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 6, June 2016

A Review on Proxy Signature Based Data Sharing in Large Dynamic Groups in the Cloud

Shaikh Muddassar Shahanawaj, Dr. K. V. Metre

ME Student, Department of Computer Engineering, MET's Institute of Engineering, BKC Nashik, Maharashtra, India

Associate Prof., Department of Computer Engineering, MET's Institute of Engineering, BKC, Nashik, Maharashtra, India

ABSTRACT: Cloud computing is more popular today and it is useful for an economical purpose and gives efficient solution for sharing group resource to the cloud users with the properties like character of low maintenance and high reliability. Unfortunately, sharing data in a multi-owner manner while preserving data and identity privacy from a cloud which cannot be trusted is one of the challenging issue, due to the frequent change of the membership. A secure multi-owner data sharing scheme, for dynamic groups in the cloud can be proposed. Every user on a cloud used leveraging group signature and dynamic broadcast encryption techniques and can anonymously share data with others. Proxy signature technique can be applied so that group leader can grant the special rights of group management to one or more chosen group members. Meanwhile, the storage overhead and encryption computation cost are independent with the number of revoked users.

KEYWORDS- Secure group sharing, public cloud computing, group key agreement, proxy signature.

I. INTRODUCTION

Cloud computing is an information technology field which provide us with intrinsic resource-sharing and low-maintenance characteristics. In cloud computing, there are many cloud service providers (CSPs), such as Amazon, which provide various services to cloud user with the help of powerful datacenters. By moving the systems from the local data management systems into cloud servers, we can enjoy high-quality services and can save significant investments on local infrastructures [1].

Data storage is one of the most fundamental services offered by cloud providers. Consider an example of a practical data application. To store and share files in same group or department, IT company allows its staffs the storage into the cloud. Cloud can be utilized for the staff by completely releasing them from troublesome local data storage and maintenance. However, significant risk to the confidentiality of the stored files can be a threat. The servers managed by cloud providers cannot be fully trusted by users as the data files stored in the cloud may be sensitive and confidential, such as project plans, etc. To hide contain of data, there is a solution to encrypt data files, and then the encrypted data should be store into the cloud. Unfortunately, providing a high level, efficient and secure data sharing scheme for groups in the cloud is not an easy task [2].

There are many security schemes for data sharing on un-trusted cloud servers provided by CSP's have been proposed. In these approaches, user stores the encrypted data files in un-trusted cloud storage and distributes the corresponding decryption keys only to authorized data owners. Thus, unauthorized users as well as cloud storage servers cannot learn the content of the files because they have no information about the decryption keys. But, the complexities of user revocation and participation in this manner are linearly increasing with the number of revoked data owner and number of users for the data file, respectively. Lu et al. proposed a secure scheme based on the cipher text-policy attribute-based encryption technique by setting a group with a single attribute, which allows any user in a group to share data with other user in the group. But, the issue of user revocation was not considered in their idea. Based on the key policy attribute based encryption (KP-ABE) technique, Yu et al. presented a fine-grained data access control and scalable scheme in cloud computing. Uncertainly, the single owner manner makes it difficult to adoption of their

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 6, June 2016

search into the case, where user is granted to store and share data. [4][5][6][7][8][9].

II. RELATED WORK

Plutus is a cryptographic storage system technique that enables secure file sharing on un-trusted servers proposed by Kallahalla et al. In these the file is divided into file groups and each file group is encrypted with a unique file-block key. The file-block key can be shared with other user to decrypt the file by the data owner. However, heavy key distribution overhead for large-scale file sharing has to be done by the data owner. If user revocation occurs again, the file-block key needs to be updated [4].

Un-trusted server includes two parts of the file stored on the cloud server:

1. File metadata.
2. File data.

The file metadata contains the access control information, a series of encrypted key blocks, etc. each of which is encrypted and present inside the public key of authorized users. Thus, the size of the file metadata is dependents upon number of authorized users. The file metadata needs to be updated, if user revocation takes place which is an intractable issue especially for large-scale sharing. Further research has been done and the NNL construction is used for efficient key revocation. However, the private key of each user in an NNL system needs to be recomputed when a new user joins the group, which may limit the application for dynamic groups. Another problem is that the computation overhead increases linearly with the sharing scale [5] [10].

Ateniese et al. borrowed proxy re-encryptions to secure distributed storage. Specifically, with unique and symmetric content keys, the data owner encrypts blocks of content, which are further encrypted under a master public key. The cloud server uses proxy cryptography to directly re-encrypt the content of the key(s) from the master public key for access control to the other user. Unfortunately, the un-trusted cloud server and any revoked malicious user can launch a collusion attack, due to which they can learn the decryption keys of all the encrypted blocks [6].

Yu et al. proposed a fine-grained data access control and scalable scheme in cloud computing based on the KPABE technique. Random key is used by the data owner to encrypt a file, where the random key is again further encrypted using KPABE. The group manager then assigns an access structure and the related corresponding secret key to authorized users, so that a user can only decrypt a cipher text if the access structure satisfies the data file attributes. User revocation can be achieved, if the manager delegates tasks of data-file re-encryption and user secret key update to cloud servers. However, the single owner manner makes it difficult for the implementation of applications with the scheme, where any member in a group should be allowed to store and share data files with others into the cloud [3].

After these a secure provenance scheme was presented by Lu et al., which was built upon group signatures and cipher-text-policy attribute-based encryption techniques. Particularly, the system is a set with a single attribute that was presented in this scheme. Every user is given two keys after the registration *Group signature key and Attribute key.*

Thus, with the help of attribute-based encryption, any user is able to encrypt a data file and by using attribute key the others member in the group can decrypt the encrypted data. Meanwhile, the user signs encrypted data with his/her group signature key for privacy preserving and traceability. However, their system does not support user revocation. Observation can be done from the above analysis, how to securely share data files in a multiple-owner manner while preserving identity privacy for dynamic groups from an un-trusted cloud still remains to be a challenging issue [7].

III. PROPOSED SYSTEM

The system described in this scheme mainly consists of three components: Administrator, Group Leader and Group Member.

A. System Architecture

In the proposed system, there are 3 kinds of users in cloud based group sharing applications:

1) Administrator
2) Group Leader (GL)
3) Group Member (GM)

Administrator: In the system, there is only one admin, who is responsible for group creation and the top level group administrator. Administrator should buy or obtains storage and computing resources from the cloud service provider.

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 6, June 2016

He/she can give rights (privilege) to specific group members to manage the group, and these rights (privilege) of management can also be removed by him/her. Initial group security parameters for all group members in the group are also provided by administrator.

Group Leader (GL): In the system, there will be one or more group leader, who will be the leader of the group. GL responsibility is to maintain group membership, and acts as a responsible person to implement group key updating. GL privileges of management can be revoked by the administrator at any time. GL can also enjoy all the functions of basic group members, such as uploading and downloading.

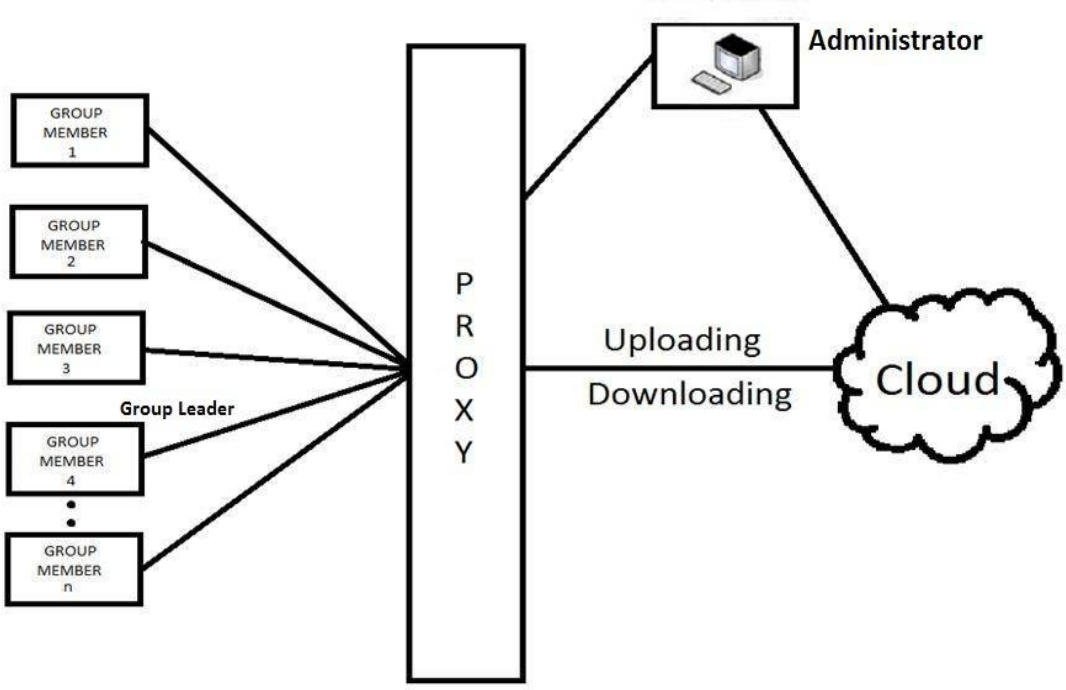

Fig.1 Proposed System Architecture

Group Member (GM): In the system, there will be one or more group member in the group. Every group member can implement file upload and download operations in the authenticated group. Each GM can get information from cloud servers related with public information and can compute the specific set of security parameters, such as group key pair.

Here, Admin $\in GA \subseteq GM$, $m < n$.

B. Proxy Signature

A signature scheme in which an original signer can authorize his/her signing authority to a proxy signer, and then proxy signer generates a signature on behalf of the original signer is called as *Proxy Signature.* A cloud service provider (CSP) can be convinced of the original signer's agreement on the signed message which was done by a proxy signature. [11] [12].

Three kinds of proxy signature algorithms have been proposed by researchers: full delegation, partial delegation and partial delegation by warrant. Full delegation and partial delegation are eliminated by partial delegation with warrant, which has been proved to be more secure and practical.

Consider 'A' as an original signer and 'B' as a proxy signer who consist of an authentic key pair (PrK_A and PuK_A), and (PrK_B and PuK_B) respectively. Consider m_w be A's warrant information for the delegation, which has semantic means including the original signer's identity, some information about the proxy signer (for example the identity), period of delegation validity, the qualification of messages on which the proxy signer can sign, etc. Let δ_A= Sign (PrK_A, m_w) be A's signature on the warrant m_w using his/her private key PrK_A. A transmits δ_A to the proxy signer B. Then partial delegation with warrant based proxy signature scheme is described as follows:

1) Proxy signature key generation (PKG) is a proxy signature key generating algorithm that takes original signers

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 6, June 2016

signature δ_A and proxy signers private key PrK_B as inputs, and outputs a proxy signature key pair ($PPrK_B$, $PPuK_B$). It is executed by the proxy signer:

$$(PPrK_B, PPuK_B) \leftarrow PKG\ (\delta_A, PrK_B). \quad (1)$$

2) Proxy signing (PS) is a proxy signing algorithm that takes proxy signature private key $PPrK_B$ and message m as inputs, and outputs proxy signature δ_p. It is executed by the proxy signer B:

$$\delta_p \leftarrow PS\ (m, PPrK_B). \quad (2)$$

3) Proxy signature verifying (PSV) is a proxy signature verifying algorithm that takes (δ_P, m, m_w, PuK_A, PuK_B) as inputs, and outputs either accept or reject. It is executed by any verifier:

$$PSV\ (\delta_P, m, m_w, PuK_A, PuK_B) = accept\ or\ reject. \quad (3)$$

C. Algorithm

The algorithms such as group initialization, member leaving process, group administrator leaving process, etc.

1) Group Initialization:

After obtaining storage and computing resource from the cloud provider, admin generates a shortest binary tree with n leaf nodes, where n is the number of group members (including GL itself) in the initial group. Each node of this binary tree can only have either zero (as a leaf node) or two child nodes, which means a node with one child is not allowed. Each one of these n leaf node is associated with one different group member. Administrator chooses a random number for each leaf node, and uses it to generate a secret key for the associated group member.

2) Member Leaving Process:

When a group member leaves, in order to provide backward secrecy, the group key pair should be updated, and all digital envelopes related to the sharing data in this group should be also updated and encrypted by the new group public key. In our scheme, one GA should mandate leaving group member's position in the binary tree and act as a sponsor to implement the group member leaving process. Detailed pseudo-code of the group member leaving process is shown in below in algorithm 2.

3) Group Leader Leaving Process:

Usually each GL mandates more than one leaf node, and he/she knows the secret keys of these leaf nodes. When a GL leaves, another GL should mandate these leaf nodes and change the security keys instead of him/her. The new mandating GL chooses a random secret key for each of the leaving GL's mandated leaf nodes, and computes the secret keys and blinded keys of all nodes in the path from each of these new mandated leaf nodes to the root node. All the paths from each of these leaf nodes to the root node form a sub-tree of the binary tree. Detailed pseudo-code of group administrator leaving process is given below in algorithm 3.

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 6, June 2016

Algorithm 1 : Group Initialization

INITIALIZE( )

1: **Generate a shortest binary tree T with n leaf nodes.**
2: each node in T only have 0 or 2 children
3: $\mathbf{PrK_G = INIT\_BINARY\_T(ROOT)}$
4: Root is the root node of T
5: $\mathbf{PuK_G = g^{PrK_G} \bmod p}$
6: **Associate each leaf node to a group member** $M_i, i = 1, 2, ..., n$ =0

INIT_BINARY_T(s)

1: **if** (s.left && s.right)
2: **do** ln = s.left
3: rn = s.right
4: $ln.BK = g^{INIT\_BINARY\_T(ln)} \bmod p$
5: $K = ln.BK^{INIT\_BINARY\_T(rn)} \bmod p$
6: $ln.version = rn.version$
7: **else**
8: **do** k = random()
9: s.K = K
10: $s.BK = g^{s.K} \bmod p$
11: **return** K

MEM-LEAVING-PROC(leav_mem_n)

1: **if** leav_mem_n.sibling is also mandated by an GA
2: **do** Merge leav_mem_n, leav_mem_n.sibling,
3: and leav_mem_n.parent to one node
4: Set this merged node
5: as the current mandated node cur_mand_n
6: **else** Set leav_mem_n as cur_mand_n.
7: NODE_UPDATING_PROCESS(cur_mand_n)

NODE_UPDATING_PROCESS(cur_mand_n)

1: n.k = random()
2: $n.BK = g^{n.k} \bmod p$
3: n.version = 0
4: r = n.sibling
5: s = n
6: **for** t is each node in the path from n.parent to the root node
7: **if** $(t \neq root)$
8: **do** $t.K = s.K^{r.BK} \bmod p$
9: $t.BK = g^{t.K} \bmod p$
10: $t.version++$
11: $s = t$
12: $r = s.sibling$
13: **else** Exit;

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 6, June 2016

Algorithm 3 : Group Administrator leaving Process

ADMIN-LEAVING-PROCESS(leav_admin)

1: Paths from each of leav_admin's mandated nodes and associated node to the root node form a subtree T'
2: UPDAT_SUBTREE(Root in T')

UPDAT_SUBTREE(n)

1: **if** n is not a leaf node in T'
2: **do** $ln = n.left\ in\ T'$
3: $rn = n.right\ in\ T'$
4: **if** ln is in T'
5: **do if** rn is in T'
6: **do** $rn.BK = g^{UPDAT\_SUBTREE(rn)} \bmod p$
7: **else do** get rn.BK in T from Cloud Servers
8: $ln.K = UPDAT\_SUBTREE(ln)$
9: $K = rn.BK^{ln.k} \bmod p$
10: **else if** ln is not in T' and rn in T'
11: **do** $rn.K = UPDAT\_SUBTREE(rn) \bmod p$
12: $rn.BK = g^{rn.k} \bmod p$
13: get ln.BK in T from Cloud Servers
14: $K = ln.BK^{rn.k} \bmod p$
15: **else if** n is a leaf node in T'
16: **do** $K = random()$
17: n.K = K
18: $n.BK = g^K \bmod p$
19: return K

IV. CONCLUSION

A dynamic secure group sharing framework in public cloud computing environment and the management privilege can be granted to some specific group members based on proxy signature scheme, all the sharing files are secured stored in cloud servers and all the session key are protected in the digital envelopes. A user is able to share data with others in the group without revealing identity privacy to the cloud. Additionally, it will support efficient user revocation and new user joining. More specially, efficient user revocation can be achieved through a public revocation list without updating the private keys of the remaining users. The storage overhead and the encryption computation cost are constant.

ACKNOWLEDGEMENT

We are very much thankful to all the authors of the papers which we have referred and glad to express my sentiments of gratitude to all of them who rendered their valuable contribution and help to make this work successful.

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 6, June 2016

REFERENCES

[1] Xuefeng Liu, Yuqing Zhang, Member, IEEE, Boyang Wang, and Jingbo Yan," Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud", *IEEETransactions* on parallel and distributed systems, Vol. 24, NO. 6, June 2013.
[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia," A View of Cloud Computing", *Comm. ACM* vol. 53, no. 4, pp. 50-58, Apr. 2010.
[3] S. Kamara and K. Lauter," Cryptographic Cloud Storage", *Proc. Intl Conf.*Financial Cryptography and Data Security (FC), pp. 136-149, Jan. 2010.
[4] S. Yu, C. Wang, K. Ren, and W. Lou, Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing, *Proc. IEEE* infocom, pp. 534-542, 2010.
[5] 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.
[6] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, Sirius: Securing Remote Untrusted Storage, *Proc.* Network and Distributed Systems Security Symp. (NDSS), pp. 131-145, 2003.
[7] 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.
[8] 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.
[9] B. Waters, Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization, *Proc.* Intl Conf. Practice and Theory in Public Key Cryptography Conf. Public Key Cryptography, 2008.
[10] D. Naor, M. Naor, and J.B. Lotspiech, Revocation and Tracing Schemes for Stateless Receivers, *Proc.* Ann. Intl Cryptology Conf. Advances in Cryptology (CRYPTO), pp. 41-62, 2001.
[11] S. Kim, S. Park, and D. Won, Proxy signatures, revisited, *Proc.* 1st Int. Conf. Inf. Commun. Security, pp. 223232, 1997.
[12] B. Lee, H. Kim, and K. Kim, Secure mobile agent using strong non designated proxy signature, *Proc.* 6th Australian Conf. Inf. Security Privacy, vol. 2119, pp. 474486, 2001.

BIOGRAPHY

Shaikh Muddassar S. received a bachelor's degree in Computer Engineering from Pune University in 2011. Now pursuing M.E (4th sem) degree in Computer Engineering from Pune University, Pune, Maharashtra, India in 2016 and currently working as a Assistant Professor in polytechnic college. My research interest includes Network Security and cloud computing.