Hindawi
Security and Communication Networks
Volume 2019, Article ID 5141395, 11 pages
https://doi.org/10.1155/2019/5141395

Research Article

New Authentication Scheme to Secure against the Phishing Attack in the Mobile Cloud Computing

Munivel E and Kannammal A

Department of Electronics and Communication Engineering, PSG College of Technology, Coimbatore, India

Correspondence should be addressed to Munivel E; mailtomunivel@gmail.com

Received 13 December 2018; Revised 11 March 2019; Accepted 7 April 2019; Published 8 May 2019

Academic Editor: Stelvio Cimato

A phishing attack is one of the severe threats to the smartphone users. As per the recent lookout report, mobile phishing attack is increasing 85% year to year and going to become a significant threat to the smartphone users. This social engineering attack attempts to get the user's password by disguising as trusted service provider. Most of the smartphone users are using the Internet services outside of the traditional firewall. Cloud-based documents are one of the primary targets of this phishing attack in mobile cloud computing. Also, most smartphone users are using the cloud storage in their device. To secure against this password attack in a mobile cloud environment, we propose a new authentication scheme to provide novel security to the mobile cloud services. This scheme will verify the user and service provider without transmitting the password using the Zero-knowledge proof based authentication protocol. Moreover, the proposed scheme will provide mutual authentication between the communication entities. The effectiveness of proposed scheme would be verified using protocol verification tool called Scyther.

1. Introduction

Mobile cloud is a hybrid computing technology, which combines the advantages of cloud computing and the cellular technology to develop new paradigm called mobile cloud computing (MCC) [1]. Figure 1 shows the general view of mobile cloud computing, MCC is the technology will help to exceed the hardware limitation like computation, storage, and networking in the end-user mobile devices [2–4].

Authentication is one of the critical security challenges in mobile cloud environment. Authentication is an approach to verify the originality of user identity. In mobile cloud computing, user identity can be verified using mobile device and/or one or more other authentication approaches. In the recent scenario, maximum protocols are sharing or sending the password in the form of hash value or the encrypted form to the verifier or the authentication server [1, 5–7]. The transmitting password can get captured by the intruder. Hence, this will encourage the phishers to develop fake website or service to capture the user password.

The objective of this paper is not to send the user password to the authentication server or cloud service providers during any stage of communication process. Hence, this paper aims to not allow delivering the user password out of end-user device, even to the trusted third party.

1.1. Related Work. Authentication is an essential security service in any system or network communications [8–10]. It is classified as user authentication, remote authentication, mutual authentication, message authentication, and implicit authentication [11, 12]. The current authentication review shows the different attributes, based on password, hash value, identity, digital signature, hierarchical model, mobile number, group key, and biometric [13].

In 1981, Lamport et al. [3] proposed an authentication scheme to send the hash value of the password, instead of a real password to a remote server to verify the authentication process. In 2014, Chaurasia et al. [7] introduced an authentication as a service in cloud environment. In this method, Chaurasia et al. [7] uses two-factor authentication scheme to verify the users between different group of services. Also, the actual user identity is not sharing, instead shares the hash value of user identity between the communication entities. Recent years cloud and mobile computing gradually developed with the help of latest wireless technologies.

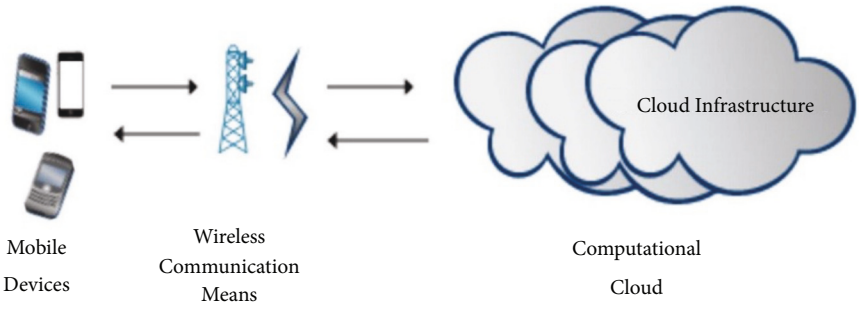

Figure 1: General view of mobile cloud computing.

Hence, in 2017, Roy et al. [14] proposed a new method to use mobile-based authentication in cloud computing. In this scheme, Roy et al. [14] introduced universal subscriber identity module (USIM) based identity verification method. This scheme used USIM as a primary identity to initiate the authentication process. However, when the mobile device gets stolen, authentication will get disabled, and the entire process will get revoked [10]. In 2012, Grzonkowski et al. [15] introduced improved authentication scheme based on the smart card based authentication protocol. This method entirely depends on the smart card generator (SCG). SCG working as a trusted third party and also this scheme was using the secure channel to share the session between the communication entities.

In 2013, Mohil et al. [16] proposed a scheme based on PIN number and the preconfigured voice prints to verify the identity of authentication user. However, this method is proved to use more computation. Hence, it is not useful due to more computation and power usage in a mobile device.

In 2015, Lin et al. [12] introduced secure method in the smart learning application in the cloud environment. This scheme registers the user with original user ID in the authentication server (AS). This scheme sends the hash value of password to the authentication server in the encrypted form. The AS decrypts and can get the hash value of the password. This scheme was secure against the man-in-the-middle attack, but not safe against the phishing attack due to password sharing between the communication entities.

In 2016, Kalra et al. [17] and Huang et al. [18] proposed strong authentication based one-time password (OTP) and Message Digest value. This [17] scheme uses USIM with a secure channel to share the user identity. Hence, this scheme is not defined when the mobile device is missed or stolen. This [18] scheme uses the traditional password to verify the authentication phase, but the chance of cracking password in the server side. Hence, this is prone to phishing attack by the server side.

Dynamic identity-based authentication technique is proposed by Li et al. [19] to secure the user identity. In this technique, real user identity is mapped with new dynamic identity in every communication. But, Li et al. [19] did not concentrate on the password security. Still user password is sharing as a hash value to the remote server in the registration phase. Also, in Stage II authentication, received hash value of the user is verified with the stored hash value from the remote server as Ramport et al. [3] authentication. Hence, in this scheme, the user password may prone to crack by the remote server.

The new secure authentication was proposed by Zhou et al. [20] using the smart card generator. But, cloud service providers using the master key to verify the user and data owner. The initial authentication is based on validating the hash value of identity and password. This scheme is not sharing the identification of the communicating entities in all the stage of authentication. Also, the intruder may disturb the communication with fake hash value to make null every time to consume computation in a mobile device. All these findings may be prone to phishing attack along with replay attack and man-in-the-middle attack in a mobile device.

To achieve mutual authentication in mobile cloud computing Grzonkowski et al. [21], He et al. [22], and Miler et al. [23] are proposed different authentication protocols in the mobile cloud service environment. According to the Miler et al. [23] scheme, the user ID is sharing using the secure channel, but the SCG generates the public key of the user and sends along with the randomly generated nonce to secure against the replay attack. However, the session key is not encrypting or not sending over a secure channel. Authentication phase not carrying the sender and receivers ID along with the session key. Hence, Miler et al. [23] scheme prone to man-in-the middle attack and phishing attack.

Smart card based or the trusted third party based authentications are the most common technique to prevent illegal access in an insecure mobile cloud environment [6]. Many authentication protocols proposed [13, 18, 24–27] to verify the originality of end user. However, most of these protocols may not be satisfying the security against a phishing attack.

Phishing attack is an essential problem in the current generation of mobile cloud authentication services [22]. Hence, to improve the security as noted earlier, mobile cloud authentication systems are vulnerable to various types of security attacks. Such attacks do not only affect the user's identity, but also affect the device performance [28].

In this paper, we endeavour to progress the mobile cloud computing security by introducing new authentication scheme based on Zero-knowledge proof technique.

The proposed scheme is aimed to secure against the replay, man-in-the-middle, denial of service, server-side spoofing and phishing attacks, Malicious Insider, and other generic attacks in the mobile cloud environment without

sharing the real username and password to the any of communication entities.

1.2. Our Contribution. In the paper, we present the outline of our proposed authentication protocol. To achieve the security against the phishing attack, we are not going to transfer the actual password to the authentication server or cloud service providers or any other communication entities during the registration and authentication stage. Here summarizing the significant contributions of our paper as follows.

First, we review the He et al. [22] scheme in mobile cloud computing. In some cases, this scheme is compromising the phishing attack. Moreover, also we show that this scheme is not entirely satisfying the user anonymity. Hash value of the user password knew by the trusted third party or the authentication server.

Second, we propose a new authentication scheme to secure against phishing attack without sharing the real username and password to the authentication server and cloud service providers. Moreover, the new scheme supports the mutual authentication with Zero-knowledge of proof.

Finally, we provide detail security verification methods to prove our proposed scheme is secure and efficient and also meets the requirements of mobile cloud services.

1.3. Organization of the Paper. This paper is organized into six sections; the second section presents the preliminaries of proposed authentication scheme. The third section reviews the brief He et al. [22] scheme and presents its security problems. The fourth section presents the details of proposed mobile cloud authentication scheme with different phases like initial registration, user registration, and authentication phases. The fifth section presents the analysis of proposed security scheme nonformal verification. The sixth section compares with similar schemes, list the code, and display the formal verification result by the Scyther. And the last seventh section explains the performance analysis of the proposed scheme and its efficiency.

2. Preliminaries

2.1. Zero-Knowledge Proof. The Zero-knowledge protocol is a method based proof of verifying the originality of the prover without disclosing further knowledge about the prover to the verifier [23]. The Zero-knowledge protocol is based on Zero-knowledge proofs and can be classified as interactive Zero-knowledge and noninteractive Zero-knowledge based on the working methods [24]. The interactive Zero-knowledge protocol uses multiple authentication steps of communications between the prover and verifier. The noninteractive Zero-knowledge protocol uses only one communication message called proof between the prover and verifier [24]. The properties of Zero-knowledge proof can be distinguished as follows.

(i) Completeness. "If the requested statement is correct, the honest verifier will prove that the requested statement is true to the honest verifier".

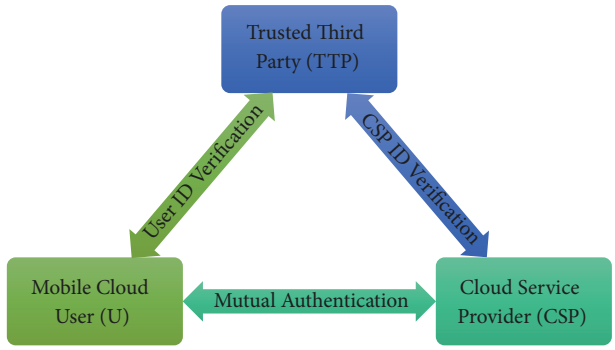

Figure 2: System model.

(ii) Soundness. "If the requested statement is false, there is no way to fake the result to the verifier that the requested statement is true".

(iii) Zero-Knowledge. "If the requested statement is right, the verifier may not know anything about the prover other than that the requested statement is true".

2.2. System Model. A typical authentication network model of the proposed mobile cloud scheme is shown in Figure 2. Here, we are using three participants in the proposed scheme.

(i) Cloud User (U). He/she is a mobile cloud user. He/she is registering as a new user with the TTP using one-time password to confirm the original identity. Then the user uses its user ID and password to generate the public key with using mobile application and then sends the mobile number, user ID, and public key with client URL to the TTP.

(ii) Trusted Third Party (TTP). TTP is working as authentication server (AS), responsible for verifying requested user and the cloud service provider (CSP). After initial verification, it is receiving the public key from the cloud user.

(iii) Cloud Service Provider (CSP). CSP provides services like storage, computation, and communication service to the mobile cloud user. It verifies the user request with its URI. If URI is on the approved list, it will ask the TTP for verification. Then TTP verifies the mobile number and the user ID. Finally the user ID, TTP nonce, and public key will send to CSP to confirm the cloud user.

3. Review of He et al. Scheme (2017)

This section describes the privacy aware authentication scheme in mobile cloud environment proposed by [22]. This protocol is developed based on identity-based signature scheme partially and also this scheme includes three phases: (1) system setup phase, (2) registration phase, and (3) authentication phase. Notations used in this protocol scheme are listed in Table 1.

Table 1: Notation and Description.

Notation	Description
‖	Concatenation
⊕	XOR Operation
h(U)	Hash Value of User ID
h(PW)	Hash Value of Password
U	Mobile Cloud User
S	Cloud Service Provider
AS	Authentication Server
Uid	Client URI with Mobile No
sk(U)	Private Key of User
pk(U)	Public Key of User
sk(S)	Private Key of CSP
pk(S)	Public Key of CSP
Ns	Fresh Authentication ID
R	Random Value Gen by User
Na, Nu1, Nu2, Nu3	Fresh Nonce
h()	Hash Function

3.1. System Setup Phase. Smart card generator (SCG) is a trusted third party (TTP) in this scheme. SCG is generating its private and public keys using bilinear pairing.

(1) Smart card generator selects a random nonce s as a master key.

(2) Smart card generator generates the public key K+SCG based on the master key s.

(3) SCG selects five hash values based on the group G1 and G2.

(4) Finally it publishes its parameter by using public key and its hash values and also saves its secret key s.

3.2. Registration Phase. In this phase, user U and the cloud service provider (CSP), registering with the SCG to get their private key through the following steps over a secure channel.

(1) User U sends his user ID to the SCG.

(2) SCG generates the user's private key by using its master key I and sends the private key K–U to the requested user through the secure channel.

(3) CSP sends its ID to SCG through the secure channel.

(4) SCG generates the CSP private key by using its master key s and sends the private key K–CSP to the requested user through the secure channel.

3.3. Authentication Phase

Step 1. User enters a password only, and does not enter a username or user ID. However, client device calculates the hash value of user ID and password. Moreover, encrypts by using its session key and, finally, the user sends to the cloud service provider as follows:

$$U \longrightarrow CSP : \left\{ID_{U_i}, PW_{U_i}\right\}_{K_{Su_i}} \quad (1)$$

Step 2. Cloud service provider (CSP) selects a random nonce a and computes A with nonce a and prime P, and A sends to the user.

$$CSP \longrightarrow U : \{A\} \quad (2)$$

Step 3. In this step, user Ui selects random nonce

$$b, r \in Z_q^* \quad (3)$$

and computes B as follows:

$$B = g^b \quad (4)$$

The session key K_{ij} and other functions are computed as follows:

$$K_{ij} = h_2\left(A \parallel B \parallel A^b\right) \quad (5)$$

$$K_2 = b.\left(P_{pub} + h_1\left(ID_{CSP_j}\right).P\right) \quad (6)$$

$$R = g^r \quad (7)$$

$$\omega_{U_i} = h_3\left(ID_{U_i} \parallel ID_{CSP_j} \parallel A \parallel B \parallel K_{ij} \parallel K_2 \parallel R\right) \quad (8)$$

$$\Xi_{U_i} = \left(r + \omega_{U_i}\right) S_{U_i} \quad (9)$$

$$C_i = h_4\left(B\right) \oplus \left(ID_{U_i} \parallel \omega_{U_i} \parallel \Xi_{U_i}\right) \quad (10)$$

User U_i sends K_2, C_i to CSP_j

$$U \longrightarrow CSP : \{K_2, C_i\} \quad (11)$$

Cloud service provider CSP_j computes session key and other functions B, K_{ij}, X and B, K_{ij}, XR as follows:

$$B = \left\{K_2, S_{CSP_j}\right\} \quad (12)$$

$$K_{ij} = h_2\left(A \parallel B \parallel B^a\right) \quad (13)$$

$$X = \left(ID_{U_i} \parallel \omega_{U_i} \parallel \Xi_{U_i}\right) \quad (14)$$

$$Y = h_4\left(B\right) \oplus C \quad (15)$$

$$X = Y \quad (16)$$

$$\left(ID_{U_i} \parallel \omega_{U_i} \parallel \Xi_{U_i}\right) = h_4\left(B\right) \oplus C \quad (17)$$

$$R = \left\{\Xi_{U_i}, P_{pub} + h_1\left(ID_{ui}\right).P\right\}.g^{-\omega_{U_i}} \quad (18)$$

Cloud service provider CSP_j verifies the following:

$$\omega_{U_i} = h_3\left(ID_{U_i} \parallel ID_{csp_j} \parallel A \parallel B \parallel K_{ij} \parallel K_2 \parallel R\right) \quad (19)$$

If not matching, CSP_j rejects the service request, or else cloud service provider CSP_j computes D_i as follows:

$$D_i = h_4\left(ID_{CSP_j} \parallel ID_{U_i} \parallel A \parallel K_{ji} \parallel K_2 \parallel B\right) \quad (20)$$

Finally CSP_j sends D_i to U_i

$$CSP \longrightarrow U : \{D_i\} \quad (21)$$

U_i Checks whether D_i is equal to $h4(ID_{CSPj} \parallel ID_{Ui} \parallel A \parallel K_{ji} \parallel K_2 \parallel B)$ and U_i confirm the CSP_j. Else U_i terminates the service.

3.4. Analysis of He et al. Scheme. In the login phase of this [22] scheme, the user enters the only username. As per (1) user encrypts the hash value username and password. However, this step does not have a precise definition of how the username is taken and may be taken from the secure memory of the mobile device.

The first finding is, if the user is taking the encrypted hash value from the mobile device secure memory, then how this scheme will resist the stolen mobile device attack, because it does not have any method to verify that the username and password are entered by the user or malicious software.

The second finding is that, in the login phase, the user sends the encrypted hash values of username and password to the cloud service provider (CSP), to verify the sender and the receiver, not mentioning anything in the communication. This may be prone to man-in-the-middle attack.

$$U \longrightarrow CSP : \left\{ID_{U_i}, PW_{U_i}\right\}_{K_{Su_i}} \tag{22}$$

The third finding is that this scheme allows sending the password to the cloud service provider. Once the cloud service provider decrypts, it can know the hash value and may try to crack the password. Hence, the phishing attack remains open in this scheme.

This [22] scheme uses smart card generator as trusted third party in the initial phase. However, it is not using the trusted party to ensure the mobile user and the cloud service provider in the authentication phase. The resistance of spoofed client attack and spoofed cloud service provider attack are unanswered in this scheme.

4. Proposed Scheme

The proposed authentication scheme has three phases. The first phase creates a group called G and its members. TTP shares the elements of the group to the communication entities. The second phase handles the registration of cloud user and CSP with the authentication server or a trusted third party. The third phase verifies the cloud user and the service provider to achieve the mutual authentication.

4.1. Initial Registration. The given group G is having set of values. G0, G1 are carrier set of random elements of group G [2, 23, 24, 29]. Hence, the public key may be the G, G0.

Group G is a carrier set cordiality of the order of Group $|G|$. Hence the digital payment like Bitcoin which uses the sec256k1 group, based on elliptic curve. Element size of this group is 256-bit strings, which is very hard in this type of group [21].

4.2. Registration Phase. The second phase accepts the registration of cloud user and cloud service provider by the authentication server.

The new user generates a request with the authentication server (AS) with its mobile number being of original identity. AS verifies the available list of available registered mobile numbers. If the number is new, the AS sends the OTP or else terminates the communication. The entered OTP will get verified with AS. Once OTP is verified, then the user enters the new user ID and password; the mobile browser generates the hash value of the user ID as H1 and generates a hash value of password as H2. Client browser generates the public key P using the hash value of the user password H2. Finally, the user sends the hash value of user ID H1 and public key P along with mobile number (as Client URL) to register in the trusted user list of AS over the secure channel as explained in Figure 3. All the above communications are happening over the secure channel between user and AS. In Figure 4, cloud service provider, service and domain registration with authentication server (TTP), and new cloud service provider (CSP) generate a new request to AS. AS verifies the existence of the new domain in the existing list. If free, the AS accepts the request and generates the domain tag with the new unique one-time key. Domain tag sends to the CSP. Moreover, the CSP has to keep the tag in the document root of its domain and verifies with the AS. If AS verifies the domain tag, accepts the registration request, and stores the hash value of domain's URI in the trusted list and shares the CSP public key to AS, all of the above communications are happening over the secure channel between AS and CSP.

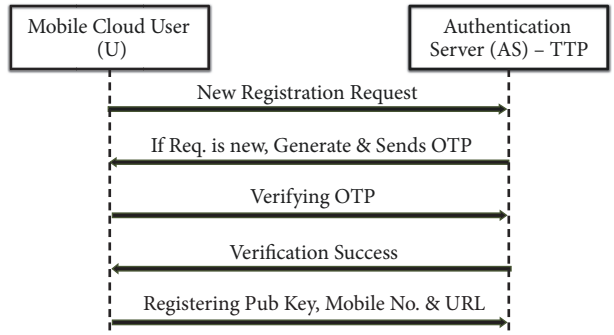

Figure 3: User registration with authentication server (TTP).

4.3. Authentication Phase. In this phase, authentication server (AS), mobile cloud user (U), and cloud service provider (CSP) are participating in verifying the user and CSP through the AS to achieve mutual authentication without revealing the real password between the communication entities. The proposed authentication is using the mobile number as an original identity to register the user. Once the user is registered successfully as per Section 4.2, the AS will generate the unique link to the client as follows. User mobile number is 9xxx7; when this user registers with the authentication server myauth.in, they will get unique web URI called client URI as 9xxx7.myauth.in. The AS will maintain the mobile number, client URI, hash value of user ID, and public key of user as client profile like Table 2.

User U sends the service request to CSP with his/her mobile identity to the Cloud service provider (CSP).

Once the user's public key is shared from the AS to CSP after verifying the user and CSP as per the communication explained in Figure 5, second stage of protocol will work as per the following steps of equation and also is explained overall in Figure 6.

Table 2: Sample Client Profile in AS.

Sl. No	Mobile No	Hash value	Pub Key	URI	Allowed Services
1	9xxx7	0A 04 1B 94	User Pub-Key	9xxx7.myauth.in	https://mobilecloudsr1.s3.amazonaws.com

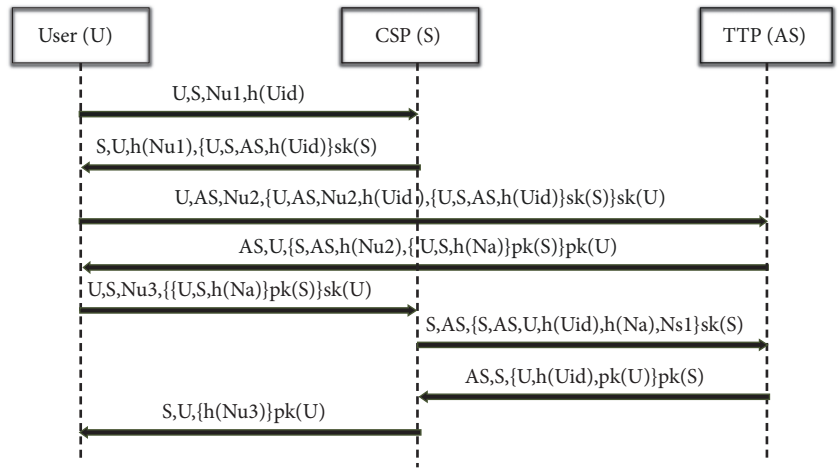

Figure 4: Cloud service provider and service and domain registration with authentication server (TTP).

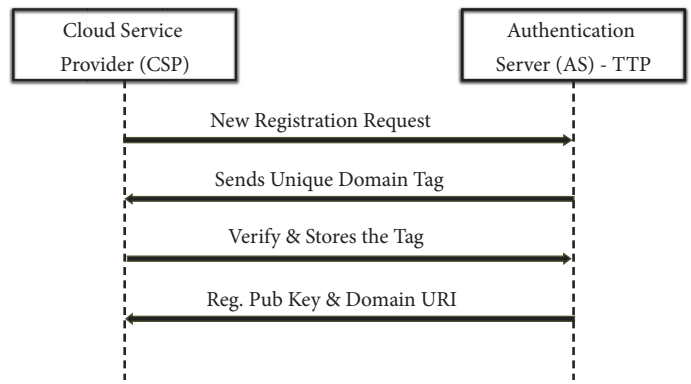

Figure 5: Proposed Stage I authentication protocol.

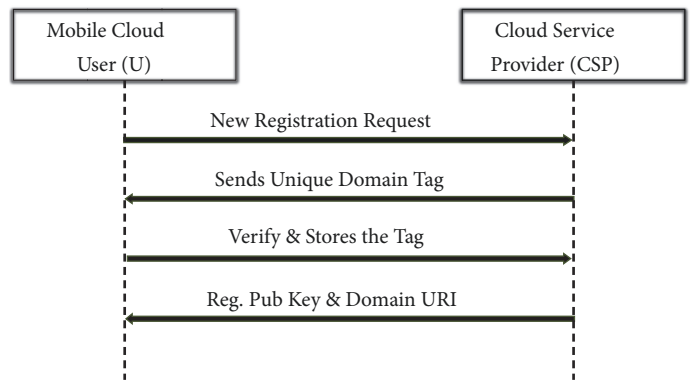

Figure 6: Proposed Stage II authentication protocol against phishing attack.

Step 1. User U requests the service or visits the service. Hence, the user enters the login and clicks the register button:

$$U- > S : U, S, \mathrm{Re}q \tag{23}$$

Step 2. Cloud service provider S sends the random token Ns as authentication ID to the user's request after verifying the client's URI.

$$S- > U : S, U, Ns \tag{24}$$

Step 3. User enters user ID and password in the client browser. Browser plug-in generates the hash value of user ID and password. Based on this hash values, the browser generates the value x as follows. Hence, the password is not leaving the client browser.

$$x = h\left(PW_u\right) \tag{25}$$

Then the user computes Pu (public key of user U) with using x and the shared group g_0:

$$Pu = g_0^x \tag{26}$$

Then the user generates the random value $r \in g$ and calculates Q:

$$Q = {g_0}^{r_x} \tag{27}$$

By using Q, user calculates the value C and Zx as follows:

$$C = h\left(Pu, Q, Ns\right) \tag{28}$$

$$Zx = Rx - Cx \tag{29}$$

Finally, the user sends the C and Zx to the server.

$$U- > S : C, Zx \tag{30}$$

The server S calculates the value Q as follows:

(1) Server receives C and Zx.

(2) The server has the users Ns, public key Pu, and shared group element g_0.

The server calculates Q:

$$Q = {Pu}^C {g_0}^{Zx} \tag{31}$$

Then the server S checks whether the $C = h(Pu, Q, Ns)$.

In this proposed protocol the random value is generated by the user, but this value is constructed by the server S with using above functions as follows.

As per (27) and (29), $Q = {g_0}^{r_x}$ and $Zx = Rx - Cx$.

We can prove the following with using simple substitution: (27) $Q = {g_0}^{r_x}$ and (31), $Q = {Pu}^C {g_0}^{Zx}$

Hence, ${g_0}^{r_x} = {Pu}^C {g_0}^{Zx}$,

$${g_0}^{r_x} = \left({g_0}^x\right)^c {g_0}^{(rx-cx)} \tag{32}$$

$${g_0}^{r_x} = {g_0}^{cx} {g_0}^{rx-cx} \tag{33}$$

$${g_0}^{r_x} = {g_0}^{cx+rx-cx} \tag{34}$$

$${g_0}^{r_x} = {g_0}^{rx} \tag{35}$$

Now user's random value r is constructed by the server S to verify that the user is genuine and also user is proved that the server's random value Ns is known by the user to achieve the mutual authentication.

5. Security Analysis and Verification

The proposed scheme is nonformally proved to resist against the significant attacks like phishing attack, replay attack, impersonation attack and other generic attacks explained in the following subsections.

5.1. Phishing Attack. The proposed authentication scheme is not sending the password to the server. It is generating the public key by using the hash value of the password. As explained in (25) to (29), we compute the C and Zx values and send to the CSP. In the CSP side C and Zx construct and verify the user identity with available public key. Hence, this scheme is resistance against the phishing attack [22, 28].

5.2. Strong Replay Attack. In this proposed scheme, nonce Nu1, Nu2, Nu3, and Na are used to check that the communication is fresh in both the stages of authentication as well as shown in Figures 5 and 6. The Ns is used as fresh authentication ID in Stage I authentication. For an example, in our Stage I authentication, user U sends Nu1 (nonce to avoid replay attack) and the hash value of actual user-identity h(Uid) (to avoid user anonymity attack) to the server S as (U,S,Nu1,h(Uid)) an authentication request to avoid the replay and user anonymity.

5.3. Server Impersonation Attack. The proposed scheme is resistance against the user impersonation attack. Not only is this scheme using the user ID but also it is using the user profile as URI, which includes the mobile number, and URI or user like 9xxx7.myauth.in. When a request comes from any user, the server verifies the client URI first. If the user URI is correct, then only the server accepts the request for impersonation and masquerade attack [10].

5.4. Generic Attacks. Also, the proposed scheme satisfies the generic attack like denial of service attack by specifying the active participants of each communication [21]. Man-in-the-middle attack is satisfying by every communication which is carrying the actual sender and the receiver identity [11, 15]. It is carrying the server URI and the client URI in every step and also its getting verified by the server using the user's profile.

5.5. Forward Secrecy and Mutual Authentication. The proposed scheme is using the asymmetric key cryptosystem to verify the communication entities to maintain the forward secrecy and to achieve the mutual authentication [18, 20, 23, 30]. Even the key is generated by the end user to avoid the server impersonation attack (U,S,Nu3,U,S,h(Na)pk(S)sk(U)).

6. Comparison and Formal Verification

In this section, we compare the proposed authentication schemes with significant security protocols listed in Table 3, and also the formal verification is done with Scyther [18, 31] to prove Stage I is secure against the significant attacks. Finally, Stage II authentication also is verified and listed.

Table 3: Comparing Resistance of Attacks.

Sl. No	Scheme	[7]	[1]	[2]	[8]	[9]	[10]	[11]	[12]	[13]	[15]	Ours
1	Formal Security Proof	No	No	Yes	No	No	No	Yes	Yes	Yes	No	Yes
2	Forward Secrecy	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
3	Login Phase Efficiency	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
4	Resistant to DoS Attack	Yes	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
5	Resistant to Password Guessing Attack	No	No	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes
6	Resistant to Phishing Attack	No	No	No	No	No	No	No	No	No	Yes	Yes
7	Resistant to Privileged Insider Attack	Yes	No	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
8	Resistant to Server Impersonation Attack	No	Yes	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
9	Resistant to Stolen Mobile Device Attack	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes
10	Resistant to Strong Reply Attack	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
11	Resistant to Strong User Anonymity	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
12	Resistant to User Impersonation Attack	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
13	Secure Mutual Authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
14	Verification using Scyther or Other Tools	No	No	No	No	No	No	Yes	No	No	No	Yes

6.1. Formal Verification. The proposed authentication protocol is verified using the automated protocol verification tool called Scyther, developed by Cremers et al. [31]. Scyther is having the features like unbounded verification, attack finding, and visualisation, also supporting the classical properties like secrecy, agreement, aliveness, and synchronisation [29, 31]. The proposed protocol can be written in security protocol description language. Mutual authentication between the user, AS, and CSP is verified in Scyther as follows.

```
/* Stage I Authentication Protocol */
const Fresh: Function;
const hash: Function;
hashfunction h;
const pk: Function;
const sk: Function;
inversekeys (pk,sk);
const h,Uid;
protocol KeyShareProto (U,S,AS)
{
role U
{
fresh Nu1,Nu2,Nu3: Nonce;
fresh Ns1,Na:Nonce;

send_1 (U,S,Nu1,h(Uid));
recv_2 (S,U,h(Nu1),{U,S,AS,h(Uid)}sk(S));
send_3 (U,AS,Nu2,{U,AS,Nu2,h(Uid),{U,S,AS,
h(Uid)}sk(S)}sk(U));
recv_4 (AS,U,{S,AS,h(Nu2),{U,S,
h(Na)}pk(S)}pk(U));
send_5 (U,S,Nu3,{{U,S,h(Na)}pk(S)}sk(U));
recv_8 (S,U,{h(Nu3)}pk(U));
}
role S
{
fresh Nu1,Nu2,Nu3: Nonce;
fresh Ns1,Na:Nonce;
recv_1 (U,S,Nu1,h(Uid));
send_2 (S,U,h(Nu1),{U,S,AS,h(Uid)}sk(S));
recv_5 (U,S,Nu3,{{U,S,h(Na)}pk(S)}sk(U));
send_6 (S,AS,{S,AS,U,h(Uid),h(Na),Ns1}sk(S));
recv_7 (AS,S,{U,h(Uid),pk(U)}pk(S));
send_8 (S,U,{h(Nu3)}pk(U));
}
role AS
{
fresh Nu1,Nu2,Nu3: Nonce;
fresh Ns1,Na:Nonce;

recv_3 (U,AS,Nu2,{U,AS,Nu2,h(Uid),{U,S,AS,
h(Uid)}sk(S)}sk(U));
send_4 (AS,U,{S,AS,h(Nu2),{U,S,
h(Na)}pk(S)}pk(U));
recv_6 (S,AS,{S,AS,U,h(Uid),h(Na),Ns1}sk(S));
send_7 (AS,S,{U,h(Uid),pk(U)}pk(S));
}
}
```

Our Stage I authentication protocol is verified by Scyther. The source code of Stage I is listed above and the output is shown in Figure 7.

```
/* Stage II Authentication Protocol */
const Fresh: Function;

protocol AuthProto (U,S)
{
role U
{
fresh Ns:Nonce;
const Req;
const C;
const Zx;

send_1 (U,S,Req);
recv_2 (S,U,Ns);
send_3 (U,S,C,Zx);
claim(U,Secret,C);
```

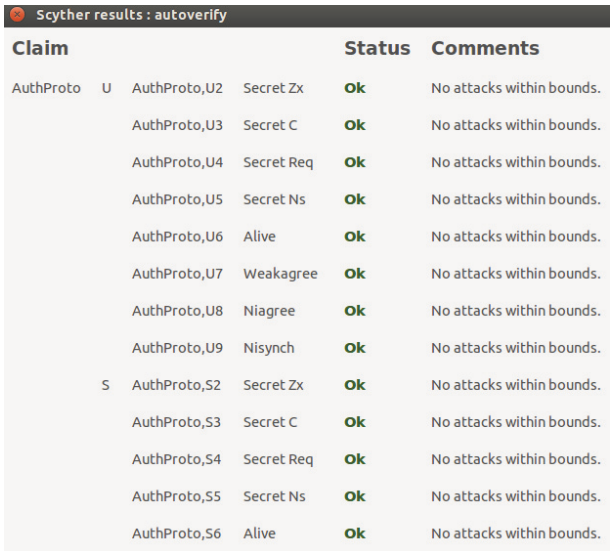

Figure 7: Proposed Stage I authentication protocol autoverification using Scyther.

```
claim(U,Secret,Zx);
claim(U,Alive);
claim(U,Weakagree);
claim(U,Commit,S,Ns);
claim(U,Niagree);
claim(U,Nisynch);
}
role S
{
fresh Ns:Nonce;
const Req;
const C;
const Zx;

recv_1(U,S,Req);
send_2 (S,U,Ns);
recv_3 (U,S,C,Zx);

claim(S,Secret,C);
claim(S,Secret,Zx);
claim(S,Alive);
claim(S,Weakagree);
claim(S,Commit,U,Ns);
claim(S,Niagree);
claim(S,Nisynch);
}
}
```

Stage II authentication protocol is also verified by Scyther verification tool. The source code of Stage II is also listed and the output is shown in Figure 8.

7. Performance Analysis

In the trusted third party based authentication schemes, performance is one of the important factors to concentrate on. As we know, performance is having close relation with the security. Most of the third party based authentication protocols are using two-stage authentication process. Stage I is registration phase or initial authentication phase and Stage II is login and authentication phase. Mostly, the registration phase or Stage I authentication is a one-time process at the time of registering the user. Hence, to calculate the performance of our proposed authentication scheme in terms of computation and number of communications, we consider only Stage II authentication. In the proposed scheme, we assumed the size of identity is 32 bits and hash size is 160 bits (we use SHA-1). As we mentioned above, registration or Stage one authentication is happening only once. Hence, we consider only Stage two authentication (login and authentication phase) for calculating the computation and communication. During Stage two, Step 1, user sends the authentication request as mentioned in (23); the size of the identity and the request is 96 bits; Step 2, CSP verifies as mentioned in (24); the size of the user identity and fresh authentication ID is again 96 bits only; Step 3, user device calculates the values C and Zx as explained in (25) to (29) and then sends the C and Zx to CSP; size of the identity is 64 and the hash values of C,Zx are 160+160. Last communication message size is 384 (32+32+160+160) bits. Hence the total transmission size is 576 bits in 3 communications. Also, Table 4 shows the proposed scheme is more efficient than the recent similar authentication schemes.

Table 4: Performance Analysis with Recent Schemes.

Sl. No	Schemes	No. of Bits	No. of Messages
1	Lee et al. [2]	1184	7
2	Dey et al. [11]	1280	4
3	Lin et al. [12]	1536	4
4	Roy et al. [14]	864	2
5	Binu et al. [29]	2304	7
6	Our Scheme	576	3

In the proposed authentication scheme performance analysis, we use few cryptographic operations and its notations as follows:

(i) Hash function as Th

(ii) Multiplication or key generation or verification as Tm

We used SHA-1 to calculate hash function Th, used ECC for multiplication, and used key generation and verification Tm to compute the C and Zx values. As per the equation numbers from (25) to (29), 5 equations are used to compute the values C and Zx. We use 2 Th, 2 Tm in mobile device. We ignored the cost of XOR operations due to negligible computation load. Table 5 explains and compares the computation cost of multiplication Tm and hash function Th with recent similar schemes.

As we know, the computation capacity of smartphone is growing day by day. Nowadays octa-core processor with 3GB RAM smartphone cost is under $100. Moderate to high end smartphone is having 8GB RAM. Hence, our scheme is working good in recent smartphones. Also, we are testing our

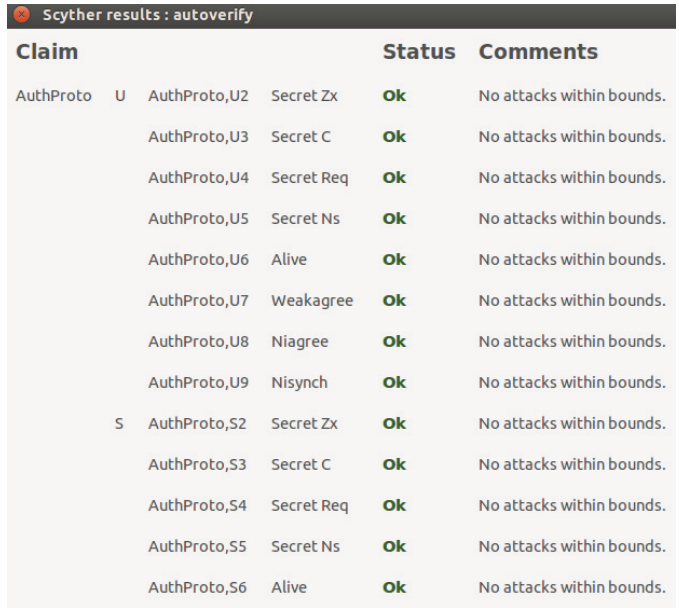

Figure 8: Proposed Stage II authentication protocol verification using Scyther.

Table 5: Computation Cost Analysis with Recent Schemes.

Sl. No	Scheme	Cost of Computation
1	Lee et al. [2]	$4Th+3Tm$
2	Dey et al. [11]	$5Th+4Tm$
3	Lin et al. [12]	$10Th+2Tm$
4	Roy et al. [14]	$9Th+1Tm$
5	Binu et al. [29]	$9Th+3Tm$
6	Our Scheme	$2Th+2Tm$

scheme to use Dynamic Computation Offloading technique in our future work to get best performance while using our scheme in the Low Powered Device. Due to this reason, we did not explain the execution time of recent similar schemes. Tables 4 and 5 show that our scheme is using less number of message communications with tiny data between communication entities. Also, our scheme uses less number of mathematical functions to achieve best computation cost and efficient security in mobile devices.

8. Conclusion

The traditional authentication methods are not suitable for mobile cloud computing due to its dynamic nature and support of various cloud services. In this paper, we presented new authentication scheme to secure the user password from the phishing attack. The proposed authentication scheme is not sending the password in any form of the existing methods like hash value and encrypted key or a digital signature to verify the identity of mobile user and the cloud service provider. In this scheme, we have used the Zero-knowledge proof technique to satisfy the authentication process. Also, the proposed scheme verified by the University of Oxford developed protocol verification tool, Scyther. The security verification and the experimental result brought about an exhibit that the proposed scheme is more secure against the phishing attack in the mobile cloud computing. In the future, we would like to explore more attributes to provide efficient mobile cloud authentication in the multicloud environment with Dynamic Computational Offloading Technique.

Data Availability

The formal verification of Scyther code Stage I and Stage II authentication used during the current study is included in the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this research paper.

Acknowledgments

This research work is partially supported by Ministry of Electronics and IT, Govt. of India's Information Security Education Awareness Project Phase II.

References

[1] M. Alizadeh, S. Abolfazli, M. Zamani, S. Baaaharun, and K. Sakurai, "Authentication in mobile cloud computing: a survey," *Journal of Network and Computer Applications*, vol. 61, pp. 59–80, 2016.

[2] A. Lee, "Authentication scheme for smart learning system in the cloud computing environment," *Journal of Computer Virology and Hacking Techniques*, vol. 11, no. 3, pp. 149–155, 2015.

[3] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[4] Z. Ahmad, K. E. Mayes, S. Dong, and K. Markantonakis, "Considerations for mobile authentication in the Cloud," *Information Security Technical Report*, vol. 16, no. 3-4, pp. 123–130, 2011.

[5] K. Akherfi, M. Gerndt, and H. Harroud, "Mobile cloud computing for computation offloading: issues and challenges," *Applied Computing and Informatics*, vol. 14, no. 1, pp. 1–16, 2016.

[6] A. Kannammal and S. Subha Rani, "Authentication and encryption for medical image security system," *International Journal of Robotics and Automation*, vol. 29, no. 4, pp. 448–455, 2014.

[7] B. K. Chaurasia, A. Shahi, and S. Verma, "Authentication in cloud computing environment using two factor authentication," in *Proceedings of the Third International Conference on Soft Computing for Problem Solving*, vol. 259 of *Advances in Intelligent Systems and Computing*, pp. 779–785, Springer, New Delhi, India, 2014.

[8] J. Wei, X. Hu, and W. Liu, "An improved authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 36, no. 6, pp. 3597–3604, 2012.

[9] H.-T. Yeh, B.-C. Chen, and Y.-C. Wu, "Mobile user authentication system in cloud environment," *Security and Communication Networks*, vol. 6, no. 9, pp. 1161–1168, 2013.

[10] A. Ahmed-Nacer and M. A.-N. Samovar, "Strong authentication for mobile cloud computing," in *Proceedings of the 13th International Conference on New Technologies for Distributed Systems*, France, 2016.

[11] S. Dey, S. Sampalli, and Q. Ye, "MDA: message digest-based authentication for mobile cloud computing," *Journal of Cloud Computing*, vol. 5, no. 1, p. 18, 2016.

[12] H. Lin, "Efficient mobile dynamic ID authentication and key agreement scheme without trusted servers," *International Journal of Communication Systems*, vol. 30, no. 1, Article ID e2818, 2017.

[13] S. Namasudra and P. Roy, "A new secure authentication scheme for cloud computing environment," *Concurrency Computation: Practice and Experience*, vol. 29, no. 20, p. e3864, 2017.

[14] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services," *IEEE Access*, vol. 5, pp. 25808–25825, 2017.

[15] S. Grzonkowski, P. M. Corcoran, and T. Coughlin, "Security analysis of authentication protocols for next-generation mobile and CE cloud services," in *Proceedings of the IEEE International Conference on Consumer Electronics*, pp. 83–87, Berlin, Germany, 2011.

[16] P. Mohit, R. Amin, A. Karati, G. P. Biswas, and M. K. Khan, "A standard mutual authentication protocol for cloud computing based health care system," *Journal of Medical Systems*, vol. 41, no. 4, p. 50, 2017.

[17] S. Kalra and S. K. Sood, "Advanced password based authentication scheme for wireless sensor networks," *Journal of Information Security and Applications*, vol. 20, pp. 37–46, 2015.

[18] D. Huang and H. Wu, "Mobile cloud security: attribute-based access control," *Mobile Cloud Computing*, pp. 181–211, 2018.

[19] C.-T. Li, C.-C. Lee, and C.-Y. Weng, "A dynamic identity-based user authentication scheme for remote login systems," *Security and Communication Networks*, vol. 8, no. 18, pp. 3372–3382, 2015.

[20] K. Zhou, M. H. Afifi, and J. Ren, "ExpSOS: secure and verifiable outsourcing of exponentiation operations for mobile cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2518–2531, 2017.

[21] S. Grzonkowski, A. Mosquera, L. Aouad, and D. Morss, "Smartphone security: an overview of emerging threats," *IEEE Consumer Electronics Magazine*, vol. 3, no. 4, pp. 40–44, 2014.

[22] 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*, vol. 99, pp. 1–11, 2017.

[23] A. Miller, "Zero-knowledge proof notation and vocabulary," in *Lecture 7 - Zero Knowledge Proofs*, ECE/CS 598AM: Cryptocurrency Security, pp. 1–4, 2016.

[24] B. Lum Jia Jun, "Implementing zero-knowledge authentication with zero knowledge," in *Proceedings of the PyCon Asia-Pacific*, 2010.

[25] E. Munivel and J. Lokesh, "Design of secure group key management scheme for multicast networks using number theory," in *Proceedings of the 2008 International Conference on Computational Intelligence for Modelling Control and Automation, CIMCA 2008*, pp. 124–129, Austria, December 2008.

[26] J. Zhang, Z. Zhang, and H. Guo, "Towards secure data distribution systems in mobile cloud computing," *IEEE Transactions on Mobile Computing*, vol. 16, no. 11, pp. 3222–3235, 2017.

[27] S. L. Albuquerque and P. R. L. Gondim, "Security in cloud-computing-based mobile health," *IT Professional*, vol. 18, no. 3, pp. 37–44, 2016.

[28] D. Huang and H. Wu, "Mobile cloud offloading models," in *Mobile Cloud Computing*, pp. 115–152, Morgan Kaufmann, 2018.

[29] S. Binu, M. Misbahuddin, and P. Raj, "A strong single sign-on user authentication scheme using mobile token without verifier table for cloud based services," in *Computer and Network Security Essentials*, K. Daimi, Ed., pp. 237–261, Springer International Publishing, Cham, Switzerland, 2018.

[30] S. Chen, D. L. Chiang, C. Liu et al., "Confidentiality protection of digital health records in cloud computing," *Journal of Medical Systems*, vol. 40, no. 5, 2016.

[31] C. J. F. Cremers, "The scyther tool: verification, falsification, and analysis of security protocols," in *Computer Aided Verification*, Springer, Berlin, Germany, 2008.