A Novel Authentication Scheme for Multi-Server Environment of Industrial Internet

Yu Zhang¹,²*, Guangmin Sun¹
¹Faculty of Information Technology, Beijing University of Technology, Beijing, China
²Jining University, Shandong, China
*Corresponding Author.

Abstract

Aiming at the security problems of authentication in multi-server environments, a novel three-factor authentication scheme for multi-server environments of industrial Internet is proposed. After verifying password and face, a temporary session key is established for the user and server. Then the user obtains the permission of application services and accessing resources. In process of verifying password, hash function is used to hide password. The method of verifying face is the face recognition based on singular value decomposition. During the key agreement phase, only four dot multiplication operations based on elliptic curve cryptography is used to realize one-time key for cryptograph transmission and mutual authentication. Through security analysis and performance comparison, the proposed scheme has stronger robustness, higher security, better convenience and less computation cost than other similar schemes, and has high application value for multi-server environments of industrial Internet.

Keywords: Multi-server environment, industrial Internet, authentication, singular value decomposition, key agreement

I. Introduction

With the widespread application of distributed systems, the authentication schemes of application services in traditional single-server environment require users to memorize many different passwords. For this reason, people began to propose authentication protocol for the multi-server environment of industrial Internet. The users can access many servers after registering only once. The details are shown in Figure 1.

Aiming at the security problems of authentication in multi-server environments, many multi-factor authentication schemes[1-6] were proposed. However, Yoon’s scheme[1] cannot resist impersonation and other attacks. Preeti’s scheme[2] doesn’t have strong robustness, and each authentication requires server participation; He’s scheme[3] cannot effectively resist impersonation attack. Odelu’s scheme[4] doesn’t have strong robustness, and may suffer insider attack.

![Diagram of Authentication Scheme](image-url)
In general, the schemes for multi-server environment can be classified into five groups in terms of the underlying intractability problem: based on discrete-logarithm-problem\textsuperscript{(7)}, based on pairing\textsuperscript{(8)}, based on chaotic-map\textsuperscript{(9)}, based on integer-factorization-problem\textsuperscript{(10)} and based on hash\textsuperscript{(11)}.

The RC online authentication mode generally leads to increase calculation and communication cost, and reduces the real-time response speed of the system. The RC offline authentication mode is vulnerable to masquerade attack. Aiming at the problems above, a novel scheme is proposed in this paper.

II. Paper contents

Section 3: the related theories. Section: the proposed scheme. Section 5: security analyses and comparison with other similar schemes. Section 6: performance comparison. Section 7: the conclusion is given.

III. Related theory

3.1 Face recognition based on singular value decomposition

Tai et al.\textsuperscript{(12)} propose a novel method named learning discriminative singular value decomposition representation (LDSVDR) for face recognition. The details of the LDSVDR model are as follows.

Letting $k = rank(A)$, face image $A$ can be represented as

\[ A = USV^T = \sum_{i=1}^{k} \lambda_i u_i v_i^T \]  \hspace{1cm} (1)

\[ \text{Vec}(A) = \sum_{i=1}^{k} \lambda_i \text{Vec}(u_i v_i^T) \]  \hspace{1cm} (2)

Where $A \in \mathbb{R}^{m \times n}$, $k$ is the rank of $A$, $k = n$, $\text{Vec}()$ is a function that converts the matrix into vector. The optimal set is $n \times 1$. For each face image, we can always build a basis set as $(\text{Vec}(u_1), \ldots, \text{Vec}(u_n)) \in \mathbb{R}^{mn\times1}$.

The rank of the newly formed representation image needs to introduce a scalar $r_i \leq n$. Eq. (2) can be rewritten as

\[ \text{Vec}(A) = \sum_{i=1}^{r} \lambda_i \text{Vec}(u_i v_i^T) = \sum_{i=1}^{r} \lambda_i \text{Vec}(u_i v_i^T) \]  \hspace{1cm} (3)

Where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r > 0 = \lambda_{r+1} = \ldots = \lambda_n = 0$.

Letting $\xi_i = \text{Vec}(u_i v_i^T) \in \mathbb{R}^{mn\times1}$. Eq. (3) can be rewritten as

\[ \text{Vec}(A) = \sum_{i=1}^{r} \lambda_i \xi_i = (\xi_1, \xi_2, \ldots, \xi_r) \left( \lambda_1, \lambda_2, \ldots, \lambda_r \right)^T \in \mathbb{R}^{r\times1} \]  \hspace{1cm} (4)

where $\Psi = (\xi_1, \xi_2, \ldots, \xi_n) \in \mathbb{R}^{mn\times1}$ and $\Phi = (\lambda_1, \lambda_2, \ldots, \lambda_r)^T \in \mathbb{R}^{r\times1}$. $\Psi$ is the newly constructed basis set and $\Phi$ is the coefficient set of $\Psi$.

The next step of the method is to seek the most discriminant weight vector $\Phi = (\lambda_1^+, \lambda_2^+, \ldots, \lambda_r^+)^T$, $\lambda_1^+ \geq \lambda_2^+ \geq \ldots \geq \lambda_r^+ > 0$, for constructing a new representation image $B$ of the original image $A$, which can be written as
Then, the optimal solution can be gotten through solving several quadratic programming subproblems.

3.2 Elliptic curve public key cryptography

Because of the lower computational cost of an elliptic curve point multiplication operation, researchers began to pay attention to protocols based on point multiplication operation.

In 2017, Wang\textsuperscript{13} presented a key agreement protocol by making use of operation amount is four. In 2019, Sun\textsuperscript{14} presented Wang’s protocol could not resist the temporary secret key leakage attack, proposed an improved protocol. This improved protocol is efficient with four operations of point multiplication in key agreement phase.

IV. The proposed scheme

4.1 Notations description

| Symbol | Description |
|--------|-------------|
| RC | Registration center |
| KGC | Key generate center |
| $G$ | Additive cyclic group |
| $q$ | The order of additive cyclic group $G$ |
| $p$ | A large prime number (k-bit) |
| $P$ | a generator of $G$ in finite field $F_p$ |
| $s$ | Master private key |
| $P_{sys}$ | Master public key |
| $H_{i}()$ | Hash function |
| $s_{j}$ | Application server |
| $SID_{j}$ | The unique identity of application server $s_{j}$ |
| $t_{i}, t_{j}$ | The current time |
| $\Delta t$ | The effective time interval |
| $r_{i}, r_{i}^{\alpha}, r_{i}^{\beta}$ | Random number |
| $d_{i}$ | Private key |
| $P_{i}$ | Public key |
| $U_{i}$ | user |
| $ID_{i}$ | The unique identity of user $U_{i}$ |
| $pw_{i}$ | Password of user |
| $B_{i}$, $B_{i}^{\alpha}$ | The biometric data of the user's face image |
| $\oplus$ | Exclusive OR (XOR) |
| $||$ | Concatenation operation |
| $\Psi_{i}$ | The SVD basis set of a face image |
4.2 Framework of the proposed scheme

The authentication system includes three parts: authentication server (including RC), mobile terminal and application server.

Registration center (RC): including Key generate center (KGC) module, key data management module, user authority management module.

Mobile terminal: including registration module (including functions such as storing and locking data), face recognition and verification module, password verification module, encryption and decryption module, key agreement module, application interface module.

Application server: including registration module, encryption and decryption module, key agreement module, application platform interface module.

The scheme is shown in Figure 2.

4.3 The proposed scheme

4.3.1 System initialization
KGC firstly selects the security parameter $k$, then structure $(E/F_r,G,q,P)$, where $P$ is a k-bit large prime number, $E/F_r$ is an elliptic curve of finite field $F_r$, $P$ is a generator of $F_r$, $G$ is an additive cyclic group of $F_r$, $Q$ is the order of $G$.

KGC chooses a random element $s$ from group $Z_2^*$ as the master private key, computes $P_{\alpha} = sP$ (is non-infinity point) and regards the computational result as the public master key, and selects two hash function.

\[
H_1: [0,1]^* \times G \rightarrow Z_2^*,
\]

\[
H_2: [0,1]^* \times [0,1]^* \times [0,1]^* \rightarrow [0,1]^*;
\]

KGC preserves the master private key $s$ secretly, publishes $(E/F_r,G,q,P_{\alpha},H_1,H_2)$.

\[
\text{Server} \quad \{\text{SID}_i,t_i\} \quad \text{RC}
\]

\[
\text{Publishes} \quad \{\text{SID}_i,P_i,d_i,t_i,h_i,t_j\}
\]

\[
\text{Chooses} \quad r_j \quad \text{computes} \quad R_j = r_jP_i
\]

\[
b_h = H_i(\text{SID}_i,\{R_j\})
\]

\[
d_j = r_j + h_i
\]

\[
\text{and} \quad P'_h = R_j + h_{P_{\alpha}}.
\]

Fig. 3. Application server registration phase

4.3.2 Registration phase

(1) Application server registration phase (as shown in Fig. 3)

Step 1. Server $S_i$ selects the unique identity $\text{SID}_i$, then sends $\text{SID}_i$ and $t_i$ to RC via secure channel.

Step 2. RC receives the requesting registration information $\{\text{SID}_i,t_i\}$ from $S_i$, and checks firstly if the value of the timestamp $t_i$ exceeds the allowed maximum range $\Delta t$ or not. RC refuses repetitive registration request from the same one within the time interval $\Delta t$. If $\Delta t$ meet the requirement, RC checks whether the identity $\text{SID}_i$ has already been registered, in order to avoid the same identity for different server's registration. If the result holds, KGC chooses a random element $r_j$ from group $Z_2^*$ for server $S_i$, computes $R_j = r_jP_i$, $b_h = H_i(\text{SID}_i,\{R_j\})$, privet key $d_j = r_j + h_i$ and public key $P'_h = R_j + h_{P_{\alpha}}$. Then RC stores $\{\text{SID}_i,P_i,d_j,t_i,h_j,t_j\}$ in the key data management module, and sends the data above to application server $S_i$ via security channel.

Step 3. After receiving $\{\text{SID}_i,P_i,d_j,t_i,h_j,t_j\}$, application server $S_i$ ensures the security of $d_j$, $r_j$, $h_j$, and $t_j$. Then the server $S_i$, publishes own identity $\text{SID}_i$, and public key $P_i$.

(2) User registration phase (as shown in Fig. 4)

\[
\text{User} \quad \text{Selects} \quad D_r \quad \text{pw}, \text{and} \quad r_{\text{en}}, \text{obtains faceted data} \quad B^{\text{en}}, \quad \text{computes} \quad M_{\text{en}} = \{D_r, \text{pw}, B^{\text{en}}, c_{\text{en}}\}
\]

\[
C_i = [c_{\text{en}}]P = (x_i,y_i), Q_i = [c_{\text{en}}]S_{\text{en}} = (x_i,y_i).
\]

\[
t_i = H_i(\text{SID}_i,\{Q_i\}), C_i = M_{\text{en}} \otimes t_i.
\]

\[
C_i = H_i(\text{SID}_i,\{Q_i\}) \otimes M_{\text{en}}, \text{extracts} \quad t_i.
\]

\[
C = \{t_i| ID_i, C_i | C_i | C_i\}
\]

\[
\text{Computes} \quad PW_{\text{en}} = H_i(\text{pw} | ID_i).
\]

\[
M_i^{\text{en}} = PW_{\text{en}} \otimes (H_i(\text{pw} | c_{\text{en}}) \otimes M_{\text{en}}).
\]

\[
M_i^{\text{en}} = \{t_i, R_i, h_i, PW, PW_{\text{en}}, d_i, P_i, \text{pw}, \Phi_i\}
\]

\[
\text{RC} \quad \text{Decrypts the ciphertext} \quad C
\]

\[
\text{gets} \quad D_r, \text{pw}, B^{\text{en}}, t_j,
\]

\[
\text{computes} \quad \text{Vec}(B^{\text{en}}) = \Psi_i \Phi_i,
\]

\[
\Psi_i = (\xi_i, \xi_i^2, \ldots, \xi_i^k) \in R^{k \times m},
\]

\[
\Phi_i = (c_{\text{en}}, c_{\text{en}}^2, \ldots, c_{\text{en}}^k) \in R^{k \times m},
\]

\[
r_j = H_i(\text{SID}_i,\{Q_i\} \otimes \text{en} \otimes \ldots \otimes \text{en} \otimes \text{pw} | k).
\]

\[
R_j = r_j + h_j.
\]

\[
P'_h = R_j + h_{P_{\alpha}}.
\]

\[
M_i^{\text{en}} = \{t_i, R_i, h_i, PW, PW_{\text{en}}, d_i, P_i, \text{pw}, \Phi_i\}
\]

\[
M_i^{\text{en}} = H_i(\text{pw} | c_{\text{en}}) \otimes (PW_{\text{en}} \otimes M_i^{\text{en}}).
\]

\[
M = \{M_i^{\text{en}} | H_i (M_i^{\text{en}}) | t_j\}
\]

Fig. 4. User registration phase
Step 1. User $U$, chooses the unique identity $ID$, and passwords $pw$, then cooperates with the face recognition and verification module to use the camera of the terminal device to get face image in accordance with the requirements for image collection. After face detection, the registration module obtains the data $B''$ of user’s face image, chooses a random element $r''$ from group $Z'_r$ and composes registration information $M''=\{ID, pw, B'', r''\}$.

Step 2. The encryption and decryption module of mobile terminal encrypts the registration information $M''=\{ID, pw, B'', r''\}$ with the master public key of RC. Firstly, the module above computes point $C_i=[x''|P=(x, y)]$ of the elliptic curve $E/F$, and point $Q=[x''|\mathbb{Z}_m=(x, y)]$, then converts data type of $x_2$ and $y_2$ to bit string if the point $Q$ is non-infinity point, otherwise choosing $r''$ again. Then the module above computes $t=H(x_2 \cdot y_2)$. If the bit string of $t$ is all zeros, the random element $r''$ must be chosen again. Then the module above computes $C_i=M'' \oplus t$ and $C_i=H(x_2 \cdot y_2 \cdot M'')$. After extracting current time $t$, the user $U$, sends the message $C=[t, ID, C_i, C_2, C_3]$ to RC via public network.

Step 3. After receiving $C=[t, ID, C_i, C_2, C_3]$, RC checks firstly if the timestamp $t$ exceeds the allowed maximum range $\Delta t$ or not. RC refuses to repetitive registration request from the same one within the time interval $\Delta t$. If the result holds, in order to avoid the same identity for different server’s registration, RC checks whether the identity $ID$ has already been registered. If the user’s identity $ID$ has already been registered, RC examines the latest time record of the user’s registration and refuses repetitive registration request from the same one within the allowed time. If it meets the requirements, RC takes out the bit string of $C_i$ from the message $C$, converts the data type of the bit string to the point of elliptic curve. Then RC examines whether the point from $C_i$ satisfies the elliptic curve. If the result doesn’t hold, RC reports an error and exits. If the result holds, RC computes elliptic curve point $Q'=rC_i=(x', y')$ and examines whether the point $Q'$ is the infinity point. If the result doesn’t hold, RC reports an error and exits. If the point $Q'$ is non-infinity point, RC converts data type of $x'_2$ and $y'_2$ to bit string. Then RC computes $t'=H(x'_2 \cdot y'_2)$ and examines whether the result $t'$ is the bit string of all zero. If $t'$ is all zero, RC reports an error and exits. If the result holds, RC takes out $C_i$ from the message $C$, computes $M''''=C_i \oplus t'$ and $C_i=H(x'_2 \cdot y'_2 \cdot M'')$. Then RC takes out $C_i$ from the message $C$ and compares $C_i$ with $C_i$. If $C_i \neq C_i$, RC reports an error and exits. If $C_i=C_i$, RC confirms $M''''=M''''=\{ID, pw, B'', t\}$, decrypts the ciphertext into plaintext and gets user’s information $ID, pw, B'', t$.

Step 4. After getting the user’s face image data $B''$, RC computes $\Psi(B'')=\Psi \Phi_0$ with singular value decomposition, where $\Psi=(\xi_1, \xi_2, \ldots, \xi_n) \in \mathbb{R}^{n \times m}$ and $\Phi_0=(\lambda_1^2, \lambda_2^2, \ldots, \lambda_n^2) \in \mathbb{R}^{m \times m}$. $\Phi_0$ is the singular value base set of the face image data $B''$ of user $U$. $\Phi_0$ is the coefficient of $\Psi$.

### Table 2 User’s permission table

| Identity | Status         | Class of identity | Authority            | User’s information         |
|----------|----------------|-------------------|----------------------|----------------------------|
| *****01  | valid          | System Administrator | All authorized       | $ID, PW$ and so on.        |
| *****02  | valid          | Normal User       | Read only            | $ID, PW$ and so on.        |
| *****ab  | valid          | Senior User       | Read and modify      | $ID, PW$ and so on.        |
| *****3y  | invalid        | Normal User       | Read and write       | $ID, PW$ and so on.        |
| ...      | ...            | ...               | ...                  | ...                        |

Step 5. RC computes $r_i=H_i(x'_2 \cdot \lambda_i^2 \cdot \lambda_i \cdot \ldots \cdot \lambda_i \cdot pw, k)$ ($r_i \in Z'_r$), $R_i=rP_i$, $h_i=H_i(ID, \Box R_i)$, $PW=fH_i(pw, \Box ID)$ (certificate for authentication), $PW''''=H_i(pw, \Box ID)$ (certificate for multiple registration), $d_i=r_i+h_iS_i$ (private key of user), $P_i=R_i+h_iP_{sw}$ (public key of user).

ISSN: 0010-8189  
© CONVERTER 2020  
www.converter-magazine.info
Step 6. RC stores user’s important information \(ID, PW, PW^{\text{a}}, PW^{\text{b}}, \Psi, \Phi_a, R, h_i\) and \(P_i\) in the user’s permission table of the server, publishes \(ID\) and \(P_i\) of user \(U_i\). (As shown in Table 2).

Step 7. RC encrypts the registration information \(M_{\text{enc}}^{\text{reg}} = \{r_i, R_i, h_i, PW, PW^{\text{a}}, PW^{\text{b}}, \Psi, \Phi_a\}\) with the algorithm as \(M_{\text{enc}} = H(\text{pw} || \text{r}) \oplus (\text{PW}^{\text{reg}} \otimes M_{\text{sec}}^{\text{reg}})\), then sends the message \(M_{\text{enc}}^{\text{reg}}, H(M_{\text{enc}}^{\text{reg}})\) to \(U_i\) via public network.

Step 8. After receiving \(M_{\text{enc}}^{\text{reg}}, H(M_{\text{enc}}^{\text{reg}})\), user \(U_i\), checks firstly if the value of timestamp \(\tau_i\) exceeds the allowed maximum range \(\Delta \tau\) or not. If the value of the timestamp exceeds the allowed maximum range, the user \(U_i\) must discard the message received and re-register. If it meets the requirements, the user \(U_i\) computes \(H(M_{\text{enc}}^{\text{reg}})\) and verifies that the information is the same as the received \(H(M_{\text{enc}}^{\text{reg}})\).

If the result doesn’t hold, the user \(U_i\) must discard the message received and re-register. If it meets the requirements, the user \(U_i\) computes \(PW^{\text{reg}} = H(\text{pw} || ID)\) and \(M_{\text{enc}}^{\text{reg}} = PW^{\text{reg}} \oplus (H(\text{pw} || \text{r}) \oplus M_{\text{sec}}^{\text{reg}})\), sequentially gets the registration information \(M_{\text{enc}}^{\text{reg}} = \{r_i, R_i, h_i, PW, PW^{\text{a}}, PW^{\text{b}}, \Psi, \Phi_a\}\).

Step 9. The user \(U_i\) publishes the information \(ID, P_i\) and \(R_i\), stores the information \(r_i, h_i, PW, PW^{\text{a}}, PW^{\text{b}}, \Psi, \Phi_a\) secretly in the registration module.

In order to avoid user secret information leakage problem because of the loss of terminal, registration module encrypts user’s secret information with symmetric cipher before storing in the smart terminal (password of the user as key). The number of password authentication error is within limits. Only after password authentication is passed, the registration module automatically decrypts the stored information and makes it available to users.

4.3.3 Login and authentication phase (as shown in Fig. 5)

![Diagram](image.png)

**Fig. 5 Login and authentication phase**

Step 1. User \(U_i\) types in the password \(PW\) to decrypt information stored in the registration module if the password input is correct. Then the registration module computes \(H(\text{pw} || \text{r})\) and compares the calculation consequence to the stored information \(PW = H(\text{pw} || \text{r})\). If the result is different, password authentication cannot be passed.
which results in terminating the authentication service. If the result is the same, password authentication can be passed.

Step 2. The user \( U \) uses the camera to obtain the face image, extracts the user’s face data \( B \), through face detection, computes \( \text{Vec}(B) = \Psi, \Phi_\alpha \) and compares the user’s face data \( B \) with the result of calculation \( \text{Vec}(B) \). Similarity that does not reach the set threshold cannot pass face authentication. Otherwise face authentication can be passed.

Step 3. The user \( U \) selects a random element \( e_i \) from group \( Z^*_i \) as the temporary private key, computes the temporary public key \( E_i = e_i P \). After extracting current time \( t_i \), the user \( U \) sends \( \{R, E, t_i\} \) to \( S_i \) via public network.

Step 4. The application server \( S_i \) selects a random element \( e_i \) from group \( Z^*_i \) as the temporary private key, computes the temporary public key \( E_i = e_i P \). After extracting current time \( t_i \), the application server \( S_i \) sends \( \{R, E, t_i\} \) to \( U_i \).

Step 5. After receiving \( \{R, E, t_i\} \), \( U_i \) checks firstly if the timestamp \( t_i \) exceeds the allowed maximum range \( \Delta t \) or not. If the value of the timestamp exceeds the allowed maximum range, the user \( U_i \) must refuse the request. If the result meets the requirements, the user \( U_i \) computes \( P_j = R_j + H_i(SID_j \supset R_j)P_{new} \), \( K^1_j = (d_j + 2e_j)(P_j + 2E_j) \), \( K^2_j = (d_j + e_j)(P_j + E_j) \) and the shared and temporary session key \( SK_{ij} = H_2(ID, SID, E_j, E_j \supset K^1_j \supset K^2_j) \). 

Step 6. After receiving \( \{R_i, E_i, t_i\} \), the application server \( S_i \) checks firstly if the timestamp \( t_i \) exceeds the allowed maximum range \( \Delta t \) or not. If the value of the timestamp exceeds the allowed maximum range, the application server \( S_i \) must refuse the request. If the result meets the requirements, the application server \( S_i \) computes \( P_i = R_i + H_j(SID_i \supset R_i)P_{new} \), \( K^1_i = (d_i + 2e_i)(P_i + 2E_i) \), \( K^2_i = (d_i + e_i)(P_i + E_i) \) and the shared session key \( SK_i = H_2(ID_i, SID_i, E_i, E_i \supset K^1_i \supset K^2_i) \).

The equations \( K^1_i = K^1_j \), \( K^2_i = K^2_j \) and \( SK_i = SK_{ij} \) can be verified by calculation. Therefore, the temporary session key can ensure the following communication security after mutual authentication.

4.3.4 Password change phase
User \( U \) chooses new password \( pw_{new} \) firstly, then incorporates a new password \( pw_{new} \) into the registration information \( M^{new}_{i} = \{ID_i, pw, pw_{new}, B_j, r_j\} \).

The following process is almost the same as the user registration phase except RC needing verify the user’s face data in the password phase.

4.3.5 Re-registration
If terminal is lost or changed, the user must use another one to register afresh and login the authentication system because the user’s data stored in the original terminal cannot be obtained.

The user \( U_i \) chooses a new \( ID_{new} \) firstly, then incorporates \( ID_{new} \) into the registration information \( M^{new}_{i} = \{ID_i, ID_{new}, pw, B^{new}_j, r^{new}_j\} \).

The following process is almost the same as the password change phase.

V. Security analysis of the proposed scheme

5.1 The robustness of the scheme
RC participates in the authentication process only during the register process of users and application servers in the proposed scheme. Authentication process requires only temporary key and shared session key, without the participation of RC. This method enhances the robustness of the scheme.

5.2 Password guessing attack

This proposed scheme requires user’s correct password and face information. Even if the attacker may get password by guessing, the attacker still cannot pass the authentication.

5.3 Terminal lost attack

The user’s information stored in the terminal is encrypted. The attacker cannot decrypt the secret information without the correct password.

5.4 Impersonation attack

The identities are public. After obtaining the identity of the user, the attacker cannot sponsor the impersonation attack without the user’s password, face data and the private key. Similarly, the attacker cannot sponsor the impersonation attack without the private key.

5.5 Man-In-The-Middle attack

The transmitted information is encrypted. Attacker cannot obtain the user’s temporary session key and sponsor Man-In-The-Middle attack.

5.6 Denial of service attack

Registration or authentication phase in the scheme has the maximum limitation for the time interval of repetitive operation and the number of request within a certain time. The design can effectively control the frequent registration or authentication requests initiated by the attacker.

5.7 Replay attack

In the process of information transmission, time information added in the message has the function of timestamp and can effectively avoid replay attack.

5.8 Privileged insider attack

In this scheme, the user’s password and private key are not saved in RC. The hash function is irreversible. Even if the system administrator can obtain the other information of the user, he cannot calculate the pivotal and secret information $r, pw$ and $pw''$ of the user.

5.9 Forward secrecy

The temporary session key is one-time. The session key generated from each authentication is different. The scheme can ensure forward security.

5.10 Mutual authentication
In this paper, only the legal user and the legal application server have their own private key. Only the calculation results of legal user or application server which competes the session key with own private keys are correct and identical.

The security comparisons are shown in Table 3.

| Security aspect                  | Odelu[4] | Chuang[5] | Irshad[6] | Proposed scheme |
|----------------------------------|----------|-----------|-----------|-----------------|
| Robustness of scheme             | No       | Yes       | Yes       | Yes             |
| Terminal lost attack             | Yes      | Yes       | No        | Yes             |
| Impersonation attack             | Yes      | Yes       | No        | Yes             |
| Denial of service attack         | No       | No        | No        | Yes             |
| Man-In-The-Middle attack         | Yes      | Yes       | Yes       | Yes             |
| Privileged insider attack        | No       | Yes       | Yes       | Yes             |
| Perfect forward secrecy          | Yes      | Yes       | Yes       | Yes             |
| Mutual authentication            | Yes      | Yes       | Yes       | Yes             |

VI. Performance comparison

The consumption is measured on an Intel Core i5-3470 platform. The results of execution time are shown in table 4 and 5.

| Variety of computing                  | Execution time (s) |
|---------------------------------------|--------------------|
| XOR operation: $T_x$                  | tiny and negligible|
| Hash operation(MD5): $T_\mu$          | 0.00097            |
| Symmetric encryption(DES): $T_\sigma$ | 0.01600            |
| Symmetric decryption(DES): $T_\sigma$ | 0.01600            |
| Chebyshev chaotic mapping $T_x \mod p$: $T_\mu$ | 0.61122 |
| Obtaining P and R from $w$: $T_{ow}$  | 0.78097            |
| Recovering R from $w'$ and P: $T_{hp}$ | 0.18097            |
| Public key encryption based on elliptic curve: $T_{pu}$ | 0.93896 |
| Public key decryption based on elliptic curve: $T_{pd}$ | 0.93896 |
| Elliptic curve point multiplication operation: $T_\sigma$ | 0.00223 |
| Compute the singular value of the matrix: $T_{svd}$ | 0.25000 |
Table 5 Execution time comparison of login and authentication

| Scheme          | User          | Application server | RC                  | Total time (s) |
|-----------------|---------------|--------------------|---------------------|----------------|
| Odelu[4]        | $7T_u + T_s + T_\text{rec} + 3T_{\text{msg}}$ | $T_s + 6T_u + 2T_{\text{msg}} + T_p$ | $T_s + T_{\text{rec}} + 2T_u + 11T_p$ | 0.28           |
| Chuang[5]       | $5T_u + 2T_s + 4T_p$ | $T_s + 4T_u + 4T_p$ |                      | 1.36           |
| Irshad[6]       | $5T_s + 8T_u$ | $2T_s + 8T_u$ | 0                    | 0.02           |
| Proposed scheme | $3T_u + 5T_c + T_{\text{ino}} + T_h$ | $2T_u + 5T_c$ | 0                    | 0.27           |

VII. Conclusion

Aiming at solving password memory and exposure in traditional identity authentication, the scheme proposed in this paper combines password hiding, face recognition and elliptic curve cryptography to realize three-factor authentication. The results show that the proposed scheme has stronger robustness, higher security, better convenience and less computation cost than other similar schemes.

Acknowledgements

This research was supported by National Natural Science Foundation of China (Grant No. 41706201).

Reference

[1] E. J. Yoon and K.-Y Yoo, Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem, Super Computing, vol. 63, no. 1, 2016, pp. 235-255.
[2] P. Chandrakar, H. Om, Cryptanalysis and extended three-factor remote user authentication scheme in multi-server environment, Arabian Journal for Science and Engineering, vol. 42, no. 2, 2017, pp. 765-786.
[3] D. He, D. Wang, Robust biometrics-based authentication scheme for multi-server environment, Systems Journal IEEE, vol. 9, no. 3, 2016, pp. 816-823.
[4] V. Odelu, A. Das, A. Goswami, A secure biometrics-based multi-server authentication protocol using smart cards, IEEE Transactions on Information Forensics and Security, vol. 10, no. 9, 2016, pp. 1953-1966.
[5] M. C. Chuang, M. C. Chen, An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics, Expert Systems with Applications, vol. 41, no. 4, 2016, pp. 1411-1418.
[6] A. Irshad, M. Sher, S. A. Chaudhary, et al., An efficient and anonymous multi-server authenticated key agreement based on chaotic map without engaging Registration Centre, The Journal of Supercomputing, vol. 72, no. 4, 2017, pp. 1623-1644.
[7] N. M. Lwamo, L. Zhu, C. Xu, et al., Suua: A secure user authentication scheme with anonymity for the single & multi-server environments, Information Sciences, vol. 477, 2019, pp. 369-385.
[8] S. Roy, A. K. Das, S. Chatterjee, et al., Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing-based healthcare applications, IEEE Transactions on Industrial Informatics, vol. 15, no. 1, 2019, pp. 457-468.
[9] S. Chatterjee, S. Roy, A. K. Das, et al., Secure biometric-based authentication scheme using Chebyshev chaotic map for multi-server environment, IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 5, 2018, pp. 824-839.
[10] Z. Xu, D. He, X. Huang, Secure and efficient two-factor authentication protocol using RSA signature for multi-server environments, International Conference on Information and Communications Security,
2017, pp. 595-605.
[11] S. Barman, H. P. Shum, S. Chattopadhyay, D. Samanta, A secure authentication protocol for multi-server-based E-healthcare using a fuzzy commitment scheme, IEEE Access, vol. 7, 2019, pp. 12557-12574.
[12] Y. Tai, J. Yang, L. Luo, F. Zhang, J. Qian, Learning Discriminative Singular Value Decomposition Representation for Face Recognition, Pattern Recognition, vol. 50, 2016, pp. 1-16.
[13] Z. Wang, Z. Ma, S. Luo, Identity-based efficient authentication and key agreement protocol for mobile Internet, Journal on Communications, vol. 38, no. 8, 2017, pp. 19-27.
[14] H. Sun, L. Li, L. Zhang, et al., An enhanced identity-based authentication key agreement protocol for mobile Internet. Computer Engineering, vol. 45, no. 9, 2019, pp. 153-160.