ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

Secure Data Framework for Multi-Owner Architecture Using Threshold Cryptography

Prasad Bangar[1] Kundan Randive[2], Jinesh Jain[3] Ganesh Bhise[4] Jayanthi E[5]

B.E., Dept. of Computer, Sinhgad College of Engineering, Pune., India[1234]

Assistant Professor, Dept. of Computer, Sinhgad College of Engineering, Pune., India[5]

ABSTRACT: Cloud computing is very popular in organizations and institutions because it provides storage and computing services at very low cost. However, it also introduces new challenges for ensuring the confidentiality, integrity and access control of the data. Some approaches are given to ensure these security requirements but they are lacked in some ways such as violation of data confidentiality due to collusion attack and heavy computation (due to large no keys). To address these issues we propose a scheme that uses threshold cryptography in which data owner divides users in groups and gives single key to each user group for decryption of data and, each user in the group shares parts of the key. In this paper, we use capability list to control the access. This scheme not only provides the strong data confidentiality but also reduces the number of keys.

KEYWORDS: Energy Outsourced data, malicious outsiders, access control, authentication, capability list, threshold cryptography.

I. Introduction

Cloud computing is a new and fast growing technology in field of computation and storage of data. It provides storage and computing as a service at very attractive cost. It 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. These characteristics make cloud significant. Industries and institutions are exploiting these characteristics of cloud computing and increasing their profit and revenue [1]. That is why, industries are shifting their businesses towards cloud computing. However, data security is a major obstacle in the way of cloud computing. People are still fearing to exploit the cloud computing. Some people believe that cloud is unsafe place and once you send your data to the cloud, you lose complete control over it.They are more or less right.

Data of data owners are processed and stored at external servers. So, confidentiality, integrity and access of data become more vulnerable. Since, external servers are operated by commercial service providers, data owner can't trust on them as they can use data for their benefits and can spoil businesses of data owner [4]. Data owner even can't trust on users as they may be malicious. 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 cloud service provider and heavy computation (due to large no keys). 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 groups on some basis such as location, project and department and, corresponding to each group, there is a single key for encryption and decryption of data. Each user in the group shares parts of the key. Data can be decrypted when at least threshold number of users will present.

This scheme not only provides data confidentiality by all means but also reduces the number of keys. To achieve fine-grained data access control, the approach has used capability list [6]. It is basically row-based decomposition of access matrix. In capability list authorized data and operations for a user are specified. It is better suit than Access Control List (ACL) [4][5][6] because ACL specifies users and their permitted operation for each data and file. It is practically inefficient that two users require same data and have same operations on it.

DOI: 10.15680/IJIRCCE.2016. 0401047

II. OBJECTIVES

1) The Objective is to maintain the capability list.
2) In this system we use only the single key for encryption and decryption.
3) Data can be decrypted when at least threshold number of users will present. This scheme not only provides data confidentiality by all means but also reduces the number of keys.
4) In capability list authorized data and operations for a user are specified. It is better suit than Access Control List (ACL) because ACL specifies users and their permitted operation for each data and file.

III. CLOUD COMPUTING

Cloud computing is a new and fast growing technology in field of computation and storage of data. It provides storage and computing as a service at very attractive cost. It 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. These characteristics make cloud significant. Industries and institutions are exploiting these characteristics of cloud computing and increasing their profit and revenue [1]. That is why, industries are shifting their businesses towards cloud computing.

However, data security is a major obstacle in the way of cloud computing. People are still fearing to exploit the cloud computing. Some people believe that cloud is unsafe place and once you send your data to the cloud, you lose complete control over it[4][5][6]. They are more or less right. Data of data owners are processed and stored at external servers. So, confidentiality, integrity and access of data become more vulnerable. Since, external servers are operated by commercial service providers, data owner can't trust on them as they can use data for their benefits and can spoil businesses of data owner. Data owner even can't trust on users as they may be malicious. 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 cloud service provider and heavy computation (due to large no keys).

IV. PROBLEM DEFINITION

To Develop a system for data security in multi owner cloud environment using threshold cryptography.

V. RELATED WORK

In existing system, there is a single key corresponding to each group of users for decryption process and all users of the group know that key. Here, number of keys is reduced but there is a problem of collusion attack of CSP and a user because a single malicious user can leak whole data of the group to CSP. We know that CSP is not trusted party. It can use data owner’s data for its commercial benefits. Consider following two aspects:

1) Capability list:
To achieve fine-grained data access control, the approach has used capability list [6]. It is basically row-based decomposition of access matrix. In capability list authorized data and operations for a user are specified. It is better suit than Access Control List (ACL) [4][5][6] because ACL specifies users and their permitted operation for each data and file. It is practically inefficient that two users require same data and have same operations on it.

2) Single key for encryption and decryption:
There is a single key for encryption and decryption of data. Each user in the group shares parts of the key. Data can be decrypted when at least threshold number of users will present. This scheme not only provides data confidentiality by all means but also reduces the number of keys.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

Disadvantages of Existing System:

1) There is a single key corresponding to each group of users for decryption process and all users of the group know that key. Here, number of keys is reduced but there is a problem of collusion attack of CSP and a user because a single malicious user can leak whole data of the group to CSP.

2) ACL specifies users and their permitted operation for each data and file. It is practically inefficient that two users require same data and have same operations on it

VI. PROPSED SYSTEM MECHANISM

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.

VII. SYSTEM ARCHITECTURE

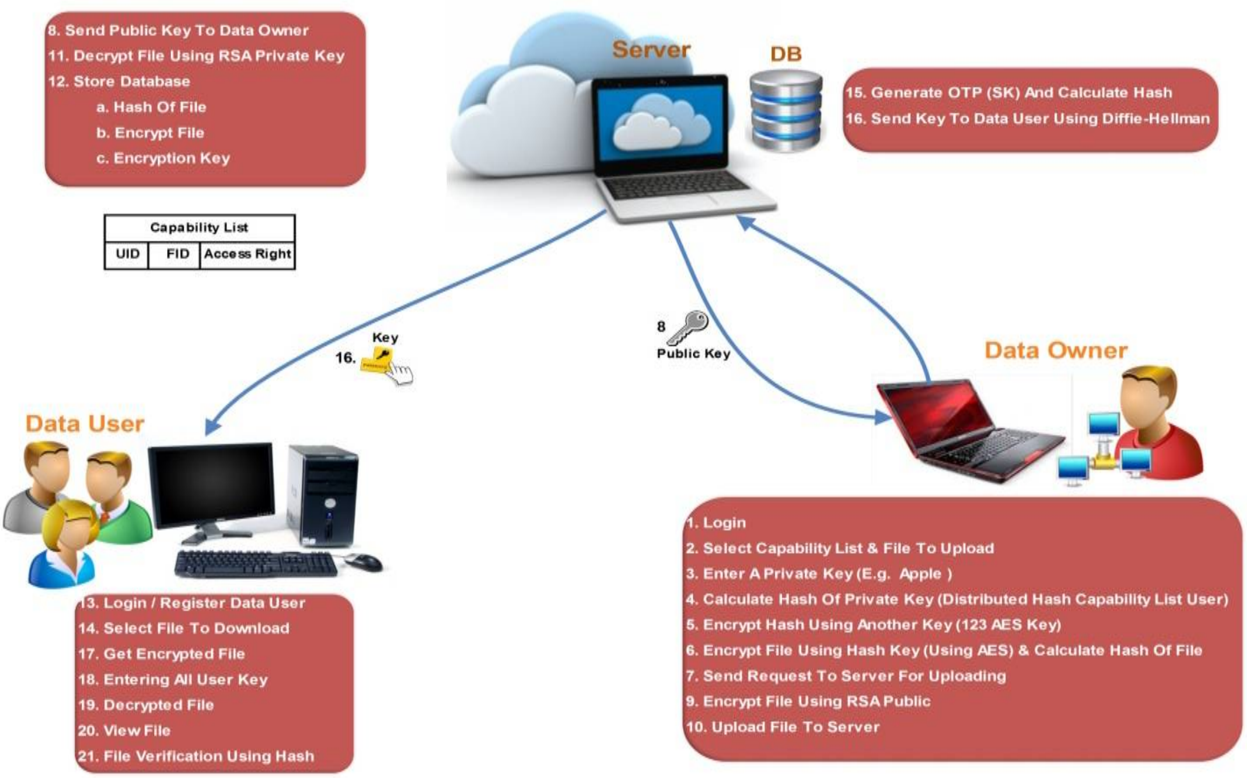
8. Send Public Key To Data Owner
11. Decrypt File Using RSA Private Key
12. Store Database
a. Hash Of File
b. Encrypt File
c. Encryption Key
Capability List
UID
FID
Access Right
Server
DB
15. Generate OTP (SK) And Calculate Hash
16. Send Key To Data User Using Diffie-Hellman
Key
16.
8
Public Key
Data Owner
Data User
13. Login / Register Data User
14. Select File To Download
17. Get Encrypted File
18. Entering All User Key
19. Decrypted File
20. View File
21. File Verification Using Hash
1. Login
2. Select Capability List & File To Upload
3. Enter A Private Key (E.g. Apple)
4. Calculate Hash Of Private Key (Distributed Hash Capability List User)
5. Encrypt Hash Using Another Key (123 AES Key)
6. Encrypt File Using Hash Key (Using AES) & Calculate Hash Of File
7. Send Request To Server For Uploading
9. Encrypt File Using RSA Public
10. Upload File To Server

Fig.1 System Architecture

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.

VIII. EXPERIMENTAL RESULTS

Our experimental record on this scheme, there are basically three entities: Data Owner (DO), Cloud Service Provider (CSP) and Users. Users are divided in groups on some basis such as location, project and department and, corresponding to each group, there is a single key for encryption and decryption of data. Each user in the group shares parts of the key. Data can be decrypted when at least threshold number of users will present. This scheme not only provides data confidentiality by all means but also reduces the number of keys. To achieve fine-grained data access control, the approach has used capability list [6]. It is basically row-based decomposition of access matrix. In capability list authorized data and operations for a user are specified. It is better suit than Access Control List (ACL) [5] because ACL specifies users and their permitted operation for each data and file. It is practically inefficient that two users require same data and have same operations on it. In this paper, the approach has used the modified Diffie-Hellman algorithm to generate one time shared session-key between CSP and user to protect the data from outsiders. To ensure data integrity the approach has used MD5 [4]. In this paper review we used three algorithms.

IX. PROPOSED ALGORITHM

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 ( (Fi))) || (CPList))
        CPList || Ekt (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, (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 that File directly to CSP.
```

Algorithm 2 illustrates the procedure required after a new File creation. When a new File is created, DO fills entries for that File in Capability List containing UID, FID and AR. DO generates a symmetric key (KT) and encrypts File with that symmetric key (KT). Now, DO encrypts the updated CPList, Encrypted File and symmetric key (KT) with its private key after that public key of CSP and sends these to the CSP. When CSP receives these, it updates Capability List, Encrypted File List and sends encrypted symmetric key (KT) to respective user group. 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(((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 3 describes how data are exchanged securely between CSP and the user by use of modified Diffie-Hellman algorithm. We called it modified D-H algorithm as we encrypt the D-H parameters using the public key of one side and, using nonce in each direction during session key (KS) generation and data transfer. It helps to counter the man-in-the middle attack. After available of keys and tokens, the user may request for data to CSP. CSP initiates modified D-H key exchange with the user, if request is authentic. We assume that the session key (KS) is shared between CSP and the user by modified Diffie-Hellman algorithm. Now, CSP encrypts the encrypted File (Fi) and its digest (Di) with the shared session key (KS) and sends it to the user.

X. MODULE INFORMATION

Data Owner(DO) :
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.all the data is resides on the DO side.user are request for the data to the DO.as well as data owner is responsible for the maintain the CL(capability list).

Cloud Service Provider(CSP) :
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 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.

User or Clients :
User are responsible for the get the data from the DO through the CSP. Numbers of users are connected to the CSP. Data can be decrypted when at least threshold number of users will present. This scheme not only provides data confidentiality by all means but also reduces the number of keys.

XI. 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 data, the scheme has used capability list.

REFERENCES

1. Sushil Kr Saroj CSE Department Greater Noida "Threshold Cryptography Based Data Security in Cloud Computing" , Computers Networks,Systems and Industrial Engineering (CNSI), IEEE International Conference on Computational Intelligence & Communication Technology,vol :2,2015.
2. H. Zhong, and H. Zhen, "An Efficient Authenticated Group Key Agreement Protocol," Security Technology, 2007 41st Annual IEEE International Carnahan Conference on, vol., no., pp.250-254, 8-11 Oct. 2014.
3. 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 ACIS/JNU International Conference on, vol., no., pp.248-251, 23-25 May 2011.

4. 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.
5. S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, “Over-encryption: Management of access control evolution on outsourced data,” in Proc. of VLDB’07, 2007.
6. 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.

BIOGRAPHY

Prasad Bangar,Kundan Randive,Jinesh Jain,Ganesh Bhise are students in Sinhgad College Of Engineering,Computer Department,Savitribai Phule Pune University.Our research interests are Computer Network etc.

DOI: 10.15680/IJIRCCE.2016. 0401047