An Enhanced Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services

Ling Xiong[1], Daiyuan Peng[1], Tu Peng[2] and Hongbin Liang[3]
[1] School of Information Science and Technology, Southwest Jiaotong University,
Chengdu 611756, Sichuan- P. R. China
[e-mail: lingdoghua99@163.com, dypeng@swjtu.edu.cn]
[2] School of Software, Beijing Institute of Technology
Beijing 100081 - P. R. China
[e-mail: pengtu@bit.edu.cn]
[3]School of Transportation and Logistics, Southwest Jiaotong University
Chengdu 611756, Sichuan- P. R. China
[e-mail:hbliang@swjtu.edu.cn]
*Corresponding author: Hongbin Liang

Received March 26, 2017; revised August 7, 2017; raccepted August 23, 2017;
published December 31, 2017

Abstract

With the fast growth of mobile services, Mobile Cloud Computing(MCC) has gained a great deal of attention from researchers in the academic and industrial field. User authentication and privacy are significant issues in MCC environment. Recently, Tsai and Lo proposed a privacy-aware authentication scheme for distributed MCC services, which claimed to support mutual authentication and user anonymity. However, Irshad et.al. pointed out this scheme cannot achieve desired security goals and improved it. Unfortunately, this paper shall show that security features of Irshad et.al.'s scheme are achieved at the price of multiple time-consuming operations, such as three bilinear pairing operations, one map-to-point hash function operation, etc. Besides, it still suffers from two minor design flaws, including incapability of achieving three-factor security and no user revocation and re-registration. To address these issues, an enhanced and provably secure authentication scheme for distributed MCC services will be designed in this work. The proposed scheme can meet all desirable security requirements and is able to resist against various kinds of attacks. Moreover, compared with previously proposed schemes, the proposed scheme provides more security features while achieving lower computation and communication costs.

Keywords: Mobile cloud computing, single sign-on, trust third party, authentication, user anonymity

This research was supported by National Science Foundation of China under Grant number (Grant No. 61502034 and 61571375).

https://doi.org/10.3837/tiis.2017.12.026 **ISSN : 1976-7277**

1. Introduction

With the development of wireless communication technologies, mobile devices such as smart phones, notebook PC, and PDA have become an essential part of our daily life. These technological revolutions bring a lot of conveniences to mobile users. They can access various kinds of mobile services(e.g. Mobile payment services[1], mobile social[2,3], mobile healthcare[4]) from anywhere at anytime. However, limited resources (e.g. Computational ability, memory, battery capacity) and communications (e.g.Low bandwidth and security) have impeded the qualities of mobile services[1,5]. Cloud computing as a network-based infrastructure provides computing resources such as operating systems, storage, networks, hardware, databases, and even entire software applications to users as on-demand fashion[6-8]. Therefore, Mobile Cloud Computing(MCC), which combines mobile environment and cloud computing, has been a new computing paradigm[3], provides us a lot of services and brings conveniences to our life and work[9-12].

In the MCC environment, mobile users can access multiple MCC service providers through wireless local area network or 3G/4G/5G telecommunication network[10]. Due to the openness of wireless networks, a malicious adversary could control the communication channel easily, i.e., he/she is able to eavesdrop, insert, block, and alter the transmitted messages. Thus, MCC is subject to various types of attacks. It is indispensable to achieve mutual authentication between mobile users and MCC service providers[12]. Additionally, the leakage of users' identities may reveal their locations, movements, and purchasing preferences, so it is of great concern to protect users' identities[13].

To access different MCC services, the mobile user needs to log in each MCC service provider with different identities and passwords according to the traditional single-server authentication schemes. Thus, the user needs to manage many identities and passwords. To reduce password fatigue from different identities and passwords, Single Sign-on(SSO) has been introduced into MCC services environment. Based on SSO, the mobile user needs to provide only one identity and password to access multiple MCC service providers[14]. Although SSO authentication schemes such as Passport[15] and OpenID[16] have brought a lot of conveniences to mobile users, these schemes require a fully trusted third party to participate in each user authentication phase, which may make the trusted third party being a bottleneck of security[13,14]. Therefore, it is necessary to design a privacy-aware authentication scheme for MCC services without the help of online trusted third party participation.

Multi-server authentication technique, which only requires the mobile user to register once at the third party, is a special way of SSO. Fig.1 illustrates the multi-server architecture. At first, each mobile user and server need to register with Registration Center(RC) through a secure channel. Then, the mobile user can access multiple servers without the help of RC during the user authentication phase. RC is responsible for issuing secret information for servers and mobile users, as well as generating public parameters. **Fig. 1** shows that Multi-server authentication technique is suitable for MCC services environment. Tsai and Lo's work[14] and Irshad et.al.'s work[17] are typical application examples.

Traditional single-factor password authentication schemes suffer from a series of security issues. Multiple factors have been introduced into authentication protocols to address these problems. There are three factors generally admitted by authentication, namely[18], something you know; something you have; something you are. This paper will develop on the basis of two factors: something you know and something you have. However, the major drawback of two-factor authentication schemes must be the limitation between two-factor security and wrong password login/update attack. Similarly, most of the multi-factor authentication schemes fail to satisfy both attacks at the same time. The 'fuzzy verifier' method proposed by Wang et.al.[19,20] will be used in this work to solve this problem.

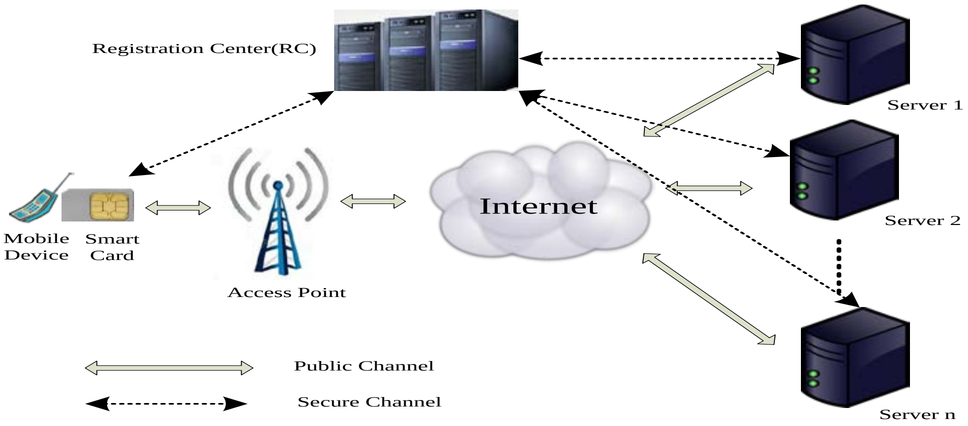


Fig. 1. Multi-server architecture

Based on the above analysis, the main security requirements of MCC services will be summarized as follows.

✧ **Mutual authentication:** It ensures that the MCC service providers and mobile users can successfully authenticate each other.

✧ **User anonymity:** It ensures that the adversary cannot obtain mobile users' identities through the transmitted messages in the public channel.

✧ **Un-traceability:** It ensures that the adversary cannot trace mobile users' behaviors from the transmitted messages in the public channel.

✧ **Truly two-factor security:** It ensures that the scheme for MCC services should satisfy the following two conditions: (1) the adversary has got the mobile device and obtains the secret information in the mobile device, but he/she should not be able to perform the offline password dictionary attack; and (2) the adversary who knows the password should not be able to perform impersonate attack without secret information in the mobile device.

✧ **Resistance to wrong password login/update attack:** It ensures that a wrong password in the login phase or password update phase does not cause the waste of computation and communication resources in MCC service provider side.

✧ **Secure session key agreement:** It ensures that two participants should be able to agree with a secure session key, which will protect transmitted messages in future communications.

✧ **Perfect forward secrecy:** It ensures that the adversary is unable to obtain the session key generated in previous sessions even if the long-term private keys of the two participants are leaked.

✧ **No online trust third party:** It ensures that the two participants achieve mutual authentication without the help of online trust third party.

- ✧ **Resistance to various attacks:** It ensures that various attacks should be prevented in MCC services, such as replay attack, man-in-the-middle attack, privileged insider attack and stole-verifier attack.

In this paper, following the viewpoint of Irshad et.al.'s work[17], an enhanced privacy-aware authentication scheme for distributed MCC services without the help of online RC will be designed. The proposed scheme can achieve mutual authentication, user anonymity and truly two-factor security, etc. Besides, it is able to resist various kinds of attacks such as wrong password attack and service provider compromising attack. Moreover, the performance analysis shows the proposed scheme has much better computation and communication efficiencies than Irshad et.al.'s work[17].

The rest of this paper is organized as follows. Section II gives some related works and our research contributions, followed by the detailed procedure of the proposed scheme in Section III. Section IV gives security analysis of our scheme. Finally, Section V concludes the paper.

2. Related Work and Our Research Contributions

2.1 Relate Work

User authentication is critical to distributed MCC services, which prevents illegal users from accessing the service providers. In 1991, Chang and Wu[21] proposed the first authentication scheme using both the password and the smart card. A smart card was issued to the remote user in this scheme when the user registered for the system. To achieve better performance, several two-factor authentication schemes for cloud computing were proposed[22-24]. Compared with Chang and Wu's scheme, these schemes have many advantages. However, they are still prone to some security flaws, such as masquerading attack, OOB attack and offline dictionary attack[25]. Moreover, these single-server authentication schemes[22-24] cannot be directly applied to MCC services environment. The reason is that they require mobile users to log in each service provider with different identities and passwords. Thus, the mobile user needs to manage various identities and passwords.

SSO is an excellent technique to resolve this problem. At present, OpenID[16], SAML[26] and OAuth[27] are mainly emerging SSO protocols. OpenID is an open protocol, which depends on session cookies as verification mechanism[16]. SAML is one of the most popular SSO protocol, which is mainly used for enterprises and universities[26]. OAuth is designed to provide a secure authentication mechanism for websites, which has two versions, namely the OAuth 1.0 and the OAuth 2.0[27]. These SSO authentication schemes bring huge conveniences to users. However, these schemes require a fully trusted third party to participate in each user authentication phase, which may make the trusted third party being a bottleneck of security[13,14]. Besides, most of these SSO authentication schemes establish communication connections through SSL or TLS[28]. SSL and TLS implementations rely on public-key infrastructure(PKI), which needs heavy computation cost and an extra key management system for certificate management[29]. As a result, these SSO authentication schemes may be unsuitable for MCC services environment.

Recently, several multi-server authentication schemes without online RC participation have been proposed[30~33], which only need mobile users to register once at RC. 2015, Tsai and Lo first proposed a privacy-aware MCC authentication scheme using multi-server authentication[14], which claimed to support mutual authentication and privacy protection. But later, Jiang et.al.[13] pointed out that their scheme fails to provide mutual authentication and suffers from some design flaws, such as wrong password login attack and no consideration user revocation. However, Jiang et.al. do not give a solution to achieve mutual authentication and leave it for their future work.

In the same year, a number of improved schemes[17,34-36] have been put forward, which have more security advantages than Tsai and Lo's scheme. However, all of these schemes still suffer from minor design flaws such as the problem of wrong password login and no user revocation. Irshad et.al.[17] proposed an improved multi-server authentication scheme for distributed MCC services, which addresses the security problems in Tsai and Lo's scheme. Unfortunately, we find that Irshad et.al.'s scheme is the limitation of low computing efficiency for using three bilinear pairing operations, one map-to-point hash function operation, etc. Besides, it is still vulnerable to truly three-factor security and no user revocation and re-registration. Because the verification data $D_i = h(ID_i \,||\, PW_i \,||\, H_b(f_i))$ is stored in the smart card, where PW_i is the password, and f_i is the fingerprint of the user. The three factors are the smart card, PW_i and f_i, respectively. Obviously, when the secret information in smart card and f_i are leaked, it is easy to guess the password. Additionally, in Irshad et.al.'s scheme, RC does not maintain identity information, therefore it cannot consider user revocation and re-registration[13,37]. Odelu et.al.[34] presented a provably secure authenticated key agreement scheme for MCC services using the signcryption, which can support mutual authentication and user anonymity. However, as in the reference[17], it cannot provide truly three-factor security. He et.al.[35] proposed an efficient privacy-aware authentication protocol for MCC services. This scheme still suffers from some minor design flaws, including wrong password login attack, no user password update phase and no revocation and re-registration. Two reasons contribute to these defects. One is that no password verification data exists in the mobile device to reject the wrong password. The other is that the trust third party does not maintain an identity information table.

Amin et.al.[36] presented an authentication scheme for MCC services, which requires all registered mobile users in the system to be honest. However, in practice, there may exist many malicious users in MCC services environment, who can launch impersonation attack at any time. In Amin et.al.'s scheme, the MCC service provider ID_{Sj} achieve mutual authentication with all registered mobile users through the sharing secret value PK_j, while all MCC service providers' secret values $<(ID_{S1},PK_1), (ID_{S2}, PK_2),..., (ID_{Sj}, PK_j),....>$ are stored in the mobile device. Therefore, it is easy for a malicious user to impersonate a legitimate user ID_i or a MCC service provider ID_{Sj} because he/she can decrypt secret values in his/her mobile device to obtain PK_j. To address all of these issues, this paper proposes an enhanced authentication scheme for distributed MCC services based on Irshad et.al.'s scheme.

2.2 Research Contributions

In this paper, an enhanced privacy-aware SSO authentication scheme is designed for distributed MCC services. The major contributions are summed up as follows:

- First, the proposed scheme can achieve mutual authentication without the help of online RC participation.
- Second, the proposed scheme is resilient to various kinds of known attacks, such as wrong password login/update attack, impersonation attack. Besides, security analysis shows that it can achieve mutual authentication, user anonymity, perfect forward secrecy, etc.
- Finally, by comparing with previously proposed scheme, the proposed scheme provides more security features while keeping lower computation and communication costs.

3. The Proposed Scheme

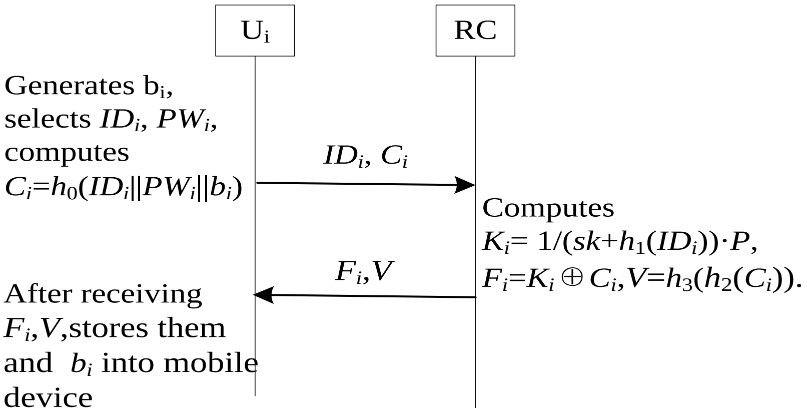
The proposed scheme consists of five phases: initialization phase, user registration phase, services registration phase, authentication phase and password update phase.

3.1 Initialization Phase

In the initialization phase, RC chooses an additive group of point G_1 with order q, and a cyclic multiplicative group G_2 with order q, where P is a generator of G_1, g is a generator of G_2. RC generates the system private key sk and calculates $PK = sk \cdot P$. Then RC chooses a bilinear pairing $e: G_1 \times G_1 \to G_2$, $g = e(P,P)$ and five secure hash functions $h_0, h_1, h_3, h_4 : \{0,1\}^* \to Z_q^*$, $h_2 : \{0,1\}^* \to \{0,1,2,...,255\}$. RC stores the system private key sk in its secure memory and publishes the system parameters $\{G_1,G_2,e,q,g,P,PK,h_0,h_1,h_2,h_3, h_4\}$.

3.2 User Registration Phase

When a user U_i wants to access a service provider, he/she needs to register in RC first. As shown in **Fig. 2**, the process of registration is shown as follows.



U_i
RC
Generates b_i, selects ID_i, PW_i, computes $C_i=h_0(ID_i||PW_i||b_i)$
ID_i, C_i
Computes $K_i= 1/(sk+h_1(ID_i))\cdot P$, $F_i=K_i \oplus C_i, V=h_3(h_2(C_i))$.
F_i,V
After receiving F_i,V,stores them and b_i into mobile device

Fig. 2. The user registration phase

(1) U_i selects identity ID_i and password PW_i, generates a random number b_i. Then U_i computes $C_i = h_0(ID_i \| PW_i \| b_i)$, U_i transmits $\{ID_i,C_i\}$ to RC through a secure channel.

(2) RC checks whether ID_i exists in the user information table, if it exists, RC rejects this request. Otherwise, RC computes $K_i = \frac{1}{sk + h_1(ID_i)} \cdot P$, $F_i = K_i \oplus C_i, V = h_3(h_2(C_i))$. After that, RC updates the user identity information table with new entry $\{ID_i\}$, and sends $\{F_i,V\}$ to U_i via a secure channel.

(3) After receiving $\{F_i,V\}$ from RC, U_i stores $\{F_i,V\}$ and b_i into the mobile device.

3.3 Service Provider Registration Phase

Similar to the user registration phase, the process of service provider registration is described as follows.

(1) The service provider S_j selects identity ID_j and transmits $\{ID_j\}$ to RC through a secure channel.

(2) RC checks whether ID_j exists in the service provider information table, if it exists, RC rejects this request. Otherwise, RC computes $K_j = \frac{1}{sk + h_1(ID_j)} \cdot P$. After that, RC updates the service provider identity information table with new entry $\{ID_j\}$, and sends K_j to S_j via a secure channel.

(3) After receiving K_j from RC, S_j stores it the secure memory.

3.4 Authentication Phase

When a user U_i wants to log in a service provider S_j, U_i needs to access to S_j. As shown in **Fig. 3**, the process of authentication is shown as follows.

(1) U_i inputs ID_i and PW_i into the mobile device MD_i. MD_i computes $C_i = h_0(ID_i \| PW_i \| b_i)$, $V' = h_3(h_2(C_i))$, and checks whether V' and V are equal. If not, MD_i terminates the session. Otherwise, MD_i generates a random number $x \in Z_q^*$, and computes $X = g^x$, $R_1 = x \cdot PK + xh_1(ID_j) \cdot P$, $CT = ID_i \oplus h_0(X)$. Then U_i sends $\{R_1, CT\}$ to S_j through the public channel.

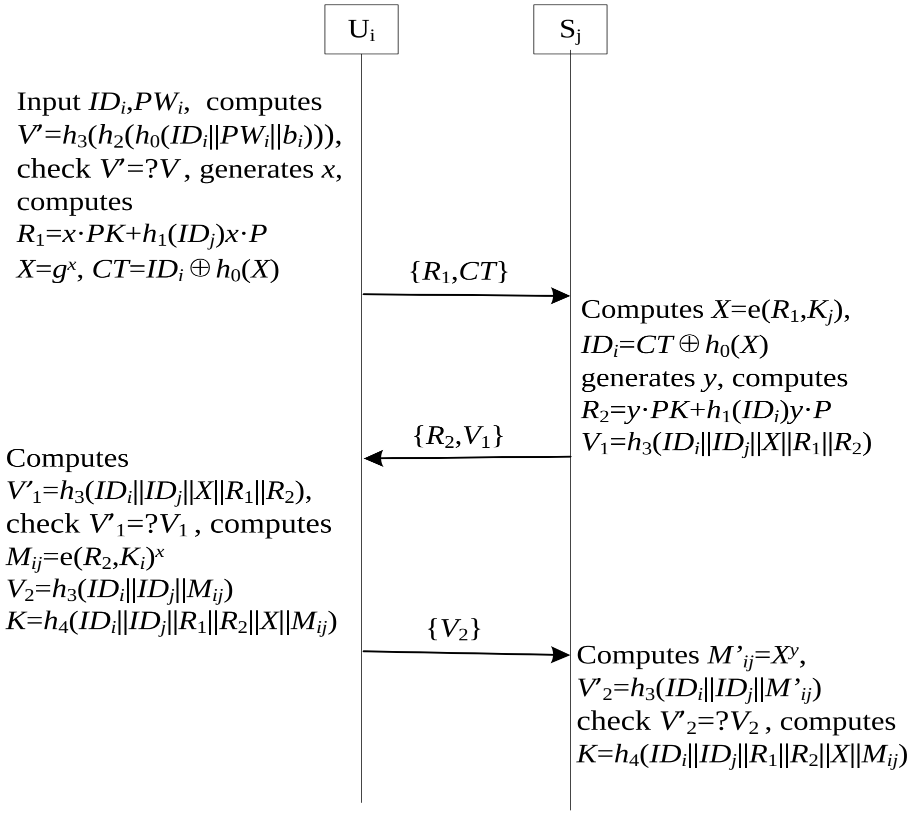

U_i
S_j
Input ID_i,PW_i, computes
V'=$h_3(h_2(h_0(ID_i\|PW_i\|b_i)))$,
check V'=?V, generates x,
computes
R_1=$x \cdot PK$+$h_1(ID_j)x \cdot P$
X=g^x, CT=$ID_i \oplus h_0(X)$
$\{R_1,CT\}$
Computes X=e(R_1,K_j),
ID_i=$CT \oplus h_0(X)$
generates y, computes
R_2=$y \cdot PK$+$h_1(ID_i)y \cdot P$
V_1=$h_3(ID_i\|ID_j\|X\|R_1\|R_2)$
$\{R_2,V_1\}$
Computes
V'_1=$h_3(ID_i\|ID_j\|X\|R_1\|R_2)$,
check V'_1=?V_1, computes
M_{ij}=e(R_2,K_i)^x
V_2=$h_3(ID_i\|ID_j\|M_{ij})$
K=$h_4(ID_i\|ID_j\|R_1\|R_2\|X\|M_{ij})$
$\{V_2\}$
Computes M'_{ij}=X^y,
V'_2=$h_3(ID_i\|ID_j\|M'_{ij})$
check V'_2=?V_2, computes
K=$h_4(ID_i\|ID_j\|R_1\|R_2\|X\|M_{ij})$

Fig. 3. The user authentication phase

(2) S_j computes $X = e(R_1, K_j)$, $ID_i = CT \oplus h_0(X)$. Then, S_j generates a random number $y \in Z_q^*$, and computes $R_2 = y \cdot PK + yh_1(ID_i) \cdot P$, $V_1 = h_3(ID_i \| ID_j \| X \| R_1 \| R_2)$. S_j sends $\{R_2, V_1\}$ to U_i.

(3) After receiving the message, U_i computes $V_1' = h_3(ID_i \| ID_j \| X \| R_1 \| R_2)$, and checks whether V_1' and V_1 are equal to each other. If not, U_i terminates the session. Otherwise, U_i computes $M_{ij} = e(R_2, K_i)^x$, $V_2 = h_3(ID_i \| ID_j \| M_{ij})$, $K = h_4(ID_i \| ID_j \| R_1 \| R_2 \| X \| M_{ij})$. Then U_i transmits V_2 to S_j.

(4) S_j computes $M_{ij}' = X^y$, $V_2' = h_3(ID_i \| ID_j \| M_{ij}')$ and checks whether V_2' and V_2 are equal. If not, S_j fails to authenticate the user U_i, and the session is terminated. Otherwise, S_j verifies U_i successful. Then S_j computes session key $K = h_4(ID_i \| ID_j \| R_1 \| R_2 \| X \| M_{ij})$.

Finally, U_i and S_j have the same session key K.

3.4 Password Update Phase

When a user U_i wants to update the password, he/she should run as follows:

(1) U_i inputs ID_i, password PW_i into the mobile device MD_i. MD_i computes $C_i = h_0(ID_i \| PW_i \| b_i)$, $V' = h_3(h_2(C_i))$, and checks whether V' and V are equal. If not, the mobile device fails to authenticate U_i, and rejects the request of password update. Otherwise U_i inputs a new password PW^*_i.

(2) MD_i computes $C_i^* = h_0(ID_i \| PW_\mathrm{i}^* \| b_i)$, $F_i^* = F_i \oplus C_i \oplus C_i^*, V^* = h_3(h_2(C_i^*))$.

(3) Finally, F^*_i and V^* are stored in MD_i to replace F_i and V respectively.

4. Security Analysis and Performance Comparisons

4.1 Security Model

Protocol participant. The proposed scheme involves four participants, the registration center RC, the service provider S_j, the mobile device MD_i and the mobile user U_i. RC is a trusted third party and it generates secure parameters. S_j is the MCC service provider who is assessed by mobile users identified.

Protocol execution. The proposed scheme has five phases:the initialization phase, the user registration phase, the service provider registration phase, the authentication phase and the password update phase. The initialization phase, the user registration phase and the service provider registration phase are assumed to be executed securely.

Adversary model. The adversary A has two goals. One is that A can successfully impersonate U_i authenticating to S_j, and the other is A can successfully impersonate S_j authenticating to U_i. Assume that A is a probabilistic polynomial time attacker, and the feasible attacks are summarized as follows:

(1) A can control the channel between the user and service provider. It means that A can eavesdrop, insert, block, and alter the transmitted messages through the communication channel.

(2) A can obtain one of the two factors: the mobile device or the password. If A has obtained the mobile device, he/she can extract the secret information in the mobile device. Then he/she has the capability of enumerating the password space $|D_{PW}|$.

(3) A may be another legitimate but malicious user in the system.

(4) A may be a legitimate but malicious service provider.

Security Model. In order to prove the security of the proposed scheme, we follow the security model presented in references[38,39]. According to this model, the security of the proposed scheme is defined by a game played by the adversary A and a challenger ζ. Further detail of the model can be found in references[38,39]. Let instance $\prod_U^s$ be the user oracle in session *s*, $\prod_S^s$ be the service provider oracle in session *s*. *A* can make following oracle queries.

- ✧ ***h$_i$*(*m$_i$*):** This query simulates hash function. When *A* asks the query m_i, ζ generates a random $h_i \in Z_q^*$ and returns h_i to *A*.
- ✧ **Register(*ID$_i$*):** This query simulates *A* registration as a legitimate user or service provider. *A* issues identity and receives secret information of the mobile device.
- ✧ **Send(*P,s,P',M*):** This query simulates *P'* sends message *M* to $\prod_P^s$. Then the oracle takes the actions specified by the protocol and outputs a response to *A*. If *P'* and *M* are null and *P* is user oracle, it means creating a new instance.
- ✧ **Reveal(*ID$_i$*):** This query simulates the leakage of session key attack, and will output the session key *K*.
- ✧ There are three corruption queries:
  (1) **Corrupt(*ID$_i$,PW$_i$*):**This query simulates the password leakage attack, and will output the user password PW_i.
  (2) **Corrupt(*ID$_i$, MD$_i$*):**This query simulates the mobile device stolen attack, and will output the secret information stored in the mobile device.
  (3) **Corrupt(*S$_j$*):**This query simulates the service provider compromise attack.
- ✧ **Test(*P,s*):** This query simulates the semantic security of the session key. ζ chooses a random bit $b \in \{0,1\}$. If *b*=1, ζ returns the session key *K* to *A*. Otherwise, ζ returns a random number to *A*.

Definition 1: Matching sessions: a session in the instance $\prod_U^s$ and a session in the instance $\prod_S^{s'}$ are said to be matching if *s* = *s'*, $pid_U = S$, $pid_S = U$ and both instances have accepted, where pid_U and pid_S denote as a peer identity.

Definition 2: Secure protocol: the proposed scheme is secure if the following properties hold:

(1) $\prod_U^s$ and $\prod_S^s$ are matching session and they accept each other.
(2) The probability of $\prod_S^s$ accepted *A* as $\prod_U^s$ is negligible.
(3) The probability of $\prod_U^s$ accepted *A* as $\prod_S^s$ is negligible.
(4) The session key is indistinguishable with a random number.
(5) When *A* has obtained the secret key in the mobile device, the probability of *A* knew the password is negligible.

4.2 Provable Security

To prove the security of the proposed scheme, assume that the scheme is defined by a game played between an adversary *A* and a challenger ζ. Firstly, three mathematical problems used for the security analysis will be defined as follows[14,35].

Definition 3 Discrete Logarithm (DL) Problem: Given $X = x \cdot P$, where $x \in Z_q^*$, $X \in G_1$, it is infeasible to compute *x*.

Definition 4 Collusion Attack Algorithm with *k*-traitors(*k*-CAA problem): Given *k* elements $e_1, e_2, ..., e_k \in Z_q^*$ and *k*+2 elements *P*, $sk \cdot P$, $\frac{1}{sk+e_1} \cdot P$, $\frac{1}{sk+e_2} \cdot P$,..., $\frac{1}{sk+e_k} \cdot P \in G_1$, it is infeasible to compute $\frac{1}{sk+e_0} \cdot P$, where $e_0 \notin \{e_1, e_2, ..., e_k\}$ and *sk* is a unknown element in

Z_q^*.

Definition 5 Modified Bilinear Inverse Diffie-Hellman with k Value Problem(k-mBIDH): Given k elements $e_1,e_2,...,e_k \in Z_q^*$ and k+3 elements P, $sk \cdot P$, $y \cdot P$ $\frac{1}{sk+e_1} \cdot P$, $\frac{1}{sk+e_2} \cdot P$,..., $\frac{1}{sk+e_k} \cdot P \in G_1$, it is infeasible to compute $e(P,P)^{\frac{y}{sk+e_0}}$, where $e_0 \notin \{e_1,e_2,...,e_k\}$, y and sk is a unknown element in Z_q^*.

Lemma 1. (Secure user authentication): In the proposed scheme, if hash functions h_0,h_1,h_3,h_4 are ideal random functions and $\prod_S^s$ has accepted, then no polynomial adversary against the proposed scheme can forge a legal user authentication message with a non-negligible probability.

Proof. Assume that the adversary A can forge a legal user authentication message with a non-negligible probability ϵ. Then there is a challenger ζ can solve the k-mBIDH problem with a non-negligible probability.

Given an instance {($e_1,e_2,...,e_k \in Z_q^*$),P, $sk \cdot P$, $y \cdot P$ $\frac{1}{sk+e_1} \cdot P$, $\frac{1}{sk+e_2} \cdot P$,..., $\frac{1}{sk+e_k} \cdot P \in G_1$ } of k-mBIDH problem, the task of ζ is to compute $e(P,P)^{\frac{y}{sk+e_0}}$. ζ sends the system parameters $\{G_1,G_2,e,q,g,P,PK,h_0,h_1,h_2,h_3,h_4\}$ to A. Assume that ID_0 is the identity of challenge. Then ζ interacts with A queries as follows:

✧ ***h*₁(*ID*ᵢ)** : The hash query maintains a list L_{h1} initialized empty. ζ checks whether the message ID_i exists in L_{h1}. If it exists, ζ returns its value h_i to A. Otherwise, If $ID_i = ID_0$, ζ sets $h_1(ID_0) \leftarrow e_0$. Otherwise, ζ sets $h_1(ID_i) \leftarrow e_i$. Then, ζ stores the tuple (ID_i,h_i) into L_{h1} and returns h_i to A.

✧ ***h*ᵢ(*m*ᵢ)** : The hash query $h_i(m_i)$, i = 0,3,4 maintains a list L_{hi} initialized empty. ζ checks whether the message m_i exists in L_{hi}. If it exists, ζ returns its value h_i to A. Otherwise, ζ generates a random number h_i, stores the tuple (m_i,h_i) into L_{hi} and returns h_i to A.

✧ **Register(*ID*ᵢ)**: In this query ζ maintains a list L_R with initialized empty. When A asks this query with identity ID_i, ζ checks whether the tuple of ID_i exists in L_R. If it exists, ζ returns ID_i to A. Otherwise, ζ operates as follows:

◆ If $ID_i = ID_0$, ζ sets $h_1(ID_i) = e_0$, $K_i = \perp$, and stores (ID_i,e_0) and (ID_i,K_i) into L_{h1} and L_R respectively. ζ returns ID_i to A.

◆ If $ID_i \neq ID_0$, ζ sets $h_1(ID_i) = e_i$, $K_i = \frac{1}{sk+e_i} \cdot P$ and stores (ID_i,e_i) and (ID_i,K_i) into L_{h1} and L_R respectively. ζ returns ID_i to A.

✧ **Send(*U*ᵢ,*s*,*S*ⱼ,*M*)**: ζ checks whether S_j and M are empty. If they are empty, ζ operates according to the specification of the proposed scheme and returns $\{R_1,CT\}$ to A. Otherwise, ζ checks whether $U_i = ID_0$. If they are not equal, ζ operates according to the specification of the proposed scheme and returns V_2 to A. Otherwise, ζ aborts the game.

- **Send(S_j,s,U_i,M)**: ζ operates according to the specification of the proposed scheme and returns the result of response to A.
- **Reveal(ID_i):** ζ returns the session key K of ID_i to A.
- **Corrupt(ID_i,PW_i)**: ζ returns the password PW_i of ID_i to A.
- **Corrupt(ID_i,MD_i)**: ζ checks whether ID_i and ID_0 are equal. If not, ζ returns (F_i,V_i) to A. Otherwise, ζ aborts the game.
- **Corrupt(S_j)**: ζ returns the state of S_j to A.

Based on these assumptions, if A can forge an authentication message V_2 of ζ, A is able to successfully authenticate to service provider. There may be two cases to forge V_2.

Case 1: A correctly guesses the value of V_2 without knowing M_{ij}. The probability of this case is equal to the probability of hash collision, that is $\frac{1}{2^{l/2}}$, where l is the output bit length of h_3.

Case 2: A obtains M_{ij} and asks the h_3 query. It means that M_{ij} is the solution of the k-mBIDH problem. The probability that ζ solves the k-mBIDH problem is analyzed as follows. In order to make the description clearer, four events will be defined.

E₁: Assume that A attacks at least once among k+1 session, but ζ does not know which one A is going to attack unless ζ does abort in Send query. The probability of this case is $Pr[E_1]=\frac{k^{q_s\text{-}1}}{(k+1)^{q_s}}$, where q_s denotes the number of Send queries.

E₂: A passes user authentication in this session.

E₃: $h_1(ID_0)$ has been chosen correctly from L_{h1}.

E₄: $h_3(ID_i||ID_j||M_{ij})$ has been chosen correctly from L_{h1}.

It is known that $Pr[E_1] \geq \frac{k^{q_s\text{-}1}}{(k+1)^{q_s}}$, then it can be concluded that $Pr[E_2|E_1] \geq \epsilon$, $Pr[E_3|E_1 \wedge E_2] \geq 1/q_{h1}$, $Pr[E_4|E_1 \wedge E_2 \wedge E_3]1/ \geq q_{h3}$, where q_{h1} and q_{h3} denote the number of h_1 queries and h_3 queries. Therefore, the probability of that ζ solves the k-mBIDH problem is computed below.

$$\Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4] = \Pr[E_1]\cdot\Pr[E_2 \mid E_1]\cdot\Pr[E_3 \mid E_1 \wedge E_2 \wedge E_3] \cdot\Pr[E_4 \mid E_1 \wedge E_2 \wedge E_3]$$
$$\geq \frac{k^{q_s\text{-}1}}{(k+1)^{q_s} q_{h1} q_{h2}} \cdot \epsilon \qquad (1)$$

It is clear that the probability of ζ solving the k-mBIDH problem is non-negligible since ϵ is non-negligible. Obviously, it is a contradicts assumption. Therefore, there is no polynomial adversary can forge a legal user's authentication message with non-negligible probability.

Lemma 2. (Secure service provider authentication): In the proposed scheme, if hash functions h_0,h_1,h_3,h_4 are ideal random functions and $\prod_U^s$ has accepted, then no polynomial adversary against the proposed scheme can forge a legal service provider authentication message with a non-negligible probability.

Proof. Assume that the adversary A can forge a legal service provider authentication message with a non-negligible probability ϵ. Then there is a challenger ζ can solve the k-mBIDH problem with a non-negligible probability.

Given an instance {($e_1, e_2, ..., e_k \in Z_q^*$), P, $sk \cdot P$, $y \cdot P$ $\frac{1}{sk+e_1} \cdot P$, $\frac{1}{sk+e_2} \cdot P$,..., $\frac{1}{sk+e_k} \cdot P \in G_1$ } of k-mBIDH problem, the task of ζ is to compute $e(P,P)^{\frac{y}{sk+e_0}}$. ζ sends the system parameters {$G_1, G_2, e, q, g, P, PK, h_0, h_1, h_2, h_3, h_4$} to A. Assume that ID_0 is the identity of challenge. ζ answers the h_i(i=0,1,3,4) query, Register query, Reveal query and Corrupt(ID_i,PW_i) query as he dose in the proof of Lemma 1. Then ζ answers other queries as follows:

- **Send(U_i,s,S_j,M)**: ζ operates according to the specification of the proposed scheme and returns the result of the response to A.
- **Send(S_j,s,U_i,M)**: ζ checks whether $S_j = ID_0$. If they are not equal, ζ operates according to the specification of the proposed scheme and returns {R_2,V_1} to A. Otherwise, ζ aborts the game.
- **Corrupt(ID_i,MD_i)**: ζ returns (F_i,V_i) to A.
- **Corrupt(S_j)**: ζ checks whether $S_j = ID_0$. If they are not equal, ζ returns the state of S_j to A. Otherwise, ζ aborts the game.

Based on these assumptions, if A can forge an authentication message V_1 of ζ, A is able to authenticate to the user successfully . There may be two cases to forge V_1.

Case 1: A correctly guesses the value of V_1 without knowing X. The probability of this case is equal to the probability of hash collision, that is $\frac{1}{2^{l/2}}$, where l is the output bit length of h_3.

Case 2: A obtains X and asks the h_3 query. It means that X is the solution of the k-mBIDH problem. The probability that ζ solves the k-mBIDH problem is analyzed as the proof of Lemma 1. Therefore, the probability of that ζ solves the k-mBIDH problem is computed below.

$$\Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4] \geq \frac{k^{q_s\text{-}1}}{(k+1)^{q_s} q_{h1} q_{h2}} \cdot \epsilon \tag{2}$$

It is clear that the probability of ζ solving the k-mBIDH problem is non-negligible due to ϵ is non-negligible. Obviously, it is a contradicts assumption. Therefore, there is no polynomial adversary can forge a legal service provider's authentication message with non-negligible probability.

Lemma 3. (Secure key agreement): In the proposed scheme, if hash functions h_0,h_1,h_3,h_4 are ideal random functions, $\prod_S^s$ and $\prod_U^s$ have accepted, then no polynomial adversary against the proposed scheme can distinguish between session key and a random number with a non-negligible probability.

Proof. The adversary A asks the h_i(i=0,1,3,4) query, Register query, Send query, Corrupt query and Test query, ζ chooses a random bit $b \in \{0,1\}$. If b=1, ζ returns the session key K to A. Otherwise, ζ returns a random number to A. If A can distinguish between the session key K and a random number, he must know M_{ij}. According to the proof of Lemma 1, if A can obtain M_{ij}, he must know the solution of the k-mBIDH problem. Obviously, it is a contradicts assumption. Therefore, there is no polynomial adversary can distinguish the session key and a random number with a non-negligible probability.

Theorem 1

The proposed scheme is secure mutual authentication protocol, if: (A) h*ash functions* h_0,h_1,h_3,h_4 *are ideal random functions*; *(B) the k-mBIDH problem is hard.*

Proof. Lemma 1 and Lemma 2 show that there is no polynomial adversary can forge a legal user or service provider if *k*-mBIDH problem is hard. According to Lemma 3, the session key is indistinguishable with a random number if *k*-mBIDH problem is hard. Besides, if the adversary has asked corrupt(ID_i, MD_i) query and gets the secret data in the mobile device, the probability of the adversary knowing the password is $256/|D_{PW}|$, which is negligible. Therefore, according to the definition 2 in section 4.1, the proposed scheme is a secure protocol.

4.3 Further Security Analysis of The Proposed Scheme

4.3.1 Resistance to privileged insider attack

In the user registration phase, U_i sends ID_i and $C_i = h_0(ID_i \| PW_i \| b_i)$ to RC, where b_i is a random number unknown to RC. Therefore, the insider cannot guess the password PW_i due to the irreversible property of the one way hash function. Thus, the proposed scheme can resist privilege insider attack.

4.3.2 Resistance to stolen-verifier attack

In the proposed scheme, the MCC service provider has no any information related to the mobile user. Therefore, it is impossible to achieve stolen-verifier attack.

4.3.3 Resistance to user impersonation attack

In the proposed scheme, the adversary has to generate a valid login request (F_i,PW_i) to impersonate as U_i, where F_i stored in the device. It means that an adversary who wants to impersonate the mobile user U_i must obtain the user’s password and mobile device at the same time. In this scheme, if the adversary has got one of the two factors, he/she still does not know another factor. Therefore, the proposed scheme can resist user impersonation attack.

4.3.4 Resistance to replay attack

In the proposed scheme, the random number is used to prevent the replay attack. The random x and y are fresh and different at every authentication. Therefore, when the two participants accept each other, it must be the current session, not previous session. Therefore, the proposed scheme can resist against the replay attack.

4.3.5 Resistance to man-in-the-middle attack

In the improved scheme, the adversary can not forge legal authentication messages without knowing K_i or K_j, which has been proved in Lemma 1 and Lemma 2. Therefore, the proposed scheme can resist against the man-in-the-middle attack.

4.3.6 Two-factor security

Two-factor security is defined as that the protocol is secure even if one of the two factors is leaked. In the proposed scheme, it is obvious that the adversary is unable to impersonation user when he/she only knows the mobile user’s password. On the other hand, suppose the adversary has stolen the user’s mobile device and obtained the data (F_i,V), $F_i = K_i \oplus C_i$, $V = h_3(h_2(C_i))$. The adversary still cannot guess the correct password, because there exist $|D_{PW}|/256$ candidates of the password, where $|D_{PW}|$ is the space of password[19,20]. Therefore, the proposed scheme can provide two-factor security.

4.3.7 Resistance to wrong password login/update attack

In the proposed scheme, secret information $V = h_3(h_2(C_i))$ stored in the mobile device is designed to check user login or password update, where $C_i = h_0(ID_i || PW_i || b_i)$. If the user or adversary inputs wrong password $PW*_i$, the mobile device will reject this login/update request by checking whether V and $V' = h_3(h_2(ID_i || PW*_i || b_i))$ are equal. Therefore, the proposed scheme can quickly detect unauthorized login/update with the wrong password.

4.3.8 User revocation re-registration

In the proposed scheme, RC stores and maintains an identity table of mobile users. Once the mobile device is lost or stolen, the mobile user must revoke his/her account and re-register to RC with a new identity. Besides, if the adversary signs up for registration with the same identity of U_i, RC can verify the whether the identity is valid. Therefore, the revocation and invalid re-registration will be checked[13,37].

4.3.9 Mutual authentication

According to the proofs of Lemma 1 and Lemma 2, there is no polynomial adversary that can forge a legal authentication message. Thus, the user and the service provider can successfully authenticate each other.

4.3.10 User anonymity and un-traceability

In the proposed scheme, U_i sends $CT = ID_i \oplus h_0(X)$ instead of the plain of ID_i to the service provider. Besides, the value of CT changes at every session due to the fresh of X. Anyone who does not know x can not know the value of X. Therefore, our proposed scheme can provide user anonymity and un-traceability.

4.3.11 Perfect forward secrecy

Perfect forward secrecy means that the previously established session keys remain secure when the long-term keys of the mobile user and service provider are disclosed. In the proposed scheme, even if the value K_i and K_j are compromised, the session key K of the previous session remains secure, because the adversary cannot compute M_{ij} without knowing the value of x or y. Therefore, this proposed scheme can achieve perfect forward secrecy.

4.4 Security comparisons

As shown in **Table 1,** this section will compare security features of the proposed scheme with the prior related schemes[14,17,34,35]. **Table 1** demonstrates that the proposed scheme is the only one that is capable of resisting all known attacks and fulfills the desirable security features.

Table 1. Security features comparisons

Attribute	Ref. [14]	Ref. [17]	Ref. [34]	Ref. [35]	Our scheme
Resistance to privileged insider attack	Yes	Yes	Yes	Yes	Yes
Resistance to stolen-verifier attack	Yes	Yes	Yes	Yes	Yes
Resistance to user impersonation attack	Yes	Yes	Yes	Yes	Yes
Resistance to replay attack	Yes	Yes	Yes	Yes	Yes
Multi-factor security	No	No	No	Yes	Yes

Resistance to man-in-the-middle attack	Yes	Yes	Yes	Yes	Yes
Resistance to wrong password login/update attack	No	Yes	Yes	No	Yes
User revocation and re-registration	No	No	Yes	No	Yes
Mutual authentication	No	Yes	Yes	Yes	Yes
User anonymity and un-traceability	Yes	Yes	Yes	Yes	Yes
Perfect forward secrecy	Yes	Yes	Yes	Yes	Yes

4.5 Performance comparisons

The qualitative comparisons between the proposed scheme and the prior related schemes[14,17,34,35] will be summarized in Table 2. The analysis includes using online third party or not, number of security factors, number of rounds, and number of bits. To achieve convincing comparisons in number of bits, assume that the bit length of request login, identity, the validity lifetime L_{ti}[34] and hash output are 32, 32, 32 and 160 bits, the bit length of the element in G_1 and G_2 are 160 and 512 bits, respectively. Therefore, the bit length of an elliptic curve point is 320 bits. In the proposed scheme, the messages $\{R_1,CT\}$,$\{R_2,V_1\}$, and$\{V_2\}$ require (320+160) = 480, (320+160) = 480 and 160 bits, respectively. Adding the three values, the total number communication bits of the proposed scheme is 1120 bits. The total number communication bits of the prior related schemes will be computed using this similar method. From comparison in **Table 2**, it can be concluded that the proposed scheme has the least communication cost among the above schemes.

Table 2. Qualitative comparisons

Attribute	Ref. [14]	Ref. [17]	Ref. [34]	Ref. [35]	Our scheme
Using online third party or not	No	No	No	No	No
Number of security factors	2	3	3	2	2
Number of rounds	4	4	3	4	3
Number of bits	1696 bits	2016 bits	1504 bits	1536 bits	1120 bits

For computational efficiency analysis, the comparison for the time complexity of the proposed scheme and the prior related schemes[14,17,34,35] has been shown in **Table 4**. Because the registration phase and password update phase are not used frequently, only the authenticated key agreement phase can be compared here. To facilitate analysis, the notations are defined as follows.

T_e: the time complexity for exponentiation operation in G_2.

T_b: the time complexity for bilinear paring operation.

T_m: the time complexity for point multiplication operation in G_1.

T_h: the time complexity of the general one-way hash function.

T_{mtp}: the time complexity for map-to-point hash function in G_1;

T_{pa}: the time complexity for point addition operation in G_1;

Almost all of the operations in the proposed scheme and the prior related relevant schemes are appeared in He et.al.'s scheme [35]. Hence, this paper continues to follow these running time. The running time of the mobile device(U_i) and the MCC service provider(S_j) operations will be listed in **Table 3**.

Table 3. Running time of operations(millisecond)

	T_{mtp}	T_b	T_m	T_{pa}	T_e	T_{mul}	T_h
U_i	33.582	32.713	13.405	0.081	2.249	0.008	0.056
S_j	5.493	5.427	2.165	0.013	0.339	0.001	0.007

The results of computational efficiency comparisons are summarized in **Table 4**. From **Table 4**, it is clear that the computation cost of the proposed scheme is lower than the prior related schemes[14,17,34,35].

Table 4. Performance comparisons

Scheme	U_i	S_j	Total cost
Ref. [14]	$T_{mtp}+4T_m+T_e+5T_h+2T_{pa}\approx89.893$	$2T_b+2T_{pa}+2T_m+2T_e+4T_h\approx16.096$	105.989
Ref. [17]	$T_{mtp}+T_b+4T_m+2T_e+8T_h+2T_{pa}\approx125.023$	$2T_b+3T_{pa}+4T_m+2T_e+5T_h\approx20.446$	145.469
Ref. [34]	$2T_{mtp}+3T_m+2T_e+5T_h+T_{pa}\approx112.238$	$2T_b+T_{pa}+T_m+3T_e+T_{mul}+5T_h\approx14.085$	126.323
Ref. [35]	$T_{mtp}+3T_m+3T_e+5T_h+T_{pa}\approx80.905$	$2T_b+T_{pa}+T_m+3T_e+T_{mul}+5T_h\approx14.085$	94.99
Our scheme	$8T_h+2T_m+T_b+2T_e\approx64.469$	$8T_h+2T_m+T_b+2T_e\approx10.131$	74.6

5. Conclusion

This paper proposed an enhanced privacy-aware authentication scheme for distributed MCC services. The proposed scheme can achieve mutual authentication without the help of online RC. Security analysis shows that the proposed scheme can fulfill mutual authentication, user anonymity, perfect forward secrecy, etc. Moreover, it can resist various kinds of known attacks, such as wrong password attack, impersonation attack. At last, the proposed scheme has been compared with the prior related schemes. The comparison results have shown that the proposed scheme not only provides useful security features but also has high computational and communication efficiency. Finally, We thank the anonymous referees for their valuable comments and constructive suggestions which greatly improved the quality of this paper.

References

[1] Z. Qin, J. Sun, A. Wahaballa, W. Zheng, H. Xiong, and Z. Qin, "A secure and privacy-preserving mobile wallet with outsourced verification in cloud computing," *Computer Standards & Interfaces*, vol. 54, Part 1, pp.55-60, 2017. Article (CrossRef Link)

[2] A. A. Mohammed, X. Kong, L. Liu, F. Xia, S. Abolfazli, Z. Sanaei, and A. Tolba., "BoDMaS: Bio-inspired Selfishness Detection and Mitigation in Data Management for Ad-hoc Social Networks," *Ad Hoc Networks*, vol.55, pp.119-131, 2017. Article (CrossRef Link)

[3] Z. Ning, F. Xia, X. Kong, and Z. Chen, "Social-oriented resource management in cloud-based mobile networks," *IEEE Cloud Computing*, vol.3, no. 4, pp.24-31, 2016. Article (CrossRef Link)

[4] C.Doukas, T. Pliakas, L. Maglogiannis, "Mobile healthcare information management utilizing cloud computing and android OS,"in *Proc. of Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, 2010. Article (CrossRef Link)

[5] X. Yang, X. Huang, J. Han, and C. Su,"Improved handover authentication and key pre-distribution for wireless mesh networks," *Concurrency and Computation: Practice and Experience*, vol. 28, no.10, pp. 2978-2990, 2016. Article (CrossRef Link)

[6] B. Alami Milani and N. Jafari Navimipour, "A systematic literature review of the data replication techniques in the cloud environments," *Big Data Research*, 2017. Article (CrossRef Link)

[7] G. Chen, H. Jin, D. Zou, B. B. Zhou, and W. Qiang, "A lightweight software fault-tolerance system in the cloud environment," *Concurrency and Computation: Practice and Experience*, vol. 27, no.12, pp. 2982-2998, 2015. Article (CrossRef Link)

[8] S. Z. Mohammadi and J. N. Navimipour, "Invalid cloud providers' identification using the support vector machine," *International Journal of Next-Generation Computing*, 2017. Article (CrossRef Link)

[9] T. H. Dinh, C. Lee, D. Niyato and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *WIRELESS COMMUNICATIONS AND MOBILE COMPUTING*, vol.13, no.18, pp.1587-1611, 2013. Article (CrossRef Link)

[10] M. R. Rahimi, J. Ren, C. H. Liu, A.V. Vasilakos, N. Venkatasubramanian, "Mobile Cloud Computing: A Survey, State of Art and Future Directions," *Mobile Netw Appl*, vol.19, pp.133-143, 2014. Article (CrossRef Link)

[11] A. R. Khan, M. Othman, S.A. Madani, S. U. Khan, "A Survey of Mobile Cloud Computing Application Models," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol.16, no.1, pp.393-413, 2014. Article (CrossRef Link)

[12] A. N. Khan, M.L. MatKiah, S.U. Khan, S.A. Madani, "Towards secure mobile cloud computing: A survey," *Future Generation Computer Systems*, vol.29, no.5, pp.1278-1299, 2013. Article (CrossRef Link)

[13] Q. Jiang, J.F Ma, And F.S. Wei, "On the Security of a Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services," *IEEE Systems Journal*, vol.99, pp.1-4, 2016. Article (CrossRef Link)

[14] J.L.Tsai and N.W. Lo, "A privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Systems Journal*, vol.9, no.3, pp. 805-815, 2015. Article (CrossRef Link)

[15] Microsoft,Windows Live ID, 2011, [Online]. Available: https://account.live.com/.

[16] OpenID Foundation, OpenID Authentication 2.0, 2007, [Online]. Available: http://openid.net/specs/ openid-authentication-2_0.html

[17] A. Irshad, M. Sher, H.F. Ahmad, B. A. Alzahrani, S. A.Chaudhry, R. Kumar, "An improved Multi-server Authentication Scheme for Distributed Mobile Cloud Computing Services," *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS*, vol.10, no.12, pp.5529-5552, 2016. Article (CrossRef Link)

[18] D. Pointcheval, S. Zimmer, "Multi-factor authenticated key exchange," *Applied cryptography and network security*, 2008. Article (CrossRef Link)

[19] D. Wang, D. He, P. Wang, and C.H. Chu,"Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Dependable Secure Comput.*, vol.12, no.4, pp. 428-442, 2015. Article (CrossRef Link)

[20] D.Wang and P.Wang,"On the usability of two-factor authentication," *in Proc. 10th Int. Conf. Security Privacy Commun*, 2014. Article (CrossRef Link)

[21] C.C. Chang; T.C.Wu, "Remote Password Authentication with Smart Cards," *IEEE Proceedings*, vol.138, pp.165-168, 1991. Article (CrossRef Link)

[22] S. Lee, I.Ong, H.T. Lim, H.J. Lee,"Two factor authentication for cloud computing," *International Journal of KIMICS*, vol.8, no.4, pp. 427-432, 2010. Article (CrossRef Link)

[23] A.J. Choudhury, P. Kumar, M. Sain, et al., "A Strong User Authentication Framework for Cloud Computing," in *Proc. of IEEE Asia -Pacific Services Computing Conference*, 2011. Article (CrossRef Link)

[24] N. Chen, R. Jiang, "Security Analysis and Improvement of User Authentication Framework for Cloud Computing," *Journal of Networks*, vol.9, no.1, pp.198-203, 2014. Article (CrossRef Link)

[25] H.X. Li , F.H. Li, C.G. Song, Y.L. Yan,"Towards Smart Card Based Mutual Authentication Schemes in Cloud Computing," *KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS*, vol.9, no.7, pp.2719-2735, 2015. Article (CrossRef Link)

[26] J. Hughes,"Profiles for the OASIS Security Assertion Markup Language(SAML)V2.0," *OASIS Standard*, 2005. Article (CrossRef Link)

[27] E. Chen, Y. Pei and S. Chen,"OAuth Demystified for Mobile Application Developers," in *Proc. of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, Scottsdale, Arizona, USA, pp.892-903, 2012. Article (CrossRef Link)

[28] A. Armando, R. Carbone, L. Compagna, J. Cuellar, G. Pellegrino, A. Sorniotti,"An authentication flaw in browser-based Single Sign-On protocols," *Impact and remediations*, *Computers and Security*, vol.33, pp.41-58, 2013. Article (CrossRef Link)

[29] S. Halevi and H. Krawczyk, "Public-key cryptography and password protocols," *ACM Trans. Inform. Syst. Secur.*, vol. 2, pp. 230–268, 1999. Article (CrossRef Link)

[30] D.B He, S.Zeadally, N.Kumar, W.Wu, "Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures," *IEEE Transactions on information forensics and Security*, vol.9, pp.2052-2064, 2016. Article (CrossRef Link)

[31] W.B. Hsieh and J.S. Leu,"An anonymous mobile user authentication protocol using self-certified public keys based on multi-server architectures," *The Journal of Supercomputing*, vol.70, no.1, pp.133-148, 2014. Article (CrossRef Link)

[32] R. Amin and G. P. Biswas,"Design and analysis of bilinear pairing based mutual authentication and key agreement protocol usable in multi-server environment," *Wireless Personal Communications*, vol.84, no.1, pp.439-462, 2015. Article (CrossRef Link)

[33] Y.P. Liao and C.M. Hsiao,"A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients," *Future Generation Computer Systems*, vol.29, no.3, pp.886–900,2013. Article (CrossRef Link)

[34] V. Odelu, A. K. Das, S. Kumari, X. Huang, M. Wazid,"Provably secure authenticated key agreement scheme for distributed mobile cloud computing services," *Future Generation Computer Systems*,vol.68, pp.74-88, 2017. Article (CrossRef Link)

[35] D He, N Kumar, MK Khan, L Wang,"Efficient Privacy-Aware Authentication Scheme for Mobile Cloud Computing Services," *IEEE Systems Journal*, vol.PP, no.99, pp.1-11, 2016. Article (CrossRef Link)

[36] R. Amin, S.H. Islam, G.P. Biswas, D. Giri, M.K. Khan,"Kumar N., A more secure and privacy-aware anonymous user authentication scheme for distributed mobile cloud computing environments," *Security and Communication Network*, vol.9, no.17, pp.4650-4666, 2016. Article (CrossRef Link)

[37] V. Odelu, A. Kumar and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Transactions on information forensics and Security*, vol.9, pp.1953-1966, 2015. Article (CrossRef Link)

[38] M. Bellare, D. Pointcheval, and P. Rogaway,"Authenticated key agreement secure against dictionary attacks," *in Proc. of EUROCRYPT*,pp. 139-155, 2000. Article (CrossRef Link)

[39] M. Jakpbsson and D. Pointcheval," Mutual authentication for low-power mobile devices," *in Proc. of FC*, pp. 178-195, 2001. Article (CrossRef Link)

Ling Xiong received Master's degree from Southwest Jiaotong University, Chengdou, Sichuan. Currently, she is a Ph.D. candidate fellow in the school of information science and technology of Southwest Jiaotong University. Her research interests include the formal analysis of cryptographic protocol, provable security technology, and authentication protocol in cloud computing services environment.
(Email: lingdonghua99@gmail.com).

Daiyuan Peng is a professor fellow in the school of information science and technology of Southwest Jiaotong University. His current research is the formal analysis of cryptographic protocol, spread spectrum sequence analysis and design, information theory and coding.
(Email: dypeng@swjtu.edu.cn).

Tu Peng is an associate professor fellow in the school of software of Beijing Institute of Technology. His current research is software reliability, fault localization, and cryptographic protocol.
(Email: pengtu@bit.edu.cn).

Hongbin Liang is an associate professor fellow in School of Transportation and Logistics of Southwest Jiaotong University. His current research is wireless communication, cloud computing, and large data technology.
(Email: hbliang@swjtu.edu.cn).