_1^* \) to check whether it is equal to \( H_1 \):

\[
H_1^* = h(K_{RG} \| \text{SID}_T \| t_T \| t_G)
\]  

(23)

If results are all valid, \( R \) generates a timestamp \( t_R \).

(5) \( R \) computes \( \text{VID}_T^* \) using received information and \( h(x_R) \) as follows:

\[
\text{VID}_T^* = \text{SID}_T \oplus h \left( h(x_R) \| t_T \right)
\]

(24)

And it retrieves the secret information \([\text{ID}_T, \text{VID}_T, x_T, t_I]\) from its database using \( \text{VID}_T^* \) as a keyword. If it is impossible to get \( \text{VID}_T^* \) same as \( \text{VID}_T^* \), then \( R \) rejects \( G \)'s request for authentication and stops communication with \( G \).
(6) $R$ computes $T$'s new virtual identity $VID'_T$ as shown in (25) and stores it safely to substitute the existing virtual identity $VID_T$:

$$VID'_T = h(ID_T \parallel x_T \parallel t_R).$$

(25)

This step makes every virtual identity supposed to be used only once. In other words, it guarantees the freshness of all virtual identities and it finally provides unlinkability of the proposed scheme.

(7) After the renewal of the virtual identity, $R$ computes the common secret key supposed to be shared between $G$ and $T$ using $T$'s private key $x_T$. Also, it computes the verification values $V_1, V_2, V_3,$ and $V_4$ for $T$ and the verification value $V_5$ for $G$ as follows:

$$K_{GT} = h(x_T) \parallel ID_T \parallel ID_G \parallel t_T \parallel t_G,$$

$$V_1 = h(x_T) \parallel ID_G \parallel t_T,$$

$$V_2 = h(K_{GT} \parallel ID_G \parallel t_T),$$

$$V_3 = h(VID'_T \parallel ID_G \parallel t_T),$$

$$V_4 = K_{GT} \oplus h(K_{RG} \parallel t_G).$$

Since the common secret key $K_{GT}$ is derived from a secret hashing value $h(x_T)$ and changing timestamps, the secrecy and the freshness of the common secret key are always guaranteed.

(8) $R$ computes the following hashing value $H_2$ and sends the message $m_3 = \{ID_R, V_1, V_2, V_3, V_4, H_2, t_R\}$ to $G$:

$$H_2 = h(K_{RG} \parallel V_1 \parallel V_2 \parallel V_3 \parallel V_4 \parallel t_R).$$

(27)

(9) After receiving $m_3$, $G$ checks $t_R$ and $H_2$. If they are all valid, $G$ computes the verification value $V_5$ after computing the common secret key $K_{GT}$. And it sends the message $m_5 = \{ID_G, V_1, V_2, V_3, V_4, V_5, t_G\}$ to $T$. $K_{GT}$ and $V_5$ are as shown in the following formulas:

$$K_{GT} = V_4 \oplus h(K_{RG} \parallel t_G),$$

$$V_5 = K_{GT} \oplus h(K_{GT} \parallel t_T).$$

(28)

(10) Upon receiving $m_4$, $T$ computes the following $V_1^*$ after checking $t_G$:

$$V_1^* = h(x_T) \parallel ID_G \parallel t_T.$$  

(29)

The fact that $V_1^*$ is the same as $V_1$ implies that $T$ recognizes the fact that $R$ checked $ID_G$.

(11) $T$ computes $K_{GT}^*$ and $V_2^*$ such that

$$K_{GT}^* = h(x_T) \parallel ID_T \parallel ID_G \parallel t_T \parallel t_G,$$

$$V_2^* = h(K_{GT}^* \parallel ID_G \parallel t_T).$$

(30)

to check the equivalence between $V_2^*$ and $V_2$. The sameness of two values means that $V_2$ is equal to $V_2^*$. The equality indicates $R$ computes $T$'s new virtual identity $VID'_T$ normally. And then $T$ stores its new virtual identity $VID'_T$ securely. As a result, the virtual identity is renewed by replacing the existing $VID_T$ with the new $VID'_T$.

(12) $T$ checks the equivalence between $V_5^*$ and $V_5$ after computing $V_5^*$ as shown in

$$V_5^* = K_{GT}^* \oplus h(K_{GT}^* \parallel t_T).$$

(32)

The valid verification of $V_5^*$ implies that $T$ and $G$ share the common secret key successfully.

3.3. Communication Phase. The things located in $G$'s domain authenticate each other and share session keys to communicate in a secure way. We assume that $T_1$ and $T_2$ already shared common secret keys called $K_{GT_1}$ and $K_{GT_2}$ respectively, with $G$ in the join phase. $T_1$ performs the following communication phase to share a session key with $T_2$. The communication phase is presented in Figure 5.

(1) $T_1$ generates a timestamp $t_{T_1}$ and computes the new shadow identity $SID'_{T_1}$ as shown in

$$SID'_{T_1} = VID_{T_1} \oplus h(x_R) \parallel t_{T_1}.$$  

(33)

to substitute the existing shadow identity $SID_{T_1}$.

(2) $T_1$ computes the verification value $V_6$ and the hashing value $H_3$ as follows:

$$V_6 = SID'_{T_1} \oplus h(K_{GT_1} \parallel t_{T_1}),$$

$$H_3 = h(K_{GT_1} \parallel SID_{T_1} \parallel V_6 \parallel t_{T_1}).$$

(34)

And it sends the communication request message $m_5 = \{CommReq, SID'_{T_1}, ID_G, V_6, H_3, t_{T_1}\}$ to $T_2$. The message header $CommReq$ stands for $T_1$'s request for communicating with $T_2$.

(3) Upon receiving $m_5$, $T_2$ generates a timestamp $t_{T_2}$ after checking $t_{T_1}$ and computes its new shadow identity $SID'_{T_2}$, the verification value $V_7$, and the hashing value $H_4$ such that

$$SID'_{T_2} = VID_{T_2} \oplus h(x_R) \parallel t_{T_2},$$

$$V_7 = SID'_{T_2} \oplus h(K_{GT_2} \parallel t_{T_2}),$$

$$H_4 = h(K_{GT_2} \parallel V_7 \parallel V_5 \parallel H_3 \parallel SID_{T_1} \parallel SID_{T_2} \parallel t_{T_1} \parallel t_{T_2}).$$

(35)
After receiving $m_6$, $G$ firstly checks $t_{T1}$, SID$_{T1}$, and SID$_{T2}$, to confirm that $T_1$ and $T_2$ were authenticated by itself in the join phase. And then it checks that $H_3^*$ is equal to $H_3$, and $H_4^*$ is equal to $H_4$ after computing the following $H_3^*$, $H_4^*$:

\[
\begin{align*}
H_3^* &= h(K_{GT1}, \| SID_{T1} \| V_6 \| t_{T1}), \\
H_4^* &= h(K_{GT2}, \| V_6 \| \| V_7 \| \| H_3 \| \| SID_{T1} \| \| t_{T1} \| \| t_{T2})).
\end{align*}
\]

(37)

And it sends the authentication request message $m_6$ = {AuthReq, SID$_{T1}$, SID$_{T2}$, $V_6$, $V_7$, $H_3$, $H_4$, $t_{T1}$, $t_{T2}$} to $G$. The message header AuthReq means that $T_2$ requests $G$ to authenticate $T_1$.

(4) After receiving $m_6$, $G$ firstly checks $t_{T1}$, SID$_{T1}$, and SID$_{T2}$, to confirm that $T_1$ and $T_2$ were authenticated by itself in the join phase. And then it checks that $H_3^*$ is equal to $H_3$, and $H_4^*$ is equal to $H_4$ after computing the following $H_3^*$, $H_4^*$:

\[
\begin{align*}
H_3^* &= h(K_{GT1}, \| SID_{T1} \| V_6 \| t_{T1}), \\
H_4^* &= h(K_{GT2}, \| V_6 \| \| V_7 \| \| H_3 \| \| SID_{T1} \| \| t_{T1} \| \| t_{T2})).
\end{align*}
\]

After computing $H_3^*$, $H_4^*$, $G$ generates $t_{T2}$, SID$_{T2}$, $V_6$, and $H_4$ using Equation (37).

(5) If all the hashing values are verified successfully, $G$ computes the new shadow identities of $T_1$ and $T_2$ using $V_6$, $V_7$, $K_{GT1}$, and $K_{GT2}$. The new shadow identities SID'$_{T1}$ and SID'$_{T2}$ are calculated from

\[
\begin{align*}
\text{SID}'_{T1} &= V_6 \oplus h(K_{GT1}, \| t_{T1}), \\
\text{SID}'_{T2} &= V_7 \oplus h(K_{GT2}, \| t_{T2}).
\end{align*}
\]

(37)

And it replaces the existing shadow identities SID$_{T1}$ and SID$_{T2}$ with SID'$_{T1}$ and SID'$_{T2}$. It stores SID'$_{T1}$ and SID'$_{T2}$ securely in its database. According to this step, the freshness of all shadow identities is guaranteed.
As a result, it preserves unlinkability of the proposed scheme.

(6) $G$ generates a timestamp $t_{G\tau_1}$ and computes the session key $SK$ and the parameters $K_1$ and $K_2$ using $K_{G\tau_1}$ and $K_{G\tau_2}$ as follows:

$$SK = h \left( K_{G\tau_1} \parallel K_{G\tau_2} \parallel SID_{T_1} \parallel SID_{T_2} \parallel t_{T_1} \parallel t_{T_2} \right).$$

$$K_1 = SK \oplus h \left( K_{G\tau_1} \parallel SID_{T_1} \parallel t_{T_1} \right).$$

$$K_2 = SK \oplus h \left( K_{G\tau_2} \parallel SID_{T_2} \parallel t_{T_2} \right).$$

(38)

(7) $G$ computes the verification and the hashing values, such as $V_8$, $V_9$, $H_5$, and $H_6$ as shown in

$$V_8 = h \left( SK \parallel ID_G \parallel t_{T_1} \right),$$

$$V_9 = h \left( SK \parallel ID_G \parallel t_{T_1} \right),$$

$$H_5 = h \left( K_{G\tau_1} \parallel K_1 \parallel V_8 \parallel t_{G\tau_1} \right),$$

$$H_6 = h \left( K_{G\tau_1} \parallel K_2 \parallel V_9 \parallel t_{G\tau_2} \right).$$

And then it sends the messages $m_7 = \{ID_G, K_2, V_9, H_6, t_{G\tau_1}\}$ to $T_2$ and $m_8 = \{ID_G, K_1, V_8, H_5, t_{G\tau_2}\}$ to $T_1$.

(8) Upon receiving $m_7$, $T_2$ checks $t_{G\tau_1}$ and $H_6$ and computes the following session key $SK^*$ shared with $T_1$ using the common secret key $K_{G\tau_2}$ and the received $K_2$:

$$SK^* = K_2 \oplus h \left( K_{G\tau_2} \parallel SID_{T_1} \parallel t_{T_1} \right).$$

(40)

After that, it checks the equivalence between the computed $V_9^*$ and the received $V_9$, where $V_9^*$ is calculated by the following equation:

$$V_9^* = h \left( SK^* \parallel ID_G \parallel t_{T_1} \right).$$

(41)

If the result is valid, $T_2$ recognizes the fact that $G$ computed the session key $SK$.

(9) Meanwhile, after receiving $m_8$, $T_1$ also checks the validities of $t_{G\tau_2}$ and $H_5$ at first. Then it computes the session key $SK^*$ using $K_{G\tau_1}$ and $K_1$ and the verification value $V_8^*$ using $SK^*$ as follows:

$$SK^* = K_1 \oplus h \left( K_{G\tau_1} \parallel SID_{T_1} \parallel t_{T_1} \right),$$

$$V_8^* = h \left( SK^* \parallel ID_G \parallel t_{T_1} \right).$$

(42)

$T_1$ confirms the sameness of $V_8^*$ and $V_8$ to check the session key $SK$ computed by $G$. Finally, $T_1$ shares the session key $SK$ with $T_2$ successfully.

4. Security Analysis

In this section, we prove the proposed scheme is secure against particular attacks. The assumptions are as follows: an adversary can eavesdrop any messages transmitted between $T$, $G$, and $R$ and he/she can obtain a legitimate thing normally registered for $R$.

4.1. Anonymity. Our scheme provides anonymity against an adversary who acquires a valid thing.

4.1.1. Anonymity in Join Phase. An adversary can eavesdrop $T$’s shadow identity $SID_T$ and the timestamp $t_T$ from the message $m_1$ in the join phase. If the adversary obtains a valid thing $T'$ registered for $R$, he/she can have a knowledge of the secret information $[ID_T, VID_T, x_T, h(x_T), ID_\|]$ stored in $T'$. Thus, the adversary can get $T$’s virtual identity $VID_T$ by computing $VID_T = SID_T \oplus h(h(x_T) \parallel t_T)$ using $SID_T$ and $t_T$ from $m_1$ and $h(x_T)$ in $T'$. To acquire $T$’s real identity $ID_T$ from $VID_T = h(ID_T \parallel x_T \parallel t_T)$, the adversary guesses a candidate identity $ID_T'$ firstly, and then he/she tries to check the equivalence between $ID_T'$ and $ID_T$. Since an identity is short and has a certain format, it is guessable. However, the adversary cannot check whether $VID_T = h(ID_T' \parallel x_T \parallel t_T)$ is satisfied or not without knowing $T$’s private key $x_T$ and the timestamp $t_T$. Therefore, it is impossible for the adversary to check whether the real identity is equal to the candidate identity or not.

4.1.2. Anonymity in Communication Phase. In the communication phase, an adversary can get $T_1$’s shadow identity $SID_{T_1}$ and its timestamp $t_{T_1}$ from the message $m_7$ or $m_8$ and $h(x_T)$ from a legitimate thing $T'$. By using them, the adversary can gain $T_1$’s virtual identity $VID_{T_1} = SID_{T_1} \oplus h(h(x_1) \parallel t_{T_1})$ as he/she does in the join phase. Then the adversary tries to confirm whether $VID_{T_1}' = h(ID_{T_1}' \parallel x_{T_1} \parallel t_{T_1})$ is satisfied or not by guessing a candidate identity $ID_{T_1}'$. Unfortunately, the adversary cannot check the equivalence between the real identity and the candidate identity without knowing $x_{T_1}$ and $t_{T_1}$. For that reason, the adversary gets no information related to $T_1$’s identity.

4.2. Untraceability. The proposed scheme provides untraceability for a thing. Because all transmitting messages always include the unique shadow and virtual identities in the join and the communication phases, an adversary cannot know the fact that all messages are originated from the same thing.

4.2.1. Untraceability in Join Phase. If a common secret key shared between $T$ and $G$ is renewed periodically or $T$ tries to join many different $Gs$, an adversary can collect all shadow identities from $T$’s join request messages in every session by eavesdropping. Besides, the adversary who has a valid thing can obtain $T$’s virtual identity $VID_T$ by computing $VID_T = SID_T \oplus h(h(x_T) \parallel t_T)$ because he/she can know $SID_T$, $h(x_T)$, and $t_T$. Meanwhile, if the shadow identity or the virtual identity always has a same value in every session, the adversary can recognize that one thing sends the join request message repeatedly while he/she cannot know that thing’s real identity. In other words, the adversary can trace the fact that a certain thing requests to join continuously. However, the whole shadow identities are all different, since they include a unique timestamp. And the registration server always issues the new virtual identity to renew the old one in every session. Therefore, the join request message sent by the thing includes the different shadow identity all the time.
It means the proposed scheme guarantees untraceability for the thing.

4.2.2. Untraceability in Communication Phase. The shadow identity in the communication phase also includes a unique timestamp. And everything always computes the new shadow identity and sends it to the gateway when they request to communicate. The gateway stores this new shadow identity to use it in the next session only if it checks that the old shadow identity is valid. Thus, shadow identities in every session have a uniqueness. Due to this, an adversary cannot trace the action of the thing in the communication phase.

4.3. Replay Attacks. Although an adversary resends past transmitting messages after collecting them in a certain session, participants involved in the join and the communication phases can detect a retransmission and stop to communicate with each other.

4.3.1. Replay Attacks in Join Phase. In the join phase, an adversary can gather the messages \( m_1, m_2, m_3, \) and \( m_4 \) between \( T, G, \) and \( R \). When the adversary resends the old message \( m_1 = \{ \text{JoinReq}, \text{SID}_T, \text{ID}_G, t_T \} \) to \( G \), \( G \) checks a time interval between a current time and \( t_T \). And it rejects the join request if the time interval exceeds a predefined value. In addition, \( R \) can also recognize the old message \( m_2 = \{ \text{AuthReq}, \text{SID}_T, \text{ID}_G, H_t, t_T, t_G \} \) sent by the adversary by checking the time period between a current time and \( t_G \). In this case, \( R \) absolutely breaks a connection.

4.3.2. Replay Attacks in Communication Phase. An adversary who collects the messages \( m_2, m_5, m_7, \) and \( m_8 \) in the communication phase sends \( m_2 \) to \( T_2 \) and \( m_8 \) to \( G \). After receiving the message, \( T_2 \) and \( G \) can know whether the message is resent or not by checking a timestamp and a current time. As a result, resending old messages is always detectable.

4.4. Forgery Attacks. In our scheme, an adversary cannot forge messages to perform the valid join and communication phases.

4.4.1. Forgery Attacks in Join Phase. When the message \( m'_1 = \{ \text{JoinReq}, \text{SID}'_T, \text{ID}_G, t_T \} \) forged from \( m_1 \) by an adversary is sent to \( G \) in the join phase, \( G \) makes \( m'_2 = \{ \text{AuthReq}, \text{SID}'_T, \text{ID}_G, H'_t, t_T, t_G \} \) and sends it to \( R \). Then \( R \) computes \( \text{VID}'_2 = \text{SID}'_T \oplus h(h(x_R)) \parallel t_T \) using \( h(x_R) \) that is the hashing value of its private key \( x_R \) and tries to search \( T \)'s secret information using \( \text{VID}'_2 \) as a keyword. Since there is no matched virtual identity in \( R \)'s database, \( R \) stops to communicate with \( G \) after recognizing that \( m'_2 \) is originated from the forged message.

4.4.2. Forgery Attacks in Communication Phase. An adversary sends the message \( m'_5 = \{ \text{CommReq}, \text{SID}'_T, \text{ID}_G, V'_6, H'_6, t'_T \} \) to \( T_2 \) after forging \( T_1 \)'s message \( m_5 \) in the communication phase. Upon receiving \( m'_5 \), \( T_2 \) sends \( m'_6 = \{ \text{AuthReq}, \text{SID}'_T, \text{SID}_T, V'_6, V_7, H'_6, t_1, t_2, t_G \} \) to \( G \). Then \( G \) can easily know that no shadow identity equals to \( \text{SID}'_1 \) in its database. Finally, \( G \) rejects \( T_2 \)'s request.

4.5. Impersonation Attacks. The proposed scheme is secure against attacks that an adversary who obtains a shadow identity of the thing impersonates that thing.

4.5.1. Impersonation Attacks in Join Phase. An adversary who tries to impersonate the valid thing \( T \) easily obtains \( T \)'s shadow identity \( \text{SID}_T \) from the message \( m_1 \) in the join phase. The adversary makes the message \( m'_1 = \{ \text{JoinReq}, \text{SID}_T, \text{ID}_G, t'_T \} \) using the obtained \( \text{SID}_T \) and a timestamp \( t'_T \) generated by him/her in a new session. And he/she sends \( m'_1 \) to \( G \). After receiving \( m'_1 \), \( G \) decides \( m'_1 \) is a valid request message because \( t'_T \) and \( H'_1 \) are verified successfully. And then \( G \) sends the message \( m'_2 = \{ \text{AuthReq}, \text{SID}_T, \text{ID}_G, H'_T, t'_T, t_G \} \) to \( R \). Checking \( t'_G, \text{ID}_G, \) and \( H'_G \) by \( R \) is also successful. \( R \) computes \( \text{VID}'_2 = \text{SID}_T \oplus h(h(x_R)) \parallel t_T \). There is no equivalent virtual identity to \( \text{VID}'_2 \) in its database. Surely, \( R \) recognizes it is an abnormal case and rejects \( G \)'s request.

4.5.2. Impersonation Attacks in Communication Phase. An adversary gets \( \text{SID}_T \) from \( m_2 \) in the communication phase. To impersonate \( T_1 \), the adversary generates a new timestamp \( t'_1 \) makes the message \( m'_5 = \{ \text{CommReq}, \text{SID}_T, \text{ID}_G, V'_6, H'_6, t'_T \} \), and sends \( m'_5 \) to \( V'_6 \) in a new session. Here, the adversary cannot compute the valid \( V'_6 \) and \( H'_6 \) because he/she does not know \( T_1 \)'s common secret key \( K_{GT} \). After receiving \( m'_5 \), \( V'_6 \) makes the message \( m'_6 = \{ \text{AuthReq}, \text{SID}_T, \text{SID}_T, V'_6, H'_6, H'_T, t'_T, t'_T \} \) and sends it to \( G \). \( G \) verifies that all \( t'_T, \text{SID}_T, \) \( \text{SID}_T, \) and \( H'_T \) are valid but it detects that \( H'_T = (h(K_{GT}) \parallel \text{SID}_T \parallel V'_6 \parallel t'_T) \) and the received \( H'_T \) are not equal. As a result, \( G \) recognizes \( T_1 \) is an invalid thing and refuses \( T_2 \)'s request.

4.6. Known Session Key Attacks. Disclosure of a common secret key in the join phase or a session key in the communication phase does not affect revelations about the thing's identity, other common secret keys, and other session keys.

4.6.1. Known Session Key Attacks in Join Phase. A common secret key shared between \( T \) and \( G \) is computed using a timestamp in the join phase. Because a timestamp always has a different value, every common secret key is unique. Although an adversary knows \( K_{GT}(h(h(x_R)) \parallel \text{ID}_G \parallel \text{ID}_G \parallel t_G) \), he/she cannot obtain the thing's identity and the common secret key in other sessions. The reasons are as follows. Firstly, the adversary still cannot get the thing's identity \( \text{ID}_{RG} \) from \( K_{GT} \) without knowing \( T \)'s secret information \( h(x_R) \). Moreover, let us suppose the adversary tries to get other common secret keys \( K'_{GT} \) in the next session after computing \( h(K_{RG}) \parallel t_G \) using \( V_4 \) and \( K_{GT} \). In that next session, the adversary eavesdrops \( V_4 = (h(K'_{GT}) \oplus h(K_{RG}) \parallel t_G) \) and computes \( K'_{GT} \parallel h(K_{RG}) \parallel t_G \parallel t_G = \).
Table 3: Performance comparisons.

| Scheme | Ours | Chang et al.'s | Lee's |
|--------|------|----------------|-------|
| Join phase |
| Hash | 7 | 7 | 7 |
| XOR | 2 | 5 | 2 |
| Rounds | 4 | 8 | 7 |
| Communication phase |
| Hash | 5 | N/A | N/A |
| XOR | 3 | N/A | N/A |
| Rounds | 3 | N/A | N/A |

$V'_i \oplus h(K_{RG} \parallel t_G)$ using $V'_4$ and $h(K_{RG} \parallel t_G)$. Due to the difference of $t_G$ and $t'_G$, $h(K_{RG} \parallel t'_G) \oplus h(K_{RG} \parallel t_G)$ is not eliminated in the above equation and the adversary cannot get $K_{GT}$. As a result, it is impossible to calculate the new session key even though the former is revealed.

4.6.2. Known Session Key Attacks in Communication Phase. All session keys in the communication phase are independent since all of them include their own unique timestamps. If the session key SK shared between $T_1$ and $T_2$ is revealed to an adversary in a certain session, he/she can compute $h(K_{GT_{12}} \parallel SID_{T_1} \parallel t_{T_1}) = K_{12} \oplus SK$ using $K_{12}$ and the revealed SK. In the next session, the adversary eavesdrops $K'_1 = (SK' \oplus h(K_{GT} \parallel SID_{T_1} \parallel t_{T_1}))$ and tries to compute $K'_{T_1} \oplus h(K_{GT} \parallel SID_{T_1} \parallel t_{T_1}) = K_1' \oplus h(K_{GT_{12}} \parallel SID_{T_1} \parallel t_{T_1})$ using $K_1'$ and $(K_{GT_{12}} \parallel SID_{T_1} \parallel t_{T_1})$. Since $t_{T_1}$ is not equal to $t'_{T_1}$, $h(K_{GT_{12}} \parallel SID_{T_1} \parallel t'_{T_1})$ and $h(K_{GT_{12}} \parallel SID_{T_1} \parallel t_{T_1})$ are not erased. Therefore, there is no way to obtain the new session key $SK'$ from the old session key SK.

5. Performance Analysis

This section partially compares the performance of our scheme with those of Chang et al.'s and Lee's [22] schemes. Since the environment, the component, and the composition of ours and others' are all different, we firstly select similar factors in all schemes and evaluate them. Table 3 shows the performance comparison results. In Table 3, Hash denotes a hash operation, XOR denotes an exclusive-OR operation, and rounds denote communication rounds. We evaluate the number of hash and exclusive-OR operations. We compare the join phase of our scheme and the authentication and session key establishment phases of other schemes, because each phase has similar purpose and procedure. In addition, we only focus on the operations on the part of the limited computing power devices, such as a thing and a mobile user. Since the performance of the whole scheme depends on the capability of those devices, we do not compare the whole operations.

As Table 3 shows, our scheme requires less or the same exclusive-OR operations in the join phase. Namely, the computation complexity of ours is a little better than others'. Our communication rounds in the join phase are reduced by half in comparison with Chang et al.'s and less than that in Lee's. Considering the wireless environments, reducing the communication cost is meaningful. Meanwhile, there is no corresponding phases of Chang et al.'s and Lee's schemes to the communication phase of our scheme. So we only show the number of operations and communication rounds of our scheme in Table 3. Definitely, hash operations and communication rounds in the communication phase are less than those in the join phase. The thing accomplishes the communication phase more often than the join phase, so the less number of operations and communication rounds in the communication phase is remarkable. As a result, our scheme provides advantages of the communication complexity as well as the computation complexity.

6. Conclusions

In this paper, we propose a novel anonymous authentication scheme that ensures the things in IoT environments can communicate with one another safely. The proposed scheme uses the virtual identity of the thing to make it be anonymized and authenticated at the same time. To provide untraceability, we keep the uniqueness of all shadow and virtual identities. Namely, duplicated shadow and virtual identities are never used in every session through the renewal process. Even if the valid virtual identity issued by the registration server is revealed, an adversary cannot obtain the real identity from that virtual identity. It is impossible for an adversary to perform replay, forgery, and impersonation attacks. Moreover, he/she cannot know the valid common secret key shared between the registration server, the gateway, and the thing, even though a specific common secret key is exposed. The valid session key shared between two things is also safe when a particular session key is disclosed to an adversary. We only use low cost functions, such as hash functions and exclusive-OR operations in consideration of fast computation and low energy consumption of the thing. In conclusion, our scheme is suitable for IoT environments as a secure and efficient authentication mechanism.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion (IITP) Grant funded by the Korea government (MSIP) (no. R0126-15-III, The Development of Risk-based Authentication-Access Control Platform and Compliance Technique for Cloud Security).

References

[1] Z.-K. Zhang, M. C. Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, and S. Shieh, “IoT security: ongoing challenges and research opportunities,” in Proceedings of the 7th IEEE International Conference on Service-Oriented Computing and Applications (SOCA ’14), pp. 230–234, IEEE, Matsue, Japan, November 2014.
[2] J. Liu, Y. Xiao, and C. P. Chen, “Authentication and access control in the Internet of things,” in Proceedings of the 32nd IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW ’12), pp. 588–592, IEEE, Macau, China, June 2012.

[3] A. Alcaide, E. Palomar, J. Montero-Castillo, and A. Ribagorda, “Anonymous authentication for privacy-preserving IoT target-driven applications,” Computers & Security, vol. 37, pp. 111–123, 2013.

[4] B. Ndibanje, H. J. Lee, and S. G. Lee, “Security analysis and improvements of authentication and access control in the Internet of things,” Sensors, vol. 14, no. 8, pp. 14786–14805, 2014.

[5] N. Ye, Y. Zhu, R.-C. Wang, R. Malekian, and Q.-M. Lin, “An efficient authentication and access control scheme for perception layer of internet of things,” Applied Mathematics and Information Sciences, vol. 8, no. 4, pp. 1617–1624, 2014.

[6] X.-J. Lin, L. Sun, and H. Qu, “Insecurity of an anonymous authentication for privacy-preserving IoT target-driven applications,” Computers & Security, vol. 48, pp. 142–149, 2015.

[7] 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.

[8] C. H. Lin and C. Y. Lee, “Cryptanalysis of a new authentication scheme with anonymity for wireless environments,” in Proceedings of the 2nd International Conference on Advances in Mobile Multimedia, pp. 399–402, 2004.

[9] 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.

[10] 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.

[11] J.-S. Lee, J. H. Chang, and D. H. Lee, “Security flaw of authentication scheme with anonymity for wireless communications,” IEEE Communications Letters, vol. 13, no. 5, pp. 292–293, 2009.

[12] C.-C. Chang, C.-Y. Lee, and W.-B. Lee, “Cryptanalysis and improvement of a secure authentication scheme with anonymity for wireless communications,” in Proceedings of the 5th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP ’09), pp. 902–904, IEEE, Kyoto, Japan, September 2009.

[13] J. Xu and D. Feng, “Security flaws in authentication protocols with anonymity for wireless environments,” ETRI Journal, vol. 31, no. 4, pp. 460–462, 2009.

[14] W. Jeon, J. Kim, J. Nam, Y. Lee, and D. Won, “An enhanced secure authentication scheme with anonymity for wireless environments,” IEICE Transactions on Communications, vol. 95, no. 7, pp. 2505–2508, 2012.

[15] 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.

[16] 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.

[17] T. Zhou and J. Xu, “Provable secure authentication protocol with anonymity for roaming service in global mobility networks,” Computer Networks, vol. 55, no. 1, pp. 205–213, 2011.

[18] C. Chen, D. He, S. Chan, J. Bu, Y. Gao, and R. Fan, “Lightweight and provably secure user authentication with anonymity for the global mobility network,” International Journal of Communication Systems, vol. 24, no. 3, pp. 347–362, 2011.

[19] 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. 1-2, pp. 214–222, 2012.

[20] L. Kun, X. Anna, H. Fei, and D. H. Lee, “Anonymous authentication with unlinkability for wireless environments,” IEICE Electronics Express, vol. 8, no. 8, pp. 536–541, 2011.

[21] J.-L. Tsai, N.-W. Lo, and T.-C. Wu, “Secure anonymous authentication protocol with unlinkability for mobile wireless environment,” in Proceedings of the International Conference on Anti-Counterfeiting, Security and Identification (ASID ’12), pp. 1–5, IEEE, Taipei, Taiwan, August 2012.

[22] T.-F. Lee, “User authentication scheme with anonymity, unlinkability and untrackability for global mobility networks,” Security and Communication Networks, vol. 6, no. 11, pp. 1404–1413, 2013.
