Wireless Netw (2013) 19:1423–1440
DOI 10.1007/s11276-013-0542-8

A wide-adapted bantam protocol for roaming across wireless areas

Jung-San Lee · Wan-Ting Tseng

Published online: 10 February 2013

Abstract To guarantee stable convenience and feasibility of the widely applied wireless network, the handover technique is playing an important role nowadays to meet the epidemic mobile device usage. As is known, the wireless network authentication protocol can decide security for whole operating scheme and help protect rights of legal users, while there are still some flaws that prevent entire architecture from working favorably. Considering such flaws including computation overload as well as the key management burden, we propose a brand new authentication protocol to ensure secure message transmission and reduce computation overload. The proposed mechanism can be used both in the intra-domain and inter-domain, catering to the aim of decreasing management burden for the key distribution center and expediting the validation process efficiently. Furthermore, we provide the formal security analysis of BAN logic to demonstrate the applicability of the protocol.

Keywords Handover · Mutual authentication · EAP-based protocol · BAN logic

1 Introduction

With the wide and convenient access of wireless networks, the requirement for ubiquitous Internet service is increasing day by day. At the same time, it also concerns that whether the security of wireless networks can be guaranteed efficiently. IEEE 802.11i [10] defines a new security model for 802.11 a/b/g networks, and since 2001, IEEE 802.1X [9] has been implemented for enhancing user authentication process and managing key distribution. 802.1X is a port-based network access control mechanism that is able to provide an extensible authentication protocol (EAP) [2, 5]. EAP can offer a framework for addressing some concerns, within which protocol there exist a number of methods applied for authentication. As is known, RFC 4017 [17] describes the requirements for EAP methods in secure wireless LANs. Some essential properties, which have to be considered seriously to establish a secure wireless networks, are listed as follows [19–27].

1. Mutual authentication support:
   The EAP method needs to provide an authentication mode that allows the communicating entity to confirm each other mutually. Only the one-way verification methods are far from satisfying this requirement; the integrated framework has to employ general mutual authentication architecture to support this requirement.
2. Synchronization of state:
   The EAP ought to provide a synchronization mechanism. This contains cryptographic keys and data encryption methods, which are referred to when messages are exchanged between entities.
3. Protection against malicious attacks:
   (1) *Resistance to the man-in-the-middle attack* takes advantage of the trust between the two sides, thus the attacker can not eavesdrop on what the two sides exchange to gain secrets.
   (2) *Resistance to impersonation attack* provides a mechanism for working against the impersonation cheating. In authentication process, this mechanism must be able to guarantee that the

J.-S. Lee (✉) · W.-T. Tseng
Department of Information Engineering and Computer Science,
Feng Chia University, Taichung 40724, Taiwan,
Republic of China
e-mail: mankindlee@gmail.com

communication is unable to be camouflaged by malicious attackers.

(3) *Session independence* makes sure that every session stays independent, thus pledging no future or prior session can be compromised from the certain one.

4. Protected cipher suite negotiation:
The EAP method negotiation shared key is applied in encrypting the communication data, while the EAP method itself has to ensure the security of this negotiation shared key instead of the direct data for authentication.

In order to achieve the security requirements mentioned above, the EAP methods are well defined to establish critical essentials for wireless networks. The definition of EAP in 802.1X does not use special technique for authentication; it only specifies certain frameworks to achieve authentication procedures. In these EAP methods, there is a set of methods based on the asymmetric cryptosystem and the certificate for the authentication process, and there is also a set of methods according to the symmetric cryptosystem for the verification process.

Concerning about the various methods, EAP-TLS makes use of the asymmetric cryptosystem for authentication [1, 18]. Although this method can ensure the message robustness, it requires heavy computation for achieving the attempt. During the whole verification procedure, the system must verify the identity of each entity at the first authentication process, including user, server and the key distribution centre. Unfortunately, this method suffers from the impersonate attack, in which an attacker can forge a certificate and steal the secret key. Hence, EAP-TTLS [7] is proposed in 2008, which extends the EAP-TLS to improve the security. EAP-TTLS inherits the property of EAP-TLS to protect the communication data, and additionally enhances the security for establishing a secure channel. However, EAP-TTLS increases largely the computation burden at the same time of constructing a secure channel which guarantees its strong security. In order to decrease computation burden, EAP-LEAP [13] is proposed, which is based on the symmetric-key authentication protocol. EAP-LEAP achieves the session independence referring to the one-time key used by each session. Hence, even if a single session is compromised, the future or prior sessions shall never be revealed, thus enhancing the robustness of the whole mechanism. This method also provides a protection mechanism based on password authentication protocol that can easily resist the denial of service attacks. Based on the password authentication protocol, EAP-LEAP can certainly decrease the computation cost, while it still faces the challenge of the dictionary attack, seeing to the fact that it is not via a secure channel [6, 11, 12].

In RFC 3748 [2] and RFC 5247 [3], EAP-MD5 mechanism based on the symmetric cryptosystem is described, which uses the username and password to achieve authentication process and encrypts message via MD5 hashing algorithm. Simple as this method is, there exist significant weaknesses within it. As is known, EAP-MD5 can not offer exchanged key after each session; it can not achieve session independence either. In addition, this method is easy to suffer the man-in-the-middle attack considering its oversimplified procedure.

From the developing history of the standard mentioned above, it is obvious that the computation and security has been the most important consideration in designing an authentication mechanism. In 2009, Huang et al. [8] presented the One-time key Secure Network Protocol (OSNP) scheme, which is regarded as an efficient authentication mechanism for mobile devices and could be integrated with the 802.1X. Specifically, it can not only resist the man-in-the-middle attack but also reduce the computation load. Moreover, the usage of one-time key can greatly mitigate the risk of guessing attacks. Nevertheless, we have found that there still exist several defects in this method. For this reason, we aim to provide a new handover mechanism for the wireless networks, on the purpose of enhancing the secure property as well as reducing the computation cost [14].

The rest of the paper is organized as follows. In Sect. 2, we review Huang et al.'s scheme in detail. The proposed scheme is presented in Sect. 3. Security analysis and discussions are described in Sect. 4, followed with the achievement of mutual authentication proved by BAN logic in Sect. 5 [4, 16]. Finally, the conclusions are given in Sect. 6.

2 Review of Huang et al.'s scheme

In this section, we briefly review the One-time key Secure Network Protocol (OSNP) scheme and point out the weaknesses [8]. OSNP is mainly concerned with two domains: the intra-domain and inter-domain. The intra-domain authentication consists of three phases: initial, subsequent and handover phases. Notations used in this method are listed in Table 1.

2.1 Review of intra-domain authentication

2.1.1 The initial authentication phase

Before accessing to the server S, the user U has to be authorized in this phase first.

(1) U uses its identity, random nonce, and password to compute the user's one-time key $OTK_U = Hash(U, N_U, PW_U)$. Then U sends the authentication request $authRQ_U = U, N_U, \{U, N_U\}_{OTK_U}$ to the server S.

Table 1 Notations used in Huang et al.'s scheme

TKT_X	Ticket issued by X
CH_X	Challenge issued by X
$RESP_X$	Response to CH_X
A_X	Authenticator issued by X
U_a	User principle in domain a
S_a	Server principle in domain a
KDC_a	Key distribution center in domain a
PW_X	Password of X shared with corresponding KDC
N_X	Nonce generated by X
K_{SS}	Session key used during communication
K_g	Group key for all servers under the same KDC
K_{TU}	Temporal user key for subsequent authentication
OTK_X	One-time key of X
SID	Session identification
VT_X	Validation time (Expiration) of X
TU_X	Temporary user identity of X
rt	The remaining time of the validated ticket
ct	The current time of the local host
$\{\cdot\}_x$	The symmetric en/decryption with secret key K
$Hash$ ()	Secure one-way hash function

(2) After S receives $authRQ_U$, the server generates a corresponding authentication token as $authRQ_S = S, N_S, \{S, N_S\}_{OTK_S}$. Then S sends $authRQ_U$ and $authRQ_S$ to the key distribution center KDC.

(3) KDC generates a unique identity $SID = (U, S, N_U)$ and checks the one-time key OTK_U and OTK_S to verify the requests. After successfully authenticating these keys, KDC randomly constructs a session key K_{SS} and a temporary user key K_{TU}. Then KDC sends SID, $authAK_S = \{N_S, U, K_{SS}\}_{OTK_S}$, and $authAK_U = \{N_U, S, K_{SS}, K_{TU}\}_{OTK_U}$ back to the server S.

(4) Once getting the message, S decrypts $authAK_S$ to get the session key K_{SS} and computes $CH_S = \{S, N'_S\}_{K_{SS}}$ and $TKT_S = SID, \{U, VT_S, K_{SS}\}_{OTK_S}$. Subsequently S sends $authAK_U$, CH_S, and TKT_S to U.

(5) After U decrypts $authAK_U$, U can further generate a response message $RESP_S = \{U, N'_S\}_{K_{SS}}$ and verify the message CH_S. Next U sends $RESP_S$ and $A_U = \{S, VT_U, K_{SS}\}_{K_{TU}}$ to the server for further authentication.

2.1.2 *The subsequent authentication phase*

While the user tries to access to the same server within the specific time, U has to be re-authenticated by S with the authentication ticket, which is generated from the previous session.

(1) U sends $sauthRQ_U = (N_U, TKT_S)$ to the server S, where TKT_S is a ticket as certificate.

(2) S decrypts $sauthRQ_U$ to get TKT_S, and then makes sure that the ticket is expired or not. If it is valid, S generates a new random nonce and a session key to compute the response message $sauthAK_U = \{N_U, K'_{SS}\}_{K_{SS}}$, as well as calculates a new challenge $CH_S = \{S, N_S\}_{K'_{SS}}$ for mutual authentication with U. Finally, S concatenates $sauthAK_U$, CH_S, and A_U, and then sends it back to user U.

(3) After receiving the message, U decrypts it to check the validity. If this message is validated, U computes the response message $RESP_S = \{U, N_S\}_{K'_{SS}}$.

2.1.3 *The handover authentication phase*

When a user wants to request the service from a new server S belonging to the same KDC, the user must be re-authenticated from the previous server S_{old}.

(1) U sends the request message $hauthRQ_U = (U, N_U, TKT_{S_{old}})$ to the new server S for re-authentication.

(2) The new server S generates $CH_S = \{S, N_S\}_{K_g}$ and forwards CH_S with $hauthRQ_U$ to the previous server S_{old}.

(3) S_{old} computes remaining validation time for the ticket $rt = (VT_{S_{old}} - ct)$. Then S_{old} sends SID and $hauthVF_{S_{old}} = \{U, K_{SS_{old}}, rt, A_{U_{old}}\}_{K_g}$ to the new server S.

(4) Upon receiving the message, S decrypts the message to get rt and computes $VT_S = (ct + rt)$. After that, S generates $hauthAK_U = \{N_U, K_{SS}\}K_{SS_{old}}$, $CH_S = \{S, N'_S\}_{K_{SS}}$, and the new ticket TKT_S. Finally, S sends $hauthAK_U || CH_S || TKT_S || A_{U_{old}}$ back to user U.

(5) After getting the message, U checks the validity of message first. If this message is valid, U calculates the response message $RESP_S = \{U, N'_S\}_{K_{SS}}$ and a new temporary certificate $A_U = \{S, VT_U, K_{SS}\}_{K_{TU}}$. Subsequently, U sends $RESP_S$ and A_U to the new server S.

2.2 Review of the inter-domain authentication

When the user roams to another domain, it must be re-authenticated by the new KDC. Similar to the handover authentication phase in the intra-domain, it also requests the authentication information from the previous domain for the new access one. A hierarchical structure is used to represent the system as shown in Fig. 1. Steps for communicating are described in the below.

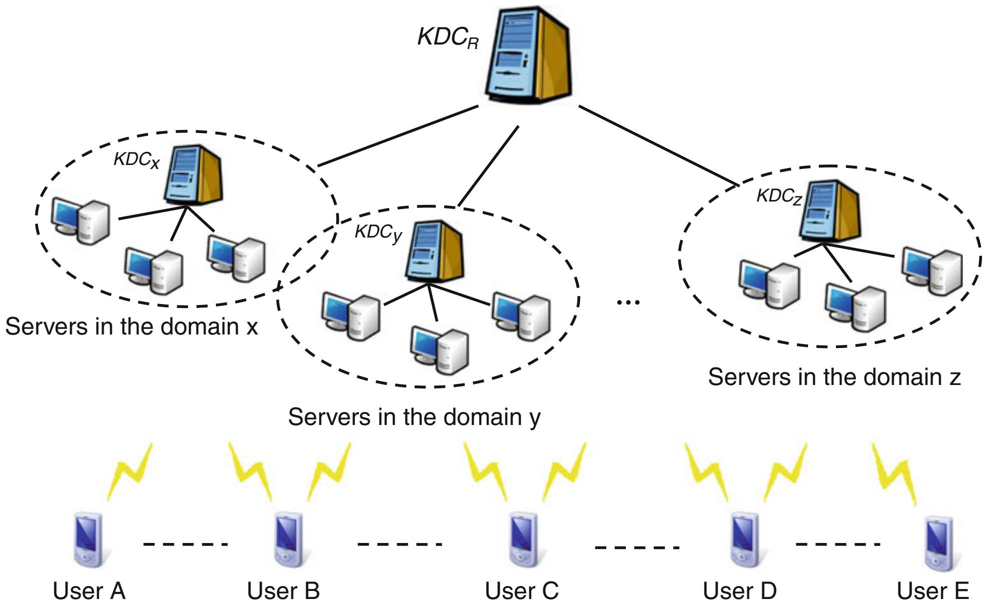

Fig. 1 System environment

(1) The roaming user TU_X sends the request message $authRQ_{TU_X} = (TU_X, N_{TU_X}, \{TU_X, N_{TU_X}\}_{K_{TU_X}})$ to the server S_Y of new access domain.

(2) When the server S_Y receives the request information, it immediately generates the same format request message $authRQ_{S_Y}$ and forwards it to KDC_Y along with $authRQ_{TU_X}$.

(3) After KDC_Y gets the request message, it sends $iauthRQ_{KDC_Y}$ combined with $authRQ_{S_Y}$, identity of KDC_Y, and $authRQ_{TU_X}$ to the root key distribution center KDC_R.

(4) KDC_R decrypts the received message to check whether the message is valid or not. Moreover, KDC_R generates a new temporary user key and a new session key. Subsequently, KDC_R delivers

$$iauthFW_{KDC_X} = (authRQ_{TU_X}, \{TU_Y, K_{TU_Y}, K_{SS}\}_{K_{KDC_X}})$$

and

$$iauthAK_{KDC_Y} = \{N_{KDC_Y}, TU_Y, K_{TU_Y}, K_{SS}\}_{K_{KDC_Y}} \text{ to } KDC_x.$$

(5) When KDC_X decrypts $iauthFW_{KDC_X}$ to get the essential information, KDC_X directly sends $iauthAK_{KDC_Y}$ and $iauthAK_{TU_X} = \{N_{TU_X}, TU_Y, K_{TU_Y}, K_{SS}\}_{K_{TU_X}}$ to KDC_Y.

(6) After KDC_Y receives the message, it checks the validity first. Then KDC_Y generates a new random nonce N_{S_Y} and the response message as

$$iauthAK_{S_Y} = \{N_{S_Y}, TU_Y, K_{SS}\}_{OTK_{S_Y}},$$

where $OTK_{S_Y} = Hash(S_Y, N_{S_Y}, PW_{S_Y})$. Finally, KDC_Y sends SID, $iauthAK_{S_Y}$, and $iauthAK_{TU_X}$ to S_Y.

(7) In the next, S_Y generates $CH_{S_Y} = \{S_Y, N'_{S_Y}\}_{K_{SS}}$ and a service ticket $TKT_{S_Y} = (SID, \{TU_Y, VT_{S_Y}, K_{SS}\}_{OTK_{S_Y}})$. Similar to the intra-domain, S_Y sends these messages back to the user TU_X.

(8) Finally, TU_X generates the certificate $A_{TU_Y} = \{S_Y, VT_{TU_Y}, K_{SS}\}_{K_{TU_Y}}$ for the next authentication phase. And TU_X sends the response message $RESP_{S_Y} = \{TU_Y, N'_{S_Y}\}_{K_{SS}}$ and A_{TU_Y} to S_Y.

2.3 Weakness of Huang et al.'s scheme

1. In the initial authentication phase of the intra-domain, we assume that an attacker is able to steal PW_U. Since OTK_U consists of U, N_U and PW_U, the OTK_U is easy to be reconstructed by the attacker. When KDC generates a temporary user key K_{TU} for the subsequent and the handover authentication phases, KDC uses OTK_U to encrypt K_{TU} and sends this encrypted message to the server. Thus K_{TU} can readily be gained by the attacker. As mentioned above, it is obvious that Huang et al.'s scheme can not confirm the session independence property.
2. In the inter-domain, this mechanism suffers the dictionary attack similar to that in the intra-domain. When the user roams to another domain, the user employs its own information in the initial phase to negotiate the new session key. As analyzed above, if an attacker can get the previous temporary key K_{TU}, this mechanism can not achieve the session independence property.

Table 2 Protocol notations

ID_{U_x}	User identity in domain "x"
$ID_{S_{x_i}}$	Server identity in domain "x" for the ith server, $i \in N$
ID_{K_x}	Key distribution center identity in domain "x"
K_{AB_x}	The secret key shared between "A" and "B" in domain "x"
$K_{K_{Rx}}$	The secret key shared between KDC_x and KDC_R
$K_{K_{Ry}}$	The secret key shared between KDC_y and KDC_R
K_G	Group key for all servers under KDC in the same domain
K_A	Temporal user key of "A" for subsequent authentication
N_{A_x}	The nonce generated by "A" in domain "x"
T_{A_x}	The timestamp generated by "A" in domain "x"
T_i	The timestamp generated by U for the ith authentication request, $i \in N$
$H(\cdot)$	Collision-free hash function

3 The proposed protocol

The system environment is the same as Fig. 1. Notations used in the proposed protocol are listed in Table 2.

There are three main participants in the wireless environment, the mobile user U, the server S, and the key distribution center KDC. Each network domain is formed as a hierarchical tree, and KDC_R is assigned as the root of all hierarchical trees. The handover process occurred in the same domain is referred to the intra-domain authentication, while that happened in different network areas is considered as the inter-domain authentication.

3.1 Intra-domain authentication

The intra-domain authentication process consists of three phases: initial, subsequent, and handover phases. In the initial authentication phase, user asks for mutual authentication to join the visiting server. As to the second phase, the user sends the ith authentication request to the same server for the ith communication service, where $i > 1$ and $i \in N$. Regarding to the handover authentication phase, it describes how to transfer the user's service from the current server to another in the same network domain. In the following, we are going to illustrate the details of these three phases in Sects. 3.1.1–3.1.3, respectively. The flowcharts of three phases are depicted in Figs. 2, 3, and 4.

3.1.1 Phase 1: initial authentication in the visiting server

Assume that the network is dominated by a key distribution center KDC, each user U shares a secret key K_{UK} with KDC, and each server S keeps a secret key K_{SK} shared with KDC.

Step 1: When a client U moves to a new access area and asks for new communication service, U has to generate an authentication request containing a timestamp T_1 and the user's identity, which are encrypted by the secret key shared between U and KDC. This request is sent to the server for verification.

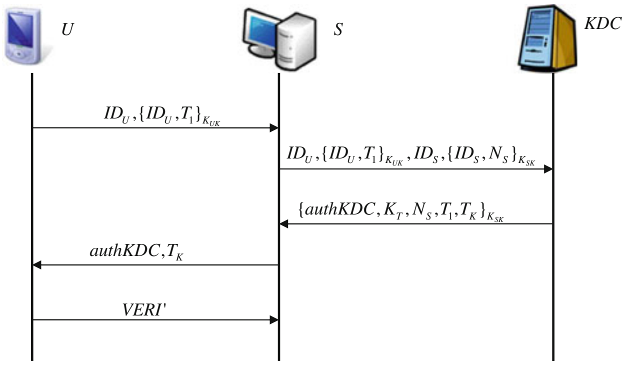
U
S
KDC
$ID_U, \{ID_U, T_1\}_{K_{UK}}$
$ID_U, \{ID_U, T_1\}_{K_{UK}}, ID_S, \{ID_S, N_S\}_{K_{SK}}$
$\{authKDC, K_T, N_S, T_1, T_K\}_{K_{SK}}$
$authKDC, T_K$
$VERI'$

Fig. 2 The flowchart of Phase 1

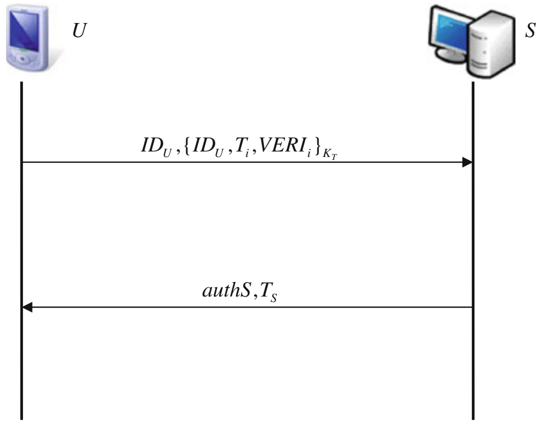

Fig. 3 The flowchart of Phase 2

Step 2: After the server S receives the request, S generates a nonce N_S and computes $\{ID_S, N_S\}_{K_{SK}}$. Afterwards the server sends $ID_U, \{ID_U, T_1\}_{K_{UK}}, ID_S, \{ID_S, N_S\}_{K_{SK}}$ to KDC.

Step 3: When KDC receives the information, it first employs the corresponding shared secret keys K_{UK} and K_{SK} to decrypt $\{ID_U, T_1\}_{K_{UK}}$ and $\{ID_S, N_S\}_{K_{SK}}$. Then it checks the nonce N_S to confirm whether the messages are legal or not. If one of them is not valid, the authentication process is terminated; otherwise, KDC generates a timestamp T_K to compute $authKDC = H(K_{UK}, T_K)$ and $K_T = H(K_{UK}, T_1, T_K)$. Then KDC forwards $\{authKDC, K_T, N_S, T_1, T_K\}_{K_{SK}}$ to S.

Step 4: Once S obtains the message, it uses K_{SK} to decrypt $\{authKDC, K_T, N_S, T_1, T_K\}_{K_{SK}}$. S further computes $VERI = H(K_T, T_1)$ and keeps this in the database with T_1 for future authentication. Moreover, S transmits $authKDC$ and T_K to U.

Step 5: When U gets the information, U computes $authKDC' = H(K_{UK}, T_K)$, and compares it with the received one. If they are not the same, the process is terminated; otherwise, server is authenticated. U then

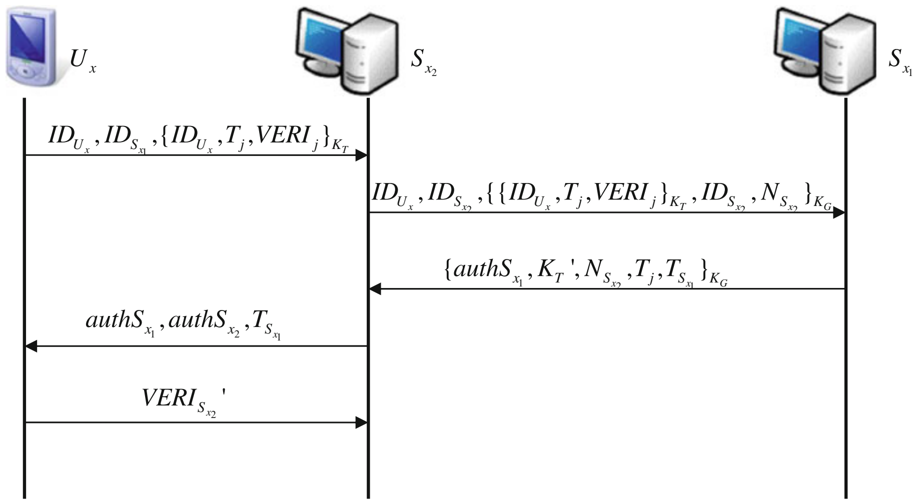
U_x
S_{x_2}
S_{x_1}
$ID_{U_x}, ID_{S_{x_1}}, \{ID_{U_x}, T_j, VERI_j\}_{K_T}$
$ID_{U_x}, ID_{S_{x_2}}, \{\{ID_{U_x}, T_j, VERI_j\}_{K_T}, ID_{S_{x_2}}, N_{S_{x_2}}\}_{K_G}$
$\{authS_{x_1}, K_T', N_{S_{x_2}}, T_j, T_{S_{x_1}}\}_{K_G}$
$authS_{x_1}, authS_{x_2}, T_{S_{x_1}}$
$VERI_{S_{x_2}}'$

Fig. 4 The flowchart of Phase 3

computes $K_T' = H(K_{UK}, T_1, T_K)$ and $VERI' = H(K_T', T_1)$. Finally, U sends $VERI'$ back to S.

Step 6: While receiving the information from U, S compares the received $VERI'$ with the one kept in the database. If $VERI' = VERI$, the mutual authentication between U and S is confirmed; otherwise, it fails.

3.1.2 Phase 2: subsequent authentication between the same server and user

Step 1: While U stays in the same service domain and requests for new communication session, U needs to generate a timestamp T_i and computes $VERI_i = H(T_{i-1}, T_i, K_T)$, where $T_{i\text{-}1}$ is the timestamp of previous session. Then U computes and sends $\{ID_U, T_i, VERI_i\}_{K_T}$ to server along with the identity.

Step 2: When S receives the request from U, S decrypts $\{ID_U, T_i, VERI_i\}_{K_T}$ and computes $VERI_i' = H(T_{i-1}, T_i, K_T)$, where T_{i-1} is the timestamp stored in the database for the previous authentication. Later on, S compares $VERI_i'$ with the decrypted one. If they are not the same, the process is terminated; otherwise, U is verified and the stored timestamp is updated to T_i. Next, S generates a timestamp T_S to compute a new session key $K_T' = H(K_T, T_i, T_S)$ and $authS = H(K_T', T_S)$ Afterwards, S transmits $authS$ and T_S to U.

Step 3: After receiving the message, U first computes $authS' = H(K_T', T_S)$ and compares $authS'$ with the received $authS$. If $authS' = authS$, the authentication is success; otherwise, the process is halted.

3.1.3 Phase 3: handover authentication between the original server and another server in the same domain

While U moves to the area dominated by another server S_{x_2}, the communication provider must be changed from S_{x_1} to S_{x_2}.

Step 1: Once the handover procedure occurs, the user U_x needs to generate a timestamp T_j and compute $VERI_j = H(T_{j-1}, T_j, K_T)$, where T_{j-1} is the timestamp for previous session. U_x then computes and sends $\{ID_{U_x}, T_j, VERI_j\}_{K_T}$ to the new server S_{x_2} along with the identity.

Step 2: Upon receiving the request, S_{x_2} generates a nonce $N_{S_{x_2}}$ and constructs $\{\{ID_{U_x}, T_j, VERI_j\}_{K_T}, ID_{S_{x_2}}, N_{S_{x_2}}\}_{K_G}$ using the group key of domain x. It then forwards the encrypted result to S_{x_1} along with the corresponding identities.

Step 3: When receiving the request from S_{x_2}, S_{x_1} decrypts $\{\{ID_{U_x}, T_j, VERI_j\}_{K_T}, ID_{S_{x_2}}, N_{S_{x_2}}\}_{K_G}$ and $\{ID_{U_x}, T_j, VERI_j\}_{K_T}$ first. It further computes and compares $VERI_j' = H(T_{j-1}, T_j, K_T)$ with the decrypted one. If they are not identical, the process is terminated; otherwise, S_{x_1} generates a timestamp $T_{S_{x_1}}$ to compute $K_T' = H(K_T, T_j, T_{S_{x_1}})$ and $authS_{x_1} = H(K_T, T_{S_{x_1}})$. After that, S_{x_1} computes and sends $\{anthS_{x_1}, K_T', N_{S_{x_2}}, T_j, T_{S_{x_1}}\}_{K_G}$ to S_{x_2}.

Step 4: After S_{x_2} gets the message, it first decrypts $\{anthS_{x_1}, K_T', N_{S_{x_2}}, T_j, T_{S_{x_1}}\}_{K_G}$. Next, it computes and stores $VERI_{S_{x_2}} = H(K_T', T_j)$ in the database along with T_j. Later on, S_{x_2} transmits $authS_{x_1}$, $authS_{x_2}$, and $T_{S_{x_1}}$ to U_x.

Step 5: When obtaining the message, U_x computes and compares $authS_{x_1}' = H(K_T, T_{S_{x_1}})$ with the received $authS_{x_1}$. If they are different, the process is halted; otherwise, U_x further uses $T_{S_{x_1}}$ to compute $K_T'' = H(K_T, T_j, T_{S_{x_1}})$ and $authS_{x_2}' = H(K_T'', T_{S_{x_1}})$. Subsequently, U_x checks whether $authS_{x_2}'$ is identical with the received $authS_{x_2}$. If it holds, U_x sends $VERI_{S_{x_2}}' = H(K_T'', T_j)$ to S_{x_2}; otherwise, the process is terminated.

Step 6: Once receiving the message from U_x, S_{x_2} compares the receiving message $VERI_{S_{x_2}}'$ with the one stored in the database. If $VERI_{S_{x_2}}' = VERI_{S_{x_2}}$, the handover procedure is confirmed; otherwise, the authentication is failure.

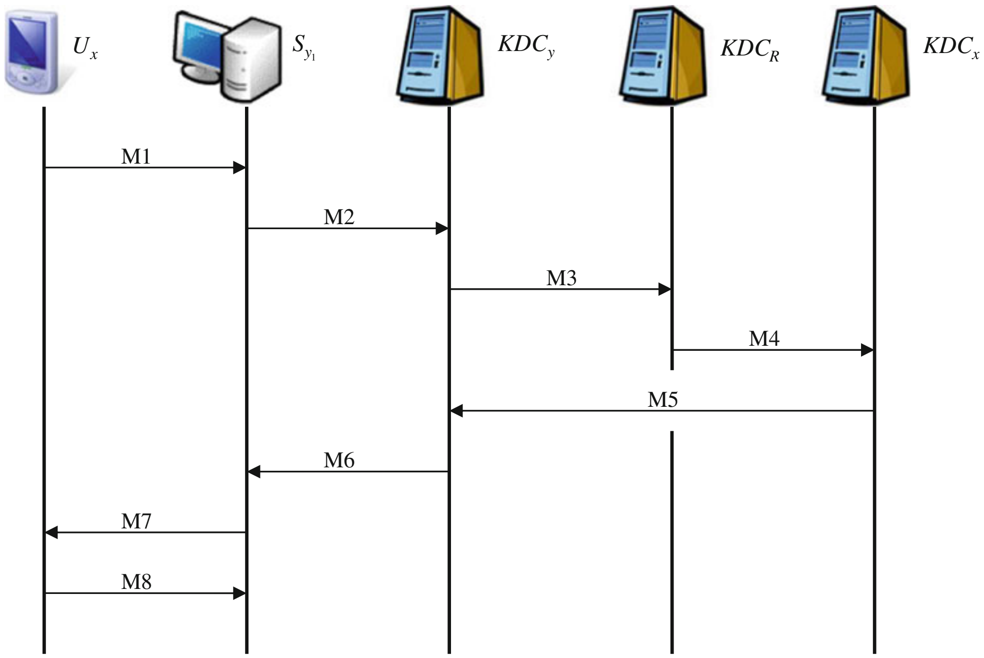

Fig. 5 The flowchart of inter-domain authentication

3.2 Inter-domain authentication

In this subsection, we describe the protocol of the inter-domain authentication. In the case of inter-domain authentication, we consider the scenario that the user U_x moves from the service area x dominated by KDC_x to the area y controlled by KDC_y. The flowchart is displayed in Fig. 5. Abbreviations listed in Fig. 5 are defined as follows.

M1: $U_x \rightarrow S_{y_1}$:
$ID_{U_x}, ID_{K_x}, \{ID_{U_x}, T_{U_x}\}_{K_{UK_x}}$

M2: $S_{y_1} \rightarrow KDC_y$:
$ID_{S_{y_1}}, \{ID_{U_x}, ID_{K_x}, \{ID_{U_x}, T_{U_x}\}_{K_{UK_x}}\}_{K_{SK_y}}$

M3: $KDC_y \rightarrow KDC_R$:
$ID_{S_{y_1}}, ID_{K_y}, \{ID_{U_x}, ID_{K_x}, \{ID_{U_x}, T_{U_x}\}_{K_{UK_x}}, N_y\}_{K_{K_{Ry}}}$

M4: $KDC_R \rightarrow KDC_x$:
$ID_{S_{y_1}}, ID_{K_y}, \{ID_{U_x}, ID_{K_x}, \{ID_{U_x}, T_{U_x}\}_{K_{UK_x}}, N_y\}_{K_{K_{Rx}}}$

M5: $KDC_x \rightarrow KDC_y$:
$\{authKDC_x, K_{T_x}, T_{U_x}, T_{K_x}\}_{N_y}$

M6: $KDC_y \rightarrow S_{y_1}$:
$\{authKDC_x, authKDC_y, K_{T_y}, T_{U_x}, T_{K_x}, T_{K_y}\}_{K_{SK_y}}$

M7: $S_{y_1} \rightarrow U_x$:
$authKDC_x, authKDC_y, T_{K_x}, T_{K_y}$

M8: $U_x \rightarrow S_{y_1}$: $VERI'_{S_{y_1}}$

Step 1: Once U_x roams to a new area and asks for communication service, U_x has to generate a timestamp T_{U_x} and compute $\{ID_{U_x}, T_{U_x}\}_{K_{UK_x}}$ first, where K_{UK_x} is the secret key shared between U_x and KDC_x. Subsequently, U_x sends the result to the foreign server S_{y_1} along with user's identity and KDC_x.

Step 2: When receiving the request, S_{y_1} employs the secret key shared with KDC_y to encrypt the received message and forwards to the result to KDC_y along with its identity.

Step 3: While KDC_y receives the message from S_{y_1}, KDC_y decrypts $\{ID_{U_x}, ID_{K_x}, \{ID_{U_x}, T_{U_x}\}_{K_{UK_x}}\}_{K_{SK_y}}$ and generates a nonce N_y. Then KDC_y encrypts ID_{U_x}, ID_{K_x}, $\{ID_{U_x}, T_{U_x}\}_{K_{UK_x}}$ and N_y using the secret key $K_{K_{Ry}}$ shared between KDC_y and KDC_R. Thereafter KDC_y sends the encrypted result to KDC_R along with the identities of S_{y_1} and itself.

Step 4: Upon obtaining the message from KDC_y, KDC_R decrypts $\{ID_{U_x}, ID_{K_x}, \{ID_{U_x}, T_{U_x}\}_{K_{UK_x}}, N_y\}_{K_{SK_y}}$ to get N_y first. KDC_R then encrypts ID_{U_x}, ID_{K_x}, $\{ID_{U_x}, T_{U_x}\}_{K_{UK_x}}$ and N_y using $K_{K_{Rx}}$ shared between KDC_x and KDC_R. Next, KDC_R forwards S_{y_1}, KDC_y, and the encryption message to the KDC_x.

Step 5: After receiving the message, KDC_x first decrypts $\{ID_{U_x}, ID_{K_x}, \{ID_{U_x}, T_{U_x}\}_{K_{UK_x}}, N_y\}_{K_{K_{Rx}}}$ and $\{ID_{U_x}, T_{U_x}\}_{K_{UK_x}}$ to get N_y and T_{U_x}. Then it generates a timestamp T_{K_x} to compute $authKDC_x = H(K_{UK_x}, T_{K_x})$ and $K_{T_x} = H(K_{UK_x}, T_{U_x}, T_{K_x})$. Afterwards, KDC_x applies N_y to encrypt $authKDC_x$, K_{T_x}, T_{U_x} and T_{K_x}. KDC_x subsequently transmits the encryption message to KDC_y.

Step 6: When KDC_y gets the message from KDC_x, it decrypts $\{authKDC_x, K_{T_x}, T_{U_x}, T_{K_x}\}_{N_y}$ and generates a timestamp T_{K_y} to compute $K_{T_y} = H(K_{T_x}, T_{U_x}, T_{K_y})$ and $authKDC_y = H(K_{T_x}, T_{K_y})$. KDC_y thus can employ the secret key K_{SK_y} to encrypt $authKDC_x$, $authKDC_y$, T_{U_x}, T_{K_x}, T_{K_y} and K_{T_y}. Then KDC_y sends the encrypted result to S_{y_1}.

Step 7: While S_{y_1} acquires the information, it decrypts $\{authKDC_x, authKDC_y, T_{U_x}, T_{K_x}, T_{K_y}, K_{T_y}\}_{K_{SK_y}}$ first. It then computes and stores $VERI_{S_{y_1}} = H(K_{T_y}, T_{U_x})$ in itself database along with T_{U_x}. Furthermore, S_{y_1} delivers $authKDC_x$, $authKDC_y$, T_{K_x}, and T_{K_y} to U_x.

Step 8: After getting the information, U_x computes $authKDC'_x = H(K_{UK_x}, T_{K_x})$ and compares it with the received $authKDC_x$. If they are not identical, the process is terminated; otherwise, U_x computes $K_{T_x} = H(K_{UK_x}, T_{U_x}, T_{K_x})$ and $authKDC'_y = H(K_{T_x}, T_{K_y})$. U_x then compares $authKDC'_y$ with the received one. If they are the same, U_x computes $K_{T_y} = H(K_{T_x}, T_{U_x}, T_{K_y})$ and $VERI'_{S_{y_1}} = H(K_{T_y}, T_{U_x})$; otherwise, this process is terminated. Finally, U_x sends $VERI'_{S_{y_1}}$ to S_{y_1}.

Step 9: While obtaining the message from U_x, S_{y_1} compares $VERI'_{S_{y_1}}$ with the one stored in itself database. If they are identical, the mutual authentication is confirmed; otherwise, the handover procedure is failure.

4 Security analysis and discussions

In this section, we discuss how the proposed protocol can resist diverse attacks. In addition, we compare other related works with ours in terms of computational costs and confirmed requirements.

4.1 Security analysis

The security of the new protocol is based on the symmetric cryptosystem and the one-way hash function [15].

1. Symmetric cryptosystem: Given a plaintext n, it is easy to encrypt n with a symmetric key K, $\{n\}_K$; however, it is computationally infeasible to extract n from $\{n\}_K$ without K.
2. Hash function: Given a value m, it is easy to compute $H(m)$, but it is computationally infeasible to achieve the following.
   - Given a value m', to find m such that $m' = H(m)$.
   - Given $H(m)$, to find m' such that $m' \neq m$ and $H(m') = H(m)$.

4.1.1 Analysis 1: resistance to malicious attacks

Proof Assume that there exists an intruder Eve being capable of eavesdropping the communications in the wireless network. Here, we discuss three cases in the initial authentication phase of the intra-domain to show that the new protocol can resist the impersonation attacks.

a. Eve camouflages *KDC* to monitor the session communications
b. Man-in-the-middle attack: Eve camouflages the server S to fool *KDC* and U.
c. Eve camouflages the mobile user U to cheat *KDC*.

In the first case, Eve has to obtain the current session K_T in advance. There are only two ways to achieve this. Eve can intercept the response message $\{authKDC, K_T, N_S, T_1, T_K\}_{K_{SK}}$ and try to retrieve K_T. It is clear that this must fail since it has contradicted the first assumption without the knowledge of K_{SK}. The second is to compromise the secret key via figuring out a K_E such that $K_E = K_T$. Under the assumption of one-way hash function, this is computationally infeasible without (T_1, T_K, K_{UK}). Hence, we can conclude that Eve can not succeed in monitoring the session communications.

In the second case, if Eve wants to impersonate the server S to fool *KDC*, she has to forward $(ID_U, \{ID_U, T_1\}_{K_{UK}}, ID_S, \{ID_S, N_S\}_{K_{SK}})$ to *KDC*. After checking the freshness of the random nonce N_S, *KDC* must detect this illegal attempt due to the fact that it is a replayed one. Furthermore, she may launch this attack by generating a valid random nonce N_E and embedding it into $\{ID_S, N_E\}_{K_{SK}}$. Then she can send $(ID_U, \{ID_U, T_1\}_{K_{UK}}, ID_S, \{ID_S, N_E\}_{K_{SK}})$ to *KDC*. Nevertheless, it is impossible for her to achieve this without the secret key K_{SK} according to the symmetric cryptosystem assumption. Thus, Eve can not pretend a valid server to cheat *KDC*. On the other hand, if Eve intends to impersonate S to fool U, she has to intercept *authKDC* and T_K in Step 4 of the initial authentication phase first. Once Eve forwards the intercepted messages to U, the user is able to detect that this is an invalid message via verifying the freshness of timestamp T_K. In case that she replaces T_K with a current timestamp T_E, U still can find that the message is an invalid one via computing and comparing $authKDC' = H(K_{UK}, T_E)$ with *authKDC*. Under the assumption of one-way hash function, it is computationally infeasible for her to construct a valid *authKDC'* without knowing the information of K_{UK}. Thus, the new protocol can resist this kid of malicious attempt.

Finally, in case that Eve wants to masquerade the mobile user U to cheat *KDC*, she has to generate and embed a current timestamp T_E into $\{ID_U, \{ID_U, T_E\}_{K_{UK}}\}$. This is due to that *KDC* can check the validity of this message by the decrypted timestamp. Without the shared key K_{UK}, Eve can not achieve this embedding according to the assumption of symmetric cryptosystem. Therefore, this protocol is confirmed to resist this cheating attack. □

4.1.2 Analysis 2: session independence

We focus on the case occurred in the intra domain to demonstrate how the new method can confirmed this requirement.

Proof After Step 4 of Sect. 3.1.1, S is able to obtain the session K_T by computing $\{\{authKDC, K_T, N_S, T_1, T_K\}_{K_{SK}}\}_{K_{SK}}$. On the other hand, U can generate the session key via computing $K'_T = H(K_{UK}, T_1, T_K)$ in Step 5. It is clear that there are two timestamps contributing to the session key. Due to the fact that the timestamp changes perpetually from this session to others and the sensitivity of one-way hash function, we know that the session keys belonging to different sessions must be different. That means the secret key employed to confirm each session communication is individual, thus implying the achievement of session independence. □

4.1.3 Analysis 3: 802.1X identity privacy

Proof Assume that an intruder Eve is able to intercept the messages during the network transmission. In the new protocol, Eve can intercept the identity of user in four cases.

Case 1: Block Step 1 of initial phase in the intra-domain

Once Eve gets the identity of U and tries to use it to login the network system for service access, she must fail. This is due to that, under the assumption of one-way hash function, she can not embed the identity and a valid timestamp into the request message of this step without the knowledge of K_{UK}.

Case 2: Block Step 1 of subsequent phase in the intra-domain

Even Eve can obtain the identity of U and use it to join the network system for service access, she must fail. This is due to the fact that it is computationally infeasible for her to compute K_T to make a valid request of this step. That is, she has to face the difficulty of compromising the one-way hash function, which is assumed to be impossible in our system.

Case 3: Block Steps 1 and 2 of handover phase in the intra-domain

The reason is the same as that of case 2.

Case 4: Block Step 1 of the intra-domain authentication

The reason is the same as that of case 1.

We then can conclude that the new protocol can comply with 802.1X to confirm the identity privacy. □

4.2 Discussions

In this subsection, we provide the comparisons between some related EAP based authentication schemes and our proposed scheme. We first describe how the new method is able to ensure the requirement listed in Table 3. *Mutual authentication* denotes that involved participants can confirm the identities of each other. The demonstration that our proposed scheme is able to confirm this essential under BAN logic model is given in the next section. As to the requirement of *Synchronization of state*, it claims that the new method shall provide the synchronization mechanism containing the functionality of data encryption. This has been achieved in the new method according to the fact that the symmetric secret keys used to encrypt data are constructed by corresponding timestamps. Considering the *Protected cipher suite negotiation*, a shared key has to be negotiated to encrypt the communications between involved participants. The hash value with the input of two timestamps and one previous secret key is constructed as the session key. In particular, one of the timestamp is kept secret all the time during the transmission. This results in that no one can forge a valid session key even the previous common session key is compromised somehow. Consequently, this property is ensured in the new method. Furthermore, the essentials of *Protection to the man-in-the-middle attack*, *Resistance impersonation attack*, *Session independence* and *802.1X Identity Privacy* have been illustrated in Sect. 4.1. According to the comparisons in Table 3, it is obvious that only the proposed scheme can guarantee all general requirements of the existing EAP based authentication protocols. Especially, the resistance to impersonation attack is just confirmed in the new method, which is considered as one of the most important security challenge.

In Table 4, we show the computation simulation of the symmetric key encryption (AES) and the hash function (SHA-1), which are the main components of new method. Simulators were executed in the VC6.0 language with Intel

Table 3 Requirements comparisons

Requirements	EAP-MD5	EAP-TLS	EAP-LEAP	EAP-OSNP	Proposed
Mutual authentication	No	Yes	No	Yes	Yes
Synchronization of state	Yes	Yes	Yes	Yes	Yes
Resistance to MITM attack	No	Yes	Yes	Yes	Yes
Resistance to impersonation attack	No	No	Yes	No	Yes
Session independence	No	No	Yes	No	Yes
Protected cipher suite negotiation	Yes	–	Yes	Yes	Yes
802.1X identity privacy	No	Yes	No	Yes	Yes

Table 4 A comparison of time taken

Operation	Time (S)
Symmetric key encryption	0.0087
One way hash function	0.0005

Table 5 Performance comparison: computation costs in intra-domain

Intra-domain phase	Operation	EAP-OSNP [8]				Proposed			
		KDC	S_{old}	*S*	*U*	*KDC*	S_{old}	*S*	*U*
Initial phase	Random number	2	–	2	1	1	–	1	1
	Hash	2	–	1	1	2	–	1	3
	En/decryption	5	–	5	5	3	–	2	1
Subsequent phase	Random number	–	–	2	1	–	–	1	1
	Hash	–	–	1	0	–	–	3	3
	En/decryption	–	–	4	4	–	–	1	1
Handover phase	Random number	–	0	3	1	–	1	1	1
	Hash	–	1	1	0	–	3	2	5
	En/decryption	–	3	5	5	–	3	2	1

L2300 CPU, and the size of inputted message is 512 kb [28].

Unquestionably, we can know that the symmetric cryptosystem takes much more execution time than the hash function needs. That is, the number of symmetric key encryption must dominate the efficiency of authentication. According to Sects. 3.1 and 3.2, it is clear that each participant in the intra-domain needs to execute the random number generation, the one-way hash function and symmetric en/decryption for authentication. The numbers of these operations performed by involved participants are listed in Table 5. No matter which phase we concern about, the new method outperforms the related work in terms of the authentication performance. This is due to that the number of required symmetric en/decryption in the related work is more than that in the new method.

As to the case of inter-domain, we summarize the number of operations performed by involved users in Table 6. The total number of symmetric en/decryption demanded in EAP-OSNP is nineteen while that in the proposed method is twelve. This comparison has helped highlight the superiority of the new method over the related work.

5 Mutual authentication by BAN logic

In this section, we give the formal analysis of the new authentication protocol via BAN logic. The notations of derivation follow those in BAN logic model [4, 16]. *P* and *Q* denote principals; *X* and *Y* range over statements; *K* denotes a symmetric key. Others are shown in Table 7.

For the correctness derivation, we apply four logic postulates including *message-meaning rule*, *nonce-verification rule*, *jurisdiction rule*, and *freshness conjuncatenation rule*. The detail can be referred to [4, 16].

5.1 Correctness of authentication within the intra domain

The accuracy of initial, subsequent, and handover authentication is discussed in Sects. 5.1.1–5.1.3, respectively.

Table 7 BAN notations

$P \mid\equiv X$	*P* believes in *X*
$P \triangleleft X$	*P* sees *X* (*P* receives *X*)
$P \mid\sim X$	*P* once said *X* (*P* sent *X*)
$P \mid\Rightarrow X$	*P* has jurisdiction over *X* (*P* can control *X*)
$\#(X)$	The statement *X* is fresh
$P \xleftrightarrow{Y} Q$	*P* and *Q* can use the shared key *K* to communicate
$P \overset{Y}{\rightleftharpoons} Q$	The statement *Y* is a secret known only to *P* and *Q*
$\{X\}_K$	The statement *X* encrypted by the secret key *K*
$\langle X\rangle_Y$	The statement *X* combined with the formula *Y*

Table 6 Performance comparison: computation costs in inter-domain

Inter-domain phase	EAP-OSNP [8]					Proposed				
Operation	KDC_R	KDC_Y	KDC_X	S_Y	U_X	KDC_R	KDC_Y	KDC_X	S_Y	U_X
Random number	2	1	0	2	1	0	2	1	0	1
Hash	0	1	0	0	0	0	2	2	1	5
En/decryption	3	3	3	5	5	2	4	3	2	1

5.1.1 Correctness of initial authentication

In the initial authentication protocol, U and S can establish a shared key K_T and negotiate a common secret T_1 via KDC. K_T and T_1 are then kept for next authentication. The protocol must achieve the following goals.

G 1.	$U\mid\!\equiv U\overset{K_T}{\longleftrightarrow}S$	**G 2**.	$S\mid\!\equiv U\overset{K_T}{\longleftrightarrow}S$
G 3.	$U\mid\!\equiv S\mid\!\equiv U\overset{K_T}{\longleftrightarrow}S$	**G 4**.	$S\mid\!\equiv U\mid\!\equiv U\overset{K_T}{\longleftrightarrow}S$
G 5.	$U\mid\!\equiv U\overset{T_1}{\rightleftharpoons}S$	**G 6**.	$S\mid\!\equiv U\overset{T_1}{\rightleftharpoons}S$
G 7.	$U\mid\!\equiv S\mid\!\equiv U\overset{T_1}{\rightleftharpoons}S$	**G 8**.	$S\mid\!\equiv U\mid\!\equiv U\overset{T_1}{\rightleftharpoons}S$

According to the analyzing procedure of BAN logic, we have to transfer the protocol into a formal form first. During the transformation, we employ $\langle X\rangle_K$ to replace the hash value $H(K\parallel X)$; thus, we obtain the following messages

Msg 1.	$U\rightarrow S:$	$\{T_1\}_{K_{UK}}$
Msg 2.	$S\rightarrow KDC:$	$\{T_1\}_{K_{UK}},\ \{\ N_S\ \}_{K_{SK}}$
Msg 3.	$KDC\rightarrow S:$	$\{\langle\ T_K\ \rangle_{K_{UK}},\ N_S,\ K_T,\ T_1\ ,T_K\}_{K_{SK}}$
Msg 4.	$S\rightarrow U:$	$\langle\ T_1,T_K\ \rangle_{K_{UK}},\ T_K$
Msg 5.	$U\rightarrow S:$	$\langle\ T_1,T_K\ \rangle_{K_T}$

Later on, we transfer the above messages into the idealized form as follows.

I 1.	$U\rightarrow S:$	$\{U\overset{T_1}{\rightleftharpoons}S\}_{K_{UK}}$
I 2.	$S\rightarrow KDC:$	$\{U\overset{T_1}{\rightleftharpoons}S\}_{K_{UK}},\ \{N_S\}_{K_{SK}}$
I 3.	$KDC\rightarrow S:$	$\{\langle U\overset{T_K}{\rightleftharpoons}KDC\rangle_{K_{UK}},N_S,U\overset{K_T}{\longleftrightarrow}S,U\overset{T_1}{\rightleftharpoons}S,U\overset{T_K}{\rightleftharpoons}KDC\}_{K_{SK}}$
I 4.	$S\rightarrow U:$	$\langle U\overset{T_1}{\rightleftharpoons}S,U\overset{T_K}{\rightleftharpoons}KDC,U\overset{K_T}{\longleftrightarrow}S\rangle_{K_{UK}},\langle\overset{T_K}{\rightleftharpoons}KDC\rangle_{K_{UK}}$
I 5.	$U\rightarrow S:$	$\langle U\overset{T_1}{\rightleftharpoons}S,U\overset{T_K}{\rightleftharpoons}KDC,U\overset{K_T}{\longleftrightarrow}S\rangle_{KT}$

Finally, we give the following fundamental assumptions:

A 1.	$U\mid\!\equiv U\overset{K_{UK}}{\longleftrightarrow}KDC$	**A 2**.	$KDC\mid\!\equiv U\overset{K_{UK}}{\longleftrightarrow}KDC$
A 3.	$S\mid\!\equiv S\overset{K_{SK}}{\longleftrightarrow}KDC$	**A 4**.	$KDC\mid\!\equiv S\overset{K_{SK}}{\longleftrightarrow}KDC$
A 5.	$U\mid\!\equiv\#(T_1)$	**A 6**.	$S\mid\!\equiv\#(N_S)$
A 7.	$KDC\mid\!\equiv\#(T_K)$	**A 8**.	$U\mid\!\equiv KDC\mid\Rightarrow(U\overset{K_T}{\longleftrightarrow}S)$
A 9.	$S\mid\!\equiv KDC\mid\Rightarrow(U\overset{K_T}{\longleftrightarrow}S)$	**A 10**.	$KDC\mid\!\equiv U\mid\Rightarrow(U\overset{T_1}{\rightleftharpoons}S)$
A 11.	$S\mid\!\equiv U\mid\Rightarrow(U\overset{T_1}{\rightleftharpoons}S)$		

Derivation of goals:

It is obviously that G 5 is true since T_1 is generated by U. According to I 3, A 3, and the *message-meaning rule*, we can further get,

D 1. $$S\mid\!\equiv KDC\mid\sim\left(\langle U\overset{T_K}{\rightleftharpoons}KDC\rangle_{K_{UK}},N_S,U\overset{K_T}{\longleftrightarrow}S,\right.$$
$$\left.U\overset{T_1}{\rightleftharpoons}S,\quad U\overset{T_K}{\rightleftharpoons}KDC\right).$$

From A 6 and the *freshness conjuncatenation rule*, we have

D 2. $$S\mid\!\equiv\#\left(\langle U\overset{T_K}{\rightleftharpoons}KDC\rangle_{K_{UK}},N_S,U\overset{K_T}{\longleftrightarrow}S,\right.$$
$$\left.U\overset{T_1}{\rightleftharpoons}S,\quad U\overset{T_K}{\rightleftharpoons}KDC\right).$$

Consequently, by D 1, D 2, and the *nonce-verification rule*, we can derive

D 3. $$S\mid\!\equiv KDC\mid\!\equiv\left(\langle U\overset{T_K}{\rightleftharpoons}KDC\rangle_{K_{UK}},N_S,U\overset{K_T}{\longleftrightarrow}S,\right.$$
$$\left.U\overset{T_1}{\rightleftharpoons}S,\quad U\overset{T_K}{\rightleftharpoons}KDC\right),$$

and thus leading to

D 4. $S\mid\!\equiv KDC\mid\!\equiv\langle U\overset{T_K}{\rightleftharpoons}KDC\rangle_{K_{UK}}$,

D 5. $S\mid\!\equiv KDC\mid\!\equiv U\overset{K_T}{\longleftrightarrow}S$,

D 6. $S\mid\!\equiv KDC\mid\!\equiv U\overset{T_1}{\rightleftharpoons}S$,

D 7. $S\mid\!\equiv KDC\mid\!\equiv U\overset{T_K}{\rightleftharpoons}KDC$.

Given D 5, A 9, D 6, and A 11, we can apply the *jurisdiction rule* to infer

D 8. $S\mid\!\equiv U\overset{K_T}{\longleftrightarrow}S$,

which proves the goal of G 2. And we can obtain the following message from I 4,

D 9. $$U\triangleleft\left(\langle U\overset{T_1}{\rightleftharpoons}S,U\overset{T_K}{\rightleftharpoons}KDC,U\overset{K_T}{\longleftrightarrow}S\rangle_{K_{UK}}\right).$$

We subsequently transform A 1 into

D 10. $U\mid\!\equiv U\overset{K_{UK}}{\rightleftharpoons}KDC$

since K_{UK} can be regarded as a secret. From D 9, D 10, and the *message-meaning rule*, we have

D 11. $$U\mid\!\equiv KDC\mid\sim\left(U\overset{T_1}{\rightleftharpoons}S,\quad U\overset{T_K}{\rightleftharpoons}KDC,\quad U\overset{K_T}{\longleftrightarrow}S\right).$$

From A 5 and the *freshness conjuncatenation rule*, we have

D 12. $U \mid\equiv \#\left(U \overset{T_1}{\rightleftharpoons} S, \quad U \overset{T_K}{\rightleftharpoons} KDC, \quad U \overset{K_T}{\longleftrightarrow} S\right).$

Consequently, by D 11, D 12, and the *nonce-verification rule*, we can derive

D 13. $U \mid\equiv KDC \mid\equiv \left(U \overset{T_1}{\rightleftharpoons} S, \quad U \overset{T_K}{\rightleftharpoons} KDC, \quad U \overset{K_T}{\longleftrightarrow} S\right),$

and thus leading to

D 14. $U \mid\equiv KDC \mid\equiv U \overset{K_T}{\longleftrightarrow} S.$

According to A 8, D 14 and the *jurisdiction rule*, we can infer

D 15. $U \mid\equiv U \overset{K_T}{\longleftrightarrow} S,$

This leads to the achievement of the goal G 1. Now, U can make sure that KDC has confirmed the secret T_1 shared between U and S. We can naturally derive that

D 16. $U \mid\equiv S \mid\equiv U \overset{T_1}{\rightleftharpoons} S,$

D 17. $U \mid\equiv S \mid\equiv U \overset{K_T}{\longleftrightarrow} S,$

since KDC must have sent T_1 to S before D 9. This leads to the achievement of the goals G 3 and G 7. We subsequently transform D 8 into

D 18. $S \mid\equiv U \overset{K_T}{\rightleftharpoons} S.$

From I 5, D 18, and the *message-meaning rule*, we can obtain

D 19. $S \mid\equiv U \mid\sim \left(U \overset{T_1}{\rightleftharpoons} S, \quad U \overset{T_K}{\rightleftharpoons} KDC, \quad U \overset{K_T}{\longleftrightarrow} S\right),$

Next, based on D 12 and D 20, we get

D 20. $S \mid\equiv U \mid\equiv \left(U \overset{T_1}{\rightleftharpoons} S, \quad U \overset{T_K}{\rightleftharpoons} KDC, \quad U \overset{K_T}{\longleftrightarrow} S\right),$

under the *nonce-verification rule*, and thus leading to

D 21. $S \mid\equiv U \mid\equiv U \overset{T_1}{\rightleftharpoons} S,$

D 22. $S \mid\equiv U \mid\equiv U \overset{K_T}{\longleftrightarrow} S.$

Based on A 11, D 21, and the *jurisdiction rule*, we can derive

D 23. $S \mid\equiv U \overset{T_1}{\rightleftharpoons} S.$

G 4, G 6, and G 8 then are proved. The proof of initial authentication is completed.

5.1.2 Correctness of subsequent authentication

Suppose that U and S have achieved the $(i-1)$th mutual authentication through the previous shared key K_T and the secret T_{i-1}. Consequently, we give the following goals for the ith authentication.

G 1.	$U \mid\equiv U \overset{T_i}{\rightleftharpoons} S$	**G 2**.	$S \mid\equiv U \overset{T_i}{\rightleftharpoons} S$
G 3.	$U \mid\equiv S \mid\equiv U \overset{T_i}{\rightleftharpoons} S$	**G 4**.	$S \mid\equiv U \mid\equiv U \overset{T_i}{\rightleftharpoons} S$
G 5.	$U \mid\equiv U \overset{T_S}{\rightleftharpoons} S$	**G 6**.	$S \mid\equiv U \overset{T_S}{\rightleftharpoons} S$
G 7.	$U \mid\equiv U \overset{K'_T}{\longleftrightarrow} S$	**G 8**.	$S \mid\equiv U \overset{K'_T}{\longleftrightarrow} S$

We then convert the protocol into the formal message as follows.

Msg 1.	$U \rightarrow S:$	$\{T_i,\ \langle\ T_i\ \rangle_{K_T}\}_{K_T}$
Msg 2.	$S \rightarrow U:$	$\langle T_i, T_S\rangle_{K'_T}, T_S$

The formal messages are then transferred into the following idealized form.

I 1.	$U \rightarrow S:$	$\{T_i, \langle U \overset{T_i}{\rightleftharpoons} S\rangle_{K_T}\}_{K_T}$
I 2.	$S \rightarrow U:$	$\langle U \overset{T_i}{\rightleftharpoons} S, U \overset{T_S}{\rightleftharpoons} S, U \overset{K'_T}{\longleftrightarrow} S\rangle_{K'_T}, \langle U \overset{T_S}{\rightleftharpoons} S\rangle_{K_T}$

The essential assumptions of the derivation are defined below.

A 1.	$U \mid\equiv U \overset{K_T}{\longleftrightarrow} S$	**A 2**.	$S \mid\equiv U \overset{K_T}{\longleftrightarrow} S$
A 3.	$U \mid\equiv \#(T_i)$	**A 4**.	$S \mid\equiv \#(T_S)$
A 5.	$S \mid\equiv U \mid\Rightarrow (U \overset{T_i}{\rightleftharpoons} S)$	**A 6**.	$U \mid\equiv S \mid\Rightarrow (U \overset{T_S}{\rightleftharpoons} S)$

Derivation of goals:

According to A 3, A 4, and the fact that T_i and T_S are generated by U and S respectively, G 1 and G 6 hold naturally. Based on I 1 and A 2, the *message-meaning rule* shows that

D 1. $S \mid\equiv U \mid\sim \left(T_i, \langle U \overset{T_i}{\rightleftharpoons} S\rangle_{K_T}\right).$

Since A 3 is a valid timestamp, we can deduce that

D 2. $U \mid\equiv \#\left(T_i, \langle U \overset{T_i}{\rightleftharpoons} S\rangle_{K_T}\right).$

By D 1, D 2, and the *nonce-verification rule*, we obtain

D 3. $S \mid\equiv U \mid\equiv \left(T_i, \langle U \overset{T_i}{\rightleftharpoons} S\rangle_{K_T}\right),$

and thus leading to

D 4. $S\mid\equiv U\mid\equiv\left(\langle U \overset{T_i}{\rightleftharpoons} S\rangle_{K_T}\right)$.

We subsequently transform A 2 into

D 5. $S\mid\equiv U \overset{K_T}{\rightleftharpoons} S$.

Based on D 4 and D 5, we can infer the following naturally

D 6. $S\mid\equiv U\mid\equiv U \overset{T_i}{\rightleftharpoons} S$.

This leads to the achievement of the goal G 4. According to A 5, D 6, and the *jurisdiction rule*, we can derive the following

D 7. $S\mid\equiv U \overset{T_i}{\rightleftharpoons} S$,

which proves G 2. From I 2 and A 1, we employ the *message-meaning rule* to derive that

D 8. $U\mid\equiv S\mid\sim U \overset{T_S}{\rightleftharpoons} S$.

Since A 4 is a valid timestamp, we can deduce that

D 9. $S\mid\equiv\#\left(U \overset{T_S}{\rightleftharpoons} S\right)$.

By D 8, D 9, and the *nonce-verification rule*, we obtain

D 10. $U\mid\equiv S\mid\equiv U \overset{T_S}{\rightleftharpoons} S$.

Based to A 6, D 10, and the *jurisdiction rule*, we can confirm G 5 as the following derivation,

D 11. $U\mid\equiv U \overset{T_S}{\rightleftharpoons} S$.

According to the fact that we know G 1, A 1, and D 11, we can naturally derive

D 12. $U\mid\equiv U\overset{K'_T}{\longleftrightarrow}S$,

which proves G 7. We subsequently transform D 12 into

D 13. $U\mid\equiv U \overset{K'_T}{\rightleftharpoons} S$.

According to I 2 and D 13, we employ the *message-meaning rule* to derive that

D 14. $U\mid\equiv S\mid\sim\left(U \overset{T_i}{\rightleftharpoons} S, U \overset{T_S}{\rightleftharpoons} S, U\overset{K'_T}{\longleftrightarrow}S\right)$.

From A 4 and the *freshness conjuncatenation rule*, we have

D 15. $S\mid\equiv\#\left(U \overset{T_i}{\rightleftharpoons} S, U \overset{T_S}{\rightleftharpoons} S, U\overset{K'_T}{\longleftrightarrow}S\right)$

By D 14, D 15, and the *nonce-verification rule*, we obtain

D 16. $U\mid\equiv S\mid\equiv\left(U \overset{T_i}{\rightleftharpoons} S, U \overset{T_S}{\rightleftharpoons} S, U\overset{K'_T}{\longleftrightarrow}S\right)$,

and thus leading to

D 17. $U\mid\equiv S\mid\equiv U \overset{T_i}{\rightleftharpoons} S$,

D 18. $U\mid\equiv S\mid\equiv U\overset{K'_T}{\longleftrightarrow}S$.

This leads to the achievement of the goal G 3. Since we know A 2, G 6, D 7, and D 18, we can naturally derive

D 19. $S\mid\equiv U\overset{K'_T}{\longleftrightarrow}S$,

which confirm G 8.

5.1.3 Correctness of handover authentication

U_x and the new server S_{x_2} are the targets that need to confirm the mutual authentication in the handover authentication phase. Finally, U_x must be able to share with S_{x_2} a secret T_j and new shared key K'_T. Based on above results, we define the following goals.

G 1.	$U_x\mid\equiv U_x\overset{K'_T}{\longleftrightarrow}S_{x_2}$	**G 2**.	$S_{x_2}\mid\equiv U_x\overset{K'_T}{\longleftrightarrow}S_{x_2}$
G 3.	$U_x\mid\equiv S_{x_2}\mid\equiv U_x\overset{K'_T}{\longleftrightarrow}S_{x_2}$	**G 4**.	$S_{x_2}\mid\equiv U_x\mid\equiv U_x\overset{K'_T}{\longleftrightarrow}S_{x_2}$
G 5.	$U_x\mid\equiv U_x \overset{T_j}{\rightleftharpoons} S_{x_2}$	**G 6**.	$S_{x_2}\mid\equiv U_x \overset{T_j}{\rightleftharpoons} S_{x_2}$
G 7.	$U_x\mid\equiv S_{x_2}\mid\equiv U_x \overset{T_j}{\rightleftharpoons} S_{x_2}$	**G 8**.	$S_{x_2}\mid\equiv U_x\mid\equiv U_x \overset{T_j}{\rightleftharpoons} S_{x_2}$

The formal form of the communicating messages is listed as following.

Msg 1.	$U_x \rightarrow S_{x_2}$:	$\{T_j, \langle T_j\rangle_{K_T}\}_{K_T}$
Msg 2.	$S_{x_2} \rightarrow S_{x_1}$:	$\{\{T_j, \langle T_j\rangle_{K_T}\}_{K_T}, N_{S_{x_2}}\}_{K_G}$
Msg 3.	$S_{x_1} \rightarrow S_{x_2}$:	$\{\langle T_{S_{x_1}}\rangle_{K_T}, K'_T, N_{S_{x_2}}, T_j, T_{S_{x_1}}\}_{K_G}$
Msg 4.	$S_{x_2} \rightarrow U_x$:	$\langle T_{S_{x_1}}\rangle_{K_T}, \langle T_j, T_{S_{x_1}}\rangle_{K'_T}, T_{S_{x_1}}$
Msg 5.	$U_x \rightarrow S_{x_2}$:	$\langle T_j, T_{S_{x_1}}\rangle_{K'_T}$

We further convert these formal messages into the idealized form.

I 1.	$U_x \rightarrow S_{x_2}$:	$\{T_j, \langle U_x \overset{T_j}{\rightleftharpoons} S_{x_2}\rangle_{K_T}\}_{K_T}$
I 2.	$S_{x_2} \rightarrow S_{x_1}$:	$\{\{T_j, \langle U_x \overset{T_j}{\rightleftharpoons} S_{x_2}\rangle_{K_T}\}_{K_T}, N_{S_{x_2}}\}_{K_G}$
I 3.	$S_{x_1} \rightarrow S_{x_2}$:	$\{\langle U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1}\rangle_{K_T}, U_x\overset{K'_T}{\longleftrightarrow}S_{x_2}, N_{S_{x_2}}, U_x \overset{T_j}{\rightleftharpoons} S_{x_2},$ $U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1}\}_{K_G}$
I 4.	$S_{x_2} \rightarrow U_x$:	$\langle U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1}\rangle_{K_T}, \langle U_x \overset{T_j}{\rightleftharpoons} S_{x_2}, U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1}, U_x\overset{K'_T}{\longleftrightarrow}S_{x_2},\rangle_{K'_T}$
I 5.	$U_x \rightarrow S_{x_2}$:	$\langle U_x \overset{T_j}{\rightleftharpoons} S_{x_2}, U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1}, U_x\overset{K'_T}{\longleftrightarrow}S_{x_2},\rangle_{K'_T}$

Before the derivation, we give the following assumptions.

A 1. $U_x \mid\equiv U_x \overset{K_T}{\longleftrightarrow} S_{x_1}$

A 2. $S_{x_1} \mid\equiv U_x \overset{K_T}{\longleftrightarrow} S_{x_1}$

A 3. $S_{x_1} \mid\equiv S_{x_1} \overset{K_G}{\longleftrightarrow} S_{x_2}$

A 4. $S_{x_2} \mid\equiv S_{x_1} \overset{K_G}{\longleftrightarrow} S_{x_2}$

A 5. $U_x \mid\equiv \#(T_j)$

A 6. $S_{x_1} \mid\equiv \#(T_{S_{x_1}})$

A 7. $S_{x_2} \mid\equiv \#(N_{S_{x_2}})$

A 8. $U_x \mid\equiv S_{x_1} \mid\Rightarrow (U_x \overset{K'_T}{\longleftrightarrow} S_{x_2})$

A 9. $S_{x_2} \mid\equiv S_{x_1} \mid\Rightarrow (U_x \overset{K'_T}{\longleftrightarrow} S_{x_2})$

A 10. $S_{x_2} \mid\equiv U_x \mid\Rightarrow (U_x \overset{T_j}{\rightleftharpoons} S_{x_2})$

A 11. $S_{x_2} \mid\equiv S_{x_1} \mid\Rightarrow (U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1})$

Derivation of goals:

G 5 is obtained immediately since T_j is generated by U.

According to the *message-meaning rule*, we can get

$$\textbf{D 1}.\quad S_{x_2} \mid\equiv S_{x_1} \mid\sim \left(\langle U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1} \rangle_{K_T}, U_x \overset{K'_T}{\longleftrightarrow} S_{x_2}, N_{S_{x_2}}, U_x \overset{T_j}{\rightleftharpoons} S_{x_2}, U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1} \right),$$

based on I 3 and A 4. Given A 7 and D 1, we can infer that

$$\textbf{D 2}.\quad S_{x_2} \mid\equiv \# \left(\langle U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1} \rangle_{K_T}, U_x \overset{K'_T}{\longleftrightarrow} S_{x_2}, N_{S_{x_2}}, U_x \overset{T_j}{\rightleftharpoons} S_{x_2}, U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1} \right).$$

Then we derive the followings with D 1 and D 2, and the *nonce-verification rule*,

$$\textbf{D 3}.\quad S_{x_2} \mid\equiv S_{x_1} \mid\equiv \left(\langle U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1} \rangle_{K_T}, U_x \overset{K'_T}{\longleftrightarrow} S_{x_2}, N_{S_{x_2}}, U_x \overset{T_j}{\longleftrightarrow} S_{x_2}, U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1} \right),$$

and thus leading to

$$\textbf{D 4}.\quad S_{x_2} \mid\equiv S_{x_1} \mid\equiv U_x \overset{K'_T}{\longleftrightarrow} S_{x_2}.$$

Taking D 4, A 9, and the *jurisdiction rule*, we can conclude the followings,

$$\textbf{D 5}.\quad S_{x_2} \mid\equiv U_x \overset{K'_T}{\longleftrightarrow} S_{x_2},$$

thus leading to G 2.

From A 1, I 4, and the *message-meaning rule*, we can get

$$\textbf{D 6}.\quad U_x \mid\equiv S_{x_1} \mid\sim U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1}.$$

Based on A 6 and the *freshness conjuncatenation rule*, we can infer that

$$\textbf{D 7}.\quad S_{x_1} \mid\equiv \# \left(U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1} \right).$$

We then apply D 6, D 7 and the *nonce-verification rule* to derive

$$\textbf{D 8}.\quad U_x \mid\equiv S_{x_1} \mid\equiv U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1}.$$

Using the *jurisdiction rule*, we can apply A 8 and D 8 to derive

$$\textbf{D 9}.\quad U_x \mid\equiv U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1}.$$

Therefore, we know A 1, G 5, A 8 and D 9 that it can easily derive G 1

$$\textbf{D 10}.\quad U_x \mid\equiv U_x \overset{K'_T}{\longleftrightarrow} S_{x_2}.$$

We subsequently transform D 10 into

$$\textbf{D 11}.\quad U_x \mid\equiv U_x \overset{K'_T}{\rightleftharpoons} S_{x_2}.$$

Based on D 4 and D 11, we employ the *message-meaning rule* to derive that

$$\textbf{D 12}.\quad U_x \mid\equiv S_{x_2} \mid\sim \left(U_x \overset{T_j}{\rightleftharpoons} S_{x_2}, U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1}, U_x \overset{K'_T}{\longleftrightarrow} S_{x_2} \right).$$

We also have the following result from A 5 and the *freshness conjuncatenation rule*,

$$\textbf{D 13}.\quad U_x \mid\equiv \# \left(U_x \overset{T_j}{\rightleftharpoons} S_{x_2}, U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1}, U_x \overset{K'_T}{\longleftrightarrow} S_{x_2} \right).$$

We subsequently have

$$\textbf{D 14}.\quad U_x \mid\equiv S_{x_2} \mid\equiv \left(U_x \overset{T_j}{\rightleftharpoons} S_{x_2}, U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1}, U_x \overset{K'_T}{\longleftrightarrow} S_{x_2} \right),$$

according to D 12, D 13, and the *nonce-verification rule*. Follow the formula D 14, we get

$$\textbf{D 15}.\quad U_x \mid\equiv S_{x_2} \mid\equiv U_x \overset{T_j}{\rightleftharpoons} S_{x_2},$$

$$\textbf{D 16}.\quad U_x \mid\equiv S_{x_2} \mid\equiv U_x \overset{K'_T}{\longleftrightarrow} S_{x_2}.$$

This leads to the achievement of the goal G 3 and G 7. Next, we transform D 5 into

$$\textbf{D 17}.\quad S_{x_2} \mid\equiv U_x \overset{K'_T}{\rightleftharpoons} S_{x_2}.$$

Using the *message-meaning rule*, we can apply I 5 and D 17 to derive

$$\textbf{D 18}.\quad S_{x_2} \mid\equiv U_x \mid\sim \left(U_x \overset{T_j}{\rightleftharpoons} S_{x_2}, U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1}, U_x \overset{K'_T}{\longleftrightarrow} S_{x_2} \right).$$

From D 13, D 18 and the *nonce-verification rule*, we obtain

D 19. $S_{x_2} \mid\equiv U_x \mid\equiv \left(U_x \overset{T_j}{\rightleftharpoons} S_{x_2}, U_x \overset{T_{S_{x_1}}}{\rightleftharpoons} S_{x_1}, U_x \overset{K'_T}{\longleftrightarrow} S_{x_2} \right)$,

and thus leading to

D 20. $S_{x_2} \mid\equiv U_x \mid\equiv U_x \overset{T_j}{\rightleftharpoons} S_{x_2}$,

D 21. $S_{x_2} \mid\equiv U_x \mid\equiv U_x \overset{K'_T}{\longleftrightarrow} S_{x_2}$.

The goals of G 4 and G 8 hold. According to A 10 and D 20, we can apply the *jurisdiction rule* to derive

D 22. $S_{x_2} \mid\equiv U_x \overset{T_j}{\rightleftharpoons} S_{x_2}$.

Hence, the goal of G 6 is satisfied. All beliefs are confirmed.

5.2 Correctness of authentication within the inter domain

In this case, U_x and S_{y_1} have to proceed the mutual authentication with the help of KDC_x. First U_x has to establish a secret T_{U_x} shared with S_{y_1}, KDC_y, KDC_R, and KDC_x. Then KDC_y generates a random nonce shared with KDC_x. In addition, KDC_y establishes a secret key K_{T_y} shared with U_x and S_{y_1}. Based on above results, we aim to prove that the new protocol can achieve the goals of G1 to G 10.

G 1.	$U_x \mid\equiv U_x \overset{K_{T_y}}{\longleftrightarrow} S_y$	**G 2**.	$S_y \mid\equiv U_x \overset{K_{T_y}}{\longleftrightarrow} S_y$
G 3.	$U_x \mid\equiv S_y \mid\equiv U_x \overset{K_{T_y}}{\longleftrightarrow} S_y$	**G 4**.	$S_y \mid\equiv U_x \mid\equiv U_x \overset{K_{T_y}}{\longleftrightarrow} S_y$
G 5.	$U_x \mid\equiv U_x \overset{T_{U_x}}{\rightleftharpoons} S_y$	**G 6**.	$S_y \mid\equiv U_x \overset{T_{U_x}}{\rightleftharpoons} S_y$
G 7.	$U_x \mid\equiv S_y \mid\equiv U_x \overset{T_{U_x}}{\rightleftharpoons} S_y$	**G8**.	$S_y \mid\equiv U_x \mid\equiv U_x \overset{T_{U_x}}{\rightleftharpoons} S_y$

The formal format of the communicating message is described below.

Msg 1.	$U_x \rightarrow S_{y_1}$:	$\{T_{U_x}\}_{K_{UK_x}}$
Msg 2.	$S_{y_1} \rightarrow KDC_y$:	$\{\{T_{U_x}\}_{K_{UK_x}}\}_{K_{SK_y}}$
Msg 3.	$KDC_y \rightarrow KDC_R$:	$\{\{T_{U_x}\}_{K_{UK_x}},\ N_y\}_{K_{K_{Ry}}}$
Msg 4.	$KDC_R \rightarrow KDC_x$:	$\{\{T_{U_x}\}_{K_{UK_x}},\ N_y\}_{K_{K_{Rx}}}$
Msg 5.	$KDC_x \rightarrow KDC_y$:	$\{\langle T_{U_x}, T_{K_x}\rangle_{K_{UK_x}}, K_{T_x}, T_{U_x}, T_{K_x}\}_{N_y}$
Msg 6.	$KDC_y \rightarrow S_{y_1}$:	$\{\langle T_{U_x}, T_{K_x}\rangle_{K_{UK_x}}, \langle T_{U_x}, T_{K_x}, T_{K_y}\rangle_{K_{T_x}},$ $K_{T_y}, T_{U_x}, T_{K_x}, T_{K_y}\}_{K_{SK_y}}$
Msg 7.	$S_{y_1} \rightarrow U_x$:	$\langle T_{U_x}, T_{K_x}\rangle_{K_{UK_x}}, \langle T_{U_x}, T_{K_x}, T_{K_y}\rangle_{K_{T_x}},$ T_{K_x}, T_{K_y}
Msg 8.	$U_x \rightarrow S_{y_1}$:	$\langle T_{U_x}, T_{K_x}, T_{K_y}\rangle_{K_{T_y}}$

Further, we transfer the above messages into the idealized form as follows.

I 1.	$U_x \rightarrow S_{y_1}$:	$\{U_x \overset{T_{U_x}}{\rightleftharpoons} S_y\}_{K_{UK_x}}$
I 2.	$S_{y_1} \rightarrow KDC_y$:	$\{\{U_x \overset{T_{U_x}}{\rightleftharpoons} S_y\}_{K_{UK_x}}\}_{K_{SK_y}}$
I 3.	$KDC_y \rightarrow KDC_R$:	$\{\{U_x \overset{T_{U_x}}{\rightleftharpoons} S_y\}_{K_{UK_x}},\ KDC_x \overset{N_y}{\longleftrightarrow} KDC_y\}_{K_{K_{Ry}}}$
I 4.	$KDC_R \rightarrow KDC_x$:	$\{\{U_x \overset{T_{U_x}}{\rightleftharpoons} S_y\}_{K_{UK_x}},\ KDC_x \overset{N_y}{\longleftrightarrow} KDC_y\}_{K_{K_{Rx}}}$
I 5.	$KDC_x \rightarrow KDC_y$:	$\{\langle U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{K_{T_x}}{\longleftrightarrow} S_y\rangle_{K_{UK_x}},$ $U_x \overset{K_{T_x}}{\longleftrightarrow} S_y, U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x\}_{N_y}$
I 6.	$KDC_y \rightarrow S_{y_1}$:	$\{\langle U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{K_{T_x}}{\longleftrightarrow} S_y\rangle_{K_{UK_x}},$ $\langle U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{T_{K_y}}{\rightleftharpoons} KDC_y,$ $U_x \overset{K_{T_y}}{\longleftrightarrow} S_y\rangle_{K_{T_x}}\ U_x \overset{K_{T_y}}{\longleftrightarrow} S_y, U_x \overset{T_{U_x}}{\rightleftharpoons} S_y,$ $U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_y\}_{K_{SK_y}}$
I 7.	$S_{y_1} \rightarrow U_x$:	$\langle U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{K_{T_x}}{\longleftrightarrow} S_y\rangle_{K_{UK_x}},$ $\langle U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{T_{K_y}}{\rightleftharpoons} KDC_y,$ $U_x \overset{K_{T_y}}{\longleftrightarrow} S_y\rangle_{K_{T_x}}$
I 8.	$U_x \rightarrow S_{y_1}$:	$\langle U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{T_{K_y}}{\rightleftharpoons} KDC_y,$ $U_x \overset{K_{T_y}}{\longleftrightarrow} S_y\rangle_{K_{T_y}}$

We then define the basic assumptions in the following.

A 1.	$U_x \mid\equiv U_x \overset{K_{UK_x}}{\longleftrightarrow} KDC_x$	**A 2**.	$KDC_x \mid\equiv U_x \overset{K_{UK_x}}{\longleftrightarrow} KDC_x$
A 3.	$S_y \mid\equiv S_y \overset{K_{SK_y}}{\longleftrightarrow} KDC_y$	**A 4**.	$KDC_y \mid\equiv S_y \overset{K_{SK_y}}{\longleftrightarrow} KDC_y$
A 5.	$KDC_x \mid\equiv KDC_x \overset{K_{K_{Rx}}}{\longleftrightarrow} KDC_R$	**A 6**.	$KDC_R \mid\equiv KDC_x \overset{K_{K_{Rx}}}{\longleftrightarrow} KDC_R$
A 7.	$KDC_y \mid\equiv KDC_y \overset{K_{K_{Ry}}}{\longleftrightarrow} KDC_R$	**A 8**.	$KDC_R \mid\equiv KDC_y \overset{K_{K_{Ry}}}{\longleftrightarrow} KDC_R$
A 9.	$U_x \mid\equiv \#(T_{U_x})$	**A 10**.	$KDC_x \mid\equiv \#(T_{K_x})$
A 11.	$KDC_y \mid\equiv \#(T_{K_y})$	**A 12**.	$KDC_y \mid\equiv \#(N_y)$
A 13.	$U_x \mid\equiv KDC_y \mid\Rightarrow (U_x \overset{K_{T_y}}{\longleftrightarrow} S_y)$	**A 14**.	$U_x \mid\equiv KDC_x \mid\Rightarrow (U_x \overset{K_{T_x}}{\longleftrightarrow} S_y)$
A 15.	$S_y \mid\equiv U_x \mid\Rightarrow (U_x \overset{T_{U_x}}{\rightleftharpoons} S_y)$	**A 16**.	$S_y \mid\equiv KDC_y \mid\Rightarrow (U_x \overset{K_{T_y}}{\longleftrightarrow} S_y)z$

Derivation of goals:

G 5 is obtained immediately since T_{U_x} is generated by U. According to the *message-meaning rule*, we can get

D 1.

$$S_y \mid\equiv KDC_y \mid\sim \left(\left\langle U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{K_{T_x}}{\longleftrightarrow} S_y \right\rangle_{K_{UK_x}} \right),$$

D 2. $$S_y \mid\equiv KDC_y \mid\sim \left(\left\langle U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{T_{K_y}}{\rightleftharpoons} KDC_y, U_x \overset{K_{T_y}}{\longleftrightarrow} S_y \right\rangle_{K_{T_x}} \right),$$

D 3. $$S_y \mid\equiv KDC_y \mid\sim \left(U_x \overset{K_{T_y}}{\longleftrightarrow} S_y, U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{T_{K_y}}{\rightleftharpoons} KDC_y \right),$$

based on A 3 and I 6. Given D 3 and A 11, we can infer that

D 4. $$KDC_y \mid\equiv \#\left(U_x \overset{K_{T_y}}{\longleftrightarrow} S_y, U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{T_{K_y}}{\rightleftharpoons} KDC_y \right).$$

Then we derive the followings with D 3 and D 4, and the *nonce-verification rule*,

D 5. $$S_y \mid\equiv KDC_y \mid\equiv \left(U_x \overset{K_{T_y}}{\longleftrightarrow} S_y, U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{T_{K_y}}{\rightleftharpoons} KDC_y \right),$$

and thus leading to

D 6. $$S_y \mid\equiv KDC_y \mid\equiv \left(U_x \overset{K_{T_y}}{\longleftrightarrow} S_y \right).$$

Considering D 6, A 16, and the *jurisdiction rule*, we can conclude the following,

D 7. $$S_y \mid\equiv \left(U_x \overset{K_{T_y}}{\longleftrightarrow} S_y \right),$$

thus proving G 2. We subsequently transform A 1 into

D 8. $$U_x \mid\equiv U_x \overset{K_{UK_x}}{\rightleftharpoons} KDC_x.$$

From I 7, D 8, and the *message-meaning rule*, we can get

D 9. $$U_x \mid\equiv KDC_x \mid\sim \left(U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{K_{T_x}}{\longleftrightarrow} S_y \right).$$

Based on A 10 and the *freshness conjuncatenation rule*, we can infer that

D 10. $$KDC_x \mid\equiv \#(T_{K_x}).$$

We then apply D 9, D 10, and the *nonce-verification rule* to derive

D 11. $$U_x \mid\equiv KDC_x \mid\equiv \left(U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{K_{T_x}}{\longleftrightarrow} S_y \right),$$

and thus leading to

D 12. $$U_x \mid\equiv KDC_x \mid\equiv \left(U_x \overset{K_{T_x}}{\longleftrightarrow} S_y \right).$$

Using the *jurisdiction rule*, we can apply D 12 and A 14 to derive

D 13. $$U_x \mid\equiv U_x \overset{K_{T_x}}{\longleftrightarrow} S_y.$$

We subsequently transform D 13 into

D 14. $$U_x \mid\equiv U_x \overset{K_{T_x}}{\rightleftharpoons} S_y.$$

Based on I 7 and D 14, we employ the *message-meaning rule* to infer

D 15. $$U_x \mid\equiv S_y \mid\sim \left(U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{T_{K_y}}{\rightleftharpoons} KDC_y, U_x \overset{K_{T_y}}{\longleftrightarrow} S_y \right).$$

We also have the following result from A 9 and the *freshness conjuncatenation rule*,

D 16. $$U_x \mid\equiv \#\left(U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{T_{K_y}}{\rightleftharpoons} KDC_y, U_x \overset{K_{T_y}}{\longleftrightarrow} S_y \right).$$

Therefore, we can have

D 17. $$U_x \mid\equiv S_y \mid\equiv \left(U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{T_{K_y}}{\rightleftharpoons} KDC_y, U_x \overset{K_{T_y}}{\longleftrightarrow} S_y \right),$$

according to D 15, D 16, and the *nonce-verification rule*. Following D 17, we get

D 18. $$U_x \mid\equiv S_y \mid\equiv \left(U_x \overset{T_{U_x}}{\rightleftharpoons} S_y \right),$$

D 19. $$U_x \mid\equiv S_y \mid\equiv \left(U_x \overset{K_{T_y}}{\longleftrightarrow} S_y \right).$$

This leads to the achievement of the goals G 3 and G 7. Due to the fact that S_y is authorized by KDC_y, we can plainly derive

D 20. $$U_x \mid\equiv KDC_y \mid\equiv \left(U_x \overset{K_{T_y}}{\longleftrightarrow} S_y \right).$$

Under the *jurisdiction rule*, we can employ A 13 and D 20 to derive

D 21. $$U_x \mid\equiv \left(U_x \overset{K_{T_y}}{\longleftrightarrow} S_y \right).$$

We subsequently transform D 21 into

D 22. $U_x \models \left(U_x \overset{K_{T_y}}{\rightleftharpoons} S_y\right)$.

According to the *message-meaning rule*, we can apply I 8 and D 22 to infer

D 23. $$S_y \models U_x \mid\sim \left(U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{T_{K_y}}{\rightleftharpoons} KDC_y, U_x \overset{K_{T_y}}{\longleftrightarrow} S_y\right).$$

From D 16, D 23, and the *nonce-verification rule*, we can obtain

D 24. $$S_y \models U_x \models \left(U_x \overset{T_{U_x}}{\rightleftharpoons} S_y, U_x \overset{T_{K_x}}{\rightleftharpoons} KDC_x, U_x \overset{T_{K_y}}{\rightleftharpoons} KDC_y, U_x \overset{K_{T_y}}{\longleftrightarrow} S_y\right),$$

and thus leading to

D 25. $S_y \models U_x \models \left(U_x \overset{T_{U_x}}{\rightleftharpoons} S_y\right)$,

D 26. $S_y \models U_x \models \left(U_x \overset{K_{T_y}}{\leftrightarrow} S_y\right)$,

which prove the goals of G 4 and G 8. According to A 15 and D 25, we can adopt the *jurisdiction rule* to derive

D 27. $S_y \models \left(U_x \overset{T_{U_x}}{\rightleftharpoons} S_y\right)$.

Hence, the goal of G 6 is satisfied. All beliefs are confirmed.

6 Conclusions

In this paper, we have proposed an efficient authentication scheme based on EAP protocol. We only apply the symmetric cryptosystem and one-way hash function to confirm the system security, which can diminish the key management load and calculation cost. Without the use of public key cryptosystem, the new scheme can be integrated with the 802.1X. In particular, the method is very suitable for mobile device circumstance considering its feature of light-weight. In addition, we have adopted BAN logic to prove the correctness of mutual authentication to highlight its security.

References

1. Aboba, B., & Simon, D. (1999). *PPP EAP TLS authentication protocol*. RFC 2716, IETF, October 1999. http://www.ietf.org/rfc/rfc2716.txt.
2. Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., & Levkowetz, H. (2004). *Extensible authentication protocol (EAP)*. RFC 3748, IETF, June 2004. http://www.ietf.org/rfc/rfc3748.txt.
3. Aboba, B., Simon, D., & Eronen, P. (2008). *Extensible authentication protocol (EAP) key management framework*. RFC 5247, IETF, August 2008. http://www.ietf.org/rfc/rfc5247.txt.
4. Burrows, M., Abadi, M., & Needham, R. (1990). A logic of authentication. *ACM Transactions on Computer Systems, 8*(1), 18–36.
5. Blunk, L., & Vollbrecht, J. (1998). *PPP extensible authentication protocol (EAP)*. RFC 2284, IETF, March 1998. http://www.ietf.org/rfc/rfc2284.txt.
6. Dantu, R., Clothier, G., & Atri, A. (2007). EAP methods for wireless networks. *Computer Standards & Interfaces, 29*(3), 289–301.
7. Funk, P., & Blake-Wilson, S. (2008). *Extensible authentication protocol tunneled transport layer security authenticated protocol version 0 (EAP-TTLSv0)*. RFC 5281, IETF, August 2008. http://www.ietf.org/rfc/rfc5281.txt.
8. Huang, Y. L., Lu, P. H., Tygar, J. D., & Joseph, A. D. (2009). OSNP: Secure wireless authentication protocol using one-time key. *Computers & Security, 28*(8), 803–815.
9. IEEE. (2001). *IEEE 802.1X: Medium access control (MAC) security enhancements*. June 2001.
10. IEEE. (2003). *IEEE 802.11i: Medium access control (MAC) security enhancements*. May 2003.
11. Liang, W., & Wang, W. Y. (2005). On performance analysis of challenge/response based authentication in wireless networks. *Computer Networks, 48*(2), 267–288.
12. Lee, J. S., Lin, P. Y., & Chang, C. C. (2009). Lightweight secure roaming mechanism between GPRS/UMTS and wireless LANs. *Wireless Personal Communications, 53*(4), 569–580.
13. Macnally, C. (2001). *Cisco LEAP protocol description*. September 2001. http://lists.cistron.nl/pipermail/cistron-radius/2001-September/002042.html.
14. Ohba, Y., Das, S., & Dutta, A. (2007) Kerberized Handover keying: A media independent handover key management architecture. In *Proceedings of the 2nd ACM/IEEE international workshop on mobility in the evolving internet architecture*, Kyoto, Japan, No. 9, pp. 1–7, August 2007.
15. Schneier, B. (1996). *Applied cryptography* (2nd ed.). New York: Wiley.
16. Syverson, P. F., & Cervesato, I. (2001). The logic of authentication protocols. *Lecture Notes in Computer Science, 2171*, 63–136.
17. Stanley, D., Walker, J., & Aboba, B. (2005). *Extensible authentication protocol (EAP) method requirements for wireless LANs*. RFC 4017, IETF, March 2005. http://www.ietf.org/rfc/rfc4017.txt.
18. Simon, D., Aboba, B., & Hurst, R. (2008). *The EAP-TLS authentication protocol*. RFC 5216, IETF, March 2008. http://www.ietf.org/rfc/rfc5216.txt.
19. Tseng, Y. M. (2006). GPRS/UMTS—aided authentication protocol for wireless LANs. *IEE Proceedings-Communications, 153*(6), 810–817.
20. Tseng, Y. M. (2009). USIM-based EAP-TLS authentication protocol for wireless local area networks. *Computer Standards & Interfaces, 31*(1), 128–136.
21. Tsai, H. C., Chang, C. C., & Chang, K. J. (2009). Roaming across wireless local area networks using SIM-based authentication protocol. *Computer Standards & Interfaces, 31*, 381–389.
22. Wu, T. Y., & Tseng, Y. M. (2010). An efficient user authentication and key exchange protocol for mobile client–server environment. *Computer Networks, 54*(9), 1520–1530.
23. Yang, C. C., Tang, Y. L., Wang, R. C., & Yang, H. W. (2005). A secure and efficient authentication protocol for anonymous channel in wireless communications. *Applied Mathematics and Computation, 169*(2), 1431–1439.

24. Yao, L., Wang, L., Kong, X. W., Wu, G. W., & Xia, F. (2010). An inter-domain authentication scheme for pervasive computing environment. *Computers and Mathematics with Applications, 60*(2), 234–244.
25. Chuang, M. C., & Lee, J. F. (2011). A lightweight mutual authentication mechanism for network mobility in IEEE 802.16e wireless networks. *Computer Networks, 55*(16), 3796–3809.
26. Lee, T. F., & Hwang, T. (2011). Provably secure and efficient authentication techniques for the global mobility network. *Journal of Systems and Software, 84*(10), 1717–1725.
27. Li, C. T., & Lee, C. C. (2012). A novel user authentication and privacy preserving scheme with smart cards for wireless communications. *Mathematical and Computer Modelling, 55*, 35–44.
28. The OpenSSL Project. http://www.openssl.org.

Author Biographies

Jung-San Lee received the B.S. degree in computer science and information engineering in 2002 and his Ph.D in computer science and information engineering in 2008, both from National Chung Cheng University, Chiayi, Taiwan. Since 2012, he has worked as an associate professor in the Department of Information Engineering and Computer Science at Feng Chia University, Taichung, Taiwan. His current research interests include electronic commerce, information security, cryptography, and mobile communications.

Wan-Ting Tseng received the M.S. degree in computer science and information engineering in 2011 from National Chung Cheng University, Chiayi, Taiwan. Her current research interests include information security and wireless communications.