JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 35, 341-360 (2019)
DOI: 10.6688/JISE.201903_35(2).0006

An Efficient Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services without Bilinear Pairings*

LING XIONG[1,2,+], TU PENG[3], DAI-YUAN PENG[1,2], HONG-BIN LIANG[4] AND ZHI-CAI LIU[2]
[1]*School of Computer Engineering*
Chengdu Technological University
Chengdu, 611730 P.R. China
E-mail: lingdonghua99@gmail.com; idle@gmail.com
[2]*School of Information Science and Technology*
[4]*School of Transportation and Logistics*
Southwest Jiaotong University
Chengdu, 610031 P.R. China
E-mail: lingdonghua99@163.com; {dypeng; hbliang}@swjtu.edu.cn
[3]*School of Software*
Beijing Institute of Technology
Beijing, 100081 P.R. China
E-mail: pengtu@bit.edu.cn

Constructing efficient privacy-aware authentication scheme for Mobile Cloud Computing (MCC) services environment has been an important research topic with ever-increasing mobile devices. However, to the best of our knowledge, most of the schemes for MCC services use heavy bilinear pairings and map-to-point hash operations, which are two very time-consuming operations in modern public key cryptography. In this work, we construct an efficient privacy-aware authentication scheme for MCC services using elliptic curve cryptography. As a result, due to the fact that no bilinear pairings and map-to-point hash operations are involved in the execution, the proposed scheme has much better computation and communication efficiencies than existing related schemes. Detailed performance analysis shows that the computation and the communication costs of our scheme are about 17.44 and 10.64% less than that of the most effective related schemes. Besides, the security analysis shows that our scheme is provably secure in the random oracle model.

Keywords: mobile cloud computing, authentication, privacy, provable security, multi-server

1. INTRODUCTION

Nowadays, mobile devices (*e.g.*, mobile phone, notebook PC and PDA) are becoming more and more important to people as the most effective and convenient communication tools [1]. This triggers a huge demand for high-quality mobile services. However, the service quality is impeded by limited resources (*e.g.*, computing power, storage, bandwidth, and security) of mobile devices. To break through the limitation, cloud computing has been introduced into the mobile environment because it provides shared computation resources and data to computers and other devices on demand [2]. This rev-

Received October 11, 2017; revised February 20, 2018; accepted April 9, 2018.
Communicated by Jiann-Liang Chen.

[+] Corresponding author.

[*] This research was supported by National Science Foundation of China under Grant number (Grant No. 61502034 and 61571375) and the civil aviation administration of China (No. PSDSA201802).

olutionary technology, Mobile Cloud Computing (MCC), provides us a lot of services and brings conveniences to our life and work [1-5].

Nevertheless, due to the openness wireless network, the adversary can easily eavesdrop, insert, block, and alter the transmitted messages in the MCC services environment. As a result, the MCC services environment is more vulnerable to be attacked than the traditional cloud computing services environment. To prevent the malicious adversary from accessing the MCC service provider, it is indispensable to achieve mutual authentications between mobile users and MCC service providers. Additionally, the leakage of users' identities may reveal their locations, movements and purchase preferences, *etc.* Hence, it attracts lots of concerns in the literature to protect user's identities.

To achieve mutual authentication, Lee *et al.* [6] proposed a two-factor single-server authentication scheme for cloud computing environment using public key infrastructure. However, Lee *et al.*'s scheme can be easily intercepted due to the plain form of transmitted message. In order to get better security, several enhanced single-server authentication schemes have been proposed for the cloud computing environment [7, 8]. However, these single-server authentication schemes cannot be directly applied to MCC services environment. The reason is that these schemes require the mobile user to log in each service provider with different identities and passwords. Thus, the user is needed to manage various identities and passwords. To reduce password fatigue from different identities and passwords, single sign-on (SSO) authentication and multi-server authentication have been introduced into MCC services environment [9, 10].

The mobile user can log in multiple servers using a single identity and password in SSO authentication mechanism. At present, OpenID [11], SAML [12] and OAuth [13, 14] are mainly emerging SSO protocols. OpenID is an open protocol, which depends on session cookies as verification mechanism [11]. SAML is one of the most popular SSO protocol, which is mainly used for enterprises and universities [15]. 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 [16]. These SSO authentication schemes bring huge conveniences to mobile users. However, there are two main security problems in them. Firstly, 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 [17, 18]. Secondly, these schemes do not protect users' privacy because the third party always knows to which service provider the mobile user accesses. Thus, the trust third party can easily track mobile users [19]. Meanwhile, most of these SSO authentication schemes establish communication connections through SSL or TLS [20]. SSL or TLS implementation needs heavy computation and communication cost. As a result, these SSO authentication schemes may be unsuitable for MCC services environment.

Multi-server authentication only needs the mobile user to register once at registration center. Then the mobile user can access all registered servers. Li *et al.* [21] proposed the first password-based multi-server authentication scheme in 2001, then a series of remarkable schemes (*e.g.* [22-28]) have been proposed. However, these schemes require a trust third party to participate in each authentication phase. To solve this issue, several multi-server authentication schemes without online third party participation have been proposed [29-33].

The above analysis shows that multi-server authentication without online third-party participation is more suitable for MCC environment. The first MCC authentication scheme using multi-server authentication was proposed by Tsai and Lo [17] in 2015. They claimed that their scheme could support mutual authentication and privacy protection. However, a series of articles [18, 34-37] point out that their scheme cannot meet the desired security goals. It is vulnerable to mutual authentication, user anonymity, and service provider spoofing attack, *etc.* Then, a number of improved schemes [34-37] have been put forward, which have more security advantages than Tsai and Lo's scheme. However, most of them use heavy bilinear pairings and map-to-point hash operations, which are two very time-consuming operations in modern public key cryptography [38]. Besides, 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.*'s scheme [34] is vulnerable to truly three-factor security, as pointed out in [38]. The three factors of this scheme are the smart card, the password PW_i, and the fingerprint f_i, respectively. Obviously, when f_i and the password verification data $D_i = h(ID_i \| PW_i \| H_b(f_i))$ stored in the smart card are leaked, the password is easy to be guessed. Moreover, their scheme suffers from the problems of no user revocation and re-registration, because the trust third party does not maintain an identity information table. As in the reference [34], Odelu *et al.*'s [35] cannot provide truly three-factor security. He *et al.*'s scheme [36] suffers from some wrong password login attack, no user password update phase and no revocation and re-registration. All of the above our schemes [17, 34-36] and multi-server authentication schemes [29-33] use heavy computation operations, such as the bilinear pairings and the map-to-point hash operations, which require much computation cost. Although Amin *et al.*'s scheme [37] is a lightweight privacy-aware authentication protocol. However, it is easy for a malicious user to impersonate a legitimate user or an MCC service provider [39].

To make the above description clearer, we briefly summarize and compare the security features and performance of the above schemes [17, 34-36] in Table 1. The details are described in sections 4.4 and 5.1. Since the Amin *et al.*'s scheme [37] is used in a special environment, we do not compare with it in this paper. In Table 1, the symbol '√' represents that the scheme achieves the corresponding security feature; the symbol '×' denotes that the scheme cannot provide the corresponding security feature.

Table 1. The security and performance analysis of schemes mentioned above.

security features	Ref.[17]	Ref.[34]	Ref.[35]	Ref.[36]
mutual authentication	×	√	√	√
user anonymity and untraceability	×	√	√	√
multi-factor security	√	×	×	√
user revocation and re-registration	×	×	√	×
resistance to MCC service provider spoofing attack	×	√	√	√
resistance to wrong password login/update attack	×	√	√	×

Table 1 shows that the previously proposed schemes mainly exist three defects. Here we give the solution to these flaws.

(1) Multi-factor security and resistance to wrong password login/update attack are two important security properties in the authentication protocol. Multi-factor (assuming there are n factors, generally, $n = 2$ or $n = 3$) security implies the protocol is still secure when $n-1$ of n factors are lost. The wrong password login/update attack mainly

exploits the inefficient login or password update to increase the computation and communication costs of the service provider. However, it is difficult to satisfy the two security features at the same time. This reason is that the password verification value must be stored in the mobile device to prevent incorrect password login/update attack, while it causes the problem of truly multi-factor security. In this paper, we use the 'fuzzy verifier' method proposed by Wang *et al.* [40-42] to address this issue. Since 'fuzzy verifier' method maps the password verification value to {0, 1, 2, …, 1023}, there are $|D_{PW}|/1023$ password candidates, where $|D_{PW}|$ is the space of password. It might occur mapping collision problem. In this case, Wang *et al.* [40-42] pointed out this will rarely occur in reality. Therefore, if the adversary has obtained the mobile device and gets password verification value in the mobile device, he/she cannot guess the correct password.

(2) User revocation and re-registration is a design problem that is easily overlooked. Except for Odelu *et al.*'s scheme [35], each of the schemes mentioned above ignored this security issue. In this work, we take two measures to prevent the problem of user revocation. The first is the blacklist mechanism. Once the user is revoked, the SCG will notify all MCC service providers to add the revoked user to the blacklist. The second is the expiration time method. The mobile user's secret value K_i is bound by an expiration time. When the time expired, the old K_i cannot be used again. For the problem of user re-registration, we use the identity information table stored in the SCG side to avoid it. When the user or the service provider registers with the system, the SCG will check whether the identity has been registered.

(3) All schemes in Table 1 use the heavy bilinear pairings and the map-to-point hash operations, which may cost heavy computation resources. In our paper, motivated by the several existing authentication schemes [43-45] based on elliptic curve cryptography (ECC), we construct a new efficient authentication scheme for MCC services using ECC. Besides, we found that it is unnecessary to utilize the map-to-point hash operation in above-mentioned schemes. Take He *et al.*'s work [36] for example, since the value K_i is in G_1, the designer maps $ID_i||PW_i$ to G_1, which consumes a large amount of computation resource. In fact, we only need to convert the value of S_{ui} into a bit sequence and use the bit sequence of $h(ID_i||PW_i)$ to protect the bit sequence of S_{ui}, that is $E_{ui} = S_{ui} \oplus h(ID_i||PW_i)$, where h is the general hash function. Through this method, the computational efficiency will be significantly enhanced.

Altogether, this paper proposes an efficient privacy-aware authentication scheme for MCC services without bilinear pairings. The main contributions of this paper are summarized as follows:

(1) Our scheme does not require heavy computation operations such as bilinear pairings and the map-to-point hash operations.
(2) Our scheme does not need an online trust third party (SCG) to participate in each user authentication phase.
(3) Security analysis shows that our scheme does not only meet a variety of security requirements for MCC services but also resist various kinds of known attacks.
(4) Compared with the previously related schemes, our scheme provides more security features while demanding less computation and communication costs.

The rest of this paper is organized as follows. Section 2 introduces the security requirements for MCC services. Section 3 presents the detailed procedure of our scheme. Section 4 gives security analysis of our scheme. The computation and communication costs analysis of the proposed scheme are discussed in section 5. Finally, section 6 concludes this paper.

2. SECURITY REQUIREMENTS FOR MCC SERVICES

He *et al.* [36] pointed out MCC services should meet many security requirements, including mutual authentication, user anonymity, untraceability, key establishment, known session key security, perfect forward secrecy, no clock synchronization and resistance to various known attacks. In addition to the security requirements mentioned above, we believe that an authentication scheme for MCC services should also satisfy the following security properties.

(1) Resistance to wrong password login/update attack: To avoid the waste of computation and communication resources for invalid login, it is necessary to check the correctness of the password in the user login phase. Besides, once a mistake occurs in the password update phase, a valid user can no longer log in the service provider using the same mobile device. Therefore, authentication schemes for MCC should consider quick detection mechanism to avoid wasting the service provider's resources [46].

(2) Efficient and user-friendly password update: Users can freely update passwords and should be allowed updating passwords without MCC service provider's assistance [40, 41, 45].

(3) Two-factor security: It ensures that the two-factor scheme for MCC services should be able to satisfy the following requirement; (1) If the adversary has obtained the mobile device and gets the secret value in the mobile device, he should not be able to perform the off-line password guessing attack; (2) The adversary who knows the password should not be able to perform impersonation attack without secret value in the mobile device [40, 41].

(4) User revocation and re-registration: It ensures that the scheme for MCC services should support user revocation and re-registration. If the user's mobile device is lost or stolen, there must be some measures to prevent the adversary to impersonate the user. In other words, if an adversary has obtained the identity of the mobile user, he cannot impersonate the mobile user in the registration phase [18, 28].

3. THE PROPOSED SCHEME

3.1 Initialization Phase

In the initialization phase, the trusted smart card generator (SCG) choose an additive group of point G_1 with order q, and P is a generator of G_1. SCG generates the system private key sk and calculates the system public key $PK=sk \cdot P$. Then SCG chooses five secure hash functions h_0, h_1, h_3, h_4: $\{0, 1\}^* \rightarrow Z_q^*$, h_2: $\{0, 1\}^* \rightarrow \{0, 1, 2, \ldots, 1023\}$. SCG publishes the system parameters $\{G_1, q, P, PK, h_0, h_1, h_2, h_3, h_4\}$.

3.2 Registration Phase

When the user U_i wants to access a MCC service provider, he/she needs to register in SCG first. As shown in Fig. 1, the process of registration is as follows.

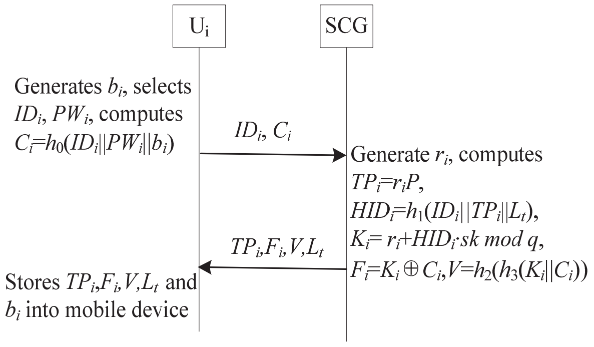

U_i
SCG
Generates b_i, selects ID_i, PW_i, computes C_i=h_0(ID_i||PW_i||b_i)
ID_i, C_i
Generate r_i, computes TP_i=r_iP, HID_i=h_1(ID_i||TP_i||L_t), K_i= r_i+HID_i·sk mod q, F_i=$K_i \oplus C_i$,V=h_2(h_3(K_i||C_i))
TP_i,F_i,V,L_t
Stores TP_i,F_i,V,L_t and b_i into mobile device

Fig. 1. The user registration phase.

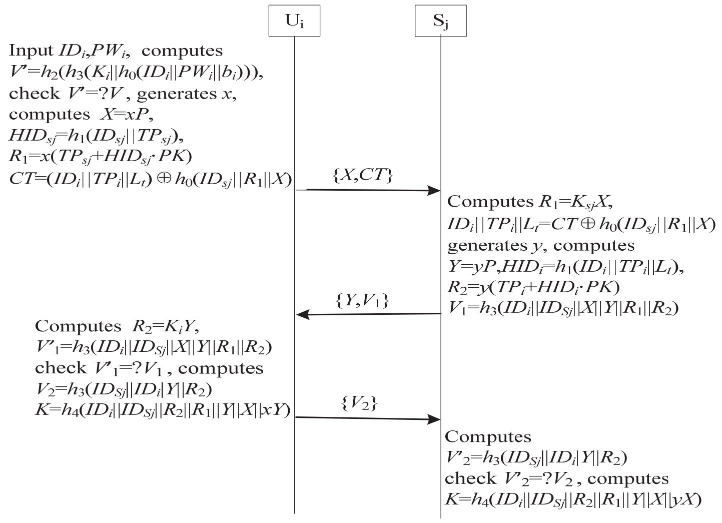
(1) U_i selects identity ID_i and password PW_i, generates a random number b_i. Then U_i computes $C_i = h_0(ID_i||PW_i||b_i)$, U_i transmits $\{ID_i, C_i\}$ to SCG through a secure channel.
(2) SCG checks whether ID_i exists in the user information table. If it exists, SCG rejects this request. Otherwise, SCG generates a random number r_i and computes $TP_i = r_iP$, $HID_i = h_1(ID_i||TP_i||L_t)$, $K_i = r_i + HID_i \cdot sk$ mod q, $F_i = K_i \oplus C_i$, $V = h_2(h_3(IK_i||C_i))$, where L_t is the valid expiration time of secret parameter. After that, SCG updates the user identity information table with the new entry $\{ID_i, L_t\}$, and sends $\{TP_i, F_i, V, L_t\}$ to U_i.
(3) After receiving $\{TP_i, F_i, V, L_t\}$ from SCG, U_i stores them and b_i into the mobile device.

The MCC service provider registration is similar to the user registration phase, the procedure is described as follows.

(1) The mobile service provider S_j selects identity ID_{S_j} and transmits $\{ID_{S_j}\}$ to SCG through a secure channel.
(2) SCG checks whether ID_{S_j} exists in the mobile service provider information table. If it exists, SCG rejects this request. Otherwise, SCG generates a random number r_{S_j} and computes $TP_{S_j} = r_{S_j}P$, $HID_{S_j} = h_1(ID_{S_j}||TP_{S_j})$, $K_{S_j} = r_{S_j} + HID_{S_j} \cdot sk$ mod q. After that, SCG updates the service provider identity information table with the new entry $\{ID_{S_j}\}$, and sends $\{K_{S_j}, TP_{S_j}\}$ to S_j.
(3) After receiving K_{S_j}, TP_{S_j} from SCG, S_j keeps K_{S_j} as secret and declares the message $\{ID_{S_j}, TP_{S_j}\}$ to all users. (Note: The number of service providers is limited, so we can declare them when service providers register.)

3.3 Authentication Phase

When the user U_i wants to log in the MCC service provider S_j, U_i needs to achieve mutual authenticate with S_j. As shown in Fig. 2, the process of mutual authentication is as follows.



U_i
S_j
Input ID_i,PW_i, computes
$V'=h_2(h_3(K_i||h_0(ID_i||PW_i||b_i)))$,
check $V'=?V$, generates x,
computes $X=xP$,
$HID_{sj}=h_1(ID_{sj}||TP_{sj})$,
$R_1=x(TP_{sj}+HID_{sj}\cdot PK)$
$CT=(ID_i||TP_i||L_t)\oplus h_0(ID_{sj}||R_1||X)$
$\{X,CT\}$
Computes $R_1=K_{sj}X$,
$ID_i||TP_i||L_t=CT\oplus h_0(ID_{sj}||R_1||X)$
generates y, computes
$Y=yP$,$HID_i=h_1(ID_i||TP_i||L_t)$,
$R_2=y(TP_i+HID_i\cdot PK)$
$V_1=h_3(ID_i||ID_{Sj}||X||Y||R_1||R_2)$
$\{Y,V_1\}$
Computes $R_2=K_iY$,
$V'_1=h_3(ID_i||ID_{Sj}||X||Y||R_1||R_2)$
check $V'_1=?V_1$, computes
$V_2=h_3(ID_{Sj}||ID_i||Y||R_2)$
$K=h_4(ID_i||ID_{Sj}||R_2||R_1||Y||X||xY)$
$\{V_2\}$
Computes
$V'_2=h_3(ID_{Sj}||ID_i||Y||R_2)$
check $V'_2=?V_2$, computes
$K=h_4(ID_i||ID_{Sj}||R_2||R_1||Y||X||yX)$

Fig. 2. The authentication phase.

(1) U_i inputs ID_i and PW_i into the mobile device. The mobile device computes $C_i=h_0(ID_i||PW_i||b_i)$, $K_i=F_i\oplus C_i$, $V'=h_2(h_3(K_i||C_i)$, and checks whether V' and V are equal. If not, mobile device terminates the session. Otherwise, it generates a random number $x\in Z_q^*$, and computes $X=xP$, $HID_{S_j}=h_1(ID_{S_j}||TP_{S_j})$, $R_1=x(TP_{S_j}+HID_{S_j}\cdot PK)$, $CT=(ID_i||TP_i||L_t)\oplus h_0(ID_{Sj}||R_1||X)$. Then U_i sends $\{X, CT\}$ to S_j through the public channel.

(2) After receiving the message, S_j computes $R_1=K_{S_j}X$, $ID_i||TP_i||L_t=CT\oplus h_0(ID_{Sj}||R_1||X)$. S_j checks the validity of L_t, and generates a random number $y\in Z_q^*$, computes $Y=yP$, $HID_i=h_1(ID_i||TP_i||L_t)$, $R_2=y(TP_i+HID_i\cdot PK)$, $V_1=h_3(ID_i||ID_{S_j}||X||Y||R_1||R_2)$. S_j sends $\{Y, V_1\}$ to U_i through the public channel.

(3) U_i computes $R_2=K_iY$, $V_1'=h_3(ID_i||ID_{S_j}||X||Y||R_1||R_2)$, and checks whether V_1' and V_1 are equal. If not, U_i fails to authenticate S_j. Otherwise, U_i computes $V_2=h_3(ID_{S_j}||ID_i||Y||R_2)$, $K=h_3(ID_i||ID_{S_j}||R_2||R_1||Y||X||xY)$ and transmits V_2 to S_j.

(4) S_j computes $V_2'=h_3(ID_{S_j}||ID_i||Y||R_2)$, and checks whether V_2' and V_2 are equal. If not, S_j fails to authenticate U_i, and the session is terminated. Otherwise, S_j calculates the session key $K=h_3(ID_i||ID_{S_j}||R_2||R_1||Y||X||yY)$, and verifies U_i successfully.

3.4 Password Update Phase

When the 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. The mobile device computes $C_i=h_0(ID_i||PW_i||b_i)$, $K_i=F_i\oplus C_i$, $V'=h_2(h_3(K_i||C_i)$, and checks whether V' and V are equal.

If not, the mobile device fails to authenticate U_i, and rejects the request of the password update. Otherwise U_i inputs a new password PW_i^*.

(2) The mobile device computes $C_i^* = h_0(ID_i||PW_i^*||b_i)$, $F_i^* = F_i \oplus C_i \oplus C_i^*$, $V^* = h_2(h_3(K_i||C_i^*)$.

(3) Finally, F_i^* and V^* are stored in the mobile device to replace F_i and V respectively.

4. SECURITY ANALYSIS

4.1 Security Model

Protocol Participant: Our scheme involves four participants, the registration center SCG, the MCC service provider S_j, the mobile device MD_i and the user U_i. SCG is a trusted third party and it generates secure parameters. S_j is MCC service provider who runs a service assessed by identified users. The user U_i accesses to S_j using mobile device MD_i.

Protocol Execution: Our scheme has four phases: the initialization phase, the registration phase, the authentication phase and the password update phase. The registration phase is assumed to be executed in a secure channel.

Adversary Model: The goal of an adversary A has three goals. One is that A can successfully impersonate U_i authenticating to S_j. The other is that A can successfully impersonate S_j authenticating to U_i. And the last is that A can obtain the session key or distinguish the session key with a random number. We assume that A is a probabilistic polynomial time attacker, and the feasible attacks are summarized as follows:

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

♦ A can obtain one of the two authentication factors, mobile device or password. If A has obtained the mobile device, he can know the secret value in the mobile device. While A has the password, he can offline enumerate the password space $|D_{PW}|$.

♦ A may be another legitimate but malicious user in the system.

♦ A may be a legitimate but malicious MCC service provider.

Security Model: Based on references [17, 36], we propose a security model for our scheme. The security model of our scheme is defined by a game played by the adversary A and a challenger ζ. Let instance $\prod_S^s$ 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 number $h_i \in Z_q^*$ and returns h_i to A.
- **Register(*ID*$_i$):** This query simulates A registration as a legitimate user. A issues identity and receives secret information of the mobile device.
- **Send(*P*, *s*, *P′*, *M*):** This query simulates P' sending 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 to create a new instance.

- **Reveal(ID_i, s):** This query simulates the leakage of session key attack and will output the session key K.
- There are three corruption queries:
  - ♦ **Corrupt(ID_i, PW_i):** This query simulates password leakage attack, and will output the user password PW_i.
  - ♦ **Corrupt(ID_i, MD_i)**:This query simulates the mobile device stolen attack, and will output the secret information stored in the mobile device MD_i.
  - ♦ **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 $\prod_P^{s'}$ are said to be matching if $s = s'$, $pid_U = S$, $pid_S = U$ and both have accepted, where pid_U and pid_S denote as a peer identity.

Definition 2: Secure protocol: we say that our scheme is secure if the following properties hold:

- ♦ $\prod_U^s$ and $\prod_S^s$ are matching session and they accept each other.
- ♦ The probability of $\prod_S^s$ accepted A as $\prod_U^s$ is negligible.
- ♦ The probability of $\prod_U^s$ accepted A as $\prod_S^s$ is negligible.
- ♦ The probability of distinguishing session key from a random number is negligible.
- ♦ When A has obtained the secret key in the mobile device, the probability of A knowing the password is negligible.

4.2 Provable Security

To prove the security of our proposed scheme, we assume that our scheme is defined by a game played between an adversary A and a challenger ζ. At first, we give two mathematical problems used for our security analysis.

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: The Computational Diffie-Hellman (CDH) Problem: Given $X=x \cdot P$, $Y=y \cdot P$, where x, $y \in Z_q^*$, X, $Y \in G_1$, it is infeasible to compute $xy \cdot P$.

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

Proof: We assume that the adversary A can forge a legal authentication message with a non-negligible probability ϵ. Then there is a challenger ζ who can solve the CDH problem with a non-negligible probability.

Given an instance (P, $A = K_iP$, $B = yP$) of CDH problem, the task of ζ is to compute yK_iP. ζ sends the system parameters $\{G_1, q, P, PK, h_0, h_1, h_2, h_3, h_4\}$ to A. ζ randomly selects a user's identity ID_C as the challenge identity and answers A's queries as follows:

- $\boldsymbol{h_i(m_i)}$**:** The hash query $h_i(m_i)$, $i = 0, 1, 3, 4$ maintains a list L_{hi} with 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, and 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 $ID_i = ID_C$, ζ generates four random numbers r_i, ε_i, c_i, $v_i \in Z_q^*$, computes $TP_i = r_i \cdot P$, sets $h_1(ID_i \| TP_i \| L_t) = \varepsilon_i$, $h_0(ID_i \| PW_i \| b_t) = c_i$, $K_i = \perp$, $F_i = K_i \oplus c_i$, $V = v_i$, and stores (ID_i, r_i, TP_i, K_i, F_i, V, L_t) into L_R, ($ID_i \| TP_i \| L_t$, ε_i) into L_{h1}, ($ID_i \| PW_i \| b_i$, c_i) into L_{h0}, and ($K_i \| h_2$ (c_i),v_i) into L_{h3} respectively. ζ returns ID_i to A.
  ♦ If $ID_i \neq ID_C$, ζ generates four random numbers r_i, ε_i, c_i, $v_i \in Z_q^*$, computes $TP_i = r_i \cdot P - \varepsilon_i \cdot PK$, sets $h_1(ID_i \| TP_i \| L_t) = \varepsilon_i$, $h_0(ID_i \| PW_i \| b_i) = c_i$, $K_i = r_i$, $F_i = K_i \oplus c_i$, $V = v_i$, and stores (ID_i, r_i, TP_i, K_i, F_i, V, L_t) into L_R, ($ID_i \| TP_i \| L_t$, ε_i) into L_{h1}, ($ID_i \| PW_i \| b_i$, c_i) into L_{h0}, and ($K_i \| h_2(c_i)$,v_i) into L_{h3} respectively. ζ returns ID_i to A.
- **Send(*ID$_i$*, *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 $\{X, CT\}$ to A. Otherwise, ζ checks whether $ID_i = ID_0$. 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*, *ID$_i$*, *M*):** ζ operates according to the specification of the proposed scheme and returns the result of response to A.
- **Reveal(*ID$_i$*, *s*):** ζ returns the session key K of ID_i in session s to A.
- **Corrupt(*ID$_i$*, *PW$_i$*):** ζ returns the password PW_i of ID_i to A.
- **Corrupt(*ID$_i$*,*MD$_i$*):** ζ searches whether the tuple (ID_i, r_i, TP_i, K_i, F_i, V, L_t) in the L_R, if exist, return (F_i, V, L_t) to A, otherwise, A asks Register(ID_i) query.
- **Corrupt(*S$_j$*):** ζ returns the state of S_j to A.

Based on the above queries, if A is able to forge the authentication message V_2 of ζ, A can successfully authenticate to MCC service provider. There may be two cases to forge the message V_2.

Case 1: A can guess V_2 correctly without knowing R_2. The probability of this case equals to the probability of hash collision. That is $1/2^{l/2}$, where l is the output bit length of h_3.
Case 2: A gets R_2 and asks the h_3 query. It means that R_2 is the solution of the CDH problem. The probability ζ solving the CDH problem is analyzed as follows. In order to be easy to explain, we define four events as follows.

E$_1$: ζ knows which one A is going to attack.
E$_2$: A passes user authentication in this session.
E$_3$: $h_1(ID_0 \| TP_i \| L_t)$ has been chosen correctly from L_{h1}.

E$_4$: $h_3(ID_{S_j}||ID_0|Y||R_2)$ has been chosen correctly from L_{h3}.

We 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 aborts in Send query. We have known that $Pr[E_1] \geq \frac{k^{q_s\text{-}1}}{(k+1)^{q_s}}$, then we can get $Pr[E_2|E_1] \geq \epsilon$, $Pr[E_3|E_1 \wedge E_2] \geq 1/q_{h1}$, $Pr[E_4|E_1 \wedge E_2 \wedge E_3] \geq 1/q_{h3}$, where q_{h1}, q_{h3} and q_s denote the number of h_1 query, h_3 query and Send query. Therefore, the probability of ζ solving the CDH problem is computed as below.

$$\begin{aligned} Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4] &= Pr[E_1] \cdot Pr[E_2|E_1] \cdot Pr[E_3|E_1 \wedge E_2] \cdot Pr[E_4|E_1 \wedge E_2 \wedge E_3] \\ &\geq \frac{k^{q_s\text{-}1}}{(k+1)^{q_s} q_{h1} q_{h3}} \cdot \epsilon \end{aligned} \tag{1}$$

It is clear that the probability of ζ solving the CDH problem is non-negligible because ϵ is non-negligible. Obviously, it is a contradictory assumption. Therefore, there is no polynomial adversary can forge a legitimate user's authentication message with a non-negligible probability.

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

***Proof*:** We assume that the adversary A can forge a legal MCC service provider authentication message with a non-negligible probability ϵ. Then there is a challenger ζ who can solve the CDH problem with a non-negligible probability.

Given an instance $(P, A = K_{S_j}P, B = xP)$ of CDH problem, the task of ζ is to compute $xK_{S_j}P$. ζ sends the system parameters $\{G_1, q, P, PK, h_0, h_1, h_2, h_3, h_4\}$ to A. Assuming 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 query as he does 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 response to A.

✧ **Send(S_j, s, U_i, M)**: ζ checks whether $S_j = ID_0$ holds. If not, ζ operates according to the specification of the proposed scheme and returns $\{Y, V_1\}$ to A. Otherwise, ζ aborts the game.

Based on the above queries, if A can forge the message $\{Y, V_1\}$ of ζ, A is able to successfully authenticate to the user. There may be two cases to forge $\{Y, V_1\}$.

Case 1: A can guess the value of V_1 correctly without knowing R_1. The probability of this case is equal to the probability of hash collision. That is $1/2^{l/2}$, where l is the output bit length of h_3.

Case 2: A gets R_1 and asks the h_3 query. It means that R_1 is the solution of the CDH problem. The probability of ζ solving the CDH problem is analyzed as the proof of Lemma 1. Therefore, the probability is computed as below.

$$Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4] \geq \frac{k^{q_s\text{-}1}}{(k+1)^{q_s} q_{h1} q_{h3}} \cdot \epsilon \tag{2}$$

It is evident that the probability of ζ solving the CDH problem is non-negligible because ϵ is non-negligible. Obviously, it is a contradictory assumption. Therefore, there is no polynomial adversary can forge a legal MCC service provider's authentication message with a non-negligible probability.

Lemma 3 (Secure key agreement): In our proposed scheme, if h_0, h_1, h_3, h_4 are ideal random functions, $\prod_S^s$ and $\prod_U^s$ have been accepted, then there is no polynomial adversary against the proposed scheme can distinguish the 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 K with a random number, he must know R_1 and R_2. According to the proof of Lemmas 1 and 2, if A obtains R_1 and R_2, he must know the solution of the CDH problem. Obviously, it is a contradictory assumption. Therefore, there is no polynomial adversary can distinguish the session key and a random number with a non-negligible probability.

Theorem 1: Our proposed scheme is a secure protocol, if: (A) $\prod_S^s$ and $\prod_U^s$ have been accepted; (B) h_0, h_1, h_3, h_4 are ideal random functions; (C) the CDH problem is hard.

Proof: Based on Lemmas 1 and 2, we can know that there is no polynomial adversary who can forge a legal user or MCC service provider login if CDH problem is hard. Besides, even if we have asked corrupt(ID_i, MD_i) query and known the secret value in the mobile device, the probability of the adversary knowing the password is $1024/|D_{PW}|$, which is negligible. According to Definition 2 in Section 4.1, the proposed scheme is a secure protocol.

4.3 Further Security Analysis of Our Scheme

4.3.1 Mutual authentication

According to the proofs of Theorem 1, no polynomial adversary can forge a legitimate user's or a MCC service provider's authentication message. Thus, the user and the MCC service provider can successfully authenticate each other.

4.3.2 User anonymity and untraceability

In our scheme, the user's ID_i is protected by the secret value R_1. That is $CT=(ID_i||$

$TP_i||L_t) \oplus h_0(ID_{S_j}||R_1||X)$. According to Lemma 1, there is no polynomial adversary can get R_1 because the CDH problem is difficult. Thus, it is almost impossible for an adversary to get ID_i from CT, which implies our scheme is able to support user anonymity and untraceability.

4.3.3 Known session key security

In our scheme, the value $X=xP$ and $Y=yP$ are fresh and different at every session. If the adversary got the session keys in previous sessions, he/she could not compute the current session key without knowing the value of R_1 or R_2, because CDH problem is hard. Therefore, our scheme can provide known session key security.

4.3.4 Perfect forward security

In our scheme, the value $X=xP$ and $Y=yP$ are fresh and different at every session. If the adversary has obtained the private keys of the mobile user and the MCC service provider, he/she still cannot compute the session key $K = h_4(ID_i||ID_{S_j}||R_2||R_1||Y||X||xY)$ without the value x and y in previous sessions. Therefore, our scheme can provide perfect forward security.

4.3.5 Two-factor security

Obviously, the adversary cannot forge a legitimate mobile user when he only knows the mobile user's password. On the other hand, when mobile device is not secure, which mean the user's mobile device is lost or stole by the adversary A, A obtains the data (F_i, V, L_t), $F_i = K_i \oplus C_i$, $V = h_2(h_3(K_i||C_i))$. However, A still cannot guess the correct password, because there exist $|D_{PW}|/1024$ candidates of password, where $|D_{PW}|$ is the space of password. This method is called 'fuzzy verifier' [40, 41], which prevents the adversary from obtaining the exacting correct password. Therefore, our proposed scheme can provide two-factor security.

4.3.6 User revocation and re-registration

In our scheme, SCG 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 SCG with a new identity. In this case, the adversary A has obtained the secret data in the mobile device. Accord to the analysis of two-factor security, A still can not get the identity and password of legal users. If A obtains the identity and password of legal users through another way, the user may launch the revocation to SCG. Then SCG informs all MCC service providers that the old identity cannot log in [18, 28]. Besides, we use the expiration time method. The mobile user's secret value K_i is bound by an expiration time L_t. When L_t expired, the old K_i cannot be used again. For the problem of user re-registration, we use the identity information table stored in the SCG side to avoid it. Therefore, the proposed scheme can provide user revocation and re-registration.

4.3.7 Privileged insider attack

In the user registration phase, U_i sends ID_i and C_i instead of PW_i to SCG, where $C_i = h_0(ID_i||PW_i||b_i)$, and b_i is unknown to SCG. In this process, the insider cannot access the password PW_i due to the irreversible property of the one-way hash function. Thus, our scheme can resist against privilege insider attack.

4.3.8 Stolen verifier table attack

In our scheme, no any verifier table of the mobile user is stored in the MCC service provider side. Therefore, our scheme can resist stolen verifier table attack.

4.3.9 User impersonation attack

In our scheme, in order to forge U_i, the adversary has to generate a valid value V_2. However, Lemma 1 shows that it is infeasible due to the hardness CDH problem. Therefore, our proposed scheme can resist against user impersonation attack.

4.3.10 MCC service provider spoofing attack

Theorem 1 shows that no polynomial adversary can forge a legitimate mobile user's or a MCC service provider's authentication message without the secret value K_i or K_{S_j}. In our scheme, the MCC service provider only has his own secret value and does not know the secret value of other MCC service provider and all mobile users. Therefore, he cannot spoof any user to other MCC service provider.

4.3.11 Replay attack

Our scheme uses challenge-response mechanism to prevent the replay attack. The random number x and y is fresh and different at every session. Therefore, when the mobile user and the MCC service provider accept each other, it must be the current session, not previous session. So, our scheme can avoid the replay attack.

4.3.12 Man-in-the-middle attack

In our scheme, the message transmitted is protected by R_1 and R_2, anyone without K_i or K_j cannot forge legal authentication message. Therefore, our scheme can resist man-in-the-middle attack.

4.3.13 Wrong password login/update attack

In our scheme, the password verification data $V = h_2(h_3(K_i||C_i))$ is stored in the mobile device, which is designed to check the correctness of password. If the user inputs a wrong password PW_i^*, the sotred V and $V' = h_2(h_3(K_i^*||h_0(ID_i||PW_i^*||b_i)))$ will not be equal [41]. So, our scheme can quickly detect unauthorized login and password update.

4.4 Security Comparisons

In this section, we compare security features of our scheme with the prior related schemes [17, 34-36]. The results of the comparison are listed in Table 2. From Table 2, we can see that our scheme is the only one which is able to resist against all known attacks and fulfill the desirable security features. Therefore, our scheme has better security than the previously related schemes.

Table 2. Security features comparisons.

security features	Ref.[17]	Ref.[34]	Ref.[35]	Ref.[36]	Ours
mutual authentication	×	√	√	√	√
user anonymity and untraceability	×	√	√	√	√
known session key security	√	√	√	√	√
perfect forward security	√	√	√	√	√
multi-factor security	√	×	×	√	√
user revocation and re-registration	×	×	√	×	√
resistance to privileged insider attack	√	√	√	√	√
resistance to stolen verifier table attack	√	√	√	√	√
resistance to user impersonation attack	√	√	√	√	√
resistance to MCC service provider spoofing attack	×	√	√	√	√
resistance to replay attack	√	√	√	√	√
resistance to man-in-the-middle attack	√	√	√	√	√
resistance to wrong password login/update attack	×	√	√	×	√

√: achieve the corresponding security feature; ×: cannot provide the corresponding security feature.

5. PERFORMANCE ANALYSIS

5.1 Computation Analysis

For efficiency analysis, we compare the computation cost of our scheme with the prior related schemes [17, 34-36]. Because initialization phase, registration phase, and password update phase are not used frequently, we only compare authenticated key agreement phase. Almost all of the operations in our scheme and the prior related schemes are appeared in He *et al.*'s scheme [36], as shown in Table 3. We continue to follow the running time of all operations in He *et al.*'s scheme. To facilitate analysis, we use the following notations and their running time to measure computation cost.

The results of computation cost comparisons are summarized in Table 4. Table 4 shows that the computation cost of our scheme has much better computation efficiency than existing related schemes.

Table 3. Running time of operations (millisecond).

	T_{mtp}	T_{bp}	T_{sm}	T_{pa}	T_{exp}	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

T_{mtp}: the time complexity for map-to-point hash function in G_1; T_{bp}: the time complexity for bilinear pairing operation; T_{sm}: the time complexity for point multiplication operation in G_1; T_{pa}: the time complexity for point addition operation in G_1; T_{exp}: the time complexity for exponentiation operation in G_2; T_{mul}: the time complexity for multiplication operation in G_2; T_h: the time complexity of general hash function.

Table 4. Computation cost comparisons.

Scheme	U_i	S_i	Total	Improvement
Ref.[17]	$T_{mtp}+4T_{sm}+T_{exp}+5T_h+2T_{pa}$ $\approx$89.893s	$2T_{bp}+2T_{pa}+2T_{sm}+2T_{exp}+$ $4T_h\approx$16.096s	105.989s	26%
Ref.[34]	$T_{mtp}+T_{bp}+4T_{sm}+2T_{exp}+8T_h$ $+2T_{pa}\approx$125.023s	$2T_{bp}+3T_{pa}+4T_{sm}+2T_{exp}+$ $5T_h\approx$20.446s	145.469s	46.09%
Ref.[35]	$2T_{mtp}+3T_{sm}+2T_{exp}+5T_h+T_{pa}$ $\approx$112.238s	$2T_{bp}+T_{pa}+T_{sm}+3T_{exp}+T_{mul}$ $+5T_h\approx$14.085s	126.323s	37.92%
Ref.[36]	$T_{mtp}+3T_{sm}+3T_{exp}+5T_h+T_{pa}$ $\approx$80.905s	$2T_{bp}+T_{pa}+T_{sm}+3T_{exp}+T_{mul}$ $+5T_h\approx$14.085s	94.99s	17.44%
Ours	$8T_h+5T_{sm}+T_{pa}\approx$67.554s	$5T_h+5T_{sm}+T_{pa}\approx$10.873s	78.427s	–

5.2 Communication Analysis

In this section, we compare communication cost of our proposed scheme with the prior related schemes [17, 34-36]. To achieve convincing comparisons, we assume that the bit length of request login, identity, L_t 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. Table 5 shows the communication cost comparison among our scheme and the prior related schemes. In our scheme, the message $\{X, CT\}$, $\{Y, V_1\}$, and $\{V_2\}$ require (320+32+320+32) = 704, (320+160) = 480 and 160 bits, respectively. Adding the three values, we get the total communication cost of our scheme 1344 bits. From comparison in Table 5, we can conclude that our scheme has the least communication cost among the above schemes.

Table 5. Communication cost comparisons.

Scheme	Ref.[17]	Ref.[34]	Ref.[35]	Ref.[36]	Ours
Number of rounds	4	4	3	4	3
Number of bits	1696 bits	2016 bits	1504 bits	1536 bits	1344 bits
Improvement	20.75%	33.33%	10.64%	12.5%	–

6. CONCLUSIONS

To enhance the computation and communication efficiency, in this paper, we propose an efficient and provably secure privacy-aware authentication scheme for MCC services without bilinear pairings. The proposed scheme address security issues such as wrong password login/update attack and truly multi-factor security existing in most of the prior related schemes. The security analysis shows that our scheme is able to resist against various kinds of attacks and fulfills the desirable security requirements. The performance analysis shows that our scheme has much better computation and communication efficiencies than existing related schemes. High security and efficiency indicate that our scheme is more suitable for the lightweight mobile device than previously proposed schemes.

REFERENCES

1. T. 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, 2013, pp. 1587-1611.

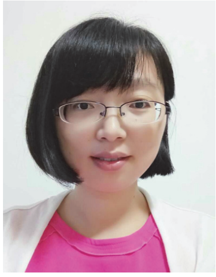
2. K. D. Chang, C. Y. Chen, J. L. Chen, and H. C. Chao, "Internet of things and cloud computing for future internet," in *Proceedings of International Conference on Security-Enriched Urban Computing and Smart Grid*, 2011, pp. 1-10.
3. C. K. Chu, W. T. Zhu, J. Han, J. K Liu, J. Xu, and J. Zhou, "Security concerns in popular cloud storage services," *IEEE Pervasive Computing*, Vol. 12, 2013, pp. 50-57.
4. J. Baek, Q. H. Vu, J. K. Liu, X. Huang, and Y. Xiang, "A secure cloud computing based framework for big data information management of smart grid," *IEEE Transactions on Cloud Computing*, Vol. 3, 2015, pp. 233-244.
5. F. Xhafa, J. Wang, X. Chen, *et al.*, "An efficient PHR service system supporting fuzzy keyword search and fine-grained access control," *Soft Computing*, Vol. 18, 2014, pp. 1795-1802.
6. S. Lee, I. Ong, H. T. Lim, and H. J. Lee, "Two factor authentication for cloud computing," *International Journal of KIMICS*, Vol. 8, 2010, pp. 427-432.
7. A. J. Choudhury, P. Kumar, M. Sain, *et al.*, "A strong user authentication framework for cloud computing," in *Proceedings of IEEE Asia-Pacific Services Computing Conference*, 2011, pp. 110-115.
8. N. Chen and R. Jiang, "Security analysis and improvement of user authentication framework for cloud computing," *Journal of Networks*, Vol. 9, 2014, pp. 198-203.
9. S. Ahmad and B. Ehsan, "The cloud computing security secure user authentication technique," *International Journal of Scientific and Engineering Research*, Vol. 4, 2013, pp. 2166-2171.
10. N. Fotiou, A. Machas, G. C. Polyzos, and G. Xylomenos, "Access control as a service for the Cloud," *Journal of Internet Services and Applications*, Vol. 6, 2015, pp. 1-15.
11. OpenID Foundation, "OpenID authentication 2.0," http://openid.net/specs/openid-authentication-2_0.html, 2007.
12. J. Hughes, "Profiles for the OASIS security assertion markup language (SAML) V2.0," http://docs.oasis-open.org/security/saml/v2.0/, 2005.
13. Internet Engineering Task Force, "The oauth 1.0 protocol," http://tools.ietf.org/html/rfc5849.
14. Internet Engineering Task Force (IETF), "The oauth 2.0 authorization framework," http://tools.ietf.org/html/rfc6749.
15. K. D. Lewis and J. E. Lewis, "Web single-sign on authentication using SAML," *International Journal of Computer Science*, Vol. 1, 2009, pp. 41-48.
16. E. Chen, Y. Pei, and S. Chen, "OAuth demystified for mobile application developers," in *Proceedings of ACM SIGSAC Conference on Computer and Communications Security*, 2012, pp. 892-903.
17. J. L. Tsai and N. W. Lo, "A privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Systems Journal*, Vol. 9, 2015, pp. 805-815.
18. 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. 12, 2016, pp. 2039-2042.
19. D. Fett, R. Küsters, and G. Schmitz, "SPRESSO: A secure, privacy-respecting single

sign-on system for the web," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1358-1369.

20. A. Armando, R. Carbone, L. Compagna, *et al.*, "An authentication flaw in browser-based single sign-on protocols, impact and remediations," *Computers and Security*, Vol. 33, 2013, pp. 41-58.
21. L. Li, I. Lin, and M. Hwang, "A remote password authentication scheme for multi-server architecture using neural networks," *IEEE Transactions on Neural Networks*, Vol. 12, 2001, pp. 1498-1504.
22. I. Lin, M. Hwang, and L. Li, "A new remote user authentication scheme for multi-server architecture," *Future Generation Computer Systems*, Vol. 19, 2003, pp. 13-22.
23. W. J. Tsaur, C. C. Wu, and W. B. Lee, "A smart card-based remote scheme for password authentication in multi-server internet services," *Computer Standards and Interfaces*, Vol. 27, 2004, pp. 39-51.
24. J. L. Tsai, "Efficient multi-server authentication scheme based on one-way hash function without verification table," *Computers and Security*, Vol. 27, 2008, pp. 115-121.
25. C. T. Li, C. C. Lee, C. Y. Weng, and C. I. Fan, "A secure dynamic identity based authentication protocol with smart cards for multi-server architecture," *Journal of Information Science and Engineering*, Vol. 31, 2015, pp. 1975-1992.
26. E. J. Yoon and K. Y. Yoo, "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem," *The Journal of Supercomputing*, Vol. 63, 2013, pp. 235-255.
27. D. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Systems Journal*, Vol. 9, 2015, pp. 816-823.
28. V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Transactions on Information Forensics and Security*, Vol. 10, 2015, pp. 1953-1966.
29. Y. H. Chuang and Y. M. Tseng, "Towards generalized ID-based user authentication for mobile multi-server environment," *International Journal of Communication Systems*, Vol. 25, 2012, pp. 447-460.
30. 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, 2013, pp. 886-900.
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, 2014, pp. 133-148.
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, 2015, pp. 439-462.
33. D. He, S. Zeadally, N. Kumar, and 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. 11, 2016, pp. 2052-2064.
34. A. Irshad, M. Sher, H. F. Ahmad, B. A. Alzahrani, S. A. Chaudhry, and R. Kumar, "An improved multi-server authentication scheme for distributed mobile cloud computing services," *KSII Transactions on Internet and Information Systems*, Vol. 10,

2016, pp. 5529-5552.

35. V. Odelu, A. K. Das, S. Kumari, X. Huang, and M. Wazid, "Provably secure authenticated key agreement scheme for distributed mobile cloud computing services," *Future Generation Computer Systems*, Vol. 68, 2017, pp. 74-88.
36. D. He, N. Kumar, M. K. Khan, and L. Wang, "Efficient privacy-aware authentication scheme for mobile cloud computing services," *IEEE Systems Journal*, Vol. PP, 2016, pp. 1-11.
37. R. Amin, S. H. Islam, G. P. Biswas, *et al.*, "A more secure and privacy-aware anonymous user authentication scheme for distributed mobile cloud computing environments," *Security and Communication Network*, Vol. 9, 2016, pp. 4650-4666.
38. X. Huang, Y. Xiang, A. Chonka, *et al.*, "A generic framework for three-factor authentication: Preserving security and privacy in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, 2011, pp. 1390-1397.
39. L. Xiong, D. Peng, T. Peng, and H. Liang, "An enhanced privacy-aware authentication Scheme for Distributed Mobile Cloud Computing Services," *KSII Transactions on Internet and Information Systems*, Vol. 11, 2017, pp. 6196-6187.
40. D. Wang, D. He, P. Wang and C. H. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Transactions on Dependable and Secure Computing*, Vol. 12, 2015, pp. 428-442.
41. D. Wang and P. Wang, "On the usability of two-factor authentication," in *Proceedings of the 10th International Conference on Security Privacy Communication*, 2014, pp. 141-150.
42. D. Wang and P. Wang, "Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks," *Ad Hoc Network*, Vol. 20, 2014, pp. 1-15.
43. D. He, H. Wang, M. K. Khan, and L. Wang, "Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography," *IET Communications*, Vol. 10, 2016, pp. 1795-1802.
44. Y. M. Tseng, S. S. Huang, and M. L. You, "Strongly secure ID-based authenticated key agreement protocol for mobile multi-server environments," *International Journal of Communication systems*, Vol. 30, 2017, pp. 1-13.
45. S. K. Islam and M. K. Khan, "Provably secure and pairing-free identity-based handover authentication protocol for wireless mobile networks," *International Journal of Communication systems*, Vol. 29, 2016, pp. 2242-2456.
46. D. Mishra, "On the security flaws in ID-based password authentication schemes for telecare medical information systems," *Journal of Medical Systems*, Vol. 39, 2015, pp. 1-16.



Ling Xiong (熊玲) received the M.S. degree in the School of Information Science and Technology of Southwest Jiaotong University. He is currently pursuing the Ph.D. degree in the School of Information Science and Technology of Southwest Jiaotong University. Her research interests include the formal analysis of cryptographic protocol, the security and privacy in cloud computing services environment and wireless sensor networks environment.

Tu Peng (彭图) 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.

Dai-Yuan Peng (彭代渊) 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.

Hong-Bin Liang (梁宏斌) 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.

Zhi-Cai Liu (刘志才) is an Associate Professor Fellow in School of Computer and Software Engineering, Xihua university. His current research is the formal analysis of cryptographic protocol, intrusion detection.