Computers, Materials & Continua CMC, vol.64, no.1, pp.297-312, 2020

Efficient Hierarchical Multi-Server Authentication Protocol for Mobile Cloud Computing

Jiangheng Kou[1], Mingxing He[1,*], Ling Xiong[1], Zihang Ge[2] and Guangmin Xie[1]

Abstract: With the development of communication technologies, various mobile devices and different types of mobile services became available. The emergence of these services has brought great convenience to our lives. The multi-server architecture authentication protocols for mobile cloud computing were proposed to ensure the security and availability between mobile devices and mobile services. However, most of the protocols did not consider the case of hierarchical authentication. In the existing protocol, when a mobile user once registered at the registration center, he/she can successfully authenticate with all mobile service providers that are registered at the registration center, but real application scenarios are not like this. For some specific scenarios, some mobile service providers want to provide service only for particular users. For this reason, we propose a new hierarchical multi-server authentication protocol for mobile cloud computing. The proposed protocol ensures only particular types of users can successfully authenticate with certain types of mobile service providers. The proposed protocol reduces computing and communication costs by up to 42.6% and 54.2% compared to two superior protocols. The proposed protocol can also resist the attacks known so far.

Keywords: Multi-server authentication, cryptography, hierarchical authentication, mobile cloud computing.

1 Introduction

With the development of mobile technology and communication technology, a variety of mobile devices appear in our lives like laptops, tablets, mobile phones, etc. According to a recent report from Pew Research Center [Silver (2019)], it is estimated that more than 5 billion people have mobile devices and over half of these connections are smartphones. People in advanced economies are more likely to have mobile phones in particular. China, a country with a huge population in the world, has about 900 million mobile users, which brings a huge opportunity for the development of mobile cloud computing. With the increase of mobile users, a variety of mobile cloud services appear in our lives, such as mobile banking, mobile online shopping, mobile online game, and mobile TV, which

[1] School of Computer and Software Engineering, Xihua University, Chengdu, 610039, China.

[2] Illinois Institute of Technology, Miles (Main) Campus, Chicago, 60616, USA.

[*] Corresponding Author: Mingxing He. Email: he_mingxing64@aliyun.com.

Received: 17 January 2020; Accepted: 04 March 2020.

all can be accessed from anywhere at any time. The revolution in wireless communication and mobile technology brings great benefits to mobile users.

The traditional single-server architecture for mobile cloud computing was proposed [Lamport (2019); Shen, Tan, Wang et al. (2015); Li, Dai, Tian et al. (2009)] to allow users to use the services remotely. However, due to the increasing number of mobile users and mobile service providers, traditional single-server architecture becomes inefficient [He, Zeadally, Kumar et al. (2016)] because of the limitation of computation and communication capabilities. For changing this situation, researchers introduced the multi-server architecture for mobile cloud computing. A multi-server architecture for mobile cloud computing is shown in Fig. 1.

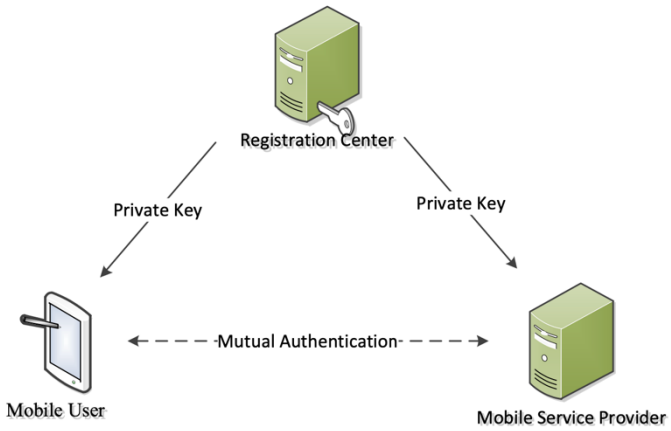

Figure 1: Network model

With the development of multi-server architecture for mobile cloud computing, multi-server architecture authentication protocol for mobile cloud computing has been widely implemented. However, there exist some security issues which any adversary can easily intercept, modify, replay, and delay the transmission of messages [Fang, Li, Yun et al. (2019)]. Security is an important aspect of the authentication protocol for mobile cloud computing, and a good protocol must give solutions to these security issues [Gopinath and Bhuvaneswaran (2019); Jiang, Liu, Yang et al. (2019); Kou, Shi, Zhang et al. (2019)]. The current authentication protocols can be divided into two types [Odelu, Das, Kumari et al. (2017)]: (i) authentication protocol with online registration center [Yoon and Yoo (2013); He and Wang (2015); Odelu, Das and Goswami (2015); Chandrakar and Om (2017); Feng, He, Zeadally et al. (2018); Amin, Kumar, Biswas et al. (2018)], (ii) authentication protocol without online registration center [Choi, Hwang, Lee et al. (2005); Tsai and Lo (2015); Tseng, Huang, Tsai et al. (2016); Odelu, Das, Kumari et al. (2017); He, Zeadally, Kumar et al. (2016); Xie, Wong, Wang et al. (2017); Xiong, Peng, Peng et al. (2017a); Xiong, Peng, Peng et al. (2017b); Jiang, Ma and Wei (2018); Chatterjee, Roy, Das et al. (2018)].

Because the second type of authentication protocols have many advantages, in recent years, many researchers have been working on it and have proposed many protocols that improve

performance and security [He, Zeadally, Kumar et al. (2016)]. However, there is a situation, in the second type of authentication protocol, that has not been considered. In the second type authentication protocol, there is no registration center involved in the authentication phase; therefore, no third party can verify the authentication of the mobile users accurately. When a mobile user registers at the registration center, he/she can authenticate with all mobile service providers who have registered at the registration center and free access to the resources provided by mobile service providers. The lake of this function prevents the deployment of the existed protocol in various real-time applications. Fig. 2 shows an environment that particular mobile users are authenticated with different level mobile service providers, which cannot be achieved by existing protocols.

It remains a significant challenge to construct a hierarchical multi-server authentication protocol for mobile cloud computing with better efficiency and security to protect the authorized users' rights for various practical mobile applications.

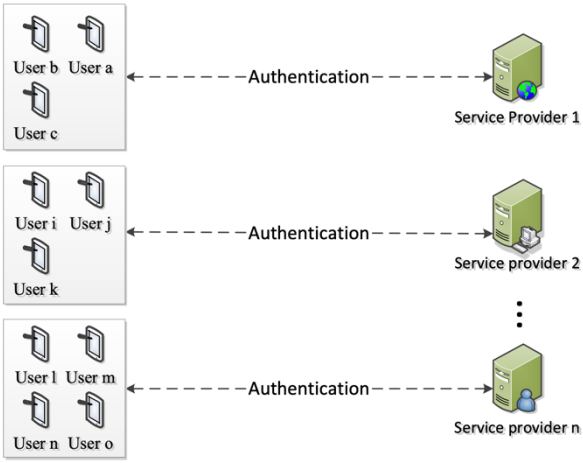

Figure 2: Hierarchical authentication model

1.1 Organization of the paper

The remainder of this paper is sketched as follows. Section 2 discusses the related work. Section 3 describes our contribution. Section 4 describes preliminaries. In Section 5, we show the details of the proposed protocol. Section 6 gives out the formal security proof of the proposed protocol. Section 7 presents a comparison of our protocol with two superior protocols on security, computation, and communication. Section 8 concludes the paper.

2 Related work

When people realized that the single-server architecture is not efficient enough for practical use, research on multi-server architecture began. Li et al. [Li, Dai, Tian et al. (2009)] found that the protocols in single-server architecture cannot apply to multi-server architecture. Hence, they proposed a multi-server architecture authentication protocol using neural network. In recent years many improvements in the authentication protocol of multi-server

architecture have been made by researchers. However, these protocols transmit the user's identity without protection; therefore, they cannot provide user anonymity.

Yoon et al. [Yoon and Yoo (2013)] proposed a biometrics-based authentication protocol which can resist smart card stolen attack. He et al. [He and Wang (2015)] constructed the first truly three-factor authentication protocol for the multi-server environment. Odelu et al. [Odelu, Das and Goswami (2015)] proposed an improved protocol to solve the security problems in He et al. [He and Wang (2015)]. Chandrakar et al. [Chandrakar and Om (2017)] proposed a new security-enhanced three-factor protocol to get more security. Feng et al. [Feng, He, Zeadally et al. (2018)] proposed an enhanced protocol that can resist several attacks and have user anonymity. Amin et al. [Amin, Kumar, Biswas et al. (2018)] proposed a lightweight authentication protocol that has lower computational and communication costs. However, the previous protocols require that the registration center be always be online, which increases communication costs and complexity.

To address the above problems, Choi et al. [Choi, Hwang, Lee et al. (2005)] proposed the first authentication protocol without the online registration center. Tseng et al. [Tseng, Huang, Tsai et al. (2016)] proposed a list-free ID-based authentication protocol using bilinear pairings for multi-server architecture. However, Tseng et al. [Tseng, Huang, Tsai et al. (2016)] did not provide credentials privacy and un-traceability for users. Odelu et al. [Odelu, Das, Kumari et al. (2017)] and He et al. [He, Zeadally, Kumar et al. (2016)] proposed new protocols that saved the computation and communication costs, and fixed the security problems. Xie et al. [Xie, Wong, Wang et al. (2017)] proposed an enhanced protocol. Afterward, Xiong et al. [Xiong, Peng, Peng et al. (2017a)] proposed an enhanced protocol for distributed mobile cloud and at the same time Xiong et al. [Xiong, Peng, Peng et al. (2017b)] proposed a new lightweight Anonymous Authentication Protocol. Jiang et al. [Jiang, Ma and Wei (2018)] performed a security analysis of Tsai et al. [Tsai and Lo (2015)] protocol and pointed out their defects. Chatterjee et al. [Chatterjee, Roy, Das et al. (2018)] proposed a biometric-based protocol using the chaotic map to enhance the security of multi-server architecture. These protocols did bring more security to the authentication, but cannot effectively control the user's access rights. So registered users can have access to all registered service providers, which seems unreasonable in practical applications.

3 Our contribution

Our contributions in this paper are presented as follows.

The proposed protocol embeds an authentication right parameter into the user's private key to achieve hierarchical authentication functionality that is not implemented by other protocols. In the proposed protocol, at the authentication phase, the session key is established between the service provider and user without involving the RC. This process significantly reduces communication costs and makes the authentication process faster and more efficient. The proposed protocol can satisfy the security requirement of multi-server architecture and is provably secure in the general security model. The proposed protocol reduces computing costs and communication costs by up to 42.6% and 54.2% compared to two top protocols.

4 Preliminaries

In this section, we introduce the mathematical preliminaries of the proposed protocol.

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be an additive cyclic group and a multiplicative cyclic group, both of them has a large prime order q. Let $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ denote a bilinear map. Suppose P is a generator of $\mathbb{G}_1$, g is a generator of $\mathbb{G}_2$. A bilinear map $\hat{e}$ has properties below.

• Bi-linearity: For all $P, Q \in \mathbb{G}_1$ and for all $a, b \in Z_q^*$, $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.

• Computability: There exists an algorithm that can successfully compute $\hat{e}(P, Q)$ for all $P, Q \in \mathbb{G}_1$.

• Non-degeneracy: There exists $P, Q \in \mathbb{G}_1$ such that $\hat{e}(P, Q) \neq 1$, where 1 is the identity element of $\mathbb{G}_2$.

We list the hard problems that we used in the proposed protocol as follows.

• **Discrete Logarithm (DL) Problem:** Given an element $x \in \mathbb{G}_2$, it is hard to compute $a \in Z_q^*$ such that $x = g^a$.

• **Computational Diffie-Hellman (CDH) Problem:** Given two elements $g^a, g^b \in \mathbb{G}_2$, it is hard to compute $g^{a \cdot b} \in \mathbb{G}_2$, where a and b are unknown and randomly choose from Z_q^*.

• **Modified Bilinear Inverse Diffie-Hellman with k value (k-mBIDH) Problem:** Given k elements $\{\alpha_1, \alpha_2, \ldots, \alpha_k\}(\alpha_i \in Z_q^*)$ and $k+2$ elements $\left\{\tau \cdot P, \eta \cdot P, \frac{1}{\tau+\alpha_1} \cdot P, \frac{1}{\tau+\alpha_2} \cdot P, \ldots, \frac{1}{\tau+\alpha_k} \cdot P\right\}$ each of them is in $\mathbb{G}_1$, it is hard to compute $\hat{e}(P, P)^{\frac{\eta}{\tau+\alpha}}$, where $\alpha \notin \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$, τ and η are two unknown elements in Z_q^*.

5 The proposed protocol

5.1 Registration center initialization phase

The registration center RC runs the generation function $Gen(1^n)$ which takes security parameter $n \in Z^+$ as input and outputs parameters as follows.

• Step 1: RC chooses the bilinear map groups $\mathbb{G}_1$ and $\mathbb{G}_2$ with a prime order q, the generator $P \in \mathbb{G}_1$ and $g = \hat{e}(P, P) \in \mathbb{G}_2$, where $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ is a bilinear map.

• Step 2: RC chooses cryptographic hash functions $H_1: \{0,1\}^* \to Z_q^*$, $H_2: \mathbb{G}_2 \to Z_q^*$, $H_3: \{0,1\}^* \to \mathbb{G}_1$, $H_4: \{0,1\}^* \to \{0,1\}^n$.

• Step 3: RC chooses a random number s from Z_q^* as master key, computes the corresponding public key $P_{pub} = sP \in \mathbb{G}_1$ and selects a set of authentication right parameters $\{\varphi_1, \varphi_2, \cdots, \varphi_n\}$, each parameter φ_i represents the authentication right level.

• Step 4: RC publishes $\{\mathbb{G}_1, \mathbb{G}_2, q, \hat{e}, P, P_{pub}, g, H_1, H_2, H_3, H_4\}$.

5.2 Mobile user registration phase

If a mobile user U_i wants to register at the registration center RC, the following steps are executed. The main steps are provided in Tab. 1.

• Step 1: U_i sends his/her identity ID_{U_i} to RC via a secure channel.

• Step 2: RC selects the authentication right parameter φ_i according to mobile user's level, and computes the U_i's private key $d_{U_i} = \frac{1}{s+H_1(ID_{U_i}\|e\|\varphi_i)} \cdot P$, where e is the expire date of the private key. RC sends d_{U_i} to U_i via a secure channel.

• Step 3: U_i computes $(\sigma_i,\theta_i) \leftarrow f(b_i)$ using fuzzy-extractor generation procedure $f(\cdot)$ [Dodis, Reyzin and Smith (2004)], where σ_i is a biometric key, θ_i is public reproduction parameter and b_i is his/her personal biometrics. U_i computes $A = d_{U_i} \oplus H_3(pw \parallel \sigma_i)$ and $B = H_4(ID_{U_i} \parallel pw \parallel \sigma_i)$, where pw is his/her password. Finally, U_i stores $\{\theta_i, A, B, e, f(\cdot), f^{-1}(\cdot), t, H_1, H_2, H_3, H_4\}$ on its mobile device, where t is the threshold in fuzzy extractor, $f(\cdot)$ is the probabilistic generation procedure for outputting σ_i and θ_i, $f^{-1}(\cdot)$ is the deterministic reproduction procedure that can recover σ_i and θ_i from a new personal biometrics input.

Table 1: Mobile user registration phase

Mobile user U_i	Registration center RC
$ID_{U_i} \rightarrow$	
	Compute $d_{U_i} = \frac{1}{s+H_1(ID_{U_i}\|e\|\varphi_i)} \cdot P$
	$\leftarrow d_{U_i}$
computes $(\sigma_i, \theta_i) \leftarrow f(b_i)$	
$A = d_{U_i} \oplus H_3(pw \parallel \sigma_i)$,	
$B = H_4(ID_{U_i} \parallel pw \parallel \sigma_i)$	
Stores $\{\theta_i, A, B, e, f(\cdot), f^{-1}(\cdot), t, H_1, H_2, H_3, H_4\}$	

5.3 Mobile service provider registration phase

If a mobile service provider S_j wants to register at the RC, the following steps are executed. The main steps are provided in Tab. 2.

• Step 1: S_j sends his/her identity ID_{S_j} to RC via a secure channel.

• Step 2: RC computes the private key $d_{S_j} = \frac{1}{s+H_1(ID_{S_j})} \cdot P$ for S_j and sends d_{S_j} to him via a secure channel.

• Step 3: Finally, S_j saves d_{S_j}.

Table 2: Mobile service provider registration phase

Mobile service provider S_j	Registration center RC
$ID_{S_j} \rightarrow$	
	Computes private key $d_{S_j} = \frac{1}{s+H_1(ID_{S_i})} \cdot P$
	$\leftarrow d_{S_j}$
Stores d_{S_j}	

5.4 Mobile user and mobile service provider authentication phase

In this part, we will show the mutual authentication between a mobile user and a mobile service provider without involving the RC. The main steps are provided in Tab. 3.

• Step 1: U_i first inputs his/her biometrics b_i, identity ID_{U_i} and password pw to mobile device. Mobile device computes $\sigma_i = f^{-1}(\theta_i, b_i)$ and $B^* = H_4(ID_{U_i} \| pw \| \sigma_i)$, and verifies the validity of inputted biometrics and password by computing $B^* \stackrel{?}{=} B$. If it holds, mobile device retrieves U_i's private key by computing $d_{U_i} = A \oplus H_3(pw \| \sigma_i)$. Then U_i selects a random number $r_1 \leftarrow Z_q^*$, computes $g_1 = g^{r_1}$, $C = r_1 \cdot (H_1(ID_{S_j})P + P_{pub})$ by using the identity of S_j. Next U_i computes $D = H_1(ID_{U_i} \| \mathrm{e} \| ID_{S_j} \| g_1)$, $E = (r_1 + D) \cdot d_{U_i}$ and $F = \left(ID_{U_i} \| e \| ID_{S_j}\right) \oplus H_2(g^{r_1})$. Finally, U_i sends login message C, E, F to S_j.

• Step 2: After receiving $\{C, E, F\}$, S_j retrieves g_1 using his/her private key d_{S_j} as $g_1 = g^{r_1} = \hat{e}(C, d_{S_j})$. S_j retrieves D, ID_{U_i}, e, ID_{S_j} by computing $F \oplus H_2(g_1)$. S_j computes $\hat{e}(E, H_1(ID_{U_i} \| e \| \varphi_i) \cdot P + P_{pub}) \stackrel{?}{=} g_1 \cdot g^D$. If both are equal, means U_i can authenticate with S_j. Then S_j selects a random number $r_2 \leftarrow Z_q^*$ and computes $g_2 = g^{r_2}$, the session key is set as $sk = H_2(g_1^{r_2}) = H_2(g^{r_1 r_2})$. Finally, S_j calculates $G = H_4\left(sk \| g_1 \| g_2 \| ID_{S_j} \| C\right)$ and sends g_2 and G to U_i.

• Step 3: Upon receiving g_2 and G, U_i computes $sk = H_2(g_2^{r_1}) = H_2(g^{r_1 r_2})$, $G^* = H_4(sk \| g_1 \| g_2 \| ID_{S_j} \| C)$ and checks whether G and G^* are equal. If both are not equal, U_i aborts the session. Otherwise, U_i confirms S_j as a valid service provider and sets sk as the session key between U_i and S_j.

Table 3: Mobile user and mobile service provider authentication phase

Mobile user U_i	Mobile service provider S_j
$r_1 \leftarrow Z_q^*, g_1 = g^{r_1}$	
$C = r_1 \cdot \left(H_1\left(ID_{S_j}\right) \cdot P + P_{pub}\right)$	
$D = H_1(ID_{U_i} \| e \| ID_{S_j} \| g_1)$	
$E = (r_1 + D) \cdot d_{U_i}$	
$F = \left(ID_{U_i} \| e \| ID_{S_j}\right) \oplus H_2(g_1)$	
$C, E, F \rightarrow$	
	Retrieves $g_1 = g^{r_1} = \hat{e}(C, d_{S_j})$
	$\left(ID_{U_i} \| e \| ID_{S_j}\right) = F \oplus H_2(g_1)$
	$\hat{e}(E, H_1(ID_{U_i} \| e \| \varphi_i) \cdot P + P_{pub}) \stackrel{?}{=} g_1 \cdot g^D$
	Accept/reject
	$r_2 \leftarrow Z_q^*, g_2 = g^{r_2}$

	$sk = H_2(g_1^{r_2})$
	$G = H_4(sk \parallel g_1 \parallel g_2 \parallel ID_{S_j} \parallel C)$
	$\leftarrow G, g_2$
Computes $sk = H_2(g_2^{r_1})$	
$G^* = H_4(sk \parallel g_1 \parallel g_2 \parallel ID_{S_j} \parallel C)$	
Checks $G^* \stackrel{?}{=} G$	
Accept/reject sk	

6 Formal security analysis

We present a security model for the proposed protocol based on previous security model [Canetti and Krawczyk (2001)]. In this model, an adversary can control the communications between different parties and knows all public parameters. $\mathcal{A}$ can make queries to all hash functions.

There are U_i and S_j at the authentication phase of our protocol. The security of the proposed protocol is defined by a game played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. Let Π_Λ^l denote the lth instance of the participant of $\Lambda \in \{U_i, S_j\}$ respectively. In this game, $\mathcal{A}$ can issue queries to $\mathcal{C}$ and get answers from it as follows.

• $H_i(q_j)$: At any time $\mathcal{A}$ issues query q_j where q_j can be any string, $\mathcal{C}$ picks a random number $r_j \in Z_q^*$ and stores $< q_j, r_j >$ into list $\mathcal{H}_i^{list}$, where $i \in \{1,2,3,4\}$ and $j \in \{poly(n)\}$. Finally, $\mathcal{C}$ sends r_j to $\mathcal{A}$.

• $Execute(U_i, S_j)$: This query simulates the passive attacks and allows $\mathcal{A}$ to learn all transmitted messages between U_i and S_j.

• $Send(\ )$: This query simulates active attacks. In this query $\mathcal{A}$ can modify the transmitted messages between U_i and S_j. The oracle returns the corresponding response to $\mathcal{A}$.

• $EKReveal(\ )$: This query allows $\mathcal{A}$ to learn the session-specific ephemeral secrets held by the oracle, and the output should not have the long-term secret key of U_i or S_j.

• $SKReveal(\ )$: This query allows $\mathcal{A}$ to lean session key which created by oracle.

• $Corrupt(\ )$: This query shows the perfect forward secrecy of the session key on oracle; In this query, $\mathcal{A}$ can obtain long-term private key of U_i or S_j.

• $Expire(\ )$: This query clears the session key of a completed session created by oracle.

• $Test(\ )$: This query returns a session key or a random string.

After issuing the queries above, $\mathcal{A}$ outputs b', where b' is about the coin b produced in $Test(\ )$. $\mathcal{A}$ violates the authentication key agreement (AKA) of the proposed protocol, if $\mathcal{A}$ can guess b correctly. If $\Pr[Succ]$ denotes the probability that $\mathcal{A}$ violates the AKA of the proposed protocol, the advantage of $\mathcal{A}$ violates the AKA of the proposed protocol becomes $Adv(\mathcal{A}) = |2\Pr[Succ] - 1|$. The proposed protocol is secure under random oracles if $Adv(\mathcal{A}) < \epsilon$, where ϵ is an extremely small number.

Let $\mathcal{A}$ be a polynomial time adversary and he/she can make no more than q_s Send queries, q_e Excution queries, and q_h Hash queries. We have the advantage of $\mathcal{A}$ as follows.

$$Adv(\mathcal{A}) \leq \frac{O(q_s+q_e)^2}{2p} + \frac{O(q_h^2)}{2^n} + \frac{O(q_s)}{2^{n+p}} + O\left(q_h \cdot Adv_{CDH}(t')\right) \quad (1)$$

where $t' = O\left(t + (q_s + q_e)T_{\exp}\right)$ and T_{exp} notes the group exponentiation operation in bilinear pairing groups.

We define a sequence of games, starting with the real attack $Game_0$ to $Game_4$. For each game, we give an event $Succ_i$ denotes $\mathcal{A}$ successfully guessed b in the i_{th} instance of game.

$\mathbf{Game_0}$: This game shows the advantage that $\mathcal{A}$ violates the real game of the proposed protocol. From the definition, we have,

$$Adv(\mathcal{A}) = |2\Pr[Succ_0] - 1| \quad (2)$$

$\mathbf{Game_1}$: In this game, we simulate all the oracles for each query and keep lists to store the results of the oracles. L_H stores the results from H_i. $L_{\mathcal{A}}$ denotes $\mathcal{A}$ queries the random oracles. L_T denotes the transcript in this channel. $Game_0$ and $Game_1$ are indistinguishable. Therefore, we have,

$$\Pr[Succ_1] = \Pr[Succ_0] \quad (3)$$

$\mathbf{Game_2}$: In this game, we simulate all random oracles in $Game_1$, but avoid some collisions in the transcripts and hashes which is queried by $\mathcal{A}$. The $Game_1$ and $Game_2$ are indistinguishable unless the collisions of the group points and hash value occurred. According to the birthday paradox, we have,

$$|\Pr[Succ_2] - \Pr[Succ_1]| \leq \frac{O((q_s+q_e)^2)}{2p} + \frac{O(q_h^2)}{2^n} \quad (4)$$

$\mathbf{Game_3}$: In this game, we abort the process if $\mathcal{A}$ has been successfully guessing the value without the help of random oracle. This only happens in $Send$ queries. Therefore $Game_2$ and $Game_3$ are perfectly indistinguishable unless U_i rejects the response from $\mathcal{A}$. We get,

$$|\Pr[Succ_3] - \Pr[Succ_2]| \leq \frac{O(q_s)}{2^{n+p}} \quad (5)$$

$\mathbf{Game_4}$: In this game, we consider the session-key security. The purpose of this game is that $\mathcal{A}$ cannot obtain past session keys even if the secret information of U_i and S_j is leaked. The session key is computed as $H_4(g^{r_1 \cdot r_2})$. The advantage that $\mathcal{A}$ guesses session key correctly is equals to $\mathcal{C}$ break the CDH problem. Therefore $Game_3$ and $Game_4$ is indistinguishable if the CDH problem holds. So, we have

$$|\Pr[Succ_4] - \Pr[Succ_3]| \leq O\left(q_h \cdot Adv_{CDH}(t')\right). \quad (6)$$

In $Game_4$, all of the random oracles are simulated. The advantage of $\mathcal{A}$ guess b correctly is listed as follows.

$$\Pr[Succ_4] = \frac{1}{2} \quad (7)$$

Therefore, we get the equation as follows:

$$|\Pr[Succ_4] - \Pr[Succ_1]| \leq |\Pr[Succ_4] - \Pr[Succ_3]| + |\Pr[Succ_3] - \Pr[Succ_1]| \leq$$
$$|\Pr[Succ_4] - \Pr[Succ_3]| + |\Pr[Succ_3] - \Pr[Succ_2]| + |\Pr[Succ_2] - \Pr[Succ_1]| \quad (8)$$

$$|\Pr[Succ_1] - 1/2| \leq \frac{O(q_s+q_e)^2}{2p} + \frac{O(q_h^2)}{2^n} + \frac{O(q_s)}{2^n+p} + O\left(q_h \cdot Adv_{CDH}(t')\right) \quad (9)$$

Finally, we get equation $Adv(\mathcal{A}) \leq \frac{O(q_s+q_e)^2}{2p} + \frac{O(q_h^2)}{2^n} + \frac{O(q_s)}{2^n+p} + O\left(q_h \cdot Adv_{CDH}(t')\right)$.

6.1 Security experiment

We show the formal security analysis above, which can prove the theoretical security of the proposed protocol. In this part, we demonstrate that the proposed protocol passes security verification.

For formal security verification, we use the broadly accepted ProVerif tool that can explore the complete state space of security protocols. ProVerif uses the Dev-Yao model as the base model to simulate attacks on security protocols [Roy, Das, Chatterjee et al. (2019)]. First, we declare parameters, constants, open channels, functions, etc., which used in the proposed protocol. Second, we create two separate subprocesses to simulate the authentication process of the mobile user and the mobile service provider. Third, we set attackers that attack the proposed protocol. Finally, we execute the simulation by running ProVerif in our experiment environment. We show the result as follows.

1) RESULT not attacker(r1[]) is true.

2) RESULT not attacker(r2[]) is true.

3) RESULT not attacker(dSj[]) is true.

4) RESULT not attacker(dUi[]) is true.

5) RESULT event(evUserEndAuth(x)) ==> event(evUserEndParamGen(x)) is true.

6) RESULT inj-event(evUserEndAuth(x_21)) ==> event(evServerEndAuth(x_21)) is true.

7) RESULT inj-event(evServerEndAuth(x_22)) ==> event(evUserEndParamGen(x_22)) is true.

The above result shows that the adversary cannot get security parameters $\mathrm{r1}, \mathrm{r2}, \mathrm{d_{U_i}}, \mathrm{d_{S_j}}$, and he/she cannot pass the authentication verification. Therefore, the proposed protocol passes the security verification.

6.2 Security requirements comparison

In this part, we compare security requirements between the proposed protocol and other protocols in Tab. 4. For simplicity, we denote SR-1 to SR-11 as security requirements described in Tab. 4.

Table 4: Security requirements

Notation	Functionality	Odelu et al.	He et al.	Ours
SR-1	Single Registration	Yes	Yes	Yes
SR-2	Mutual Authentication	Yes	Yes	Yes
SR-3	User Anonymity	Yes	Yes	Yes
SR-4	Un-traceability	Yes	Yes	Yes
SR-5	Session Key Agreement	Yes	Yes	Yes
SR-6	Perfect Forward Secrecy	Yes	Yes	Yes
SR-7	Stolen Device Attack	Yes	Yes	Yes
SR-8	No Verifier Table	Yes	Yes	Yes
SR-9	No Online Registration Center	Yes	Yes	Yes
SR-10	The Resistance of Various Attacks	Yes	Yes	Yes
SR-11	Hierarchical Authentication	No	No	Yes

According to Tab. 4, both He et al. and Odelu et al.'s protocol cannot provide hierarchical authentication for limiting user access to service providers. In contrast, the proposed protocol can satisfy all eleven security requirements.

7 Performance comparison

In this part, we show the computation and communication costs of the proposed protocol. We will also compare its performance with other protocol. For the purpose of getting a trusted security level (1024-bit RSA algorithm), an Ate pairing: $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is used. $\mathbb{G}_1$ with order q is generated by a point on a super-singular elliptic curve $E(F_p): y^2 = x^3 + 1$ which is defined on the finite field F_p. Order q is a 160-bit prime number and p is a 512-bit prime number.

7.1 Computation cost comparison

We give the running time of various operations performed in the proposed protocol, and we compare the results with He and Odelu. In this section, we use the following notations for the following running times in this paper:

- T_{bp}: The running time of a bilinear pairing operation.
- T_{sm}: The running time of a scalar multiplication operation in $\mathbb{G}_1$.
- T_{mtp}: The running time of a map-to-point hash function in $\mathbb{G}_1$.
- T_{pa}: The running time of a point addition operation in $\mathbb{G}_1$.
- T_{exp}: The running time of an exponentiation operation in $\mathbb{G}_2$.
- T_{mul}: The running time of a multiplication operation in $\mathbb{G}_2$.
- T_h: The running time of a general hash operation.

The above operations are implemented with MIRACL library on a Samsung Galaxy S5 with Quad-core 2.45 GHz processor, 2-gigabyte memory running in Android 4.4.2, and a personal computer with I5-4460S 2.9 GHz processor, 4-gigabyte memory running in Windows 8. The running time of those operations is listed in Tab. 6.

Table 6: The running time of related operations (Millisecond)

	T_{mtp}	T_{bp}	T_{sm}	T_{pa}	T_{exp}	T_{mul}	T_h
User	33.582	32.713	13.405	0.081	2.249	0.008	0.056
Server	5.493	5.427	2.165	0.013	0.339	0.001	0.007

Based on the running time of the user and the service provider, we compare the computation costs of the proposed protocol with He and Odelu, the result described in Tab. 7 and Fig. 3.

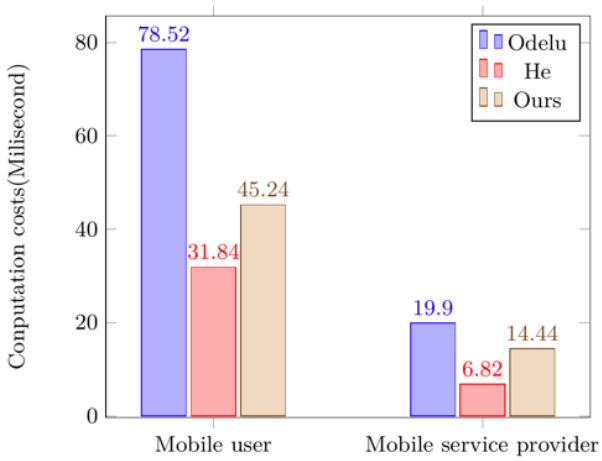

Figure 3: Computation cost comparison

Table 7: The running time of related operations (millisecond)

	Odelu et al.	He et al.	Ours
User	$T_{mtp} + 4T_h + 3T_{sm} + 2T_{exp} = 78.519$	$2T_{sm} + T_{pa} + 2T_{exp} + 8T_h = 31.837$	$3T_{sm} + 2T_{exp} + T_{pa} + 4T_h = 45.242$
Server	$T_{mtp} + 4T_h + 4T_{exp} + T_{sm} + T_{mul} + 2T_{bp} = 19.897$	$T_{bp} + 4T_{exp} + 2T_{mul} + 5T_h = 6.82$	$2T_{bp} + 4T_{exp} + T_{sm} + T_{pa} + T_{mul} + 6T_h = 14.44$

7.2 Communication cost comparison

According to the description of the trusted security level, q is a 160-bits prime number and p is a 512-bits prime number. The size of an element in $\mathbb{G}_1, \mathbb{G}_2$ is 1024 bits. The size of the hash function's output is 160 bits, identity parameter and expire parameter are both

32 bits. In our protocol, on the mobile user side, message C, E, F requires 320+320+96=736; on the mobile service provider side, message G, g_2 requires 160+512=672. The total communication cost is 1408 bits. The communication round comparison shows in Fig. 4. and communication cost comparison shows in Fig. 5.

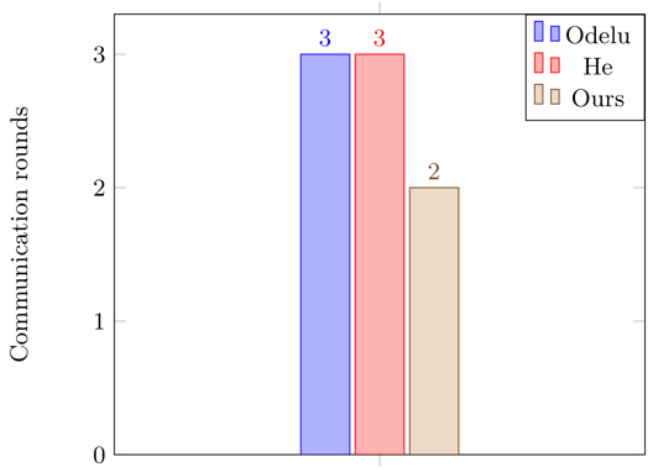

Figure 4: Communication round comparison

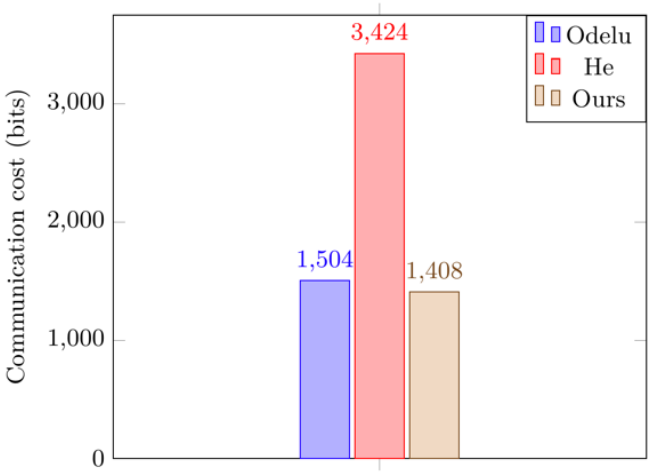

Figure 5: Communication cost comparison

8 Conclusion

In the past few years, many authentication protocols for mobile cloud computing have been proposed. However, they cannot provide hierarchical authentication functionality. This paper shows a novel authentication protocol with reasonable computation and communication costs and has implemented hierarchical authentication functionality, which restricts mobile user access to mobile service provider. The security proof has demonstrated that our protocol is provably secure. The computation and communication costs show the proposed protocol is suitable for the application of mobile cloud computing.

Funding Statement: This work is funded by the Chengdu Science and Technology Bureau No. 2016-XT00-00015-GX and the Civil Aviation Administration of China No. PSDSA201802.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Amin, R.; Kumar, N.; Biswas, G. P.; Iqbal, R.; Chang, V. (2018): A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment. *Future Generation Computer Systems—the International Journal of Escience*, vol. 78, no. 3, pp. 1005-1019.

Canetti, R.; Krawczyk, H. (2001): Analysis of key-exchange protocols and their use for building secure channels. *Advances in Cryptology-Eurocrypt*. Springer, Berlin, Heidelberg.

Chandrakar, P.; Om, H. (2017): A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ECC. *Computer Communications*, vol. 110, no. 15, pp. 26-34.

Chatterjee, S.; Roy, S.; Das, A. K.; Chattopadhyay, S.; Kumar, N. et al. (2018): Secure biometric-based authentication scheme using Chebyshev chaotic map for multi-server environment. *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 824-839.

Choi, K.; Hwang, J.; Lee, D.; Seo, I. (2005): ID-based authenticated key agreement for low-power mobile devices. *Information Security and Privacy*, vol. 3574, no. 12, pp. 494-505.

Dodis, Y.; Reyzin, L.; Smith, A. (2004): Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. *Advances in Cryptology-Eurocrypt*, pp. 523-540.

Fang, L.; Li, Y.; Yun, X.; Wen, Z.; Ji, S. et al. (2019): A novel authentication scheme to prevent multiple attacks in SDN-based IoT Network. *IEEE Internet of Things Journal*, DOI: 10.1109/JIOT.2019.2944301.

Feng, Q.; He, D.; Zeadally, S.; Wang, H. (2018): Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment. *Future Generation Computer Systems—the International Journal of Escience*, vol. 84, pp. 239-251.

Gopinath, V.; Bhuvaneswaran, R. S. (2018): Design of ECC based secured cloud storage mechanism for transaction rich applications. *Computers, Materials & Continua*, vol.57, no. 2, pp. 341-352.

He, D.; Wang, D. (2015): Robust biometrics-based authentication scheme for multiserver environment. *IEEE Systems Journal*, vol. 9, no. 3, pp. 816-823.

He, D.; Zeadally, S.; Kumar, N.; Wu, W. (2016): Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server

architectures. *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2052-2064.

Jiang, Q.; Ma, J.; Wei, F. (2018): On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services. *IEEE Systems Journal*, vol. 12, no. 2, pp. 2039-2042.

Jiang, X; Liu, M.; Yang, C.; Liu, Y.; Wang, R. (2019): A blockchain-based authentication protocol for WLAN mesh security access, *Computers, Materials & Continua*, vol. 58, no. 1, pp. 45-59.

Kou, L.; Shi, Y.; Zhang, L.; Liu, D.; Yang, Q. (2019): A lightweight three-factor user authentication protocol for the information perception of IoT, *Computers, Materials & Continua*, vol. 58, no. 2, pp. 545-565.

Kumari, S.; Das, A. K.; Li, X.; Wu, F.; Khan, M. K. et al. (2018): A provably secure biometrics-based authenticated key agreement scheme for multi-server environments. *Multimedia Tools and Applications*, vol. 77, no. 2, pp. 2359-2389.

Lamport, L. (1981): Password authentication with insecure communication. *Communications of the ACM*, vol. 24, no. 11, pp. 770-772.

Li, H.; Dai, Y.; Tian, L.; Yang, H. (2009): Identity-based authentication for cloud computing. *Cloud Computing*. Springer, Berlin, Heidelberg.

Miracle. (2015): *BWorld Robot Control Software.* https://github.com/miracl/MIRACL.

Odelu, V.; Das, A. K.; Goswami, A. (2015): A secure biometrics-based multi-server authentication protocol using smart cards. *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953-1966.

Odelu, V.; Das, A. K.; Kumari, S.; Huang, X.; Wazid, M. (2017): Provably secure authenticated key agreement scheme for distributed mobile cloud computing services. *Future Generation Computer Systems—the International Journal of Escience*, vol. 68, pp. 74-88.

Roy, S.; Das, A. K.; Chatterjee, S.; Kumar, S.; Chattopadhyay, S. et al. (2019): Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing based healthcare applications. *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 457-468.

Shen, J.; Tan, H.; Wang, J.; Wang, J.; Lee, S. (2015): A novel routing protocol providing good transmission reliability in underwater sensor networks. *Journal of Internet Technology*, vol. 16, no. 1, pp. 171-178.

Silver, L. (2019): Smartphone ownership is growing rapidly around the world, but not always equally. https://www.pewresearch.org/global/2019/02/05/smartphone-ownership-is-growing-rapidly-around-the-world-but-not-always-equally/.

Tsai, J.; Lo, N. (2015): A privacy-aware authentication scheme for distributed mobile cloud computing services. *IEEE Systems Journal*, vol. 9, no. 3, pp. 805-815.

Tseng, Y. M.; Huang, S. S.; Tsai, T. T.; Ke, J. H. (2016): List-free id-based mutual authentication and key agreement protocol for multi-server architectures. *IEEE Transaction on Emerging Topics in Computing*, vol. 4, no. 1, pp. 102-112.

Xie, Q.; Wong, D. S.; Wang, G. L.; Tan, X.; Chen, K. F. et al. (2017): 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.

Xiong, L.; Peng, D.; Peng, T.; Liang, H. (2017a): An enhanced privacy-aware authentication scheme for distributed mobile cloud computing services. *KSII Transactions on Internet and Information Systems*, vol. 11, no. 12, pp. 6169-6187.

Xiong, L.; Peng, D. Y.; Peng, T.; Liang, H. B.; Liu, Z. C. (2017b): A lightweight anonymous authentication protocol with perfect forward secrecy for wireless sensor networks. *Sensors*, vol. 17, no. 11, pp. 1-28.

Xu, G.; Qiu, S.; Ahmad, H.; Xu, G.; Guo, Y. et al. (2018): A multi-server two-factor authentication scheme with un-traceability using elliptic curve cryptography. *Sensors*, vol. 18, no. 7, pp. 1-19.

Yoon, E. J.; Yoo, K. Y. (2013): Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. *Journal of Supercomputing*, vol. 63, no. 1, pp. 235-255.