A Novel Privacy-preserving User Authentication Protocol for Big Data Environment

Jiabing Liu, Xudong He, Huoye Tang, Dejun Wang, and Bo Meng
(Corresponding author: Bo Meng)

School of Computer Science, South-Central University for Nationalities
Wuhan 430074, China
Email: mengscuec@gmail.com

(Received Dec. 4, 2019; Revised and Accepted July 23, 2020; First Online Apr. 24, 2021)

Abstract

Many authentication protocols use private information as a factor to implement user authentication. The authentication server can obtain user private information. Therefore, there exists a risk of privacy leakage with the rapid development of data mining technology. Hence, it's important to protect privacy in the authentication protocol. In this paper, we first model the PPMUAS (privacy-preserving remote user authentication protocol) protocol using Applied PI calculus and analyze it with ProVerif(Protocol Verifier). We found that it has three vulnerabilities. And then, we propose the PPUAPBDE protocol (privacy-preserving user authentication protocol for big data environment), which uses signcryption, homomorphic encryption, and fuzzy hash to protect user privacy of multi-behavior features. After that, PPUAPBDE is modeled using Applied PI calculus and analyzed with ProVerif. The result shows that it achieves confidentiality, authentication, and privacy. Finally, we develop an authentication system based on PPUAPBDE to evaluate Recall(recall) and FPR (false positive rate). The Recall is about 94.8%, and the FPR is 5.1%, which is better than PPMUAS.

Keywords: Multi-behavior; Authentication; Privacy; Fuzzy Hash; Homomorphic Encryption

1 Introduction

Many authentication protocols used private information, such as biometrics, behavior characteristics and hardware information, as factors to implement identity authentication [2, 12, 13, 16, 18, 19, 21, 27]. But they do not consider the privacy of private information. When private information are sent to authentication server, the authentication server can obtain user privacy. Therefore, there exists a risk of private information leakage with big data mining tools and techniques [26]. Hence it's very important to protect the privacy of the private information in authentication protocol.

In general, there are two methods to protect privacy. On the one hand, authentication server applies privacy protection technology to protect privacy, such as data anonymity, data distortion or cryptography [6, 25]. On the other hand, authentication server adopts authentication protocol provide authentication and privacy. The second method is adopted mostly as it solves the problem fundamentally. Privacy is protected before being sent to authentication server. Therefore, privacy-preserving authentication protocol [7, 11, 14, 15, 17, 23, 24] has attracted attention. The main contributions of this paper are summarized as follows.

1) Apply Applied PI calculus in the symbolic model to formalize PPMUAS protocol and analyze its confidentiality, authentication and privacy with ProVerif. The result shows that it achieves confidentiality and privacy, but AS(Authentication Server) cannot authenticate DB(Database), and AS cannot authenticate User mutually;

2) Present PPUAPBDE protocol, which protects privacy of user private information, such as user keyboard, mouse usage habits, system processes and network behavior, with signcryption, homomorphic encryption and fuzzy hash. PPUAPBDE protocol provides the mutual authentication between User and AS, the authentication from DB to AS and the authentication from User to DB;

3) Apply Applied PI calculus in the symbolic model to formalize PPUAPBDE protocol, and then analyze its confidentiality, authentication, and privacy with ProVerif. The results indicate that it achieves confidentiality, authentication and privacy;

4) Develop an authentication system based on PPUAPBDE protocol to evaluate the Recall and FPR. The Recall is about 94.8%, and the FPR is 5.1%, which are better than PPMUAS protocol.

2 Related Work

Nowadays, lots of private information, such as biometric, behavior characteristic and hardware information, are used for authentication protocols [2, 13, 16, 18, 19, 21, 27], but the privacy of private information has not been implemented. Hence privacy-preserving authentication protocols that provides privacy are introduced [7, 11, 17, 23, 24].

Without privacy preservation: Based on biometrics, S. X. Fang *et al.* [27] expounded the advantages of biometrics identification technology compared with traditional identification technology. The characteristics of face recognition methods are analyzed and compared. A. Rassan and H. Alshaher [21] used fingerprint recognition as authentication factor and proposed an authentication method with user handwritten response code. The method consists of registration and login phase. In the registration phase, the user fingerprint is collected with mobile device and is stored in the database after the pre-processing. In the login phase, the fingerprint is collected again. After pre-processing, the corresponding entry in the database is matched to judge whether user is valid or not. R. H. Li and Y. Z. Li [16] proposed a dynamic continuous authentication system. After user login successfully, the character combination of Keystroke behavior and mouse behavior is used to monitor user operation behavior. It is used to judge whether operation object is genuine. In general, the security of the system is guaranteed. G. Kambourakis *et al.* [19] explored the potential of keystroke dynamics for touchscreen-equipped smartphones. A touchstroke system is implemented in the Android platform. And different methodologies, such as every pair of pressed keys and the average value of pressed keys, are executed under different scenarios to estimate the effectiveness in authenticating the end-user. The result shows that there still need advanced privacy protection. M. Babaeizadeh [2] used user keystroke duration as authentication factor and proposed a method for authenticating mobile users. When the user registers, the information of user keystroke duration is stored in the database after user registers successfully. And the session expiration time is set to force the user to log in before the end of the session. Unauthorized personnel cannot log into the system for session hijacking. S. Kang *et al.* [13] designed a practical method for biometric authentication based on electrocardiogram signals which is collected from mobile or wearable devices. The proposed approach can reduce the time required to achieve the target FAR (false acceptance rate) and FRR (false rejection rate). The proposed method are implemented in a wearable watch to verify its feasibility. In the experiment results, the FAR and FRR are 5.2% and 1.9%, respectively. H. Jeong and E. Choi [18] proposed a security authentication using profiling techniques for access control and user authentication, in which the profile consists of user information (name, ID, personal preference, hobby, etc) and service information (service name, provider name, context, frequency value, etc). However, details and requirements are not presented.

With privacy preservation: Based on homomorphic encryption, Hatin [7] proposed a privacy-preserving biometric authentication protocol. The protocol relies on the homomorphic Goldwasser-Micali cryptosystem [10]. And, they proved its security against malicious, but cannot resist insider attacks. Ren *et al.* [17] proposed a privacy-preserving authentication and access control scheme to insure the interaction security between mobile user and service in PCEs. The proposed scheme integrates underlying cryptographic primitives: Blind signature and hash chain, into highly flexible and lightweight authentication protocol. Based on the ring signature, Gamage *et al.* [11] proposed an identity-based ring signature scheme to create privacy-preserving authenticatable messages. Ruj *et al.* [23] proposed a privacy-preserving authenticated access control scheme to insure the security of private information in clouds. In the proposed scheme, the cloud verifies the authentication of the user without knowing the user identity before storing information. The cloud, however, does not know the identity of the user who stores information, but only verifies the user credential, which protect the user privacy effectively. Vorugunti [24] claimed that they proposed the first privacy-preserving remote user authentication protocol(PPMUAS), which uses the user-specific information and behavioral features. They claimed PPMUAS is the first authentication protocol to consider multi-factors and hybrid profile for privacy-preserving remote user authentication. Because of Proverif's support for cryptographic primitives [5], Proverif is used to analyze and validate security protocols described in Horn clause or Applied PI calculus. Based on Applied PI calculus [1], we use ProVerif [4] to analyze the PPMUAS. And we found that it has three security vulnerabilities.

3 PPMUAS Protocol

In this section, the messages in PPMUAS are given. And then, we model the PPMUAS protocol using Applied PI calculus and analyze the protocol with ProVerif. We found that it has protection mechanism for user privacy, but it has three authentication vulnerabilities.

3.1 Message of PPMUAS

In 2017, C. Vorugunti [24] proposed PPMUAS, which uses user password and keystroke dynamics to generate user profile. The messages in PPMUAS protocol are shown in Figure 1.

Registration request: User sends message 1 to AS to launch the registration. The message 1 is composed of User ID Id_i. User number i, User profile Cup_i and Rpw_i. The inputs to function Fuzzy-Hash Pw_i include U_OlInfo and U_BrInfo , where U_OlInfo is User online information, and U_BrInfo is User browser information. The inputs to function FH_Enc consists of U_KeySD, U_AMM and

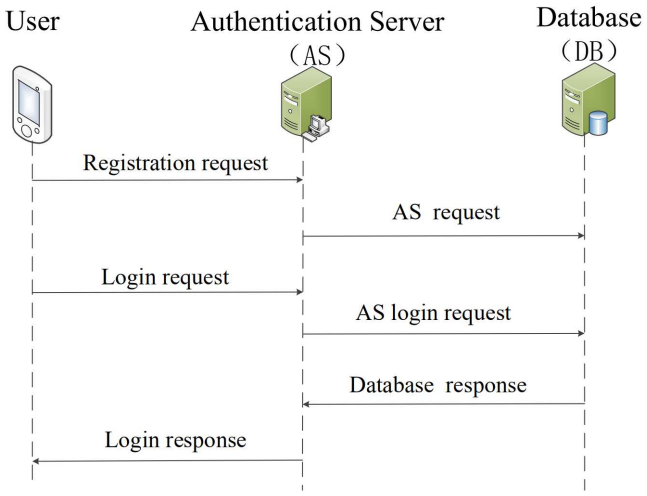

Figure 1: PPMUAS messages

U_GM, where *U_KeySD* is User keystroke dynamics information, *U_AMM* is accelerometer measurement information, *U_GM* is gyroscope measurement information. At last, message 1 is sent to AS through public channel c.

Authentication server request: After receiving message 1, Function Hash uses ID_i, RPW_i and i to generate hash value V_i. And AS stores it locally. At the same time, it creates message 2. The message contains α, $Rpup_i$ and i. $\alpha = E_{PUK}\ (r_i,\ k_n)$, which means α is obtained by encrypting the random number r_i which are generated by AS, and the User specific secret key k_n with the public key $Pu(k)$ of DB. $Rpup_i$ is received by the following method: Firstly, permute the user profile Cup_i with $Pup_i = \pi\ (Cup_i)$ to produce Pup_i, and then Pup_i is XORed with the random number r_i to get $Rpup_i$ with $Rpup\_i = Pup_i \bigoplus r_i$.After that, message 2 is generated and sent to DB through public channel c. After DB receives message 2, it decrypts α with the private key $Pr(k)$ to get r_i and k_n and obtains Pup_i by $Rpup_i$ and r_i through XOR. Finally, r_i, k_n, and Pup_i are saved in the database, and User registration is completed.

Login request: When User logs in, User enters the user id Id_i, the password Pw_i, and uses related information to generate Cup_i'. And then, message 3 is created in which it contains Id_i, Rpw_i, Cup_i' and i. Finally, the message 3 is sent to AS through public channel c.

AS login request: After AS receives message 3, V_i' is calculated firstly by the hash function and is compared with V_i saved in AS. If V_i and V_i' are equal, message 4 is generated. Message 4 contains α', $Rpup_i$' and i, where α' is created by the random number r_i' and the User session key k_{nn} by the public key $Pu(k_1)$ of DB. $Rpup_i$' is obtained by the following method: Firstly, permute the user profile Cup_i with Pup_i'$= \pi(Cup_i$') to produce Pup_i', and then Pup_i' is XORed with the random number r_i' to get $Rpup_i$' with $Rpup_i$' = Pup_i' *bigoplus* r_i'. Finally, message 4 is sent to DB through public channel c.

Database response: After receiving message 4, DB first decrypts with the private key to obtain r_i' and k_{nn}. Then $Rpup_i$' is XORed with r_i' to obtain Pup_i'. Finally, Pup_i' is compared with Pup_i stored in DB. If $HW((Pup_i' \bigoplus Pup_i)) \leq \triangle t$, it is legitimate user. Otherwise, AS denies access. HW is the Hamming weight. After that, message 5 is produced and sent to AS through public channel c, in which it contains "yes" or "no" only.

Login response: On receiving the message "yes" from DB, AS sends Content Service Ticket to User to access the content server resources, otherwise AS rejects and ends the connection.

3.2 Formal Modeling of PPMUAS

3.2.1 Function and Equational Theory

The functions and equations used in the modeling process are described in this section. We use Applied PI calculus to formalize PPMUAS protocol. The message x is encrypted by function *aenc(x, Pu)* with public key *Pu*, and message y is decrypted by function *adec(y, Pr)* with private key *Pr*. *XOR(x, y)* performs *XOR* calculation on message x and y. The Hamming weight algorithm *HW(x, y)* performs a Hamming weight measurement on the message x and y. The permutation function P permutes the message x. Figure 2 depicts the PPMUAS protocol function and equation theory.

$$\begin{bmatrix} fun\ FuzzyHash/1 \\ fun\ Hash/2 \\ fun\ FH\_Enc/1 \\ fun\ Hashone/3 \\ fun\ P/1 \\ fun\ XOR/2 \\ fun\ adec/2 \\ fun\ aenc/2 \\ fun\ Pu/1 \\ fun\ HW/2 \\ equation\ adec(aenc(x, Pu(y)), \Pr(y)) = x \\ equation\ HW(x, y) = (x, y) \\ equation\ XOR(XOR(x, y), y) = x \end{bmatrix}$$

Figure 2: Function and equation theory

3.2.2 Process

The whole PPMUAS protocol process mainprocess consists of three processes: *processUser*, *processAS* and *processDB*. They constitute the main process together, as shown in Figure 3.

$$mainprocess$$
$$(!\,processUser\,|!\,processAS\,|!\,processDB)$$

Figure 3: PPMUAS mainprocess

The all processes are shown in Figures 4, 5 and 6, respectively.

$$\begin{bmatrix} let\ processUser = \\ new\ U\_OlInfo \\ new\ U\_BrInfo \\ new\ U\_KeySD \\ new\ U\_AMM \\ new\ U\_GM \\ let\ Cup_i = \begin{bmatrix} (FuzzyHash(Pw_i),\ FuzzyHash(U\_OlInfo), \\ FuzzyHash(U\_BrInfo),\ FH\_Enc(U\_KeySD), \\ FH\_Enc(U\_AMM),\ FH\_Enc(U\_GM)) \end{bmatrix} in \\ let\ Rpw_i = Hash(Pw_i, b_i)\ in \\ let\ e = (ID_i, Rpw_i, Cup_i, i)\ in \\ out(c,e) \quad //registration\ phase \\ \\ new\ U\_OlInfo' \\ new\ U\_BrInfo' \\ new\ U\_KeySD' \\ new\ U\_AMM' \\ new\ U\_GM' \\ let\ Cup_i' = \begin{bmatrix} (FuzzyHash(Pw_i),\ FuzzyHash(U\_OlInfo'), \\ FuzzyHash(U\_BrInfo'),\ FH\_Enc(U\_KeySD'), \\ FH\_Enc(U\_AMM'),\ FH\_Enc(U\_GM')) \end{bmatrix} in \\ let\ Rpw_i = Hash(Pw_i, b_i)\ in \\ let\ e' = (ID_i, Rpw_i, Cup_i', i)\ in \\ out(c,e') \\ in(c,ticket) \quad //\log in\ phase \end{bmatrix}$$

Figure 4: PPMAUS processUser

$$\begin{bmatrix} let\ processAS = \\ in(c,m_1) \\ new\ r_i \\ new\ k_n \\ let\ (ID_i, Rpw_i, Cup_i, i) = m_1\ in \\ let\ V_i = Hashone(ID_i, Rpw_i, i)\ in \\ let\ Pup_i = P(Cup_i)\ in \\ let\ Rpup_i = XOR(Pup_i, r_i)\ in \\ let\ a = aenc((r_i, k_n), Pu(k_1))\ in \\ let\ DB = (a, Rpup_i, i)\ in \\ out(c,DB) \quad //registration\ phase \end{bmatrix} \begin{bmatrix} in(c,m_3) \\ new\ yes \\ let\ (ID_i, Rpw_i, Cup_i', i) = m_3\ in \\ let\ V_i' = Hashone(ID_i, Rpw_i, i)\ in \\ if\ V_i' = V_i\ then \\ new\ r_i' \\ new\ k_m \\ let\ Pup_i' = P(Cup_i')\ in \\ let\ Rpup_i' = XOR(Pup_i', r_i')\ in \\ let\ a' = aenc((r_i', k_m), Pu(k_n))\ in \\ let\ DB' = (a', Rpup_i', i)\ in \\ out(c,DB') \\ in(c,yes) \\ out(c,ticket) \quad //\log in\ phase \end{bmatrix}$$

Figure 5: PPMAUS processAS

$$\begin{bmatrix} let\ processDB = \\ in(c,m_2) \\ let\ (a, Rpup_i, i) = m_2\ in \\ let\ (r_i, k_n) = adec(a, Pu))\ in \\ let\ Pup_i = XOR(Rpup_i, r_i)\ in \\ //registration\ phase \end{bmatrix} \begin{bmatrix} in(c,m_4) \\ let\ (a', Rpup'_i, i) = m_4\ in \\ let\ (r_i, k_m) = adec(a', Pu)\ in \\ let\ Pup_i' = XOR(Rpup_i', r_i')\ in \\ if\ HW(Pup_i', Pup_i) = (Pup_i', Pup_i)\ then \\ new\ yes \\ out(c,yes) \quad //\log in\ phase \end{bmatrix}$$

Figure 6: PPMAUS processDB

3.3 Security Analysis of PPMUAS

In this section, the security analysis results of PPMUAS are given. The results indicate that PPMUAS has achieved confidentiality and privacy. But in the aspect of authentication, it has three security vulnerabilities that User cannot authenticate AS mutually and AS cannot authenticate DB.

3.3.1 Confidentiality

Here we use query attacker: Pw_i to model confidentiality of user password Pw_i. The analyzed result is shown in Figure 7. Figure 7 is the result of the confidentiality of Pw_i. The result is true, which proves that the user password Pw_i is secret. Because Pw_i is hashed by user to obtain a hash value. And then, the hash value is sent to AS. Owning to the one-way property of the hash function, and the attacker cannot obtain the password Pw_i through the hash value.

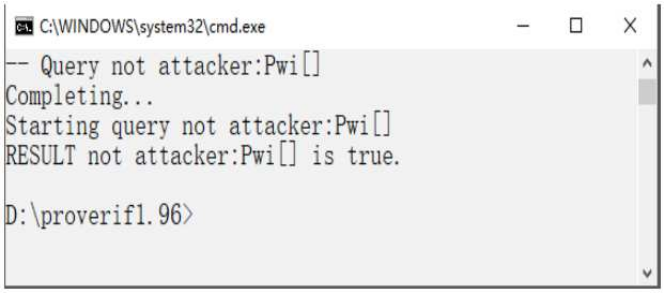

Figure 7: The confidentiality of Pw_i

3.3.2 Authentication

Here we use non-injective agreements to model authentication among User, AS and DB. The non-injective agreement is used to model authentication in ProVerif. PPMUAS protocol authentications are shown in Table 1.

Table 1: Authentication model

Non-injective agreement	Authentication
ev:endauthAs_User(x)⟶ ev:beginaauthAs_User(x)	AS authenticates User
ev:endauthUser_As(x)⟶ ev:beginaauthUser_As(x)	User authenticates AS
ev:endauthDB_As(x)⟶ ev:beginaauthDB_As(x)	DB authenticates AS
ev:endauthAs_DB(x)⟶ ev:beginaauthAs_DB(x)	AS authenticates DB

Figures 8 and 9 are mutual authentication results between AS and DB. The authentication result of DB to AS in Figure 8 is "true", it indicates that the authentication of DB to AS is achieved. AS sends Authentication Server login request message to DB, in which it contains α', $Rpup_i$' and i, where $Rpup_i$' is generated by AS. After receiving Authentication Server login request message, DB first decrypts with the private key to get r_i' and $Rpup_i$', and then creates Pup_i', which is compared to Pup_i saved in DB. If they are equal, it indicates that Authentication Server login request message is produced by the AS. Therefore, DB can authenticate AS. The authentication result of AS to DB in Figure 9 is "Can Not Be Proved", indicating that AS cannot authenticate DB. Because the DB response message is without any security mechanism. Thus the attacker can launch a impersonate attack. Therefore, AS cannot authenticate DB.

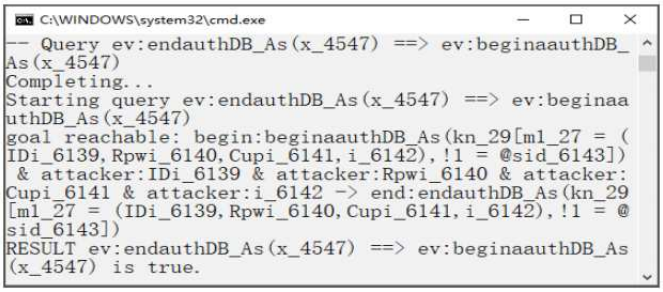

Figure 8: The authentication result of DB to AS

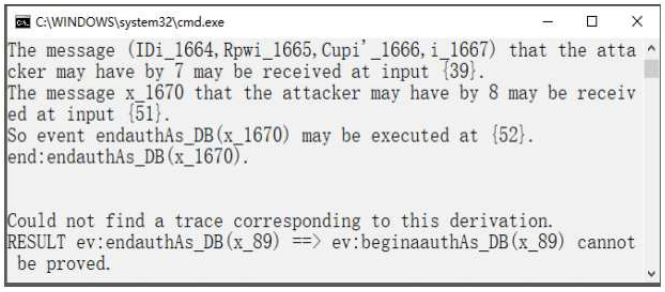

Figure 9: The authentication result of AS to DB

Figures 10 and 11 are mutual authentication results between AS and User. The results of both are "Can Not Be Proved", indicating that AS and User Cannot authenticate mutually. Because the attacker can pretend to be User to launch an impersonate attack. The Login response message sent by AS to User contains only one parameter ticket/reject, and the message does not have any security mechanisms. The attacker can initiate an impersonate attack, so User cannot authenticate AS.

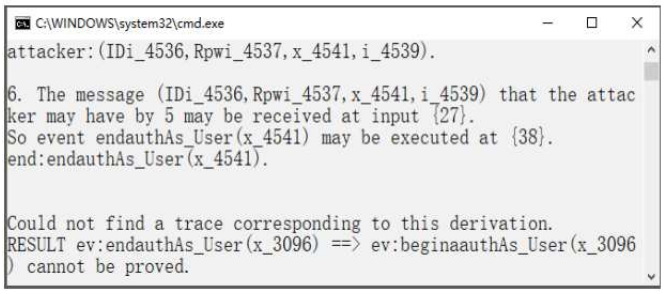

Figure 10: The authentication result of AS to User

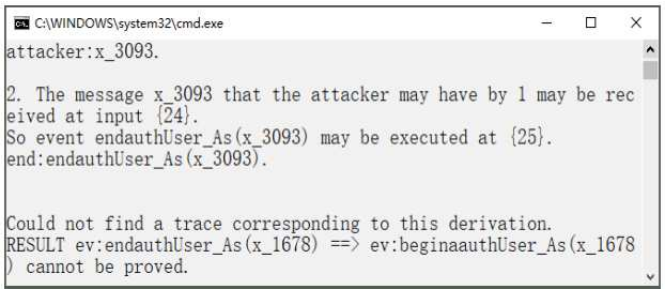

Figure 11: The authentication result of User to AS

3.3.3 Privacy

Here the privacy is modeled as the confidentiality of profile. We use query attacker: Cup_i and Cup_i' to model the privacy of Cup_i and Cup_i'. The results of user profile privacy analysis are shown in Figures 12 and 13. The results of Cup_i and Cup_i' are true, indicating that the privacy of PPMUAS has been implemented. Because Cup_i and Cup_i' are ciphertext containing private information, and they have confidentiality.

4 PPUAPBDE Protocol

The PPMUAS protocol claimed to be privacy protection while achieving remote user authentication. But with the analysis of section III, we find that it has not supported the one-way authentication from AS to DB and

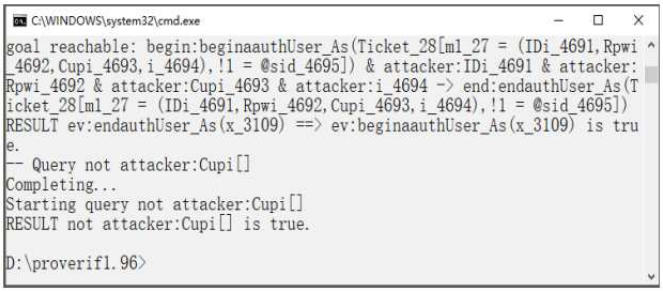

Figure 12: Privacy analysis of Cup_i

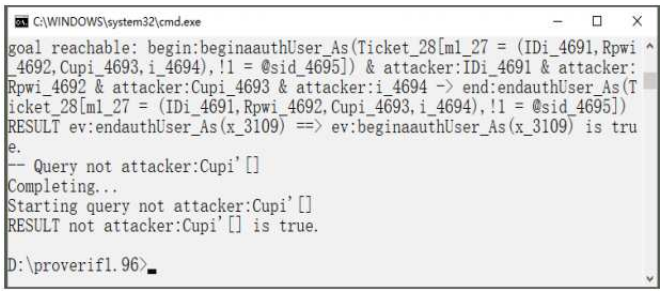

Figure 13: Privacy analysis of Cup_i'

the mutual authentication between User and AS. In order to achieve remote user authentication and privacy, the PPUAPBDE protocol is proposed based on user profile which contains the behavioral features of user mouse movement and keystroke, system processes, and network as user authentication factors. At the same time, it applies homomorphic encryption and fuzzy hash to protect the security and privacy of authentication factors. The fuzzy hash scheme can solve the avalanche effect of the normal hash algorithm. PPUAPBDE protocol provides the mutual authentication between User and AS, the authentication from DB to AS and the authentication from User to DB.

4.1 Message of PPUAPBDE

The PPUAPBDE protocol includes three roles: User, Authentication Server (AS) and Database (DB) and is composed of the registration phase and login phase. In the registration phase, User generates the encrypted user profiles using homomorphic signcryption and fuzzy hash with the behavioral patterns of user mouse movement and keystroke, system processes, and network behavior. And then the encrypted user profiles are sent to DB. In the login phase, AS verifies password and regenerates profiles, and then the regenerated profiles are sent to DB. DB compares the user profiles produced in the Login phase and in the registration phase and produces a result. If the result is higher than the preset threshold value, the User is allowed to log in, otherwise denied. The PPUAPBDE protocol mainly uses signcryption [3], fuzzy hash [22] and homomorphic encryption(HE) [8]. Notations in PPUAPBDE protocol are explained in Table 2.

The messages of PPUAPBDE protocol are shown in Figure 14. It has two phases, including eight messages. Registration phase includes Registration request, Authentication Server request, and Registration response messages. Login phase includes Login request, Authentication Server login request, Database response, User grant, and Login response messages.

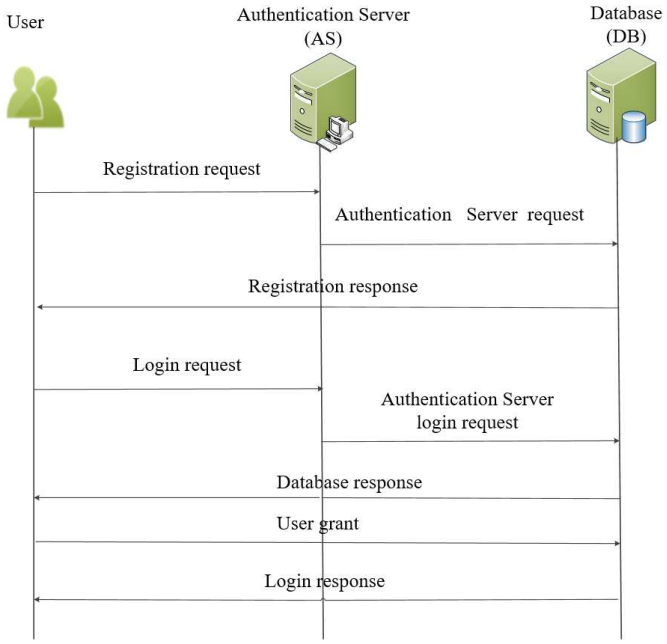

Figure 14: PPUAPBDE messages

Registration request: When User registers, it sends message 1 to AS:

——***Registration request***: $t_1, c_{reg}, R_{reg}, s_{reg}$ ——

Message 1 is composed of timestamp t_1, signcryption parameters c_{reg}, R_{reg} and s_{reg}. The calculation steps of signcryption parameters are shown in Table 3.

Authentication Server request: After AS receives message 1, it verifies whether the timestamp t_1 is in the time skew T or not. If t_1 exceeds T, the message 1 is a replay message, and is ignored, otherwise, $r_{reg} \cdot D$ will be compared with R_{reg}, which is shown in Table 4.

If $r_{reg} \cdot D$ and R_{reg} are equal, User password hash value h_{Pw} will be saved locally. And then, message 2 will be generated:

——**Authentication Server request**:t_2,c_{reg}',R_{reg}',s_{reg}'——

It includes timestamps t_2, signcryption parameters C_{reg}', R_{reg}' and s_{reg}'. C_{reg}', R_{reg}' and s_{reg}' are calculated , which is shown in Table 5.

Finally, message 2 is sent to DB.

Table 2: Notations and definitions

Symbol	Definition	Symbol	Definition
UP_i	Encrypted User Profile	Dis_{FHE}	Metric-data Characteristic deviation
C	Signcryption parameter	$\mathrm{Dis}_{FuzzyHash}$	String Characteristic deviation
R	Signcryption parameter	t_i	Timestamp
S	Signcryption parameter	T	Effective Time Period
PU_i,PR_i	Public/Private Key of HE	c	Public Channel
$FuzzyHash$	Fuzzy Hash Function	$\triangle t_1$	String Characteristic deviation
P_i,d_i	Public/Private Key of Signcryption	$\triangle t_2$	Metric-data Characteristic deviation

Table 3: Registration request calculation

Step1: $(PU_{UID}, PR_{UID} = THE_{keygen}$.
Step2: $d_{UID} \in (0,1,2,\cdots,p-1)$, $PU_{UID} = d_{UID} \cdot D$, $D \in n$, n is a large prime number.
Step3: $UP_{reg} = \{FuzzyHash(pw_i), FuzzyHash(U\_OlInfo), FuzzyHash(U\_BrInfo), FH\_Enc(U\_KeySD), FH\_Enc(U\_AMM), FH\_Enc(U\_GM)$
Step4: $k_1 = h(k \cdots D), k_2 = h(k \cdot P_{AS}), k \in (1,2,\cdots,n-1)$.
Step5: h_{Pw}=hash(Password).
Step6: c_{reg}=E((h_{Pw},UID,UP_{reg}),k_2), UID is user id.
Step7: r_{reg}=hash(k_1,h_{Pw}).
Step8: $s_{reg} = \frac{k}{r_{reg}+d_{UID}} \bmod n$.
Step9: $R_{reg} = r_{reg}D$.

Table 4: AS request comparison

step1: choose $\mathrm{d}_{AS} \in$ (0,1,2...p-1), P_{AS}=d_{AS} D.
step2: k_1=hash(s_{reg}(P_{UID}+R_{reg})).
step3: k_2=hash(s_{reg}(d_{AS}(P_{UID}+R_{reg}))).
step4: (h_{PW},UID,UP_{reg})=D(c_{reg},k_2).
step5: r_{reg}=hash(k_1,$\mathrm{h}_p w$).
step6: r_{reg} D?=R_{reg}.

Table 5: AS request calculation

step1: choose k, k $\in$ (0,1,2...n-1).
step2: $k_3 = h(kD), k_4 = h(kP_{DB})$.
step3: c_{reg}'=E((UID,UP_{reg}),k_4).
step4: r_{reg}'=hash(k_3,UID).
step5: $s'_{reg} = (\frac{k}{r'_{reg}+d_{AS}}) \bmod n$.
step6: $R'_{reg} = r'_{reg}D$.

Table 6: Registration responce calculation

step1: $d_{DB} \in (0,1,\cdots,n-1), P_{DB} = d_{DB}D$.
step2: k_3=hash(s_{reg}'(P_{AS}+R_{reg}')).
step3: k_4=hash(s_{reg}'(d_{DB}(P_{AS}+R_{reg}'))).
step4: (UID,UP_{reg})=D(s_{reg}',k_4).
step5: r_{reg}'=hash(k_3,UID).
step6: $r'_{reg}D = R'_{reg}$.

Registration response: After DB receives message 2, and then it verifies whether the timestamp t_2 is in the time skew T or not. If t_2 exceeds T, message 2 is a replay message, and is ignored, otherwise, r_{reg}' $\cdot D$ and R_{reg}' are calculated, which is shown in Table 6.

If r_{reg}' $\cdot D$ and R_{reg}' are equal, the User profile UP_{reg} will be saved locally. And then, message 3 will be generated:

——Registration responce: *Yes*——

Where the message "Yes" means that the User registers successfully. And then, message 3 is sent to AS.

Login request: When User enters id and password to log in, UP_{login}, c_{login}, R_{login}, s_{login} will be generated just like the calculation in the registration phase. And then, message 4 will be created:

——Login request: t_3, c_{login}, R_{login}, s_{login}——

where it contains a timestamp t_3, signcryption parameter c_{login}, R_{login}, s_{login}. Finally, message 4 is sent to AS.

Authentication Server login request: After AS receives message 4, and then it verifies whether the timestamp t_3 is in the time skew T or not. If t_3 exceeds T, message 4 is a replay message and is ignored. Otherwise, $r_{login} \cdot D$ is compared with R_{login}. If they are equal, we will continue to compare the User password hash value h_{Pw}' saved in login phase and the

User password hash value h_{Pw} saved locally. If h_{Pw}' and h_{Pw} are equal, message 5 is generated, too.

——AS Login request: t_4,c_{login}', R_{login}', s_{login}'——

Where it contains timestamp t_4, Signcryption parameters c_{login}', R'_{login}, s'_{login}. Signcryption parameters c'_{login}, R'_{login}, s'_{login} are produced by the same method in Registration phrase. Finally, message 5 is sent to DB. If h_{Pw}' and h_{Pw} are not equal, User will be requested to login again.

Database response: After DB receives message 5, and then it verifies whether the timestamp t_4 is in the time skew T or not. If t_4 exceeds T, Authentication Server login request message is ignored, otherwise, r_{login}'D is compared with R_{login}'. If they are equal, the following computation will be executed.

$$
\begin{aligned}
&Dis_{FuzzyHash} = \\
&\{Fuzzycmp(FuzzyHash(U\_SFP), \\
&FuzzyHash(U'\_SFP)), \\
&Fuzzycmp(FuzzyHash(U\_SP), \\
&FuzzyHash(U'\_SP)), \\
&Fuzzycmp(FuzzyHash(U\_NS), \\
&FuzzyHash(U'\_NS)).\}
\end{aligned}
$$

TThus, we get the String Characteristic Deviation $Dis_{FuzzyHash}$ of the login phase profile UP_{reg} and the registration phase profile UP_{login}. If $Dis_{FuzzyHash}$ is in the range of deviation Δt_1, we continue to calculate the Metric-data Characteristic deviation Dis_{FHE} of the User profile.

$$
\begin{aligned}
&Dis_{FHE} = \\
&\{FHE_{sub}(FHE(U\_AFI, PU_{UID}), \\
&FHE(U'\_AFI, PU_{UID}), PU_{UID}), \\
&FHE_{sub}(FHE(U\_MI, PU_{UID}), \\
&FHE(U'\_MI, PU_{UID}), PU_{UID}), \\
&FHE_{sub}(FHE(U\_KI, PU_{UID}), \\
&FHE(U'\_KI, PU_{UID}), PU_{UID}).\}
\end{aligned}
$$

Finally, DB will sign the Dis_{FHE} with the private key of DB. Thus, the result is sent to User:

——Database responce: Sign(Dis_{FHE})——

User grant: When User receives the response message 6, the signature is verified first. And then, Dis_{FHE} is decrypted by the private key of Homomorphic Encryption to get the plaintext of User profile Metric-data Characteristic deviation Dis_{FHE-pt}. After that the deviation is sent to DB after signing with the private key of User:

——User grant: Sign(Dis_{FHE-pt})——

Finally, the message 7 is sent to DB.

Login response: When DB receives message 7, it verifies the signature to get Dis_{FHE-pt}. Then, it compares Dis_{FHE}' with Dis_{FHE-pt}, PU_{UID}. If they are equal, we continue to compare whether Dis_{FHE-pt} is in the range of deviation Δt_2. If this condition is met, it means that the user is legitimate. And message 8 will be generated. Otherwise, the User is rejected.

——Login responce:Ticket)——

4.2 Formal Modeling of PPUAPBDE

In this section, the Applied PI calculus is used to describe the PPUAPBDE protocol formally, and the non-injective and Query are used to model the authentication and confidentiality. Then the software tool ProVerif is used to formalize and prove the confidentiality and authentication of the PPUAPBDE. Finally, the confidentiality of user privacy information is analyzed, and the privacy analysis results of the PPUAPBDE protocol is given.

4.2.1 Function and Equational Theory

Function and equation theory mainly contains public-key encryption algorithm *E(x, Pu)* and decryption algorithm *D(y, Pu)*. The public key encryption algorithm *E(x, Pu)* encrypts the message x using the public key Pu to generate the ciphertext. The decryption algorithm *D(y, Pu)* decrypts the message y using the public key Pu to obtain the plaintext. The sign algorithm Sign*(x,Pr(y))* signs the message x with private key *Pr(y)*, while the signature is verified with public key *Pu(y)*. Its formal modeling is shown in Figure 15.

```
fun FuzzyHash/1.
fun hash/1.
fun FHE/2.
fun E/2.
fun D/2.
fun mod/2.
fun FHE_keygen/0.
fun Fuzzycmp/2.
fun h/1.
fun Pu/1.
fun Pr/1.
fun sign/2.
fun versign/2.
fun FHE_Decrypt/2.
fun FHE_sub/2.
equation D(E(x,Pu(y)),Pu(y)) = x.
equation versign(sign(x,Pr(y)),Pu(y)) = x.
equation FHE_Decrypt(FHE_sub((FHE(x,PU(y)),
FHE(y,PU(y))),Pu(y)),Pr(y)) = x - y.
```

Figure 15: Function and equation theory

4.2.2 Process

The PPUAPBDE protocol is composed of *processUser*, *processAS* and *processDB*. The three processes constitute the main process, as shown in Figure 16.

$$\begin{bmatrix} Mainprocess \\ (!\,processUser|!\,processAS|!\,processDB) \end{bmatrix}$$

Figure 16: PPUAPBDE mainprocess

We model the three processes using Applied PI calculus. The *processUser* all processes are shown in Figures 17, 18 and 19, respectively.

4.3 Security Analysis of PPUAPBDE

In this section, the security analysis results of PPUAPBDE are given. The results indicate that PPUAPBDE has implemented confidentiality, authentication, and privacy.

4.3.1 Confidentiality

In this section, the confidentiality of user password Psw_i is analyzed. And we use query attacker: Password in ProVerif to verify the confidentiality of the user password Psw_i. The result of query attacker: Password is shown in Figure 20. The result is "true" to prove that the confidentiality of Password has been satisfied. This is because the User uses the hash function to process the Password to get the hash value, and the hash value is sent to the AS. Owning to the one-way property of the hash function, the attacker cannot obtain the User password plaintext.

4.3.2 Authentication

The non-injective agreement is used to model authentication in ProVerif. The authentication model of PPUAPBDE is the same as shown in Table 1.

The authentication results between User and AS are shown in Figures 21 and 22. The results of both are true, indicating that the mutual authentications between AS and User have been achieved. This is because the Login request message and Registration request message sent by User to AS, are signed with their private keys. Only the signcryption public key can verify the signature. Therefore, User can authenticate AS mutually.

In Figure 23, User authenticates DB. In Figure 24, DB authenticates AS. The results of both are true, indicating that User can authenticate DB and DB can authenticate AS. The Registration response message is handled by the signcryption function. And the Authentication server Login request message and Authentication server request message sent by AS to DB, are signed with their private keys, too.

$$\begin{bmatrix}
new\ t_1; \\
new\ UID; \\
new\ Password; \\
new\ U_{_SFP}; \\
new\ U_{_SP}; \\
new\ U_{_NS}; \\
new\ U_{_AFI}; \\
new\ U_{_MT}; \\
new\ U_{_KI}; \\
let\ UP_{reg} = \begin{bmatrix} FuzzyHash(U_{_SFP}), FuzzyHash(U_{_SP}), FuzzyHash(U_{_NS}) \\ FHE(U_{_AFI}, PU_{UID}), FHE(U_{_MT}, PU_{UID}), FHE(U_{_KI}, PU_{UID}) \end{bmatrix} in \\
let(PU_{UID}, PR_{UID}) = FHE_keygen\ in \\
in(c, P_{AS}); \\
\cdots \\
let\ c_{reg} = E((h_{pw}, UID, UP_{reg}), k_1)\ \ in \\
\cdots \\
let\ s_{reg} = (\frac{k}{r_{reg} + d_{UID}}) \bmod\ n\ in \\
\cdots \\
out(c, m_1); //\ registration\ phase \\
\\
new\ t_3; \\
new\ Password'; \\
new\ U_{_SFP}'; \\
new\ U_{_SP}'; \\
new\ U_{_NS}'; \\
new\ U_{_AFI}'; \\
new\ U_{_MT}'; \\
let\ UP_{login} = \begin{bmatrix} FuzzyHash(U_{_SFP}'), FuzzyHash(U_{_SP}'), FuzzyHash(U_{_NS}') \\ FHE(U_{_AFI}', PU_{UID}), FHE(U_{_MT}', PU_{UID}), FHE(U_{_KI}', PU_{UID}) \end{bmatrix} in \\
\cdots \\
out(c, m_3); \\
in(c, Ticket). //\ \log in\ phase
\end{bmatrix}$$

Figure 17: PPUAPBDE processUser

4.3.3 Privacy

Here the privacy is modeled as the confidentiality of profile. We use query attacker: UP_{reg} and UP_{login} to model the privacy of UP_{reg} and UP_{login}.The results of user profile privacy analysis are shown in Figures 25 and 26. The results of UP_{reg} and UP_{login} are true, indicating that it has User profile privacy of UP_{reg} and UP_{login}. This is because the homomorphic signcryption adopted by the PPUAPBDE is to provide privacy protection of user profile UP_{reg} and UP_{login}.

5 Evaluation and Analysis

In order to evaluate the performance of PPUAPBDE protocol, we develop an authentication system with Visual Studio Community 2015 and MySQL Community 6.3. The signcryption scheme is the open-source software library [9]. The fuzzy hash is the sdhash proposed by Roussev [22]. Homomorphic encryption is Microsoft's Open Encrypted Arithmetic Library project [20], which

$$
\begin{aligned}
&in(c, m_1);\\
&in(c, P_{UID});\\
&in(c, P_{DB});\\
&let(t_1, c_{reg}, R_{reg}, s_{reg}) = m_1 \ in\\
&let\ k_1 = hash(s_{reg}(P_{UID} + R_{reg}))\ in\\
&let\ k_2 = hash(s_{reg}(d_{AS}(P_{UID} + R_{reg})))\ in\\
&let\ (h_{pw}, UID, UP_{reg}) = D(c_{reg}, k_2)\ in\\
&if\ R_{reg} = hash(k_1, h_{pw}) \times D\ then\\
&new\ t_2\\
&let\ k_3 = h(k \times D)\ in\\
&let\ k_4 = h(k \times P_{DB})\ in\\
&let\ c_{reg}' = E((UID, UP_{reg}), k_4)\ in\\
&let\ r_{reg}' = hash(k_3, UID)\ in\\
&let\ s_{reg}' = (\frac{k}{r_{reg}' + d_{AS}}) \bmod n\ in\\
&let\ R_{reg}' = r_{reg}' \times D\\
&let\ m_2 = (t_2, c_{reg}', R_{reg}', s_{reg}')\ in\\
&out(c, m_2);//registration\ phase
\end{aligned}
\qquad
\begin{aligned}
&in(c, m_3);\\
&let\ (t_1, c_{\log in}, R_{\log in}, s_{\log in}) = m_3\ in\\
&\cdots\\
&if\ R_{\log in} = r_{\log in} \times D\ then\\
&let\ c_{\log in}' = E(UID, k_4)\ in\\
&let\ r_{\log in}' = hash(k_3, UID)\ in\\
&let\ s_{\log in}' = (\frac{k}{r_{\log in}' + d_{AS}}) \bmod n\ in\\
&let\ R_{\log in}' = r_{\log in}' \times D\\
&let\ m_4 = (t_4, c_{\log in}', R_{\log in}', s_{\log in}')\ in\\
&\cdots\\
&out(c, m_4).//\log in\ phase
\end{aligned}
$$

Figure 18: PPUAPBDE processAS

$$
\begin{aligned}
&new\ t;\\
&in(c, m_2);\\
&in(c, P_{AS});\\
&let\ (t_2, c_{reg}', R_{reg}', s_{reg}') = m_2\ in\\
&let\ k_3 = hash(s_{reg}'(P_{AS} + R_{reg}'))\ in\\
&let(UID, UP_{reg}) = D(c_{reg}', k_4)\ in\\
&\cdots\\
&if\ R_{reg}' = hash(k_3, UID) \times D\ then\\
&out(c, yes);//registration\ phase\\
&\\
&in(c, m_4);\\
&let\ (t_4, c_{\log in}', R_{\log in}', s_{\log in}') = m_4\ in\\
&\cdots\\
&if\ R_{\log in}' = c_{\log in}' \times D\ then\\
&let\ Dis_{FuzzyHash} = \begin{bmatrix} Fuzzycmp(FuzzyHash(U_{\_SFP}), FuzzyHash(U_{\_SFP}')) \\ Fuzzycmp(FuzzyHash(U_{\_SP}), FuzzyHash(U_{\_SP}')) \\ Fuzzycmp(FuzzyHash(U_{\_NS}), FuzzyHash(U_{\_NS}')) \end{bmatrix} in\\
&let\ Dis_{FHE} = \begin{bmatrix} FHE_{stib}(FHE(U_{\_AFI}, PU_{UID}), FHE(U_{\_AFI}', PU_{UID}'), PU_{UID}) \\ FHE_{stib}(FHE(U_{\_MT}, PU_{UID}), FHE(U_{\_MT}', PU_{UID}), PU_{UID}) \\ FHE_{stib}(FHE(U_{\_KI}, PU_{UID}), FHE(U_{\_KI}', PU_{UID}), PU_{UID}) \end{bmatrix} in\\
&\cdots\\
&newTicket;\\
&out(c, Ticket);//\log in\ phase
\end{aligned}
$$

Figure 19: PPUAPBDE processDB

Figure 20: The confidentiality of Psw_i

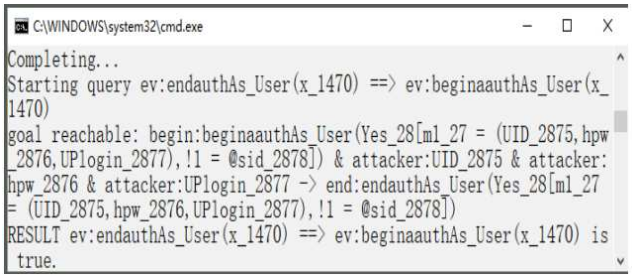

Figure 21: AS authenticates User

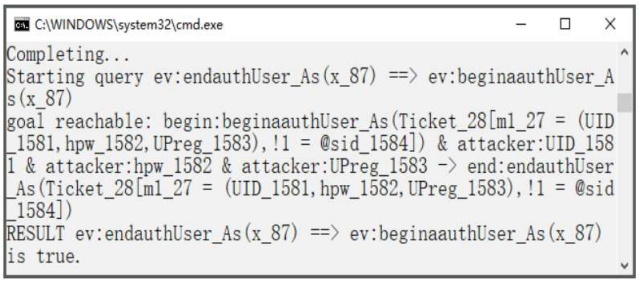

Figure 22: User authenticates AS

is an easy-to-use but powerful homomorphic encryption library that is called via API. We choose two closely related important parameters: Recall and FPR to evaluate the PPUAPBDE authentication system. The Recall R=TP/(TP+FN) represents the probability that a legitimate user will successfully log in to the system, in which TP is the number of times a legitimate user attempts to log in, and FN represents the number of times an illegal user attempts to log in. FPR that is, the probability that the system will misjudge the attacker as a legitimate user. We select 663 students in a university as a user to evaluate the Recall and FPR in 31 days. The related evaluation data is shown in Table 7, and the data in the table is taken from the database log. The experimental results are shown in Figure 27. The Recall is 94.8%. The last FPR is 5.1%. The cloud-based privacy protection mobile user authentication system PPMUAS based on big data characteristics described in [24] has a Recall of 84.9% and an FPR of 12.6%. The Recall value continues to increase, and the FPR value continues to decrease, indicating that the PPUAPBDE can be used in the authentication protocol effectively. If we want to put the experimental authentication system into practice, there is a work on using

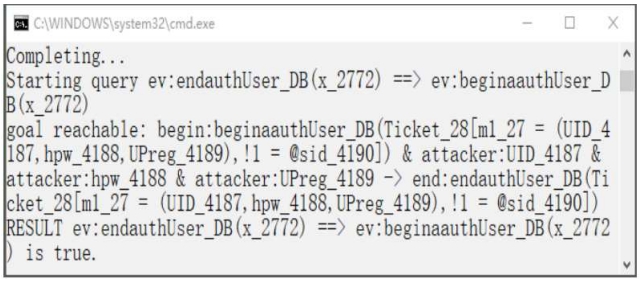

Figure 23: User authenticates DB

Figure 25: Privacy analysis of UP_{reg}

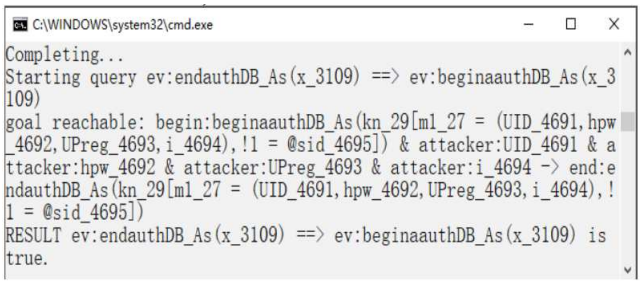

Figure 24: DB authenticates AS

Figure 26: Privacy analysis of UP_{login}

better algorithms for collecting dynamic behavioral features to improve the FPR.

6 Conclusion

Many authentication protocols used private information as authentication factors to implement user authentication. The authentication server can get private data. Therefore, with the rapid development of data mining technology, there exists a risk of private information leakage in the authentication server. In general, there are two methods to protect privacy. One is that the authentication server uses privacy protection technology, such as data anonymity, data distortion, or cryptography, to protect privacy. This method depends on authentication server cooperation. The other is to develop the privacy-preserving authentication protocol that provides not only user authentication but also privacy preservation of private data. The second method is adopted mostly as it solves the problem fundamentally. The privacy information is protected before being sent to the authentication server. Hence, we analyzed the authentication and confidentiality of PPMUAS protocol using Applied PI calculus and found it has some security vulnerabilities. In order to implement remote authentication and privacy of user information, the PPUAPBDE protocol is proposed based on user profile which contains behavioral patterns of user mouse movement and keystroke, system processes, and network as user authentication factors. At the same time, it applies homomorphic encryption and fuzzy hash to protect the security and privacy of authentication factors. And then, the confidentiality, authentication and privacy of the PPUAPBDE protocol are analyzed using Applied PI calculus. The results show that the confidentiality, authentication, and privacy of PPUAPBDE have been implemented. Finally, based on the PPUAPBDE protocol, we design, implement and test the PPUAPBDE authentication system. The Recall is about 94.8%, and FPR is 5.1%, which are better than PPMUAS protocol. The scheme has not been put into practical use yet, only for testing the actual effect of multi-behavior features to protect user privacy in the authentication protocol. PPMUAS was proposed by Vorupunti *et al.* in 2017, and they claimed that it is the first privacy-preserving remote user authentication. Hence, compared to PPMUAS protocol from the key three aspects:

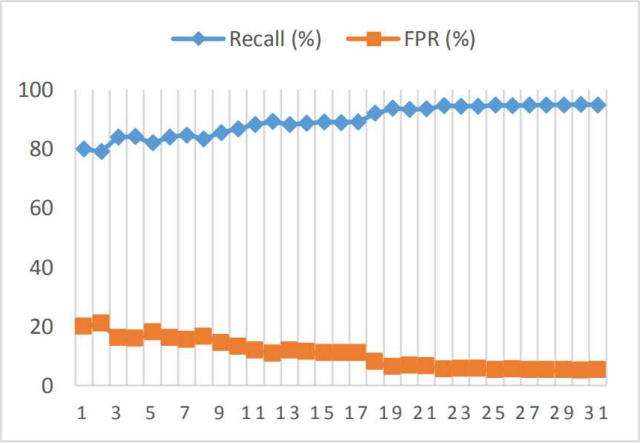

Figure 27: Experimental results

- Authentication factor. PPUAPBDE uses multi-factor mechanisms: User mouse movement and keystroke, system processes, and network as authentication factors, while PPMUAS only use a single factor;;

Table 7: Related evaluation data

Day	TP	FN	Recall(%)	FPR(%)
1	531	132	80.0	19.9
2	524	139	79.0	20.9
3	557	106	84.0	15.9
4	558	105	84.1	15.8
5	544	119	82.0	17.9
6	557	106	84.0	15.9
7	561	102	84.6	15.3
8	553	110	83.4	16.5
9	567	96	85.5	14.4
10	576	87	86.8	13.1
11	585	78	88.2	11.7
12	592	71	89.2	10.7
13	585	78	88.2	11.7
14	588	75	88.6	11.3
15	591	72	89.1	10.8
16	590	73	88.9	11.0
17	591	72	89.1	10.8
18	610	53	92.0	7.9
19	621	42	93.7	6.2
20	619	44	93.3	6.6
21	619	44	93.4	6.5
22	627	36	94.6	5.3
23	626	37	94.4	5.5
24	626	37	94.4	5.5
25	628	35	94.7	5.2
26	627	36	94.6	5.3
27	628	35	94.7	5.2
28	629	34	94.8	5.1
29	629	34	94.8	5.1
30	630	33	95.0	4.9
31	629	34	94.8	5.1

- Authentication. PPUAPBDE protocol provides the mutual authentication between User and AS, the authentication from DB to AS and the authentication from User to DB, while PPMUAS does not provide authentication from AS to DB and the mutual authentication between User and AS;

- Recall and FPR. Recall and the FPR of the PPUAPBDE authentication system are tested, and the results show that PPUAPBDE is better than PPMUAS.

There are open issues in the PPUAPBDE protocol. For example, the Recall of the PPUAPBDE authentication system is not high enough. In the future, we can establish a contour model for the normal behavior of legitimate users, and then use the similarity curve of the behavior profile as the credential to authenticate the legitimate user to improve the Recall.

Acknowledgments

The research is supported in part by the Fundamental Research Funds for the Central Universities, South-Central University for Nationalities No. CZZ21001 and QSZ17007, and in part by the natural science foundation of Hubei Province under the grants No.2018ADC150.

References

[1] M. Abadi and C. Fournet, "Mobile values, new names, and secure communication," in *Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, vol. 36, no. 3, pp. 104–115, 2001.

[2] M. Babaeizadeh, "Keystroke dynamic authentication in mobile cloud computing," *International Journal of Computer Applications*, vol. 90, no. 1, pp. 29–36, 2014.

[3] S. M. Bellovin, P. Gutmann and M. Blaze, "An IBE-based signcryption scheme for group key management," *arXiv: Cryptography and Security*, 2016. arXiv:1603.09526.

[4] B. Blanchet, "An efficient cryptographic protocol verifier based on prolog rules," in *Proceedings of 14th IEEE Computer Security Foundations Workshop*, pages 82–96, 2001.

[5] Z. F. Cao, "New development of cryptography (in Chinese)," *Advanced Engineering Sciences*, vol. 47, no. 1, pp. 1–12, 2015.

[6] Y. C. Chen, W. L. Wang, M. S. Hwang, "RFID authentication protocol for anti-counterfeiting and privacy protection", in *The 9th International Conference on Advanced Communication Technology*, pp. 255–259, 2007.

[7] E. Cherrier, J. Hatin and J. Schwartzmann, "Privacy preserving transparent mobile authentication," in *The 3rd International Conference on Information Systems Security and Privacy*, pp. 354–361, 2017.

[8] C. Gentry, M. Dijk and S. Halevi, "Fully homomorphic encryption over the integers," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 24–43, 2010.

[9] A. Ghani, *et al.*, S. Ashraf, A. Irshad, "An efficient signcryption scheme with forwarding confidentiality and public verifiability based on hyper elliptic curve cryptography," *Multimedia Tools and Applications*, vol. 74, no. 5, pp. 171–173, 2015.

[10] S. Goldwasser and S. Micali, "Probabilistic encryption & how to play mental poker keeping secret all partial information," in *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, pp. 365–377, 1982.

[11] B. Gras, C. Gamage and B. Crispo, "An identity-based ring signature scheme with enhanced privacy," *Second International Conference on Security and Privacy in Communication Networks and the Workshops*, pp. 1–5, 2006.

[12] C. C. Lee, C. H. Liu, M. S. Hwang, "Guessing attacks on strong-password authentication protocol", *International Journal of Network Security*, vol. 15, no. 1, pp. 64–67, 2013.

[13] Y. Lee, S. Kang and I. Cho, "ECG authentication system design based on signal analysis in mobile and wearable devices," *IEEE Signal Processing Letters*, vol. 23, no. 6, pp. 805–808, 2016.

[14] C. T. Li, M. S. Hwang, Y. P. Chu, "Further improvement on a novel privacy preserving authentication and access control scheme for pervasive computing environments", *Computer Communications*, vol. 31, no. 18, pp. 4255–4258, Dec. 2008.

[15] C. T. Li, M. S. Hwang, Y. P. Chu, "A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks", *Computer Communications*, vol. 31, no. 12, pp. 2803-2814, July 2008.

[16] R. H. Li and Y. Z. Li, "Research on behavior characteristics in dynamic continuous identity authentication system," (in Chinese), *Computer and Digital Engineering*, vol. 46, no. 1, pp. 138–143, 2018.

[17] W. Lou, K. Ren and K. Kim, "A novel privacy preserving authentication and access control scheme for pervasive computing environments," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 4, pp. 1373–1384, 2016.

[18] H. Jeong and E. Choi, "User authentication using profiling in mobile cloud computing," *AASRI Procedia*, vol. 2, pp. 262–267, 2012.

[19] G. Kambourakis and D. Damopoulos, "Introducing touchstroke: keystroke-based authentication system for smartphones," *Security & Communication Networks*, vol. 9, no. 6, pp. 542–554, 2016.

[20] Microsoft/SEAL. (`https://github.com/microsoft/seal`)

[21] A. Rassan and H. Alshaher, "Securing mobile cloud computing using biometric authentication (SMCBA)," in *International Conference on Computational Science and Computational Intelligence*, vol. 1, pp. 157–161, 2014.

[22] V. Roussev, "An evaluation of forensic similarity hashes," *Digital Investigation*, vol. 8, pp. S34-S41, 2011.

[23] S. Ruj and M. Stojmenovic, "Privacy preserving access control with authentication for securing data in clouds," in *The 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 556–563, 2012.

[24] C. Vorugunti, "PPMUAS: A privacy preserving mobile user authentication system for cloud environment utilizing big data features," in *IEEE International Conference on Advanced Networks and Telecommunications Systems*, pp. 1–6, 2016.

[25] C. H. Wei, M. S. Hwang, Augustin Y. H. Chin, "A secure privacy and authentication protocol for passive RFID tags," *International Journal of Mobile Communications*, vol. 15, no. 3, pp. 266–277, 2017.

[26] L. Yao, J. Lu and X. He, "A security analysis method for security protocol implementations based on message construction," *Applied Sciences*, vol. 8, no. 12, pp. 2543, 2018.

[27] L. W. Zhang, S. X. Fang and H. L. Wang, "Research on the development trend of identity authentication technology based on face biometrics," *Information Security Research*, vol. 003, no. 006, pp. 533–537, 2017.

Biography

Jiabing Liu. was born in 1994 and is now a postgraduate at the School of Computer Science, South-Central University for Nationalities. His research interests include Blockchain and Smart contract security.

Xudong He. was born in 1991 and is now a postgraduate at school of Computer Science, South-Central University for Nationalities. His research interests include: Security protocol implementations and reverse engineering.

Huoye Tang. was born in 1991 and is now a postgraduate at school of Computer Science, South-Central University for Nationalities. His research interests include security protocol implementations and reverse engineering.

Dejun Wang. was born in 1974 and received his Ph.D. in information security at Wuhan University in China. Currently, he is an associate professor in the school of computer, South-Center University for Nationalities, China. He has authored/coauthored over 20 papers in international/national journals and conferences. His current research interests include security protocols and formal methods.

Bo Meng. was born in 1974 in China. He received his M.S. degree in computer science and technology in 2000 and his Ph.D. degree in traffic information engineering and control from Wuhan University of Technology at Wuhan, China in 2003. From 2004 to 2006, he worked at Wuhan University as a postdoctoral researcher in information security. Currently, he is a full Professor at the school of computer, South-Center University for Nationalities, China. He has authored/coauthored over 50 papers in International/National journals and conferences. In addition, he has also published a book "secure remote voting protocol" in the science press in China. His current research interests include Cyberspace security.