CONVERTER MAGAZINE
Volume 2021, No. 3

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

Yu Zhang[1,2*], Guangmin Sun[1]

[1]Faculty of Information Technology, Beijing University of Technology, Beijing, China

[2]Jining University, Shandong, China

**Corresponding Author.*

Abstract

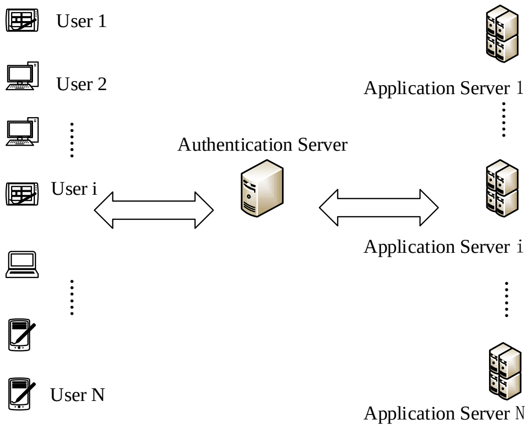
Aiming at the security problems of authentication in multi-server environments, a novel three-factor authentication scheme for multi-server environments of industrial Internet is proposed. After verifying password and face, a temporary session key is established for the user and server. Then the user 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.



ISSN: 0010-8189

Fig. 1. Authentication framework of multi-server environment

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[7], based on pairing[8], based on chaotic-map[9], based on integer-factorization-problem[10], and based on hash[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.[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

$$A = USV^T = \sum_{i=1}^{k} \lambda_i u_i v_i^T \tag{1}$$

$$Vec(A) = \sum_{i=1}^{k} \lambda_i Vec(u_i v_i^T) \tag{2}$$

Where $A \in R^{m \times n}$, $m \geq n$, k is the rank of A, $k = n$, $Vec(\cdot)$ 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 $(Vec(u_1 v_1^T), \cdots, Vec(u_n v_n^T)) \in R^{(m \times n) \times n}$.

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

$$Vec(A) = \sum_{i=1}^{k} \lambda_i Vec(u_i v_i^T) \approx \sum_{i=1}^{n_r} \lambda_i Vec(u_i v_i^T) \tag{3}$$

Where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k > 0 = \lambda_{k+1} = \cdots \lambda_n = 0$.

Letting $\xi_i = Vec(u_i v_i^T) \in R^{(m \times n) \times 1}$, Eq. (3) can be rewritten as

$$Vec(A) \approx \sum_{i=1}^{n_r} \lambda_i \xi_i = (\xi_1, \xi_2, \cdots, \xi_{n_r})(\lambda_1, \lambda_2, \cdots, \lambda_{n_r})^T \triangleq \Psi\Phi \tag{4}$$

where $\Psi = (\xi_1, \xi_2, \cdots, \xi_{n_r}) \in R^{(m \times n) \times n_r}$ and $\Phi = (\lambda_1, \lambda_2, \cdots, \lambda_{n_r})^T \in R^{n_r \times 1}$. Ψ is the newly constructed basis set and Φ is the coefficient set of Ψ.

The next step of the method is to seek the most discriminant weight vector $\Phi_S = (\lambda_1^S, \lambda_2^S, \cdots, \lambda_{n_r}^S)^T$, $\lambda_1^S \geq \lambda_2^S \geq \cdots \geq \lambda_{n_r}^S > 0$, for constructing a new representation image B of the original image A, which can be written as

$$Vec(B) = \Psi\Phi_S \tag{5}$$

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[13] presented a key agreement protocol by making use of operation amount is four. In 2019, Sun[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

Table 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
$H_1(\cdot), H_2(\cdot)$	Hash function
S_j	Application server
SID_j	The unique identity of application server S_j
t_i, t_j	The current time
Δt	The effective time interval
r_i, r_i^{reg}, r_j	Random number
d_j	Private key
P_j	Public key
U_i	user
ID_i	The unique identity of user U_i
pw_i	Password of user
B_i^{reg}, B_i	The biometric data of the user's face image
$\oplus$	Exclusive OR (XOR)
$\Vert$	Concatenation operation
Ψ_i	The SVD basis set of a face image

ISSN: 0010-8189

Φ_{iS}	The coefficient set of Ψ_i
P_i	Public key of user
d_i	Private key of user
e_i, e_j	Temporary private key
E_i, E_j	Temporary public key
SK_{ij}, SK_{ji}	Shared session key
$D_A(M)$	Digital signature about message M
$E_A(M)$	Encrypt message M

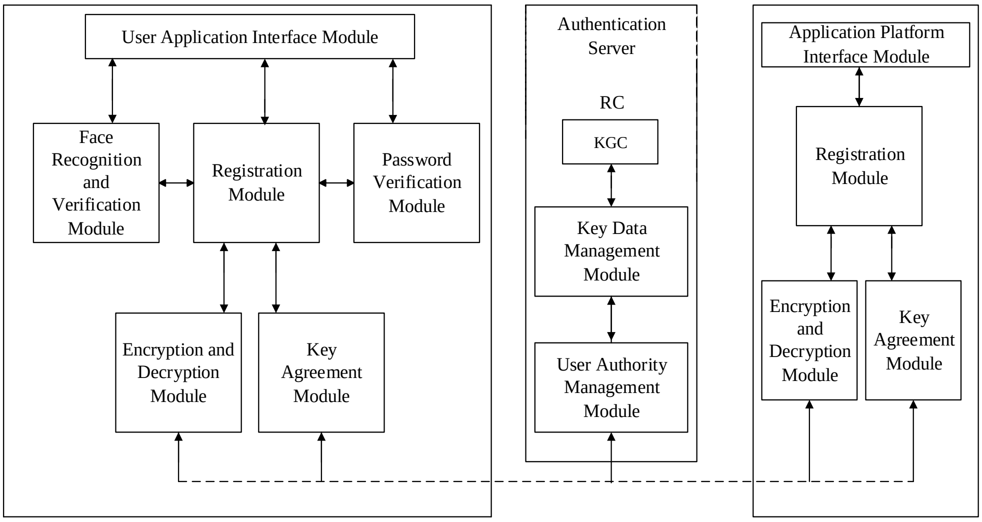

User Application Interface Module
Face Recognition and Verification Module
Registration Module
Password Verification Module
Encryption and Decryption Module
Key Agreement Module
Authentication Server
RC
KGC
Key Data Management Module
User Authority Management Module
Application Platform Interface Module
Registration Module
Encryption and Decryption Module
Key Agreement Module

Fig. 2. Framework of the proposed scheme

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 ($E/F_p,G,q,P$), where p is a k-bit large prime number, E/F_p is an elliptic curve of finite field F_p, P is a generator of F_p, G is an additive cyclic group of F_p, q is the order of G.

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

$H_1:\{0,1\}^*\times G\to Z_q^*$,
$H_2:\{0,1\}^*\times\{0,1\}^*\times G^4\to\{0,1\}^k$;

KGC preserves the master private key s secretly, publishes ($E/F_p,G,q,P,P_{sys},H_1,H_2$).

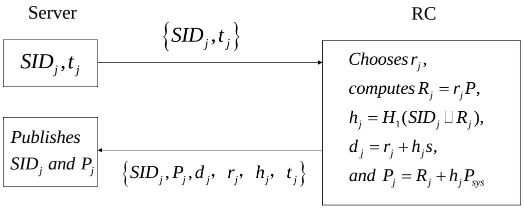


Fig. 3. Application server registration phase

4.3.2 Registration phase

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

Step 1. Server S_j selects the unique identity SID_j, then sends SID_j and t_j to RC via secure channel.

Step 2. RC receives the requesting registration information $\{SID_j,t_j\}$ from S_j and checks firstly if the value of the timestamp t_j exceeds the allowed maximum range Δt or not. RC refuses repetitive registration request from the same one within the time interval Δt. If Δt meet the requirement, RC checks whether the identity SID_j 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_q^* for server S_j, computes $R_j=r_jP$, $h_j=H_1(SID_j\Box R_j)$, privet key $d_j=r_j+h_js$ and public key $P_j=R_j+h_jP_{sys}$. Then RC stores $\{SID_j,P_j,d_j,\ r_j,\ h_j,\ t_j\}$ in the key data management module, and sends the data above to application server S_j via security channel.

Step 3. After receiving $\{SID_j,P_j,d_j,\ r_j,\ h_j,\ t_j\}$, application server S_j ensures the security of d_j, r_j,h_j and t_j. Then the server S_j publishes own identity SID_j and public key P_j.

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

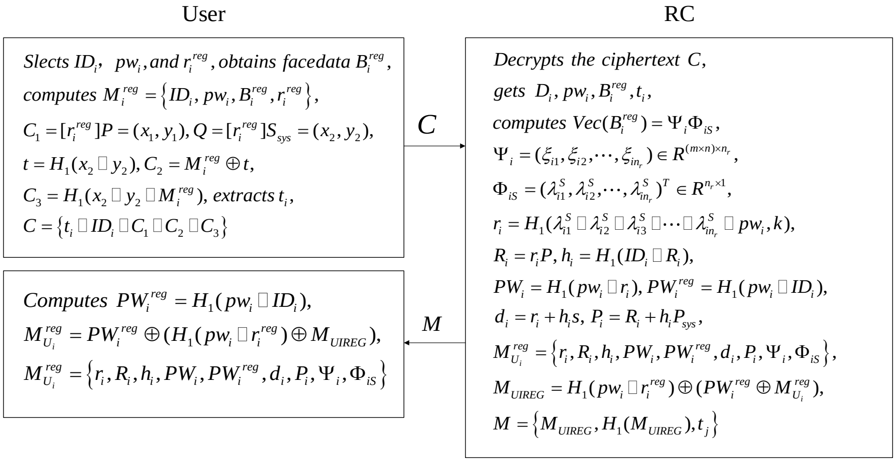

User
RC
$Slects\ ID_i,\ pw_i, and\ r_i^{reg}, obtains\ facedata\ B_i^{reg},$
$computes\ M_i^{reg}=\{ID_i,pw_i,B_i^{reg},r_i^{reg}\},$
$C_1=[r_i^{reg}]P=(x_1,y_1),Q=[r_i^{reg}]S_{sys}=(x_2,y_2),$
$t=H_1(x_2\Box y_2),C_2=M_i^{reg}\oplus t,$
$C_3=H_1(x_2\Box y_2\Box M_i^{reg}),\ extracts\ t_i,$
$C=\{t_i\Box ID_i\Box C_1\Box C_2\Box C_3\}$
C
$Decrypts\ the\ ciphertext\ C,$
$gets\ D_i,pw_i,B_i^{reg},t_i,$
$computes\ Vec(B_i^{reg})=\Psi_i\Phi_{iS},$
$\Psi_i=(\xi_{i1},\xi_{i2},\cdots,\xi_{in_r})\in R^{(m\times n)\times n_r},$
$\Phi_{iS}=(\lambda_{i1}^S,\lambda_{i2}^S,\cdots,\lambda_{in_r}^S)^T\in R^{n_r\times 1},$
$r_i=H_1(\lambda_{i1}^S\Box\lambda_{i2}^S\Box\lambda_{i3}^S\Box\cdots\Box\lambda_{in_r}^S\Box pw_i,k),$
$R_i=r_iP,h_i=H_1(ID_i\Box R_i),$
$PW_i=H_1(pw_i\Box r_i),PW_i^{reg}=H_1(pw_i\Box ID_i),$
$d_i=r_i+h_is,\ P_i=R_i+h_iP_{sys},$
$M_{U_i}^{reg}=\{r_i,R_i,h_i,PW_i,PW_i^{reg},d_i,P_i,\Psi_i,\Phi_{iS}\},$
$M_{UIREG}=H_1(pw_i\Box r_i^{reg})\oplus(PW_i^{reg}\oplus M_{U_i}^{reg}),$
$M=\{M_{UIREG},H_1(M_{UIREG}),t_j\}$
M
$Computes\ PW_i^{reg}=H_1(pw_i\Box ID_i),$
$M_{U_i}^{reg}=PW_i^{reg}\oplus(H_1(pw_i\Box r_i^{reg})\oplus M_{UIREG}),$
$M_{U_i}^{reg}=\{r_i,R_i,h_i,PW_i,PW_i^{reg},d_i,P_i,\Psi_i,\Phi_{iS}\}$

Fig. 4. User registration phase

Step 1. User U_i chooses the unique identity ID_i and passwords pw_i, 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_i^{reg} of user's face image, chooses a random element r_i^{reg} from group Z_q^* and composes registration information $M_i^{reg}=\{ID_i, pw_i, B_i^{reg}, r_i^{reg}\}$.

Step 2. The encryption and decryption module of mobile terminal encrypts the registration information $M_i^{reg}=\{ID_i, pw_i, B_i^{reg}, r_i^{reg}\}$ with the master public key of RC. Firstly, the module above computes point $C_1=[r_i^{reg}]P=(x_1,y_1)$ of the elliptic curve E/F_p and point $Q=[r_i^{reg}]S_{sys}=(x_2,y_2)$, then converts data type of x_2 and y_2 to bit string if the point Q is non-infinity point, otherwise choosing r_i^{reg} again. Then the module above computes $t=H_1(x_2 \Box y_2)$. If the bit string of t is all zeros, the random element r_i^{reg} must be chosen again. Then the module above computes $C_2=M_i^{reg}\oplus t$ and $C_3=H_1(x_2 \Box y_2 \Box M_i^{reg})$. After extracting current time t_i, the user U_i sends the message $C=\{t_i \Box ID_i \Box C_1 \Box C_2 \Box C_3\}$ to RC via public network.

Step 3. After receiving $C=\{t_i \Box ID_i \Box C_1 \Box C_2 \Box C_3\}$, RC checks firstly if the timestamp t_i exceeds the allowed maximum range Δt or not. RC refuses to repetitive registration request from the same one within the time interval Δt. If the result holds, in order to avoid the same identity for different server's registration, RC checks whether the identity ID_i has already been registered. If the user's identity ID_i 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_1 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_1 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 $Q'=[s]C_1=(x_2',y_2')$ 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 $t'=H_1(x_2' \Box y_2')$ 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_2 from the message C, computes $M_i'^{reg}=C_2\oplus t'$ and $C_3'=H_1(x_2' \Box y_2' \Box M_i'^{reg})$. Then RC takes out C_3 from the message C and compares C_3 with C_3'. If $C_3'\neq C_3$, RC reports an error and exits. If $C_3'=C_3$, RC confirms $M_i'^{reg}=M_i^{reg}=\{ID_i, pw_i, B_i^{reg}, t_i\}$, decrypts the ciphertext into plaintext and gets user's information $ID_i, pw_i, B_i^{reg}, t_i$.

Step 4. After getting the user's face image data B_i^{reg}, RC computes $Vec(B_i^{reg})=\Psi_i\Phi_{iS}$ with singular value decomposition, where $\Psi_i=(\xi_{i1},\xi_{i2},\cdots,\xi_{in_r})\in R^{(m\times n)\times n_r}$ and $\Phi_{iS}=(\lambda_{i1}^S,\lambda_{i2}^S,\cdots,\lambda_{in_r}^S)^T\in R^{n_r\times 1}$ ($\lambda_{i1}^S\geq\lambda_{i2}^S\geq\cdots\geq\lambda_{in_r}^S>0$). Ψ_i is the singular value base set of the face image data B_i^{reg} of user U_i, Φ_{iS} is the coefficient of Ψ_i.

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 $r_i=H_1(\lambda_{i1}^S \Box \lambda_{i2}^S \Box \lambda_{i3}^S \Box \cdots \Box \lambda_{in_r}^S \Box pw_i, k)$ ($r_i\in Z_q^*$), $R_i=r_iP$, $h_i=H_1(ID_i \Box R_i)$, $PW_i=H_1(pw_i \Box r_i)$ (certificate for authentication), $PW_i^{reg}=H_1(pw_i \Box ID_i)$ (certificate for multiple registration), $d_i=r_i+h_is$ (private key of user), $P_i=R_i+h_iP_{sys}$ (public key of user).

Step 6. RC stores user's important information $ID_i, PW_i, PW_i^{reg}, B_i^{reg}, \Psi_i, \Phi_{iS}, R_i, h_i$ and P_i in the user's permission table of the server, publishes ID_i and P_i of user U_i (As shown in the Table 2).

Step 7. RC encrypts the registration information $M_{U_i}^{reg} = \{r_i, R_i, h_i, PW_i, PW_i^{reg}, d_i, P_i, \Psi_i, \Phi_{iS}\}$ with the algorithm as $M_{UIREG} = H_1(pw_i \| r_i^{reg}) \oplus (PW_i^{reg} \oplus M_{U_i}^{reg})$, then sends the message $\{M_{UIREG}, H_1(M_{UIREG}), t_j\}$ to U_i via public network.

Step 8. After receiving $\{M_{UIREG}, H_1(M_{UIREG}), t_j\}$, user U_i checks firstly if the value of timestamp t_j exceeds the allowed maximum range Δt 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 $H_1'(M_{UIREG})$ and verifies that the information is the same as the received $H_1(M_{UIREG})$.

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 $PW_i^{reg} = H_1(pw_i \| ID_i)$ and $M_{U_i}^{reg} = PW_i^{reg} \oplus (H_1(pw_i \| r_i^{reg}) \oplus M_{UIREG})$, sequentially gets the registration information $M_{U_i}^{reg} = \{r_i, R_i, h_i, PW_i, PW_i^{reg}, d_i, P_i, \Psi_i, \Phi_{iS}\}$.

Step 9. The user U_i publishes the information ID_i, P_i and R_i, stores the information $r_i, h_i, PW_i, PW_i^{reg}, d_i, \Psi_i, \Phi_{iS}$ 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)

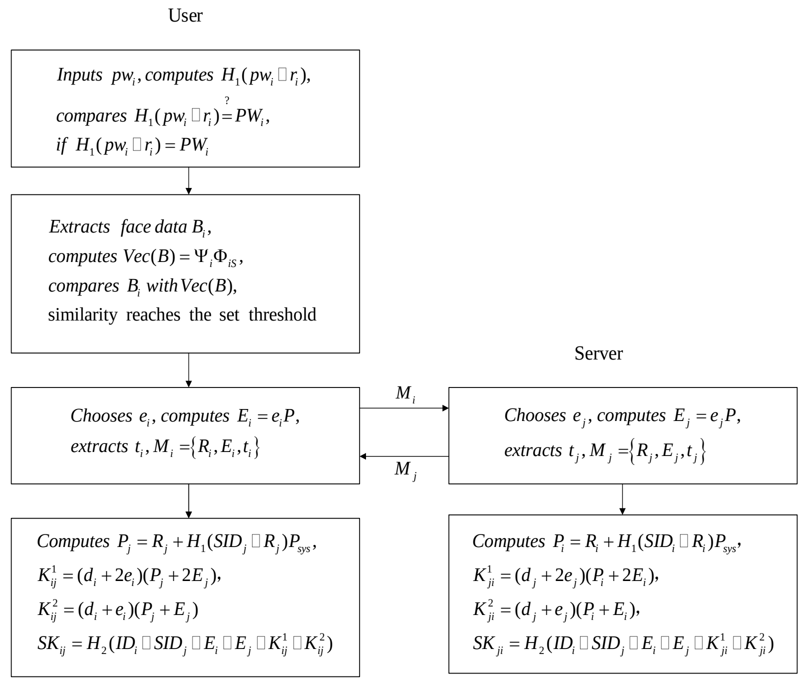

User
Inputs pw_i, computes $H_1(pw_i \| r_i)$,
compares $H_1(pw_i \| r_i) \overset{?}{=} PW_i$,
if $H_1(pw_i \| r_i) = PW_i$
Extracts face data B_i,
computes $Vec(B) = \Psi_i \Phi_{iS}$,
compares B_i with $Vec(B)$,
similarity reaches the set threshold
Server
M_i
M_j
Chooses e_i, computes $E_i = e_i P$,
extracts t_i, $M_i = \{R_i, E_i, t_i\}$
Chooses e_j, computes $E_j = e_j P$,
extracts t_j, $M_j = \{R_j, E_j, t_j\}$
Computes $P_j = R_j + H_1(SID_j \| R_j)P_{sys}$,
$K_{ij}^1 = (d_i + 2e_i)(P_j + 2E_j)$,
$K_{ij}^2 = (d_i + e_i)(P_j + E_j)$
$SK_{ij} = H_2(ID_i \| SID_j \| E_i \| E_j \| K_{ij}^1 \| K_{ij}^2)$
Computes $P_i = R_i + H_1(SID_i \| R_i)P_{sys}$,
$K_{ji}^1 = (d_j + 2e_j)(P_i + 2E_i)$,
$K_{ji}^2 = (d_j + e_j)(P_i + E_i)$,
$SK_{ji} = H_2(ID_i \| SID_j \| E_i \| E_j \| K_{ji}^1 \| K_{ji}^2)$

Fig. 5 Login and authentication phase

Step 1. User U_i types in the password pw_i to decrypt information stored in the registration module if the password input is correct. Then the registration module computes $H_1(pw_i \| r_i)$ and compares the calculation consequence to the stored information PW_i ($PW_i = H_1(pw_i \| r_i)$). 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_i uses the camera to obtain the face image, extracts the user's face data B_i through face detection, computes $Vec(B) = \Psi_i \Phi_{iS}$ and compares the user's face data B_i with the result of calculation $Vec(B)$. Similarity that does not reach the set threshold cannot pass face authentication. Otherwise face authentication can be passed.

Step 3. The user U_i selects a random element e_i from group Z_q^* as the temporary private key, computes the temporary public key $E_i = e_i P$. After extracting current time t_i, the user U_i sends $\{R_i, E_i, t_i\}$ to S_j via public network.

Step 4. The application server S_j selects a random element e_j from group Z_q^* as the temporary private key, computes the temporary public key $E_j = e_j P$. After extracting current time t_j, the application server S_j sends $\{R_j, E_j, t_j\}$ to U_i.

Step 5. After receiving $\{R_j, E_j, t_j\}$, U_i checks firstly if the timestamp t_j exceeds the allowed maximum range Δt 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 $P_j = R_j + H_1(SID_j \Box R_j)P_{sys}$, $K_{ij}^1 = (d_i + 2e_i)(P_j + 2E_j)$, $K_{ij}^2 = (d_i + e_i)(P_j + E_j)$ and the shared and temporary session key $SK_{ij} = H_2(ID_i \Box SID_j \Box E_i \Box E_j \Box K_{ij}^1 \Box K_{ij}^2)$.

Step 6. After receiving $\{R_i, E_i, t_i\}$, the application server S_j checks firstly if the timestamp t_i exceeds the allowed maximum range Δt or not. If the value of the timestamp exceeds the allowed maximum range, the application server S_j must refuse the request. If the result meets the requirements, the application server S_j computes $P_i = R_i + H_1(SID_i \Box R_i)P_{sys}$, $K_{ji}^1 = (d_j + 2e_j)(P_i + 2E_i)$, $K_{ji}^2 = (d_j + e_j)(P_i + E_i)$ and the shared session key $SK_{ji} = H_2(ID_i \Box SID_j \Box E_i \Box E_j \Box K_{ji}^1 \Box K_{ji}^2)$.

The equations $K_{ij}^1 = K_{ji}^1$, $K_{ij}^2 = K_{ji}^2$ and $SK_{ij} = SK_{ji}$ 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_i chooses new password pw_i^{new} firstly, then incorporates a new password pw_i^{new} into the registration information $M_i^{reg} = \{ID_i, pw_i, pw_i^{new}, B_i, r_i\}$.

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_i^{new} firstly, then incorporates ID_i^{new} into the registration information $M_i^{reg} = \{ID_i, ID_i^{new}, pw_i, B_i^{reg}, r_i^{reg}\}$.

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 requestion 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_i, PW_i and PW_i^{reg} 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.

Table 3 Security comparison

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.

Table 4 Consumption time table

Variety of computing	Execution time (s)
XOR operation: T_X . Face matching: T_{B_i}	tiny and negligible
Hash operation(MD5) : T_H	0.00097
Symmetric encryption(DES) : T_E	0.01600
Symmetric decryption(DES) : T_D	0.01600
Chebyshev chaotic mapping $T_n(x) \bmod \mathrm{p}$: T_M	0.61122
Obtaining P and R from w : T_{Gen}	0.78097
Recovering R from w' and P: T_{Rep}	0.18097
Public key encryption based on elliptic curve: T_{PM}	0.93896
Public key decryption based on elliptic curve: T_{PD}	0.93896
Elliptic curve point multiplication operation: T_m	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]	$7T_H+T_E+T_X+T_{\mathrm{Rep}}+3T_{PM}$	$T_E+6T_H+2T_{PM}+T_D$	$T_E+T_{PM}+2T_D+11T_H$	0.28
Chuang[5]	$5T_H+2T_X+4T_M$	$T_X+4T_H+4T_M$	0	1.36
Irshad[6]	$5T_X+8T_H$	$2T_X+8T_H$	0	0.02
Proposed scheme	$3T_H+5T_m+T_{SVD}+T_{B_i}$	$2T_H+5T_m$	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., Suaa: 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.