INCREASE SECURITY IN CLOUD COMPUTING USING HMAC AND KERBEROS ALGORITHMS

M Varaprasad Rao[1], G Vishnu Murthy[2]

[1]*Dept.of CSE, Anurag Group of Instiitutions, Hyderabad, India*
[2]*Dept.of CSE, Anurag Group of Instiitutions, Hyderabad, India*

Abstract

The Cloud Computing is a service, based on Internetwork technology. The customers can do data transfer and resource sharing among number of services provided in cloud computing. Any service is easily accessed by a customer from anywhere and anytime in cloud thru Internetworking, therefore it has two good properties called availability and QoS. It has its pros and cons for the customers to create and store the information in cloud server. The application and data management software tools are not that much trustworthiness, then it implies on security aspects of QoS and availability in cloud. Thus we proposed a method for improving security in cloud data storage using Kerberos algorithm and HMAC Public Key Infrastructure.

Keywords: *Cloud Computing, Third Party, Cloud Security, Kerberos algorithm, HMAC, Cloud Provider*

----------***----------

1. INTRODUCTION

Cloud computing rapidly growing to achieve the prosperity for the human, and with this increase in penetrate malicious program in the cloud security becomes more important. The Cloud Computing is a powerful technology and the computing architecture consists of Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The SaaS services are used to transform data center into small areas of computation on large size [1], e.g., AWS EC2, S3 [2] [3]. In this paper, Kerberos algorithm is used to authenticate users for granting ticket to access a service and distribute the session keys in encrypted format over IP network. Every user must create profile and UserID in the third party to connect Cloud. The user's passwords are hashed and saved in the Database for more secure. To retrieve the UsedID and password in a secured mode, we use Kerberos and HMAC algorithms. It has the following procedure

- User request to AS for Ticket Granting Ticket (TGT)
- AS will verify the user's request, create TGT and Session key; encrypt using user's password.
- User receives reply from AS and the same will communicate with Cloud service provider to get Ticket Granting Service (TGS).
- A pair of ticket (Ticket, Session key) is generated and sends to user through HMAC algorithm.
- The user sends ticket to Cloud Service Provider/Server.
- The server will authenticate and grants access to user.

Here to access large scale database secured in the cloud every user should have profile and password. The remaining sections of the paper are organized as cloud services and models, problem statement, implementation process.

- In cryptography, a pair of HMAC algorithm and secret shared key will be used in cryptographic iterative hash function, e.g., MD5, SHA-1. These algorithms may be used depends on application. The strength of HMAC is based on properties of hash function and simultaneously verifies both authentication and integrity of message. An iterative hash algorithmic function divides the long information into number of blocks of fixed size. Compression algorithm is applied on these blocks. The HMAC algorithms such as SHA-1 and MD5 are operated on 512-bit block size of the information. The hash code size of SHA-1 is 160 bits and MD5 is 128 bits, the size of this hashing may be truncated if desired.

HMAC (K, m) = H (K XOR outpad, H (K XOR inpad, text))

Here
K – Secret key
m – Message or information
H – Hash function
outpad – Outer padding
inpad – Inner padding

Python has a library called a hmac module and it is defined as follows
import hmac
def hmac_md5(key, msg):
return hmac.HMAC (key, msg, md5)

The HMAC depends on the size of secret key, hence there is a chance of brute force attack on HMAC. The attack can be separated from regular collision attacks with shared secret key and therefor it is not possible to find collisions with the sufficient combinations. The forgery and pre-image attacks are not actively involved in proving security of HMAC [11], but they provide insights to HMAC based on existing hash code. Timing attack can be performed by digit by digit to find a HMAC code in less secured systems.

Cloud Services

Cloud computing [6] [7] [8] has the following services

- **Infrastructure as a Service** (Iaas) provides all hardware parts at reasonable pricing model.
- **Platform as a Service** (PaaS) it allows developers to build applications and services.
- **Software as a Service** (SaaS) specifies to access any software applications.

Cloud Models

- **Public Cloud** is pooled shared physical resources, and accessible over a public network.
- **Private Cloud** is a secured cloud based environment, which can be operated by an authorized user.
- **Hybrid Cloud** is an integrated cloud service uses both private and public clouds.

2. PROBLEM STATEMENT

2.1 System Architecture

Cloud data storage model with Kerberos is explained in Figure1. It consists of eight parts and explained in the following Table1.

Table1: Parts of Kerberos Algorithm applied in Cloud Data Storage

User	User request to create an account with third party data base and replies a pair of password and session key to place data in Cloud Database.
Cloud Service Provider	Microsoft Azure, Google Apps and Amazon Web Services etc., are offered cloud services and solutions. The resources are accessed on sharing mechanism thru Internetworking anywhere and anytime.
Kerberos Authentication Service (KAS)	KAS transmits the data in encrypted form for the given plain text passwords over a secured network between clients and servers. Kerberos protocol works on non-trusted networks for authenticates users.
Authentication Service (AS)	AS knows all the users' passwords, where passwords are stored in centralized database. In addition to this, the AS shares a unique secret key with each server.
Ticket Granting Service (TGS)	TGS will issue tickets to authenticated users.
Data Base	This consists of information about users and their services. It is shared between third party and Kerberos. The same is shared to principal.
Third Party	The third party defines - who is the correctness, expertise, capabilities to access and utilize the specified cloud.
HMAC	Will encrypt the ticket and sends it to user to access the cloud service by using SHA-1.

- The database record consists the following.
  - Principal entry
  - Validity of a ticket
  - Encrypted key.
  - Renewal time of Ticket
  - Flags
  - Password validity
  - Validity of the principal

2.2 Design Objectives

The objectives of this model are work and trustworthiness. These will ensure the security in cloud to access data from large size databases.

Work – with the minimum time the user can register and do operations on storage

Trustworthiness – only the authorized user can access data in secured manner.

3. IMPLEMENTATION PROCESS

The user's data stored in cloud is accessed through only referring cloud service provider. Therefore the database may be frequently updated by insert, delete, append etc. operations. To ensure that these operations in secured manner, we have introduced a model using Kerberos and HMAC algorithms. This has the following steps.

A. User sends request to third party for registration. The third-party will make a request to access TGT by userID and profile. Then the user is authenticated by authenticated service. It has 4 events as realm, option, time and nonce. Realm – is client or user or consumer or customer. Option – is a flag(s) used to set the behavioral aspects of the user's service. Time – is used to set the time of a ticket and its validity and Nonce – is a random value and prevents re-play attack.

B. To know user's data, refer to a TGT. Once user's data is known then simply encrypt a block of message using a key based user's password. The generated block of message will have a session key and is used to communicate between the user and TGS with its own nonce, flag, status, timing and validity.

C. The client sends a request to the TGS to obtain service-granting ticket from the desired cloud service based on existing TGT.

D. The TGS will decrypt and verifies with hash code to the incoming ticket. It checks the lifetime of ticket; if lifetime is alive, then to authenticate the user it compares userID and network address. If any user is permitted to access request service, then TGS will issue a ticket to access the specified cloud. The service-granting provider ticket contains a timestamp, this timestamp is used once again; if a user needs to access same cloud, then user can make use previously obtained

ticket. And the user need not worry about password. Here, the ticket is encrypted with a secret key (K_v) and this is known only to the TGS and server, this leads to prevent alteration of the message.

E. Finally, to access respective cloud service it is required that mutual authentication and the process is as follows.

- Subkey: A session is secured or protected using encrypted key. If the subkey is not considered then use the session key of ticket ($k_{c,v}$).
- Sequence number: It is a random value nonce generated for a session. It is an optional field and will be used to send and/or receives the messages between client and cloud server. The sequence number will help in detecting replay attacks. The Table2 shows the implementation process [5].

Table 2: Information Service Exchange in Cloud using Kerberos Service

A. Authentication Service: Ticket-Granting Ticket
1) C→ AS: OptionsID_cRealm$_c$$ID_{tgs}$Times Nonce$_1$
2) ASC→:Realm$_c$$ID_c$Ticket$_{tgs}Ek_c$[$k_{c,tgs}$TimesNonce$_1$Realm$_{tgs}$$ID_{tgs}$ Ticket$_{tgs}$=E$_{ktgs}$[Flages$k_{c,tgs}$Realm$_c$$ID_cAD_c$Times]
B. Ticket-Granting Cloud Service: CloudService-Granting Ticket
3) CTGS→OptionsID_vTimesNonce$_2$Ticket$_{tgs}$ Authenticator$_c$
4) TGSC→:Realm$_c$$ID_c$Ticket$_vEk_{c,tgs}$ [$k_{c,v}$Times Nonce$_2$ Realm$_2$$ID_v$] Ticket$_{tgs}$=E$_{ktgs}$[Flags $k_{c,tgs}$Realm$_c$$ID_cAD_c$Times] Ticket$_v$=E$_{kv}$[Flags $k_{c,v}$Realm$_c$$ID_cAD_c$Times] Authenticator$_c$=HMAC(E$_{Kc,tgs}$[$ID_c$Realm$_c$ TS_1])
C. Client/Server Authentication: CloudService
5) CTGS→: Options Ticket$_v$Authenticator$_c$
6) TGSC→:E$_{kc,v}$[TS_2SubkeySeq#] Ticket$_v$ =E$_{kv}$ [Flags $k_{c,v}$Realm$_c$IDcAD$_c$Times] Authenticator$_c$=HMAC(E$_{kc,v}$ [ID_cRealm$_c$$TS_2$SubkeySeq#])

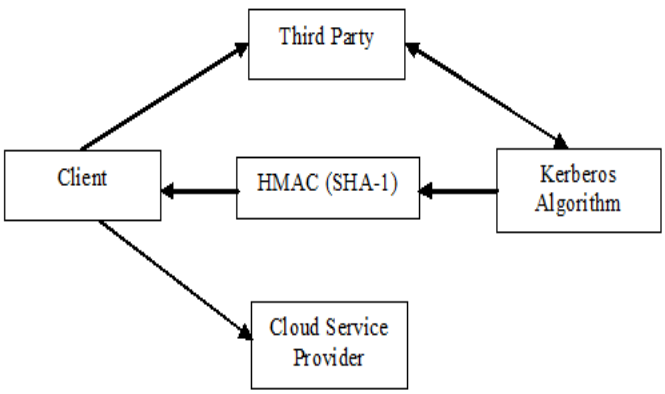

Fig1: Architecture of Cloud Data Storage

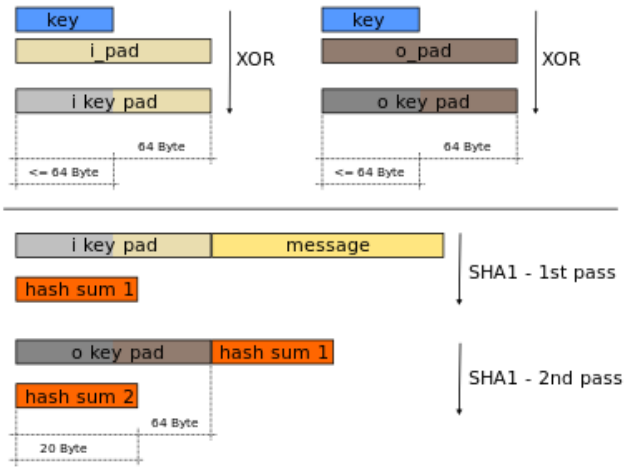

Fig 2: HMAC format

4. CONCLUSION

We have introduced a new effective and flexible scheme, which includes Kerberos authentication algorithm, HMAC and third party. Kerberos authentication algorithm is used to authenticate the users in the network and will helps the user to communicate two way communication between user and cloud server. The third party checks the users, weather the user is correct and expertise, then the user has capabilities to access cloud service provider over secured networks. HMAC will encrypt the ticket and sends it to user to access the cloud service by using SHA-1.

REFERENCES

[1] Cong Wang,QianWang,KuiRen, "Ensuring Data Storage Security in Cloud computing", cit by: 23 IEEE International Conference on Computer and Information Technology; 2009, pp.1-9

[2] N. Gohrin "Amazon's S3 down for several hours," Online at http://www.pcworld.com/business-center/article/ 142549 /amazons s3 down for several hours.html, 2008.

[3] Amazon.com, "Amazon Web Services (AWS)," Online at http://aws.amazon.com, 2008.

[4] MILL88Miller. S; Neuman,B.; Schiller,j.; and Saltzer, j. "Kerberos authentication and authorization System" Section E.2.1, Project Athena Technical plan, M.I.T. Project Athena, Cambridge, MA. 27 October 1998.

[5] William Stallings, "Cryptography and Network Security", second edition,2002.

[6] John Harauz, Lori M. Kaufman, Bruce Potter, " Data Security in the World of Cloud Computing", IEEE Security and Privacy, July/Aug. 2009, vol. 7, issue. 4, pp. 61-64.

[7] Loganayagi B, Sujatha.S., " Cloud Computing in Stax Platform", IEEE International Conference on Computer Communication and Electrical Technology, (IEEE-ICCCET 2011); 18-19 Mar. 2011, pp.1-5.

[8] Dawei Sun, Guiran Chang, QiangGuo, Chuan Wang, Xingwei Wang., "A Dependability Model to Enhance

Security of Cloud Environment Using System-Level Virtualization Techniques", First International Conference on Pervasive Computing, Signal Processing and Applications (PCSPA); 2010, pp. 305-310.

[9] http://publib.boulder.ibm.com

[10] http://en.wikipedia.org/wiki/Hash-based_message_authentication_code

[11] Jongsung Kim, Alex Biryukov, Bart Preneel, and Seokhie Hong, "On the Security of HMAC and NMAC Based on HAVAL, MD4, MD5, SHA-0 and SHA-1", 5th International Conference on Security and Cryptography for Networks pp 242-256