MATEC Web of Conferences **40**, 09002 (2016)
DOI: 10.1051/matecconf/20164009002

Key Based Mutual Authentication (KBMA) Mechanism for Secured Access in MobiCloud Environment

A. Cecil Donald[1] and Dr. L. Arockiam[2]

[1]*Doctoral Research Scholar, Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India*
[2]*Associate Professor, Department of Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli, Tamil Nadu, India*

Abstract. Mobile Cloud Computing (MCC) fuels innovation in Mobile Computing and opens new pathways between mobile devices and infrastructures. There are several issues in MCC environment as it integrates various technologies. Among all issues, security lies on the top where many users are not willing to adopt the cloud services. This paper focuses on the authentication. The objective of this paper is to provide a mechanism for authenticating all the entities involved in accessing the cloud services. A mechanism called Key Based Mutual Authentication (KBMA) is proposed which is divided into two processes namely registration and authentication. Registration is a one-time process where the users are registered for accessing the cloud services by giving the desired unique information. Authentication process is carried out mutually to verify the identities of Device and Cloud Service Provider (CSP). Scyther tool is used for analysing the vulnerability in terms of attacks. The result claims show that the proposed mechanism is resilient against various attacks.

1 Introduction

It has been observed that the significant technologies like Cloud Computing and Mobile devices are on trend for the last few years. The extensive adoption of these two technologies are changing our lives. Researchers and data analysts show exactly how these technologies have extremely created a reverberation in the technological landscape around the world. The cloud is already playing a much larger role in Information Technology. By the end of 2015, the propagation of Mobile Cloud Networking (MCN) development will become a natural extension of the Bring Your Own Device (BYOD) environment. According to the recent survey from Juniper Research, the number of mobile cloud computing users are expected to grow promptly in the next 5 years [1]. In 2016, it is expected that mobile cloud market will produce annual revenue of $9.5 billion from $400 million (2009), at an average annual increase of 88%.

Mobile Cloud Computing (MCC) is an emerging technology where there are several issues as it integrates the three trending concepts, Mobile Computing, Cloud Computing and Networks. There are several benefits in using the MCC services. But, due to the constraints present in the devices, there are several issues in MCC environment. Among all the issues, security lies on the top where many users are not willing to adopt the services. In order to access the cloud, the user has to connect their mobile to mobile station. Then, the mobile station connects it to its internet server, which allows the user to access its cloud server. There are a numerous number of intermediate servers used for accessing the internet which makes the security a big concern. The major security issues are Authentication, Access Control, Availability, Confidentiality, Identity Management, Integrity, Application Security and Privacy.

This paper addresses only the security issues present in Mobile Cloud Computing Environment, especially Authentication. The rest of the paper is organized as follows. Section 2 discusses the related works. Section 3 provides the motivation and section 4 delivers the objective of the proposed work. Section 5 and section 6 explains the working concept and algorithm of the proposed mechanism. Section 7 discusses the results and findings of the work. Finally, Section 8 concludes the paper.

2 Related Works

Cloud computing users prove their identities with digital credentials, typically passwords and digital certificates. If an attacker could fake or steal these credentials, the cloud system will suffer from spoofing attacks [2]. In MCC, the problem is even severe because mobile devices often lack computing power to execute complex security algorithms. Moreover, it is difficult to enforce a standardized credential protection mechanism due to the variety of mobile devices.

In recent years, several authentication mechanisms for cloud environment have been proposed to withstand against the attacks. Omri et al. [3] presented an application that uses handwriting recognition as an

Article available at http://www.matec-conferences.org or http://dx.doi.org/10.1051/matecconf/20164009002

authentication pattern to access mobile cloud. Rassan et al. [4] projected a solution for authenticating mobile cloud users using the normal mobile device camera as a fingerprint reader to get the fingerprint image, process and realize it. Based on the activity logs, cloud security policies shall be revised and re-configured. Deepak et al. [5] proposed an authentication algorithm to verify the authenticity of the user. The proposed algorithm stands against the Brute Force attack and Man-in-the Middle attack but increases the computation time and consumes more energy than conventional methods.

3 Motivation

From the security perspective, all interfaces have the danger of exposing sensitive information and receiving malicious data. In addition, eavesdropping and spoofing are easier in wireless networks than wired network. Dealing with threats is a major challenge. Mobile cloud is highly virtualized and highly federated in nature. Most of the users expect to access the cloud resources and applications without any complex authentication process. Users are not willing to carry any external devices for authentication and also expect positive user experience. Users are willing to use their existing mobile devices for authentication. Several authentication mechanisms focus only on authenticating the users but not the entities participating in accessing the cloud services. Thus, an approach needs to be developed to control and manage identities across different clouds.

4 Objective

The primary objective of this paper is to provide a mechanism for authenticating all the entities participating in accessing the cloud services (i.e. Mutual Authentication). This will be helpful in providing secure communications over the internet and to authenticate the identity of each other in a secure manner.

5 Working of MobiCloud

The primary mode of mobile device communication is HTTP over Wi-Fi while the communication between the Unified Cloud Authenticator and the Cloud Service Provider (CSPs) is over HTTPS.

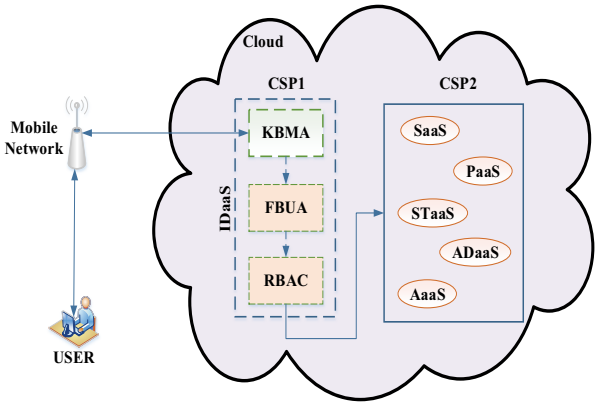

Figure 1. Skeleton Architecture of MobiCloud

Figure 1. Shows the skeleton architecture of the MobiCloud environment which holds the Identity as a Service (IDaaS) and the other services like SaaS, IaaS, STaaS, AaaS, etc. The user is allowed to access the cloud resources only after completing all the three processes given in IDaaS. This paper details only the KBMA mechanism which authenticates the entities participating in accessing the cloud services (Device and CSP).

The working procedure of the Mobile Cloud Technology is given as follows [6]. Initially, the user's requests are communicated over the Mobile Networks (MN) which holds the Base Station (BS), AAA sever and a user repository. Base Station acts as the transmitter and the AAA server authenticates the requests in the MN. Then, the requests are directed to the Unified Cloud Authenticator (UCA). The UCA plays a major role in security which contains the Authentication Server (AS) for authenticating not only the users but also their roles for accessing their respective services. The user repository is the place where all the user credential data are stored. Another major component is Cloud Service Provider (CSP) who provides the service to the users, which obviously holds the Cloud Service Server (CSS) and the Cloud Storage Repository (CSR).

6 KBMA Operations

The working procedure of KBMA is divided into two main processes. They are:

a) Registration Process
b) Authentication Process

6.1 Registration Process

Registration is the one time process carried out by the cloud server. For setting up an account, the user ID, password and other unique information like credit card details used for the payment on pay-per-use basis are given as inputs during registration process. Those input attributes are transferred over a standard protocol like Secure SHell (SSH) for secure transaction. The user, the hashed password and the device information will be stored in the master lookup table. The processes are shown in Figure 2.

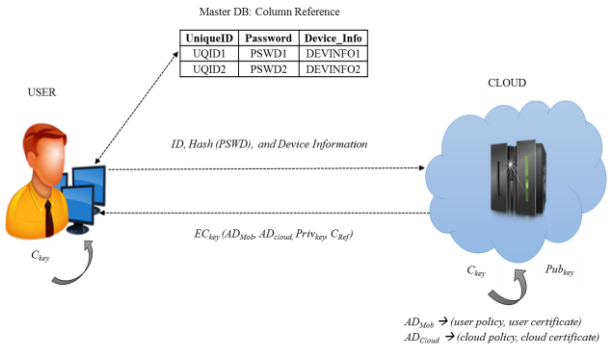

Figure 2. Registration Process

A key C_{key} is generated using the user ID and hashing the password at both ends which is used throughout the authentication process.

$$C_{key} = Id \veebar H\ (Password) \quad \ldots (1)$$

During the registration process, a cryptographic hash function is applied at user (AD_{mob}) and cloud (AD_{cloud}) side for the purpose of authentication.

AD_{mob} contains the user certificate and user policy such as cloud service access policy, user access policy.

$$AD_{mob} = H\ (U_{pol}||U_{cert}) \quad \ldots (2)$$

AD_{cloud} contains the cloud certificate and the cloud policy such as cloud resource policy, user policy.

$$AD_{cloud} = H\ (C_{pol}||C_{cert}) \quad \ldots (3)$$

The CSP will send a message in an encrypted form to the user's device using the key C_{key} generated at both sides (1) which contains the digest of user, cloud, private key and the Column Reference.

$$EC_{key}\ (AD_{mob}||AD_{cloud}||PRI_{key}||C_{ref}) \quad \ldots (4)$$

The pseudocode for the process of user registration is given below.

Table 1. Algorithm for User Registration

User Registration
Registration_User()
Declarations
U_{pol} → *User Access Policy*
U_{cert} → *User Access grant Certificate*
C_{pol} → *Cloud Policy*
C_{cert} → *Cloud Certificate*
H() → *Hash function*
AD_{mob} → *Digested data for device authentication*
AD_{cloud} → *Digested data for cloud authentication*
C_{key} → *Cryptography Key*
C_{ref} → *Column reference for user detail in lookup table*
INP_{user} → *User input for registration*
ENC_user_reg_conf → *Encrypted message of user confirmation*
Input ← *User ID, Password and other unique information*
1. *Start*
2. $USER_{req\_reg}$ = *User request for registration*
3. C_{key} ← *Generate a cryptographic key from* INP_{user} *//Generate a digested data by applying XOR on user policy and user certificate*
4. AD_{mob} ← ***H***$(U_{pol}||U_{cert})$ *//Generate a digested data by applying XOR on cloud policy and cloud certificate*
5. AD_{cloud} ← ***H***$(C_{pol}||C_{cert})$
6. *Generated Asymmetric key in cloud side* *//*PUB_{key} ← *Public key ,* PRI_{key} ← *Private key*
7. *Encrypt the value of* AD_{mob} *,* AD_{cloud} *,*PRI_{key} *and* C_{ref} *ENC_user_reg_conf* ←***EC***$_{key}(AD_{mob}\ ||AD_{cloud}\ ||\ Pri_{key}||\ C_{ref})$
8. *Send this encrypted message (ENC_user_reg_conf) to user as the registration confirmation*
9. *User is successfully registered with Mobicloud*
End

6.2 Authentication Process

There is a need to authenticate each other i.e. Mobile Device and the Cloud Service Provider (CSP) to ensure each other identity. The authentication in this phase is subdivided into two activities. They are as follows.

a) CSP authenticating Device
b) Device authenticating the CSP

6.2.1 CSP authenticating Device

In this phase, two different activities take place, namely encryption and decryption. The following section reveals how the authentication process is carried out in the MobiCloud Environment.

6.2.1.1 Encryption

Figure 3. Shows the encryption process carried out during authentication in the MobiCloud environment.

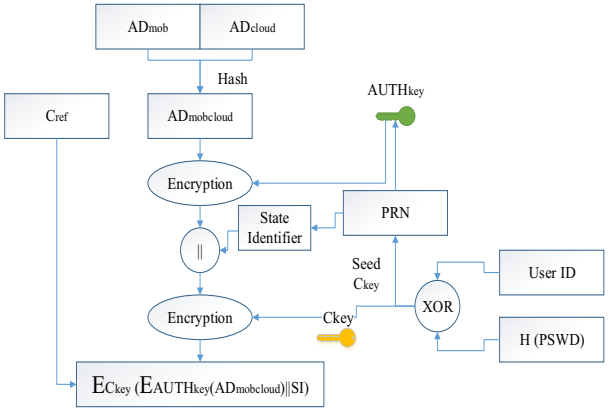

Figure 3. Workflow Diagram

A key C_{key} will be generated when the user requests for a service. Using the key C_{key}, a random number is generated for generating an authentication key $AUTH_{key}$ to avoid the duplication of credential generation. The $AUTH_{key}$ is used for encrypting the AD_{mob} and AD_{cloud} as the Digestive Access Mechanism which is based on the cryptographic encryption techniques.

$$EAD_{mobcloud} = EAUTH_{key}\ (AD_{mob}||AD_{cloud}) \quad \ldots (5)$$

Then, the C_{key} which was already generated at (1), encrypts the already encrypted authentication digest $EAD_{mobcloud}$.

Finally, the encrypted data is sent to the cloud Server along with its Column Reference (C_{ref}).

$$EC_{key}\ (EAD_{mobcloud}||SI) \quad \ldots (6)$$

6.2.1.2 Decryption

The Cloud Server searches for the user's ID and Password in the server database. Once the credentials are matched, a key C_{key} will be generated at the server side. The generated key C_{key} is used for decrypting the encrypted data which was already sent to the server to get the State Identifier and the Authentication Digest.

$$C_{key}\ Decrypt\ (EAD_{mobcloud}||SI) \quad \ldots (7)$$

Again, the same key C_{key} is used for decrypting the Authentication Digest to obtain the plain data.

$$C_{key}\ Decrypt\ (EAD_{mobcloud}) \quad \ldots (8)$$

This process is done for matching the credential data into the server. The search will be based on the Column Reference C_{ref}.

$$AD_{mobcloud} \quad \ldots (9)$$

Once the decrypted credentials are matched, the CSP authenticates the device of the user.

Table 2. Algorithm for User Registration

Authentication Process
CSP_Authenticating_Device()
Declarations
INP_{user} → *User input for cloud access*
C_{key} → *Cryptography Key*
PRN → *Pseudo Random Number*
$AUTH_{key}$ → *Authentication Key*
AD_{mob} → *Digested data for device authentication*
AD_{cloud} → *Digested data for cloud authentication*
$AD_{mobcloud}$ → *Digested data of device and cloud*
SI → *State Identifier*
ENC_AUTH_DIG → *Encrypted digest of device and cloud*
ENC_CAD → *Encrypted data sent to cloud*
Input ← *User ID, Hashed Password*
1. *Start*
// *Encryption Process*
2. $USER_{login}$ = *User logs in for accessing cloud resources*
3. C_{key} ← *Generate a cryptographic key from* $USER_{login}$
4. $AUTH_{key}$ ← *PRN* ← C_{key}
//Generate an Authentication key from Random number generated using cryptographic key
5. $AD_{mobcloud}$ ← $\boldsymbol{H}(AD_{mob}, AD_{cloud})$
6. *ENC_AUTH_DIG* ← $\boldsymbol{E}AUTH_{key}(AD_{mobcloud})$
//Digested data of both device and cloud are encrypted using Authentication key
7. $AUTH_{key}$ ← n^{th} *Sequence of PRN*
8. *ENC_CAD* ← $\boldsymbol{E}C_{key}$*(ENC_AUTH_DIG||SI)*
//Encrypt the data with the State Identifier and send it to the cloud server
// *Decryption Process*
9. *CSP* ← *ID, PSWD*
// Cloud Service Provider searches for the user ID, Password using Column Reference
10. C_{key} ← *Generates a cryptographic key*
11. C_{key} ← $\boldsymbol{D}$*(ENC_CAD)*
12. C_{key} → *PRN , SI →* n^{th} *bit*
// C_{key} acts as the input for PRN and SI is used to identify the n^{th} bit
13. $\boldsymbol{D}AUTH_{key}$ *(ENC_AUTH_DIG)*
//$AUTH_{key}$ decrypts the encrypted digested data of device and cloud
14. *If(Cloud_*$AD_{mobcloud}$*.equals(*$AD_{mobcloud}$*)*
Legitimate User
else
Adversarial User
End

6.2.2 Device Authenticating CSP

This is the second process in Authentication process where the mobile device authenticates the CSP. A Digital Signature DS is generated at CSP side in which it consists of Authentication Digest of device and the cloud $AD_{mobcloud}$ encrypted using a Public Key PUB_{key}.

$$DS = PUB_{key}\ Encrypt\ (AD_{mobcloud}) \quad \ldots (10)$$

The Mobile Device decrypts the encrypted DS with the key Ckey which was already generated and stored at the mobile device.

$$MOB \xleftarrow{Decrypts} PUB_{key}\ E(AD_{mobcloud}) \quad \ldots (11)$$

If the decrypted Authentication digest ADmobcloud matches with the ADmobcloud stored at the CSP, then it is found to be legitimate user.

7 Results and Discussions

The operations of the proposed authentication mechanism is simulated using the Scyther Protocol Analyzer. This tool analyses and computes the vulnerability of each parameter. Table 3 shows that the proposed KBMA mechanism is secure and resilient against various attacks.

The security of the proposed mechanism is determined using the Vulnerability score Vs as given in (12). The Vs is the measure of number of attacks. It ranges from 0.0 to 1.0.

$$Security_{KBMA} : 0.0 <= Vs <= 1.0 \quad \ldots (12)$$

The lowest Vulnerability Score Vs indicates that the proposed system is more secure.

Table 3. Vulnerability Analysis of KBMA

Role	**Claim**	**Status**	**Comment**
Device	Secret $AUTH_{key}$	OK	No Attacks
	Secret SI	Ok	No Attacks
	Secret PSWD	OK	No Attacks
	Secret ID	Ok	No Attacks
	Secret AD_{mob}	OK	No Attacks
	Secret AD_{cloud}	Ok	No Attacks
	Alive	OK	No Attacks
	$Weak_{agree}$	Ok	No Attacks
	Ni_{agree}	OK	No Attacks
	Ni_{synch}	OK	No Attacks
CSP	Secret $AUTH_{key}$	OK	No Attacks
	Secret PSWD	OK	No Attacks
	Secret ID	OK	No Attacks
	Secret SI	OK	No Attacks
	Alive	OK	No Attacks
	$Weak_{agree}$	OK	No Attacks
	Ni_{agree}	OK	No Attacks
	Ni_{synch}	OK	No Attacks

The claims of the security analysis validated by the Scyther protocol analyser are given below.

7.1 Claim 1: Secret $Auth_{key}$

This Authentication key is used for encrypting and decrypting the digested data ($AD_{mobcloud}$). The device encrypts the $AD_{mobcloud}$ and sends it to the CSP using the $AUTH_{key}$. This Authentication key is generated using the State Identifier (SI). $AUTH_{key}$ remains secret as it is not transmitted through the communication channel.

7.2 Claim 2: Secret SI

State Identifier (SI) is used for specifying the n^{th} sequence of the generated Random Number (PRN). Only after encrypting the SI using C_{key}, it is sent to the CSP. It is neither shared nor sent to anyone.

7.3 Claim 3: Secret PSWD

Here, the security of the password depends on the user. The length of the password can be 512 bit string. It is hashed and a replica is stored at the cloud authentication database during the registration process. The Password is not transferred during communication instead it is used at both ends.

7.4 Claim 4: Secret ID

The ID of user is sent along with the password to the CSP only once during the registration process. The communicating device does not send the ID to the CSP during Authentication process. Hence, it remains safe. But, the security of ID really depends on the user.

7.5 Claim 5: Secret AD_{mob}

AD_{mob} is the hashed data related to the user which holds the unique information, access policy, etc. Before transmission, every information is encrypted and hashed using the key C_{key}.

7.6 Claim 6: Secret AD_{cloud}

AD_{cloud} is the hashed data related to the cloud which holds the unique information and cloud policy information. Before transmission, AD_{cloud} is hashed and encrypted with AD_{mob} using the Authentication Key ($AUTH_{key}$).

7.7 Claim 7: Secret $Weak_{agree}$

The proposed mechanism proves that the device is in weak agreement with the CSP. The device is running the proposed mechanism with the CSP and likewise the CSP is running with the device. Hence, it is evident that the communication is not attacked by adversarial users during the authentication process.

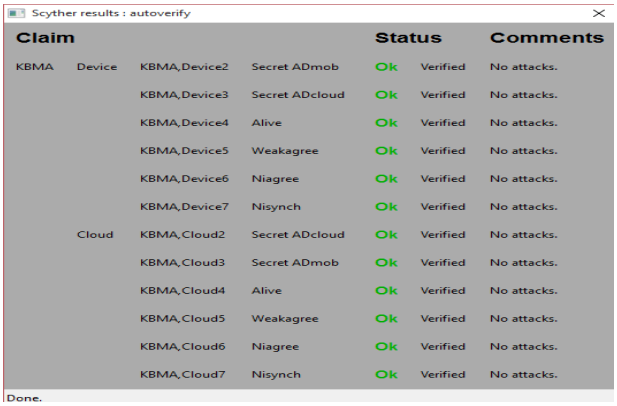

Figure 4. Auto Claims Result Window of KBMA Mechanism

Figure 4 shows the auto claims result of the proposed authentication mechanism which has no attacks.

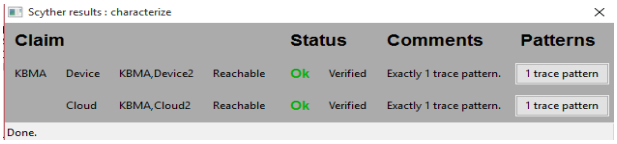

Figure 5. Trace Pattern of KBMA Mechanism

The trace pattern of the entities (Device and Cloud) involved in accessing the cloud resource is given in Figure 5. The result shows that there are no attacks during authentication process.

8 Conclusion

In recent times, the MCC is becoming a new hot technology and security of the same has become a research issue. Also, several existing authentication techniques and methods for mobile cloud computing are discussed. In this paper, a mechanism called KBMA is proposed which verifies the identity of the entities participating in accessing the cloud services. The proposed KBMA mechanism involves two processes namely registration and authentication. The algorithm of the KBMA is implemented and analysed using Scyther tool. The derived results show that the proposed KBMA is resilient against various attacks and the claims of the security is also analysed. From the results, it is evident that the proposed mechanism authenticates the entities and provides secure channel for communication. User Authentication will be incorporated in future research.

References

1. Mobile's next great leap will happen in the cloud, (2014), http://www.infoworld.com/d/cloud-computing/mobiles-next-great-leap-will-happen-in-the-cloud-236891
2. E. Ghazizadeh, M. Zamani, J. Manan, A. Pashang, *Proceedings of 2012 IEEE 4th International Conference on Cloud Computing Technology and Science*, 62-68 (2012)
3. Z. Sanaei, S. Abolfazli, A. Gani, R. Buyya, IEEE Comm. Sur. & Tutorials, 1-24 (2013)
4. IehabALRassan and HananAlShaher, IJNSA **5**, 41-53 (2013)
5. G. Deepak, B. S. Pradeep, S. Srinath, IJSR **3**, 598-602 (2014)
6. A. C. Donald, L. Arockiam, *Proc. of International Conference on Computer Communication and Informatics (ICCCI 2015)*, IEEE, 1-6, (January 2015)