Received January 16, 2020, accepted February 17, 2020, date of publication February 27, 2020, date of current version March 10, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2976431*

# A Privacy-Preserved E2E Authenticated Key Exchange Protocol for Multi-Server Architecture in Edge Computing Networks

**CHIEN-LUNG HSU**[1,2,3,4,5], **TUAN-VINH LE**[2], **CHUNG-FU LU**[6], **TZU-WEI LIN**[2], **AND TZU-HSIEN CHUANG**[1]

[1]Department of Information Management, Chang Gung University, Taoyuan 33302, Taiwan
[2]Graduate Institute of Business and Management, Chang Gung University, Taoyuan 33302, Taiwan
[3]Healthy Aging Research Center, Chang Gung University, Taoyuan 33302, Taiwan
[4]Department of Visual Communication Design, Ming Chi University of Technology, Taoyuan 24301, Taiwan
[5]Department of Nursing, Taoyuan Chang Gung Memorial Hospital, Taoyuan 33044, Taiwan
[6]Department of Information Management, Chihlee University of Technology, New Taipei City 24243, Taiwan

Corresponding author: Chung-Fu Lu (peter61@mail.chihlee.edu.tw)

This work was supported in part by the Chang Gung Memorial Hospital under Grant CMRPG5D0183 and Grant CMRPD3D0063, in part by the Ministry of Science and Technology in Taiwan under Grant MOST-105-2923-E-182-001-MY3, Grant MOST-107-2221-E-182-006, and Grant MOST-108-2221-E-182-011, in part by the Healthy Aging Research Center, Chang Gung University from the Featured Areas Research Center Program within the Framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan, under Grant EMRPD1I0481, Grant EMRPD1H0421, and Grant EMRPD1H0551.

**ABSTRACT** Edge computing has played an important role in enabling 5G technology which supports a great number of connected narrow-band IoT devices. In an edge computing architecture enabled with global mobile network, edge or IoT devices are wirelessly connected to the edge of the network. Data acquisition and processing will be handled at or close to the edge of the network in a distributed way. Since edge computing is a heterogeneous distributed interactive system with multiple domains and entities, it might suffer from potential attacks and threats. To provide a trusted edge computing, there must have a robust scheme that allows all participants to mutually authenticate in a secure and privacy-preserved way. With the rapid development of IoT technologies, mobile networks and edge computing architecture, single server has been unable to meet the needs of users. In this paper, we propose a privacy-preserved end-to-end password-based authenticated key exchange protocol for multi-server architecture in edge computing networks. Our protocol allows an end user to use an easy-to-remember password to login to the server, then through foreign agent compute a shared key with another end user for specific use of services. The proposed protocol provides strong user anonymity during communication process. Besides, the proposed protocol is proved to be secure using BAN logic and AVISPA tool. Furthermore, performance analysis shows that the proposed protocol gains stronger security and better computational efficiency. Providing lightweight computation with short key size of ECC, our work is a solution to lower latency and improve efficiency in edge computing networks.

**INDEX TERMS** Edge computing, IoT, end-to-end, privacy protection, password-based, key exchange.

## I. INTRODUCTION

Development of ubiquitous computing technologies and wireless sensing devices has driven various innovative services and applications of the Internet of Things (IoT), such as smart home, smart healthcare, smart city, intelligent transportation, and etc. IoT is a global and heterogeneous infrastructure comprising a number of functional blocks based on existing and evolving interoperable wireless information and communication technologies [1] such as 2G / 3G / 4G, WiFi, Bluetooth, etc. Functional blocks include internet-enabled sensing devices, communication, services, management, security, and applications [2]. It enables advanced intelligent context-aware services by interconnecting physical things, virtual things or hybrid things. Recently, Internet of Everything (IoE) focused on the intelligent connection of people, processes, data, and everything has been introduced. The proliferation of the IoE and 5G network architecture [3]

The associate editor coordinating the review of this manuscript and approving it for publication was Marco Martalo.

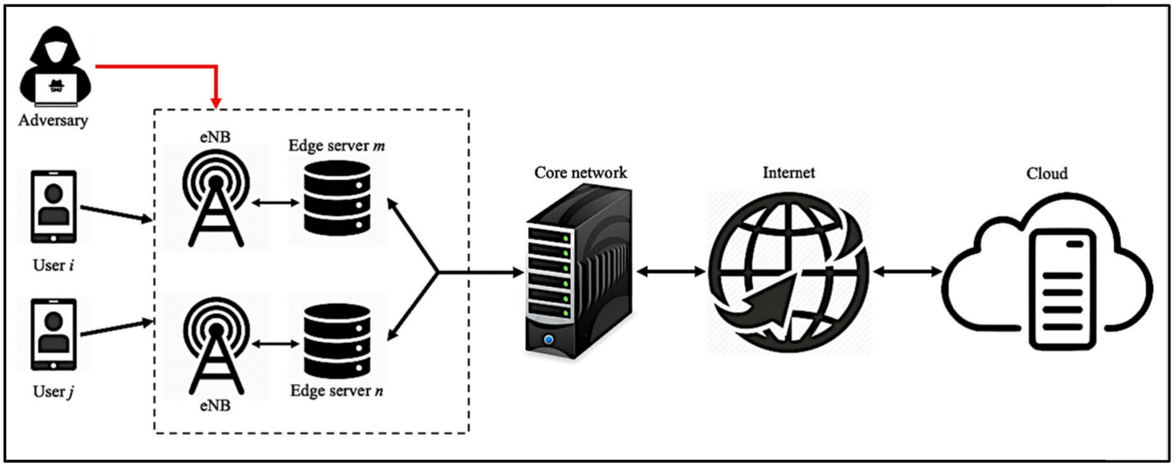


**FIGURE 1.** Attack scenario in an edge computing network.

enables powerful collaborative computation, data processing, and rich service interface for the devices. However, traditional centralized cloud computing model for IoE will inefficiently support IoE-based application services due to the following problems: (i) Multi-sources data processing requirements of massive data at the edge of network might not be met. (ii) The communicational bandwidth and speed might be a bottleneck due to large scale of user access. (iii) It is a big challenge to deal with user privacy and users' sensitive data in edge devices. Therefore, it is desired to combine existing cloud computing and edge computing to efficiently deal with the massive data processing problems at the edge of the network [2].

In an edge computing architecture, data acquisition and processing will be handled at or close to the edge of the network in a distributed way. It can offload the computation and communication burden and gain better quality of service. In the other word, edge computing enables storing and processing data at the edge of the network [4]. Thereby, edge computing addresses heavyweight computation problem from cloud computing [5]. Thus, edge computing has played an important role in enabling 5G technology, where narrow-band (NB) IoT devices are the essential entity. However, since edge computing is a heterogeneous distributed interactive system with multiple domains and entities, it might suffer from potential security issues and challenges in processing massive data. These issues and challenges include data security, secure computation, secure transmission, entity authentication, access control, privacy protection [2]. To provide a trusted edge computing, it should allow all participants to mutually authenticate for withstanding potential threats. Figure 1 presents communication in an edge computing network. Users may ask request to use services from providers (home servers or foreign agents). Besides, end users can communicate with each other to compute conversation key for specific purpose. Since this communication is carried out via a public channel, it is threatened to various attacks, such as man-in-the-middle attack [6], replay attack [7], impersonation attack [8], stolen verifier attack [9] and so on. An adversary can access message and steal desired information. Besides, identity anonymity [10] is very important to user. Therefore, a robust authentication scheme securing this communication is essential. Recently, a lot of works have been conducted to address security and privacy for sensitive information distributed using IoT devices in mobility networks or edge computing networks [11]–[15].

An edge computing architecture enabled with global mobility network provides effective global roaming services for personal communicating users and IoT devices. Through the universal roaming technology, legitimate mobile users can enjoy ubiquitous services [16] and manage IoT devices. A global mobility network includes three communicating parties: mobile user, home server and foreign agent. Mobile user in global mobility networks access service provider using IoT devices. User can directly use service from home server. Besides, he/she can communicate with foreign agent to obtain the service through home server [17]. With the rapid development of mobility network technology, people can use various services through mobile devices anytime and anywhere with edge computing. In order to address user security and privacy, lots of authentication and key exchange protocols used for global mobility networks have been introduced [18]–[21]. For instance, replay attack was prevented in [22], [23].

Furthermore, Sood [24] proposed a smart identity authentication protocol based on a dynamic identity card, which is obtained from improvement of Bellovin and Merritt [25]'s protocol. Sood used congruent multiplication and exponent to calculate user identity and password, then stores these in a verification table. However, Sood's protocol is not free from stolen verifier attack when attacker steals the verification table. Therefore, some scholars have proposed password-based authentication mechanism without verification table to withstand this attack [26], [27].

Recently, Gope and Hwang [16] proposed a strong anonymity mutual authentication and key agreement scheme for global mobile networks. Mobile communication architecture introduced in their work provided user a cross-domain server mutual authentication method. However, server in Gope and Hwang' protocol needs to maintain a verification table at registration center, which causes certain threats. Their work did not introduce a strong two-factor authentication. Besides, Gope and Hwang's scheme cannot achieve the goal of end-to-end communication.

With the rapid development of IoT technologies, global mobile networks and edge computing networks, single server has been unable to meet the needs of users. The number of servers has increased remarkably to provide more services for the end user [28]. The conventional schemes allow user to access service only with a single server. More servers will lead to more identities and passwords that user must remember, which causes considerable inconvenience. It is not secure that user uses the same set of identities and passwords to register with different servers. Therefore, many researchers have proposed identity authentication mechanism suitable for a multi-server environment so that user can obtain services from multiple servers using a single password. A multi-server architecture in the edge computing network allows users to access service without complicated registration and authentication. For instance, Li *et al.* [29] proposed a secure dynamic identity based authentication protocol with smart card for multi-server architecture.

In this paper, we propose a privacy-preserved end-to-end authenticated key exchange protocol for multi-server architecture in distributed edge computing networks. The proposed protocol allows a mobile user to use an easy-to-remember password to login and authenticate different servers in the network. Edge computing network enables 5G technology architecture that supports a massive number of connected NB-IoT devices. The users of these devices may want to directly connect to each other for specific purposes such as sharing services, establishing common subscriptions, etc. To this end, our proposed scheme allows end users to communicate with each other and compute a shared key through the help of home server and foreign agent. User privacy is protected during communication process. Multi-server architecture introduced in our work deals with the overhead. Besides, Elliptic Curve Cryptography (ECC) with small key size is employed in our scheme. Hence, the proposed scheme favors end-to-end communication and is well suited for 5G enabled edge computing networks. Our proposed scheme is favored by the help of smart card, which can provide personal identification, authentication, data storage, and application processing [30].

The rest of this paper is organized as follows. Section II, we briefly review Gope and Hwang's scheme. Section III, we propose a privacy-preserved end-to-end authenticated key exchange protocol for multi-server architecture in edge computing networks. Section IV and Section V, we respectively present formal and informal security analysis of the proposed protocol. Section VI, we compare performance of the proposed protocol with its related works. Section VII, an implementation of the proposed protocol is described. Finally, the conclusions and future research directions are given in Section VIII.

## II. REVIEW OF GOPE AND HWANG'S SCHEME

In this section, we briefly describe Gope and Hwang's scheme, which consists of three phases: registration phase, mutual authentication and key agreement phase, and password update phase. After that, we point out some weaknesses of their protocol.

### A. REGISTRATION PHASE

*Step 1 — Mobile user (MU) sends registration information to home agent (HA). They perform the following sub-steps.*

Step 1-1: MU submits his/her identity $ID_M$ to HA via a secure channel.

Step 1-2: HA generates a random number $\mathrm{n_h}$ and then computes $K_{uh} = h(ID_M||n_h) \oplus ID_h$.

Step 1-3: HA generates a set of unlinkable pseudo-$IDs$ $PID = \{pid_1, pid_2, \ldots\}$, where for each $pid_j \in PID$, $pid_j = h(ID_M||r_i K_{uh})$, $r_i$ a random number.

Step 1-4: HA generates a unique track sequence number $Tr_{seq}$, which is basically a sequence number of 64-bit.

Step 1-5: HA stores $K_{uh}$ and $ID_M$ in its database.

Step 1-6: HA stores $K_{uh}$, PID, $Tr_{seq}$, $h(\cdot)$ in the smart card and sends smart card to MU.

*Step 2 — The shared key $K_{uh}$ between mobile user MU and home agent HA is stored in smart card.*

Step 2-1: MU chooses a password $PSW_M$ and submits it to the smart card.

Step 2-2: Smart card computes $K_{uh}^* = K_{uh} \oplus h(ID_M||PSW_M)$, $PID^* = PID \oplus h(ID_M||PSW_M)$.

Step 2-3: MU replaces $K_{uh}$ and $K_{uh}^*$ with $PID$ and $PID^*$ respectively. Then smart card contains $\{K_{uh}^*, PID^*, Tr_{seq}, h(\cdot)\}$.

### B. MUTUAL AUTHENTICATION AND KEY AGREEMENT (MAKA) PHASE

*Step 1 — Smart card computes the shared key $K_{uh}$ of mobile user MU and home agent HA with the legitimate $ID_M$ and $PSW_M$ , and sends an authentication request to foreign agent FA.*

Step 1-1: MU inserts his/her smart card into the reader and enters his/her identity $ID_M$ and password $PSW_M$.

Step 1-2: Smart card generates two random numbers $N_m, N_m'$ and computes $P = N_m \oplus N_m'$.

Step 1-3: Smart card computes $K_{uh} = K_{uh}^* \oplus h(ID_M||PSW_M)$, $AID_M = h(ID_M||K_{uh}||N_m||Tr_{seq})$, where $Tr_{seq}$ denotes the most recent track sequence number, received from the home agent HA. In case of loss of synchronization, the user needs to choose one of the unused $pid_j^*$ then submits his/her identity $ID_M$ and password $PSW_M$ and computes

$pid_j = pid_j^* \oplus h(ID_M||PSW_M)$. Subsequently, assigns the $pid_j$ as $AID_M$, i.e. $AID_M = pid_j$. In that case, user needs not to include the track sequence number $Tr_{seq}$ in $M_{B_1}$.

Step 1-4: MU forms $M_{B_1} = \{AID_M, \{N'_m||p\}E_{K_{uh}}, Tr_{seq}$ $(if\ req.), ID_h\}$, and sends request message $M_{B_1}$ to FA.

*Step 2 — FA sends MU's authentication request information to HA. Foreign agent FA performs the following sub-steps.*

Step 2-1: FA generates two random numbers $N_f$, $N'_f$ and computes $Q = N_f \oplus N'_f$.

Step 2-2: FA computes $V_1 = h\{M_{B_1}||K_{fh}||N_f) \oplus Q$.

Step 2-3: FA forms a message $M_{B_2} = \{AID_M, \{N'_m||p\}E_{K_{uh}}, Tr_{seq}, \{N'_f||E_{K_{uh}}, V_1\}$, and send $M_{B_2}$ to HA.

*Step 3 — After receiving the $M_{B_2}$, HA verifies the legitimacy of the mobile user MU and the foreign agent FA. HA performs the following sub-steps.*

Step 3-1: HA checks whether the track sequence number $Tr_{seq}$ is valid.

Step 3-2: HA decrypts $\{N'_m||P\}E_{K_{uh}}$ and $\{N'_f|Q\}E_{K_{fh}}$ with shared key $K_{uh}$ and $K_{fh}$.

Step 3-3: HA computes and verifies the parameters $V_1$, $AID_M$.

Step 3-4: HA computes $x = \{N_m||N_f\}E_{K_{fh}}$, $Tr = h(K_{uh}||ID_M||N_m) \oplus Tr_{seq_{new}}$, $V_2 = h(x||K_{fh}||N_f)$, $y = h(N_m||K_{uh}) \oplus N_f$, $V_3 = h(y||N'_m||K_{uh}||Tr)$.

Step 3-5: HA sends $M_{B_3} = \{x, Tr, y, V_2, V_3\}$ to FA.

*Step 4 — After receiving $M_{B_3}$ transmitted by HA, FA authenticates HA and establishes a conversation key with MU. FA performs the following sub-steps.*

Step 4-1: FA decrypts $x$ using $K_{fh}$, checks the integrity of $x$, and verifies $N_f$ by computing and comparing $V_2^*$ with $V_2$.

Step 4-2: FA computes the session key $SK = N_m \oplus N_f$.

Step 4-3: FA forms a response message $M_{B_4} = \{y, Tr, V_3\}$ and sends $M_{B_4}$ to MU.

*Step 5 — After receiving $M_{B_4}$ transmitted by FA, MU authenticates HA and FA, then establishes a conversation key with FA. MU performs the following sub-steps.*

Step 5-1: Using $y$ and $Tr$ from $M_{B_4}$, MU computes $V_3^*$. Then it verifies if $V_3^*$ and $V_3$ are equal.

Step 5-2: Using $y$ and $Tr$ from $M_{B_4}$, MU computes $N_f = h(N_m||K_{uh}) \oplus y$, $Tr_{seq_{new}} = h\{K_{uh}||ID_M||N_m)$ $\oplus Tr$, and $SK = N_m \oplus N_f$.

Step 5-3: MU updates $Tr_{seq} = Tr_{seq_{new}}$.

### C. PASSWORD UPDATE PHASE

*Step 1 — MU needs to insert his/her identity $ID_M$ and current password $PSW_M$ to smart card, then computes $K_{uh} = K_{uh}^* \oplus h(ID_M||PSW_M)$, $PID = PID^* \oplus h(ID_M||PSW_M)$. After verifying user's legitimacy, MU enters the new password $PSW_M^*$.*

*Step 2 — Using the new password, smart card computes $K_{uh}^{**} = K_{uh}^* \oplus h(ID_M||PSW_M^*)$, $PID^{**} = PID^* \oplus h(ID_M||PSW_M^*)$.*

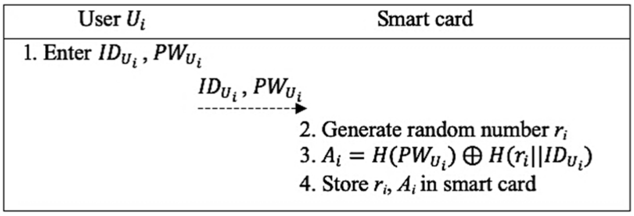


**FIGURE 2.** Smart card registration phase.

*Step 3 — The device will replace $K_{uh}^*$ with $K_{uh}^{**}$, $PID^*$ with $PID^{**}$, then store them for further communication.*

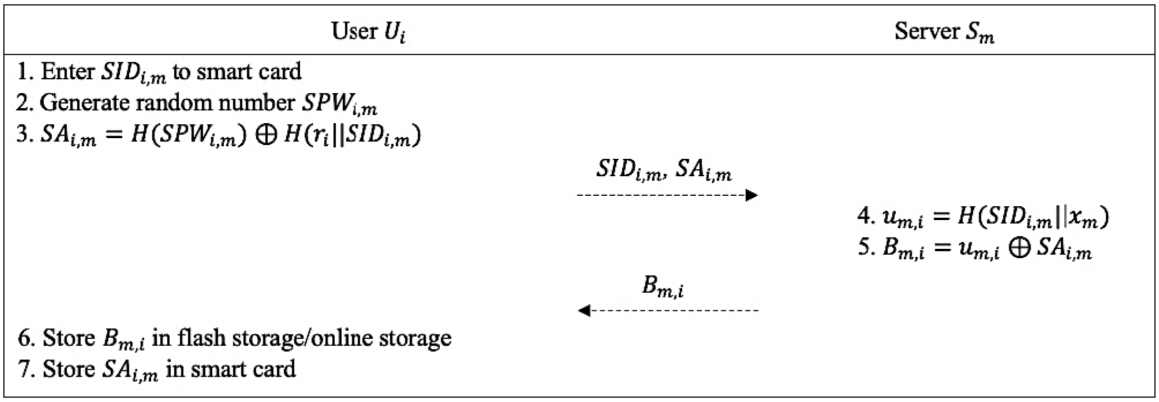
### D. WEAKNESSES OF GOPE AND HWANG'S SCHEME

Gope and Hwang [16] claimed that their protocol can resist various known attacks. However, we found that their protocol has certain weaknesses as follows:

- Unsecure against man-in-the-middle attack: This attack happens when an attacker attempts to intercept the message transmitted between the sender and the receiver who believe that they are directly communicating with each other. He/she tries to impersonate legitimate parties or obtain secret information. At the registration phase of Gope & Hwang's scheme, the home agent (HA) personalizes a smart card with $\{K_{uh}, PID, Tr_{seq}, h(\cdot)\}$ and issues it to MU and then stores a copy of $K_{uh}$ in its database for further communication. An adversary in registration center may use this parameter to impersonate the user and obtain his/her service from foreign agent.
- Unsecure against stolen-verifier attack: Similarly, Gope & Hwang's scheme needs a verification table at registration center. This table may be leaked out and the adversary can use it to impersonate the legitimate user.
- Lacks strong two-factor authentication: This mechanism includes password and smart card in authentication process so as to enhance security. In Gope & Hwang's scheme, MU inserts his/her smart card into the reader and enters his/her identity $ID_M$ and password $PSW_M$. However, smart card registration was not available. The smart card then was not used to verify the user by confirming the input information. Therefore, their scheme doesn't achieve strong two-factor authentication.
- Lacks user end-to-end communication: in Gope & Hwang's scheme, user is only able to communicate with the foreign agent to obtain its service. An end-to-end communication between user and user was not introduced in their work. In many scenarios, users want to communicate with each other to compute the shared key for further purposes. Thus, a robust authentication scheme that secures this communication is essential.

## III. THE PROPOSED PROTOCOL

Our proposed protocol includes four roles/actors: user $U_i$, user $U_j$, remote server $S_m$ and remote server $S_n$. The proposed protocol consists of six phases: system initialization phase, smart card registration phase, server registration phase, login phase, mutual authentication & key exchange phase, and



1. Enter $SID_{i,m}$ to smart card
2. Generate random number $SPW_{i,m}$
3. $SA_{i,m} = H(SPW_{i,m}) \oplus H(r_i||SID_{i,m})$
4. $u_{m,i} = H(SID_{i,m}||x_m)$
5. $B_{m,i} = u_{m,i} \oplus SA_{i,m}$
6. Store $B_{m,i}$ in flash storage/online storage
7. Store $SA_{i,m}$ in smart card

**FIGURE 3.** Server registration phase.

password update phase. During the protocol, all of the parties including $U_i$, $FA_p$, $S_m$, $U_j$, $FA_q$, $S_n$ participate in the communication that lets the user $U_i$ and $U_j$ compute a conversation key. For simplicity, only communication among $U_i$, $FA_p$, $S_m$ is described. Table 1 describes notations and cryptographic functions used in this paper.

**TABLE 1.** Notations and cryptographic functions.

| Symbol | Description |
|---|---|
| $ID_U$ | *ID* of user *U* used at smart card registration |
| $SID$ | *ID* of user *U* used at server registration |
| $ID_S$ | *ID* of remote server *S* |
| $PW_U$ | Password of user *U* generated at smart card registration |
| $SPW$ | Password of user *U* generated at server registration |
| $r$ | Random number that smart card generate when user *U* login to remote server *S* |
| $\oplus$ | Exclusive Or operation |
| $H(\,)$ | One-way hash function |
| $x$, $k$ | Secret key of the server *S* |
| $V$ | Public key of the server *S* |
| $a$ | Randomly selected number of user *U* 's smart card |
| $b$ | Randomly selected number of the server *S* |
| $N_f$ | Randomly selected number of foreign agent *FA* |
| $R$ | encryption/decryption key of user *U* |
| $f$ | Shared key between foreign agent and remote server |
| $E_R(\,)$ / $D_R(\,)$ | Symmetric encryption / decryption functions with key *R* |
| $T$ | Timestamp |
| $Cert$ | *CA* Certificate of the server *S* |
| $Sig$ | *CA* Signature of the server *S* |

### A. SYSTEM INITIALIZATION PHASE

In system intialization phase, based on elliptic curve cryptography proposed by the National Institute of Standards and Technology (NIST) [31], the system generates a curve $Ep\,(a, b) : y^2 = x^3 + ax + b(mod\ p)$ with a point $G_{(x1,y1)}$. It then computes public key for each server using the secret key $k$, $V = kG$. Besides, $f$ is the symmetric key the home server and the foreign agent. Home server registers to certificate authority *CA* and obtains their own certificate, signature, public key and private key.

### B. SMART CARD REGISTRATION PHASE

The user $U_i$ sends registration information to the smart card, the user $U_i$ and the smart card performs following steps (shown in Figure 2).

*Step 1 — $U_i$ enters $ID_{U_i}$, $PW_{U_i}$ to smart card.*

*Step 2 — Smart card generates $r_i$, then computes $A_i = H(PW_{U_i}) \oplus H(r_i||ID_{U_i})$.*

*Step 3 — Smart card stores $r_i$ and $A_i$.*

### C. SERVER REGISTRATION PHASE

The user first logins to the smart card then performs server registration. As shown in Figure 3, the user $U_i$ and the server $S_m$ perform the following steps.

*Step 1 — $U_i$ transmits the registration information to $S_m$ through smart card. $U_i$ and $S_m$ perform the following sub-steps.*

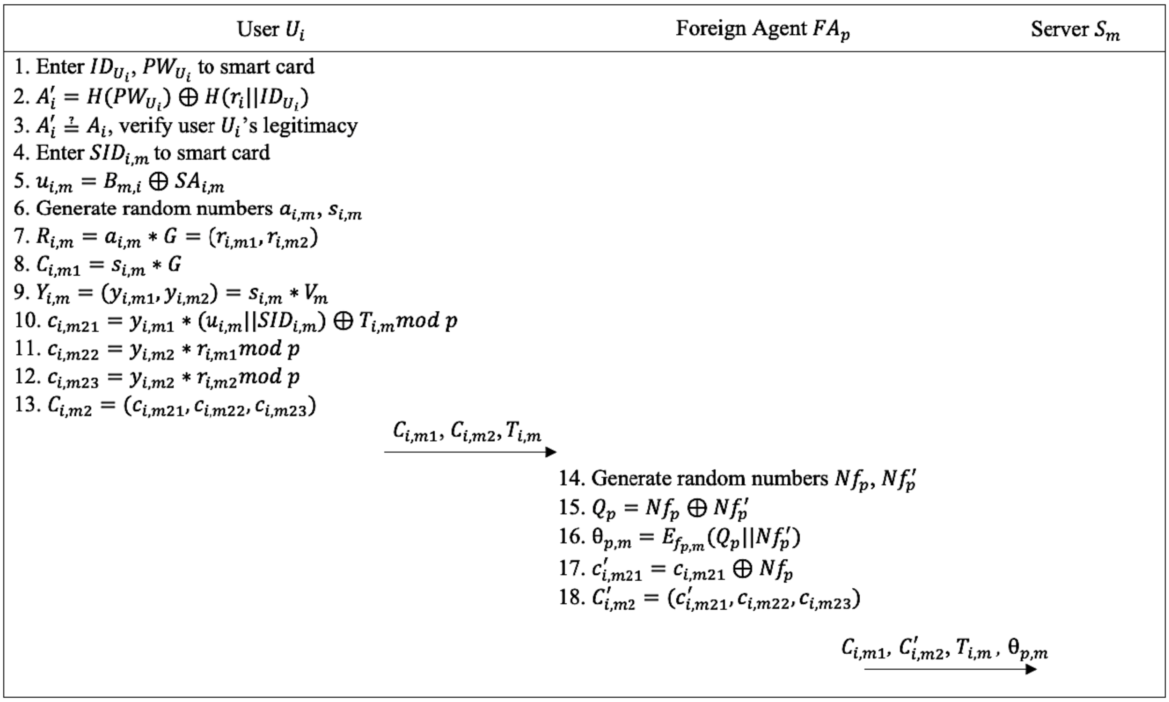
Step 1-1: $U_i$ enters $SID_{i,m}$ to smart card.
Step 1-2: Smart card generates random number $SPW_{i,m}$, then computes $SA_{i,m} = H(SPW_{i,m}) \oplus H(r_i||SID_{i,m})$.
Step 1-3: $U_i$ transmits $SID_{i,m}$ and $SA_{i,m}$ to server $S_m$.

*Step 2 — $S_m$ computes a shared value with $U_i$.*

Step 2-1: $S_m$ computes $u_{m,i} = H(SID_{i,m}||x_m)$ and $B_{m,i} = u_{m,i} \oplus SA_{i,m}$.
Step 2-2: $S_m$ transmits $B_{m,i}$ to the user $U_i$.
Step 2-3: $U_i$ stores $B_{m,i}$ and $SA_{i,m}$ in flash drive and smart card respectively.

### D. LOGIN PHASE

In login phase, the user $U_i$ first logins to smart card for verification. As shown in Figure 4, the user $U_i$ logs in to the server $S_m$, then the user $U_i$, the smart card, the foreign agent $FA_p$, and the server $S_m$ jointly perform the following steps to complete the procedure in which the user $U_i$ can



1. Enter $ID_{U_i}$, $PW_{U_i}$ to smart card
2. $A_i' = H(PW_{U_i}) \oplus H(r_i||ID_{U_i})$
3. $A_i' \overset{?}{=} A_i$, verify user $U_i$'s legitimacy
4. Enter $SID_{i,m}$ to smart card
5. $u_{i,m} = B_{m,i} \oplus SA_{i,m}$
6. Generate random numbers $a_{i,m}$, $s_{i,m}$
7. $R_{i,m} = a_{i,m} * G = (r_{i,m1}, r_{i,m2})$
8. $C_{i,m1} = s_{i,m} * G$
9. $Y_{i,m} = (y_{i,m1}, y_{i,m2}) = s_{i,m} * V_m$
10. $c_{i,m21} = y_{i,m1} * (u_{i,m}||SID_{i,m}) \oplus T_{i,m} mod\ p$
11. $c_{i,m22} = y_{i,m2} * r_{i,m1} mod\ p$
12. $c_{i,m23} = y_{i,m2} * r_{i,m2} mod\ p$
13. $C_{i,m2} = (c_{i,m21}, c_{i,m22}, c_{i,m23})$

$\xrightarrow{C_{i,m1},\ C_{i,m2},\ T_{i,m}}$

14. Generate random numbers $Nf_p$, $Nf_p'$
15. $Q_p = Nf_p \oplus Nf_p'$
16. $\theta_{p,m} = E_{f_{p,m}}(Q_p||Nf_p')$
17. $c_{i,m21}' = c_{i,m21} \oplus Nf_p$
18. $C_{i,m2}' = (c_{i,m21}', c_{i,m22}, c_{i,m23})$

$\xrightarrow{C_{i,m1},\ C_{i,m2}',\ T_{i,m},\ \theta_{p,m}}$

**FIGURE 4.** Login phase.

login to the server $S_m$ and compute conversation key with the user $U_j$.

*Step 1 — $U_i$ enters $ID_{U_i}$, $PW_{U_i}$ to smart card.*

*Step 2 — Smart card computes $A_i' = H(PW_{U_i}) \oplus H(r_i||ID_{U_i})$.*

*Step 3 — Smart card compares $A_i'$ and its $A_i$, then verifies the legitimacy of user $U_i$.*

*Step 4 — User $U_i$ inserts his/her smart card, then enters $SID_{i,m}$. Using $B_{m,i}$ from flash drive, smart card calculates shared secret number $u_{i,m} = B_{m,i} \oplus SA_{i,m}$.*

*Step 5 — Smart card generates two random numbers $a_{i,m}$, $s_{i,m}$.*

*Step 6 — Smart card computes $R_{i,m} = a_{i,m} * G = (r_{i,m1}, r_{i,m2})$, $C_{i,m1} = s_{i,m} * G$, $Y_{i,m} = (y_{i,m1}, y_{i,m2}) = s_{i,m} * V_m$, $c_{i,m21} = y_{i,m1} * (u_{i,m}||SID_{i,m}) \oplus T_{i,m} mod\ p$, $c_{i,m22} = y_{i,m2} * r_{i,m1} mod\ p$, $c_{i,m23} = y_{i,m2} * r_{i,m2} mod\ p$, $C_{i,m2} = (c_{i,m21}, c_{i,m22}, c_{i,m23})$.*

*Step 7 — User $U_i$ transmits $C_{i,m1}$, $C_{i,m2}$, $T_{i,m}$ to foreign agent $FA_p$.*

*Step 8 — Foreign agent $FA_p$ generates two random numbers $Nf_p$, $Nf_p'$, computes $Q_p = Nf_p \oplus Nf_p'$, $\theta_{p,m} = E_{f_{p,m}}(Q_p||Nf_p')$, $c_{i,m21}' = c_{i,m21} \oplus Nf_p$, $C_{i,m2}' = (c_{i,m21}', c_{i,m22}, c_{i,m23})$, and transmits { $C_{i,m1}$, $C_{i,m2}'$, $T_{i,m}$, $\theta_{p,m}$ } to server $S_m$.*

### *E. MUTUAL AUTHENTICATION AND KEY EXCHANGE PHASE*

#### 1) MUTUAL AUTHENTICATION AND EXCHANGE PHASE BETWEEN SERVERS

As shown in Figure 5, mutual authentication and key exchange process between two servers is described as follows.

*Step 1 — Server $S_m$ and $S_n$ respectively authenticate the legitimacy of user $U_i$ and $U_j$. The following sub-steps are performed by server $S_m$.*

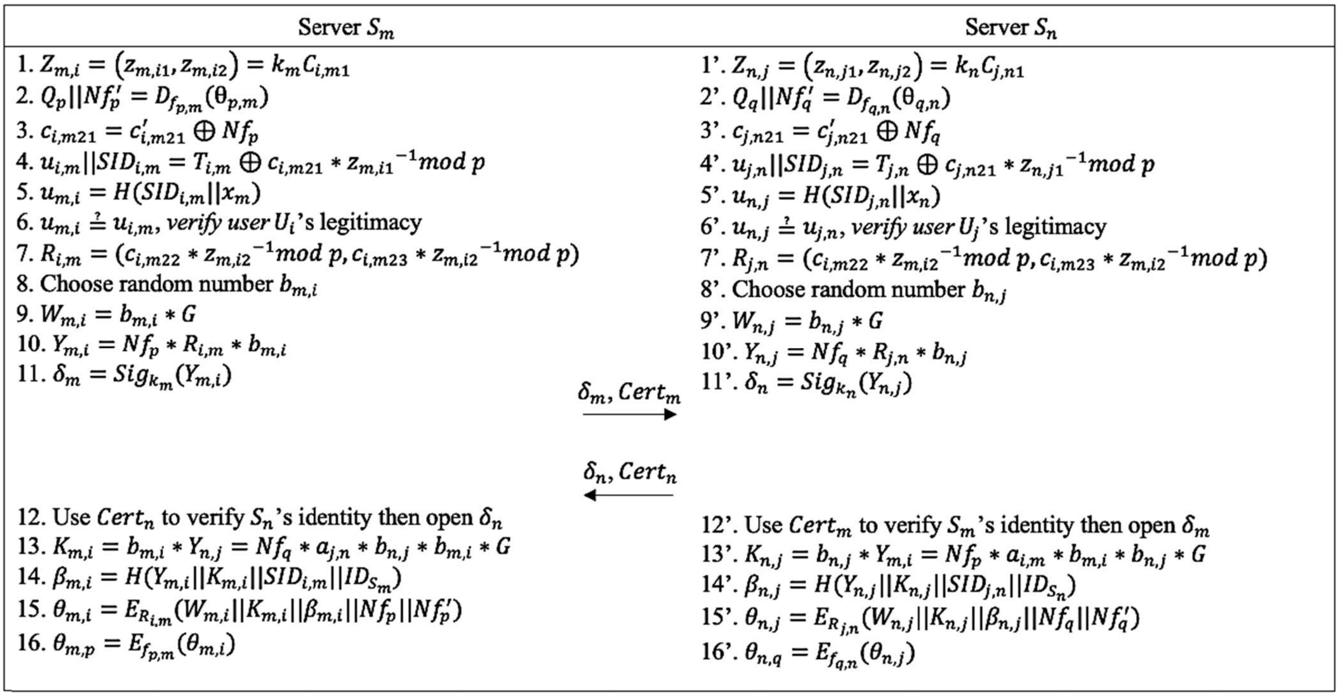
Step 1-1: $S_m$ computes $Z_{m,i} = (z_{m,i1}, z_{m,i2}) = k_m C_{i,m1}$.
Step 1-2: $S_m$ decrypts $\theta_{p,m}$ to get $Q_p$, $Nf_p'$, and computes $c_{i,m21} = c_{i,m21}' \oplus Nf_p$.
Step 1-3: $S_m$ computes $u_{i,m}||SID_{i,m} = T_{i,m} \oplus c_{i,m21} * z_{m,i1}^{-1} mod\ p$.
Step 1-4: $S_m$ uses above $SID_{i,m}$ and its secret number $x_m$ to compute $u_{m,i} = H(SID_{i,m}||x_m)$.
Step 1-5: $S_m$ confirms $u_{m,i} \overset{?}{=} u_{i,m}$ to verify user $U_i$'s legitimacy. If there is a match, $U_i$ is confirmed to be a legitimate user.
Step 1-6: $S_m$ employs the Elliptic Curve Cryptography to obtain $R_{i,m} = (c_{i,m22} * z_{m,i2}^{-1} mod\ p, c_{i,m23} * z_{m,i2}^{-1} mod\ p)$.

*Step 2 — After $S_m$ and $S_n$ authenticate the legitimacy of $U_i$ and $U_j$, Server $S_m$ performs the following sub-steps.*

Step 2-1: $S_m$ chooses random number $b_{m,i}$ and computes $W_{m,i} = b_{m,i} * G$.
Step 2-2: $S_m$ computes $Y_{m,i} = Nf_p * R_{i,m} * b_{m,i}$.
Step 2-3: $S_m$ calculates signature $\delta_m = Sig_{k_m}(Y_{m,i})$.
Step 2-4: $S_m$ transmits $\delta_m$, $Cert_m$ to $S_n$ for verification.

*Step 3 — Server $S_m$ first verifies $\delta_n$, $Cert_n$ received from the server $S_n$. After verifying $S_n$ 's identity, server $S_m$ uses received numbers to compute the following computations.*

Step 3-1: $S_m$ computes $K_{m,i} = b_{m,i} * Y_{n,j} = Nf_q * a_{j,n} * b_{n,j} * b_{m,i} * G$.



Server $S_m$

1. $Z_{m,i} = (z_{m,i1}, z_{m,i2}) = k_m C_{i,m1}$
2. $Q_p || Nf_p' = D_{f_{p,m}}(\theta_{p,m})$
3. $c_{i,m21} = c_{i,m21}' \oplus Nf_p$
4. $u_{i,m} || SID_{i,m} = T_{i,m} \oplus c_{i,m21} * z_{m,i1}{}^{-1} mod\ p$
5. $u_{m,i} = H(SID_{i,m} || x_m)$
6. $u_{m,i} \overset{?}{=} u_{i,m}$, *verify user* $U_i$'s legitimacy
7. $R_{i,m} = (c_{i,m22} * z_{m,i2}{}^{-1} mod\ p, c_{i,m23} * z_{m,i2}{}^{-1} mod\ p)$
8. Choose random number $b_{m,i}$
9. $W_{m,i} = b_{m,i} * G$
10. $Y_{m,i} = Nf_p * R_{i,m} * b_{m,i}$
11. $\delta_m = Sig_{k_m}(Y_{m,i})$

$\xrightarrow{\delta_m, Cert_m}$

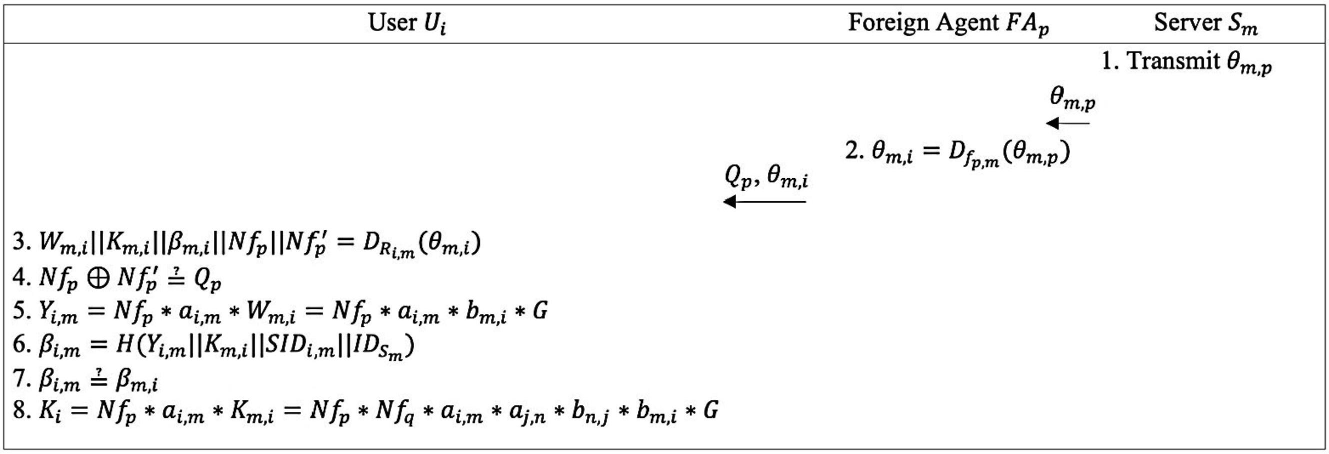
$\xleftarrow{\delta_n, Cert_n}$

12. Use $Cert_n$ to verify $S_n$'s identity then open $\delta_n$
13. $K_{m,i} = b_{m,i} * Y_{n,j} = Nf_q * a_{j,n} * b_{n,j} * b_{m,i} * G$
14. $\beta_{m,i} = H(Y_{m,i} || K_{m,i} || SID_{i,m} || ID_{S_m})$
15. $\theta_{m,i} = E_{R_{i,m}}(W_{m,i} || K_{m,i} || \beta_{m,i} || Nf_p || Nf_p')$
16. $\theta_{m,p} = E_{f_{p,m}}(\theta_{m,i})$

Server $S_n$

1'. $Z_{n,j} = (z_{n,j1}, z_{n,j2}) = k_n C_{j,n1}$
2'. $Q_q || Nf_q' = D_{f_{q,n}}(\theta_{q,n})$
3'. $c_{j,n21} = c_{j,n21}' \oplus Nf_q$
4'. $u_{j,n} || SID_{j,n} = T_{j,n} \oplus c_{j,n21} * z_{n,j1}{}^{-1} mod\ p$
5'. $u_{n,j} = H(SID_{j,n} || x_n)$
6'. $u_{n,j} \overset{?}{=} u_{j,n}$, *verify user* $U_j$'s legitimacy
7'. $R_{j,n} = (c_{i,m22} * z_{m,i2}{}^{-1} mod\ p, c_{i,m23} * z_{m,i2}{}^{-1} mod\ p)$
8'. Choose random number $b_{n,j}$
9'. $W_{n,j} = b_{n,j} * G$
10'. $Y_{n,j} = Nf_q * R_{j,n} * b_{n,j}$
11'. $\delta_n = Sig_{k_n}(Y_{n,j})$
12'. Use $Cert_m$ to verify $S_m$'s identity then open $\delta_m$
13'. $K_{n,j} = b_{n,j} * Y_{m,i} = Nf_p * a_{i,m} * b_{m,i} * b_{n,j} * G$
14'. $\beta_{n,j} = H(Y_{n,j} || K_{n,j} || SID_{j,n} || ID_{S_n})$
15'. $\theta_{n,j} = E_{R_{j,n}}(W_{n,j} || K_{n,j} || \beta_{n,j} || Nf_q || Nf_q')$
16'. $\theta_{n,q} = E_{f_{q,n}}(\theta_{n,j})$

**FIGURE 5.** Mutual authentication and key exchange process between two servers.



User $U_i$ | Foreign Agent $FA_p$ | Server $S_m$

1. Transmit $\theta_{m,p}$

$\xleftarrow{\theta_{m,p}}$

2. $\theta_{m,i} = D_{f_{p,m}}(\theta_{m,p})$

$\xleftarrow{Q_p, \theta_{m,i}}$

3. $W_{m,i} || K_{m,i} || \beta_{m,i} || Nf_p || Nf_p' = D_{R_{i,m}}(\theta_{m,i})$
4. $Nf_p \oplus Nf_p' \overset{?}{=} Q_p$
5. $Y_{i,m} = Nf_p * a_{i,m} * W_{m,i} = Nf_p * a_{i,m} * b_{m,i} * G$
6. $\beta_{i,m} = H(Y_{i,m} || K_{m,i} || SID_{i,m} || ID_{S_m})$
7. $\beta_{i,m} \overset{?}{=} \beta_{m,i}$
8. $K_i = Nf_p * a_{i,m} * K_{m,i} = Nf_p * Nf_q * a_{i,m} * a_{j,n} * b_{n,j} * b_{m,i} * G$

**FIGURE 6.** Authentication process among the server, foreign agent and user.

Step 3-2: $S_m$ computes $\beta_{m,i} = H(Y_{m,i}||K_{m,i}||SID_{i,m}||ID_{S_m})$, $\theta_{m,i} = E_{R_{i,m}}(W_{m,i}||K_{m,i}||\beta_{m,i}||Nf_p||Nf'_p)$, $\theta_{m,p} = E_{f_{p,m}}(\theta_{m,i})$.

### 2) AUTHENTICATION PHASE AMONG SERVER, FOREIGN AGENT AND USER

As shown in Figure 6, authentication process among the server, foreign agent and user is described as follows.

*Step 1 — Server $S_m$ transmits $\theta_{m,p}$ to foreign agent $FA_p$.*

*Step 2 — $FA_p$ computes $\theta_{m,i} = D_{f_{p,m}}(\theta_{m,p})$, and transmits $\theta_{m,i}$, $Q_p$ to $U_i$.*

*Step 3 — $U_i$ and $U_j$ respectively verify $S_m$ and $S_n$, then compute a conversation key. User $U_i$ performs the following sub-steps.*

Step 3-1: $U_i$ computes $W_{m,i}||K_{m,i}||\beta_{m,i}||Nf_p||Nf'_p = D_{R_{i,m}}(\theta_{m,i})$.

Step 3-2: $U_i$ verifies whether $Nf_p \oplus Nf'_p \overset{?}{=} Q_p$.

Step 3-3: $U_i$ computes: $Y_{i,m} = Nf_p * a_{i,m} * W_{m,i} = Nf_p * a_{i,m} * b_{m,i} * G$.

Step 3-4: $U_i$ computes $\beta_{i,m} = H(Y_{i,m}||K_{m,i}||ID_{U_i}||ID_{S_m})$.

Step 3-5: $U_i$ compares $\beta_{i,m}$ with the received $\beta_{m,i}$. If there is a match, the server $S_m$ is legitimate.

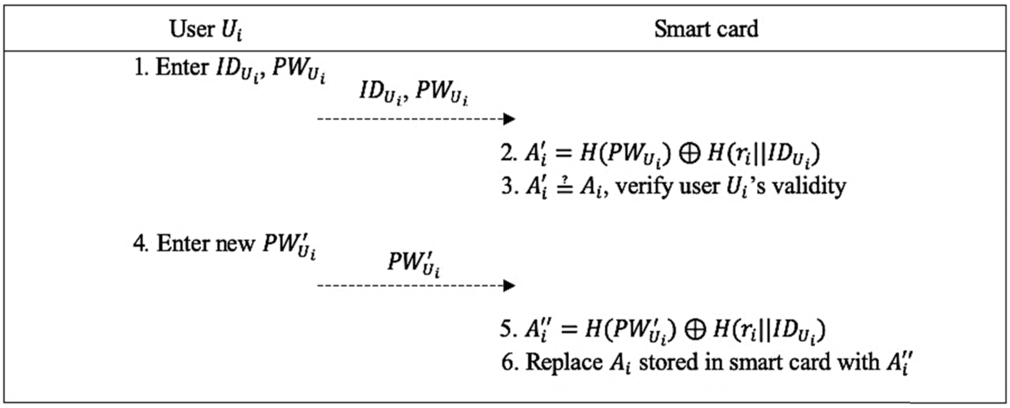


**FIGURE 7.** Password update phase.

**TABLE 2.** The notation used for logical analysis.

| Symbols | Description |
|---|---|
| $P, Q$ | Principals |
| $C$ | Communicating channel |
| $X, Y$ | Messages |
| $C(X)$ | The message $X$ is transited via channel $C$ |
| $r(C)$ / w($C$) | The set of readers/writers of channel $C$ |
| $P\|X$ | $P$ believes statement $X$. The construct is central to the logic |
| $P \triangleleft C(X)$ | $P$ sees $C(X)$. The message $X$ is transited via channel $C$ and can be observed by $P$. $P$ must be a reader of channel $C$ to read message $X$ |
| $P\|\sim X$ | $P$ once said $X$. $P$ at some time sent a message including $X$ |
| $P \triangleleft X\|C$ | $P$ sees $X$ via $C$. The message $X$ is transited via channel $C$ and can be received by $P$ |
| $(X)_K$ | $X$ is hashed with the key $K$ |
| $P \overset{K}{\leftrightarrow} Q$ | $P$ and $Q$ may use the shared key $K$ to communicate. Here $K$ will never be discovered by any principals expect for $P$ and $Q$ |
| $\#(X)$ | The formula $X$ is fresh, $X$ has not been sent in a message at any time before |

Step 3-6: Using $K_{m,i}$ sent by $S_m$, $U_i$ computes his/her conversation key $K_i = Nf_p * a_{i,m} * K_{m,i} = Nf_p * Nf_q * a_{i,m} * a_{j,n} * b_{n,j} * b_{m,i} * G$. $K_j$ is similarly computed by $U_j$ at the same time.

### F. PASSWORD UPDATE PHASE

As shown in Figure 7, the user $U_i$ and his/her smart card perform the following steps to complete password update phase.

*Step 1 — $U_i$ enters $ID_{U_i}$, $PW_{U_i}$, then smart card computes $A_i' = H(PW_{U_i}) \oplus H(r_i||ID_{U_i})$. After that, $A_i$ and $A_i'$ are compared to verify the legitimacy of $U_i$.*

*Step 2 — $U_i$ enters a new password $PW'_{U_i}$. Smart card computes $A_i'' = H(PW'_{U_i}) \oplus H(r_i||ID_{U_i})$, then replaces $A_i$ with $A_i''$.*

## IV. FORMAL SECURITY ANALYSIS

### A. LOGICAL ANALYSIS USING BAN LOGIC

This section describes the logical analysis of the proposed protocol by using BAN logic, which was defined and presented by [32], [33]. Table 2, Table 3 and Table 4 [32]–[34] respectively defines the notations, assumptions and rules used in this analysis. On the basis of the assumptions and logical analyses, the proposed protocol must realize the following four goals of authentication and key agreement as follows.

(G1) $U_i \equiv U_i \overset{K_i}{\longleftrightarrow} U_j$: User $U_i$ believes that $K_i$ is a symmetric key shared between $U_i$ and $U_j$.

(G2) $U_j \equiv U_i \overset{K_i}{\longleftrightarrow} U_j$: User $U_j$ believes that $K_i$ is a symmetric key shared between $U_i$ and $U_j$.

(G3) $U_i \equiv U_j \equiv U_i \xleftrightarrow{K_j} U_j$: User $U_i$ believes that $U_j$ is convinced of $K_j$ is a symmetric shared key between $U_i$ and $U_j$.

(G4) $U_j \equiv U_i \equiv U_i \xleftrightarrow{K_j} U_j$: User $U_j$ believes that $U_i$ is convinced of $K_j$ is a symmetric shared key between $U_i$ and $U_j$.

To accomplish Goal 1, firstly, we must prove $a_{i,m}$, $Nf_p$, and $K_{m,i}$ are trusted by $U_i$. According to [32]–[34], the proposed protocol is described in logic with the following steps.

*Step 1* — $FA_p \lhd (s_{i,m} * G, y_{i,m1} * (u_{i,m}||SID_{i,m}) \oplus T_{i,m}\ mod\ p, y_{i,m2} * r_{i,m1} mod\ p, y_{i,m2} * r_{i,m2} mod\ p, T_{i,m}$

*Step 2*— $S_m \lhd (s_{i,m} * G, y_{i,m1} * (u_{i,m}||SID_{i,m}) \oplus T_{i,m} mod\ p \oplus Nf_p, y_{i,m2} * r_{i,m1} mod\ p, T_{i,m}, \{(Nf_p \oplus Nf'_p)||Nf'_p\}_{f_{p,m}})$

*Step 3*— $FA_p \lhd (\{\{W_{m,i}||K_{m,i}||\beta_{m,i}||Nf_p||Nf'_p\}_{R_{i,m}}\}_{f_{p,m}})$

*Step 4* — $U_i \lhd (Q_p, \{W_{m,i}||K_{m,i}||\beta_{m,i}||Nf_p||Nf'_p\}_{R_{i,m}})$.

We have that

$$U_i \equiv a_{i,m} \to U_i \tag{1}$$

$$U_i \equiv Nf_p \to U_i \tag{2.1}$$

and

$$U_i \equiv K_{m,i} \to U_i \tag{2.2}$$

must hold because of interpretation (I3) and assumption (A5). Next, to accomplish Eq (2.1) and (2.2), we have that

$$\begin{aligned} S_m \equiv (&FA_p|| \sim (Nf_p \to FA_p, s_{i,m} * G, y_{i,m1} * (u_{i,m}||SID_{i,m}) \\ &\oplus T_{i,m} mod\ p \oplus Nf_p, y_{i,m2} * r_{i,m1} mod\ p, y_{i,m2} \\ &* r_{i,m2} mod\ p, T_{i,m}, \{(Nf_p \oplus Nf'_p)||Nf'_p\}_{f_{p,m}}) \\ &\to Nf_p \to FA_p) \end{aligned} \tag{3.1}$$

$$\begin{aligned} FA_p \equiv (&S_m|| \sim (\{W_{m,i}||K_{m,i}||\beta_{m,i}||Nf_p||Nf'_p\}_{R_{i,m}}) \\ &\to S_m, (\{\{W_{m,i}||K_{m,i}||\beta_{m,i}||Nf_p||Nf'_p\}_{R_{i,m}}\}_{f_{p,m}}) \\ &\to (\{W_{m,i}||K_{m,i}||\beta_{m,i}||Nf_p||Nf'_p\}_{R_{i,m}}) \to S_m) \end{aligned} \tag{3.2}$$

$$\begin{aligned} U_i \equiv (&FA_p|| \sim (Nf_p \to FA_p, Q_p, \\ &\{W_{m,i}||K_{m,i}||\beta_{m,i}||Nf_p||Nf'_p\}_{R_{i,m}}) \to Nf_p \to FA_p) \end{aligned} \tag{3.3}$$

$$\begin{aligned} U_i \equiv (&FA_p|| \sim (K_{m,i} \to FA_p, Q_p, \\ &\{W_{m,i}||K_{m,i}||\beta_{m,i}||Nf_p||Nf'_p\}_{R_{i,m}}) \to K_{m,i} \to FA_p) \end{aligned} \tag{3.4}$$

$$S_m \equiv (FA_p|| \sim Nf_p \to FA_p) \tag{4.1}$$

$$FA_p \equiv (S_m|| \sim (\{W_{m,i}||K_{m,i}||\beta_{m,i}||Nf_p||Nf'_p\}_{R_{i,m}}) \to S_m) \tag{4.2}$$

$$U_i \equiv (FA_p|| \sim (Nf_p \to FA_p) \tag{4.3}$$

and

$$U_i \equiv (FA_p|| \sim (K_{m,i} \to FA_p) \tag{4.4}$$

must hold because of assumptions (A3), (A6) and the rationality rule (R1). To accomplish Eq (4.1), (4.2), (4.3) and (4.4) we have that

$$S_m \equiv \#(Nf_p \to FA_p) \tag{5.1}$$

$$FA_p \equiv \#((\{W_{m,i}||K_{m,i}||\beta_{m,i}||Nf_p||Nf'_p\}_{R_{i,m}}) \to S_m) \tag{5.2}$$

$$U_i \equiv \#(Nf_p \to FA_p) \tag{5.3}$$

and

$$U_i \equiv \#(K_{m,i} \to FA_p) \tag{6.1}$$

must hold because of the freshness rules (F1), (F2) and assumption (A4). To accomplish Eq (5.1), (5.2), (5.3) and (6.1), we have that

$$FA_p \equiv (W(C_{FA_p,S_m}) = \{FA_p, S_m\}) \tag{7.1}$$

$$S_m \equiv (W(C_{FA_p,S_m}) = \{FA_p, S_m\}) \tag{7.2}$$

$$FA_p \equiv (W(C_{FA_p,U_i}) = \{FA_p, U_i\}) \tag{7.3}$$

$$S_m \in r(C_{FA_p,S_m}) \tag{8.1}$$

$$FA_p \in r(C_{FA_p,S_m}) \tag{8.2}$$

$$U_i \in r(C_{FA_p,U_i}) \tag{8.3}$$

$$S_m \equiv \lhd C_{FA_p,S_m}(Nf_p \to FA_p) \tag{9.1}$$

$$FA_p \equiv \lhd C_{FA_p,S_m}(\{W_{m,i}||K_{m,i}||\beta_{m,i}||Nf_p||Nf'_p\}_{R_{i,m}}) \to S_m) \tag{9.2}$$

$$U_i \equiv \lhd C_{FA_p,U_i}(Nf_p \to FA_p) \tag{9.3}$$

and

$$U_i \equiv \lhd C_{FA_p,U_i}(K_{m,i} \to FA_p) \tag{9.4}$$

must hold because of the interpretation rule (I1), the seeing rules (S1), (S2), assumptions (A1), (A2). By using interpretation rule (I3), we have $U_i \equiv K_i = a_{i,m} * Nf_p * K_{m,i}$.

Subsequently, using the same arguments of assumptions, rules, sysmetric keys, we have $K_{m,i} = b_{m,i} * Y_{n,j}$ trusted by $S_m$, and $Y_{n,j} = Nf_q * R_{j,n} * b_{n,j}$ trusted by $S_n$. Particularly for $Y_{n,j}$, the trust from Certificate Authority is needed, which is regarded as an assumption.

Finally, we have that the proposed protocol realizes

Goal 1: $U_i \equiv U_i \xleftrightarrow{K_i} U_j$

Similarly, we have that the proposed scheme realizes Goal 2: $U_j \equiv U_i \xleftrightarrow{K_i} U_j$ by using the same arguments of Goal 1.

To accomplish Goal 3, we have that

$$U_i \equiv ((U_j|| \sim U_i \xleftrightarrow{Ki} U_j) \to (U_j \equiv U_i \xleftrightarrow{K_i} U_j)) \tag{10}$$

and

$$U_i \equiv (U_j|| \sim U_i \xleftrightarrow{K_i} U_j) \tag{11}$$

must hold because of the rationality rule (R1) and assumption (A3). To accomplish Eq (11), we have that

$$U_i \equiv \#(U_i \xleftrightarrow{K_i} U_j) \tag{12}$$

must hold because of the freshness rules (F1), (F2) and assumption (A4). To accomplish Eq (12), we have that

$$U_i \lhd C_{U_i,U_j}(U_i \xleftrightarrow{K_i} U_j) \tag{13}$$

$$U_i \equiv (w(C_{U_i,U_j}) = \{U_i, U_j) \tag{14}$$

**TABLE 3.** The assumptions of the proposed protocol.

| Assumptions | Explanation |
|---|---|
| (A1) $A \in r(C_{A,B})$ | $A$ can read from channel $C_{A,B}$ |
| (A2) $A \equiv (w(C_{A,B}) = \{A, B\}$ | $A$ believes that $A$ and $B$ can write on $C_{A,B}$ |
| (A3) $A \equiv (B \Vert\sim \phi \rightarrow \phi)$ | $A$ believes that $B$ only says what it believes |
| (A4) $A \equiv \#(N_A)$ | $A$ believes that $N_A$ is fresh |
| (A5) $A \equiv a_{i,m} \rightarrow A$ | $A$ believes that $a_{i,m}$ is its secret number |
| (A6) $A \equiv A \overset{f}{\leftrightarrow} B$ | Server $A$ believes that $f$ is a symmetric key shared between $A$ and $B$ |

**TABLE 4.** The inference rules of the logic of the proposed protocol.

| Rules | Explanation |
|---|---|
| | *Seeing rules* |
| (S1) $\frac{P \lhd C(X), P \in r(C)}{P \equiv (P \lhd X \mid C), P \lhd X}$ | If $P$ receives and reads $X$ via $C$, then $P$ believes that $X$ has arrived on $C$ and $P$ sees $X$ |
| (S2) $\frac{P \lhd (X,Y)}{P \lhd X, P \lhd Y}$ | If $P$ sees a hybrid message $(X, Y)$, then $P$ sees $X$ and $Y$ separately |
| | *Interpretation rules* |
| (I1) $\frac{P \equiv (w(C) = \{P,Q\}}{\mathrm{P} \equiv (P \lhd X \mid C) \rightarrow Q \mid\sim X}$ | If $P$ believes that $C$ can only be written by $P$ and $Q$, then $P$ believes that if $P$ receives $X$ via $C$, then $Q$ said $X$ |
| (I2) $\frac{P \equiv (Q \mid\sim (X,Y))}{P \equiv (Q \mid\sim X), P \equiv (Q \mid\sim Y)}$ | If $P$ believes that $Q$ said a hybrid message $(X, Y)$, then $P$ believes that $Q$ has said $X$ and $Y$ separately |
| (I3) $\frac{U_i \equiv (a_{i,m} \rightarrow U_i).U_i \equiv (Nf_p \rightarrow FA_p).U_i \equiv (K_{m,i} \rightarrow S_m)}{P \equiv (K_i = a_{i,m} * Nf_p * K_{m,i})}$ | If $U_i$ believes that $a_{i,m}$ is a secret number, and that $Nf_p$ and $K_{m,i}$ are components from $FA_p$ and $S_m$ respectively, then $U_i$ believes that $K_i$ is a symmetric key shared between $U_i$ and $U_j$ |
| | *Freshness rules* |
| (F1) $\frac{P \equiv (Q \mid\sim X), P \equiv \# (X)}{P \equiv (Q \mid\sim X)}$ | If $P$ believes that another $Q$ said $X$ and $P$ also believes that $X$ is fresh, then $P$ believes that $Q$ has recently said $X$ |
| (F2) $\frac{P \equiv \# (X)}{P \equiv \# (X,Y)}$ | If $P$ believes that a part of a mixed message $X$ is fresh, then it believes that the whole message $(X, Y)$ is fresh |
| | *Rationality rules* |
| (R1) $\frac{P \equiv (\Phi_1 \rightarrow \Phi_2), P \equiv \Phi_1}{P \equiv \Phi_2}$ | If $P$ believes that $\Phi_1$ implies $\Phi_2$ and $P$ believes that $\Phi_1$ is true, then $P$ believes that $\Phi_2$ is true |

and

$$U_i \in r(C_{U_i,U_j}) \tag{15}$$

must hold because of the interpretation rule (I1), the assumptions (A1), (A2) and the seeing rules (S1) and (S2).

Thus, the proposed protocol realizes

Goal 3: $U_i \equiv U_j \equiv U_i \xleftrightarrow{K_j} U_j$.

Similarly, using the same arguments of Goal 3, the proposed protocol realizes Goal 4: $U_j \equiv U_i \equiv U_i \xleftrightarrow{K_j} U_j$.

Therefore, our proposed protocol realizes Goal 1, 2, 3 and 4.

### B. SECURITY VERIFICATION USING AVISPA TOOL

We verify our scheme using widely accepted Automated Validation of Internet Security Protocols and Applications

```
role user (U, S, F: agent, Kus, Rus, Kas: symmetric_key, Ks: public_key, H, Mul: hash_func, SND, RCV: channel
(dy))
played_by U def=
local State: nat,
SPWim, SAim, Ri, SIDim, Umi, Bmi, Tim, Nfp, Nfp1, Qp, Cim1, G, B1mi, Aim, Wmi, Ymi, Dm, Kmi, B2mi, Dmi,
Rim, Uim, Ynj, IDsm, Cim2, Kua, Xm, Bpm, A: text
init State := 0
transition
% Registration phase
1.  State = 0 /\ RCV(start) =|>
State':= 1
%/\ Enter SIDim to smart card
/\ SPWim' := new() /\ SAim' := xor(H(SPWim'),H(Ri.SIDim))
/\ SND({SIDim.SAim'}_Kus)
/\ secret(SIDim,g1,{U,S}) /\ secret(SAim',g2,{U,S}) /\ secret(SPWim',g3,{U})
2.  State = 1 /\ RCV({xor(Umi',xor(H(SPWim'),H(Ri.SIDim)))}_Kus) =|>
State':= 2
/\ Bmi' := xor(Umi',xor(H(SPWim'),H(Ri.SIDim)))
%/\ Store Bmi in flash storage %/\ Store SAim in smart card
% Mutual authentication and key exchange phase
3.  State = 0 /\ RCV(start) =|>
State':= 1
%/\ Enter IDUi, PWUi to smart card %/\ Smart card verify legitimacy of user %/\ Enter SIDim to smart card
/\ Uim' := xor(Bmi,SAim) /\ Aim' := new() /\ Tim' := new() /\ Cim1' := {xor((Uim'.SIDim),Tim')}_Ks
/\ SND(Cim1'.Tim')
/\ witness(U,S,u_s_tim,Tim')
4.  State = 1 /\ RCV(({Mul(B1mi'.G).Mul(B1mi'.Ynj).H(Ymi'.Kmi'.SIDim.IDsm).Nfp'.Nfp1'}_Rus).Qp) =|>
State':= 2
/\ Kua' := Mul(Nfp'.Aim.Mul(B1mi'.Ynj))
/\ request(S,U,s_u_b1mi,B1mi')
end role
```

**FIGURE 8.** **The HLPSL specification of the user.**

(AVISPA) tool [35]. AVISPA tool executes the simulated protocol specified by HLPSL language [36]. For verifying cryptographic protocol, AVISPA tool includes four backends as follows.

- On-the-fly Model-Checker (OFMC)
- Constraint Logic based Attack Searcher (CL-AtSe)
- SAT-based ModelChecker (SATMC)
- Tree Automata based on automatic approximations for the analysis of security protocols (TA4SP)

In accordance with our proposed protocol, three roles including the user $U_i$, the server $S_j$ and the foreign agent $FA_p$ are defined in the specification, HLPSL of which are shown in Figure 8, Figure 9 and Figure 10 respectively. Besides, session role, environment role and goals are also specified in HLPSL (shown in Figure 11). Since elliptic curve key generation is not supported in AVISPA, public key, private key and session key of ECC are predefined as Ks, inv(Ks) and Rus respectively. We consider six secrecy goals and two authentication properties for verification of our scheme. These goals and authentication properties are described as follows.

- secrecy_of g1: SIDim is kept secret to the U and the S.
- secrecy_of g2: SAim' is kept secret to the U and the S.
- secrecy_of g3: SPWim' is kept secret to the U.
- secrecy_of g4: Bmi' is kept secret to the U and the S.
- secrecy_of g5: Nfp' is kept secret to the U, the S and the F.
- secrecy_of g6: Nfp1' is kept secret to the U, the S and the F.
- authentication_on u_s_tim: The server S authenticates the user U based on Tim' received from the message of the user U.
- authentication_on s_u_b1mi: The user U authenticates the user U based on B1mi' received from the message of the server S.

As show in Figure 12, the analysis results of the proposed protocol using OFMC confirm that the stated security properties are satisfied for a bounded number of sessions as specified in the environment role. Therefore, the proposed protocol is safe against various attacks, which are specifically described in Section V.

## V. INFORMAL SECURITY ANALYSIS

The primary purpose of our propose protocol is to provide conversation key for two users. In other words, the secure shared key $K$ of the user $U_i$, $U_j$ and is computed through verification and authentication of the home servers $S_m$, $S_n$ and foreign agents $FA_p$, $FA_q$. The details of semantic security analysis of our proposed protocol are presented as follows.

### A. PROVIDES ROBUST VERIFICATION

In Step 1 of login phase, the smart card computes $A_i' = H(PW_{U_i}) \oplus H(r_i||ID_{U_i})$, then confirms $A_i$ and $A_i'$. The user is verified to be legitimate if there is match, otherwise the smart

```
role server (U, S, F: agent, Kus, Rus, Kas: symmetric_key, Ks: public_key, H, Mul: hash_func, SND, RCV: channel
(dy))
played_by S def=
local State: nat,
SPWim, SAim, Ri, SIDim, Umi, Bmi, Tim, Nfp, Nfp1, Qp, Cim1, G, B1mi, Aim, Wmi, Ymi, Dm, Kmi, B2mi, Dmi,
Rim, Uim, Ynj, IDsm, Cim2, Kua, Xm, Bpm, A: text
init State := 0
transition
% Registration phase
1. State = 0 /\ RCV({SIDim.xor(H(SPWim'),H(Ri.SIDim))}_Kus) =|>
State':= 1
/\ Umi' := H(SIDim.Xm) /\ Bmi' := xor(Umi',xor(H(SPWim'),H(Ri.SIDim)))
/\ SND({Bmi'}_Kus)
/\ secret(Bmi',g4,{U,S})
% Mutual authentication and key exchange phase
2.  State = 0 /\ RCV(xor(({xor((Uim'.SIDim),Tim')}_Ks),Nfp').({xor((Uim'.SIDim),Tim')}_Ks).({Qp'.Nfp1'}_Kas)) =|>
State':= 1
/\ Cim1' := xor(xor(({xor((Uim'.SIDim),Tim')}_Ks),Nfp'),Nfp') /\ B1mi' := new() /\ Wmi' := Mul(B1mi'.G) /\ Ymi' :=
Mul(Nfp'.Rim.B1mi')
%/\ Dm = Signature of Sm with Ymi'
%/\ Send Dm to server Sn %/\ Receive Dn' from Sn, verify identity of Sn, and open Dn'
/\ Kmi' := Mul(B1mi'.Ynj) /\ B2mi' := H(Ymi'.Kmi'.SIDim.IDsm) /\ Dmi' := {Wmi'.Kmi'.B2mi'.Nfp'.Nfp1'}_Rus
/\ SND({{Wmi'.Kmi'.B2mi'.Nfp'.Nfp1'}_Rus}_Kas)
/\ request(U,S,u_s_tim,Tim')
/\ witness(S,U,s_u_b1mi,B1mi')
end role
```

**FIGURE 9.** The HLPSL specification of the home server.

```
role foreign (U, S, F: agent, Kus, Rus, Kas: symmetric_key, Ks: public_key, H, Mul: hash_func, SND, RCV: channel
(dy))
played_by F def=
local State: nat,
SPWim, SAim, Ri, SIDim, Umi, Bmi, Tim, Nfp, Nfp1, Qp, Cim1, G, B1mi, Aim, Wmi, Ymi, Dm, Kmi, B2mi, Dmi,
Rim, Uim, Ynj, IDsm, Cim2, Kua, Xm, Bpm, A: text
init State := 0
transition
% Mutual authentication and key exchange phase
1. State = 0 /\ RCV(({xor((Uim'.SIDim),Tim')}_Ks).Tim') =|>
State':= 1
/\ Nfp' := new() /\ Nfp1' := new() /\ Qp' := xor(Nfp',Nfp1') /\ Bpm' := {Qp'.Nfp1'}_Kas /\ Cim2' :=
xor(({xor((Uim'.SIDim),Tim')}_Ks),Nfp')
/\ SND(xor(({xor((Uim'.SIDim),Tim')}_Ks),Nfp').({xor((Uim'.SIDim),Tim')}_Ks).Bpm')
/\ secret(Nfp',g5,{U,S,F}) /\ secret(Nfp1',g6,{U,S,F})
2. State = 1 /\
RCV({{Mul(B1mi'.G).Mul(B1mi'.Ynj).H(Mul(Nfp'.Rim.B1mi').Mul(B1mi'.Ynj).SIDim.IDsm).Nfp'.Nfp1'}_Rus}_Kas)
=|>
State':= 2
/\ SND(({Mul(B1mi'.G).Mul(B1mi'.Ynj).H(Mul(Nfp'.Rim.B1mi').Mul(B1mi'.Ynj).SIDim.IDsm).Nfp'.Nfp1'}_Rus).Qp)
end role
```

**FIGURE 10.** The HLPSL specification of the foreign agent.

card rejects the request. In Step 1 of the mutual authentication and key exchange phase, the server $S_m$ decrypts $C_{i,m1}$ and $C'_{i,m2}$ using $k_m$ to obtain $u_{i,m}$ and $SID_{i,m}$. The server $S_m$ then computes $u_{m,i} = H(SID_{i,m}||x_m)$ using $x_m$ and confirms $u_{m,i}\overset{?}{=}u_{i,m}$. Similarly, if there is a match, legitimate user is confirmed. Besides, in Step 2 of the mutual authentication and key exchange phase, public key of the server $S_n$ is verified by using certificate $Cert_n$. In Step 5 of the mutual authentication and key exchange phase, the user uses $R_{i,m}$ to decrypt $\theta_{m,i}$ then obtains $W_{m,i}$. After that, he/she calculates $Y_{i,m} = a_{i,m} * W_{m,i}$, $\beta_{i,m} = H(Y_{i,m}||K_{m,i}||ID_{U_i}||ID_{S_m})$. The user then confirms $\beta_{i,m}$ and $\beta_{m,i}$ to verify the server $S_m$. Hence, our protocol provides a robust verification of communicating participants.

### B. PROVIDES MUTUAL AUTHENTICATION

In Step 1 of the mutual authentication and key exchange phase, $u_{m,i}$ is computed to verify the user. Also, in this phase, both of the servers' signatures are verified by the corresponding certificates. In Step 5 of the mutual authentication and key exchange phase, the user decrypts $\theta_{m,i}$ and computes $\beta_{i,m} = H(Y_{i,m}||K_{m,i}||ID_{U_i}||ID_{S_m})$. The user then confirms $\beta_{m,i}$ and

```
role session (U, S, F: agent, Kus, Rus, Kas: symmetric_key, Ks: public_key, H, Mul: hash_func) def=
local SU, RU, SS, RS, SF, RF: channel (dy)
composition
user (U,S,F,Kus,Rus,Kas,Ks,H,Mul,SU,RU) /\ server (U,S,F,Kus,Rus,Kas,Ks,H,Mul,SS,RS) /\ foreign
(U,S,F,Kus,Rus,Kas,Ks,H,Mul,SF,RF)
end role
role environment() def=
const u, s, f: agent,
kus, rus, kas, kui: symmetric_key,
ks, ki: public_key,
h, mul: hash_func,
u_s_tim, s_u_b1mi, g1, g2, g3, g4, g5, g6: protocol_id
intruder_knowledge = {u,s,f,ks,ki,inv(ki)}
composition
session(u,s,f,kus,rus,kas,ks,h,mul) /\ session(i,s,f,kui,kui,kui,ks,h,mul) /\ session(u,i,f,kui,kui,kui,ks,h,mul) /\
session(u,s,i,kui,kui,kui,ks,h,mul)
end role
goal
secrecy_of g1, g2, g3, g4, g5, g6
authentication_on u_s_tim, s_u_b1mi
end goal
environment()
```

**FIGURE 11.** The HLPSL specification of the session role, environment role and goals.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/Anonymous_E2E.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.94s
  visitedNodes: 144 nodes
  depth: 6 plies
```

**FIGURE 12.** The results of the OFMC back-end.

$\beta_{i,m}$ to verify the legitimacy of the server $S_m$. Hence, our work provides a full mutual authentication during proposed protocol.

### C. PROVIDES STRONG USER ANONYMITY

The $ID_{U_i}$ of the user is securely stored in the smart card at registration, and only used when the smart card verify legitimacy of the user by computing $A_i' = H(PW_{U_i}) \oplus H(r_i||ID_{U_i})$. Even the server does not know of $ID_{U_i}$. The user registers and logins to the server using $SID_{i,m}$ instead of his/her original $ID_{U_i}$. After that, the server uses $SID_{i,m}$ to compute $u_{m,i}$ and $B_{m,i}$ for further authentication. The identity $SID_{i,m}$ is not available openly and is only known to the user and server. Even if $A_i$ or $SA_{i,m}$ are leaked out, attacker cannot obtain $ID_{U_i}$ or $SID_{i,m}$ respectively since these identities are protected by one-way hash function. On the other hand, in the mutual authentication phase, the attack does not know of $k_m$, so he/she cannot decrypt $C_{i,m21}$ to obtain $SID_{i,m}$. Besides, suppose the attacker compromises $\beta_{i,m}$, he/she still cannot know of $SID_{i,m}$ since $\beta_{i,m}$ is a hash value. Hence, our scheme provides a strong user anonymity.

### D. PROVIDES FORWARD SECRECY

Assume $Y_{m,i}$ or $Y_{i,m}$ is known to the attacker. Owning to discrete logarithm problem, the secret numbers $a_{i,m}$, $b_{m,i}$ will not be calculated. Moreover, $a_{i,m}$ and $b_{m,i}$ are randomly generated to compute session key and conversation key. These keys are different in every login time. Therefore, the attacker cannot derive correct keys from previous ones. Hence, this protocol achieves the forward secrecy.

### E. PROVIDES PASSWORD UPDATE

In the proposed protocol, we provide password update facility. In password update phase, the user enters current $PW_{U_i}$ for verification. After that, he/she can enter $PW'_{U_i}$ to update his/her password. The user is recommended to update his/her periodically for better security.

### F. RESISTS PASSWORD GUESSING ATTACK

In this case, the attacker tries to guess the password from known parameters. Suppose the attacker obtains $A_i$ in the smart card. He/she then tries to guess $PW_{U_i}$ from $A_i = H(PW_{U_i}) \oplus H(r_i||ID_{U_i})$. However, due to one-way hash value, it is not possible for the attacker to guess the correct password $PW_{U_i}$. Hence, our scheme can resist password guessing attack.

### G. RESISTS IMPERSONATION ATTACK

Assume the attacker knows of identity of the user and attempts to send a login request to the server $S_m$. Unless the attacker simultaneously steals $SA_{i,m}$ (stored in the smart

**TABLE 5.** Comparison of security properties.

| Security properties | Sood [24] | Jiang *et al.* [20] | Li *et al.* [29] | Gope and Hwang [16] | Ours |
|---|---|---|---|---|---|
| Resists stolen smart card attack | No | Yes | No | Yes | Yes |
| Resists man-in-the-middle attack | No | No | No | No | Yes |
| Resists replay attack | Yes | No | No | Yes | Yes |
| Resists password guessing attack | Yes | Yes | Yes | Yes | Yes |
| Resists user impersonation attack | No | Yes | No | Yes | Yes |
| Resists server impersonation attack | Yes | Yes | No | Yes | Yes |
| Resists stolen verifier attack | Yes | No | No | No | Yes |
| Provides mutual authentication | Yes | Yes | Yes | Yes | Yes |
| Provides session key generation | Yes | Yes | Yes | Yes | Yes |
| Provides strong two-factor authentication | No | Yes | No | No | Yes |
| Provides forward secrecy | Yes | Yes | Yes | Yes | Yes |
| Provides user end-to-end communication | No | No | No | No | Yes |
| Provides user anonymity | Yes | Yes | Yes | Yes | Yes |

**TABLE 6.** Comparison of computational complexities.

|  | Sood [24] | Jiang *et al.* [20] | Li *et al.* [29] | Gope and Hwang [16] | Ours |
|---|---|---|---|---|---|
| Smart card registration phase | $4nT_{ASED} + 2nT_H + nT_X$ | $3nT_H + nT_X$ | $5T_H + 2T_X$ | $4nT_H + 3nT_X$ | $2T_H + T_X$ |
| Server registration phase |  |  |  |  | $6T_H + 4T_X$ |
| Login phase | $5nT_{ASED} + 4nT_H$ | $2nT_{ASED} + 14nT_H + 3nT_X$ | $5T_H + 2T_X$ | $6nT_{SED} + 11nT_H + 12nT_X$ | $3T_{PM} + T_{ASED} + T_{SED} + 4T_H + 10T_X$ |
| Authentication and key exchange phase | $8nT_{ASED} + 6nT_H + nT_X$ |  | $38T_H + 40T_X$ |  | $2T_{PM} + 5T_{SED} + 3T_{ASED} + 3T_H + 3T_X$ |
| Password update phase | $4nT_{ASED} + 3nT_H$ | $2nT_H + 2nT_X$ | $6T_H$ | $4nT_H + 4nT_X$ | $4T_H + 2T_X$ |
| Total time complexities | $21nT_{ASED} + 13nT_H + 2nT_X$ | $2nT_{ASED} + 19nT_H + 6nT_X$ | $54T_H + 44T_X$ | $6nT_{SED} + 19nT_H + 19nT_X$ | $5T_{PM} + 6T_{SED} + 3T_{ASED} + 19T_H + 20T_X$ |
| Total rough estimation ($ms$) | $10969.51n$ | $1053.53n$ | 272.2 | $61.795n$ | 1943.175 |

$n$: number of server; $T_E$: time for performing an exponentiation operation; $T_{PM}$: time for performing an elliptic curve point multiplication operation; $T_{SED}$: time for performing a symmetric encryption/decryption operation; $T_{ASED}$: time for performing an asymmetric encryption/decryption operation; $T_H$: time for performing a hash function operation; $T_X$: time for performing an exclusive-or operation.According to [38]: $T_E \approx 522ms$; $T_{PM} \approx 63.075ms$; $T_{SED} \approx 8.7ms$; $T_{ASED} \approx 522ms$; $T_H \approx 0.5s$; $T_X \approx 0.005ms$

card), and $B_{m,i}$ (stored in the flash drive), he/she cannot compute correct $u_{i,m}$. He/She cannot impersonate the user without correct $PW_{U_i}$ for smart card verification in the beginning. In another case, the attacker obtains the server's identity and tries to impersonate it by generating a session key to encrypt a forged $\theta_{m,i}$. However, session key $R_{i,m}$ cannot be computed without correct random number $a_{i,m}$ and $s_{i,m}$. Furthermore, $Nf_p$ and $Nf'_p$ are unknown to the attacker, he/she cannot calculate $Q_p$. The user will terminate the process if $Q_p$ is not correct. Therefore, impersonation attack is resisted in our proposed protocol.

### H. RESISTS MAN-IN-THE-MINDDLE ATTACK

In this case, the attacker tries to tamper with $C_{i,m1}$, $C_{i,m2}$, $T_{i,m}$ of login request message. However, due to aforesaid impersonation attack resistance, he/she cannot generate correct $u_{i,m}$, and then the server $S_m$ will reject the login request. Besides, in the mutual authentication and key exchange phase, assume the attacker attempts to access $\theta_{m,i}$, but he/she does not have $R_{i,m}$ to decrypt $\theta_{m,i}$, and $b_{m,i}$ to compute $\beta_{m,i}$ respectively. Therefore, the attacker cannot act as a middle-man in any cases, and our protocol is secure against man-in-the-middle attack.

### I. RESISTS REPLAY ATTACK

Replay attack occurs when the attacker intercepts the message stolen from the last session then retransmits it to the server. In Step 1 of the mutual authentication and key exchange phase of our scheme, timestamp $T_{i,m}$ is used to resist replay attack. Specifically, $c_{i,m21}$ is generated with $T_{i,m}$ by XOR operation. The sever uses $T_{i,m}$ included in the message to check whether the message is resent. Only one message including the correct timestamp within $c_{i,m21}$ is accepted. Besides, the server will reject any message with incorrect timestamps. Therefore, our proposed protocol is free from replay attack.

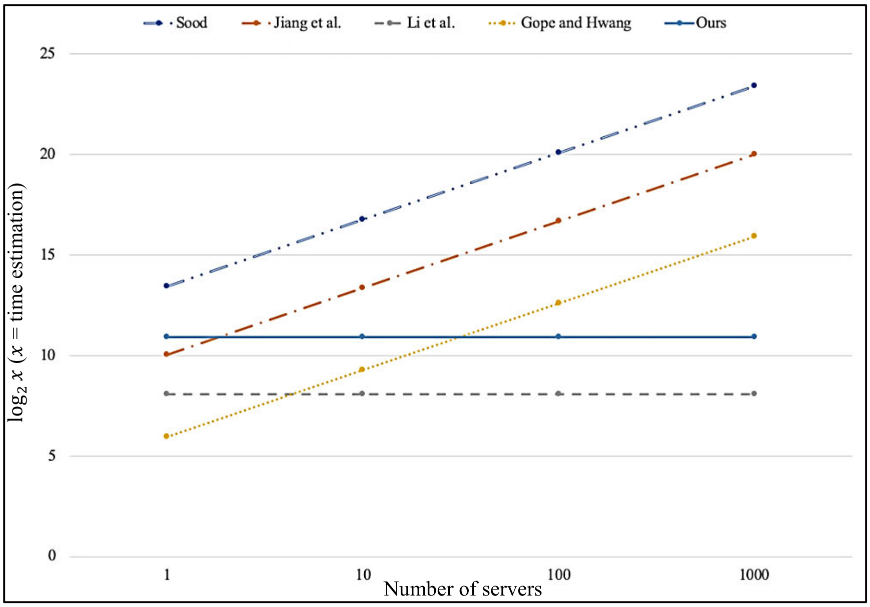


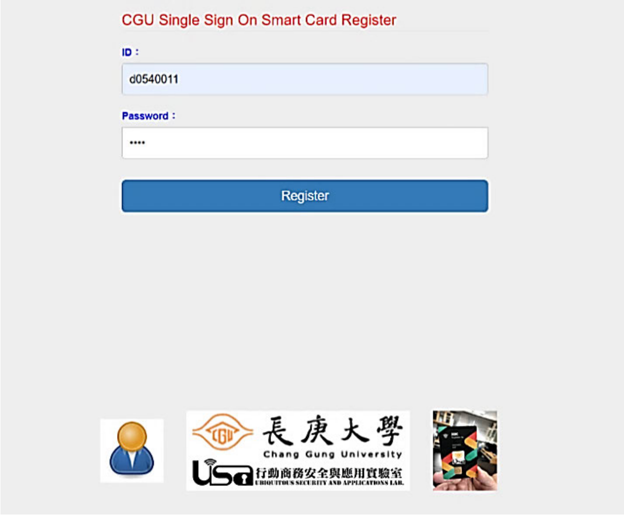
**FIGURE 13.** Running time of different schemes.

### J. RESISTS STOLEN SMART CARD ATTACK

Smart card stores random numbers $r_i$, $A_i$ and $SA_{i,m}$. In some cases, the smart card may be lost or stolen. However, the attack cannot impersonate the user since he/she does not have correct password $PW_{U_i}$. As mentioned, our protocol can provide anonymous identity and resist password guessing attack. Therefore, the stolen smart card is useless without correct $ID_{U_i}$ and $PW_{U_i}$. On the other hand, even if the attacker obtains $r_i$, $A_i$, $SA_{i,m}$, unless he/she can steal $B_{m,i}$ stored in the flash drive at the same time, the attacker cannot impersonate the legitimate user to send the login request. Hence, stolen smart card attack is avoided in the proposed protocol.

## VI. PERFORMANCE ANALYSIS

In this section, the proposed scheme is compared with the related works to judge its competence and functioning. According to Table 5, we can see that Sood [24] and Li *et al.* [29] cannot resist stolen smart card attack, which is resisted by our proposed protocol. Besides, our proposed protocol can resist man-in-the-middle attack, which is a threat in the protocols of Sood [24], Jiang *et al.* [20], Li *et al.* [29] and Gope and Hwang [16]. Our proposed protocol is secure against replay attack to which Jiang *et al.* [20] and Li *et al.* [29]'s protocols are vulnerable. Unlike ours, Sood [24], Li *et al.* [29] and Gope and Hwang [16] lacks a strong two-factor authentication. In addition, Jiang *et al.* [20], Li *et al.* [29] and Gope and Hwang [16] cannot prevent stolen verifier attack. Unlike Li *et al.* [29], our proposed protocol can resist server impersonation attack. Also, our proposed protocol can resist user impersonation attack that is a threat to Sood [24] and Li *et al.* [29]. Other than immense properties of security, our scheme bears a reasonable computational cost. As shown Figure 13, the logarithm to base 2 is defined as the running time of each scheme obtained from Table 6.


CGU Single Sign On Smart Card Register

ID : d0540011

Password : ••••

**FIGURE 14.** Smart card registration.

Specifically, the comparative value is $\log_2 x$, where $x$ is the rough estimation of running time of each scheme when $n$ (number of servers) increases from 1 to 1000. When $n$ gradually increases, our proposed protocol is explicitly more efficient than protocols of Sood [24], Jiang *et al.* [20] and Gope and Hwang [16], which were designed for single-server architecture. Only protocol of Li *et al.* [29], which was also proposed for multi-server architecture, has less running time than ours. However, as mentioned above, Li *et al.* [29]'s protocol is not accomplished, which is unsafe against well-known attacks. Our scheme is even more efficient than Sood [24]'s in single-server architecture environment. Unlike all of the previous work, our protocol can favor the end-to-end communication between the end users. Therefore, such computational cost is rational.


CGU Single Sign On Smart Card Login

ID : d0540011

Password : ••••

FIGURE 15. Smart card login.


CGU Single Sign On Create Server

Smartcard ID : d0540011

Smartcard Password : ••••

Server Name : CGMH

FIGURE 16. Server creation.

## VII. IMPLEMENTATION OF THE PROPOSED PROTOCOL

In this section, our proposed protocol is implemented with user-controlled single sign-on mechanism. Single sign-on (SSO) is a property that allows user to authenticate mobile application or web application with single username and password to access multiple applications that uses the same authentication provider [37]. SSO is consistent with multi-server architecture introduced in this paper, where user can access multiple edge servers to obtain services. In this scenario, we describe user interface of SSO system designed by Ubiquitous Security and Applications Laboratory (USA Lab.), Chang Gung University (CGU). The library of this system is written using Go Programming Language. Our system illustration includes four phases, namely, smart card registration phase, smart card login phase, server creation phase and account registration phase. In smart card registration phase (shown in Figure 14), user creates an account with identity d0540011, which is subsequently used for smart card login phase (shown in Figure 15). After having smart card login to system, the user has to create server. As shown in Figure 16, we use smart card's identity and password to create the server CGMH. After that, the user creates some more servers, namely, CGMH blockchain, CGU, GOOGLE, etc. (shown in Figure 17). Finally, in account registration phase, he/she uses ID, password and arbitrary IDs to register accounts for multiple servers so as to use potential applications developed by CGU (the applications were not described in this illustration). As show in Figure 18, smart card identity d0540011, password and user identity 01011992 are used to create an account. The user can also check the detailed information of the created accounts. Figure 19 shows that he/she has created eight accounts with two identities 01011992 and 29071991, and four servers CGU,


CGU Single Sign On Registration Phase

Smartcard ID

Smartcard Password

ID

Server : CGMH

FIGURE 17. Server query.


CGU Single Sign On Registration Phase

Smartcard ID: d0540011

Smartcard Password: ••••

ID: 01011992

Server : CGMH

FIGURE 18. Account registration.

CGU Single Sign On Check My Account

Account and Password :

Show 10 entries Search:

| ID | Server | password | seq |
|---|---|---|---|
| 01011992 | CGU | FB000003000A00F5C0B0100A8000 | 1 |
| 01011992 | CGMH | E0000003700A52B06000000000 | 1 |
| 01011992 | YAHOO | D0003E02A14F0D00020D00880000 | 1 |
| 01011992 | GOOGLE | B0000080005004402001A0106000 | 1 |
| 29071991 | CGU | 8B0000001434080568B4001E00 | 1 |
| 29071991 | CGMH | 4680000180052C0006000000000 | 1 |
| 29071991 | YAHOO | C0000E00DF00B500020804809000 | 1 |
| 29071991 | GOOGLE | F00000801C508040000040C48000 | 1 |

Showing 1 to 8 of 8 entries

**FIGURE 19.** **Account checking.**

CGMH, YAHOO and GOOGLE. The password for each account was automatically generated by the SSO system. Furthermore, system interfaces of mutual authentication and key exchange phase, and password update phase are being developed. Thereby, end user can establish conversation key and update their passwords in accordance with our proposed protocol.

## VIII. CONCLUSION

In this paper, we propose a privacy-preserved end-to-end authenticated key exchange protocol for multi-server architecture in edge computing networks. The proposed protocol is implemented with single sign-on (SSO) property and multi-server architecture. Our protocol allows mobile users to use a single easy-to-remember password to login to multiple servers then compute a conversation key for themselves during their end-to-end communication in 5G enabled NB-IoT networks. User privacy is preserved during communication process in our proposed protocol. As compared with previous works, the proposed protocol gains stronger security and better efficiency. Moreover, Elliptic Curve Cryptography with small key size is employed in our protocol. Thereby, our proposed protocol is suitable to edge computing.

Edge computing architecture plays an important role in enabling 5G technology. Thereby, security and privacy in edge computing network attract more and more attention from research community. Biometric-based authentication protocol is a good direction for providing a higher security level of communication. Also, with the increasing number of IoT or edge devices, secure authentication protocol for group communication or conference key distribution in 5G-IoT is an interesting topic for future work.

## ACKNOWLEDGMENT

The authors would like to thank anonymous referees for their valuable suggestions.

## REFERENCES

[1] C. Zavazava, "ITU work on Internet of Things," in *Proc. ICTP Workshop*, Mar. 2015, pp. 11–23.

[2] P. P. Ray, "A survey on Internet of Things architectures," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 30, no. 3, pp. 291–319, 2018.

[3] P. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5G network architecture," *IEEE Commun. Mag.*, vol. 52, no. 11, pp. 65–75, Nov. 2014.

[4] J. Zhang, B. Chen, Y. Zhao, X. Cheng, and F. Hu, "Data security and privacy-preserving in edge computing paradigm: Survey and open issues," *IEEE Access*, vol. 6, pp. 18209–18237, 2018.

[5] M. Barbareschi, A. De Benedictis, E. La Montagna, A. Mazzeo, and N. Mazzocca, "A PUF-based mutual authentication scheme for cloud-edges IoT systems," *Future Gener. Comput. Syst.*, vol. 101, pp. 246–261, Dec. 2019.

[6] F. Callegati, W. Cerroni, and M. Ramilli, "Man-in-the-middle attack to the HTTPS protocol," *IEEE Secur. Privacy*, vol. 7, no. 1, pp. 78–81, Jan./Feb. 2009.

[7] S. Malladi and J. R. Alves-Foss Heckendorn, "On preventing replay attacks on security protocols," in *Proc. Int. Conf. Secur. Manage.*, Jan. 2002, p. 8.

[8] C. Adams, "Impersonation attack," in *Encyclopedia Cryptography Security*, H. C. A. van Tilborg, Ed. Boston, MA, USA: Springer, 2005, p. 286.

[9] W.-C. Ku, H.-C. Tsai, and M.-J. Tsaur, "Stolen-verifier attack on an efficient smartcard-based one-time password authentication scheme," *IEICE Trans. Commun.*, vol. 87, pp. 2374–2376, Aug. 2004.

[10] S. Kumari and M. K. Khan, "More secure smart card-based remote user password authentication scheme with user anonymity," *Secur. Commun. Netw.*, vol. 7, no. 11, pp. 2039–2053, Nov. 2013.

[11] Y. Zhang, R. H. Deng, G. Han, and D. Zheng, "Secure smart health with privacy-aware aggregate authentication and access control in Internet of Things," *J. Netw. Comput. Appl.*, vol. 123, pp. 89–100, Dec. 2018.

[12] K. Fan, C. Zhang, K. Yang, H. Li, and Y. Yang, "Lightweight NFC protocol for privacy protection in mobile IoT," *Appl. Sci.*, vol. 8, no. 12, p. 2506, Dec. 2018.

[13] L. Bu, M. Isakov, and M. A. Kinsy, "A secure and robust scheme for sharing confidential information in IoT systems," *Ad Hoc Netw.*, vol. 92, Sep. 2019, Art. no. 101762.

[14] K. Fan, H. Xu, L. Gao, H. Li, and Y. Yang, "Efficient and privacy preserving access control scheme for fog-enabled IoT," *Future Gener. Comput. Syst.*, vol. 99, pp. 134–142, Oct. 2019.

[15] Y. Zhao, Y. Liu, A. Tian, Y. Yu, and X. Du, "Blockchain based privacy-preserving software updates with proof-of-delivery for Internet of Things," *J. Parallel Distrib. Comput.*, vol. 132, pp. 141–149, Oct. 2019.

[16] P. Gope and T. Hwang, "An efficient mutual authentication and key agreement scheme preserving strong anonymity of the mobile user in global mobility networks," *J. Netw. Comput. Appl.*, vol. 62, pp. 1–8, Feb. 2016.

[17] R. Madhusudhan and K. S. Suvidha, "An efficient and secure user authentication scheme with anonymity in global mobility networks," in *Proc. 31st Int. Conf. Adv. Inf. Netw. Appl. Workshops (WAINA)*, Mar. 2017, pp. 19–24.

[18] C.-C. Chang, C.-Y. Lee, and Y.-C. Chiu, "Enhanced authentication scheme with anonymity for roaming service in global mobility networks," *Comput. Commun.*, vol. 32, no. 4, pp. 611–618, Mar. 2009.

[19] D. He, M. Ma, Y. Zhang, C. Chen, and J. Bu, "A strong user authentication scheme with smart cards for wireless communications," *Comput. Commun.*, vol. 34, no. 3, pp. 367–374, Mar. 2011.

[20] Q. Jiang, J. Ma, G. Li, and L. Yang, "An enhanced authentication scheme with privacy preservation for roaming service in global mobility networks," *Wireless Pers. Commun.*, vol. 68, no. 4, pp. 1477–1491, Feb. 2012.

[21] Y.-F. Chang, W.-L. Tai, and M.-H. Hsu, "A secure mobility network authentication scheme ensuring user anonymity," *Symmetry*, vol. 9, no. 12, p. 307, Dec. 2017.

[22] D. Zhao, H. Peng, L. Li, and Y. Yang, "A secure and effective anonymous authentication scheme for roaming service in global mobility networks," *Wireless Pers. Commun.*, vol. 78, no. 1, pp. 247–269, Apr. 2014.

[23] P. Gope and T. Hwang, "Enhanced secure mutual authentication and key agreement scheme preserving user anonymity in global mobile networks," *Wireless Pers. Commun.*, vol. 82, no. 4, pp. 2231–2245, Feb. 2015.

[24] S. K. Sood, "An improved and secure smart card based dynamic identity authentication protocol," *Int. J. Network Security*, vol. 14, pp. 39–46, Jan. 2012.

[25] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," in *Proc. IEEE Comput. Soc. Symp. Res. Secur. Privacy*, May 1992, pp. 72–84.

[26] H.-M. Sun, "An efficient remote use authentication scheme using smart cards," *IEEE Trans. Consum. Electron.*, vol. 46, no. 4, pp. 958–961, Nov. 2000.

[27] T.-F. Lee, "An efficient dynamic id-based user authentication scheme using smart cards without verifier tables," *Appl. Math. Inf. Sci.* vol. 9, pp. 485–490, Jan. 2015.

[28] A. Kumar, and H. Om, "An improved and secure multiserver authentication scheme based on biometrics and smartcard," *Digit. Commun. Netw.*, vol. 4, no. 1, pp. 27–38, 2018.

[29] C.-T. Li, C.-C. Lee, C.-Y. Weng, and C.-I. Fan, "A secure dynamic identity based authentication protocol with smart cards for multi-server architecture," *J. Inf. Sci. Eng.*, vol. 31, pp. 1975–1992, Nov. 2015.

[30] M. Hendry, *Multi-Application Smart Cards: Technology and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[31] E. Barker, "Recommendation for key management," NIST, Gaithersburg, MA, USA, Tech. Rep. 800-57, 2016, vol. 1.

[32] M. Burrows and M. R. A. Needham, "Logic of authentication," *Oper. Syst. Rev.*, vol. 23, no. 5, pp. 1–13, 1989.

[33] L. Buttyan and S. U. S. Wilhelm, "A simple logic for authentication protocol design," in *Proc. 11th IEEE Comput. Secur. Found. Workshop*, Jun. 1998, pp. 153–162.

[34] H. K. Aslan, "Logical analysis of AUTHMAC_DH: A new protocol for authentication and key distribution," *Comput. Secur.*, vol. 23, no. 4, pp. 290–299, Jun. 2004.

[35] T. A. Team, "Automated validation of Internet security protocols and applications," Eur. Community, Brussels, Belgium, Tech. Rep. IST-2001-39252, 2006.

[36] D. von Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in *Proc. APPSEM Workshop*, Sep. 2005, pp. 1–7.

[37] I. Nongbri, P. Hadem, and S. Chettri, "A survey on single sign-on," *Int. J. Creative Res. Thoughts*, vol. 6, no. 2, pp. 595–602, Apr. 2018.

[38] S. Banerjee, V. Odelu, A. K. Das, S. Chattopadhyay, N. Kumar, Y. Park, and S. Tanwar, "Design of an anonymity-preserving group formation based authentication protocol in global mobility networks," *IEEE Access*, vol. 6, pp. 20673–20693, 2018.

**CHIEN-LUNG HSU** received the M.S. and Ph.D. degrees in information management from the National Taiwan University of Science and Technology, in 1997 and 2002, respectively. He is currently a Joint Professor with the Department of Information Management, and Graduate Institute of Business and Management, Chang Gung University, Taiwan. His research interests include smart home, mobile commence, computer and communication security, information security, applied cryptography, healthcare, digital right management, auto identification technology, and user centered service. He received lots of honors, awards, and certificates in term of information security in his research, and has a great number of publications in the related fields. He is also a member of the Institute of Information and Computing Machinery (IICM) and the Chinese Cryptology and Information Security Association (CCISA).

**TUAN-VINH LE** received the M.S. degree in business administration from the Department of Business Administration, National Formosa University, Taiwan. He is currently pursuing the Ph.D. degree with the Graduate Institute of Business and Management, Chang Gung University, Taiwan. His current research interests include information security, applied cryptography, secure networks, and blockchain-enabled data protection systems. As an international student, he was granted a Full Scholarship for his M.S. degree in business administration from National Formosa University, in 2014, and a Full Scholarship for his Ph.D. degree in business and management from Chang Gung University, in 2016.

**CHUNG-FU LU** received the B.S. and M.S. degrees in electrical engineering and the Ph.D. degree in information management from the National Taiwan University of Science and Technology, Taiwan, in 1991, 1993, and 2011, respectively. He was an Associate Professor and the Chair of the Information Management Department, Chihlee University of Technology, Taiwan, from August 2016 to August 2018. His current research includes cryptography, information security, network security, wireless sensor networks, smart home systems, and mobile commerce.

**TZU-WEI LIN** received the B.S. and M.S. degrees in information management from Chang Gung University (CGU), Taiwan, in 2011 and 2013, respectively, where he is currently pursuing the Ph.D. degree with the Graduate Institute of Business and Management. He worked at the Information Security Group, Information Technology Services, Academia Sinica, Taiwan, from 2013 to 2016. His research interests are computer and communication security, information security, applied cryptography, the Internet of Things, and wearable healthcare systems. He received best student paper awards from CISC 2018.

**TZU-HSIEN CHUANG** received the B.S. and M.S. degrees in information management from Chang Gung University, Taiwan, in 2015 and 2017, respectively. He is currently working with the Institute for Information Industry. His research interests are information security and applied cryptography.

• • •