

Secure Authentication Protocol for Efficient Computational Offloading Service in the Mobile Cloud Computing

Munivel E, Kannammal A

Department of Electronics and Communication Engineering, PSG College of Technology, India
e.munivel@gmail.com, aks.ece@psgtech.ac.in

Abstract

Battery-powered mobile devices are convenient to use anywhere and anytime. Nowadays maximum people using the mobile device, such as Smartphone, Tablet computers to use mobile services anytime and anywhere. Moreover, most of the Smartphone having wireless communication like Wi-Fi and 4G. For some applications, the requirement of computing power may be very high, but the actual configuration of mobile devices are very limited, such as CPU, memory, storage, and battery. Among these computational resources, bandwidth and battery are the most significant problems for Smartphone. Efficient Computational Offloading is the best solution for extending the usage of Smartphone by executing the resource-intensive task to offload from mobile to the remote cloud servers can extend processing capability and support for multiple categories of application. However, this technique is having difficulty in offloading the process to a remote cloud server without the proven security of entity verification. To deal with these challenges, here we are proposing new security protocol to authenticate mobile and cloud server with zero knowledge proof of authentication to verify the communication entities and recommends to offload the service. The proposed protocol will get verify by the University of Oxford developed verification tool Scyther.

Keywords: Authentication, Mobile cloud computing, Smartphone, Cloud server, Computational offloading

1 Introduction

Virtualization is a parent technology for Cloud Computing to emulate the computing infrastructure like real in the isolated environment [34]. Cloud Computing is the technology will be going to modernize the IT field shortly and will reduce the workforce with the help of cloud automation. Also, Cloud Computing is the technology will give computing resources as a service over the network [1-3]. The cloud computing is the hybrid technology to deliver on-demand computing services over the

network [20]. This future technology will provide more computing capacity to any user on the rental basis as mentioned in the Figure 1 as cloud computing architecture [20].



Figure 1. Cloud computing architecture

Mobile devices and laptops, have limited computing resources in to use the processing ability, life of battery. Therefore, this type of battery powered device is inadequate to use the high-end computational tasks. Also, there is a fast progress of based on these low power devices and extremely curtail their battery lifespan as an outcome when using the resource intensive tasks. Resource intensive mobile applications like multimedia, natural language processing and augmented reality are becomes gradually getting rigorous and also required to refine computational resources [4]. Particularly at the time of using the user-interactive applications, the mobile device has to wait more time to complete the process execution due to the restricted processing capabilities of the mobile cloud device [10].

Computational offloading, which takes place of unused computational resources available by the cloud server, and it is becoming a capable technique to resolve an amount of problems disturbing in the mobile cloud computing [9]. The main idea is to offload the processing limitation from the mobile devices through transferring the resource intensive processes from mobile cloud devices to the remote cloud servers. This

offloading method brings many prospective benefits, like refining the performance of mobile cloud applications and decreasing the battery usage and so on in the mobile devices.

1.1 Cloud Characteristics

- Request Base Service: Cloud user can apply to get the computing resources based on the requirement without the intervention of any service provider [20].
- Broad Network Access: Plenty of services grouped and get delivered over the network in a heterogeneous manner to any device [20].
- Rapid Elasticity: Computing capabilities can automatically provide to the user with the minimum and maximum threshold [20].
- Resource Pooling: The service provider's computing resources like process, memory, storage, and networking are dynamically added to serve multiple users with on-demand [20].
- Metered Service: Cloud services can control automatically with the optimized resources to the user and will get reported correctly to the provider and consumer [20].

1.2 The Cloud Models

- Cloud Software as a Service: Software services can be delivered to the end user over the network without installing on the dedicated device [25].
- Cloud Platform as a Service: Software can be developed and deployed using any end-user device. The services provider could manage the dependency of development resources and libraries.
- Infrastructure as a Service: Infrastructure service is one of the critical types of cloud service to offer the virtualized computing resources like computation, storage, and networking to the consumer over the web. The customer can decide the requirement of computer resources based on their application deployment over the metered service.

1.3 Overview of Mobile Cloud Computing

The Mobile cloud is defined to use the cloud technologies in mobile devices where processing and data storage will be on remote cloud servers as shown in the Figure 2 [23]. Nowadays so many mobile cloud computing applications are using publicly like Mobile Gmail, Google Maps, Facebook and so on [9, 32].

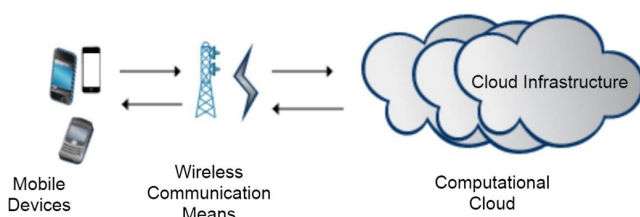


Figure 2. Mobile Cloud Computing Overview [23]

However, in recent years, most of the mobile devices having massive data storage and more processing capabilities in mobile itself [9-10]. In upcoming years this situation will get change. With the advancement of smartphones, the cloud market is turning towards building the smartphone with high-end apps, which leads to use the supercomputing power in mobile [7-8].

However, two main reasons are there, why cloud computing get disturbed in its advancement [6]. First is infrastructure, in India, most of the time we are using the internet in kilobit per second and more malware on mobile devices, especially on Android [18]. Whatever devices using the Android operating system are not clean android. Customized to use the application of the mobile manufacturers or to use the service provider's pre-installed apps. This nature will degrade the performance of smartphone life.

1.4 Related Work

Authentication is an essential security service in any system or network communications [14]. It is classified as, user authentication, remote authentication, mutual authentication, message authentication and implicit authentication [11]. The current authentication review shows the different attributes, based on password, hash value, Identity, digital signature, hierarchical model, mobile number, group key and biometric [18, 21].

To achieve mutual authentication in mobile cloud computing Grzonkowski et al. [28], Ahmed et al. [24] and Todd et al. [31] are proposed different authentication protocols in the mobile and cloud service environment. According to the Quasim et al. [27] scheme, the user ID is sharing using the secure channel, but the smart card generator, 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, Ahmed et al. [24] and Todd et al. [31] scheme prone to man-in-the middle attack and phishing attack.

Computation offloading tasks from smart-phones to remote cloud server has recently been rediscovered as a technique to enhance the performance of Smartphone applications, beyond the limit of its capacity [5, 9, 26]. Mobile phones are resource constrained devices, like a limited battery, low bandwidth, limited storage capacity, and low-speed processor [9, 13]. These limitations can be overcome by computational offloading, sending the process to the cloud server and receiving back the results from these remote servers [29-30].

The problem of the process offloading had identified in the general computing environment in the past [10]. This research gives an overall method to decide the computational offloading technique and its applications.

Computation offloading is most important for the resource-constrained mobile devices [12, 16]. The Certain process cannot execute on mobile devices due to limited resources. The only possible way to use those application programs is to offload all or part of the computation to remote cloud servers [9]. All the reviewed research normally emphasis on one or two offloading tasks such as computational resource, data storage and data transmission mark to decide the offloading choices founded on device monitoring applications through the cloud agent's application installed in the mobile cloud device.

2 Methodologies

This section explains the computations offloading technique along with the offloading methods. Finally, explains the proposed method of authentication with the resistance to certain security attacks and energy efficiency.

2.1 Computational Offloading Requirements

This section presents basic details on mobile cloud computational offloading technique and define the key challenges and problems related with the computational offloading decision making.

2.1.1 Basic Computational Offloading

The basic offloading method is having multiple modules, when user generates a request to offload the computation from their device, few details are collected to take decision like existing computational capability and also the availability of networking capability by the offloading agent. There are multiple agents combined here as Mobile Agent required in the computational offloading like process agent, power agent, bandwidth agent and memory agent. Each agent having their own responsibilities as follows,

- Process agents: This module collects the information's like amount of execution time, physical memory utilization and required amount of data during the execution of the applications running in the mobile device.
- Bandwidth Agent: This module gathers the information about the network bandwidth and the status of network connection with the history of number of times connected and disconnected.
- Power Agent: This module gathers the information about the power consumption of the application running in the mobile device also calculating the expected time to run the application in the device using the power monitoring applications.
- Memory Agent: This module monitors the total, available and required physical memory, and also collects the amount of physical memory required to run during the execution of the process.

2.1.2 Computational Offloading Criteria

Computational offloading decisions are typically taken based on a certain cost measure. This metrics decided as energy cost to calculate the amount of power being utilized. Storage cost is calculating as amount of data is being used to run in per second. The performance, robustness and safety are important metrics which need to be take full advantage before the offloading process begin. Among all these criteria, power, bandwidth and performance are significant characteristics for the concern of the mobile user.

2.1.3 Process Fragmentation and Decisions Making

On the base of the collected details, the offloading decision-making application takes the decision according to the above criteria. Then the fragmentation module is invoked to cut the process that classify the process into local and remote portion of tasks. The local process is executed by the mobile device and the remote portion is offloading to the remote cloud server over the secure channel. This application fragmentation is done either statically or dynamically.

The mobile agent sends the fragmented tasks to the cloud agent which is located in the remote cloud server over the secure connection between the mobile agent in the mobile phone and the cloud agent located in the remote cloud server.

Cloud agent module is invoked to find a suitable virtual server for offloading. The offloaded tasks interact with the local tasks in the local partition when required to share the intermediate values to complete the process execution. When finish the execution, the final results sent back to the mobile cloud device.

2.1.4 Offloading Scheme

The advanced computation offloading system divides the whole process of computation into multiple tasks without any restrict at any particular level [19]. The proposed technique divides the tasks into remote tasks, and local tasks such that the remote tasks run on the cloud server and the local tasks run on the end-user device, and also the domestic tasks and the remote tasks execute in a distributed way in the way of the actual flow of sequential control. The Figure 3 describes, how the mobile agent transfers the tasks to remote cloud server. The cloud agent is deciding the number of virtual machines to maintain the process offloading beneficial. Also, in the proposed model is transferring the piece of process over the secure channel to maintain confidentiality in the public network.

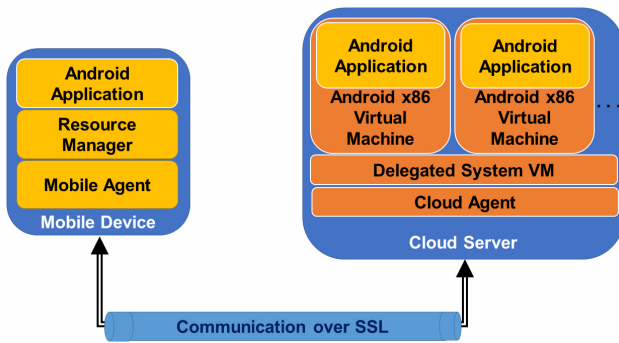


Figure 3. Computational offloading

2.1.5 Process Migration

The Mobile agent in computation offloading scheme will migrate the running process from mobile to cloud servers to get more processing capability in the resource-constrained device. The process of offloading computation helps to increase the client device computation performance and to save power on the end-user device [17].

Energy saving in the end-user device: Mobile phone is the first option for many users to use day to day activities. Using high-end applications in mobile devices will consume more power as well as processing, but this can be solved using computational offloading to cloud servers. Finally, mobile users can use the high-performance computing application with the help of offloading to cloud servers.

To solve the primary thin client or a centralized computing model of computational offloading problems, proposing a Selective Computational Offloading to use the customer device efficiently and complete the offloading with limited bandwidth.

Offloading computation to another device is not a new technique, already the thin client uses this way but dynamically offloading the task to the cloud server and the process offloading based on a specific threshold condition like running process increases more than 80%. Moreover, the efficient use of local resources is essential. So, always offloading the tasks to the remote server is not efficient one, like when running jobs are less than 30% of its capacity. If the running tasks are more than 80% will get offloaded fully. If the task level is average like in between of 30% to 80%, can be offloaded based user request to the cloud server.

2.2 Zero Knowledge Proof (ZKP)

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. The Zero-knowledge protocol is based on Zero-knowledge proofs and can classify as Interactive Zero-knowledge and Non-Interactive Zero-knowledge based on the working methods [33]. The Interactive Zero-knowledge protocol uses multiple authentication steps of communications between the prover and

verifier. The non-interactive Zero-knowledge protocol uses only one communication message called proof between the prover and verifier [33].

The Properties of zero-knowledge proof can be distinguished as follows,

- **Completeness:** “If requested statement is correct, the honest verifier will prove that the requested statement is true to the honest verifier” [22].
- **Soundness:** “If the requested statement is false, there is no way to fake the result to the verifier that the requested statement is true” [22].
- **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” [22].

2.3 Proposed Security Framework

This section presents the new framework to verify the communication entities before the computational offloading process begin. In the proposed method new concept is introduced as cloudlet. Cloudlet is an emerging research concept of cloud and mobile computing technique to deploy the cloud-in-a-box in the nearest location of the mobile user, also called as portable private edge cloud server. Here, using the cloudlet for the different propose to implement the mobile cloud service in specific location like office, home, class room or any private location. This service could be used preferably over the home, office or institutional Wi-Fi.

Aim of this proposed framework is to build and maintain a cloned copy of mobile phone in the cloudlet with synchronization of mobile cloud client device and the mobile phone, the cloned virtual machine of the cloudlet synchronizes with the mobile device when needed. Another way to present, cloudlet maintains one cloned copy of the mobile cloud application of the mobile phone. The virtual machine of the cloudlet and mobile phone is working as twin mobile (only in the cloud application perspective), both synchronized and communicating over the secure private network.

Cloudlet maintain the same version of android mobile operating system as virtual machine. Also, maintain the cloned copy of mobile cloud application.

The given group g is having set of values. g_0, g_1 are the carrier set of random elements of group g . Hence, the public key may be the g, g_0 .

The group g is a carrier set cordiality of the order of Group $|g|$ [21-23].

2.3.1 System Model

A typical cloud authentication system model of the proposed cloud scheme shown in Figure 4. Here, using three roles in the proposed method.

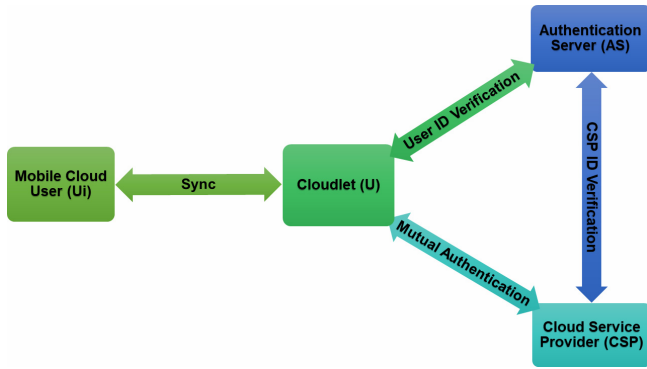


Figure 4. System model

- Mobile User (Ui): He / She is a mobile user, registering as a cloud user with the Cloudlet (U) using mobile device verification to confirm the device identity with full permission of the device over the private connection.
- Cloudlet (U): This is a clone of User Ui's mobile device, registering as a new user with the Authentication Server (S) through the Cloudlet (Cloned User) (U) using email verification to confirm the initial identity. Then the user using its user ID and password, to generate the Public Key with using mobile cloud application, then sending the digest value of the user ID and public key to the VDI Server.
- Authentication Server (AS): 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.
- Cloud Service Provider (CSP): CSP provides services computation service to the Cloudlet (Cloned User). It verifies the user request with its URI. If URI is on the approved list, it will ask the TTP to verify. Then TTP verifies the mobile number, the user ID. Finally, the user ID, TTP nonce, and public key will send to CSP to confirm the Cloud User.

The proposed mutual authentication is a type of zero-knowledge proof technique and its system model shown in the Figure 4. This technique is not going to share the user's real password to the remote cloud server. Moreover, to initiate the authentication or to verify the mutual authentication, a remote cloud server is sharing the random value as a one-time key to the mobile cloud user. Based on the one-time server key, the cloud user starting the authentication process as per the following method.

2.3.2 Initial Registration

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 trusted a third party. The third

phase verifies the cloud user and the service provider to achieve the mutual authentication. Also, the notation and description used in the offloading service is shown in the Table 1.

Table 1. Notations used in the Proposed Protocol

Notation	Description
\parallel	Concatenation
\oplus	XOR Operation
$h(U)$	Hash Value of User ID
$h(PW)$	Hash Value of Password
U_i	Mobile Cloud User
U	Cloudlet (Cloned) User
S	Cloud Service Provider
AS	Authentication Server
U_{id}	Client URI with Mobile No
$sk(U)$	Private Key of User
$pk(U)$	Public Key of User
$sk(S)$	Private Key of User
$pk(S)$	Public Key of User
$AuthID$	Fresh Authentication ID
R	Random Value Gen by User
$N_a, N_{u1}, N_{u2}, N_{u3}$	Fresh Nonce
$h()$	Hash Function

2.3.3 Registration Phase

The new user generates a request with the Authentication Server (AS) with its e-mail is original identity. AS verify the available list of available registered e-mail number. If the e-mail is new, the AS sends the OTP else terminates the communication. The entered OTP will get verified with AS. Once OTP verified; then the user enters the new user ID and password, the mobile browser generates the hash value of the user ID as H_1 and generates a hash value of Password as H_2 . Client browser generates the Public Key P using the hash value of the User Password H_2 .

Finally, the user sends the Hash value of User ID H_1 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 shown in the Figure 3. All the above communications are happening over the secure channel between User and AS. New Cloud Service Provider (CSP) generates a new request to AS. AS verify the existence of the new domain in the existing list. If free, the AS accept 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 Domains URI in the trusted list and share the CSP Public Key to AS.

2.3.4 Authentication Phase

In this phase, Authentication Server (AS) and

Mobile Cloud User (U) are participating to verify each other to achieve mutual authentication without revealing the real password. Table 1, describes the notations used in the authentication phase of the proposed protocol.

In this phase, Authentication Server (AS) and Mobile Cloud User (U) are participating to verify each other to achieve mutual authentication without revealing the real password.

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

$$U \rightarrow S : U, S, N_1 \tag{1}$$

Step 2: Cloud Service Provider S sends the random token Sid as Communication ID to the user's request after verifying the Client's IMEI number.

$$S \rightarrow U : S, U, Sid, h(N_1) \tag{2}$$

Step 3: User enters user ID and password in the mobile agent app. The app generates the hash value of User's Password. Based on this hash values, the app generates the value x as follows. Hence, the password is not leaving the client app.

$$x = h(PW_u) \tag{3}$$

Then the user computes Pu (Public Key of User U) with using x and the shared group g_0

$$Pu = g_0^x \tag{4}$$

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

$$Q = g_0^{r_x} \tag{5}$$

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

$$C = h(Pu, Q, N_s) \tag{6}$$

$$Zx = Rx - Cx \tag{7}$$

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

$$U \rightarrow S : C, Zx \tag{8}$$

The Server S calculates the value Q as follows

1. Server receives C and Zx
2. The server has the users Sid, Public Key Pu and shared group element g_0

The server calculates Q ,

$$Q = Pu^C g_0^{Zx} \tag{9}$$

Then the server S checks the C

$$C = h(Pu, Q, N_s)$$

In this proposed protocol the random value r is generated by the user, but this value r is constructed

by the Server S with using above values as follows.

As per the 5th equation, $Q = g_0^{r_x}$

And $Zx = Rx - Cx$

So, we can prove with using simple substitution as follows, above equation

$$Q = g_0^{r_x} \text{ and } Q = Pu^C g_0^{Zx}$$

$$g_0^{r_x} = Pu^C g_0^{Zx} \text{ As per above equation } Pu = g_0^x$$

$$g_0^{r_x} = (g_0^x)^C g_0^{(rx-cx)} \tag{10}$$

$$g_0^{r_x} = g_0^{cx} g_0^{rx-cx} \tag{11}$$

$$g_0^{r_x} = g_0^{cx+rx-cx} \tag{12}$$

$$g_0^{r_x} = g_0^{rx} \tag{13}$$

Now User's random value r is constructed by the server S to verify that the User is genuine or not and also User proves that the server's random value Sid is known by the Mobile Cloud User to achieve the mutual authentication.

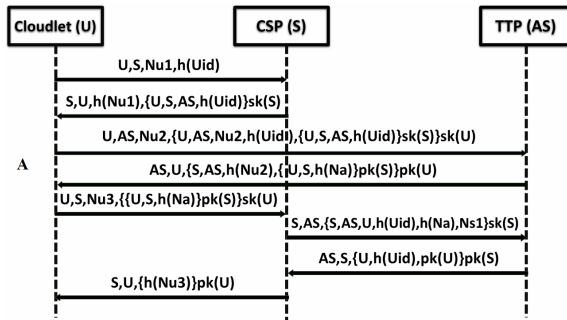
Authentication is satisfied based on the above steps. Once communication entities verified, the mobile cloud user starts to offload the process. The Figure 5 explains the detail sequential diagram of the two stages of proposed protocol.

2.3.5 Secure Computational Offloading

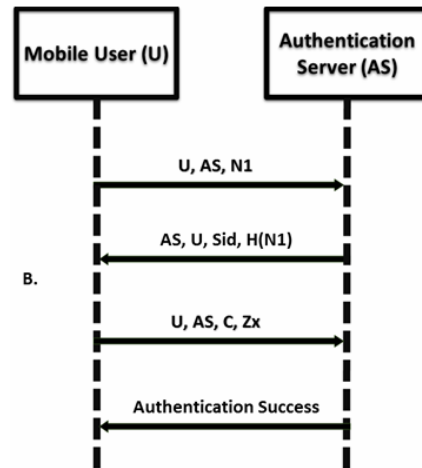
Computational offloading is a method to moderating resource intensive computation to remote cloud servers. This is valuable from the performance and energy perception, it positively shows different trials in terms of security due to increased bandwidth over networks with possibly attacks. Privacy is a main concern in mobile cloud computational offloading. Once the user tasks are offloaded to the remote cloud that are not under the users' control, privacy is having the important role in keeping the data is secure. In the midst of probable security problems are authentication and the confidentially related attacks which can be rectified by the presented scheme by the ZKP methods, once the communication entities are authenticated, then the cloudlet offload the tasks into the CSP over the private network establishing for the specific sessions. But every communication is verifying the sender and the receiver tags to confirm the sender and the receiver URL to verify the over communication is secure. Hence, encrypting the computation offloading tasks is still a study.

2.3.6 Testing Various Attacks

One of the key developments related to ZKP-based authentication protocols in the computational offloading contexts also relates to the application of



(a) Authentication Phase-I



(b) Authentication – Phase-II (Mutual Authentication)

Figure 5. Sequential diagrams of two stages of authentication

their core characteristic to the realistic problem of the development of cloud technology and the increasing importance of mobile devices a cloud service accessing device. Hence, security is one of the important concerns during the offloading, also, the offloading process must be resistance to the major attacks. This system is non-formally proved to resist against the significant attacks in the following items,

- User Anonymity: One of the important issues in the mobile cloud computing is user anonymity. Hence, the user anonymity is to be considered to secure the privacy of the Cloudlet (Cloned User). User anonymity means that a remote user’s actual user identity will be disguised throughout the authentication phase and the original identity cannot be traced by unauthorized users, and he/she cannot be linked or traced by any intruders. Hence, in this authentication scheme the real user identity is hashed with the user profile as a Tag. It cannot be extracted without the server Tag. Hence, this scheme is secure against the the user anonymity.
- Password Guessing Attack: In this scheme the user US is password is not sending to the Server S or any other communication entities. This scheme is generating the Public key using the User U User Id and the Password. This password only known by the User U only. Server is verifying the communication by using the user public key and other parameters using in this multi-identity scheme. So, the access privileges to a mobile and authentication server are negotiated by trying several combinations of usernames and the corresponding passwords till the hacker crack. So, in this scheme brute force and the dictionary cannot guess or crack anything. Hence, this scheme is secure against the password guessing attack.

- Impersonation Attack: In this scheme an impersonation attack is considered to avoid. In this scheme the user and server communication are actively verified by the tags of communication entities. Hence, the impersonation from the server side to user as well as user impersonation to server also monitor by every communication of the scheme. Hence, the user and server impersonation attacks cannot attempt by attacker in this scheme.
- Phishing Attack: Capturing user identity and password by using fake websites or mobile application, etc. used in this technology world. Hence, the communication must be secure and should satisfy the authentication service with phishing resistance. In the proposed scheme, the user password is not sharing in any means, to any destinations, it sharing only the computed blind value from the user U to server S is C, Zx. With these values attackers cannot construct any value. So, this scheme is secure against the phishing attack.
- Forward Secrecy: This scheme uses the asymmetric cryptographic method is to generate the keys of user Identity and to construct the intermediate values C, Zx to get authenticate with the server. Hence, in this scheme forward secrecy cannot compromised.
- Mutual Authentication: This scheme is mutuality verify the user U with the server S. User U is verified by the server by U’s public key which is encrypted by the U’s private key and random value R. Also, Server S is verifying by the User U, when fresh AuthID sends to U and construct the values by using its private key, finally send the C, Zx to Server S. The server S reconstruct with using U’s Public Key and AuthID to get the U’s random value R. So, both parties verify each other to confirm mutually authenticated.

3 Analysis and Verification

This section presents the formal security verification using Scyther tool, followed by computational offloading analysis and energy saving using the cloud simulation tool called GreenCloud as follows.

3.1 Security Verification Using Scyther

The security features associated with ZKP-based protocol in offloading systems correspond closely with their inherent characteristics and their most salient features as specific protocols. Namely, these benefits relate to their simplicity, their difficulty in terms of being replicated, and their ability to be utilized within multi-user cloud environment.

The scheme is verified using the automated protocol verification tool called Scyther, developed by Cremers et al. Scyther is having the features like Unbounded verification, Attack finding, visualisation, also supports the classical properties like secrecy, agreement, aliveness and synchronisation. The code of the proposed scheme is written in security protocol description language. This authentication phase is verified by the scyther and display the resistance against the significant attacks in the Figure 6.

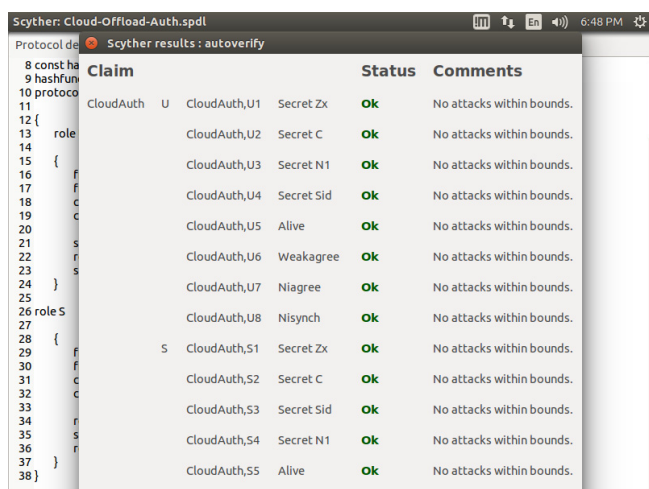


Figure 6. Auto verification result of proposed protocol

3.2 Performance and Computation Cost Analysis

In the mobile cloud authentication schemes, performance is one of the important factors to concentrate, due to the use of battery powered device. In the proposed scheme, the size of identity is assumed as 32 bits and hash size is 160 bits (uses SHA-1). As mentioned above, Initial Registration or Stage One authentication is only once in one user cycle. Hence, here consider only Stage Two authentication (Login or Authentication Phase) for calculating the computation and communication. During the Stage Two or the authentication phase, Step 1: user sends the authentication request as mentioned in the equation no.

1, the size of the identity and the request is 128 bits. Step 2: Server S verifies as mentioned in the equation no. 2, the size of the user identity and Fresh Authentication ID is again 128 bits only. Step 3: user device calculates the values C and Zx as explained in the equations no. 3 to 4. Then sends the C and Zx to the Server S, size of the identity is 64 and the hash values of C, Zx is 160+160. Last communication message size is 448(128+160+160) bits. Hence the total transmission size is 704 bits in 3 communication). Also, the above listed Table 2 shows the proposed scheme is efficient than the recent similar authentication schemes.

Table 2. Performance analysis with recent schemes

SI No.	Schemes	No. of Bits	No. of Messages
1	Lee et al. (Lee et al. 2015)	1184	7
2	Dey et al. (Dey et al. 2016)	1280	4
3	Lin et al. (Lin et al. 2017)	1536	4
4	Roy et al. (Roy et al. 2017)	864	2
5	Binu et al. (Binu et al. 2018)	2304	7
6	Our Scheme	704	3

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

- Hash Function as Th
- Multiplication or Key Generation or Verification as Tm

The SHA-1 is used to calculate the Hash Function Th, used ECC to Multiplication, Key Generation and Verification Tm to compute the C and Zx values. As per the equation nos from 3 to 7 are using to compute the values C and Zx. Here, 2 Th, 2 Tm used in Mobile Device. The cost of XOR operation is ignored due to negligible computation load. The Table 3 explains and compare the Computation cost of Multiplication Tm and Hash function Th with recent smiler schemes.

Table 3. Computation cost analysis with recent schemes

SI No.	Schemes	Cost of Computation
1	Lee et al. (Lee et al. 2015)	4Th + 3Tm
2	Dey et al. (Dey et al. 2016)	5Th + 4Tm
3	Lin et al. (Lin et al. 2017)	10Th + 2Tm
4	Roy et al. (Roy et al. 2017)	9Th + 1Tm
5	Binu et al. (Binu et al. 2018)	9Th + 3Tm
6	Our Scheme	2Th + 2Tm

The Table 2 and Table 3 shows our scheme is using fewer number of message communication with tiny data between communication entities. As well as, our scheme using fewer number of mathematical functions to achieve best computation cost and efficient Security in Mobile Devices.

3.3 GreenCloud

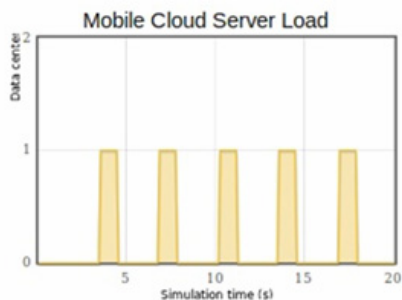
GreenCloud is a cloud simulator developed for the study of cloud computing environments by the University of Luxembourg and it is a sophisticated packet-level simulator for energy-aware cloud computing data centers with a focus on cloud communications. It offers a detailed fine-grained modeling of the energy consumed by the data center IT equipment, such as computing servers, network switches, and communication links.

- Energy Efficiency: The Green Cloud, simulator is designed to capture details of the energy consumed by data center modules (servers, switches and links) as well as packet-level communication patterns in realistic setups.
- Other Features: It has the facility to define detailed modelling of the energy consumption by virtual server and virtual switches. GreenCloud offers a thorough investigation of process and bandwidth workload distributions among other virtual cloud servers.

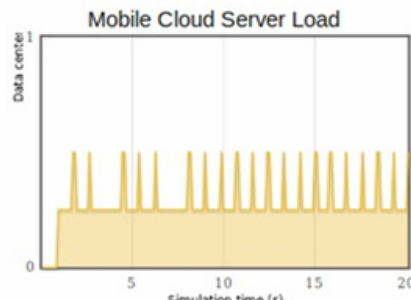
3.4 Energy Saving in the Mobile Cloud

The main problems of the mobile cloud computing technologies cannot complete the tasks in time due to limited processing capability with less power and bandwidth [17, 27]. The Figure 7, explains the offloading process load, before and after the process offloading, as well as the energy status, before and after the process offloading. Mobile cloud computing achieves the efficiency and save the energy in the mobile device when resource-intensive tasks are moving to the server, some applications like image processing, video conversion, and editing [11, 15-16, 32-34]. Multimedia applications running in the mobile phone will consume more energy and will offload to save energy to increase the battery life [33]. Some basic approaches for saving energy in cell phones,

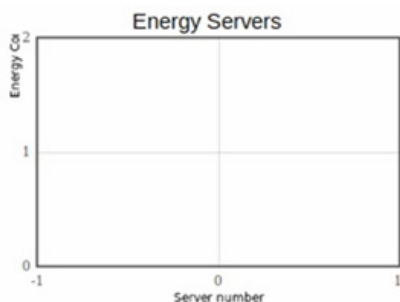
- Save the energy by turning off the mobile, when not in need.
- Execute programs slowly: which decreases the processor’s clock speed, so power is getting saved.
- Computation Offloading: Move the computation away from the mobile device, when the user wants to use more.



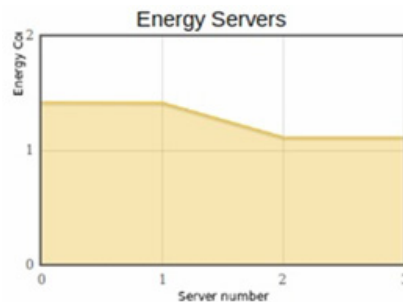
(a) Before Process Offloading



(b) After Process Offloading



(c) Before offloading the energy state



(d) After offloading the energy state

Figure 7. Green cloud simulation result

4 Conclusion

In this paper, zero knowledge proof-based authentication scheme is proposed to increase the

security of computation offloading from the mobile to the cloud server.

The presented results so far satisfy the authentication without sharing the user password to the server. The proposed authentication confirms the zero knowledge-

based technique to complete the verification mutually between the mobile and cloud server without sharing the actual password to the authentication server. Also, the simulation result of the GreenCloud proves the energy-efficient computational offloading technique saves more energy and allows to use more processing in the cloud server. The proposed authentication method is verified with Scyther tool and confirms the resistance of significant security attacks. Also, the experiments and evaluations with the GreenCloud simulator given a better result and the computation-intensive applications like multimedia conversion application will get easy to use in the smartphone with proven security.

The proposed method verifies the authentication and then, transfer the offloading task over the secure channel to the remote cloud server. Moreover, the proposed method ensures the privacy of process offloading between the communication entities.

References

- [1] S. A. Haque, S. Islam, M. J. Islam, J.-C. Grégoire, An Architecture for Client Virtualization: A Case Study, *Computer Networks*, Vol. 100, pp. 75-89, May, 2016.
- [2] M. Alizadeh, S. Abolfazli, M. Zamani, S. Baharun, K. Sakurai, Authentication in Mobile Cloud Computing: A Survey, *Journal of Network and Computer Applications*, Vol. 61, pp. 59-80, February, 2016.
- [3] S. L. Albuquerque, P. R. L. Gondim, Security in Cloud-Computing-Based Mobile Health, *IT Professional*, Vol. 18, No. 3, pp. 37-44, May-June, 2016.
- [4] J. Zhang, Z. Zhang, H. Guo, Towards Secure Data Distribution Systems in Mobile Cloud Computing, *IEEE Transactions on Mobile Computing*, Vol. 16, No. 11, pp. 3222-3235, November, 2017.
- [5] A. ur R. Khan, M. Othman, A. N. Khan, J. Shuja, S. Mustafa, Computation Offloading Cost Estimation in Mobile Cloud Application Models, *Wireless Personal Communications*, Vol. 97, No. 3, pp. 4897-4920, December, 2017.
- [6] S. Merlin, A. Chandrasekar, Towards Mobile Cloud Authentication and Gait Based Security Using Time Warping Technique, *Cluster Computing*, pp. 1-10, September, 2017.
- [7] K. Akherfi, M. Gerndt, H. Harroud, Mobile Cloud Computing for Computation Offloading: Issues and Challenges, *Applied Computing and Informatics*, Vol. 14, No. 1, pp. 1-16, January, 2018.
- [8] J. Wei, X. Hu, W. Liu, An Improved Authentication Scheme for Telecare Medicine Information Systems, *Journal of Medical Systems*, Vol. 36, No. 6, pp. 3597-3604, December, 2012.
- [9] K. Kumar, J. Liu, Y.-H. Lu, B. Bhargava, A Survey of Computation Offloading for Mobile Systems, *Mobile Networks and Applications*, Vol. 18, No. 1, pp. 129-140, February, 2013.
- [10] D. Huang, P. Wang, D. Niyato, A Dynamic Offloading Algorithm for Mobile Computing, *IEEE Transactions on Wireless Communications*, Vol. 11, No. 6, pp. 1991-1995, June, 2012.
- [11] S. Grzonkowski, A. Mosquera, L. Aouad, D. Morss, Smartphone Security: An Overview of Emerging Threats, *IEEE Consumer Electronics Magazine*, Vol. 3, No. 4, pp. 40-44, October, 2014.
- [12] W. Zhang, Z. Zhang, H.-C. Chao, Cooperative Fog Computing for Dealing with Big Data in the Internet of Vehicles: Architecture and Hierarchical Resource Management, *IEEE Communications Magazine*, Vol. 55, No. 12, pp. 60-67, December, 2017.
- [13] N. Aminzadeh, Z. Sanaei, S. H. A. Hamid, Mobile Storage Augmentation in Mobile Cloud Computing: Taxonomy, Approaches, and Open Issues, *Simulation Modelling Practice and Theory*, Vol. 50, pp. 96-108, January, 2015.
- [14] S. Grzonkowski, P. M. Corcoran, Sharing Cloud Services: User Authentication for Social Enhancement of Home Networking, *IEEE Transactions on Consumer Electronics*, Vol. 57, No. 3, pp. 1424-1432, August, 2011.
- [15] I. Elgendy, W. Zhang, C. Liu, C.-H. Hsu, An Efficient and Secured Framework for Mobile Cloud Computing, *IEEE Transactions on Cloud Computing*, pp. 1-10, June, 2018.
- [16] S. Guo, J. Liu, Y. Yang, B. Xiao, Z. Li, Energy-Efficient Dynamic Computation Offloading and Cooperative Task Scheduling in Mobile Cloud Computing, *IEEE Transactions on Mobile Computing*, Vol. 18, No. 2, pp. 319-333, February, 2019.
- [17] M. Shiraz, A. Gani, A. Shamim, S. Khan, R. W. Ahmad, Energy Efficient Computational Offloading Framework for Mobile Cloud Computing, *Journal of Grid Computing*, Vol. 13, No. 1, pp. 1-18, March, 2015.
- [18] M. B. Mollah, M. A. K. Azad, 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, April, 2017.
- [19] F. Berg, F. Dürr, K. Rothermel, Increasing the Efficiency of Code Offloading in n-tier Environments with Code Bubbling, *Mobile Networks and Applications*, Vol. 23, No. 5, pp. 1364-1375, October, 2018.
- [20] D. Huang, H. Wu, *Mobile Cloud Computing*, Morgan Kaufmann, 2018.
- [21] F. Hao, P. Ryan, J-PAKE: Authenticated Key Exchange without PKI, *Transactions on Computational Science XI-Lecture Notes in Computer Science*, Springer-Heidelberg, 2010.
- [22] A. Miller, *Zero-Knowledge Proof Notation and Vocabulary*, Lecture Series- Zero Knowledge Proofs- Cryptocurrency Security, 2016.
- [23] D. Huang, H. Wu, Mobile Cloud Security: Attribute-Based Access Control, *Mobile Cloud Computing*, Morgan Kaufmann, 2018.
- [24] A. Alzahrani, N. Alalwan, M. Sarrab, Mobile Cloud Computing: Advantage, Disadvantage and Open Challenge, *7th Euro American Conference on Telematics and Information Systems (EATIS '14)*, Valparaiso, Chile, 2014,

Article No. 21.

- [25] M. C. Murphy, M. McClelland, Computer Lab to Go: A “Cloud” Computing Implementation, *ISECON/CONISAR 2008*, Phoenix, Arizona, 2008, pp. 1-10.
- [26] K. Sinha, M. Kulkarni, Techniques for Fine-Grained, Multi-site Computation Offloading, *2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID '11)*, Newport Beach, CA, USA, 2011, pp. 184-194.
- [27] Q. B. Hani, J. P. Dichter, Secure and Strong Mobile Cloud Authentication, *2016 SAI Computing Conference (SAI)*, London, UK, 2016, pp. 562-565.
- [28] S. Grzonkowski, P. M. Corcoran, T. Coughlin, Security Analysis of Authentication Protocols for Next-generation Mobile and CE Cloud Services, *2011 IEEE International Conference on Consumer Electronics -Berlin (ICCE-Berlin)*, Berlin, Germany, 2011, pp. 83-87.
- [29] M. AbdelAty, A. Mokhtar, A Computational Offloading Framework for Object Detection in Mobile Devices, *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2017*, Cairo, Egypt, 2017, pp. 97-107.
- [30] S. Kosta, A. Aucinas, P. Hui, R. Mortier, X. Zhang, ThinkAir: Dynamic Resource Allocation and Parallel Execution in the Cloud for Mobile Code Offloading, *2012 Proceedings IEEE INFOCOM*, Orlando, FL, 2012, pp. 945-953.
- [31] T. Steiner, *An Introduction to Securing a Cloud Environment*, Sans Institute, Information Security Reading Room, November, 2012.
- [32] Z. Ahmad, K. E. Mayes, S. Dong, K. Markantonakis, Considerations for Mobile Authentication in the Cloud, *Information Security Technical Report*, Vol. 16, No. 3-4, pp. 123-130, August-November, 2011.
- [33] B. L. J. Jun, Implementing Zero-Knowledge Authentication with Zero Knowledge, *The Python Papers Monograph*, Vol. 2, Article No. 9, 2010.
- [34] P. England, J. Manferdelli, Virtual Machines for Enterprise Desktop Security, *Information Security Technical Report*, Vol. 11, No. 4, pp. 193-202, June, 2006.



Kannammal A, completed her Ph.D. in Engineering from Anna University Chennai in 2014, M.E. degree from Thiagarajar College of Engineering, Madurai, India in 2004. B.E. degree in ECE, in 2002. Presently she is working as Associate Professor in the Dept. of Electronics and Communication Engineering at PSG College of Technology, Coimbatore, India. Her interests include Medical Image Processing and Medical image security.

Biographies



Munivel E, Completed M.E. from Anna University, Chennai, India in 2008, followed by Master of Science in Computer Science, in 2006. Currently working as Scientist at National Institute of Electronics and Information Technology (MeitY, Govt. of India). His interests include Virtualization, Cloud Infrastructure, System Security, Mobile Cloud Security and Virtual Training Environment.

