Hindawi
Wireless Communications and Mobile Computing
Volume 2019, Article ID 4520685, 12 pages
https://doi.org/10.1155/2019/4520685

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

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

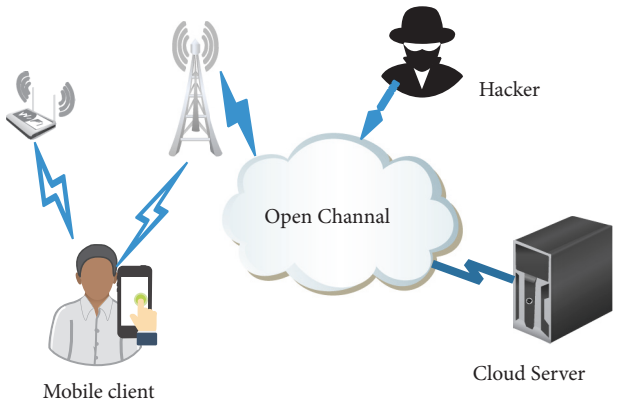

Figure 1: The model of a mobile client communicating with the cloud server.

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, Sarvabhatla-Vorugunti [21] found that Islam-Biswas's scheme [20] fails to resist replay attack and is defenseless to impersonation attack, and they 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) 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 the change 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_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^*: $\{1, 2, \ldots, q\text{-}1\}$;

s: the private key of CS;

K_{pub}: the system public key;

SCN_i: the smartcard number;

$h(\cdot)$: $\{0,1\}^* \longrightarrow \{0,1\}^k$, the collision-free one-way hash function.

2.2. Elliptic Curve Cryptosystem (ECC). Let F_p be the prime field and E_q/F_p denotes an elliptic curve E_q over a finite F_p, defined by an equation $y^2 \bmod p = (x^3 + ax + b) \bmod p$, $a, b \in F_p$ with $(4a^3 + 27b^2) \bmod p \neq 0$. The point on E_q/F_p together with an extra point O is called the point as "point at infinity." The additive elliptic curve group is defined as $G=\{(x, y): x, y \in F_p$ and $(x, y) \in E_q(a, b)\} \in \{O\}$ and we call the point O "point at infinity." Let $P,Q \in G$, l be the line containing P and Q (tangent line to E_q/F_p if $P=Q$) and the third point R intersecting l with E_q/F_p. Let l' be the line connecting R and O. Then P '+' Q is the point such that l' intersects E_q/F_p at R and O and P '+' Q. The scalar multiplication on E_q/F_p can be computed as $kP=P+P+\ldots+P$ (k times).

More details of the ECC definition can be found in [3].

2.3. Computational Problem. We review the following mathematical problems on elliptic curves in order to prove the security of our proposed protocol:

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

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

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

2.4. 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} x 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 $s \in Z_q^*$ 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 $m \in [2^4, 2^8]$ 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) $MC_i \longrightarrow CS$: $\{ID_i, RPW_i\}$, where $RPW_i=h(r_i||PW_i)$ (r_i is a random number).

(2) $CS \longrightarrow MC_i$: a smartcard containing $\{D_1, ID_s, h(\cdot), P, K_{pub}, m\}$, where $D_1= h(h(ID_i||b_i||T_i||SCN_i) \bmod m) \oplus RPW_i$ (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 $D_2 = r_i \oplus h(ID_i||RPW_i) \bmod m$ 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) $MC_i \longrightarrow CS$: $\{PID_i, X_1, M_2\}$. MC_i keys his/her ID_i and PW_i, the smartcard computes $RPW_i^*=h(r_i||PW_i)$, $D_2^* = r_i \oplus (h(ID_i||RPW_i) \bmod m)$. If $D_2^* = D_2$, the card accepts MC_i, selects a random number $r_m \in Z_q^*$, and computes $X_1=r_mP, X_2=r_mK_{pub}$, $X_3 = X_1 + X_2$, $M_1 = D_1 \oplus RPW_i$, $PID_i=(ID_i||M_1) \oplus h(X_1||X_3)$, and $M_2=h(ID_i||X_2||PID_i)$. Finally, MC_i sends $\{PID_i, X_1, M_2\}$ as a login request to CS via a public channel. Otherwise, it aborts this session.

(2) $CS \longrightarrow MC_i$: $\{Y, M_3\}$. CS first computes $X_2{}' = sX_1$, $X_3{}' = X_1 + X_2{}'$, $(ID_i'||M_1{}') = PID_i \oplus h(X_1||X_3)$, and $M_2{}' = h(ID_i'||X_2'||PID_i)$ and then checks whether $M_2{}' = M_2$ holds or not. If not, CS terminates this session. Otherwise, MC_i is authenticated, and CS finds (T_i, b_i, SCN_i) via ID_i', computes $M_1=h(ID_i'||b_i||T_i||SCN_i) \bmod m$, and verifies the condition $M_1{}' \stackrel{?}{=} M_1$. If it is false, CS aborts this session. Otherwise, CS selects a random number $r_s \in Z_q^*$ and computes $Y=r_sP$, $K_s= r_sX_1$, $SK_{s\text{-}m} = h(ID_i'||ID_S||X_2{}'||Y\ ||K_s)$, and $M_3= h(ID_i'||ID_S||X_1{}'||X_2{}'||Y\ ||K_s)$ and sends $\{Y, M_3\}$ to MC_i.

(3) $MC_i \longrightarrow CS$: $\{M_4\}$. MC_i computes $K_m=r_mY$, $M_3{}'= h(ID_i||ID_S||\ X_1||X_2||Y||K_m)$, and MC_i will abort this

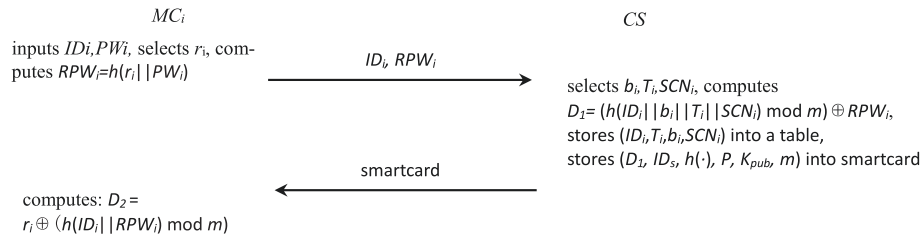

Figure 2: Registration phase.

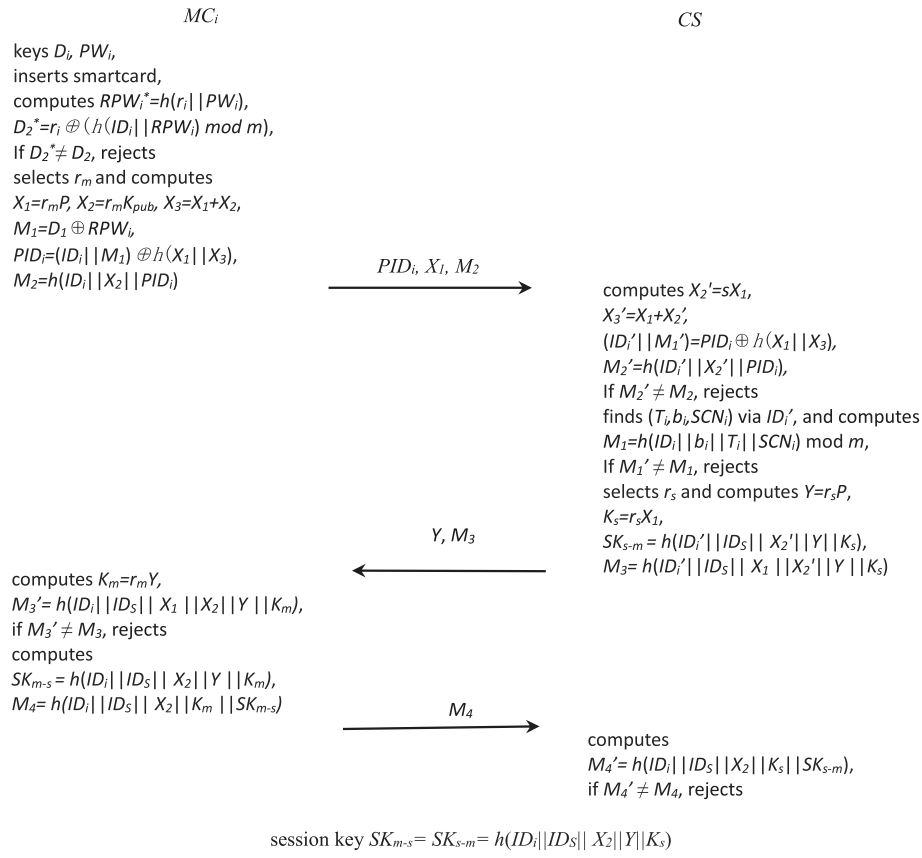

Figure 3: Authentication phase.

session if $M_3' \neq M_3$; otherwise, MC_i computes $SK_{m-s}=h(ID_i||ID_S||X_2||Y||K_m)$ and $M_4=h(ID_i||ID_S||X_2||K_m||SK_{m-s})$ and forwards $\{M_4\}$ to *CS*.

(4) *CS* computes $M_4'=h(ID_i||ID_S||X_2||K_s||SK_{s-m})$, and it exits the session if $M_4' \neq M_4$. Otherwise, it accepts $SK_{s-m}(=SK_{m-s})$ as the shared session key with MC_i.

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 inserts the smartcard and keys ID_i, PW_i.

(2) The card computes $RPW_i'=h(r_i||PW_i)$, $D_2'=r_i \oplus (h(ID_i||RPW_i) \bmod m)$ and checks whether $D_2'=D_2$ holds. If not, the card rejects MC_i's request. Otherwise, the card asks the user to input a new password PW_i^{new}.

(3) The card computes $RPW_i^{new}=h(r_i||PW_i^{new})$, $D_1^{new}=D_1 \oplus RPW_i' \oplus RPW_i^{new}$, and $D_2^{new}=r_i \oplus (h(ID_i||RPW_i^{new}) \bmod m)$ and replaces (D_1, D_2) with (D_1^{new}, D_2^{new}).

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) $MC_i \longrightarrow CS$: $\{PID_i, X_1, M_2, revoke\_request\}$. As shown in Section 3.3, the card computes PID_i, X_1, and M_2 and sends $\{PID_i, X_1, M_2, revoke\_request\}$ 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, and 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_i is hidden in PID_i, which is different with $h(X_1||X_3)$ because X_3 is changed with r_m in every session. To retrieve ID_i, the adversary has to compute X_3. However, he/she will fail because he/she has no knowledge of r_m and s. Thus, the adversary cannot get the MC_i's identity by computing $(ID_i||M_1)=PID_i \oplus h(X_1||X_3)$. Therefore, the proposed scheme achieves not only user anonymity but also untraceability.

4.1.2. Forward Secrecy. In our scheme, the session key $SK=h(ID_i||ID_S||X_2||Y||K_s)$, where $X_2=sX_1$, $Y=r_sP$, and $K_s=r_sX_1=r_sr_mP$. That is to say, the session key is generated with partial key information provided by MC_i and CS respectively and dealt with a hash function. Although the adversary can intercept X_1 and Y in the public channel, to compute $X_2 = sX_1$ and $K_s=r_sX_1=r_sr_mP$, he/she needs to know the secret key s and the random number r_s of CS, or the random number r_m of MC_i. However, his/her dream will not come true due to the hardness of ECDL problem and CDH problem.

4.1.3. Mutual Authentication. In the proposed scheme, CS with s verifies the legitimacy of MC_i by checking M_1. If M_1 is valid, CS authenticates MC_i. On the other hand, MC_i authenticates CS by checking M_3 and CS will pass the test if M_3 is valid. Thus, the proposed scheme achieves mutual authentication.

4.1.4. Offline Dictionary Attack. Suppose the lost/stolen smartcard is obtained by the adversary and he/she reveals the secret information $\{D_1, D_2, ID_s, h(\cdot), P, K_{pup}, m\}$ from the smartcard by performing the side-channel attacks[36, 37] and fully controls the public channel. We will use two aspects to demonstrate that the proposed scheme is secure against offline dictionary attack.

If the adversary uses D_2 and conduct an offline dictionary attack as follows:

(1) The adversary chooses a pair (ID_i^*, PW_i^*) from the dictionary space of D_{ID} and D_{PW}, respectively.

(2) The adversary computes $RPW'=h(r_i||PW_i^*)$ and $D_2' = r_i \oplus (h(ID_i^*||h(r_i||PW_i^*) \bmod m)$.

(3) The adversary verifies the correctness of ID_i^* and PW_i^* by checking whether $D_2' = D_2$ holds. If it holds, the adversary has found a correct pair (ID_i, PW_i). Otherwise, the adversary will repeat step (1)~(3) until $D_2' = D_2$.

However, the adversary will not succeed for the following two reasons. First, the adversary has no knowledge of r_i and 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_i and PW_i successfully. Second, suppose the adversary knows r_i; it is also infeasible for him/her to find a correct pair (ID_i, 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 $|D_{ID}| * |D_{PW}|/m \approx 2^{32}$ candidates of (ID_i, PW_i) pair. Therefore, the number of (ID_i, 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_i from $M_2 = h(ID_i||X_2||PID_i)$, PID_i and X_1 are available from the public channel and $X_2 = sX_1$. However, the adversary cannot calculate X_2 because he/she knows nothing about the secret key s of CS. Therefore, the adversary fails to conduct such an attack.

In short, the proposed scheme is secure from dictionary attack.

4.1.5. Privileged Insider Attack. In the proposed scheme, MC_i submits $\{ID_i, h(r_i||PW_i)\}$ 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 $\{PID_i, X_1, M_2\}$, and the random number r_s 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 $\{ID_i, b_i, T_i, SCN_i\}$ 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_i, PW_i) or replay the login request $\{PID_i, X_1, M_2\}$ successfully as we analyzed above. Furthermore, if he/she chooses a random number r_a and computes $X_{1a} = r_ap$, forges PID_a and M_a, constructs the login request message $\{PID_a, X_{1a}, M_a\}$, and sends it to CS. However, CS cannot compute the correct ID_i in the table according to the login request $\{PID_a, X_{1a}, M_a\}$ from the adversary, which results in the computed M_a' not being equal to the received M_a. 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

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_{send} queries of Send oracle with execution time t, $Adv^{AKE}_{\mathcal{P},\mathcal{D}}(\mathcal{A})$ denotes the adversary $\mathcal{A}$ in breaking AKE security of $\mathcal{P}$. Under the difficult assumption of CDH problem, if the one-way hash function behaves like a random oracle and the signature scheme in $\mathcal{P}$ is unforgeable against adaptive chosen message attacks, then*

$$Adv^{AKE}_{\mathcal{P},\mathcal{D}}(\mathcal{A}) \le C' \cdot q_{send}^{s'} + \varepsilon(l) \quad (1)$$

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

Proof. We prove this theorem with a series of games Gm_i (i=0,1,2,3,4,5,6). In each Gm_i, the adversary will guess a correct bit with the Test query and this event is denoted as S_i and the corresponding probability is $Pr[S_i]$.

Gm_0: This game is considered as the real attack scenario under random oracle model. According to the definition of $\mathcal{A}$'s advantage [43], we have

$$Adv^{AKE}_{\mathcal{P},\mathcal{D}}(\mathcal{A}) = Pr[S_0] \quad (2)$$

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 Gm_0. Thus, we have

$$|Pr[S_1] - Pr[S_0]| \le \varepsilon(l) \quad (3)$$

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 $\{PID_i, X_1, M_2, Y, M_3$, and $M_4\}$. If the collisions occur, we abort the game and let the adversary win. According to the birthday paradox, we have

$$|Pr[S_2] - Pr[S_1]| \le \varepsilon(l) \quad (4)$$

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_3, the Diffie-Hellman key K_s $(=r_sX_1)$ and $K_m(=r_mY)$ are removed from the input list, i.e., the session key $SK= h'(ID_i||ID_S||X_2||Y)$ and authenticator $M_3=h'(ID_i'||ID_S||X_1'||X_2'||Y)$. In Gm_2, we have ruled out the collisions of hash function and the transcripts. Thus, the adversary is capable of distinguishing Gm_3 and Gm_2 only if he/she can calculate the Diffie-Hellman key K_s or K_m in passive session and sends a query (ID_i,ID_S,X_2,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, and $b_2 \in Z_q^*$ for each session and set $U=a_1X+b_1P$ and $V=a_2Y+b_2P$. If the adversary is able to distinguish the game Gm_3 and Gm_2, a query (ID_i,ID_S,X_2,Y,K_s) is made to the hash oracle. This means that the adversary can compute $(K-a_1b_2X-a_2b_1Y-b_1b_2P)/a_1a_2$ as an answer to the CDH instance (X,Y). Under the difficulty of CDH problem, we have

$$|Pr[S_3] - Pr[S_2]| \le \varepsilon(l) \quad (5)$$

Gm_4: In this game, we start to handle the active session for Send $(CS,\{M_4\})$ 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_i. The rule of the participants process queries is modified as follows.

Compute $M_4'=h(ID_i||ID_S||X_2||K_m||SK_{m\text{-}s})$ and check whether M_4' is equal to the received M_4. If it is true, the cloud server CS looks up a record $((PID_i, X_1, M_2),(Y, M_3),(M_4))$ from the hash list L_h. We terminate the game if the record exists. The authenticator M_4 in the proposed scheme is unforgeable due to the hardness of CDH problem. Thus, we have

$$|Pr[S_4] - Pr[S_3]| \le \varepsilon(l) \quad (6)$$

Gm_5: In this game, we continue to the active session for Send$(MC_i,\{Y, M_3\})$. We also define this game by terminating the game with the following rule, where the adversary is luck 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 $(*||ID_i|| * ||K_s)$ in the hash list L_h, and we terminate the game if the result is null. Otherwise, compute the session key $SK=h(ID_i||ID_S||X_2||Y||K_s)$, $M_3=h(ID_i||ID_S||X_1||X_2||Y||K_s)$.

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

$$|Pr[S_5] - Pr[S_4]| \le \varepsilon(l) \quad (7)$$

Gm_6: In this game, we modify the simulation rule of Send$(CS, \{PID_i, X_1, M_2\})$ query for the last time. When a Send$(CS, \{PID_i, X_1, M_2\})$ query is submitted, the CS first computes X_2,X_3,ID_i,M_1,M_2', and checks whether $M_2' = M_2$ holds. If the result is true and the message $\{PID_i, X_1, M_2\}$ is forged by the adversary, we abort the simulation and let $\mathcal{A}$ win. Afterwards, we evaluate the success probability of forging the message $\{PID_i, X_1, M_2\}$. Note that the authenticator $M_2=h(ID_i||X_2||PID_i)$ and, similarly with the analysis in the previous game, we can know that the success probability of forging authenticator M_2 is negligible. Furthermore, based on the difficulty of ECDL problem, the probability of successful

forgery of message $\{PID_i, X_1, M_2\}$ is negligible. Thus, we obtain

$$\left|Pr\left[S_6\right] - Pr\left[S_5\right]\right| \leq \varepsilon(l) \tag{8}$$

In the last game, the session keys are chosen randomly and the advantage of $\mathcal{A}$ in guessing session keys is negligible and the active sessions are aborted without accepting if $\mathcal{A}$ forges the message. The only possibility for $\mathcal{A}$ to win the game is to corrupt the smartcard and guess the password of MC_i. The advantage $\mathcal{A}$ has no advantage to get the password from the game. Based on the Zipf's law, we obtain

$$|Pr\left[S_6\right] \leq C' \cdot q_{send}^{s'} \tag{9}$$

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

Theorem 2. *Assume that $\mathcal{P}$ represents the proposed scheme and $\mathcal{A}$ is a PPT adversary breaking the anonymity of $\mathcal{P}$. The advantage of $\mathcal{A}$ in breaking the anonymity of $\mathcal{P}$ is bounded by*

$$Adv_{\mathcal{P}}^{anon}(\mathcal{A}) \leq \varepsilon(l) \tag{10}$$

Proof. We suppose that $\mathcal{A}$ can break the anonymity of $\mathcal{P}$ with a nonnegligible advantage. We reach this aim by employing $\mathcal{A}$ 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_mP, sP, r_msP) and (P, r_mP, sP, r), where s is the private key of CS.

(1) Let U be a valid user owning his smartcard and password.

(2) Let $X_{U1}=r_mP$, $X_{U2}=r_msP$, and execute the subsequent procedure with CS as the protocol definition. We use sid_U^i as the session identifier of this protocol execution.

(3) Let $X_{U1}=r_mP$, $X_{U2}=r$, and execute the subsequent procedure with CS as the protocol definition. The corresponding session identifier of this protocol execution is labelled as sid_U^i. CS may respond with rejection according to the first message from user U. In this case, to make sid_U^i and sid_U^j have the same structure, U can set $r_s \in Z_q^*$ and chooses two random bit strings for M_3 and M_4, respectively.

(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 CS using X_{U1}, X_{U2}. In this case, the session identifier is denoted as sid_U^k.

(5) Two queries TestAnonymity(sid_U^i, sid_U^j) and TestAnonymity(sid_U^i, sid_U^k) are made by $\mathcal{A}$, and the returned bits are denoted as b_1 and b_2, respectively.

(6) If b_1=0 and b_2=0, output "none is a Diffie-Hellman tuple"; if b_1=0 and b_2=1, output "(P, r_mP, sP, r_msP) is a Diffie-Hellman tuple"; if b_1=1 and b_2=0, output "(P, r_mP, sP, r) is a Diffie-Hellman tuple"; if b_1=1 and b_2=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_mP is different in every protocol run. Based on the self-reducibility of CDH problem, we obtain $\mathrm{Adv}_{\mathcal{CDH},\mathcal{P}}(U) = |\Pr[U(P, r_mP, sP, r_msP)=1]\text{-}\Pr[U(P,r_mP,sP,r)=1]|$, where s is a fixed value and $r_m \in Z_q^*$. Thus, we have $\mathrm{Adv}_{\mathcal{CDH},\mathcal{P}}(U) \geqslant |\Pr[\mathrm{TestAnonymity}(sid_U^i, sid_U^j)=1] - \Pr[\mathrm{TestAnonymity}(sid_U^i, sid_U^k)=1]|$. That is to say, $\mathcal{A}$ 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. □

Table 1: Computation cost at client and server side.

	T_{PM}	T_{PA}	T_{IN}	T_H
Server	1.17ms	<0.1ms	<1ms	<0.001ms
Client	0.13s	<0.1s	<0.01s	<0.001s

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 is 512 bits and 160 bits, respectively. For the convenience of evaluating computation cost, we set $T_H, T_{PM}, T_{PA}, T_{IN}$ 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_i 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.

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.

	[22]	[26]	[27]	[28]	[24]	[29]	[25]	[30]	Ours
S1	$7T_{PM}+3T_{PA}+10T_H$	$4T_{PM}+4T_H$	$4T_{PM}+T_{PA}+5T_H$	$3T_{PM}+T_{PA}+6T_H$	$3T_{PM}+T_{PA}+9T_H$	$5T_{PM}+T_{IN}+9T_H$	$4T_{PM}+5T_H$	$4T_{PM}+3T_H$	$3T_{PM}+T_{PA}+7T_H$
S2	1.22 s	0.524 s	0.625 s	0.496 s	0.499 s	0.669 s	0.525 s	0.653 s	0.497 s
S3	$5T_{PM}+2T_{PA}+6T_H$	$5T_{PM}+5T_H$	$4T_{PM}+T_{IN}+5T_H$	$3T_{PM}+T_{PA}+6T_H$	$3T_{PM}+6T_H$	$3T_{PM}+T_{IN}+7T_H$	$4T_{PM}+5T_H$	$5T_{PM}+4T_H$	$3T_{PM}+T_{PA}+6T_H$
S4	6.056 ms	5.855 ms	5.685 ms	3.616 ms	3.516 ms	3.517 ms	4.685 ms	5.854 ms	3.616 ms

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.

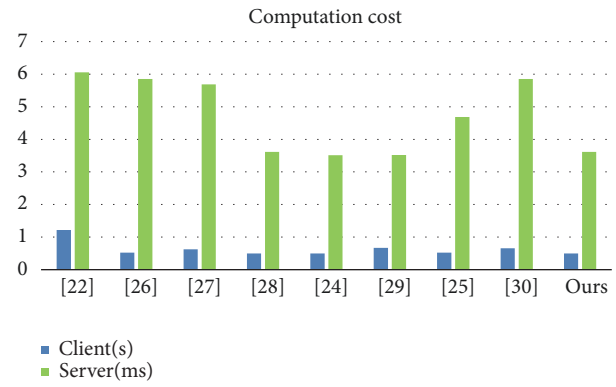

Figure 4: Computation cost comparison.

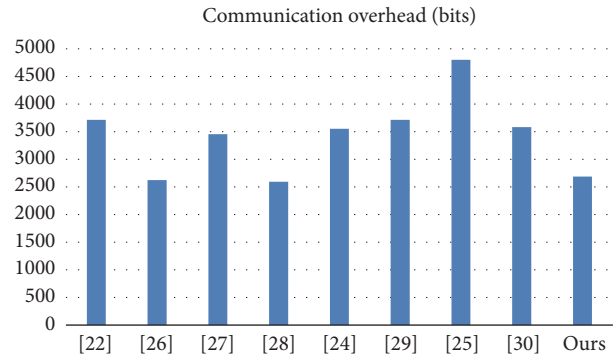

Figure 5: Communication overhead comparison.

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/s10916-015-0244-0]). (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/s11277-016-3745-3]). (6) The [28] data used to support the findings of this study have been deposited in the [Springer] repository ([DOI: 10.1007/s11042-015-3166-4]). (7) The [29] data used to support the findings of this study have been deposited in the [ScienceDirect] repository ([DOI: 10.1016/j.pmcj.2015.12.003]). (8) The [30] data used to support the findings of this study have been deposited in the [Springer] repository ([DOI: 10.1007/s11227-017-2048-0]).

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.

	[22]	[26]	[27]	[28]	[24]	[29]	[25]	[30]	Ours
I1	$3l_p+4l_h$	$l_i+2l_p+3l_h+2l_t$	$3l_p+2l_h+2l_t$	$2l_p+3l_h+2l_t$	$3l_p+3l_h$	$3l_p+4l_h$	$l_i+4l_p+4l_h$	$l_i+3l_p+3l_h$	$2l_p+4l_h$
I2	3712	2624	3456	2592	3552	3712	4800	3584	2688
I3	3	3	2	3	3	3	2	3	3

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

Table 4: Comparison of security properties.

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

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

[1] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, "Security and privacy challenges in mobile cloud computing: Survey and way ahead," *Journal of Network and Computer Applications*, vol. 84, pp. 38–54, 2017.

[2] Mordor Intelligence Industry Report, Mobile Cloud Market, https://www.mordorintelligence.com/industry-reports/global-mobile-cloud-market-industry, 2018.

[3] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[4] V. S. Miller, "Use of elliptic curves in cryptography," in *Proceedings of the Conference on The Theory And Application of Cryptographic Techniques*, vol. 218, pp. 417–426, Springer, 1985.

[5] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, "Recommendation for key management part 1: General (revision 3)," *NIST Special Publication*, vol. 800, no. 57, pp. 1–147, 2012.

[6] C.-L. Lin and T. Hwang, "A password authentication scheme with secure password updating," *Computers & Security*, vol. 22, no. 1, pp. 68–72, 2003.

[7] M. Peyravian and C. Jeffries, "Secure remote user access over insecure networks," *Computer Communications*, vol. 29, no. 5, pp. 660–667, 2006.

[8] H.-M. Sun and H.-T. Yeh, "Password-based authentication and key distribution protocols with perfect forward secrecy," *Journal of Computer and System Sciences*, vol. 72, no. 6, pp. 1002–1011, 2006.

[9] X. Li, W. Qiu, D. Zheng, K. Chen, and J. Li, "Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 2, pp. 793–800, 2010.

[10] J. Xu, W.-T. Zhu, and D.-G. Feng, "An improved smart card based password authentication scheme with provable security," *Computer Standards & Interfaces*, vol. 31, no. 4, pp. 723–728, 2009.

[11] J. Ma, W. Yang, M. Luo, and N. Li, "A study of probabilistic password models," in *Proceedings of the 35th IEEE Symposium on Security and Privacy (SP '14)*, pp. 689–704, IEEE, May 2014.

[12] J. Gosney, "Password cracking HPC," in *Passwords 2012 Security Conference*, University of Oslo, Oslo, Norway, 2012, http://bit.ly/1y00I3O.

[13] M. L. Das, *Two-Factor User Authentication in Wireless Sensor Networks*, IEEE Press, 2009.

[14] D. Nyang and M.-K. Lee, "Improvement of das's two-factor authentication protocol in wireless sensor networks," *IACR Cryptology ePrint Archive*, vol. 2009, p. 631, 2009.

[15] H.-F. Huang, Y.-F. Chang, and C.-H. Liu, "Enhancement of two-factor user authentication in wireless sensor networks," in *Proceedings of the 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP '10)*, pp. 27–30, October 2010.

[16] D. He, Y. Gao, S. Chan, C. Chen, and J. Bu, "An enhanced two-factor user authentication scheme in wireless sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 10, no. 4, pp. 361–371, 2010.

[17] Q. Xie, "Improvement of a security enhanced one-time two-factor authentication and key agreement scheme," *Scientia Iranica*, vol. 19, no. 6, pp. 1856–1860, 2012.

[18] M. K. Khan and K. Alghathbar, "Cryptanalysis and security improvements of 'two-factor user authentication in wireless sensor networks'," *Sensors*, vol. 10, no. 3, pp. 2450–2459, 2010.

[19] C.-C. Lee, C.-T. Li, and S.-D. Chen, "Two attacks on a two-factor user authentication in wireless sensor networks," *Parallel Processing Letters*, vol. 21, no. 1, pp. 21–26, 2011.

[20] S. H. Islam and G. Biswas, "Dynamic ID-based remote user mutual authentication scheme with smartcard using Elliptic Curve Cryptography," *Journal of Electronics (China)*, vol. 31, no. 5, pp. 473–488, 2014.

[21] M. Sarvabhatla and C. S. Vorugunti, "A secure and robust dynamic ID-based mutual authentication scheme with smart card using elliptic curve cryptography," in *Proceedings of the 7th International Workshop on Signal Design and Its Applications in Communications, IWSDA 2015*, pp. 75–79, India, September 2015.

[22] J. Qu and X.-L. Tan, "Two-factor user authentication with key agreement scheme based on elliptic curve cryptosystem," *Journal of Electrical and Computer Engineering*, vol. 2014, 16 pages, 2014.

[23] B. Huang, M. K. Khan, L. Wu, F. T. B. Muhaya, and D. He, "An Efficient Remote User Authentication with Key Agreement Scheme Using Elliptic Curve Cryptography," *Wireless Personal Communications*, vol. 85, no. 1, pp. 225–240, 2015.

[24] S. A. Chaudhry, H. Naqvi, K. Mahmood, H. F. Ahmad, and M. K. Khan, "An Improved Remote User Authentication Scheme Using Elliptic Curve Cryptography," *Wireless Personal Communications*, vol. 96, no. 4, pp. 5355–5373, 2017.

[25] C.-C. Chang, H.-L. Wu, and C.-Y. Sun, "Notes on "Secure authentication scheme for IoT and cloud servers"," *Pervasive and Mobile Computing*, vol. 38, pp. 275–278, 2017.

[26] M. S. Farash and M. A. Attari, "A secure and efficient identity-based authenticated key exchange protocol for mobile client-server networks," *The Journal of Supercomputing*, vol. 69, no. 1, pp. 395–411, 2014.

[27] S. A. Chaudhry, H. Naqvi, T. Shon, M. Sher, and M. S. Farash, "Cryptanalysis and improvement of an improved two factor authentication protocol for telecare medical information systems," *Journal of Medical Systems*, vol. 39, no. 6, pp. 1–11, 2015.

[28] Q. Xie, D. S. Wong, G. Wang, X. Tan, K. Chen, and L. Fang, "Provably secure dynamic ID-based anonymous two-factor authenticated key exchange protocol with extended security model," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1382–1392, 2017.

[29] Y. Lu, L. Li, H. Peng, and Y. Yang, "An anonymous two-factor authenticated key agreement scheme for session initiation protocol using elliptic curve cryptography," *Multimedia Tools and Applications*, vol. 76, no. 2, pp. 1801–1815, 2017.

[30] S. Kumari, M. Karuppiah, A. K. Das, X. Li, F. Wu, and N. Kumar, "A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers," *The Journal of Supercomputing*, pp. 1–26, 2017.

[31] H. Mun, K. Han, Y. S. Lee, C. Y. Yeun, and H. H. Choi, "Enhanced secure anonymous authentication scheme for roaming service in global mobility networks," *Mathematical and Computer Modelling*, vol. 55, no. 1-2, pp. 214–222, 2012.

[32] D. Zhao, H. Peng, L. Li, and Y. Yang, "A secure and effective anonymous authentication scheme for roaming service in global mobility networks," *Wireless Personal Communications*, vol. 78, no. 1, pp. 247–269, 2014.

[33] I. Memon, I. Hussain, R. Akhtar, and G. Chen, "Enhanced privacy and authentication: an efficient and secure anonymous communication for location based service using asymmetric cryptography scheme," *Wireless Personal Communications*, vol. 84, no. 2, pp. 1487–1508, 2015.

[34] A. G. Reddy, A. K. Das, E.-J. Yoon, and K.-Y. Yoo, "A secure anonymous authentication protocol for mobile services on elliptic curve cryptography," *IEEE Access*, vol. 4, pp. 4394–4407, 2016.

[35] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, L. Leng, and N. Kumar, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Computer Networks*, vol. 101, pp. 42–62, 2016.

[36] T. H. Kim, C. Kim, and I. Park, "Side channel analysis attacks using AM demodulation on commercial smart cards with SEED," *The Journal of Systems and Software*, vol. 85, no. 12, pp. 2899–2908, 2012.

[37] N. Veyrat-Charvillon and F. X. Standaert, *Generic Side-Channel Distinguishers: Improvements and Limitations*, Springer, Berlin, Germany, 2011.

[38] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, Springer Publishing Company, Incorporated, 2010.

[39] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pp. 62–73, ACM, 1993.

[40] Q. Jiang, J. Ma, and F. Wei, "On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Systems Journal*, pp. 1–4, 2016.

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

[42] D. He, N. Kumar, M. K. Khan, L. Wang, and J. Shen, "Efficient privacy-aware authentication scheme for mobile cloud computing services," *IEEE Systems Journal*, no. 99, pp. 1–11, 2017.

[43] F. Wei, P. Vijayakumar, Q. Jiang, and R. Zhang, "A mobile intelligent terminal based anonymous authenticated key exchange protocol for roaming service in global mobility networks," *IEEE Transactions on Sustainable Computing*, no. 99, pp. 2377–3782, 2018.

[44] D. Pointcheval, "Provable security for public key schemes," in *Contemporary Cryptology*, Advanced Courses in Mathematics - CRM Barcelona, pp. 133–190, Springer, 2005.

[45] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," *Designs, Codes and Cryptography*, vol. 19, no. 2-3, pp. 173–193, 2000.

[46] M. Scott, N. Costigan, and W. Abdulwahab, "Implementing cryptographic pairings on smartcards," in *Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems*, vol. 4249 of *Lecture Notes in Computer Science*, pp. 134–147, Springer, 2006.

[47] H. Krawczyk, "HMQV: a high-performance secure Diffie-Hellman protocol," in *Advances in Cryptology—CRYPTO 2005*, vol. 3621 of *Lecture Notes in Computer Science*, pp. 546–566, Springer, Berlin, Germany, 2005.