An improved and efficient mutual authentication scheme for session initiation protocol

Yuanyuan Zhang, Kunming Xie, Ou Ruan*

School of Computer Science, Hubei University of Technology, Wuhan, CHINA

* ruanou@163.com

Abstract

Qiu et al. made a security analysis about the protocols of Chaudhry et al. and Kumari et al. in 2018, and they pointed out that there are many security weaknesses in the protocols. To improve the security, Qiu et al. proposed an advanced authentication scheme for Session Initiation Protocol on the basis of the previous protocols and claimed that their own protocol was very secure and practical. However, we demonstrate that the protocol of Qiu et al. has a serious mistake which causes their protocol cannot be executed normally. Beyond that, we also find out that their protocol cannot withstand insider attack and denial service attack. In order to remove these weaknesses, we propose an efficient provably secure mutual authentication scheme. Furthermore, our scheme provides security analysis with the help of Burrows-Abadi-Needham (BAN) logic. Compared with their protocol, ours has greater security and better performance.

1 | Introduction

With the network of universal gradually, we often send messages, make phone calls and watch videos via the network, which brings great convenience to our lives. However, the network security issues become increasingly prominent. In order to enhance the security of the Session Initiation Protocol (SIP) in communication via the network, lots of scholars have proposed numerous solutions. The first authentication scheme about SIP on the basis of Hypertext Transport Protocol (HTP) authentication was proposed in 1999 [1], but Yang et al. [2] proved that the scheme was insecure in 2005, then they proposed an improved scheme. In fact, the scheme of Yang et al. still has security loopholes. Latterly, although some scholars want to design secure and practical schemes [3–11], the most schemes have more or less flaws.

In recent years, Zhang et al. [12] proposed a remote mutual authentication protocol with protecting user anonymity for SIP, but there are also lots of loopholes in their protocol too. It was as expected that Lu et al. [13] showed that their protocol lacks mutual authentication and cannot resist insider attack in 2016. In order to remove these weaknesses, Lu et al. proposed a new scheme protocol based on the protocol of Zhang et al., but it still had fatal weaknesses in security. Chaudhry et al. [14] found that the scheme of Lu et al. is invalid to impersonation...
attack. Besides, Kumari et al. [15] showed that the scheme of Lu et al. cannot withstand identity guessing attack and forgery attack. According to their conclusion, Chaudhry et al. proposed a mutual authentication protocol, Kumari et al. proposed an authentication protocol too. However, on the basis of the schemes of Chaudhry et al. and Kumari et al., Qiu et al. [16] demonstrated their schemes are frail for some attacks including insider attack, off-line guessing attack and so on. To overcome these defects, Qiu et al. proposed an improved mutual authentication scheme, claiming that the scheme not only could resist above certain attacks, but also adopted a method of generating random numbers instead of time labels to solve the problem that time is difficult to synchronize. We have to admit that the improved scheme of Qiu et al. did remove certain weaknesses. But after our analysis, we find that the scheme of Qiu et al. has a serious mistake which make the scheme cannot execute properly and cannot resist some attacks. Considering the worst condition, the adversary affects user password update even accesses crucial information of a legal user. In order to overcome the weaknesses, we design a new, more secure and high-performance scheme.

We will focus on the rest seven sections to describe our paper, section 2 reviews Qiu et al.’s scheme. In section 3 we pay attention to analyze the weaknesses about Qiu et al.’s scheme. Section 4 describes our proposed scheme. Next section 5 and section 6 mainly provide analysis and proof of security. Section 7 gives the performance comparison among our scheme and the relative schemes of other scholars. Finally, section 8 shows the conclusion of our scheme.

2 | Review of the scheme of Qiu et al.

In this section, we will briefly review Qiu et al.’s scheme [16], which contains three phases: registration phase, login and authentication phase and password update phase. The detailed information about the three phases is shown as follows.

Before we show each phase, the notations which throughout this paper are introduced in Table 1 firstly.

2.1 | Registration phase

During the registration phase, user $U$ and server $S$ do the following operations to finish registering.

A. $U \rightarrow S: \{Id, HId\}$:

A user $U$ registers to server $S$ with his/her identity $Id$, password $Pw$ and secret key $Up$. After that, $U$ completes $VPw = h(Pw||Up)$, $HId = h(Id||VPw)$ and transmits $\{Id, HId\}$ to sever $S$.

| Symbol | Description |
|--------|-------------|
| $U$    | A legitimate user |
| $S$    | The remote server |
| $Id$   | The identity of the $U$ |
| $Pw$   | The password of the $U$ |
| $Up$   | The secret key of the $U$ |
| $Sp$   | The private key of the $S$ |
| $Q_s = S_p \cdot P$ | The public key of the $S$ |
| $h(\cdot)$ | A one-way hash function |
| $\|$ | String concatenation operation |
| $\oplus$ | The bitwise XOR operation |
| $sk$ | The session key between $U$ and $S$ |

Table 1. The notations of the schemes.
B. After receiving \( Id \) and \( HId \), \( S \) will compute \( N=HId \oplus h(S_p) \) and store \( N \) into database.

### 2.2 | Login and authentication phase

If a user \( U \) completed the registration phase successfully and wants to access request to the server \( S \), \( U \) and \( S \) should perform the following steps.

#### A. \( U \rightarrow S: \{B,X,T\} \):

The user \( U \) accesses server \( S \) with his or her identity \( Id \), password \( Pw \) and secret key \( Up \).

After that, \( U \) calculates \( VPw = h(Pw || Up) \), generates a long random number \( r \) and computes \( HId = h(Id || VPw) \), \( W = h(Id || Up) \), \( X = r \cdot P \), \( Y = r \cdot Q_s \), \( B = W \oplus h(HId || Y) \) and \( T = h(B || Y || W) \).

After completion of calculation, \( U \) sends \( \{B,X,T\} \) to server \( S \).

#### B. \( S \rightarrow U: \{E,Auth_s\} \):

When server \( S \) receives message \( \{B,X,T\} \), it will calculate \( HId = N \oplus h(S_p) \), \( Y = S_p \cdot X \), \( W = B \oplus h(HId || Y) \) and check whether \( T' = h(B || Y || W) \) is equal to \( T \). If not valid, the session will be terminated by \( S \), otherwise server \( S \) generates a long random number \( r' \) and calculates \( E = r' \cdot Q_s \), \( sk_r = r' \cdot Y \), \( Auth_r = h(sk_r || W || Y) \). After that, \( S \) transmits information \( \{E,Auth_s\} \) to \( U \).

#### C. \( U \rightarrow S: \{Auth_u\} \):

Upon getting \( \{E,Auth_s\} \), \( U \) calculates \( sk_u = r \cdot E, Auth_r' = h(sk_u || W || Y) \). Then \( U \) checks whether \( Auth_r' = Auth_r \). If valid, then \( U \) computes \( Auth_u = h(sk_u || W || Y || E) \) and sends result \( Auth_u \) to \( S \), else terminates the session.

#### D. Once receiving the message \( \{Auth_u\} \), \( S \) computes \( Auth_u' = h(sk_u || W || Y || E) \). If \( Auth_u' \) is not equal to \( Auth_u \), \( S \) terminates the session. After that, the user \( U \) communicates with server \( S \) based on the common session key \( sk = sk_u = sk_r = r \cdot r' \cdot Q_s \).

### 2.3 | Password update phase

For a legitimate user \( U \), if he or she wants to change own password \( Pw \) for some reasons, the following steps will be performed.

#### A. \( U \rightarrow S: \{V,M\} \):

\( U \) already picks a new password \( Pw_{new} \) and a new secret key \( Up_{new} \), after that he or she inputs identity \( Id \), password \( Pw \) and secret key \( Up \). \( U \) computes \( V = h(sk || h(Id || h(Pw || Up))) \) and \( M = h(Id || sk) \oplus h(Id || h(Pw_{new} || Up_{new})) \), then \( U \) transmits message \( \{V,M\} \) to \( S \).

#### B. After obtaining message \( \{V,M\} \), \( S \) computes \( V' = h(sk || N) \) firstly, then \( S \) judges if the value of \( V' \) is equal to \( V \). If holds, \( S \) will replace \( N \) with \( N_{new} \) by computing \( N_{new} = h(S_p) \oplus h(Id || sk) \oplus M \) into database. If not, \( S \) fails to change password and exits the session.

### 3 | Weakness of scheme proposed by Qiu et al.

In this section, we analyze the weaknesses of Qiu et al.’s scheme [16] carefully. After our study, we find that their scheme has a serious mistake which causes the scheme cannot executed normally. What is more, their scheme cannot resist insider attack, denial service attack and makes user \( U \) have poor experience [17].

#### 3.1 | Serious mistake

In registration phase of Qiu et al.’s scheme, we notice that information \( N \) is stored into database alone. As is known to all, there should be some information like identity \( Id \) correspond to
In this part, we assume that a malicious insider adversary \( \text{Id} \) can observe the entire protocol execution process, so the server cannot store \( \text{Id} \), \( H\text{Id} \) and \( N \) in the database. Or else in login and authentication phase in their scheme, when user \( S \) receives \( \text{Id}^* \), it cannot match corresponding \( N \) in database without the help of the corresponding information. So, the scheme of Qiu et al. is unable to carry out normally.

Perhaps Qiu et al. just forgot corresponding information, here we help them supply corresponding information on the basis of the scheme of Qiu et al. In registration, \( S \) only knows information \( \text{Id}, H\text{Id} \) and \( N \) which relates to \( U \). \( H\text{Id} \) is the most important secret data during the entire protocol execution process, so the server cannot store \( H\text{Id} \) in the database but stores \( \text{Id} \). We assume a semi honest server \( S_0 \) has the ability of gaining and calculating the sensitive information in the sever. If \( \text{Id} \) corresponds to \( N \), we notice that an adversary \( S_0 \) can obtain \( \{\text{Id}^*, U_0^*, Pw^*\} \) of a legitimate user \( U^* \) in login and authentication phase by off-line guessing attack, the specific steps are as follows.

**Step 1:** According to the login and authentication process in the scheme of Qiu et al., \( S_0 \) will get the values of \( H\text{Id}^* \) and \( W^* \) at time of calculating \( H\text{Id}^* = N^* \oplus h(S_p) \) and \( W^* = B^* \oplus h(\text{Id}^* \oplus Y^*) \). Besides, because \( \text{Id}^* \) corresponds to \( N^* \), \( S_0 \) can get the corresponding value of \( \text{Id}^* \).

**Step 2:** After getting user \( U \)'s sensitive information \( \{\text{Id}^*, H\text{Id}^*, W^*\} \), \( S_0 \) can guess the value of \( U_0 \) from the identity space by calculating \( W^* = h(\text{Id}^* || U_0^*) \). According to the same truth, \( S_0 \) can guess the value of \( Pw^* \) from the identity space by calculating \( H\text{Id}^* = h(\text{Id}^* || VPw^*) = h(\text{Id}^* || h(Pw^* || U_0^*)) \). Obviously, \( S_0 \) can access \( \{\text{Id}^*, U_0^*, Pw^*\} \).

Through the above steps, \( S_0 \) has successfully accessed information \( \{\text{Id}^*, U_0^*, Pw^*\} \) of a legal user \( U^* \).

### 3.2 Insider attack

In this part, we assume that a malicious insider adversary \( A \) can obtain some sensitive information in the database of server \( S \). In Qiu et al.’s scheme, the adversary \( A \) can achieve insider attack by registering as a legitimate user. Firstly, he masquerades as a legitimate user in registration to input identity \( \text{Id}^* \), password \( Pw^* \) and secret key \( U_0^* \), then \( S \) will store the corresponding value \( N^* \) into database and he can get the value \( N^* \) form the database of sever \( S \). After that, \( A \) has already mastered the information \( \{\text{Id}^*, U_0^*Pw^*, N^*\} \). On the basis of formulas \( H\text{Id} = h(\text{Id} || h(Pw || U_0)) \) and \( H\text{Id} = N^* \oplus h(S_p) \), \( A \) can get \( h(S_p) = N^* \oplus h(\text{Id}^* || h(Pw^* || U_0^*)) \). In addition, \( A \) can get other user’s \( N_0 \) in the database, so he will get the user’s corresponding \( H\text{Id}_0 \) by formula \( H\text{Id}_0 = N_0 \oplus h(S_p) \). In login and authentication phase, \( A \) can impersonate to be the user \( U_0 \) to access sever \( S \) by corresponding \( N_0 \) and \( H\text{Id}_0 \). The specific steps are as follows.

**A.** \( U_0 \rightarrow S_0 \{B_0, X_0, T_0\} \):
Adversary \( A \) chooses random numbers \( r_0 \), \( W_0 \) and computes \( X_0 = r_0 \cdot P, Y_0 = r_0 \cdot Q, B_0 = W_0 \oplus h(H\text{Id}_0 || Y_0) \) and \( T_0 = h(B_0 || Y_0) || W_0 \). After completion of calculation, \( U_0 \) sends information \( \{B_0, X_0, T_0\} \) to server \( S \).

**B.** \( S \rightarrow U_0: \{E, \text{Auth}_1\} \):
Once getting message \( \{B_0, X_0, T_0\} \), \( S \) computes \( H\text{Id}_0' = N^* \oplus h(S_p), Y_0' = S_p \cdot X_0 \) and \( W_0' = B_0 \oplus h(H\text{Id}_0' || Y_0') \). Obviously, the verification equation \( T_0' = h(B_0 || Y_0' || W_0') \) is true. Next \( S \) generates random number \( r' \) and computes \( E = r' \cdot Q, sk_a = r' \cdot Y_0', \text{Auth}_a = h(sk_a || W_0' || Y_0') \).

After that, \( S \) transmits \( \{E, \text{Auth}_1\} \) to \( U_0 \).

**C.** \( U_0 \rightarrow S_0: \{\text{Auth}_a\} \):
When getting \( \{E, \text{Auth}_1\} \), \( U_0 \) computes \( sk_a = r_0 \cdot E, \text{Auth}_a = h(sk_a || W_0 || Y_0 || E) \) and sends \( \{\text{Auth}_a\} \) to \( S \).

**D.** After receiving the message \( \{\text{Auth}_a\} \), \( S \) computes \( \text{Auth}_a' = h(sk_a || W_0' || Y_0' || E) \) and gets a conclusion that \( \text{Auth}_a' \) is equal to \( \text{Auth}_a \). Finally, the user \( U_0 \) communicates with server \( S \) based on the common session key \( sk = sk_a = sk_i = r_0 \cdot r' \cdot Q \)
We draw a conclusion that A can masquerade as an arbitrary legitimate user for entering server S by insider attack.

3.3 Denial service attack

We assume that an adversary A is able to intercept message which is transmitted between U and S. At password update phase in Qiu et al.'s scheme, U sends the result values of V and M to S. The adversary A intercepts message \{V, M\} and forges M_0 by generating random number, then A transmits \{V, M_0\} to S. Apparently, A will pass verification of S by checking whether V is equal to V'. After that, S computes N' = h(S_p) \oplus h(Id \oplus k) \oplus M_0 and replaces N with N' in the database. Because of the falsity of N', the user U will fail to pass verification in next login and authentication phase.

3.4 Defects of practicality

In term of Qiu et al.'s scheme, we find that a user U needs to remember identity Id, password Pw and secret key Up. It is a burden for users to keep in mind with three different data. If a user U wants to change password, according to the password update phase of Qiu et al., he or she has to bear in mind with a new password Pw' and a new secret key Up'. For the most users, it is hard to remember so much information. Therefore, the scheme of Qiu et al. has poor practicability.

4 Our proposed scheme

In this section, in order to improve the security, we design an efficient provably secure mutual authentication scheme. Compared with the scheme of Qiu et al., our proposed scheme can resist various attacks and there is less pressure for users to remember. Our scheme consists of three phases: registration phase (see Fig 1), login and authentication phase (see Fig 2) and password update phase (see Fig 3). The proposed scheme is described as follows.

4.1 Registration phase

For a legal user U, if he or she wants to access the system, the necessary step is to register with the server S by submitting identity ID and password PW. User U and server S will perform the following steps.

1. U generates a long random number r and computes \(V_{PW} = h(PW \| ID \oplus r)\) and \(H_{Id} = h(Id \oplus r)\). Then U transmits \(\{V_{PW}, H_{Id}\}\) to S by a secure channel.

| User(U)                                                                 | Server(S)                                                                 |
|------------------------------------------------------------------------|---------------------------------------------------------------------------|
| Choose ID, PW, r                                                       | Compute \(V_{PW} = h(PW \| ID \| r)\)                                       |
| Compute \(H_{Id} = h(Id \oplus r)\)                                    | \(\rightarrow v_{PW}, H_{Id}\) Compute \(N = V_{PW} \oplus h(S_p \| H_{Id})\) |
| \(\text{Store } V_{PW} \text{ into } \text{smart card}\)              | \(R = h(S_p \oplus V_{PW})\) Store \(\{H_{Id}, N\}\) into database             |
| Smart card includes \(\{V_{R}, R\}\)                                  | Store \(R\) into \text{smart card}                                         |

Fig 1. Registration phase.

https://doi.org/10.1371/journal.pone.0213688.g001
2. After receiving the data, $S$ computes $N = VPw \oplus h(S_p || HId)$ and $R = h(S_p \oplus VPw)$. Then $S$ stores $\{HId, N\}$ into database and issues a smart card which contains $R$ to $U$ through a secure channel.

3. After receiving the smart card, $U$ computes $VR = h(PW \oplus ID) \oplus r$ and stores $VR$ into the smart card.

4.2 | Login and authentication phase

When a user $U$ wants to acquire the service from server $S$, he or she should insert his or her smart card into card reader and enter his or her identity $ID$ and password $PW$. $U$ and $S$ will perform the following steps.

1. The user $U$ generates a random number $r_a$ and computes $r' = VR \oplus h(PW \oplus ID)$, $HId' = h(ID \oplus r')$, $VPw' = h(PW||ID||r')$, $C = h(R \oplus VPw') \oplus r_a$ and $Auth_u = h(r_a \oplus HId' \oplus VPw')$. Then $U$ transmits $\{HId', C, Auth_u\}$ to the server $S$.

2. After receiving the message, $S$ extracts $N$ according to corresponding $HId'$ from the database and calculates $VPw'' = N \oplus h(S_p || HId')$, $r'_a = h(R \oplus VPw'') \oplus C$. Then checks whether the value of $Auth_u$ is equal to $h(r'_a \oplus HId' \oplus VPw'')$. If holds, $S$ picks a random number $r_s$, computes $sk_s = h(r_a \oplus HId' \oplus VPw')$. Then $S$ transmits $\{D, Auth_s\}$ to $U$. Otherwise, $S$ terminates the session.

3. When $U$ receives the message, $U$ computes $r'_a = h(R || VPw') \oplus D$, $sk_u = h(r_a \oplus r'_a || HId')$, checks whether $Auth_s$ is equal to $h(sk_u || D)$. If success, user $U$ and server $S$ share the same session key $sk_s = h(r_a \oplus r'_a || HId') = sk_u = h(r_a \oplus r'_a || HId')$. If not, the smart card aborts the session.

4.3 | Password update phase

If a user $U$ needs to change password $PW$ for a number of reasons, he or she only needs to input identity $ID$, password $PW$ and new password $PW_{new}$. User $U$ and server $S$ will perform the following steps.

---

**Input $ID, PW$**

- Compute $r' = VR \oplus h(PW \oplus ID)$
- $HId' = h(ID \oplus r')$
- $VPw' = h(PW || ID || r')$

**Generate $r_a$**

- Compute $C = h(R \oplus VPw') \oplus r_a$
- $Auth_u = h(r_a \oplus HId' \oplus VPw')$

**Extract $N$ in database**

- Compute $VPw'' = N \oplus h(S_p || HId')$
- $r'_a = h(R \oplus VPw'')$
- $Auth_s = h(r'_a \oplus HId' \oplus VPw'')$

**Generate $r_s$**

- Compute $sk_s = h(r_a \oplus HId' \oplus VPw')$
- $D = h(R || VPw'') \oplus r_s$
- $Auth_s = h(sk_s || D)$

---

Fig 2. Login and authentication phase.

https://doi.org/10.1371/journal.pone.0213688.g002
1. The user calculates 
\[ r_0 = VR \oplus h(\text{ID} \oplus h) \]
\[ HId = h(ID \oplus r) \]
\[ VPv' = h(PW \oplus ID \oplus r') \]
\[ VPw' = h(PW_{\text{new}} \oplus ID \oplus r') \]
\[ C1 = h(R \oplus VPv') \oplus VPw' \]
Generate \( r_0 \)
Compute 
\[ C2 = (VPv' \oplus VPw') \oplus r_0 \]
\[ Auth_u = h(r_0 \oplus HId \oplus VPw') \]
\[ C1, C2, HId, Auth_u \]
Compute 
\[ VPw' = N \oplus h(S_p \oplus HId') \]
\[ R' = h(S_p \oplus VPw') \]
\[ VPw_{\text{new}} = C1 \oplus h(R' \oplus VPw') \]
\[ r_a' = C2 \oplus h(VPw' \oplus VPw_{\text{new}}) \]
Check 
\[ Auth_s = h(r'_a \oplus HId' \oplus VPw_{\text{new}}) \]
Compute 
\[ N' = VPw'_{\text{new}} \oplus h(S_p \oplus HId') \]
Replace \( N \) with \( N' \)
Compute 
\[ R_{\text{new}} = h(S_p \oplus VPw_{\text{new}}) \]
\[ D = h(R' \oplus VPw'_{\text{new}} \oplus r_a') \oplus R_{\text{new}} \]
\[ Auth_s = h(VPw'_{\text{new}} \oplus R_{\text{new}}) \]

2. Upon receiving the message, \( S \) computes 
\[ VPw' = N \oplus h(S_p \oplus HId') \]
\[ R' = h(S_p \oplus VPw') \]
\[ VPw_{\text{new}} = C1 \oplus h(R' \oplus VPw') \]
\[ r_a' = C2 \oplus h(VPw' \oplus VPw_{\text{new}}) \]
Then \( S \) checks whether the value of \( Auth_u \) is equal to \( h(r'_a \oplus HId' \oplus VPw_{\text{new}}) \). If not, \( S \) terminates this session. Else continues to calculate 
\[ N' = VPw'_{\text{new}} \oplus h(S_p \oplus HId') \] to replace \( N \) with \( N' \) in the database. Finally, \( S \) computes 
\[ R_{\text{new}} = h(S_p \oplus VPw_{\text{new}}) \]
\[ D = h(R' \oplus VPw'_{\text{new}} \oplus r_a') \oplus R_{\text{new}} \]
and \( Auth_s = h(VPw'_{\text{new}} \oplus R_{\text{new}}) \). After that, message \( \{D, Auth_s\} \) will be transmitted to \( U \).

3. After receiving the message \( \{D, Auth_s\} \), \( U \) gets \( R'_{\text{new}} \) by computing 
\[ R'_{\text{new}} = D \oplus h(R' \oplus VPw'_{\text{new}} \oplus r_a) \]
and checks whether the value of \( Auth_s \) is equal to 
\[ h(VPw'_{\text{new}} \oplus R'_{\text{new}}) \]. If success, \( U \) computes 
\[ VR' = h(PW_{\text{new}} \oplus ID) \oplus r' \] replaces \( \{VR, R\} \) with \( \{VR', R'_{\text{new}}\} \), else terminates this session.

5 | Security analysis

In this part, we demonstrate our scheme is secure, practical and can provide kinds of security requirements. We assume that an adversary \( A \) might perform various attacks [18–21]. More detailed information is as follows.

5.1 | Insider attack

Assume that an adversary \( A \) can obtain \( N \) and \( HId \) of a legitimate user that stored in database. Because computational formulas are 
\[ HId = h(ID \oplus r) \]
\[ N = VPw \oplus h(S_p \parallel HId) \]
\[ VR = h(PW \oplus ID) \]
An improved and efficient mutual authentication scheme for session initiation protocol

5. 2| User anonymity

In the public channel, we do not send Id directly but transmit Hld which is computed by means of the formula Hld = h(ID ⊕ r). Even if A can access the value of Hld, A still cannot get the sensitive information Id. Because the formula contains a random number r which A does not know. Therefore, our proposed scheme provides user anonymity.

5. 3| Replay attack

In our proposed scheme, the random numbers r_s and r_a change in every login. For an adversary A, he can intercept information {Hld', C, Auth_a} and replay this message. Obviously, A can pass the verification of the server S and will receive the corresponding message {D, Auth_s} from server S. Because A does not have a knowledge of the correct values R, VPw and r_a, he cannot compute r_s' = h(R ⊕ VPw) ⊕ D and sk_a = h(r_a || r_s' || Hld'). Therefore, our scheme is security even under replay attack.

5. 4| off-line password guessing attack

In our scheme, if an adversary A accesses exchanged information {Hld', C, Auth_a, D, Auth_s} which is transmitted in a public channel [22,23]. Because of the formula Hld = h(ID ⊕ r'), A is unable to guess correct ID without r'. In addition, in the formulas VPw' = h(PW || ID || r'), C = h (R ⊕ VPw') ⊕ r_a, A cannot guess PW correctly without the value ID and r'. Therefore, our scheme has an advantage of resisting off-line password guessing attack.

5. 5| Smart card lost attack

If an adversary A steals the smart card, he is able to get the VR and R which are stored in the smart card. Because the formulas are VR = r' ⊕ h(PW ⊕ ID) and C = h(R ⊕ h(PW || ID || r')) ⊕ r_a, without the knowledge of r' and r_a, A cannot guess the correct identity ID and password PW. If A wants to communicate with S, he needs to structure legitimate {Hld', C, Auth_a}. Because of the formulas Hld' = h(ID ⊕ r'), C = h(R ⊕ h(PW || ID || r')) ⊕ r_a, Auth_a = h(r_a ⊕ Hld' ⊕ h (PW || ID || r')), without identity ID and password PW, A is unable to pass through the verification of server S. Therefore, our scheme can resist smart card lost attack.

5. 6| Impersonation attack

For an adversary A, in authentication phase, if he wants to masquerade as a legal user U and login in server S, he must forge a valid login message {Hld', C, Auth_a}. It is impossible for A to forge valid login message without legitimate identity ID, password PW, VR and R. The same is true, if A wants to masquerade as the server S, he has to counterfeit message {D, Auth_s}. Without valid information N, A is unable to obtain {D, Auth_s} which can pass through verification of the user U. Furthermore, in password update phase, A is unable to forge valid C1, C2 and Auth_s to pass the authentication of server S. In the same way, A is unable to forge valid D and Auth_s to pass the verification of user U. Therefore, our scheme can resist impersonation attack.

5. 7| Man-At-The-End attack

Man-At-The-End attack [24] contains widespread aspects and is difficult to model. The technical adversary is human that we call A here, he could authorize and limitless access to the
target. All security protections stand up to $A$ for a specific period of time. Because Man-At-The-End attack has concrete form in certain circumstance, one of the defense details is as follows:

$A$ could personate a legitimate user to register and access the sensitive value of $N^*$ from the database end. His own identity $ID^*$, password $PW^*$ are known. According to formulas $h(S_p||HId) = VPw \oplus N$, $VPw = h(PW||ID||r)$ and $HId = h(ID || r)$. Without having knowledge about the value of random number $r$, he will take unpractical time cost to compute $S_p$, by formulas $h(S_p||h(ID^*||r)) = h(PW^*||ID^*||r) \oplus N^*$. Our scheme offer a defense and $A$ is unable to obtain important information $S_p$.

Although $A$ could execute other forms and is hard to analyze, many protective devices have ability to against the attack, include software protection, hardware protection and digital asset protection, more details are reference no.24. Hence, our scheme can make a defense against Man-At-The-End attack.

5. 8| Perfect forward secrecy

When the system crashes in one session, an adversary can acquire information $ID$, $PW$, $S_p$ and intercept message {$HId'$, $C$, $Auth_u$} and {$D$, $Auth_s$}. For an adversary $A$, he wants to get $sk_s = sk_u = h(r_a || r_b || HId')$. But without the knowledge of the correct values $VR$ and $R$, he cannot compute $r_a' = h(R || VPw') \oplus C$, $r_b' = h(R || VPw') \oplus D$ correctly. Finally, $A$ still have no ability to get the session key to eavesdrop session. Hence, our scheme provides perfect forward secrecy.

6 | Security proof

Security model: Burrows–Abadi–Needham logic (also known as the BAN logic) is a set of rules for defining and analyzing information exchange protocols [25]. Specifically, BAN logic helps its users determine whether exchanged information is trustworthy, secured against eavesdropping or both.

In this section, we will demonstrate that our scheme is secure and practical by the Burrows-Abadi-Needham (BAN) logic. We list some essential BAN-Logic symbols and formulas as follows, Table 2 introduces the notations of symbol and Table 3 introduces notations of formula. Supposing that $P$ and $Q$ are the symbols of participants, $X$ and $Y$ are statements as symbols, and $K$ is the symbol for hash function key, next notations include more detailed explanation. We give reasoning process based on BAN-Logic in the following steps.

Step 1 Our goals

In order to make our scheme practicable, we list some goals which need to be achieved.

Goal 1. $U \equiv S \equiv (U \equiv S)$. 

| Symbol | Description |
|--------|-------------|
| $P \equiv X$ | $P$ believes $X$ |
| $P \Rightarrow X$ | $P$ has jurisdiction over $X$ |
| $\#(X)$ | $X$ is fresh |
| $P < X$ | $P$ sees $X$ |
| $P \neg X$ | $P$ once said $X$ |
| $(X, Y)$ | $X$ or $Y$ belongs to $(X, Y)$ |
| $(X)_K$ | $X$ is hash with the key $K$ |
| $P, S < Q$ | $P$ communicates with $Q$ using key $K$ |

https://doi.org/10.1371/journal.pone.0213688.t002
Table 3. Notations of formula.

| Formula | Description |
|---------|-------------|
| \( P_j \models (P \rightarrow \neg Q_j \land Q_j) \) | The message meaning rule |
| \( P \models \neg Q \land \neg \neg Q \) | The freshness rule |
| \( P \models (Q_j \equiv (X \land Y) \equiv Q_j) \) | The nonce-verification rule |
| \( P \models (Q \land \neg X \land \neg Y) \) | The jurisdiction rule |
| \( P \models (Q \land \neg X \land \neg Y) \) | The believe rule |

https://doi.org/10.1371/journal.pone.0213688.t003

Goal 2. \( U \models (U \leftrightarrow S) \).
Goal 3. \( S \models U \models (U \leftrightarrow S) \).
Goal 4. \( S \models (U \leftrightarrow S) \).

Step 2 Idealized form

We show the idealized message form between user \( U \) and server \( S \) as follows.

Msg 1. \( U \rightarrow S : (r_a, \ U \leftrightarrow S) \) VPw.
Msg 2. \( S \rightarrow U : (r_a, \ r_b, \ U \leftrightarrow S, \ U \leftrightarrow S, \ U \leftrightarrow S) \) VPw.

Step 3 Initial state

We transform some premises about authentication and login phase in our scheme.

Aspt 1. \( U \models \#(r_a) \).
Aspt 2. \( S \models \#(r_a) \).
Aspt 3. \( U \models S \models (U \leftrightarrow S) \).
Aspt 4. \( S \models U \models (U \leftrightarrow S) \).
Aspt 5. \( U \models (U \leftrightarrow S) \).
Aspt 6. \( S \models (U \leftrightarrow S) \).
Aspt 7. \( U \models (U \leftrightarrow S) \).
Aspt 8. \( U \models (U \leftrightarrow S) \).
Aspt 9. \( U \models (U \leftrightarrow S) \).
Aspt 10. \( S \models (U \leftrightarrow S) \).
Aspt 11. \( S \models U \models (U \leftrightarrow S) \).
Aspt 12. \( S \models U \models (U \leftrightarrow S) \).

Step 4 Derivation process

We prove that user \( U \) and server \( S \) can set up the session key by combining with the above information.

(1) The proof of goal 1 and goal 2:

S 1.

On the basis of Msg 2, we get that \( U \models (r_a, \ r_b, \ U \leftrightarrow S, \ U \leftrightarrow S, \ U \leftrightarrow S) \) VPw, in the light of Aspt 7, we know that \( U \models (U \leftrightarrow S) \). Because of the message meaning rule, we get the following results:

\[ U \models S \models (r_a, \ r_b, \ U \leftrightarrow S, \ U \leftrightarrow S, \ U \leftrightarrow S) \].

S 2.
According to S 1 and Aspt 1, we know that $U| \equiv S \sim (r_a, r_b, U \xleftarrow{r_a} S, U \leftarrow S, U \leftrightarrow S)$ and $U\equiv \#(r_a)$. Owing to the nonce verification and the freshness rule, which is listed as follows.

$$U| \equiv S| \equiv (r_a, r_b, U \xrightarrow{r_a} S, U \rightarrow S, U \leftrightarrow S).$$

S 3.
In line with S 2, we could achieve: $U| \equiv S| \equiv (U \xleftarrow{r_b} S)$ and

$$U| \equiv S| \equiv (U \leftrightarrow S).$$

Goal 1 is realized apparently.
S 4.
According to $U| \equiv S| \equiv (U \xleftarrow{r_b} S)$ and Aspt 3 $U| \equiv S \Rightarrow (U \xleftarrow{r_b} S)$, By the jurisdiction rule, we confirmedly know that $U| \equiv (U \xleftarrow{r_b} S)$. According to $U| \equiv (U \xleftarrow{r_b} S)$, Aspt 5 $U| \equiv (U \xleftarrow{r_a} S)$, Aspt 9 $U| \equiv (U \xleftarrow{HId} S)$ and $sk = h(r_a||r_b||HId)$, we make a representation is as follows:

$$U| \equiv (U \leftarrow S).$$

So, we have achieved Goal 2.
(2) The proof of goal 3 and goal 4:
S 5.
From Msg 1, we know that $S| \equiv (S_{\leftarrow r_a} U \xleftarrow{r_a} S)VPw$, because of Aspt 8 $S| \equiv (U \xrightarrow{VPw} S)$ and the message meaning rule, we can deduce that:

$$S| \equiv U| \sim (r_a, U \xrightarrow{r_a} S).$$

S 6.
On the basis of S 5 and Aspt 2, we know $S| \equiv U| \sim (r_a, U \xrightarrow{r_a} S)$ and $S|\equiv \#(r_a)$, owing to the nonce verification and the freshness rule, we obtain result what is as follows:

$$S| \equiv U| \equiv (r_a, U \xrightarrow{r_a} S).$$

S 7.
According to S 6 and Aspt 4 $S| \equiv U \Rightarrow (U \xrightarrow{r_a} S)$, because of the jurisdiction rule, we could obtain $S| \equiv (U \xrightarrow{r_a} S)$. Form Aspt 6 $S| \equiv (U \xrightarrow{r_a} S)$, Aspt 10 $S| \equiv (U \xrightarrow{HId} S)$ and the formula $sk = h(r_a||r_b||HId)$, we are able to get:

$$S| \equiv (U \xrightarrow{r_a} S).$$

Goal 4 is accomplished.
S 8.
According to S 6 $S| \equiv U| \equiv (U \xrightarrow{r_a} S)$ and Aspt 11 $S| \equiv U| \equiv (U \xrightarrow{HId} S)$, Aspt 12 $S| \equiv U| \equiv (U \xleftarrow{HId} S)$. Because of the formula $sk = h(r_a||r_b||HId)$, we have following.

$$S| \equiv U| \equiv (U \xrightarrow{r_a} S).$$

Distinctly, Goal 3 of our claim is achieved.
According to the proof above, our goals are achieved. We hold a sufficient reason to believe that both $U$ and $S$ believe that the session key $sk$ which is shared between $U$ and $S$. 


Performance comparison

The experience of users play an important role in protocol. During this part, we will show a performance comparison between our scheme and the other schemes (see Fig 4). Before making a comparative analysis, we assume that one elliptic curve point multiplication operation is $T_{pm}$, one hash function operation is $T_h$. Other operations like generating a random number and exclusive-OR operation spend less time, which have little effect on performance comparison. So, we neglect the lightweight operations at this time. Before performance simulation test, we analysis performance of other protocols and ours in theory. So, we list Table 4 to describt theoretical time spend comparison.
From Table 4, at registration phase, execution time of our scheme mainly lies on executing $5T_h$. In other three protocols, execution time mainly lies on executing $3T_h, 3T_h$ and $2T_h$. At authentication phase, execution time of our scheme mainly depends on executing $15T_h$. Meanwhile, execution time of other three protocols mainly depends on executing $12T_h + 6T_{pm}, 7T_h + 6T_{pm}, 10T_h + 6T_{pm}$. At password update phase, execution time of our scheme mainly lies on executing $19T_h$. And the other two protocols’ execution time mainly depends on executing $10T_h$ and $8T_h$. The protocol of Chaudhry et al. lacks the phase of password update. In those four protocols, total execution time are $39T_h, 25T_h + 6T_{pm}, 10T_h + 6T_{pm}, 20T_h + 6T_{pm}$ respectively.

In those four protocols, we find that our scheme performance mainly bases on hash function and the protocols of Qiu et al. [16], Chaudhry et al. [14] and Kumari et al. [15] are based on hash function and elliptic curve point multiplication. At registration phase and password update phase, we use a little more hash functions than those protocols. But the registration phase needs to be carried out only one time, so it has almost no effect on overall performance. And password update phase is not commonly used for a certain user, so it has little effect in practical applications. From authentication phase and total phases, though we use too many hash functions, other protocols all use six elliptic curve point multiplications. It is obvious that $T_{pm}$ is many times as much as $T_h$. Compared with other protocols, our scheme has a great advantage on computational costs in usual authentication phase and total phases.

We perform simulant performance comparison under the same computer simulation environment and write programs according to the schemes strictly. In the experiment, we run one hundred times to get the average data. According to the Fig 4, at registration phase, the time consumption in schemes of Qiu et al. [16], Chaudhry et al. [14] and we are all 0.00075s and Kumari et al.’s [15] scheme is 0.00088s. At password update phase, the time consumptions are 0.0018s, 0.001s, 0.0011s respectively. Although those scholars’ protocols have a little bit better performance at registration phase and update phase, in practical application, those scholars’ frequently-used authentication phases spend more time compared with our authentication phase. The costs of time are 0.0223s, 0.0201s, 0.0216s respectively, but ours is only 0.0023s. From comparative analysis, our total time is less than others. Obviously, our scheme is more efficient and more practice in application.

8 | Conclusion

In this paper, we review Qiu et al.’s protocol and find that it is vulnerable to some known attacks such as insider attack and denial service attack, then we review the scheme and carry on a strict security analysis about their scheme. Next, in order to solve these problems, we propose our more secure and more convenient scheme. Security analysis shows that our scheme can resist insider attack, off-line password guess attack and more, we give a sufficient reason. Then in security proof section, we adopt the BAN-logic to prove our scheme is secure and realizable. In the end, we make a performance comparison, the result shows that our scheme can be more suitable for users in SIP. Because under the same conditions, we can establish connections faster. In conclusion, compared to other protocols, our scheme is more security and practical.
Supporting information

S1 Fig. Transport details between user and server in registration phase.
(TIF)

S2 Fig. Transport details between user and server in login phase.
(TIF)

S3 Fig. Transport details between user and server in update phase.
(TIF)

S4 Fig. Performance Analysis Chart of the four schemes.
(TIF)

S1 Code. Zhang’s scheme. The VS code for analyzing the performance of the Zhang’s scheme.
(ZIP)

S2 Code. Qiu’s scheme. The VS code for analyzing the performance of the Qiu’s scheme.
(ZIP)

S3 Code. Kumari’s scheme. The VS code for analyzing the performance of the Kumari’s scheme.
(ZIP)

S4 Code. Chaudhry’s scheme. The VS code for analyzing the performance of the Chaudhry’s scheme.
(ZIP)

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61701173, No.61672010, No.61702168, No. 61502156), the Natural Science Foundation of Hubei Province of China (No. 2017CFB596), the fund of Hubei Key Laboratory of Transportation Internet of Things (WHUTIOT-2017B001), Guangxi Key Laboratory of Cryptography and Information Security (GCIS201717), the Ph.D. research startup foundation of Hubei University of Technology (BSQD2015028).

Author Contributions

Conceptualization: Yuanyuan Zhang.
Data curation: Yuanyuan Zhang.
Formal analysis: Yuanyuan Zhang.
Funding acquisition: Kunming Xie.
Methodology: Yuanyuan Zhang.
Resources: Yuanyuan Zhang, Ou Ruan.
Software: Kunming Xie.
Supervision: Ou Ruan.
Validation: Ou Ruan.
Writing – original draft: Kunming Xie.
Writing – review & editing: Kunming Xie.
References

1. Franks J, Hallam-Baker P, Hostetler J, Lawrence S, Leach P, Luotonen A, et al. HTTP authentication: basic and digest access authentication. IETF RFC. 1999; 2617: 1-34.
2. Yang C, Wang R, Liu W. Secure authentication scheme for Session Initiation Protocol. Computers & Security. 2005; 24(5):381-386.
3. Kumari S, Chaudhry SA, Wu F, Li X, Farash MS, Khan MK, et al. An improved smart card based authentication scheme for Session Initiation Protocol. Peer-to-Peer Networking and Applications. 2015. https://doi.org/10.1007/s12083-015-0409-0
4. Qiu SM, Xu GA, Ahmad H, Wang LC. A robust mutual authentication scheme based on elliptic curve cryptography for telecare medical information systems. IEEE Access. 2018; 6:7452-7463.
5. Tsai JL. Efficient Nonce-based Authentication Scheme for Session Initiation Protocol. International Journal of Network Security. 2009; 9(1):12–16.
6. Guo C, Chang CC, Chang SC. A Secure and Efficient Mutual Authentication and Key Agreement Protocol with Smart Cards for Wireless Communications. International Journal of Network Security. 2018; 20 (2): 323–331.
7. Azrour M, Farhaoui Y, Ouanan M. A new secure authentication and key exchange protocol for session initiation protocol using smart card. International Journal of Network Security. 2017; 19(6): 870–879.
8. Chaudhry SA, Naqvi H, Sher M. An improved and provably secure privacy preserving authentication protocol for SIP. Peer-to-Peer Networking and Applications. 2017; 10(1):1–15.
9. Xie Q. A new authenticated key agreement for session initiation protocol. International Journal of Communication Systems. 2012; 25(1):47–54.
10. Farash MS. Security analysis and enhancements of an improved authentication for session initiation protocol with provable security. Peer-to-Peer Networking and Applications. 2016; 9(1): 82–91.
11. Wu L, Fan J, Xie Y. An Improved Authentication and Key Agreement scheme for Session Initial Protocol. KSII Transactions on Internet and Information Systems (TIIIS). 2017; 11(8): 4025–4042.
12. Zhang Z, Qi Q, Kumar N. A secure authentication scheme with anonymity for session initiation protocol using elliptic curve cryptography. Multimedia Tools and Applications. 2015; 74(10): 3477–3488.
13. Lu Y, Li L, Peng H, Yang Y. A secure and efficient mutual authentication scheme for Session Initiation Protocol. Peer-to-Peer Netw Appl. 2016; 9(2):449-459.
14. Chaudhry SA, Khan I, Irshad A, Ashraf MU, Khan MK, Ahmad HF, et al. A provably secure anonymous authentication scheme for Session Initiation Protocol. Security and Communication Networks. 2016; 9 (18):5016-5027.
15. Kumari S, Karuppiah M, Das AK, Li X, Wu F, Gupta V, et al. Design of a secure anonymity-preserving authentication scheme for Session Initiation Protocol using elliptic curve cryptography. Journal of Ambient Intelligence and Humanized Computing. 2018; 9(3): 643–653.
16. Qiu S, Xu G, Ahmad H. An enhanced password authentication scheme for session initiation protocol with perfect forward secrecy. PLOS ONE. 2018; 13(3): e0194072. https://doi.org/10.1371/journal.pone.0194072 PMID: 29547619
17. Kumari S, Li X, Wu F. Design of a provably secure biometrics-based multi-cloud-server authentication scheme. Future Generation Computer Systems. 2017; 68: 320–330.
18. Kumari S, Khan MK, Li X. An improved remote user authentication scheme with key agreement. Computer & Electrical Engineering. 2014; 40(6): 1997–2012.
19. Zhang Mingwu, Zhang Yudi, Jiang Yan and Shen Jian. Obfuscating EVES algorithm and its application in fair electronic transactions in public cloud systems. IEEE System Journal. 2019. https://doi.org/10.1109/JSYST.2019.2900723
20. Khan MK, Alghathbar K. Cryptanalysis and security improvements of ‘two-factor user authentication in wireless sensor networks’. Sensors. 2010; 10(3): 2450–2459. https://doi.org/10.3390/s100302450 PMID: 22294935
21. Horng SJ, Tseng SF, Huang PH. An efficient certificateless aggregate signature with conditional privacy-preserving for vehicular sensor networks. Information Sciences. 2015; 317: 48–66.
22. Lu R, Cao Z. Off-line Password Guessing Attack on an Efficient Key Agreement Protocol for Secure Authentication. JI Network Security. 2006; 3(1): 35–38.
23. Kumari S, Li X, Wu F. A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps. Future Generation Computer Systems. 2016; 63: 56–75.
24. Akhunzada A, Sookhak M, Anuar N B. Man-At-The-End attacks: Analysis, taxonomy, human aspects, motivation and future directions. Journal of Network and Computer Applications. 2015; 48: 44–57.
25. Burrows M, Abadi M, Needham RM. A logic of authentication. Proc. R. Soc. Lond. A. 1989; 426(1871): 233–271.