Research Article

An Efficient and Provably Secure Anonymous User Authentication and Key Agreement for Mobile Cloud Computing

Jiaqing Mo, Zhongwang Hu, Hang Chen, and Wei Shen

School of Computer Science and Software, Zhaoqing University, Zhaoqing, China

Correspondence should be addressed to Jiaqing Mo; mojiaqing@126.com

Received 10 September 2018; Revised 2 December 2018; Accepted 19 December 2018; Published 4 February 2019

Academic Editor: Rüdiger C. Pryss

Copyright © 2019 Jiaqing Mo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nowadays, due to the rapid development and wide deployment of handheld mobile devices, the mobile users begin to save their resources, access services, and run applications that are stored, deployed, and implemented in cloud computing which has huge storage space and massive computing capability with their mobile devices. However, the wireless channel is insecure and vulnerable to various attacks that pose a great threat to the transmission of sensitive data. Thus, the security mechanism of how the mobile devices and remote cloud server authenticate each other to create a secure session in mobile cloud computing environment has aroused the interest of researchers. In this paper, we propose an efficient and provably secure anonymous two-factor user authentication protocol for the mobile cloud computing environment. The proposed scheme not only provides mutual authentication between mobile devices and cloud computing but also fulfills the known security evaluation criteria. Moreover, utilization of ECC in our scheme reduces the computing cost for mobile devices that are computation capability limited and battery energy limited. In addition, the formal security proof is given to show that the proposed scheme is secure under random oracle model. Security analysis and performance comparisons indicate that the proposed scheme has reasonable computation cost and communication overhead at the mobile client side as well as the server side and is more efficient and more secure than the related competitive works.

1. Introduction

Mobile cloud computing (MCC) is introduced as services of cloud computing, which is offered in mobile devices such as smart phones and tablets environment [1]. In MCC, mobile users can access resources, applications, and running results stored in the cloud and can deploy and implement a variety of services through cloud computing, enabling mobile devices to increase computing power and to increase storage capacity and contextual awareness. According to Mordor Intelligence’s research, in 2023 the MCC market will generate revenues of $94.75 billion (online, 2018) [2]. However, wireless channels supporting communication between mobile devices and the cloud service providers are insecure and are vulnerable to many kinds of attacks like impersonation attack, replay attack, and interception (see Figure 1). Additionally, when the mobile devices access cloud computing services, seamless connectivity will be required while roaming across the heterogeneous network, but their security policies vary greatly which leads to inefficiency. In addition, mobile devices have relatively limited computation capability and energy as compared with traditional computers or laptops. Therefore, a secure and efficient authentication mechanism between the mobile devices and the cloud service provider to ensure the legitimacy of each other is indispensable in preventing illegal access and withstanding potential attacks through wireless channels by the adversary.

According to the comments above, two primary issues should be considered in designing a remote user authentication scheme for mobile devices in MCC:

1) Security. Since the authentication request and the relevant messages are transmitted over public channel, the roaming authentication mechanism verifies the
identity legitimacy of mobile devices while withstanding the well-known attacks launched by the adversary so as to ensure that the private data such as the identity and the geographical location are not leaked and tracked.

(2) Efficiency. Efficiency should be taken intensively into account for the mobile devices. As mentioned above, mobile devices are constrained by computation capability and energy, and the authentication process passes through the heterogeneous network, which means latency and packet loss.

Therefore, improving security and reducing computation cost and communication overhead are very important for developing a practical authenticated scheme.

1.1. Related Works. Authentication protocols play an important role in preventing any unauthorized access from an adversary or malicious user for net-based services. Most of the traditional authentication protocols are based on public key cryptography like RSA. However, RSA cryptosystems heavily consume computation resources and have a lengthy key size making the traditional authentication schemes inefficient in mobile devices that are resource constrained. Elliptic curve cryptography (ECC) [3, 4], compared with the other public key cryptography, such as RSA, provides the same security level in RSA with smaller keys and faster computation; e.g., a 160-bit ECC based public key can provide the security level of a 1024-bit RSA based public key and a 256-bit ECC based public key has the same security level as a 3072-bit RSA public key [5]. Therefore, the authentication schemes based on ECC are more beneficial for mobile devices than other cryptosystems.

To access the resource at the remote server, the most convenient and simplest mechanism is the password-only authentication schemes [6–10]. If the user wants to login to the remote server, he must submit his identity and password to the server. Upon receiving the login request, the server checks whether the submitted identity and password are equal to the identity and password stored in the table. If the user’s identity and password match the corresponding pair of the table, the user passes authentication of the remote server and is authorized to access the system. To achieve higher security, the password is salted with a hash function in the login request. In general, these schemes only use the single factor of password to secure the security of the system, which is prone to suffer from online or offline password guessing attack [11, 12].

To overcome this issue and further improve the system security, Das firstly proposed a two-factor authentication scheme in 2009 [13], i.e., using password and smartcard, which provides greater flexibility for authentication and inspired many subsequent relevant works [14–17]. Das claimed that his scheme has the advantage of employing simple hash function, requiring less communication cost, and is secure against known attacks. Unfortunately, many researchers [14–16, 18, 19] examined Das’s scheme and identified several security weaknesses (such as insider attack, impersonation attack, and offline password guessing attack) and then put forward many improved versions.

In 2014, Islam-Biswas [20] put forward a two-factor authentication scheme on ECC for cloud computing, and claimed that their protocol is not only efficient but also secure enough to fulfill the security requirements of many authentication scenarios. However, Sarvabhaatla-Vorugunti [21] found that Islam-Biswas’s scheme [20] fails to resist replay attack and is defenseless to impersonation attack, and then they proposed an enhanced two-factor authenticated scheme to thwart the security weakness. However, extensive use of scalar multiplication made their work inefficient. Qu-Tan [22] presented a new two-factor user authentication and key agreement scheme on ECC to overcome some security weaknesses such as smartcard loss attack in the previous schemes. However, Huang et al. [23] analyzed Qu-Tan’s scheme [22] and pointed out that their scheme was unable to withstand impersonation attack, and a new enhanced key agreement for authentication was introduced by Huang et al. to mitigate the chances of security weakness. Unfortunately, Chaudhry et al. [24] found that Huang et al.’s scheme [23] was subjected to impersonation attack and has correctness issues. They introduced an improved two-factor authentication scheme over Huang et al.’s protocol [25] and claimed their scheme can resolve all the correctness issues in the previous one. However, we found that Chaudhry et al.’s scheme [24] is vulnerable to smartcard loss attack.

Independently, Farash-Attari [26] proposed an authenticated protocol to protect data transmission on ECC for mobile client-server networks. Chaudhry et al. [27] proposed an improved smartcard based authenticated protocol for telecare medical information. Xie et al. [28] extended the security model of authentication and presented a dynamic ID-based two-factor authenticated scheme to achieve user anonymity and overcome smartcard loss attack. Lu et al. [29] proposed an anonymous two-factor authentication scheme to eliminate the security weaknesses in the previous schemes for session initiation. Chang et al. [25] proposed an enhanced scheme for IoT and cloud server to fix the security issue of inability to provide mutual authentication and the mistiness of the session key, and retains the merits of the previous one. Kumari et al. [30] also proposed an improved authenticated
scheme using ECC for IoT and cloud server and claimed their proposal is resistant to known attacks. The common feature of these schemes is that they support two-factor authentication and make use of ECC to enhance security. Unfortunately, most of these two-factor authentication schemes and the similar kinds were pointed out that they cannot achieve truly two-factor security since they are vulnerable to smartcard lost attack.

In recent years, there are some other ECC based authentication protocols that were proposed for mobile devices [31–35]. Yet, there is a common issue in these schemes; that is, the authentication process between mobile devices and the remote server must be done with the help of the third party, which makes their communication overhead substantially higher.

In summary, according to the analysis above, most of the existing authentication schemes ultimately turn out to have defects as follows:

1. High computation cost and high communication overhead result in the impracticality of their scheme.
2. Not being able to preserve the user privacy leads to the tracking of sensitive information such as identity and location by the adversary.
3. The security properties of their schemes are evaluated by using their own evaluation criteria, rather than the well-known third-party evaluation criteria.

1.2. Our Contributions. Considering the comments above, a desirable remote authentication scheme for mobile cloud computing services should ensure efficiency while providing appropriate security. In this paper, we present a secure and efficient anonymous two-factor authentication and key agreement scheme for MCC by employing ID-based ECC with pairing-free. The contributions of the proposed scheme are summarized as follows:

1. Privacy-preserving. Preserving user anonymity and providing untraceability are the strong demand of the mobile client, and our protocol fulfills these security requirements.

2. Not requiring the additional third party. In our scheme, the participants, except for the mobile client and the cloud server, and the authentication process do not involve the trusted third party like the home agent.

3. Strong security and efficiency. The proposed scheme employs “fuzzy verifier” technique to resist offline dictionary attack and fulfills the security evaluation metrics; meanwhile, the performance comparison with the related two-factor schemes shows that our scheme has a better tradeoff between the security requirements and the performance.

1.3. Security Evaluation Criteria. In order to evaluate the security properties of our scheme more fairly, we will adopt the widely accepted evaluation criteria as the third-party security evaluation criteria. We brief the security evaluation criteria as follows.

1. (1) C1: No password verifier-table. The server should not maintain a table to store the password of user. (2) C2: Password friendly. The scheme should provide a mechanism for the user to change the password locally. (3) C3: No password exposure. The privileged insider cannot derive the user password. (4) C4: No smart card loss attack. If the user’s smart card is lost or stolen and obtained by the attacker, the attacker cannot reveal the identity and password of the user. (5) C5: Resistance to known attacks. The scheme should be secure against basic/sophisticated attacks, such as offline password guessing attack, impersonation attack, and replay attack. (6) C6: Sound repairability. The scheme should provide a smartcard revocation mechanism. (7) C7: Provision of key agreement. The client and the server should generate a shared session key between them. (8) C8: No clock synchronization. The scheme should be prevented from clock synchronization and time-delay problem. (9) C9: Timely typo detection. The scheme can detect the wrong password of the user. (10) C10: Mutual authentication. The client and the server should authenticate each other. (11) C11: User anonymity. The scheme should prevent the identity of the user from being known or tracked by the attacker. (12) C12: Forward secrecy. The scheme should provide the perfect forward secrecy.

1.4. Organization of This Paper. The rest of the paper is organized as follows. Some preliminaries are given in Section 2. Section 3 presents our two-factor authentication scheme for MCC and the security analysis of the proposed scheme is given in Section 4. The performance comparisons are discussed in Section 5. We concluded this paper in Section 6.

2. Preliminaries

2.1. Notations. Some notations used in this paper are introduced as follows:

- $MC_i$: the ith mobile client;
- $CS$: the cloud server;
- $ID_{C_i}$: $MC_i$’s identity;
- $PW_{i}$: $MC_i$’s password;
- $ID_{S}$: the $CS$’s identity;
- $p, q$: two large prime numbers;
- $F_p$: a finite field with $p$;
- $E_{q}^*$: an elliptic curve defined on finite field $F_p$ with order $q$;
- $G$: a cyclic additive group with order $q$;
- $P$: the generator of $G$;
- $Z_q^*$: $\{1, 2, \ldots, q-1\}$;
- $s$: the private key of $CS$;
- $K_{pub}$: the system public key;
- $SCN_i$: the smartcard number;
- $h\cdot: \{0, 1\}^* \rightarrow \{0, 1\}^k$, the collision-free one-way hash function.
2.4. Adversary Model. Understanding the adversary capabilities is extremely important for designing a truly secure "point at infinity." Let the security of our proposed protocol: 

Elliptic Curve Discrete Logarithm (ECDL) Problem: Given \( Q, \) \( P \in G \), finding an integer \( a \in \mathbb{Z}^*_q \) such that \( Q = aP \in G \) is hard.

Computational Diffie-Hellman (CDH) Problem: Given \( (P, aP, bP) \) for any \( a, b \in \mathbb{Z}^*_q \), finding \( abP \in G \) is hard.

Elliptic Curve Factorization (ECF) Problem: Given \( (P, Q) \in G \), where \( Q = rP + tP \) and \( r, t \in \mathbb{Z}^*_q \) and computation of \( rP \) and \( tP \) is impossible.

2.5. Adversary Model. Understanding the adversary capabilities is extremely important for designing a truly secure protocol. In this section, we conclude the adversary model used in this paper based on [35] as follows:

1. An attacker may control the insecure channel between the related parties. That is to say, the attacker can intercept, eavesdrop, replay, modify, delete, or insert messages over the public channel.

2. An attacker can extract the secret data stored in the smartcard by side-channel attack [36, 37] or differential power attack [38].

3. An attacker can learn the identity of the user as far as the attacks and security properties are concerned.

4. An attacker can enumerate offline all the pairs in Cartesian product \( D_{ID} \times D_{PW} \) in polynomial time, where \( D_{ID} \) and \( D_{PW} \) denote identity space and password space, respectively.

5. An attacker cannot successfully guess the random number and the secret key chosen by the communication parties within polynomial time, since they are adequately large.

6. An attacker can learn the public parameter of system like \( E_q(a, b), P, K_{pub} \).

3. Proposed Scheme

In this section, we shall describe the details of our anonymous two-factor user authentication scheme for MCC. The proposed scheme consists of three phases: system setup, registration, and authentication.

3.1. System Setup. The purpose of this phase is to generate the initial parameters for the future user registration and authentication. The working process is as follows and the notations are as defined above:

1. Choose an elliptic curve \( E_q \) over a prime field \( F_p \).

2. Select the master key \( s \in \mathbb{Z}^*_q \) and set \( K_{pub}=sP \) as the public key.

3. Publish system parameters: \( \{F_p, E_q, F_p, p, P, K_{pub}, G, h()\} \).

4. Select an integer \( m \in [2^5, 2^8] \) as the parameter of fuzzy verifier.

3.2. Registration. In this phase, \( MC_i \) with identity \( ID_i \) wants to register to the cloud server \( CS \) and \( CS \) generates registration information and delivers them to \( MC_i \). The messages to be exchanged in this phase are illustrated as follows:

1. \( MC_i \rightarrow CS: \{ID_i, RPW_i\} \), where \( RPW_i=h(r_i||PW_i) \) (\( r_i \) is a random number).

2. \( CS \rightarrow MC_i: \{ID_i, ID_P, h(), P, K_{pub}, m\} \), where \( D_i = h(h(ID_i)||b_i||T_{i}||SCN_i) \) mod \( m \rangle \in RPW_i \) (\( T_i \) is \( MC_i \)'s registration time, \( b_i \) is a random number). Furthermore, \( CS \) stores \( \{ID_i, T_i, b_i, SCN_i\} \) into a table.

3. \( MC_i \) computes \( D_2 = r_i\times h(ID_i)||RPW_i \) mod \( m \) and stores \( D_2 \) into the smartcard.

The detail of this phase is shown in Figure 2.

3.3. Authentication. In this phase, mutual authentication between \( MC_i \) and \( CS \) shall be accomplished. Meanwhile, the session key shared between them is generated. \( MC_i \) and \( CS \) perform the following steps:

1. \( MC_i \rightarrow CS: \{PID_i, X_3, M_2\} \). \( MC_i \) keys his/her \( ID_i \) and \( PW_i \), the smartcard computes \( RPW_i^* = h(r_i||PW_i) \). \( D_2^* = r_i\times h(h(ID_i)||RPW_i^*) \) mod \( m \). If \( D_2^* = D_2 \), the card accepts \( MC_i \); selects a random number \( r_m \in \mathbb{Z}^*_q \), and computes \( X_3 = r_mX_2 = r_mK_{pub}, X_3 = X_1 + X_2, M_1 = D_i\times RPW_i, PID_i=h(ID_i)||M_1 || h(X_1||X_2) \) and \( M_2 = h(ID_i||X_3||PID_i) \). Finally, \( MC_i \) sends \( \{PID_i, X_3, M_2\} \) as a login request to \( CS \) via a public channel. Otherwise, it aborts this session.

2. \( CS \rightarrow MC_i: \{Y, M_3\} \). \( CS \) first computes \( X_2^* = X_1, X_3' = X_1 + X_2 \), \( ID_i||M_3' \) and \( h(ID_i)||b_i||T_{i}||SCN_i \) mod \( m \), and verifies the condition \( M_1 = M_2 \). If it is false, \( CS \) aborts this session. Otherwise, \( CS \) selects a random number \( r_s \in \mathbb{Z}^*_q \) and computes \( Y = r_sP, K_{s} = r_sX_1, SK_{s-m} = h(ID_i)||M_3' || Y || K_{s}, M_3 = h(ID_i)||X_3'|| X_3'|| Y || K_{s} \) and sends \( \{Y, M_3\} \) to \( MC_i \).

3. \( MC_i \rightarrow CS: \{M_4\} \). \( MC_i \) computes \( K_{m} = r_s||Y, M_3' = h(ID_i)||X_3'|| Y || K_{m} \), and \( MC_i \) will abort this
session if $M_3' \neq M_3$; otherwise, $MC_i$ computes $SK_m = h(ID_i||ID_3||X_2||Y||K_m)$ and $M_4 = h(ID_i||ID_3||X_2||K_m||SK_m)$ and forwards $[M_4]$ to CS.

(4) CS computes $M_4' = h(ID_i||ID_3||X_2||Y||K_m)$, and if $M_4' \neq M_4$, rejects computes $SK_m' = h(ID_i||ID_3||X_2||Y||K_m)$, $M_4' = h(ID_i||ID_3||X_2||K_m||SK_m)$ and forwards $[M_4']$. Session key $SK_m = SK_m'$, and if $M_4' \neq M_4$, rejects.

This authentication phase is summarized in Figure 3.

3.4. Password Update. When the password of $MC_i$ is leaked out, our proposed scheme can change the password flexibly. $MC_i$ performs the following steps to change the password:

1. $MC_i$ inputs $ID_i$, $PW_i$, selects $r_i$, computes $RPW_i = h(r_i||PW_i)$, computes $D_2 = r_i \oplus (h(ID_i||RPW_i) \mod m)$, and forwards $D_2$ to $CS$.

2. $CS$ selects $b_i$, $T_i$, $SCN_i$, computes $D_2' = h(ID_i||b_i||T_i||SCN_i)$, stores $[ID_i, T_i, b_i, SCN_i]$ into a table, stores $[D_2, D_2', h(i, P, SCN_i)]$ into smartcard.

3. $MC_i$ inserts smartcard, computes $RPW_i = h(r_i||PW_i)$, if $D_2' \neq D_2$, rejects.

4. Selects $r_s$ and computes $X_s = r_s ; X_0$, $M_3 = D_2 \oplus RPW_i$, $PID = h(ID_i||M_3) \oplus b_i||X_s||X_0$, $M_4 = h(ID_i||X_0||PID)$, and forwards $\{M_4\}$ to $CS$.

5. $CS$ computes $X_s' = X_0 + X_s$, $M_4' = h(ID_i||X_s||PID)$, and if $M_4' \neq M_4$, rejects.

6. Finds $(T_i, b_i, SCN_i)$ via $ID_i'$, and computes $M_1 = h(ID_i||b_i||T_i||SCN_i)$, stores $[D_1, D_2', h(i, P, SCN_i)]$ into smartcard.

7. If $M_4' \neq M_4$, rejects.

8. Selects $r_i$ and computes $Y = r_i \oplus (h(ID_i||RPW_i) \mod m)$, $K_0 = r_i X_0$, $SK_m = h(ID_i||ID_3||X_2||Y||K_i)$, $M_4 = h(ID_i||ID_3||X_2||Y||K_i)$, and forwards $[PID, X_1, M_2]$ to $CS$.

9. Upon receipt of revival request from $MC_i$, $CS$ first validates the legitimacy of $MC_i$. If it is true, $CS$ sets $T_i, b_i, SCN_i$ as null. Thus, the card is revoked so that the card can no longer be used to login to the system unless $MC_i$ registers again. Otherwise, $CS$ rejects this revocation request.

3.5. Smartcard Revocation. If $MC_i$'s smartcard is breached, to protect the card from being abused, $MC_i$ can revoke the card as follows:

1. $MC_i$ performs step (1) in Section 3.3 to get authenticated by the card.

2. $MC_i \rightarrow CS: [PID, X_1, M_2, revoke_request]$. As shown in Section 3.3, the card computes $PID_i, X_2, M_2$, and sends $[PID_i, X_2, M_2, revoke_request]$ to $CS$.

3. Upon receipt of revocation request from $MC_i$, $CS$ first validates the legitimacy of $MC_i$. If it is true, $CS$ sets $T_i, b_i, SCN_i$ as null. Thus, the card is revoked so that the card can no longer be used to login to the system unless $MC_i$ registers again. Otherwise, $CS$ rejects this revocation request.
4. Security Analysis

In this section, we provide an informal security analysis of the proposed scheme on satisfying the security evaluation criteria of two-factor authenticated protocol, and a formal security analysis to demonstrate that our scheme is secure under random oracle model [39].

4.1. Informal Security Analysis

4.1.1. User Anonymity and Privacy. Privacy is of great importance in the area of mobile cloud computing [40–42]. It means that the attacker cannot determine the sender of the messages and also cannot distinguish whether the messages are sent by the same sender. In our scheme, user’s ID is hidden in PID, which is different with \( h(X) \) because \( X \) is changed with \( r_m \) in every session. To retrieve \( ID \), the adversary has to compute \( X \). However, \( h(X) \) is also infeasible for him/her to find a correct pair \((ID, PW_i)\) pair. Therefore, the number of \((ID, PW_i)\) candidates is too large for the adversary to conduct the offline dictionary attack successfully.

If the adversary uses \( M_2 \) and guesses \( ID_2 \) from \( \{ID_2, h(ID_2)\} \) to CS for registration. The password \( PW_i \) is protected with a random number \( r_i \) and thus CS cannot learn \( MC_i \)’s \( PW_i \) and other useful information. Therefore, the proposed scheme is secure from privileged insider attack.

4.1.6. Replay Attack. In our scheme, we make use of the random number mechanism to resist replay attack. In each session, the random number \( r_m \) is generated by \( MC_i \) to compute the login request messages \( \{PID_a, X_1, M_1\} \) and the random number \( r_i \) is chosen by CS to compute the response messages \( \{Y, M_3\} \). The freshness and validity of the messages are assured effectively by the random number mechanism for the current session. Therefore, the proposed scheme can withstand replay attack.

4.1.7. Verifier-Stolen Attack. In our scheme, the verifier table \( \{ID_2, b_1, T_i, SCN_i\} \) stored in CS and these parameters are not security-related. The adversary cannot conduct any attack if he/she compromises this table. Therefore, the proposed scheme can resist verifier-stolen attack.

4.1.8. User Impersonation Attack. If the adversary intends to impersonate \( MC_i \), he will fail since he/she cannot guess the pair \((ID_2, PW_i)\) or replay the login request \( \{PID_a, X_1, M_1\} \) successfully as we analyzed above. Furthermore, if he/she chooses a random number \( r_i \) and computes \( X_{ta} = r_i P \), forges \( PID_a \) and \( M_4 \), constructs the login request message \( \{PID_a, X_{ta}, M_4\} \) and sends it to CS. However, CS cannot compute the correct \( ID_2 \) in the table according to the login request \( \{PID_a, X_{ta}, M_4\} \) from the adversary, which results in the computed \( M_{ta} \)’s not being equal to the received \( M_4 \). This means that the adversary fails to impersonate \( MC_i \). Therefore, the proposed scheme can withstand user impersonation attack.

4.2. Formal Security Analysis. In this section, we use the random oracle model [39] to conduct a formal security analysis of the proposed scheme. For simplification, we adopt

\( r_i \) is large enough to prevent the adversary from guessing \( r_i \) successfully according to item (5) of the adversary model in Section 2.4, which results in failure of guessing \( ID \) and \( PW_i \) successfully. Second, suppose the adversary knows \( r_i \); it is also infeasible for him/her to find a correct pair \((ID, PW_i)\) because the computation of \( D_2 \) employs “fuzzy verifier” mechanism. For example, supposing \( |D_{ID}| = |D_{PW}| = 10^6 \) and \( m = 2^8 \), there are \( |D_{ID}| \cdot |D_{PW}| = 2^{32} \) candidates of \((ID, PW_i)\) pair. Therefore, the number of \((ID, PW_i)\) candidates is too large for the adversary to conduct the offline dictionary attack successfully.

In short, the proposed scheme is secure from dictionary attack.
the security model of [43] as our security model. We will provide a security proof and a privacy proof of our scheme, and they are similar to [43]. But there are two differences, one is because their authentication schemes are based on modular exponentiation, their security analyses are also based on the modular exponentiation, and our security analysis is based on ECC; the second is that our analysis result of the various games is just a rough estimate.

**Theorem 1.** Assume that \( \mathcal{P} \) represents the proposed scheme for mobile cloud computing, \( \mathcal{D} \) is a password space and its frequency distribution follows the Zipf’s law, \( \mathcal{A} \) is a probabilistic polynomial-time (PPT) adversary, and he/she makes maximum \( q_{\text{send}} \) queries of Send oracle with execution time \( t \).

\[
\text{Adv}^{\text{AKE}}_{\mathcal{P},2} (\mathcal{A}) \leq C' \cdot q_{\text{send}} + \varepsilon (l)
\]

(1)

where \( C' \) and \( s' \) are the Zipf parameters, \( l \) is the security parameter, and \( \varepsilon () \) is a negligible function.

**Proof.** We prove this theorem with a series of games \( G_m \) where \( C' \) and \( s' \) are the Zipf parameters, \( l \) is the security parameter, and \( \varepsilon () \) is a negligible function.

\[
\text{Ad}^{\text{AKE}}_{\mathcal{P},2} (\mathcal{A}) = \Pr [\mathcal{S}_0]
\]

(2)

\( \text{Gm}_1: \) This game simulates the hash function \( h(\cdot) \) by maintaining a hash list \( L_h \) with respect to our scheme \( \mathcal{P} \). We also simulate Send, Test, Execute, Reveal, and Corrupt queries as the real player’s behavior. We can see that the hash function can be modeled in PPT time and this game is indistinguishable from \( \text{Gm}_0 \). Thus, we have

\[
|\Pr [S_1] - \Pr [S_0]| \leq \varepsilon (l)
\]

(3)

\( \text{Gm}_2: \) In this game, we rule out sessions in which the collisions of random oracle queries occur during the simulation of hash function and transcripts \( \{\text{PID}_i, X_i, M_{2,i}, M_{3,i}, \text{and } M_{4,i}\} \).

If the collisions occur, we abort the game and let the adversary win. According to the birthday paradox, we have

\[
|\Pr [S_2] - \Pr [S_1]| \leq \varepsilon (l)
\]

(4)

\( \text{Gm}_3: \) In this game, we modify the simulation rules of session through Execute queries. We use the private hash function \( h'(\cdot) \) instead of \( h(\cdot) \) to calculate the session key in passive session. Furthermore, when computing the session key and the authenticator \( M_{1,i} \), the Diffie-Hellman key \( K_s = r_s X_s \) and \( K_n = r_n Y \) are removed from the input list, i.e., the session key \( SK = h(ID || ID || X_s || Y) \) and authenticator \( M_{3}= h(ID || ID || X_s || X_s || Y) \). In \( \text{Gm}_2 \), we have ruled out the collisions of hash function and the transcripts. Thus, the adversary is capable of distinguishing \( \text{Gm}_3 \) and \( \text{Gm}_2 \) only if he/she can calculate the Diffie-Hellman key \( K_s \) or \( K_n \) in passive session and sends a query \( \{ID, ID, X_s, Y, K_s\} \) to \( h(\cdot) \). However, breaking the CDH problem is computationally hard. To a CDH instance \( (X, Y) \), we use the self-reducibility [44] of CDH problem to embed this instance to the passive sessions. To do that, we select random numbers \( a_1, a_2, b_1, b_2 \) and \( b_1 \in Z^*_q \) for each session and set \( U = a_1 X + b_1 P \) and \( V = a_2 Y + b_2 P \). If the adversary is able to distinguish the game \( \text{Gm}_3 \) and \( \text{Gm}_2 \), a query \( \{ID, ID, X_s, Y, K_s\} \) is made to hash oracle. This means that the adversary can compute \( (K - a_1 b_1 X, a_1 b_1 Y, b_2 P)/a_1 a_2 \) as an answer to the CDH instance \( (X, Y) \).

Under the difficulty of CDH problem, we have

\[
|\Pr [S_3] - \Pr [S_2]| \leq \varepsilon (l)
\]

(5)

\( \text{Gm}_4: \) In this game, we start to handle the active session for Send \( (\text{CS}, \{M_{4,i}\}) \) query. And we define the game with the following rule, where the adversary may have computed the correct \( K_m \) to impersonate the mobile client \( MC \). The role of the participants process queries is modified as follows.

Compute \( M_{4,i} = h(ID || ID || X_s || SK_{m,i}) \) and check whether \( M_{4,i} \) is equal to the received \( M_{4,i} \). If it is true, the cloud server \( CS \) looks up a record \( ((\text{PID}, X_s, M_{4,i}), (Y, M_{4,i}), (M_{4,i})) \) from the hash list \( L_h \). We terminate the game if the record exists. The authenticator \( M_{4,i} \) in the proposed scheme is unforgeable due to the hardness of CDH problem. Thus, we have

\[
|\Pr [S_4] - \Pr [S_3]| \leq \varepsilon (l)
\]

(6)

\( \text{Gm}_5: \) In this game, we continue to the active session for Send \( (\text{MC}, \{Y, M_{5,i}\}) \). We also define this game by terminating the game with the following rule, where the adversary is lucky to guess \( K_s \) to impersonate the cloud server \( CS \) without asking the hash query \( h(\cdot) \). To achieve this goal, the rule of the participants process the queries is modified as follows.

Look up a record \( ((\text{PID}, X_s, M_{5,i}), (Y, M_{5,i}), (M_{5,i})) \) in the hash list \( L_h \), and we terminate the game if the result is null. Otherwise, compute the session key \( SK = h(ID || ID || X_s || Y || K_s) \) and authenticator \( M_{5} = h(ID || ID || X_s || Y || K_s) \).

The adversary wins only if \( K_s \) is correctly guessed without asking \( h(\cdot) \). Similar to the previous game, we obtain

\[
|\Pr [S_5] - \Pr [S_4]| \leq \varepsilon (l)
\]

(7)
forgery of message \{PID_{i}, X_{i}, M_{2}\} is negligible. Thus, we obtain
\[
|Pr [S_{6}] - Pr [S_{5}]| \leq \varepsilon (l)
\]  
(8)

In the last game, the session keys are chosen randomly and the advantage of \( A \) in guessing session keys is negligible and the active sessions are aborted without accepting if \( A \) forges the message. The only possibility for \( A \) to win the game is to corrupt the smartcard and guess the password of MC\(_{i}\). The advantage \( A \) has no advantage to get the password from the game. Based on the Zipf’s law, we obtain
\[
|Pr [S_{6}] \leq C' \cdot q_{\text{send}}
\]  
(9)

According to (2)–(9), we have the result of Theorem 1.

**Theorem 2.** Assume that \( \mathcal{P} \) represents the proposed scheme and \( A \) is a PPT adversary breaking the anonymity of \( \mathcal{P} \). The advantage of \( A \) in breaking the anonymity of \( \mathcal{P} \) is bounded by
\[
\text{Adv}_{\mathcal{P}}^{\text{anon}} (A) \leq \varepsilon (l)
\]  
(10)

**Proof.** We suppose that \( A \) can break the anonymity of \( \mathcal{P} \) with a nonnegligible advantage. We reach this aim by employing \( A \) to develop an algorithm to break the CDH problem with the identical nonnegligible advantage.

**Algorithm 3.** Select \( r_{m}, r \in Z_{q}^{*} \), input two tuples \((P, r_{m}P, sP), (P, r_{m}P, sP, r)\), where \( s \) is the private key of \( \text{CS} \).

1. Let \( U \) be a valid user owning his smartcard and password.
2. Let \( X_{U1}=r_{m}P, X_{U2}=r_{m}sP \), and execute the subsequent procedure with \( \text{CS} \) as the protocol definition. We use \( sid_{U} \) as the session identifier of this protocol execution.
3. Let \( X_{U1}=r_{m}P, X_{U2}=r \), and execute the subsequent procedure with \( \text{CS} \) as the protocol definition. The corresponding session identifier of this protocol execution is labelled as \( sid_{U}^{k} \).
4. Select \( r_{m} \in Z_{q}^{*} \), let \( X_{U1}=r_{m}P \) and \( X_{U2}=r_{m}K_{pub}=r_{m}sP \), and execute the subsequent procedure with the server \( \text{CS} \) using \( X_{U1}, X_{U2} \). In this case, the session identifier is denoted as \( sid_{U}^{k} \).
5. Two queries TestAnonymity\((sid_{U}^{l}, sid_{U}^{k})\) and TestAnonymity\((sid_{U}^{l}, sid_{U}^{k})\) are made by \( A \), and the returned bits are denoted as \( b_{l} \) and \( b_{k} \), respectively.
6. If \( b_{l}=0 \) and \( b_{k}=0 \), output "none is a Diffie-Hellman tuple"; if \( b_{l}=0 \) and \( b_{k}=1 \), output "\((P, r_{m}P, sP, r_{m}sP)\) is a Diffie-Hellman tuple"; if \( b_{l}=1 \) and \( b_{k}=0 \), output "\((P, r_{m}P, sP, r)\) is a Diffie-Hellman tuple"; if \( b_{l}=1 \) and \( b_{k}=1 \), output "both are Diffie-Hellman tuples."

Obviously, the Algorithm 3 can be performed within polynomial time. Furthermore, in Algorithm 3, \( sP \) is fixed while \( r_{m}P \) is different in every protocol run. Based on the self-reducibility of CDH problem, we obtain \( \text{Adv}_{\mathcal{P}}^{\text{anon}} (U) = |Pr[\text{U}(P, r_{m}P, sP, r_{m}sP)=1] - Pr[\text{U}(P, r_{m}P, sP, r)=1]| \), where \( s \) is a fixed value and \( r_{m} \in Z_{q}^{*} \). Thus, we have \( \text{Adv}_{\mathcal{P}}^{\text{anon}} (U) \leq |Pr[\text{TestAnonymity}(sid_{U}^{l}, sid_{U}^{k})=1] - Pr[\text{TestAnonymity}(sid_{U}^{l}, sid_{U}^{k})=1]| \). That is to say, \( A \) can break the untraceability of a participant \( U \) by solving the CDH problem which is believed computationally hard. This is a contradiction with the difficulty of CDH problem.

Therefore, the theorem is proved.

5. Comparison on Efficiency and Security
In this Section, we compare our protocol with other related competitive protocols such as Qu-Tan[22], Farash-Attari[26], Chaudhry et al.[27], Xie et al. [28], Chaudhry et al. [24], Lu et al. [29], Chang et al. [25], and Kumari et al. [30] in terms of computation cost and communication overhead and security during the authentication phase. The registration is a one-time process, so we have not taken it into consideration.

Here we set \( q \) and \( p \) as the order of the super singular curve or nonsupersingular curve \( E \) over a finite field \( F_{p} \) which is 512 bits and 160 bits, respectively. For the convenience of evaluating computation cost, we set \( T_{PM}, T_{PA}, T_{IN}, T_{H} \) as the time of performing a one-way hash function, the time of performing a scalar multiplication operation of point, the time of performing an addition operation of point, and the time of performing a 160 bits modular inversion, respectively. The time of performing an exclusive-or operation (XOR) and a concatenate operation are much less than a hash function [45], so their times are negligible. Combined with the analysis above, the specific performing time of these operations is shown in Table 1 based on experimental data [46]. Furthermore, we set \( l_{1} \) as the length of identity with 32 bits, \( l_{P} \) as the length of a Point with 1024 bits, \( l_{h} \) as the length of a one-way hash value with 160 bits, and \( l_{t} \) as the length of a timestamp with 32 bits, respectively.

<table>
<thead>
<tr>
<th></th>
<th>( T_{PM} )</th>
<th>( T_{PA} )</th>
<th>( T_{IN} )</th>
<th>( T_{H} )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Server</strong></td>
<td>1.17 ms</td>
<td>&lt;0.1 ms</td>
<td>&lt;1 ms</td>
<td>&lt;0.001 ms</td>
</tr>
<tr>
<td><strong>Client</strong></td>
<td>0.13 s</td>
<td>&lt;0.1 s</td>
<td>&lt;0.01 s</td>
<td>&lt;0.001 s</td>
</tr>
</tbody>
</table>

**5.1. Comparison of Computation Cost.** The comparison of computation cost between the proposed scheme and the related schemes is shown in Table 2.

According to Table 2, we can learn that the computation cost of our scheme in the mobile client is 0.497 s, which is just slightly higher than [28], while it is much less than the others [22, 24–27, 29, 30]. Meanwhile, the computation cost of our scheme in the server side is 3.616 ms, which is almost the same as [24, 28, 29] and is much less than [22, 25–27, 30]. It is evident that our scheme is still efficient as compared with other related schemes, whether in client side or in server side.

To make the comparison more clearly, the comparison graph of computation cost is shown in Figure 4.
Table 2: Comparison in computation cost.

<table>
<thead>
<tr>
<th></th>
<th>[22]</th>
<th>[26]</th>
<th>[27]</th>
<th>[28]</th>
<th>[24]</th>
<th>[29]</th>
<th>[30]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>$7T_{PM}+3T_{PA}+10T_H$</td>
<td>$4T_{PM}+4T_H$</td>
<td>$4T_{PM}+T_{PA}+2T_H$</td>
<td>$3T_{PM}+T_{PA}+6T_H$</td>
<td>$3T_{PM}+2T_{PA}+9T_H$</td>
<td>$5T_{PM}+T_{IN}+9T_H$</td>
<td>$4T_{PM}+5T_H$</td>
<td>$3T_{PM}+T_{PA}+7T_H$</td>
</tr>
<tr>
<td>S2</td>
<td>1.22 s</td>
<td>0.524 s</td>
<td>0.625 s</td>
<td>0.496 s</td>
<td>0.499 s</td>
<td>0.669 s</td>
<td>0.525 s</td>
<td>0.653 s</td>
</tr>
<tr>
<td>S3</td>
<td>$5T_{PM}+2T_{PA}+6T_H$</td>
<td>$5T_{PM}+5T_H$</td>
<td>$4T_{PM}+T_{IN}+5T_H$</td>
<td>$3T_{PM}+T_{PA}+6T_H$</td>
<td>$3T_{PM}+T_H$</td>
<td>$3T_{PM}+T_{IN}+7T_H$</td>
<td>$4T_{PM}+5T_H$</td>
<td>$5T_{PM}+4T_H$</td>
</tr>
<tr>
<td>S4</td>
<td>6.056 ms</td>
<td>5.855 ms</td>
<td>5.685 ms</td>
<td>3.616 ms</td>
<td>3.516 ms</td>
<td>3.517 ms</td>
<td>4.685 ms</td>
<td>5.854 ms</td>
</tr>
</tbody>
</table>

S1: computation cost in mobile client side; S2: executing time in mobile client side; S3: computation cost in cloud server side; S4: executing time in cloud server side.
5.2. Comparison of Communication Overhead. Table 3 compares the communication overhead of the proposed scheme with other related schemes.

From Table 3, we can see that the message size of the proposed scheme is 2688 bits, which manifests that our scheme outperforms the related schemes except for [26, 28]. We can also see that the number of total messages in the authentication phase of schemes participating in comparison can be divided into two classes, the number of which is 2 and 3, respectively. The number of total messages in our scheme is 3. Although schemes [25, 27] can complete their authentication process with 2 messages, these 2-message protocols have the significant security weakness of failing to achieve perfect forward secrecy as pointed out by Krawczyk [47]. In brief, the comparison result demonstrates that the communication overhead of our scheme is acceptable.

The comparison of communication overhead is shown in Figure 5.

5.3. Comparison of Security Properties. Finally, we make a comparison of security properties between our scheme and other related schemes in light of the evaluation metrics, and the result is given in Table 4.

From Table 4, we can see that the proposed scheme can achieve more security properties than the other related schemes, such as user anonymity and untraceability which should not be overlooked in privacy-preserving, and it is more effectively satisfied with the urgent security requirement of mobile users when their sensitive data was transmitted over the wireless network. The other schemes are more and less vulnerable to some security weaknesses, such that schemes in [22, 24, 25, 27, 29] are vulnerable to smartcard loss attack, schemes in [25, 28] fail to provide user anonymity, and schemes in [24, 25, 27, 30] cannot provide forward secrecy. Thus, it is clear that the proposed scheme can provide better protection for the mobile client in MCC.

In summary, from the three comparisons above, we can draw a conclusion that the proposed scheme is not only more powerful and efficient in computation cost and communication overhead but also is more secure in withstanding various known attacks than other related schemes.

6. Conclusion

In this paper, we have proposed a new anonymous two-factor user authentication and key agreement protocol on ECC for mobile cloud computing. The design of the proposed scheme exploits fuzzy verifier technique to prevent offline identity and password dictionary attack. Furthermore, the reasonable use of ECC makes this scheme efficient for mobile devices that are computing capability limited and energy limited with privacy-preserving property. The formal security analysis on random oracle model reveals that the proposed scheme is provably secure under ECDL problem and CDH problem. Furthermore, the comparison of performance and security shows that the proposed scheme is more efficient and secure than the related works. We believe that this proposal is practical for mobile cloud computing.

Data Availability

(1) The [22] data used to support the findings of this study have been deposited in the [ACM] repository ([DOI: 10.1155/2014/423930]). (2) The [25] data used to support the findings of this study have been deposited in the [Springer] repository ([DOI: 10.1007/s11227-014-1170-5]). (3) The [26] data used to support the findings of this study have been deposited in the [Springer] repository ([DOI: 10.1007/s0916-015-0244-0]). (4) The [27] data used to support the findings of this study have been deposited in the [IEEE Xplore] repository ([DOI: 10.1109/TIFS.2017.2659640]). (5) The [24] data used to support the findings of this study have been deposited in the [Springer] repository ([DOI: 10.1007/s10916-015-0244-0]). (6) The [28] data used to support the findings of this study have been deposited in the [Springer] repository ([DOI: 10.1007/s0916-015-0244-0]). (7) The [29] data used to support the findings of this study have been deposited in the [Springer] repository ([DOI: 10.1007/s11227-014-1170-5]). (8) The [20] data used to support the findings of this study have been deposited in the [Springer] repository ([DOI: 10.1007/s11227-014-1170-5]).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.
Table 3: Comparison in communication overhead.

<table>
<thead>
<tr>
<th></th>
<th>[22]</th>
<th>[26]</th>
<th>[27]</th>
<th>[28]</th>
<th>[24]</th>
<th>[29]</th>
<th>[25]</th>
<th>[30]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$3l_p+4l_h$</td>
<td>$l_p+2l_p+3l_h+2l_l$</td>
<td>$3l_p+2l_p+2l_l$</td>
<td>$2l_p+3l_l+2l_l$</td>
<td>$3l_h+3l_l$</td>
<td>$3l_p+4l_h$</td>
<td>$l_p+4l_p+4l_l$</td>
<td>$l_p+3l_p+3l_h$</td>
<td>$2l_p+4l_l$</td>
</tr>
<tr>
<td>2</td>
<td>3712</td>
<td>2624</td>
<td>3456</td>
<td>2592</td>
<td>3552</td>
<td>3712</td>
<td>4800</td>
<td>3584</td>
<td>2688</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

I1: communication cost; I2: the length of communication message; I3: total messages.

Table 4: Comparison of security properties.

<table>
<thead>
<tr>
<th></th>
<th>[22]</th>
<th>[26]</th>
<th>[27]</th>
<th>[28]</th>
<th>[24]</th>
<th>[29]</th>
<th>[25]</th>
<th>[30]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>No password verifier-table</td>
<td>√</td>
<td>N/A</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Password friendly</td>
<td>√</td>
<td>N/A</td>
<td>√</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>No password exposure</td>
<td>√</td>
<td>N/A</td>
<td>√</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>No smart card loss attack</td>
<td>x</td>
<td>N/A</td>
<td>x</td>
<td>√</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Resistance to known attacks</td>
<td>√</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Sound reactivity</td>
<td>x</td>
<td>N/A</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Provision of key agreement</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>No clock synchronization</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Timely typo detection</td>
<td>√</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Mutual authentication</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>User anonymity</td>
<td>√</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
<tr>
<td>Forward secrecy</td>
<td>√</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>√</td>
</tr>
</tbody>
</table>

Acknowledgments

This work was partially sponsored by the National Natural Science Foundation of China (No. 61672007) and Science and Technology Innovation Guidance Project 2017 of Zhaoqing (No. 201704030605).

References


