Scientific Journal Impact Factor (SJIF): 1.711

IJMTER

International Journal of Modern Trends in Engineering and Research

ISSN (ONLINE) : 2349 - 9745

ISSN (PRINT) : 2393 - 8161

# An Efficient Secure Data Sharing Framework for Public Cloud Based Group Sharing

Silpa Salim[1], Elizabeth Jacob[2]

*[1,2]Department of Computer Science, IIET, Nellikuzhi*

**Abstract—**Many of the public cloud computing services have appeared for data storage in group applications. Two important problems that arises when sharing group data in public cloud are the privacy and security of group member's data. Cloud service providers are separate administrative entities and users don't have access to the cloud internal operational details. Because of the semi trust nature of cloud service provider, the traditional security technologies cannot be directly applied to the public cloud based group data sharing applications. We propose an efficient secure framework for public cloud based group application. This framework combines proxy signature scheme, incremental proxy re-encryption scheme and cipher text policy attribute based encryption together into a protocol. The group leader can choose one or more of the group members for the management of the group by adopting proxy signature scheme. Incremental proxy re-encryption scheme is used to simplify the file modification by the data owner by dividing the file into different data blocks. Cipher text policy attribute based encryption is used to perform encryption on data blocks.

**Keywords—** Group sharing, public cloud, incremental proxy re-encryption

## I. INTRODUCTION

Many cloud computing service providers have introduced to provide data storage in cloud. When group data is stored in the cloud main advantage is that the data owner can share their data with the desired members in the group. The cloud is managed by cloud service provider. Cloud service provider cannot be treated as trusted because of its semi-trust nature. So the traditional security storage schemes cannot be directly applied to this public cloud based group data storage application. Therefore an efficient secure group sharing framework needs to be developed.

The cloud servers managed by cloud providers are not fully trusted by users while the data files stored in the cloud may be sensitive and confidential, such as business plans. To preserve data privacy, a basic solution is to encrypt data files, and then upload the encrypted data into the cloud. So valid users can download and decrypt the file. But the session key updating and distribution is a major problem. Another method is the use of Digital Envelope. In these methods, the computing and communication overhead of digital envelopes generation and the computational and communication overhead of session key updating are major problems. The efficiency of these schemes depends on the fact that cloud servers should be trusted otherwise they can launch the collusion attack with some leaving group members.

The group application in cloud can be formed as follows. The group leader initializes the group. Group leader is the creator of the group. He buys storage space from the cloud. Group leader can also create new user roles. Each user can issue a join request to the group leader. When the group leader accepts the request that user becomes a valid group member. Moreover the group leader can set two or more group members as group administrator. The setting and revoking of group administrator can only be performed by the group leader. Each of the group members in the group can perform file uploading and downloading. The changes of membership in group make secure data sharing extremely difficult. On one hand, the anonymous system challenges new granted users to learn the content of data files stored before their participation, because it is impossible for new granted users to contact with anonymous data owners, and obtain the corresponding decryption keys. On the other hand, an efficient membership revocation mechanism without updating the secret keys of the remaining users is also desired to minimize the complexity of key management.

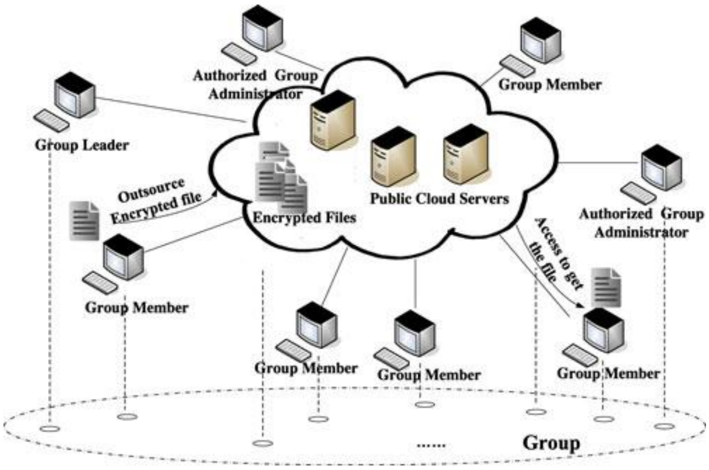

*Fig. 1. Public cloud based group sharing framework*

Some of major requirements of secure data sharing in the Cloud are as follows. Firstly the data owner should be able to specify a group of users that are allowed to view his or her data. Specified members within the group should be able to gain access to the data anytime, anywhere without the data owner’s intervention. No one, other than the data owner and the members of the group, should gain access to the data, including the Cloud Service Provider. The group leader should be able to add new users to the group. The data owner should also be able to revoke access rights against any member of the group over his or her shared data. The group leader should be able to grant the privilege of group management to any of the chosen group members and this group management privilege can be revoked at any time by the group leader.

Main contribution of this work can be summarized as follows: This work supports group key pair updating whenever a group member uploads a new file which transfers most of the computational and communication complexity to cloud servers without leaking the privacy. Group management privilege can be granted to any of the group members, which can be revoked at any time. This work enables the group to negotiate and update the group key pairs even though not all of the group members are online together. Any offline group member can launch group key synchronization when he/she becomes online again in the next time. This work allows the data owner to encrypt and upload the file to a number of users based on their role. All the corresponding group members can download the file to the specific folder. If the data owner wants to modify the encrypted file uploaded on the cloud storage, this work proposes an incremental version of proxy re-encryption scheme for improving the file modification operation by the data owner.

## II . RELATED WORKS

In the work achieving secure scalable fine grained access control in cloud computing, an encrypted file can be decrypted by a user only if he/she has all of the file’s attributes. By using proxy re-encryption, the computing complexity of digital envelope generation for a session key of a sharing file decreases at the data owner’s side. For each one-time session key, the data owner needs to compute only one digital envelope by using his/her own public key. Based on the proxy re-encryption algorithm, cloud servers can compute digital envelopes for all intended recipient.

The efficiency of this scheme relies on that there is high attribute variability between different files and high attribute variability between different users. But in group applications, different group members usually have same or similar interests, and they usually have attributes in common between them. In the scenario of interest based group sharing, the communication and computing overhead of user revocation will be dependent on the size of the group. The efficiency of these schemes depends on the assumption that cloud servers must be absolutely trusted. Otherwise, cloud servers can launch the collusion attack with some curious leaving group members. So, in order to protecting files from the prying eyes of curious cloud servers and leaving group members, the data owner needs to re-generate his key pairs and re-generate n-1 proxy-re-encryption keys when revoking a group member. This computing overhead is very high for the data owner.

In traditional studies, the security of group communication applications can be ensured by group key agreement. All of these schemes require all group members to be online together. Unfortunately, it’s difficult to have all members online together in group applications in the cloud. How to make sure that such group application in the cloud are secure and reliable remains a challenging problem. When the data owner modifies the file, the modified version of the file should be available to all the specified group members. Our scheme uses incremental proxy re-encryption scheme to simplify the file modification operation by the data owner.

## III. PROPOSED WORK

The proposed method is shown in Figure 2, which includes four phases such as group initialization, group administration and privilege management, data sharing management, incremental proxy re-encryption.

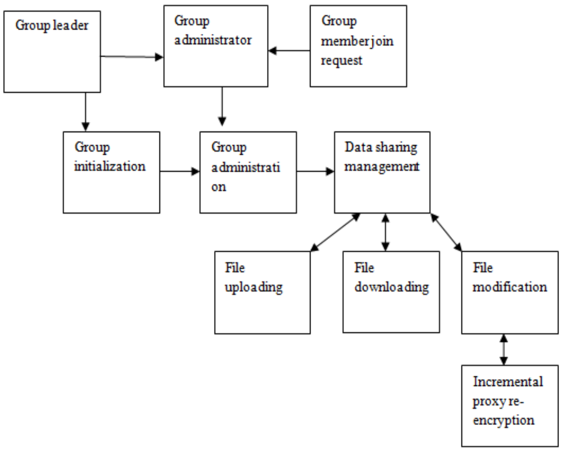

*Fig. 2. Proposed group sharing framework*

The four stages are given by:

- Group initialization phase. Each user can give a join request to the leader.
- Group administration phase. In this phase the group leader grant the group administration privilege to some specific group members
- Data sharing management phase. In this phase each valid group member can perform the operations of file uploading and downloading.
- Incremental proxy re-encryption scheme for improving the file modification operation by the data owner. The re-encryption is performed using cipher text policy attribute based encryption.

### A. Group Initialization

First group leader creates a group in the cloud. For a group, there is only one group leader who is the creator of the group. Then the group leader obtains storage space from the cloud provider. Leader can select any member to perform group management and this management permission can also be revoked by leader. Each user performs the member registration and sends a join request to the group leader. The user becomes a valid member only if the joining request is accepted by the group leader. Each group member can implement file download and upload operations in the authenticated group.

### B. Group Administration

Next phase is the group administration and privilege management step. In this phase the group leader can grant the privilege of group management to any of the chosen group members. This group member then becomes the group administrator. There is one, or more authorized group administrators in a group. They can maintain group membership such as joining and leaving of group members. Their permission of group management can be revoked by the group leader at any time. They also can perform all the functions of basic group members, such as file uploading and downloading.

### C. Data Sharing Management

Each group member in the group can perform data sharing operations. Each valid member can perform file uploading and downloading. Before uploading a file to cloud servers, the data owner symmetrically encrypts the file with a randomly chosen session key. The data owner selects a number of recipients for the file. Then all those specified members will receive the key of that particular file. The corresponding members can download the received file to the specific folder.

### D. Incremental Proxy Re-encryption

The proposed scheme uses the concept of incremental cryptographic for improving the file modification operations of proxy re-encryption scheme in terms of communication and processing overhead. In this scheme a file is divided into different data blocks. To encrypt each block the user generates random number and encrypts the message using the personal private key. Similarly, the message authentication code for each block of file is generated using cryptographic hash function. Subsequently, the generated message authentication codes are concatenated and cryptographic hash function is applied to get the final message authentication code for verifying the integrity of the uploaded file. The encryption on data blocks is performed by using cipher text policy attribute based encryption. When the data owner wants to perform file modification, only the block containing the modified part is encrypted again. After modification the modified file will be available to all corresponding group members.

```
Algorithm 1: Incremental proxy re-encryption

Input : Original file

Output : Modified file

    1. Divide the file into different data blocks

    2. Encrypt each data blocks using cp-abe and store in the cloud

    3. Calculate the message authentication code for each data blocks

    4. Calculate the final message authentication code for each file
```

5. Open an editor window for data owner
6. Make file editing
7. Save the changes
8. Perform steps 3 and 4
9. Compare the message authentication codes before and after modification to get the modified block
10. Perform Re-encryption only on the modified blocks.
11. Send new keys of modified file to all members in the group.

## IV. CONCLUSION

This work proposed an efficient secure group sharing framework in public cloud computing environment. In this scheme, the group management privilege can be granted to some specific group members and this privilege can be revoked at any time by the group leader. This new group admin can perform the operations of join and leave. All the sharing files are secured and stored in cloud servers. The proposed scheme supports the updating of the group key pair whenever group member uploads a new file, which transfers most of the computational complexity and communication overhead to cloud servers without leaking the privacy. Any offline group member can launch group key synchronization when he/she becomes online again in the next time. Incremental proxy re-encryption scheme is used which divides the file into different data blocks which simplifies the file modification operation by the data owner. The different data blocks are encrypted and stored in the cloud by using cipher text policy attribute based encryption.

## REFERENCES

[1] C. K. Wong, M. Gouda, and S. S. Lam, “Secure group communications using key graphs,” IEEE-ACM Trans. Netw., vol. 8, no. 1, pp. 16–30, Feb. 2000.
[2] K. Ren, C. Wang, and Q. Wang, “Security challenges for the public cloud,” IEEE Internet Comput., vol. 16, no. 1, pp. 69–73, Jan./Feb. 2012.
[3] K. Yang, X. Jia, K. Ren, and B. Zhang, “DAC-MACS: Effective data access control for multi-authority cloud storage systems,” in Proc. IEEE Conf. Comput. Commun., 2013, pp. 2895–2903.
[4] L.Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, “Group formation in large social networks: Membership, growth, and evolution,” in Proc. 12th Int. Conf. Knowl. Discovery Data Mining, 2006, pp. 44–54.
[5] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing,” in Proc. IEEE 29th Conf. Comput. Commun., 2010, pp. 534–542
[6] T. Jung, X. Li, Z. Wan, and M. Wan, “Privacy preserving cloud data access with multi-authorities,” in Proc. IEEE Conf. Comput. Commun., 2013, pp. 2625–2633.
[7] T. Malkin, S. Obana, Satoshi, and M. Yung, “The hierarchy of key evolving signatures and a characterization of proxy signatures,” in Proc. Int. Conf. Adv. Cryptol., 2004, pp. 306–322.
[8] Y. Kim, A. Perrig, and G. Tsudik, “Group key agreement efficient in communication,” IEEE Trans. Comput., vol. 53, no. 7, pp. 905– 921, Jul. 2004
[9] Y. Kim, A. Perrig, and G. Tsudik, “Simple and fault-tolerant key agreement for dynamic collaborative groups,” in Proc. 7th ACM Conf. Comput. Commun. Security, 2000, pp. 235–244.
[10] Y. Kim, A. Perrig, and G. Tsudik, “Tree-based group key agreement,” ACM Trans. Inf. Syst. Security, vol. 7, no. 1, pp. 60–96, 2004.