Robust Anonymous Mutual Authentication for Mobile Cloud Computing Services

Alampally Sreedev[1], Kavithagopu2

12Assistent Professor CSE, Sri Indu College of Engineering and Technology, Hyderabad, Telangana, India

Abstract

With the combination of mobile computing technology and cloud computing, mobile users have the advantage of quality, resources, and access handiness to the web to communicate data seamlessly anyplace at any time. However, the transfer of information through vulnerable channels poses security threats like man-in-middle and sniffing. This paper explores, reviews, and analyses many existing works associated with authentication in mobile cloud computing. This paper additionally aims to propose an authentication theme to secure data communication in mobile cloud computing. The proposed scheme is based on multi-factor authentication that uses usernames, passwords, and a One-Time secret. The One-Time secret is encrypted with a Diffie-Hellman key exchange algorithm to achieve mutual authentication between the mobile device and therefore the cloud server. The proposed scheme will be implemented and tested using Java.

Keywords: **mobile cloud computing, authentication, Diffie-Hellman, one-time password**

1. Introduction

With the increasing number of important and sensitive personal and business data being shared on the Internet every day, security in networking has become essential. Internet attackers are constantly formulating new strategies and tactics to disrupt online user services. [1]– [3] The technology for network security must improve and evolve to secure data communication on the Internet from the threat of network intrusions. [4]– [6] Organizations today rely on cloud computing services to fulfill their business goals and provide their customers with the highest accessibility to data sources. This service allows organizations and their customers to upload their data into a cloud server for retrieval at any time

[7]. Data communication between the server and client allows the transmission of digital data regardless of geographical location, type of data content, and technological medium.

Mobile Cloud Computing (MCC) is the combination of cloud computing, mobile computing and wireless networks to bring rich computational resources to mobile users, network operators, as well as cloud computing providers.[1][2][3] The ultimate goal of MCC is to enable execution of rich mobile applications on a plethora of mobile devices, with a rich user experience.[4] MCC provides business opportunities for mobile network operators as well as cloud providers.[5][6] More comprehensively, MCC can be defined as "a rich mobile computing technology that leverages unified elastic resources of varied clouds and network technologies toward unrestricted functionality, storage, and mobility to serve a multitude of mobile devices anywhere, anytime through the channel of Ethernet or Internet regardless of heterogeneous

Environments and platforms based on the pay-as-you-use principle."[7]
\ although significant research and development in MCC is available in the literature, efforts in the following domains is still lacking: [3] [7]

- **Architectural issues:** Reference architecture for heterogeneous MCC environment is a crucial requirement for unleashing the power of mobile computing towards unrestricted ubiquitous computing.

- **Energy-efficient transmission:** MCC requires frequent transmissions between cloud platform and mobile devices, due to the stochastic nature of wireless networks, the transmission protocol should be carefully designed.[11][12]
- **Context-awareness issues:** Context-aware and socially-aware computing are inseparable traits of contemporary handheld computers. To achieve the vision of mobile computing among heterogeneous converged networks and computing devices, designing resource-efficient environment-aware applications is an essential need.
- **Live VM migration issues:** Executing resource-intensive mobile application via Virtual Machine (VM) migration-based application offloading involves encapsulation of application in VM instance and migrating it to the cloud, which is a challenging task due to additional overhead of deploying and managing VM on mobile devices.
- **Mobile communication congestion issues:** Mobile data traffic is tremendously hiking by ever increasing mobile user demands for exploiting cloud resources which impact on mobile network operators and demand future efforts to enable smooth communication between mobile and cloud endpoints.
- **Trust, security, and privacy issues:** Trust is an essential factor for the success of the burgeoning MCC paradigm. It is because the data along with code/component/application/complete VM is offloaded to the cloud for execution. Moreover, just like software and mobile application piracy, the MCC application development models are also affected by the piracy issue.[10] Pirax[10] is known to be the first specialized framework for controlling application piracy in MCC environment.

Mutual authentication or **two-way authentication** refers to two parties authenticating each other at the same time, being a default mode of authentication in some protocols (IKE, SSH) and optional in others (TLS).

By default the TLS protocol only proves the identity of the server to the client using X.509 certificate and the authentication of the client to the server is left to the application layer. TLS also offers client-to-server authentication using client-side X.509 authentication.[1] As it requires provisioning of the certificates to the clients and involves less user-friendly experience, it's rarely used in end-user applications.

Mutual TLS authentication (**mTLS**) is much more widespread in business-to-business (B2B) applications, where a limited number of programmatic and homogeneous clients are connecting to specific web services, the operational burden is limited and security requirements are usually much higher as compared to consumer environments.

Better institution-to-customer authentication would prevent attackers from successfully impersonating financial institutions to steal customers' account credentials; and better customer-to-institution authentication would prevent attackers from successfully impersonating customers to financial institutions in order to perpetrate fraud

— *Financial Services Technology Consortium, 2005*

Most Mutual authentication is machine-to-machine, leaving it up to chance whether or not users will notice (or care) when the remote authentication fails (e.g. a red address bar browser padlock, or a wrong domain name). Non-technical mutual-authentication also exists to mitigate this problem, requiring the user to complete a challenge, effectively forcing them to notice, and blocking them from authenticating with a false endpoint.

Mutual authentication is two types:

1. Certificate based
2. User name-password based

Nevertheless, with these advantages mobile cloud computing is exposed to security threats. Security threats in mobile cloud computing is a concern to mobile users as sensitive personal information such as identity, location, job, and biometrics can be easily targeted by attackers with malicious intent. Poor security administration constrains the development and deployment of cloud connected security-sensitive applications in areas such as social media, medicine, finance, and e-government services. Security challenges in mobile cloud computing include securing mobile users and the data stored on the cloud [10]. The security issue focused on in this paper is authentication in mobile cloud computing. Despite many authentication schemes being proposed in recent years, most authentication proposed methods still lack mutual authentication between mobile devices and cloud servers [11].

The lack of mutual authentication in mobile cloud computing makes it vulnerable to man-in-the-middle attacks where attackers intercept communications and manipulates messages between the mobile user and the cloud server. The attacker can also impersonate the mobile user or the cloud server. When the security of a communication channel is breached, the confidentiality and integrity of the transmitted data is exposed to vulnerabilities. Thus, it is crucial to have mutual authentication that secures data communication between mobile devices and cloud servers. Mutual authentication is a process that verifies the identity of both the mobile device and cloud server as genuine before they are authorized for data communication.

2.Related work

This part will discuss two authentication schemes, which are authentication on the cloud server side and authentication on the user side. Figure 1 shows the classification of authentication schemes in mobile cloud computing.

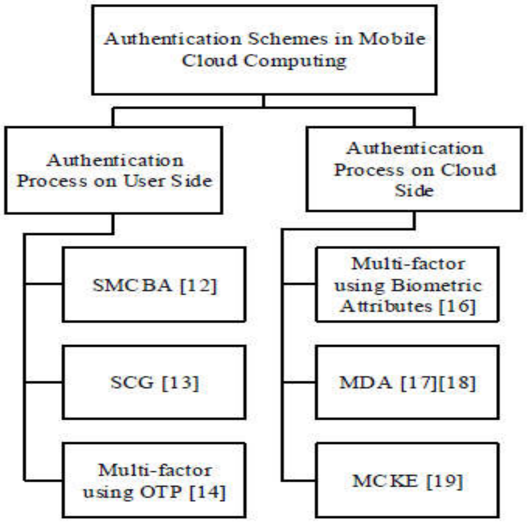

Authentication Schemes in Mobile Cloud Computing
Authentication Process on User Side
Authentication Process on Cloud Side
SMCBA [12]
SCG [13]
Multi-factor using OTP [14]
Multi-factor using Biometric Attributes [16]
MDA [17][18]
MCKE [19]

Fig 1: Classification of Authentication Schemes in Mobile Cloud Computing

A. Authentication on the User-Side

In user -side authentication, authentication operations are carried out by a mobile device. The user inputs its information at the browser or application level for it to be processed. The processed information is then verified and sends to the server to be authenticated.

A biometric authentication mechanism that uses fingerprint recognition systems to secure mobile cloud computing was proposed in [12]. The proposed authentication mechanism uses existing cameras in mobile phones to capture the fingerprint image of a cloud user. Hence, this mechanism does not require any extra sensors to be implemented into the mobile device. However, a high-quality camera is required for capturing an accurate fingerprint image for operations to be carried out. The user captures a fingerprint image using a mobile phone camera. The fingerprint image undergoes image processing such as converting an RGB image into a gray-scale image, reduced blurring, segmentation, and ridge enhancement. The processed image is then sent in a core-point detection phase where feature extraction of the fingerprint image is carried out. Finally, the cloud server checks whether the extracted fingerprint image matches the one that is stored in its database. If it matches, the user is verified and authenticated for the cloud server.

A private authentication system proposed in [13] uses a Smart Card Generator (SCG). The proposed system uses a dynamic nonce generation and bilinear pairing cryptosystem techniques. According to the researchers, the technique reduces the complexity of discrete logarithm problems. To set up the authentication system, the SCG selects a random number as a master private key and computes a public key to generate all other public parameters. It then publishes the generated public key and public parameters. The registration phase is executed after system set up is complete. Mobile users or service providers register to the SCG by providing their information while the SCG computes and securely sends the respective private keys to the mobile user and the service provider. When the service provider and mobile user want to communicate, a card provided by the trusted SCG is used to authenticate both parties. The disadvantage of this system is that there is a risk of losing the card, and the card is necessary for both the mobile client and the cloud service provider to authenticate each other.

A Multi-factor authentication method using mobile phones was proposed in [14]. The proposed method uses a mobile phone as a software token for a One-Time Password (OTP) and an additional SMS-based authentication system. When the user wants to login to a secure website, the user starts the authentication software on a mobile device. After entering their username and password, the software generates a one-time password for the user that is valid for ten minutes. The OTP is then used by the user to authenticate the website for login access. The one-time password generated in this research uses factors such as username, password, IMEI, and IMSI, which were concatenated and hashed using SHA-256.

For the SMS -based authentication system, when the user wants to login to a secure website, the user is required to send an encrypted SMS to the server. The server receives the SMS and decrypts the message into four parts, which are the sender's mobile number, username, pin, and IMEI number. The server checks its database to see if the information is genuine. If the user's information is valid, the server generates a password, encrypts it, and sends the password back to the user via SMS. According to the researchers, this authentication system saves organizational costs of purchasing and maintaining hardware tokens as it uses software tokens for verification. However, the one-time password is not encrypted and payment charges are implied when the SMS-based authentication is used. In general, the secure generation and transmission of a one-time password is challenging [15].

B. Authentication on the Cloud-Side

Authentication on the cloud side means that most of the authentication operations are carried out in the cloud server. The inputs submitted by a user are sent to the server to be verified. After the user data is verified, the server will provide access to the user in a secure communication channel. Several types of authentication systems carried out in cloud servers are elaborated on in this section.

An authentication system for smart devices using multi factors in a mobile cloud service was proposed in [16]. The system uses ID/password, International Mobile Equipment Identity (IMEI), International Mobile Subscriber Identity (IMSI), voice recognition, and face recognition as the authentication parameters. Based on Fig. 2, the authentication process begins when a user input their information to be sent to the management server. The management server then checks the VMs load on the clustered host, calculates loads, and performs load balancing.

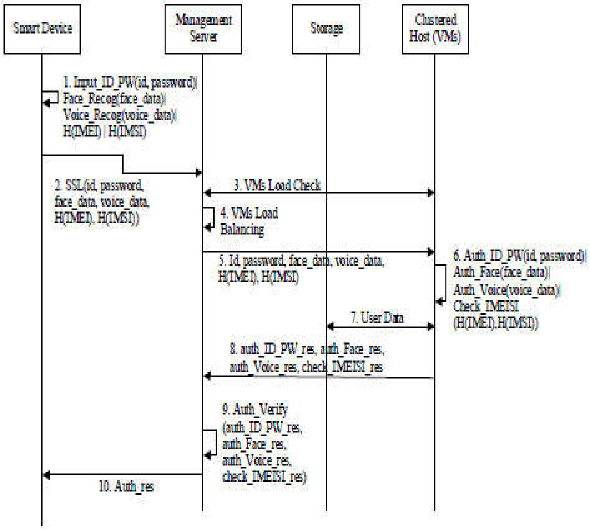


Fig 2: Authentication Process in Multi-Factor Environment [16]

The calculated load is sent to the clustered host for the Virtual Machines (VM) to authenticate the information given by the smart device user. If there is an information overlap with a user who registered earlier, the authentication process will fail and the existing user will be notified of this attempted security breach. The result of the authentication process is sent to a management server who then returns the final authentication result with the user's authentication values to the smart device. This proposed system enhances authentication performance as the factors are processed in bulk by the cloud server, but there is no mutual authentication between mobile users and the cloud server. The system also lacks usability and privacy as it requires multiple types of sensitive data.

The Message Digest-based Authentication (MDA) scheme was proposed in [17] [18]. This authentication scheme consists of two phases, one in which the cloud authenticates the mobile client and another in which the mobile client authenticates the cloud. Upon the creation of a cloud account, a key To is generated by XOR-in (Exclusive OR) a hashed user ID and user password. Based on Fig. 3, the cloud user has two message digests, which are Mouser and McLeod, in the mobile device.

For the cloud server to authenticate the mobile device, the mobile device needs to send an authentication request to the cloud server. Using PRNG, an authentication key or Auth_ key is generated using a seed Tk and a state identifier or SI to encrypt the hash value of the MDcloud and MDuser into an MD. The mobile node then sends the encrypted message, #CF

|| ETk { Eauth_keyi { MD } || SI } to the cloud server. #CF is the column reference for the database in the cloud server. After the cloud server receives this message, it searches for the specific hashed password and userID using the column reference #CF as shown in Fig. 4. Once the hashed password and userID are found, the key Tk is generated to decrypt the messages ETk { Eauth_keyi { MD} || SI } to obtain the state identifier or SI. The key Tk and SI are used by PRNG to generate the Auth_key, which is then used to decrypt the authenticated message Eauth _keyi { MD } to obtain MD. If the messages MD and MD' matches, the authenticity of the mobile device is verified.

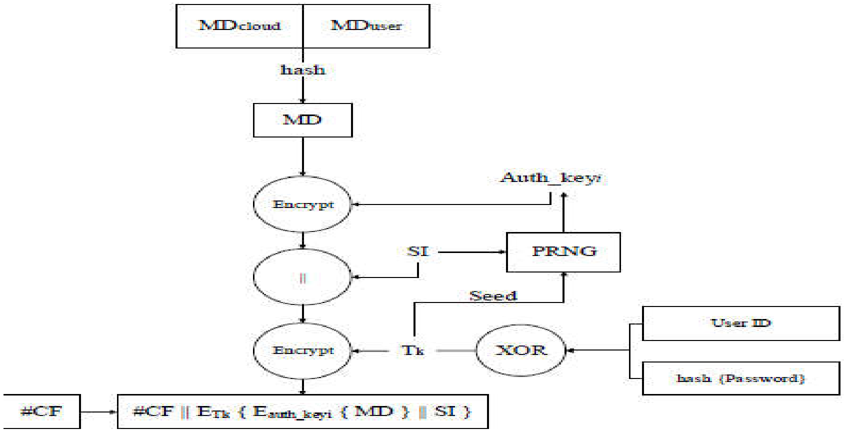


Fig 3: Mobile device sends an authentication request to the cloud server[17][18]

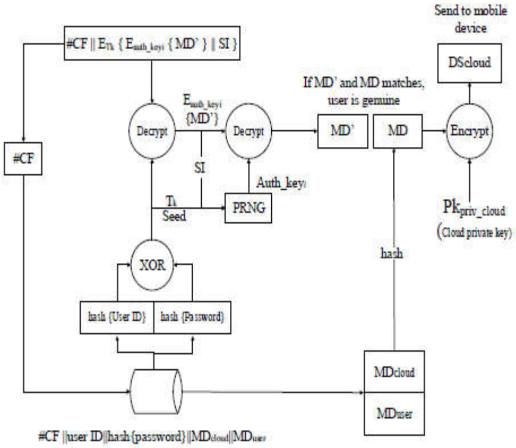


Fig 4: The cloud server authenticates the mobile device [17][18]

Based on Fig. 5, the second phase of the authentication begins where the mobile device verifies and authenticates the cloud server. The cloud server will send a digital signature encrypted by its own private key, Pkpriv_cloud to the mobile device. After receiving the digital signature, the mobile device will decrypt it with the cloud’s public key, Pkpub_cloud. If the decrypted MD matches the mobile device MD, then the authenticity of the cloud server is verified.

This study proposed a high security authentication scheme that provides mutual authentication. However, the authentication operations involve many processes such as the generation of random keys and authentication keys, the hashing of message digests, and the encryption and decryption of the message digest that will be carried out in the respective both parties. There are numerous messages being transmitted between the mobile device and the cloud server, which makes the message digest authentication scheme less efficient [11].

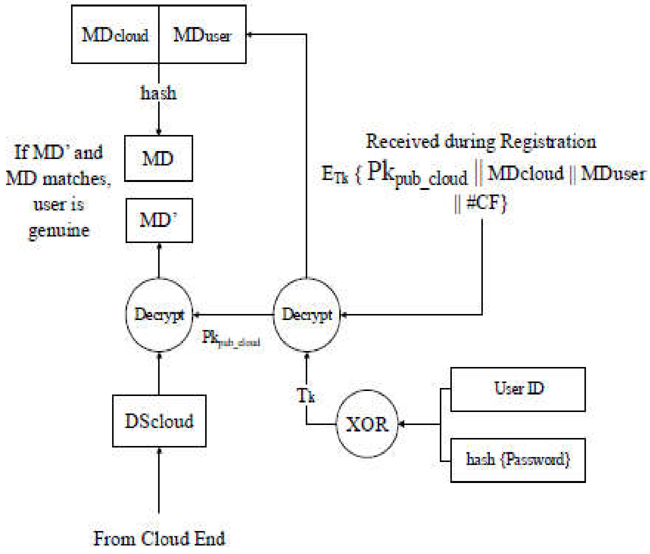

MDcloud
MDuser
hash
MD
If MD' and MD matches, user is genuine
MD'
Received during Registration
E_Tk { Pk_pub_cloud || MDcloud || MDuser || #CF}
Decrypt
Decrypt
Pk_pub_cloud
Tk
XOR
User ID
hash {Password}
DScloud
From Cloud End

Fig 5: The mobile device authenticates the cloud server [17][18]

An authentication system called Mobile Cloud Key Exchange was proposed in [19]. This system is based on the randomness-reuse strategy and the Internet Key Exchange (IKE) scheme. The researchers discuss the various authentication technologies that are used by mobile clouds such as basic authorization, signature tokens, open ID, and open Auth (OAUTH). The authentication system is setup from a Diffie-Hellman group using a large prime integer and generator number for the group that acts as a primitive root modulo.

The system uses certificate authority in a public key exchange to ensure that communication between two parties can be verified. The MCKE system is used when it receives a new task so the cloud controller (CLC) will pick a secret value, compute a public key, and broadcast the message to the domain of the server for a Diffie-Hellman key exchange to take place. The session key is now shared between the CLC and the server to encrypt their communications. For MCKE to be completed, CLC needs to generate signatures using a secret key generated from a key pair issued by the certificate authority. The signature will then be sent to the server for verification to take place. The public key contained in the certificate is used to verify the signature. Likewise, for verification in CLC, the server will also need to send its own signature, encrypted ID, and certificate to the CLC. According to the researchers, the proposed authenticated key exchange scheme reduces time consumption and computation load but is costly to implement. The researches mentioned that new strategies for providing better security-aware scheduling are required furthering enhancing symmetric-key encryption.

From the analysis of related works, an authentication scheme for this research titled “Diffie-Hellman based One-Time Password to Secure Data Communication in Mobile Cloud Computing” is proposed.

This research will focus on mitigating the limitations faced by existing studies by proposing an authentication scheme that can provide mutual authentication in a mobile cloud computing environment.

3. propose work

The proposed multi-factor authentication scheme in this research requires login credentials and a one-time password. This authentication method acts as an extra security layer for mobile users and is commonly used in the current advancement of mobile application technology. An example of multi-factor authentication can be observed during an Internet banking transaction. When a bank customer wants to make an online payment, he is prompted to fill in his login credentials, which is the first factor of authentication. To complete his payment, he is required to fill in a special pin that is send to his mobile device via SMS, which is the second factor of authentication. This process verifies that the bank customer is genuine based on “what he knows” (username and password) and “what he has” (SMS send to his mobile device). The usage of multi-factor authentication reduces the number of identity thefts on the Internet besides being exposed to phishing websites.

Therefore, the multi-factor authentication scheme of this research utilized something that the mobile user knows and something that the mobile user possesses. The first factor of this proposed authentication scheme is the login credentials of the mobile user, which is their username and password. The second factor is a one-time password locally generated by the mobile device, which acts as a software token. For this research, a software token is used because it is cheap compared to a hardware token in the long run. The generation of this one-time password does not require any extra hardware from the mobile user, which provides convenience.

There are two methods of generating OTPs, which are based on time synchronization and mathematical algorithms. In this paper, we will use a one-time password based on time synchronization [20]. The one-time password is computed using the current time and shared factors such as username, password, and the International Mobile Equipment Identity (IMEI) of the mobile device. The shared factors are concatenated with the current time to produce a one-time password that will be encrypted using the Advanced Encryption Standard (AES) algorithm. There will be a time interval for the validity of the one-time password which ranges from 30 seconds to 2 minutes. After the time interval is over, the one-time password will become invalid.

Both the client and server must compute the one-time password generation locally. The server will then check whether the one-time password received from the client

Matches the one that it generated. However, before the client sends the one-time password to the server, the one-time password is encrypted using a shared key that is exchanged between the mobile client and the cloud server using a Diffie-Hellman algorithm. Diffie-Hellman key exchange is a method of exchanging cryptographic keys on an insecure channel [21][22]. This Diffie-Hellman key exchange allows two parties that do not know each other to establish a shared secret key over a public channel.

Mobile users are required to register and sign up for an account with the cloud service provider to store their information in the server database. The Diffie-Hellman key exchange process is carried out after the client is connected to the server. The computation of public and private keys generates a symmetric key cipher that is used to encrypt the data exchange between the two parties. Hence, the generation and transmission of a one-time password in this authentication scheme is secured. Based on related works, most authentication methods do not provide mutual authentication. Hence, they are vulnerable to man-in-middle and sniffing attacks. When the communication channel is under attack, the data transferred is insecure, which causes privacy and confidentiality issues. This research proposes the use of multi-factors authentication and a one -time password encrypted with a Diffie -Hellman secret key to provide mutual authentication for both the mobile client and cloud server.

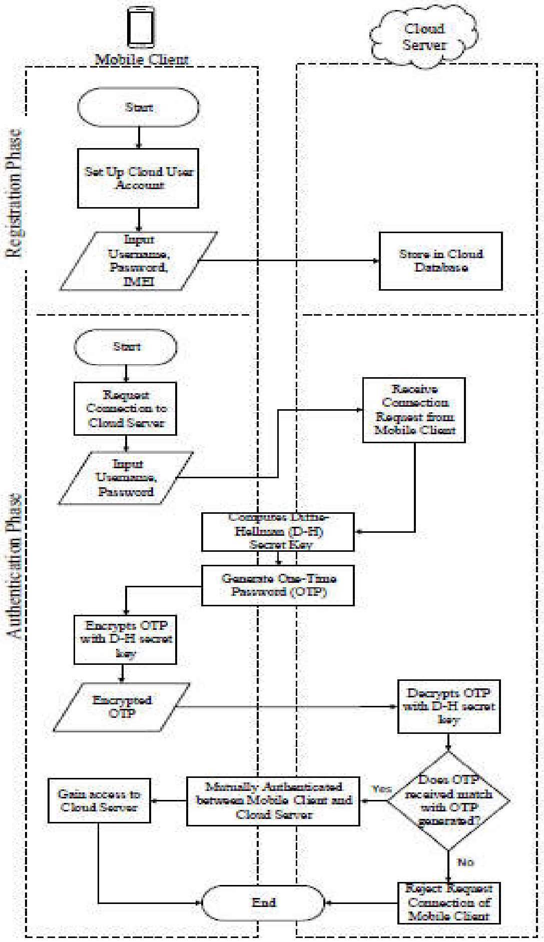
A. Design and Implementation

The proposed authentication scheme uses multi-factor authentication that is knowledge-based and possession-based. There are two phases in this proposed authentication scheme, registration and authentication. In the registration phase, the mobile client is required to set up an account on a cloud server by registering his mobile device information, username, and password. This information is stored in the cloud server database.

After the registration phase, when the mobile user wants to communicate with the cloud server, he needs to send a connection request to the cloud server, which will prompt the mobile user to enter their username and password. Upon entering his username and password, the computation of Diffie-Hellman secret keys by the mobile client and cloud server take places. Then, both the mobile client and cloud server will locally generate a one-time password at the same time. The mobile client uses the Diffie-Hellman secret key to encrypt its one-time password to be sent to the cloud server. Upon receiving the encrypted one-time password, the cloud server decrypts it using the Diffie-Hellman secret key. The cloud server then checks whether the one-time password that it receives from the mobile device matches the one-time password that is generated locally in the cloud server. If the one-time password sent by the mobile device matches the server's locally generated one-time password, then the mobile device and the cloud server are mutually authenticated. The mobile client gain access to the cloud server. However, if the one-time passwords do not match, the connection request of the mobile client is rejected and the authentication phase ends. Figure 6 shows a flowchart for the proposed authentication scheme. The authentication scheme will be simulated using Java in a real experiment. The proposed authentication scheme will be carried out by simulating a client and a server using Net beans software.

B. Testing and Evaluation

For this research, an experiment using fake mobile user accounts communicating with a cloud server will be set up. The mobile client will generate 10 OTPs at different times of the day and these will be tested to see if they matched the OTPs of the server. The factors for generating the OTPs such as username, password, and IMEI will be used to manipulate the experimental setup. Another experiment that checked the chances of generating identical OTPs with fake users will also be conducted. These experiments will be conducted to evaluate whether the authentication scheme provides mutual authentication based on encrypted OTPs with Diffie-Hellman secret keys.


Mobile Client
Cloud Server
Registration Phase
Start
Set Up Cloud User Account
Input Username, Password, IMEI
Store in Cloud Database
Authentication Phase
Start
Request Connection to Cloud Server
Input Username, Password
Receive Connection Request from Mobile Client
Computes Diffie-Hellman (D-H) Secret Key
Generate One-Time Password (OTP)
Encrypts OTP with D-H secret key
Encrypted OTP
Decrypts OTP with D-H secret key
Does OTP received match with OTP generated?
Yes
No
Mutually Authenticated between Mobile Client and Cloud Server
Gain access to Cloud Server
Reject Request Connection of Mobile Client
End

ig 6: Flowchart of the proposed authentication scheme

Conclusion

Mobile cloud computing incorporates a very massive potential user base as owning a mobile device connected to the web has become important in today's wireless world. Mobile cloud computing provides several advantages to users in terms of resources, convenience, and quality. However, security in mobile cloud computing creates challenges in having mutual authentication between the consumer and server besides providing an economical authentication method. Future works can focus on implementing and testing the proposed authentication theme.

.

REFERENCES

[1] A. A. Ahmed, A. Jan tan, and T. C. Wan, "Filtration model for the detection of malicious traffic in large-scale networks," *Comput. Commun.*, vol. 82, pp. 59–70, 2016.

[2] A. A. Ahmed and N. A. K. Zaman, "Attack intention recognition: A review," *Int. J. Network Security*, vol. 19, no. 2, pp. 244–250, 2017.

[3] A. A. Ahmed, A. Jantan, and T. C. Wan, "Real-time detection of intrusive traffic in QoS network domains," *IEEE Security & Privacy*, vol. 11, no. 6, pp. 45–53, 2013.

[4] A. A. Ahmed and Y. W. Kit, "MICIE◌ள: A Model for Identifying and Collecting Intrusion Evidences," *2016 IEEE 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) Italy*, pp. 288-294, 2016.

[5] A. A. Ahmed, A. Jantan, and M. Rasmi, "Service violation monitoring model for detecting and tracing bandwidth abuse," *Journal of Network and Systems Management*, vol. 21, no. 2, pp. 218–237, 2013.

[6] A. A. Ahmed, "Investigation Model for DDoS Attack Detection in Real-Time," *International Journal of Software Engineering and Computer Systems*, vol. 1, no.1, pp. 93–105, 2015.

[7] A. A. Ahmed and C. X. Li, "Locating and Collecting Cybercrime Evidences on Cloud Storage: Review," *2016 International Conference on Information Science and Security (ICISS) Thailand*, pp. 1-5, 2016.

[8] D. S. Yadav and P. K. Doke, "Mobile Cloud Computing Issues and Solution Framework," pp. 1115–1118, 2016.

[9] P. N. Dharmale and P. L. Ramteke, "Mobile Cloud Computing,"

[10] M. Tank, Nirbhay K. Chaubey, "Security, Privacy and Challenges in Mobile Cloud Computing (MCC):- A Critical Study and Comparison," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 4, no. 6, pp. 2257–2263, 2016.

[11] M. Alizadeh, S. Abolfazli, M. Zamani, S. Baaaharun, and K. Sakurai, "Authentication in mobile cloud computing: A survey," *J. Netw. Comput. Appl.*, vol. 61, pp. 59–80, 2016.

[12] I. Al Rassan and H. AlShaher, "Securing Mobile Cloud Computing Using Biometric Authentication (SMCBA)," *2014 Int. Conf. Comput. Sci. Comput. Intell.*, pp. 157–161, 2014.

[13] K. K. Kavitha, B. L. Gopinath, C. U. Kushalappa, and D. K. H, "Mobile Cloud Computing With A Private Authentication Scheme,"172–176, 2016.

[14] F. Aloul, S. Zahidi, and W. El-hajj, "Multi Factor Authentication Using Mobile Phones," *Int. J. Math.*, vol. 4, no. 2, pp. 65–80, 2009

[15] W. T. Meshach and K. S. S. Babu, "Secured and Efficient Authentication Scheme for Mobile Cloud (2013)," vol. 2, no. 1, pp. 242–248, 2013.

[16] D. M'Raihi, S. Machani, M. Pei, and J. Rydell, "RFC 6238 - TOTP: Time-Based One-Time Password Algorithm,", *Tools.ietf.org*, 2011. [Online]. Available: https://tools.ietf.org/html/rfc6238. [Accessed: 27 October 2017]

[17] W. Diffie, W. Diffie, and M. E. Hellman, "New Directions in Cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644– 654, 1976.

[18] M. P. Rewagad and M. Y. Pawar, "Use of digital signature with diffie hellman key exchange and aes encryption algorithm to enhance data security in cloud computing," *Proc. - 2013 Int. Conf. Commun. Syst. Netw. Technol. CSNT 2013*, pp. 437–439, 2013.