International Journal of Scientific Research Engineering & Technology (IJSRET)
Volume 2 Issue 9 pp 578-582 December 2013 www.ijsret.org ISSN 2278 – 0882

Secure Single-Sit Data Sharing for Dynamic Groups in Cloud

Ranjith.K, Pursuing M.Tech,
Department of Information Technology,
V.S.B Engineering College,
Karur, India,
mails2ranju@gmail.com.

Prasanth SP, Pursuing M.Tech,
Department of Information Technology,
V.S.B Engineering College,
Karur, India,
prasanthitboss@gmail.com.

***Abstract*—**Cloud computing has emerged as a new type of commercial paradigm. Cloud and its services got wide popularity among the IT industry in a short span of time. With the character of low maintenance and ease of use cloud services are extended for resource sharing among cloud users. Sharing among group is still a challenging issue, due to frequent change of membership. The main issues are related to identity privacy and data privacy. Group sharing may not scale well for individual data access when sharing highly confidential and sensitive data in the group. In this paper, we propose a secure dynamic group sharing among several members of a group rather than individuals. Any user can share data with others in the group, but access available in the presence of minimum no of users only. The storage overhead and cryptography is independent of no of users in the group.

***Index Terms*—Cloud computing, resource sharing, identity privacy, dynamic group, cryptography.**

I. INTRODUCTION

Cloud computing has emerged as a new type of commercial paradigm. With the character of low maintenance and ease of use, cloud computing gained more popularity among individual as well as organizational users. Mobile internet paved a new way for cloud computing. Many users are interested to join clouds, but security is still a challenging issue for cloud users. Even though, cloud computing and its popularity increases dramatically. As people rely more and more on the internet and cloud technology, security of their privacy takes more and more risks.

One of the fundamental services offered by cloud is data storage. Let us consider a real time scenario. A company allows its employees belong to same department to store and share files in cloud. By utilizing cloud, staff can get free from local data management and related issues. However it poses some confidentiality risks. The cloud service provider or third party may not fully trusted by the users. Designing an efficient and secure sharing scheme for groups in cloud is a difficult task due to following reasons. First, identity privacy is one of the challenging problems for cloud deployment. Without guarantee of identity privacy, users are unwilling to join the cloud group. Because, their real identity, other related information may vulnerable to an attack. Second, it is highly recommended that the entire cloud services should be fully used by all the users. Support of multi-owner data system is needed. In a single-owner system group manager only has full access to the cloud, whereas in multi-owner system, all user gets equal preference in the group.

Last but not least, groups are normally dynamic in practice. This dynamic nature raises key management and distribution issues in revocation. An efficient revocation mechanism without updating secret keys of remaining users is desired to reduce the key management complexity. Even though one more challenge arises due to untrusted members in the group. If the group member is an untrusted one, he decrypts all the files and discloses to other organizations or out of the group.

To solve the above listed challenges, we propose secure single-sit data sharing for dynamic group in cloud. Main contributions in this paper are:

1. We propose a secure multi-owner data system where any member in the group can share data files with others in the same group.
2. Our proposed scheme supports dynamic groups, such that without updating user's private key, any user can join or leave the group at any point of time. Such users get full access once they joined the group or loss all access rights in the group. This dynamic nature of group does not make any key management issues in the group.

3. We propose single-sit sharing scheme where any individual user can't get decrypted file by using his own private key. It needs a minimum no of keys called threshold to decrypt the data files.

II. RELATED WORK

In [1], Liu et al. proposed a dynamic group sharing system, where any user can join or leave the group at any time. User revocation is performed by the owner using a revocation list generated. Revocation can be performed without disturbing other users in the group. Similarly, newly joined user can directly decrypt the files available in the group without contacting the owner.

In [4], Kallahalla et al. proposed a cryptographic storage system that enables secure file sharing on untrusted servers called Plutus. The file is divided into no of small files each then encrypted with secret key. Users are provided with corresponding key to decrypt required file blocks. However, heavy key distribution overhead experienced for large scale file sharing. Also, to perform user revocation entire file blocks need to be re-encrypted for redistribution

Ateniese et al. [5] proposed proxy re-encryption to secure distributed storage. The user encrypts the file using symmetric keys which further encrypted by a master public key. For access control, server uses proxy cryptography to re-encrypt the content using master keys. Unfortunately, a collusion attack between untrusted server and revoked users may launch, which enables them to know the decryption keys used for the encryption.

In [2], Yu et al. brought a scalable and fine grained data access in cloud using Key Policy-Attribute based Encryption method. The data owner can use any random key to encode the file, where the chosen random key is again encrypted along with a set of attributes using KP-ABE. Then, the group manager provides an access structure and decryption keys to the user. The ciphertext can be decrypted by the user if and only if the attributes satisfy the access structure assigned. To achieve user revocation, data owner needs to update all attributes and keys. Here single owner sharing does not allow multiple owner sharing and maximum utilization of cloud resources.

Lu et al. [6] proposed secure provenance scheme, which built on group signatures and ciphertext- policy-attribute based encryption techniques. The system is set up with a single attribute. Thus, any user can obtain two keys after registration: a group signature key and an attribute key. The user is now able to encrypt files using attribute based encryption and group members can decrypt the same using attribute keys obtained. User signs files during encryption using signature keys for privacy preserving and traceability. However this scheme does not support user revocation.

From the above analysis, we can observe that how to securely share data in a multiple-owner manner for dynamic groups while preserving identity privacy from an untrusted server remains to be a challenging issue. In this paper, we propose single-sit sharing scheme for dynamic groups. Compared with existing works, our proposed work provides the following unique features.

1. Any user in the group can store and share data at any time.
2. Encryption complexity and size of cipher text is independent with the no of revoked users.
3. User revocation can be performed without updating private keys of other users.
4. Any user can decrypt files without contacting data owner but within the presence of min. no of users.

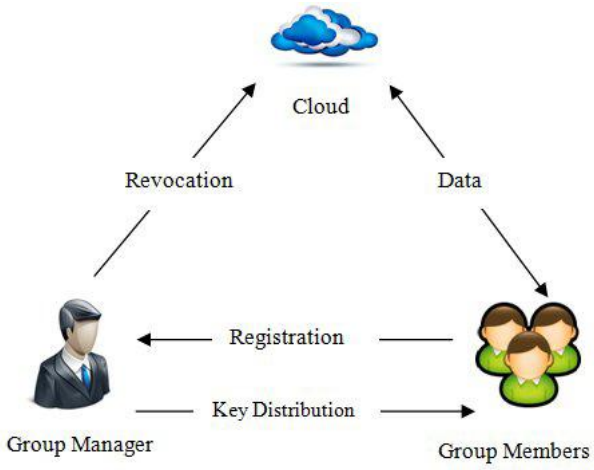

Fig.1. System Model

III.SYSTEM MODEL AND DESIGN GOALS

a) System Model

We consider a cloud architecture combining with an example that consist a company allows its staff to store and share files within the same group or other departments. The system model consist three entities: the cloud, group manager, group members as in the fig.1.

The cloud is a large repository of data files that stored by the cloud users. Cloud is managed by cloud service providers (CSPs). CSPs support the cloud users for storing and retrieving shared files. We assume that cloud is honest but curious.

Group manager is responsible for system parameter generation, user registration approval, revocation, revealing real identity of data owner when any disputes came. Possibly the group manager may be the administrator of the organization or the team leader of the group. Therefore, we assume that group manager is trusted by other parties.

Group members are set of registered users who are all allowed to store and share their private data in the cloud. Usually the group members are the team members or staffs in the organization. Group membership is dynamically changed due to the staff resignation, newly joining in the organization.

b) Design goals

Our system is designed to achieve the following goals. Each of them described briefly as follows.

Access control: Group members including group manager can access the cloud if they have valid key. Unregistered members and revoked members are strictly prohibited from accessing the cloud.

Data confidentiality: Unauthorized users can't know the content of stored data including the cloud. One of the challenging issues is to maintain confidentiality in dynamic group. Because new user should able to decrypt the files while revoked users unable to decrypt shared files.

Anonymity and traceability: Anonymity guarantees flexible access to cloud without revealing real identities of user. Although anonymity provides protection to identity it poses some insider attack risks. Traceability allows identification of real identity of an inside attacker incase of any attack.

Efficiency: Efficiency of the proposed system can be explained as follows: Any user can store and share files with other users in the cloud. User revocation can be performed without disturbing remaining users. Remaining users don't need to update their private keys. Also, newly joined users can decrypt the files without contacting data owner. Insider attack problem can be resolved. At the same time, data confidentiality can be maintained as the system requires min. no of user's private key to decrypt the files.

IV. PROPOSED SCHEME

a) Overview

In groups or large organizations, at least one inside attacker will be present. Insider attacker is more dangerous than external hacker. So, in our proposed system we add a single-sit decryption mechanism by which individual users can't decrypt the files using his/her own key alone. It requires more than one key that satisfy the min. threshold. Encryption can be performed by individual data owners. In addition to this group signature is added along with the message to identify the data owner which is maintained by the data owner. This information used only when any disputes came until which remains as a secret. User revocation is performed by group manager. So, users are not facing any burden due to user revocation or revoked user.

b) Scheme Description

System Initialization: Group Manager is responsible for system initialization. Initially creating a group and invite members relevant to the group to join the group. It defines bilinear group *G1* and forms a bilinear map

e: G1 x G2 -> G2. That returns a public key which forms master key of the group maintained by group manager.

User Registration: User Registration starts with a user *i* with identity *Idi*. The Group Manager selects any number *xi* randomly from Z and computes *Xi*, *Yi* as follows:

$$\left.\begin{aligned} Xi &= (1/\mu + xi).P \;\varepsilon\; G1 \\ Yi &= (xi/\mu + xi).G \;\varepsilon\; G1 \end{aligned}\right\} \quad (1)$$

Now, Group Manager adds (*Xi*, *xi*, *Id*) into the group of users. This can be used to trace the user later if needed. After successful registration the user *i* gets private key (*xi*,*Yi*, *Xi*), this key further used for Group Signature generation and File Decryption.

User Revocation: User Revocation is performed by Group Manager through a Revocation list that is publicly available. Revocation List is based on time stamps. Let *Idg* be the Group identity. The tuple (*Xi*, *xi*, *ti*) represents that a user *i* with partial private key (*Xi*, *xi)* is revoked at time *ti*.

File Upload: To store and share data file in the group, data owner need to perform following steps.

1. Get the revocation list from cloud. The member makes a request to cloud using group identity *Idg*. Cloud verifies group identity and returns revocation list RL corresponding to the group.
2. Verification of validity of RL received. First checks whether marked date is fresh or not. Later, verifies signature *sig(RL)* using equation *e(V, f1(RL))=e(P, sig(RL))* .If the RL is invalid user can terminate the scheme here.
3. Encrypting data file F. This encryption process considers two cases as follows:

(a) No Revoked user in the list.

i. Select unique file identity *Idf* .
ii. Choose any random number m ε Z.
iii. Compute parameters C1, C2, M, C as follows:

$$\left.\begin{aligned} C1 &= m.A\ \varepsilon\ G1 \\ C2 &= m.P\ \varepsilon\ G1 \\ M &= Z^{m}\ \varepsilon\ G2 \\ C &= Encm(M) \end{aligned}\right\} \quad (2)$$

(b) There are *n* revoked users in the list.

i. Select unique file identity *Idf* .
ii. Choose any random number m ε Z.
iii. Compute parameters C1, C2, M, C as follows:

$$\left.\begin{aligned} C1 &= m.A\ \varepsilon\ G1 \\ C2 &= m.Pn\ \varepsilon\ G1 \\ M &= Z_n^{m}\ \varepsilon\ G2 \\ C &= Encm(M) \end{aligned}\right\} \quad (3)$$

4. Select any random number £ and compute *f(£)*. The Hash value obtained may use for the deletion of data file in future. Also, data owner adds *(Idf, £)* in to his local storage.
5. The uploaded file F at time *tf* with signature σ on *(Idf, C1, C2, C, f(£),tf)* computed using private key *(X,x)*.
6. Successful updating of file F into cloud adds the file in the local shared data list maintained by the Group Manager. On receiving data, cloud checks its signature for validity. If more number of users revoked in the group, revocation verification also performed.

File Deletion: If the file is no longer needed in the group then either group manager or owner can remove the file from cloud through deletion operation. To delete any file with identity *Idf* Group Manager computes *(μ f1 (Idf))* and sends it along with *Idf* to the cloud. Now, cloud will delete the data file if *e(μ f1(Idf),P)=e(V,f1(Idf))* holds.

File Access: To decrypt the files stored in the cloud in a single-sit user needs to satisfy (*K*,*n*) threshold secret sharing scheme. Where, *K* represents the key from *n* users and *n* represents minimum number of shares needed to decrypt the file. Use revocation list obtained from the cloud to check validity of *n* users to be participated in single-sit scheme.

$$K=\{k1+k2+k3+\ldots..+kn\} \text{ ————————} \quad (4)$$

Where, *kn* is the share of *nth* participant. Individual users use their own private key (*Xi, xi)* to decrypt the file. During decryption the signature of data owner verified for ensuring integrity.

Algorithm: Share Computation algorithm for Shamir's scheme.

Data: Secret Key K, No of shares n.

Result: n shares *k1, k2,... kn.*

Set *f0=K;*

Construct *f(x) = f0+f1x+f2x+……+fk-1;*

Evaluate *ki = f(i), (i=1,2….n);*

Traceability: When any disputes occurred during the entire group operations, tracing can be performed by group manager to identify the person's real identity who made illegal actions through Signature of individual members. Group manager use his own private key to trace malicious user in the list.

V. CONCLUSION

In this paper, we proposed a dynamic grouping and data sharing scheme in a single-sit. Single-sit scheme needs minimum no of participants to decrypt the file that already fixed by group manager. No individual user able to access files using his own private key alone which considerably reduces insider attack. As we use public revocation list, user revocation can be performed without changing private keys of existing users. Detailed analysis of our system shows that the system can provide enough security as guaranteed.

REFERENCES

[1] Xuefeng Liu, Yuqing Zhang, "Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the

Cloud", IEEE Transactions on Parallel and Distributed Systems, vol. 24, No. 6, June 2013.

[2] Shucheng Yu, Cong Wang, Kui Ren, Wenjing Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing", Proc. IEEE INFOCOM, pp. 534-542, 2010.

[3] Dan Bogdanov, "Foundations and properties of Shamir's sharing scheme", Research seminar in Cryptography, University of Tartu, Institute of Computer Science, May 2007.

[4] M. Kallahalla, E.Riedel, R.Swaminathan, Q.Wang and K.Fu, "Plutus: Scalable Secure File sharing on Untrusted Storage", Proc. USENIX Conf. File and Storage Technologies, pp. 29-42, 2003.

[5] G.Ateniese, K.Fu, M.Green, and S.Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage", Proc. Network and Distributed Systems Security Symp. (NDSS),pp. 29-43, 2005.

[6] R.Lu, X.Lin, X.Liang, and X.Shen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing", Proc. ACM Symp. Information, Computer and Comm. Security, pp. 282-292, 2010.

[7] V.Goyal. O.Pandey, A.Sahai, and B.Waters, "Attribute –Based Encryption for Fine-Grained Access Control of Encrypted Data", Proc. ACM Conf. Computer and Communication Security (CCS), pp. 89-98, 2006.

[8] B.Waters, "Ciphertext-Policy Attribute based Encryption: An Expressive, Efficient, and Provably Secure Realization", Proc. Int'l Conf. Practice and Theory in Public Key Cryptography Conf. Public Key Cryptography, 2008.

[9] D.Boneh, X.Boyen, and H.Shacham, "Short Group Signature", Proc. Int'l Cryptography Conf. Advances in Cryptography (CRYPTO), pp. 41-55, 2004.

[10] A. Shamir, "How to Share secret", Comm. ACM, vol. 22, No. 11, pp. 612-613, 1979.

ABOUT THE AUTHORS

Ranjith.K received the B.E-Computer Science and Engineering Degree from Sri Jayaram Engineering College affiliated to Anna University in 2011 and currently pursuing his M.Tech-Information Technology Degree in V.S.B Engineering College, affiliated to Anna University. His area of interest includes Cloud Computing, Distributed Systems, and Computer Networks.

Prasanth SP received the B.Tech-Information Technology Degree from Nandha Engineering College, affiliated to Anna University in 2012 and currently pursuing his M.Tech-Information Technology Degree in V.S.B Engineering College, affiliated to Anna University. His area of interest includes Distributed Computing and Cloud Computing.