International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 03 | Mar-2016 www.irjet.net p-ISSN: 2395-0072

# Threshold Cryptography Based Data Security in Cloud Computing

[1]Swati S. Devare, [2] Manisha S. Shinde, [3]Nisha B. Mhaske, [4]Pooja B. Ghogare
*Student, Dept. of Computer Engineering, PREC College, Maharashtra, India*
*Student, Dept. of Computer Engineering, PREC College, Maharashtra, India*

---

Abstract - Now a day's cloud computing is very popular in large and small scale organization as it will stored large amount of data and provide low cost service. Hence it was create daily new challenges to provide secure authorization, integrity and access control. Some approaches are ensuring about security but there are also some lack in these approaches and issues due to collusion attack, heavy computation. To resolve this issue we proposed a scheme its threshold cryptography in which data owner can divide users in group and provide single key with the using key each user in group can access the data. In these studies, we use capability list to control the access. In this scheme not only provide data security but also provide reduce number of keys.

**Key Words:** Cloud computing, Outsourced data, malicious outsiders, access control, authentication, capability list, threshold Cryptography.

## 1. INTRODUCTION

Now a day's cloud computing is fast growing and data storage system as it's provide storage system in very low cost .Cloud computing provides services according to three fundamental service models: Infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS). Storage as a service is basically a platform as a service. The five characteristics of cloud computing are: on-demand service, self-service, location independent, rapid elasticity and measured scale service. The feature of cloud computing is unique. Most of the organization and institute utilized of this characteristics of the cloud computing and take benefit to gain profit [1]. Hence, industries are shifting their businesses towards cloud computing. Cloud computing uses increased day by day, however, Data security is main concern in cloud computing. Peoples are worried for stored data at cloud server. Some people believe that cloud is unsafe place and once you send your data to the cloud, you lose complete control over it [8] [9].data owner can be stored data on external server thus sometime it may be unsecured. So, confidentiality, integrity and access of data become more vulnerable. Service provides will be provide an external server that is why data owner can't trust on them as they can use data for their benefits and can spoil businesses of data owner [4]. Data users may be harmful thus data owner cannot trust on them. Data confidentiality may violet through collusion attack of malicious users and service providers. Many schemes are given to ensure these security requirements but they are suffering from collusion attack of malicious users and. To address these issues we propose a scheme. In this scheme, there are basically three entities: Data Owner (DO), Cloud Service Provider (CSP) and Users. Users are divided in location, projects and department each user use key for encryption and decrypted data. Each user in the group shares parts the key. Data will be decrypted when among the users are available on cloud server. Our proposed system focus on provide security as well as reduce the number keys. With the capability list can achieve fine-grained data access control [6]. It is basically row-based decomposition of access matrix. Authorized data which is provide by owner and operation are mentioned in capability list. It is better suit than Access Control List [5] [10] [16] because ACL specifies users and their permitted operation for each data and file. It's not suitable when two users access same type of data. In this paper, these system has been used as the Diffie-Hellman algorithm to create one time access session-key between CSP and user to keep secure the data from outsiders.

## I. LITERATURE SURVEY

In cloud computing mostly two important security requirements are keep data secure and efficient access control for data. Some time we focus on data security we not concentrate on system performance access control and have to focus on DO, CSP, users. Like sometime we use among the keys for secure data. All keys are confidential so need to be keep secure and maintain these keys which are additional work. These additional works affect the performance of the system. Hence it is needed to reduce these keys so our system help for reducing keys with provide data security as well as increased the performance of the system. Many schemes are suggested to meet these requirements. In [13] author define the group-key scheme. In this scheme, there is single key for the one user group for decryption process and this single key used by each user in group. These scheme help for the reduce key but simultaneously it will be cause for collusion attack as many cloud service providers can attack and there is a user a single malicious user can access the hole data of group and leak whole data of the group to Cloud Service Providers. As we know that Cloud Service Provider is not trusted party. Cloud service providers can access data owner's data for the commercial benefit. In the [14] Proposed scheme can define data security and data access. In this scheme data encrypted by symmetric key and this symmetric key are only known to the data owner and corresponding data users. Data owner can encrypted data same will be uploaded on cloud server. CSP can't see data stored at it as data are encrypted. Data are further encrypted by one time session key which is secret

shared with the cloud service provider and user by the modified Diffie-Hellman protocol to protect data from outsiders during the transmission between CSP and user. This proposed system provide security undoubtly but the description key has been available with each user so among of the users included in some applications, hence number of keys may increase. Hence, increases the maintenance and security Concerns of keys Communication model of the proposed scheme somehow matches with it [4] but proposed scheme is more secure and reduces number of keys. Our system useful for those application where data will be used in large amount where work done in team ex: software industries. You may think proposed scheme has limited applications but it is not as such. Threshold cryptography can use when group of users access data. Such as software and hardware industries, medicals, institutes, banks and large fields. There is provide hierarchy of access which is reliable and useful. For Example, university has HOD, teachers, clerk, counselor and students. Each one has different level of access right.

## II. PROPOSED SYSTEM

We suppose that our model is composed of three entities: a CSP, a DO and many users associated with DO. Initially, all Users are registered at DO. During registration users send their credentials to DO. We assume that user's credentials are sent securely to DO. DO then divides users in groups and provides encryption keys, tokens, algorithm (MD5) and other necessary things for secure communication to user groups in response of registration. A user can get data from CSP in a confidential manner after successful authentication of himself at CSP. We assume that CSP has a large capacity and computational power. We also assume that no one can breach the security of CSP. Further we assume that the algorithm which is used to generate the secrete keys for encryption, is secure at DO. DO has storage capacity to store some files and data and, he can execute programs also at CSP to manage his files and data. We are using modified Diffie- Hellman and public key cryptography to secure communication between CSP and user. Modified Diffie-Hellman protocol is used to create one time session-key between CSP and user. Fig.1 illustrates the secure communication between entities in the proposed scheme.

We present a complete model for secure communication between different entities and secure access to data. There are four algorithms in the proposed scheme. Algorithm 1 describes secure communication of data between DO and CSP moreover this algorithm insures data confidentiality and, authentication of DO and CSP. Algorithm 2 describes procedures which DO and CSP apply after a new file creation in respect. Algorithm 3 describes about secure communication of data between CSP and user. In this algorithm user's authorization is also checked. Algorithm 4 describes the threshold cryptography technique for decryption of a user's file. Algorithm 4 is applied at user side where number of keys is reduced (one key corresponding to one group) and no threat of collusion attack as in group-key scheme.

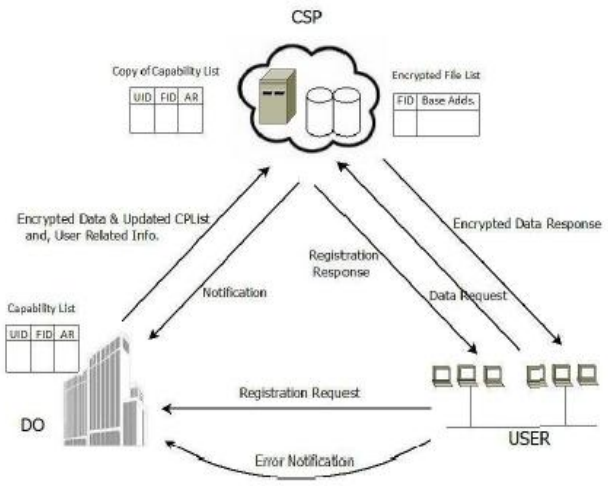


Fig. 1 Communication model in Proposed Scheme

## III. Algorithms

**Algorithm 1**: Procedure to be followed by CSP after getting encrypted File and Capability List from DO.

Step 1: CSP stores Encrypted Data and Capability List

Which are received from DO

Array ← Rece(EkPuCSP(EkPrDO ( EKkt(Fi)) || (CPList))

EKkt(Fi) ← DkPrCSP(DkPuDO( (Array))_

Step 2: CSP updates the Encrypted File List

Encptd. File List ← Encptd. File List (FID, Base Adds.)

Step 3: CSP updates Capability List

CPList ← CPList (UID, FID, AR).

Algorithm 1 describes the process what CSP do after getting encrypted data and Capability List from the DO. CSP decrypts the message using its own private key and the public Key of data owner and stores the encrypted data and capability List in its storage. CSP then updates the encrypted File List and Capability List. Since, data are encrypted using Symmetric key (KT) which is known only to DO and respected user group, CSP can't see data even though user's credential comes through it.

**Algorithm 2**: Procedure to be followed after a new File

Creation

Step 1: DO updates Capability List

CPList ← Add.(CPList, (UID, FID, AR))

Step 2: Now, DO encrypts the CPList, Encrypted File,

Symmetric key and sends these to the CSP

Send (EkPuCSP(EkPrDO(CPList,EKkt (Fi), EkPrDO

(EkPuUSR(KT, N+1, TimeStamp)))))

Step 3: CSP updates its copy of the Capability List,

Encrypted File List and sends symmetric key to indented User group.

Send (EkPuUSR(EkPrDO(EkPuUSR(KT, N+1,TimeStamp))))

Step 4: Now, the user can send actual access request for te file directly to the CSP.

Users of the user group then decrypt the message and get their own parts of the symmetric key (KT). To avoid man-in-middle and replay attack we use nonce and timestamp in each message. After getting the details, user can request to CSP for data.

**Algorithm 3**: Algorithm for secure data exchange

Between CSP and User by using Modified D-H key exchange

Step 1: User sends data access request to CSP

Send (UID, FID, AR))

Step 2: CSP matches UID, FID, AR with CPList stored at it

If ( match)

Go to step (3)

Else Go to step (6)

Step 3: CSP initiates D-H exchange with that User and

shares one time shared session key( KS)

Step 4: CSP encrypts the encrypted File with shared

Session key and sends it to User

Send ( EKk3(EKkt (Fi)))

Step 5: User decrypts the File and calculates the message

digest of that File If Calculated digest matches with stored digest then File is original

else

File is modified and User sends Error Notification to DO

Step 6: CSP sends 'invalid request' message to User

**Algorithm 4**: Algorithm for Decryption of a File for

User 1

Step 1: User 1 receives Encrypted File

M ← Rece .( EKkt(Fi))

Step 2: Initially, all bits of PKS Vector is zero. Here,

PKS Vector indicates parts of the key. User 1 will update

This PKS Vector with the components he has

PKS = PKS OR A1

Step 3: User 1 forwards M and PKS to ith user of the

Group. Who will decrypt it and update the PKS Vector

M = DKkt1 (M)

PKS = PKS OR Ai

The ith user then forwards M and PKS to next user in the

Group. The next user performs same operations as ith user did. This process is continued.

Step 4: if(PKS = = 11111.....up to d bits)

Go to step (5)

else

Go to step (3)

Step 5: Forward M to the User 1

User 1 then decrypts message (M) and get File

Fi = DKkt2(M).

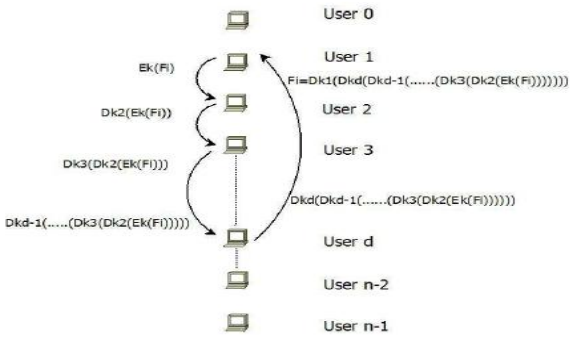


Fig.2.Process of decryption

## IV. Experimental RESULT

We have developed project to prevent the unauthorized access of file stored on the cloud storage. To upload the file on cloud storage user need to login to the system and enter the cloud id for which he is going to registered.

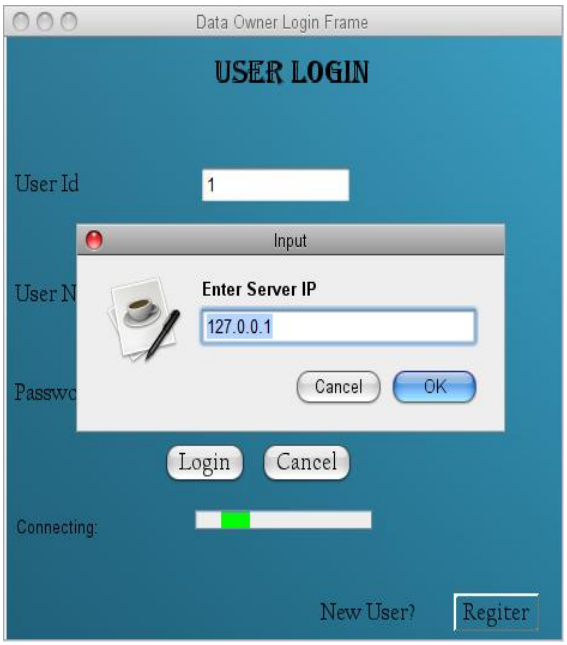


Fig.3.User Login to Cloud Server

The second image shows the options available to the data user when he login to the cloud server. The data owner can view the access rights details and can see the capability list of cloud users.

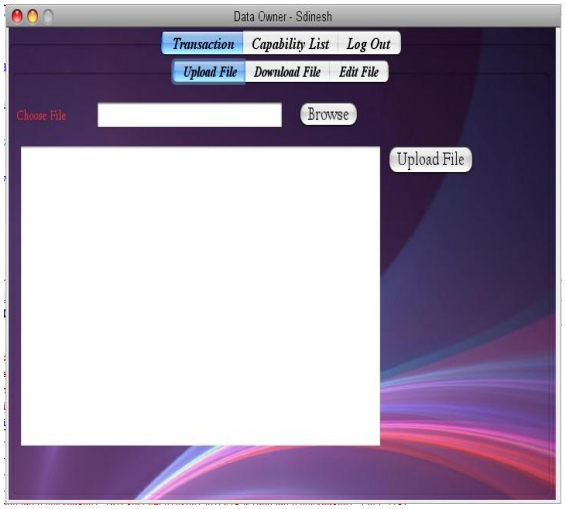


Fig. 2.Data Owner Page

When user request for particular file then cloud data owner can decide is to approve the rights or not. If owner deny the access rights then user can not able to download the file.

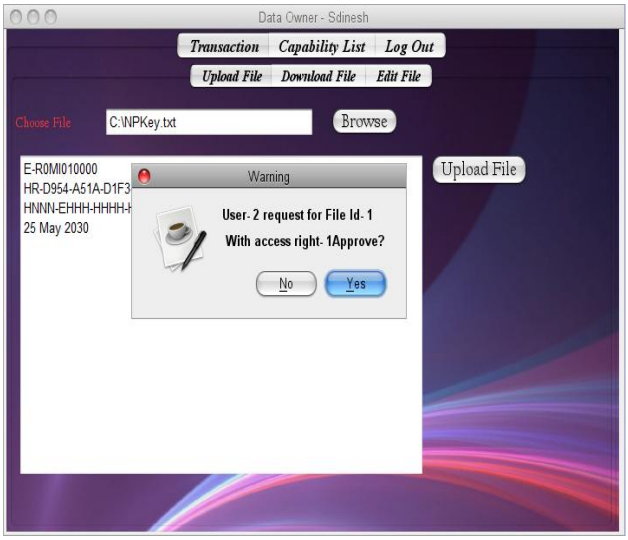


Fig.5 .Access Rights

.

## V. CONCLUSION

In this paper, we presented a new approach which provides Security for data outsourced at CSP. Some approaches are given to secure outsourced data but they are suffering from having large number of keys and collusion attack. By employing the threshold cryptography at the user side, we protect outsourced data from collusion attack. Since, DO stores its data at CSP in encrypted form and, keys are known only to DO and respected users group, data confidentiality is ensured. To ensure fine-grained access control of outsourced.

## References

[1] J. Do, Y. Song, and N. Park, "Attribute Based Proxy Re-encryption for Data Confidentiality in Cloud Computing environments," Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First 206 ACIS/JNU International Conference on, vol., no., pp.248-251, 23-25 May 2011.

[2] A. Shamir, "How to share a secret," Communications of the ACM, v.22 n.11, p.612-613, Nov. 1979. [Online]. Available:http://portal.acm.org/citation.cfm?id=359168.359176.

[3] N. Bennani, E. Damiani, and S. Cimato, "Toward Cloud-Based Key Management for Outsourced Databases," Computer Software and Applications Conference Workshops (COMPSACW), 2010 IEEE 34th Annual, vol., no., pp.232-236, 19-23 July 2010.

[4] [4] S. Sanka, C. Hota, and M. Rajarajan, "Secure data access in cloud computing," Internet Multimedia Services architecture and application (IMSAA), 2010 IEEE 4th International Conference on, vol., no., pp.1-6, 15-17 Dec. 2010.

[5] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," in Proc. of NDSS'05, 2005.

[6] C. Hota, S. Sanka, M. Rajarajan, and S. Nair, "Capability-Based Cryptographic Data Access Control in Cloud Computing," Int. J. Advanced Networking and Applications Volume: 01 Issue: 01 Page: (2011).

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," Association for Computing Machinery, in Proc. of CCS'06, 2006.

[8] T. Mather, S. Kumaraswamy, and S. Latif, "Cloud security and Privacy," O'Reilly Media, Sep. 2009.

[9] A. T. Velte, T. J. Velte, and R. Elsenpeter, "Cloud computing a practical approach," Tata McGraw-Hill Edition, 2010, ISBN-13:978- 0-07-068351-8.

[10] W. Stallings, "Cryptography and network security," LPE Forth Edition, ISBN-978-81-7758-774-6.

[11] G. Miklau, and D. Suciu, "Controlling access to published data using cryptography," in Proc. of 29th VLDB, Germany, Sept 2003.

[12] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," in Proc. of IEEE INFOCOM 2010..