ITC 2/46

Journal of Information Technology and Control
Vol. 46 / No. 2 / 2017
pp. 235-245
DOI 10.5755/j01.itc.46.2.13781

Anonymous and Authentication Protocol for Multi-Server

Received 2015/12/05 | Accepted after revision 2017/05/03

http://dx.doi.org/10.5755/j01.itc.46.2.13781

Anonymous and Authentication Protocol for Multi-Server

Wen-Chung Kuo, Po-Wei Shih

Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, e-mail: simonkuo@yuntech.edu.tw

Yu-Chih Huang

Department of Information Management, Tainan University of Technology, Taiwan, R.O.C.

Lih-Chyau Wuu

Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology

Corresponding author: simonkuo@yuntech.edu.tw

In a multi-server environment, when a user wants to login to a different server to access services, he/she needs to register another user identity and password. Recently, the single sign-on authentication method has been proposed. The major characteristic of this method is that a user only needs to remember one identity and password which can login to different servers. This reduces user inconvenience and server resource usage. On the other hand, anonymity is an important issue. If a user's identity is disclosed, an attacker can trace or masquerade as the user to login servers. Preventing disclosure of the user's identity is very important. In this paper, we will propose an anonymous and single sign-on authentication scheme based on Lagrange interpolating polynomial for a multi-server environment. According to our security analysis, this proposed scheme maintains anonymity, provides mutual authentication and also resists many attacks such as lost smart card, insider attack and replay attack.

KEYWORDS: Anonymous, mutual authentication, memory consumes, Lagrange interpolating polynomial, replay attack.

Introduction

In recent years, people have used modern technologies such as e-mail, Twitter, Facebook, etc. to communicate to each other over the Internet. However, users should use different identities and passwords for different services. Naturally, it is difficult to remember many different identities and passwords. In

This paper is an extended version of our paper published in BAI2014[6] order to alleviate this problem, many authentication schemes based on different methodologies such as password [3, 4], smart card [2, 8, 9], or one-way hash function [7] have been proposed. In 2003, Lee *et al.* [3] proposed password authentication scheme (LKH- scheme). The major contribution of this scheme is that two points are used to generate one linear equation and then the characteristics of equation are used to authenticate the user.

The use of linear equations to replace traditional encryption can reduce com- puting cost and transmission quantity for authentication. Therefore, in 2010, Liaw *et al.* [4] proposed an efficient password authentication scheme based on a geometric approach for multi-server environments. Different from the above, their proposed authentication scheme uses a 2-dimensional plane and 3-dimensional space coordi- nates to achieve authentication. Although these schemes [3, 4] can achieve mutual authentication between user and server, they cannot provide user anonymity on a public network.

In order to achieve user anonymity, we propose an anonymous and single sign-on authentication scheme based on Lagrange interpolating polynomial for multi-server environments in this paper. In other words, a user only needs to remember one identity and password to be able to login to different servers. Each server does not save the user's login information during authentication phase. This can reduce the usage of server resources. According to the security analysis, user anonymity is an important issue. If a user identity is disclosed, an attacker can trace or masquerade as the user to gain access to servers. Thus, we use linear equations to hide user identity and use the Lagrange interpolating polynomial to authenticate the user and server identities. Additionally, our scheme can address lost smart cards, resist replay attack and provide mutual authentication.

The rest of the paper is organized as follows: in Section 2, we review the Lagrange interpolating polynomial and Lee et al.'s scheme. Next, we propose an anonymous and single sign-on protocol for multi-server environments and then discuss security analysis in Sections 3 and 4, respectively. Finally, we draw some conclusions in Section 5.

Related work review

The Lagrange interpolating polynomial

The Lagrange interpolating polynomial [10] is the polynomial $P(x)$ of degree $\neq (n-1)$ that passes through n points, and is given by Eq.(1),

$$P_j(x) = y_j \prod_{k=1,k\neq j}^{n} \frac{x-x_k}{x_j-x_k} \quad (1)$$

where is the Lagrange polynomial, and x_j and y_j are the x-value and y-value of point (x_j, y_j), respectively. In our scheme, we use the Lagrange interpolating polynomial on modulus system.

The password authentication scheme

In 2003, a password authentication scheme based on a geometric approach was proposed by Lee et al. [3]. The major contribution of the LKH-scheme is not having to remember different identities and passwords for various servers. There are four phases in the LKH-scheme: (1) the registration phase, (2) the login phase, (3) the authentication phase and (4) password key update. The notations and symbols used in the LKH-scheme are shown in Table 1.

Table 1

Notations and symbols

Notation	Definition
U_i	The ith user
$\Gamma = \{S1, S2, \ldots, Sm\}$	A set of servers that U_i would like to login to
ID_i	Unique identity of U_i
PW_i	Password of U_i
(x_j, y_j)	The secret points of server S_j
T	Timestamp
L	Linear equation
$\oplus$	XOR operation
$h(\cdot)$	One-way hash function

The registration phase

Step LKH-R1. $U_i \rightarrow$Trusted Manager: ID_i, PW_i

In this step, U_i sends his ID_i and PW_i to the trusted manager.

Step LKH-R2. Trusted Manager $\rightarrow U_i$: $\{ID_i, (C_{ij}, D_{ij})\}$

After receiving the message from U_i, the trusted manager calculates (X_{ij}, Y_{ij}) and (C_{ij}, D_{ij}) where $X_{ij} = h(ID_i \oplus x_j)$, $Y_{ij} = h(ID_i \oplus y_j)$, $C_{ij} = X_{ij} \oplus PW_i$ and $D_{ij} = Y_{ij} \oplus PW_i$. Then, the trusted manager stores the message $\{ID_i, (C_{ij}, D_{ij})\}$ in a smart card and gives it to U_i.

The login phase

Step LKH-L1. $U_i \rightarrow$Reader: PW_i

In this step, U_i inserts the smart card into a reader and enters PW_i to obtain (X_{ij}, Y_{ij}) by computing $X_{ij} = C_{ij} \oplus PW_i$ and $Y_{ij} = D_{ij} \oplus PW_i$.

Step LKH-L2. Reader $\rightarrow U_i$: $\{A_{ij}, B_{ij}, R_i, T_i\}$

Smart card generates two random numbers α_i and β_i. U_i utilizes (α_i, β_i) and (X_{ij}, Y_{ij}) to generate $L_{ij}: y = f_{ij}(x) = a_{ij}x + b_{ij} \bmod p$ and calculates $A_{ij} = a_{ij} \oplus X_{ij}$, $B_{ij} = b_{ij} \oplus Y_{ij}$ and $R_i = h(ID_i||a_{ij}||b_{ij}||T_i))$, where T_i is the timestamp of the U_i.

Step LKH-L3. $U_i \rightarrow S_j: \{ID_i, A_{ij}, B_{ij}, R_i, T_i\}$

U_i forwards $\{ID_i, A_{ij}, B_{ij}, R_i, T_i\}$ to S_j.

The authentication phase

Step LKH-A1. $U_i \rightarrow S_j: \{ID_i, A_{ij}, B_{ij}, R_i, T_i\}$

After receiving the message from U_i, S_j checks ID_i and verifies whether $|T - T_i| \neq \Delta T$, where T is the current time on S_j and ΔT is the expected time interval for transmission delay and clock offset error. If the time is within the expected range, S_j uses secret points $(x_j, y_j)$and ID_i to obtain (X_{ij}, Y_{ij}). Then (a_{ij}, b_{ij}) is recovered by using $a_{ij}=A_{ij}\oplus X_{ij}$ and $b_{ij}=B_{ij}\oplus Y_{ij}$. Then, S_j calculates $R_i = h(ID_i||a_{ij}||b_{ij}||T_i))$ and checks whether R'_i is equal to R_i. If $R'_i=R_i$, then $L_{ij}=f_{ij}(x)=a_{ij}x+b_{ij} \bmod p$ can be reconstructed. This allows S_j to check whether (X_{ij}, Y_{ij}) is located on line L_{ij}.

Step LKH-A2. $S_j \rightarrow U_i: \{SID_j, R_j, T_j\}$

If U_i is authenticated, then S_j calculates $R_j = h(SID_j, h(a_{ij}), h(b_{ij}), T_j)$ and forwards $\{SID_j, R_j, T_j\}$ to U_i.

Step LKH-A3. When U_i receives the message from S_j, U_i uses SID_j and T_j to calculate $R'_j = h(SID_j, h(a_{ij}), h(b_{ij}, T_j)$. Then U_i checks whether R'_j is equal to R_j. If so, then U_i and S_j can achieve mutual authentication.

The password update phase

If U_i wants to change his/her password, he/she can enter PW_i into the smart card to obtain (X_{ij}, Y_{ij}). Then, U_i enters a new password PW_{new} to calculate $C''_{ij}=X_{ij} \oplus PW_{new}$ and $D_{ij} = Y_{ij} \oplus PW_{new}$. Finally, the new secret information $\{ID_i, (C''_{ij}, D'_{ij})\}$ is saved into the smart card. Afterwards, U_i can access the system using his/her new password PW_{new}.

Lee *et al.*[3] uses a geometric approach to propose another encryption method to reduce computing cost and memory use. However, their method cannot provide user anonymity. We extend their contribution by incorporating anonymity in our scheme and propose an anonymous authentication protocol in the next section.

The Proposed Scheme

This section proposes an anonymous authentication scheme for a multi-server environment. We assume that there are three entities in this scheme: registration center (RC), user (U), and server (S) in this scheme. The scheme includes the following four phases: (1) the registration phase; (2) the login phase; (3) the authentication phase and (4) the session key update phase. It is assumed that time is synchronous and information from smart card cannot be obtained through forced attacks. The notations and symbols used in the proposed scheme are shown in Table 2.

Table 2

Notations and symbols in our proposed scheme

Notation	Definition
U_i	User *i*
S_j	Server *j*
RC	Registration center
CID_i	Anonymous identity of U_i
ID_i	Unique identity of U_i
PW_i	Password of U_i
SID_j	Unique identity of S_j
$(x_1, y_1), (x_2, y_2)$	Registration center secret points
T	Timestamp
SK	Session key
L	Linear equation
$P(x)$	Lagrange polynomial
n, p	Two large primes ($n > p$)
N	Random nonce
$h(\cdot)$	One-way hash function

The registration phase

Preceding the registration phase, RC randomly chooses a 2D point (x_1,y_1) and a Lagrange polynomial $P(x)$: $y = ix^2 + jx + k \bmod n$ and then determines a point (x_2,y_2) so that $y_2 = ix^2 + jx_2 + k \bmod n$.

The user registration subphase

The user registration interaction sequence is depicted in Fig.1. An explanation of various messages and steps in this subphase is given below.

Step RU1. $U_i \rightarrow RC: ID_i, PW_i$

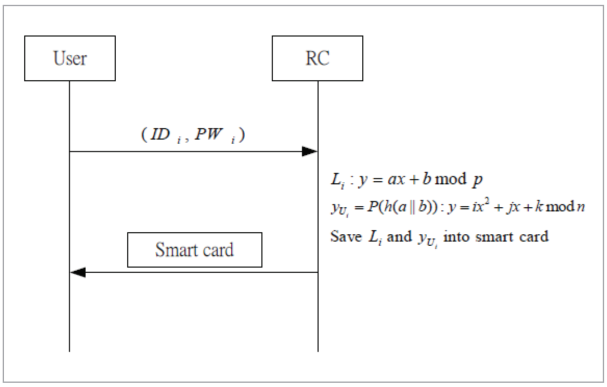
To register, U_i selects a password PW_i and then sends ID_i and PW_i to RC through a secure channel.

Step RU2. $RC \rightarrow U_i: L_i:\ y = ax + b \bmod p$ and y_{U_i}.

After receiving a registration request from U_i, RC uses (x_1, y_1) and (ID_i, PW_i) to generate linear equation $L_i: y = ax + b \bmod p$ and calculates hash value $h(a//b)$. Then, the hash value $h(a//b)$ is substituted into the Lagrange polynomial $P(x): y = ix^2 + jx + k \bmod n$ to generate the corresponding value y_{U_i}. Next, RC saves $h(a//b)$ and y_{U_i} in its database. Finally, linear equation L_i and value y_{U_i} are embedded into the smart card and sent to U_i through a secure channel.

Figure 1

Interaction sequence of the user registration subphase



The server registration subphase

The server registration interaction sequence is depicted in Fig.2. An explanation of various messages and steps in this subphase is provided below.

Step RS1. $S_j \rightarrow RC: SID_j, N_j$

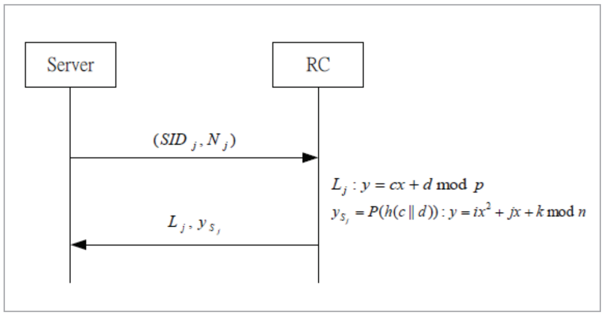
S_j selects a random nonce N_j and then sends SID_j and N_j to RC through a secure channel.

Step RS2. $RC \rightarrow S_j: L_j: y = cx + d \bmod p$ and y_{S_j}.

After receiving the registration request from S_j, RC uses (x_1, y_1) and (SID_j, N_j) to generate linear equation $L_j: y = cx + d \bmod p$ and calculates a hash value $h(c||d)$. Then, RC substitutes $h(c||d)$ into the Lagrange polynomial $P(x): y = ix2 + jx + k \bmod n$ to generate corresponding value y_{S_j}, i.e. $y_{S_j} = P(h(c||d))$. Finally, RC saves $(SID_j, N_j, h(c||d))$ and y_{S_j} in its database and sends L_j and y_{S_j} to S_j through a secure channel.

Figure 2

Interaction sequence of the server registration subphase



The login phase

When U_i wants to login to S_j, he inserts his smart card and enters his identity ID_i and password PW_i.

Step L1. The smart card checks (ID_i, PW_i) if the linear equation function $L_i: y = ax + b \bmod p$, i.e. $PW_i = a * ID_i + b \bmod p$. If it is correct, then it generates a random point (x_i, y_i) that is on the linear equation L_i and a random nonce N_i to calculate $CID_i = h(a||b||N_i||T_i)$ and $M_1 = a \oplus b \oplus N_i$. Otherwise, the smart card rejects the request.

Step L2. U_i forwards $\{CID_i, (x_i, y_i), M_1, T_i\}$ to S_j, where T_i is the timestamp of the U_i.

The authentication phase

The authentication phase interaction sequence is depicted in Fig.3. Following is an explanation of various messages and steps in this phase. When Sj receives the messages $\{CID_i, (x_i, y_i), M_1, T_i\}$ from U_i, the following steps are executed.

Step A1. $S_j \rightarrow RC: \{CID_i, (x_i, y_i), M_1, T_i, SID_j, (x_j, y_j)\}$

To prevent a replay attack, S_j ensures $|T - T_i|$ is not greater than ΔT, where T is the current time on S_j and ΔT is the expected time interval for transmission delay and clock offset error. If it is not correct, S_j terminates the session. Otherwise, S_j produces a random point (x_j, y_j) on the linear equation $L_j : y = cx+d \bmod p$, i.e., $y_j = cx_j + d \bmod p$ and forwards $\{CID_i, (x_i, y_i), M_1, T_i, SID_j, (x_j, y_j)\}$ to RC.

Step A2. $RC \rightarrow S_j : \{V_1, V_2, M_2, (x_{RC}, y_{RC}), T_{RC}\}$

When RC receives message $\{CID_i, (x_i, y_i), M_1, T_i, SID_j, (x_j, y_j)\}$ from S_j, RC performs the following steps.

1 RC checks whether $|T - T_i| \neq \Delta T$. If it is true, RC then uses SID_j to find the corresponding random nonce N_j from the database.

2 RC utilizes the point (x_1, y_1) and (SID_j, N_j) to generate $L_j^*: y = cx + d \bmod p$.

3 RC checks if (x_j, y_j) exists in L_j. If it does not exist, then RC terminates the session. Otherwise, RC utilizes points (x_j, y_j) and (x_1, y_1) to recover $L_i^*: y = ax + b \bmod p$ and calculate $N_i = a \oplus b \oplus M_1$ and $CID_i^* = h(a||b||N_i||T_i)$.

4 RC checks if CID_i^* is equal to CID_i. If the values are equal, then RC uses L_i and L_j to calculate hash values $h(a||b)$ and $h(c||d)$ to recover the corresponding y_{U_i} and y_{S_j} values.

5 RC uses the three points $(h(a||b), y_{U_i})$, $(h(c||d), y_{S_j})$ and (x_2, y_2) to recover the Lagrange polynomial $P(x) : y = ix^2 + jx + k \bmod n$. If it cannot be recovered, then the request is rejected. Otherwise, RC utilizes $(h(a||b), y_{U_i})$ and $(h(c||d), y_{S_j})$ to generate the linear equation $L_{SK} : y = ex + f \bmod p$ and then selects a random point (x_{RC}, y_{RC}) on L_{SK}.

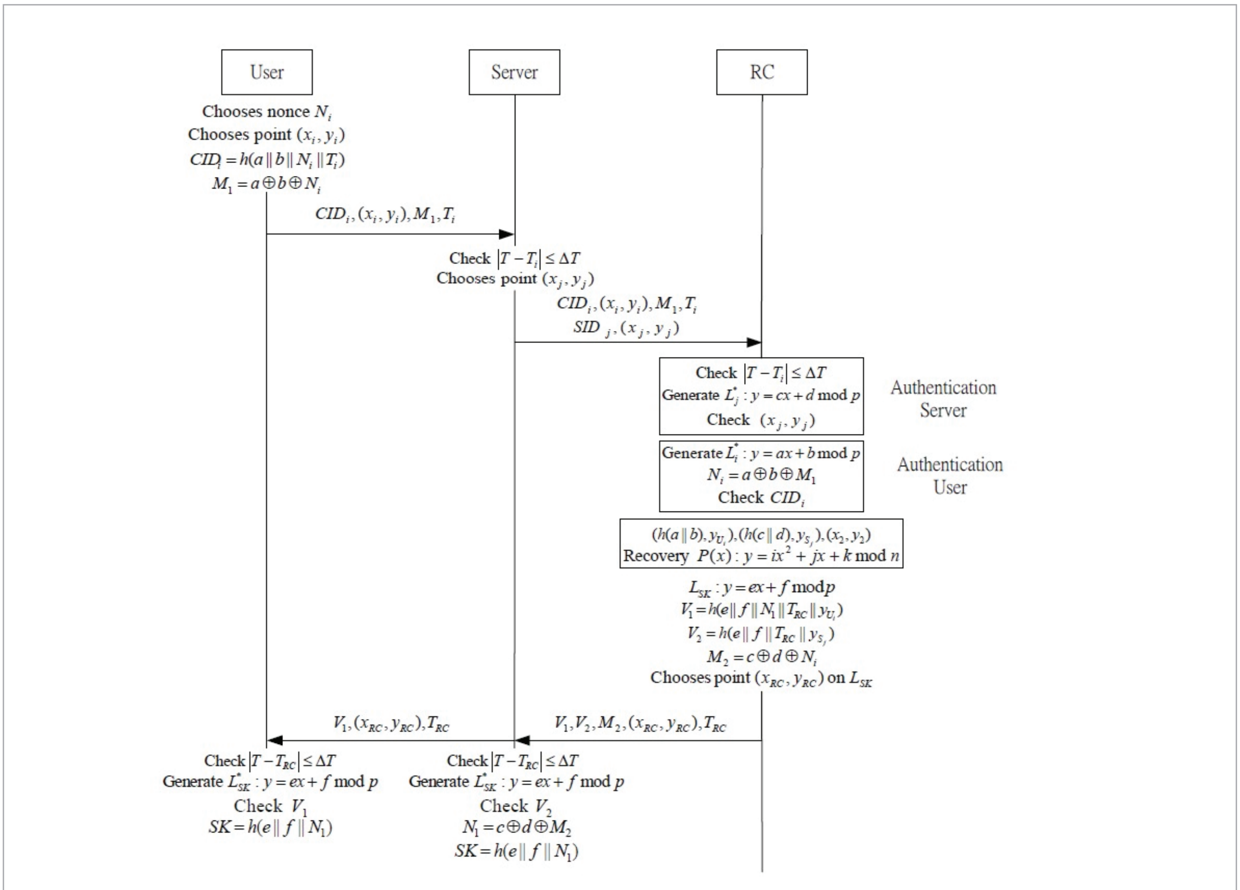
6 RC computes $M_2 = c \oplus d \oplus N_i$, $V_1 = h(e||f||N_i||T_{RC}||y_{U_i})$ and $V_2 = h(e||f||N_i||T_{RC}||y_{S_j})$. Finally, RC sends $\{V_1, V_2, M_2, (x_{RC}, y_{RC}), T_{RC}\}$ to S_j.

Step A3. $S_j \rightarrow U_i : \{V_1, (x_{RC}, y_{RC}), T_{RC}\}$

After receiving the message $\{V_1, V_2, M_2, (x_{RC}, y_{RC}), T_{RC}\}$ from RC, S_j executes the following steps.

Figure 3

Interaction sequence of the authentication phase



S_j uses $(h(c//d), y_{S_j})$ and (x_{RC}, y_{RC}) to generate $L^*_{SK}: y = ex + f \bmod p$ when $|T - T_{RC}| \neq \Delta T$.

S_j checks whether V^*_2 is equal to V_2. If so, then S_j calculates $N_i = c \oplus d \oplus M_2$. S_j calculates the session key $SK = h(e||f||N_i)$ and forwards $\{V_1, (x_{RC}, y_{RC}), T_{RC}\}$ to U_i.

Step A4. When U_i receives $\{V_1, (x_{RC}, y_{RC}), T_{RC}\}$ from S_j, U_i checks whether $|T - T_i| \neq \Delta T$. If they are not equal, U_i uses $(h(a//b), y_{U_i})$ and (x_{RC}, y_{RC}) to recover $L^*_{SK}: y = ex + f \bmod p$ and check whether V^*_1 is equal to V_1. If they are equal, then U_i calculates the session key $SK = h(e||f||N_i)$.

The session key update phase

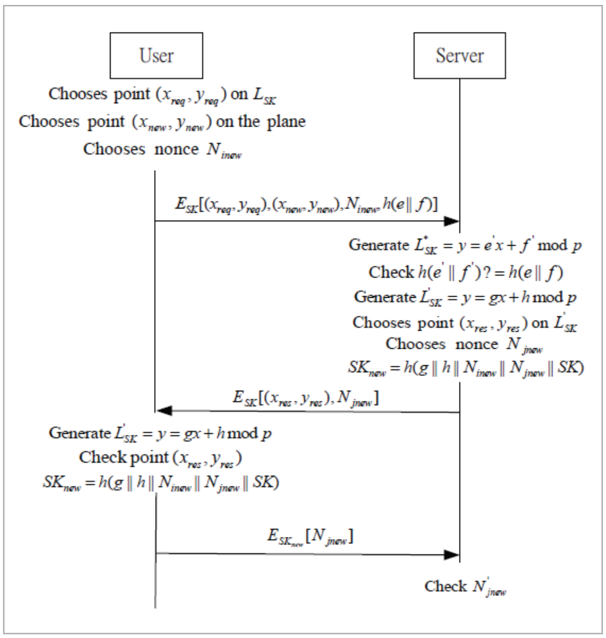
When the user wants to update the session key with the server, he/she can get a new session SK_{new} by using the below listed steps. The session key update interaction sequence is depicted in Fig.4.

Step U1. $U_i \to S_j: E_{SK}[(x_{req}, y_{req}), (x_{new}, y_{new}), N_{inew}, h(e||f)]$

U_i selects two points (x_{req}, y_{req}) and (x_{new}, y_{new}) on a plane where (x_{req}, y_{req}) lies on the linear equation L_{sk}: $y = ex + f \bmod p$. Then, U_i calculates $h(e||f)$ and selects a random nonce N_{inew}. Finally, U_i utilizes session key SK to encrypt the message $\{(x_{req}, y_{req}), (x_{new}, y_{new}), N_{inew}, h(e||f)\}$ and sends it to S_j.

Figure 4

Interaction sequence of the update session key phase



Step U2. $S_j \to U_i: E_{SK}[(x_{res}, y_{res}), N_{jnew}]$

When S_j receives messages $E_{SK}[(x_{req}, y_{req}), (x_{new}, y_{new}), N_{inew}, h(e//f)]$ from U_i, S_j executes the following steps.

1. S_j decrypts message $E_{SK}[(x_{req}, y_{req}), (x_{new}, y_{new}), N_{inew}, h(e//f)]$.
2. S_j uses $(h(c//d), y_{S_j})$ and (x_{req}, y_{req}) to recover $L^*_{SK}: y = e'x + f' \bmod p$.
3. S_j calculates $temp = h(e//f)$ and checks whether $temp$ is equal to $h(e//f)$. If equal, S_j utilizes (x_{req}, y_{req}) and (x_{new}, y_{new}) to generate new linear equation L'_{SK}: $y = gx + h \bmod p$ and new random nonce N_{jnew}.
4. S_j sends message $E_{SK}[(x_{res}, y_{res}), N_{jnew}]$ to U_i, where the point (x_{res}, y_{res}) is on the linear equation L'_{SK}.

Finally, S_j computes the new session key $SK_{new} = h(g||h||N_{inew}||N_{jnew}||SK)$.

Step U3. $U_i \to S_j: E_{SKnew}[N_{jnew}]$

Similarly, U_i will perform the following steps when U_i receives the message $E_{SK}[(x_{res}, y_{res}), N_{jnew}]$ from S_j.

1. U_i decrypts the message and verifies if $y_{res} = g * x_{res} + h \bmod p$ for (x_{res}, y_{res}).
2. If true, U_i generates a new session key $SK_{new} = h(g||h||N_{inew}||N_{jnew}||SK)$ and uses SK_{new} to encrypt N_{jnew} to send it to S_j. Otherwise, the request is dropped.

Step U4. After S_j receives the message $E_{SKnew}[N_{jnew}]$, S_j uses the new session key SK_{new} to recover N'_{jnew} and checks if it is equal to the original N_{jnew}. If it holds, the server updates to the new session key.

Security analysis

In this section, we discuss the security of our scheme against various attacks and use the Burrows-Abadi-Needham Logic (a.k.a BAN Logic) [1] mechanism toprove that the session key can be correctly updated between the user and server in the session key update process.

The authentication phase proof

We use the BAN-logic to show that our scheme correctly updates between the user and the server. Let X and Y represent the range over statements. For the BAN-logic, the logical notations of the logic are given in Table 3.

Furthermore, we also define some logical postulates that we will use in the proofs as follows:

Table 3

The logical notation

Items	Explanation
$U_i \overset{SK}{\longleftrightarrow} S_j$	U_i and S_j share a session key.
$U_i \mid\equiv X$	U_i believes a statement X.
$\#(X)$	X is fresh.
$U_i \Rightarrow X$	U_i controls X.
$U_i \triangleleft X$	U_i receives X.
$U_i \mid\sim X$	U_i sends X.
$[X,Y]_K$	X and Y are encrypted with the key K.

1. The Message-meaning rule

$$\frac{P \triangleleft [X]_K, P \mid\equiv P \overset{SK}{\longleftrightarrow} Q}{P \mid\equiv Q \mid\sim X} \quad (2)$$

If principal P believes that key K is shared with principal Q and P receives message X that is encrypted under K, then principal P believes that principal Q sent the statement X.

2. The Fresh concatenation rule

$$\frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X,Y)} \quad (3)$$

If principal P believes the freshness of statement X, then principal P believes the freshness of (X,Y).

3. The Nonce-verification rule

$$\frac{P \mid\equiv Q \mid\sim X, P \mid\equiv \#(X)}{P \mid\equiv Q \mid\equiv X} \quad (4)$$

If principal P believes that statement X was not stated before and principal Q sent the statement X, then principal P believes that principal Q believes statement X.

4. The Jurisdiction rule

$$\frac{P \mid\equiv Q \mid\equiv X, P \mid\equiv Q \Rightarrow X}{P \mid\equiv X} \quad (5)$$

If principal P believes that statement X is under principal Q's jurisdiction, then principal P believes principal Q on the validity of statement X.

Next, we show that our scheme should satisfy the following requirements:

$$G_1 : RC \mid\equiv (x_i, y_i), (x_j, y_j) \quad (6)$$

$$G_2 : S_j \mid\equiv (x_{RC}, y_{RC}) \quad (7)$$

$$G_3 : U_i \mid\equiv (x_{RC}, y_{RC}) \quad (8)$$

Before we analyze the proposed protocol, we transform to the idealized form and identify the initial state of our scheme. The initial state is assumed to be the following:

$$A_1 : U_i \mid\equiv U_i \overset{y_{Ui}}{\longleftrightarrow} \mathrm{RC} \quad (9)$$

$$A_2 : U_i \mid\equiv \#(T_{RC}) \quad (10)$$

$$A_3 : U_i \mid\equiv RC \Rightarrow (U_i \overset{y_{Ui}}{\longleftrightarrow} \mathrm{RC}) \quad (11)$$

$$A_4 : S_j \mid\equiv S_j \overset{y_{Sj}}{\longleftrightarrow} \mathrm{RC} \quad (12)$$

$$A_5 : S_j \mid\equiv \#(T_{RC}) \quad (13)$$

$$A_6 : S_j \mid\equiv RC \Rightarrow (S_j \overset{y_{Sj}}{\longleftrightarrow} \mathrm{RC}) \quad (14)$$

$$A_7 : RC \mid\equiv S_j \overset{L_j}{\longleftrightarrow} \mathrm{RC} \quad (15)$$

$$A_8 : RC \mid\equiv U_i \overset{L_i}{\longleftrightarrow} \mathrm{RC} \quad (16)$$

$$A_9 : RC \mid\equiv S_j \Rightarrow (S_j \overset{L_j}{\longleftrightarrow} \mathrm{RC}) \quad (17)$$

$$A_{10} : RC \mid\equiv U_i \Rightarrow (U_i \overset{L_i}{\longleftrightarrow} \mathrm{RC}) \quad (18)$$

$$A_{11} : RC \mid\equiv \#(\mathrm{T}_i) \quad (19)$$

Now, we use the initial assumptions and the rules of the BAN-logic to analyze the idealized form of our scheme. The proofs are described as follows:

1 By A_7, A_8 and $RC \triangleleft ((x_i, y_i), (x_j, y_j), T_i)$, we apply the message-meaning rule to derive

$$RC \mid\equiv \#\big((x_i, y_i), (x_j, y_j)\big) \quad (20)$$

2 By A_{11} and Eq.(20), we apply the fresh concatenation rule toderive

$$RC \mid\equiv \#\left((x_i, y_i),(x_j, y_j)\right) \quad (21)$$

3 By Eq.(20) and (21), we apply the nonce-verification rule to derive

$$RC \mid\equiv S_j \mid\equiv \left((x_i, y_i),(x_j, y_j)\right) \quad (22)$$

4 By A_9, A_{10} and Eq.(22), we apply the jurisdiction rule to derive Eq.(6):

$G_1 : RC \mid\equiv (x_i, y_i),(x_j, y_j)$

5 By A_4 and $S_j \lhd \left((x_{RC}, y_{RC}), T_{RC}\right)$, we apply the message-meaning rule toderive

$$S_j \mid\equiv RC \mid\sim (x_{RC}, y_{RC}), T_{RC} \quad (23)$$

6 By A_5 and Eq.(23), we apply the fresh concatenation rule and nonce-verification rule to derive

$$S_j \mid\equiv RC \mid\equiv (x_{RC}, y_{RC}) \quad (24)$$

7 By A_6 and Eq.(24), we apply the jurisdiction rule to derive Eq.(7):

$G_2 : S_j \mid\equiv (x_{RC}, y_{RC})$

8 By A_1 and $U_i \lhd \left((x_{RC}, y_{RC}), T_{RC}\right)$, we apply the message-meaning rule to derive

$$U_i \mid\equiv RC \mid\sim (x_{RC}, y_{RC}), T_{RC} \quad (25)$$

9 By A_2 and Eq.(25), we apply the fresh concatenation rule and nonce-verification rule to derive

$$U_i \mid\equiv RC \mid\equiv (x_{RC}, y_{RC}) \quad (26)$$

10 By A_3 and Eq.(26), we apply the jurisdiction rule to derive Eq.(8).

$$G_3 : U_i \mid\equiv (x_{RC}, y_{RC}) \quad (27)$$

The security analysis

In this subsection, we discuss the resiliency of our proposed scheme against some common attacks such as replay attack, lost or stolen smart card attack, and impersonation attack, while providing mutual authentication and user anonymity.

Resistance to replay attack

Since the transmitted messages CID_i, V_1 and V_2 include timestamps, the server and user can detect a replay attack directly. If an attacker re-submits the intercepted message, the attacker must choose a suitable time interval ΔT and modify these three messages. However, the attacker cannot modify them because he/she does not know L_i and L_{SK}. Note that the replay attack is avoided while the system clock synchronization and transmission delay are accounted for.

Smart card stolen attack

This assumes that the smart card is lost or stolen. If the attacker wants to use this stolen smart card to login, the attacker must know the correct *ID* and *PW* to complete the login phase. Furthermore, the *ID* and *PW* cannot be recovered from the stolen smart card. Therefore, the lost smart card attack is prevented in our proposed scheme.

Resistance impersonation attack

In the authentication phase, an attacker may intercept message $\{CID_i, (x_i, y_i), M_1, T_i\}$ and try to impersonate a legal user to pass authentication. However, the attacker cannot calculate $L_i : y = ax + b \bmod p$, because point (x_i, y_i) is a secret held by the registration center. Therefore, the attacker cannot impersonate the legal user to pass authentication. On the other hand, if the attacker wants to masquerade as a server to spoof the user and intercept message $\{V_1, V_2, M_2, (x_{RC}, y_{RC}), T_{RC}\}$, the attacker must modify V_1, but cannot do so since the attacker does not have y_{U_i}. Additionally, the attacker cannot calculate $L^*_{SK} : y = ex + f \bmod p$ and generate the session key *SK*. Thus, the attacker cannot masquerade as a server to spoof users.

Mutual authentication

In our scheme, *RC* has the important role to authenticate user U_i and server S_j. If the verification operation fails, this means that the credentials are incorrect and possibly the user U_i or server S_j is not legitimate. Therefore, mutual authentication will end. When the registration center sends message $\{V_1, V_2, M_2, (x_{RC}, y_{RC}), T_{RC}\}$ to server S_j and the server verifies V_2 as correct, the server can trust that user U_i is legitimate since the user passed the registration center authentication. Conversely, server S_j sends message $\{V_1, (x_{RC}, y_{RC}), T_{RC}\}$ to user U_i and U_i verifies if V_1 is correct. The user can

Table 4
Security Properties

Properties	LKH-scheme[3]	LYCH-scheme[4]	XHM-scheme[11]	Our scheme
User anonymous	No	No	Yes	Yes
Mutual authentication	Yes	Yes	Yes	Yes
Resist the smart card stolen	Yes	No	Yes	Yes
Security of the session key	No	No	Yes	Yes
Prevent the replay attack	Yes	Yes	Yes	Yes
Prevent the impersonation attack	Yes	Yes	Yes	Yes

trust that the server is legitimate since the authentication message cannot be forged.

User anonymity

In our scheme, the user uses CID_i instead of ID_i so the attacker cannot get the user's real identity. In addition, we use a random nonce N_i to calculate CID_i. This makes CID_i unique every time. Thus, the attacker cannot trace the user's identity. In the authentication phase, the registration center does not use ID_i to search for the corresponding Lagrange polynomial value instead, it uses the hash value $h(a//b)$ to search. Therefore, the user does not send his/her identity over the public network. Hence, the attacker cannot trace the user's real identity by intercepting a message transmitted between U_i and RC.

The security properties of our scheme are compared with other authentication schemes [3, 4, 11]. The results are shown in Table 4. As can be seen from the Table 4, our scheme achieves user anonymity, mutual authentication and the session key security; it also prevents the lost smart card, replay and impersonation attacks.

The performance evaluation

Computational Load

In Table 5, the computational load of our scheme is examined and compared with other authentication schemes [3, 4, 11]. The computation cost of a one-way function is denoted as T_H; T_M is the computation cost of modular exponentiation; T_R is the computation cost of a random nonce, select point and timestamp generation.

Note that we ignore the cost of XOR and //, because these operations require very little computation overhead. To be fair, we will not compare the session key update phase, because other schemes do not have this phase. As can be seen from the Table 5, our scheme has computation cost of 14 hashes, 7 instances of random nonce computing and 10 instances of modular exponentiation in the three phases of registration, login and authentication.

Communication Load

The communication load of our scheme was examined and compared with [11]and the results are shown in Table 6. The hash value length is assumed to be 128-bits, timestamp length is assumed to be 24-bits, the length of the each point value is assumed to be 16-bits, and each of the other elements are assumed to be 128-bits. The bits used for each interaction are added together and then divided by eight to obtain the number of bytes transferred for each interaction.

For example, from the authentication phase, there are four instances of message transmission which are:

$U_i \rightarrow S_j$, $S_j \rightarrow RC$, $RC \rightarrow S_j$ and $S_j \rightarrow U_i$. The results show the total communication load of our scheme is less than Xue's scheme.

Table 5

Computational load

Protocols		Computation cost		
		Registration	Login	Authentication
LKH-scheme[3]	Registration center User	$2TH$ 0	0 $TH + TR$	0 TH
	Server	0	0	$5TH + TR$
	Total	$2TH$	$TH + TR$	$6TH + TR$
	Registration center	$4TH$	0	0
LYCH-scheme[4]	User	0	$3TH + TR$	0
	Server	0	0	$4TH + TR$
	Total	$4TH$	$3TH + TR$	$4TH + TR$
XHM-scheme[11]	Registration center User Server	$4TH$ $3TH + TR$ TR	0 $2TH$ 0	$15TH + 2TR + 4TM$ $7TH + 2TR$ $6TH + TR$
	Total	$7TH + 2TR + 4TM$	$2TH$	$28TH + 4TR$
Proposed protocol	Registration center User Server	$4TM + 2TH$ 0 TR	0 $TH + 3TR$ 0	$5TH + 2TR + 4TM$ $3TH + TM$ $3TH + TR + TM$
	Total	$2TH + TR + 4TM$	$TH + 3TR$	$11TH + 3TR + 6TM$

Table 6

Communication load for the authentication phase

Protocols	Message length (byte)			
	$U_i \to S_j$	$S_j \to RC$	$RC \to S_j$	$S_j \to U_i$
Our protocol	37	55	53	21
XHM-scheme[11]	83	163	64	32

Conclusion

The proposed authentication scheme can achieve user anonymity and provide mu- tual authentication between server and users. In addition, this approach can resist various kinds of attacks. For a multi-server environment, our proposed scheme also provides single sign-on capabilities for users. Therefore, a user does not need to register on many different servers or remember he/she unique identity and pass- word to login to participating servers. Furthermore, we use hash function and modular computing to replace the traditional encryption method which reduces computing cost.

References

1. Burrows, M., Abadi, M., Needham, R. A logic of authentication. ACM Transactions on Computer Systems, 1990, 8(1), 18-36. https://doi.org/10.1145/77648.77649
2. Chuang, M. C., Chen, M. C. An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics. Expert

Systems with Applications, 2014, 41, 1411-1418. https://doi.org/10.1016/j.eswa.2013.08.040

3. Lee, Y. C. Kuo, W. C., Hou, J. M. Password Authentication Scheme with Multi- Servers. NCS, 2003.
4. Liaw, H. T., Yen, C. T., Chiu, M. Y., Hsiao, L. L. Efficient password authentication schemes based on a geometric approach for a multi-server environment. Journal of Zhe- jiang Uniersity Science, 2010, 11(12), 989-997.
5. Mun H. Han, K., Lee, Y. S., Yeun, C. Y., Choi, H. H. Enhanced secure anonymous authentication scheme for roaming service in global mobility networks. Mathematical and Computer Modelling, 2012, 55, 214-222. https://doi.org/10.1016/j.mcm.2011.04.036
6. Shih, P. W., Kuo, W. C., Wuu, L. C., Huang, Y. C. Anonymous and efficient authenti- cation protocol based on Lagrange approach for multi-server environment. International Conference of Business and Information (BAI 2014), 2014.
7. Tsai, J. L. Efficient multi-server authentication scheme based on one-way hash function without verification table. Computers and Security, 2008, 27(3), 115-121. https://doi.org/10.1016/j.cose.2008.04.001
8. Tsai, J. L., Lo, N. W., Wu, T. C. Novel Anonymous Authentication Scheme Using Smart Card. IEEE Transaction on Industrial Informatics, 2013, 9(4), 2004-2013. https://doi.org/10.1109/TII.2012.2230639
9. Tsai, J. L., Lo, N. W. A chaotic map-based anonymous multi-server authenticated key agreement protocol using smart card. International Journal of Communication Systems, 2015, 28(13), 1955-1963. https://doi.org/10.1002/dac.2829
10. Wuu, L. C., Hung, C. H., Chang C. M. Quorum-based Key Management Scheme in Wireless Sensor Networks. KSII Transactions on Internet and Information Systems, 2012, 6(9), 2442-2454. https://doi.org/10.1145/2184751.2184770
11. Xue, K., Hong, P., Ma, C. A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. Journal of Computer and System Sciences, 2014, 80(1), 195-206. https://doi.org/10.1016/j.jcss.2013.07.004
12. Yeh, K. H. An anonymous and lightweight authentication scheme for mobile devices. Information Technology and Control, 2015, 44(2), 206-214.https://doi.org/10.5755/j01.itc.44.2.8335

Summary / Santrauka

In a multi-server environment, when a user wants to login to a different server to access services, he/she needs to register another user identity and password. Recently, the single sign-on authentication method has been proposed. The major characteristic of this method is that a user only needs to remember one identity and password which can login to different servers. This reduces user inconvenience and server resource usage. On the other hand, anonymity is an important issue. If a user's identity is disclosed, an attacker can trace or masquerade as the user to login servers. Preventing disclosure of the user's identity is very important. In this paper, we will propose an anonymous and single sign-on authentication scheme based on Lagrange interpolating polynomial for a multi-server environment. According to our security analysis, this proposed scheme maintains anonymity, provides mutual authentication and also resists many attacks such as lost smart card, insider attack and replay attack.

Sujungtų serverių aplinkoje, vartotojui, norinčiam prisijungti prie skirtingo serverio, kad gautų prieigą prie paslaugų, reikia užregistruoti kitą vartotojo tapatybę ir slaptažodį. Neseniai buvo pasiūlytas SSO identifikavimo metodas, kurio pagrindinė savybė ta, kad vartotojas turi prisiminti tik vieną tapatybę ir slaptažodį, kuriuos naudodamas gali prisijungti prie skirtingų serverių. Taip mažesnis nepatogumas vartotojui ir mažiau naudojama serverio išteklių. Kita vertus, anonimiškumas išlieka svarbiu probleminiu klausimu. Jei vartotojo tapatybė atskleidžiama, atakuotojas gali sekti vartotoją arba juo apsimesti jungdamasis prie serverių. Labai svarbu neleisti atskleisti vartotojo tapatybės. Straipsnyje siūloma anoniminė ir SSO identifikavimo schema, paremta Lagranžo interpoliaciniu daugianariu sujungtų serverių aplinkai. Atlikta saugumo analizė rodo, kad siūloma schema išlaiko anonimiškumą, suteikia bendro identifikavimo galimybę ir gali pasipriešinti tokioms atakoms: prarasta išmanioji kortelė, atakuotojo tinklo viduje ataka ir atkartojimo ataka.