International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue VIII, August 2018- Available at www.ijraset.com

An Improved Algorithm for Multi-Owner Data Sharing using Policy based Signcryption in Clouds

Ayushi Shukla[1], Prof. Umesh Lilhore[2], Prof. Nitesh Gupta[3]
[1]*NRI Institute of Information science & Technology, Sagar, India*
[2, 3]*Asst Professor, NRI Institute of Information science & Technology, Bhopal, India*

Abstract: ***Cloud Computing is an internet based recent technology which provides storage and retrieval of Data and Sharing of Data. With the character of low maintenance, cloud computing provides an economical and efficient solution for sharing group resource among cloud users. Sharing of data between multiple users can be achieved through the wide booming technology known as cloud computing. The shared data can be accessed by the group members of the cloud. Unfortunately, this method is untrusted hence that the cloud is mentioned as untrusted cloud. The Existing methodology implemented [1] for the Multi Owner Data Sharing over Cloud provides efficient and scalable Data Sharing but takes high computational Cost of time and Encryption and decryption cost can be minimized. Hence in this paper a more scalable and efficient Multi Owner Data Sharing technique is implemented which contains a) the model is fully trusted where Data Owner and Key Generation Center and Server and Receiver is used. B) it delivers an indication to the clienteles that the CSP is not cheating by storing fewer copies. C) it rope subcontracting of active information, i.e., it provisions block-level processes, such as chunk alteration, supplement, obliteration, and attach. D) it permits official operators to flawlessly admittance the folder reproductions stowed by the CSP. E) Data owner → encrypts file → (key generation center & TPA) generate key and re-encrypt the data → stored on multiple servers → access by the authorized users. F) Includes Message Verification also.***

I. INTRODUCTION

CLOUD COMPUTING is recognized as an alternative to traditional information technology due to its intrinsic resource-sharing and low-maintenance characteristics. In cloud computing, the cloud service providers (CSPs), such as Amazon, are able to deliver various services to cloud users with the help of powerful datacenters. By migrating the local data management systems into cloud servers, users can enjoy high-quality services and save Significant investments on their local infrastructures. One of the most fundamental services offered by cloud providers is data storage. Let us consider a practical data application. A company allows its staffs in the same group or department to store and share files in the cloud.

By utilizing the cloud, the staffs can be completely released from the troublesome local data storage and maintenance. However, it also poses a significant risk to the confidentiality of those stored files. Specifically, 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. Unfortunately, designing an efficient and

secure data sharing scheme for groups in the cloud is not an easy task due to the following challenging issues. First, identity privacy is one of the most significant obstacles for the wide deployment of cloud computing. Without the guarantee of identity privacy, users may be unwilling to join in cloud computing systems because their real identities could be easily disclosed to cloud providers and attackers. On the other hand, unconditional identity privacy may incur the abuse of privacy. For example, a misbehaved staff can deceive others in the company by sharing false files without being traceable. Therefore, traceability, which enables the group manager (e.g., a company manager) to reveal the real identity of a user, is also highly desirable. Second, it is highly recommended that any member in a group should be able to fully enjoy the data storing and sharing services provided by the cloud, which is defined as the multiple-owner manner. Compared with the single-owner manner [2], where only the group manager can store and modify data in the cloud, the multiple-owner manner is more flexible in practical applications. More concretely, each user in the group is able to not only read data, but also modify his/ her part of data in the entire data file shared by the company. Last but not least, groups are normally dynamic in practice, e.g., new staff participation and current employee revocation in a company. The changes of membership 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. Several security schemes for data sharing on untrusted servers have been proposed. In these approaches, data owners store the encrypted data files in untrusted storage and distribute the corresponding decryption keys only to authorized users. Thus, unauthorized users as well as storage servers cannot learn the content of the data files because they have no knowledge of the decryption.

The concept of provenance has been extensively studied for a long time, and widely used in the archival theory to denote the documented history of some data objects [9]. Given its provenance, a data object can report who created and who modified its contents. Therefore, once a dispute rises in document stored in a cloud, provenance is important for data forensics to provide digital evidences for post investigation. However, provenance is still an unexplored area in cloud computing [10], in which we need to deal with many challenging security issues. For example, in support of data forensics in cloud computing, the provenance information must be secured, i.e., they do not violate the information confidentiality and user privacy in cloud computing. Specifically, as the essential of bread and butter of data forensics in cloud computing, secure provenance should at least satisfy the following basic requirements:

1) *Unforgeability*: a genuine provenance record in cloud computing can effectively attest the ownership and process history of data objects stored in a cloud, any adversary cannot forge a valid provenance records, i.e., modifying an item in a existing record or directly introducing a new forged record without being detected.
2) *Conditional Privacy Preservation*: To ensure information confidentiality and anonymous authentication in cloud computing, a genuine provenance record should also be conditional privacy preserving [11, 12, 13, 14]. That is, only a trusted authority has the ability to reveal the real identity recorded in the provenance, while anyone else cannot. Secure provenance is vital to the success of data to solve the challenges presented above, we propose Multi owner data sharing in cloud. The main contributions of this paper include:

a) Propose a multi-owner data sharing scheme. It implies that any user in the group can securely share data with others by the untrusted cloud.

b) Provide privacy-preserving access control to users, which guarantees any member in a group to anonymously utilize the cloud resource. Moreover, the real identities of data owners can be revealed by the group manager when disputes occur.

II. LITERATURE SURVEY

In 2010, Lan Zhou projected a scalable and fine-grained data access control scheme by defining access polices based on data attributes and KP-ABE technique [3]. The arrangement of attribute based encryption, proxy re-encryption and lazy re encryption allows the data owners to allocate the calculation tasks to untrusted server without enlightening the necessary contents of data. By using Key Policy Attribute-Based Encryption (KP-ABE), the random key is further encrypted with a set of attributes. Then the approved users are assigned an access formation and matching secret key by the Group admin. Hence onlythe users with data file attributes that gratify the access structure can decrypt a cipher text. This approach has some drawback such as multiple owner manners. In multiple owner manner it is not maintained by this system so that those single owner manners make it less flexible as only Group Admin are answerable for altering the data file shared. And user secret key needed to be updated after each revocation.

In 2013, Boyang Wang, Hui Li and Ming Li focus on a public verifier is able to audit the integrity of shared data without retrieving the entire data from the cloud, and also without learning private identity information of the group members [4]. Group dynamics such as user join and user revocation are efficiently handled by outsourcing signature updating operation store the cloud via a secure proxy re-signature scheme. A public verifier is able to efficiently audit the integrity of shared data in the cloud for a group of users without retrieving the entire data from the cloud. The public verifier, who is only a third-party to the group, is not able to reveal confidential information of the group, such as which user in the group or which block in shared data is a higher valuable target than others. The original useris the original owner of data and creates shared data in the cloud in the first place. After shared data has been created in the cloud, not only
the original user, but also group users are able to access and modify shared data. The original user also acts as the group manager, who is able to add new users to share data and revoke users from the group. The cloud offers data storage and sharing services to users. Due to the existence of hardware/software failures and internal attacks, users do not fully trust the cloud with the integrity of shared data stored in the cloud. The Third Party Auditor, who is a public verifier, is able to audit the integrity of shared data on behalf of users.

In 2014, Ning Cao, Cong Wang, Ming Li, Kui Ren and Wenjing Lou data owners are motivated to outsource their complex data management systems from local sites to the commercial public cloud for greater flexibility and economic savings [5]. Protecting data privacy the sensitive data have to be encrypted before outsourcing, which obsoletes traditional data utilization based on plaintext keyword search which is widely used on the plaintext data, cannot be directly applied on the encrypted data. Downloading all the data from the cloud and decrypt locally and all these multi keyword search schemes retrieve search results based on the existence of keywords, which cannot provide acceptable result ranking functionality.

In 2003, the E.Goh and his team proposed a system named "SIRIUS"[6].In that the files stored on the untrusted server include two parts: file metadata and file data. In the file meta data it includes a series of encrypted key blocks and each one is encrypted by the public key of the authorized users. Here also the user revocation is an intractable issue for the large-scale file sharing. Since every time the file's meta data also need to be updated. In the next version, the NNL construction is used for the efficient key revocation. But in this also whenever a new user joins in the group, there is no need to recomputed the private keys of every user.

In 2010, Yu.et.al proposed a scalable and fine grained data access control scheme in the cloud computing[7]by using the KP-ABE technique. In this scheme, the data owners encrypt the file with a rand key where this random key is further encrypted with a group of attributes y using the KP-ABE and the respected secret keys to the authorized users, then the user can only decrypt the cipher text if the data file attributes match with the access structure. To achieve the user revocation the cloud servers takes the responsibility from manager of the tasks such as file re encryption and the secret key updates. Here in this scenario, the single owner manner may create the problem with the implementation of applications where all the users can share data with the others.

In 2014, Shani Raj, Varghese paul and Nishana Rahim presented a multi owner information exchange is an model for sharing business data of large organizations which allows owners to create, manage and control their information data in cloud [8]. In policy based data sharing, each user has an access policy for the system and each file has some file access policy. The data may have multiple owners and the owners register into system as a group of users but having individual access keys and passwords. Anyone in the group can store and share the data. The policies of shared files are set by any of the owners and need approval of all the owners. Any change in file policy should need the group permission. File revocation means making file permanently inaccessible for all owners and it is done by deleting the secured decryption key with permission of all the owners along with the file policy. Custom files are shared to a set of selected users by the owners. Public files can be accessed by all users registered in the system and the access permissions are set by owners. Two kinds of users are in the system custom users and public users. For public users a public policy is available for data file access. The keys are sent to users through email and the users who clear all authentication tests are only authorized to access the data file.

III. PROPOSED METHODOLOGY

A. First of all Create Data Owner and CSP (Server & Data Center) and Receiver and Brokers and Key Generation Center.
B. Data Owner Starts Sending Files.
C. Create Dynamic Copies of the File.
D. Encrypt each of the file and send to the Server.
E. Server and Crucial Cohort Center creates a Master Secrete key and Re-encrypts the encrypted file and store to various Data Centers.
F. Receiver request to server to access the shared data.
G. Server is authenticating receiver.
H. If genuine server decrypts the dossier and send the scrambled file to the receiver.
I. Receiver finally decrypts the file.
J. Verification is done by the receiver.

The Planned Work implemented here contains the functions to be performed for the complete operations. The Data Owner uses KeyGen, CopyGen, TagGen, Perpare Update functions. The Third Party Auditor uses ExecUpdate and MasterSec and Prove. Key Generation Center runs the Algorithm MasterSec. While Verifier runs Verify algorithm.

1) *KeyGen(Pk,Sk):* Here this procedure is implemented at the Data Holder end as thriving as receiver's end. Here Pk and Sk denoted Community Key and Secluded Key respectively. This Funtion generates a couples of Solutions using Elliptic Curve Cryptography. The private Key Generated is kept private or secrete by the Data Owner, while Public Key Pk is shared publicly.
2) *Copy Gen(Ci,Fi):* This Meaning is used by the Information Proprietor for the generation of 'N' number of Copies to be shared to the Third Party Accountant to be stowed at various Cloud data Centers. Here the Quantity of Copies 'Ci' to be divided a File

'Fi' with 'N' copies as F={F1, F2,F3…..Fn}. The Generated files are then Encrypted by the Data Owner and send to the Third Party Auditor for Storing in the Haze Data Centers.

3) *TagGen(Sk,Fi):* This Function is Performed by the Data Owner on the Quantity of Files 'Fi'. Data Owner uses each of the Secrete Key Sk and the File Copies 'Fi' and Outputs a separate Tag Block set. Here each of the Tag block to be encrypted is linked with the Second Block Set by some ordered collections. The Linked List Generated by the Data Owner is then Send to the Third Party Auditor along with their Authenticating Sets.
4) *Prepare Update(D, Update Info):* This Function is track by the Information Owner to Update the Block Sets stored at the Data Centers. The Function takes an input D as Metadata that is stored at the Data Owner side and the Dynamic Operation to be achieved on the Specified Block Set. The Various Operations to be achieved on the Block set are Insertion or new Block File or deletion of some existing copy of File or Updation of existing File.
5) *Exec Update(Fi, Tag, Update Req):* This Procedure is track by the Third Party Accountant where the request from the Data Owner with Input such as File Copy number and the authenticator Tag and Update Request. Third Party Auditor first Decrypt the Copy to be stored at the Data Center and the Query the requested block and update the request and again Re-encrypt the File Block and stored at the data center
6) *Master Sec(R):* This Function is route by the Third Party Examiner for the generation of Master Secrete Key with the help of Random keys generated by the Third Party Auditor and Key Generation Center.
7) *Prove(Fi,Tag,chal):* This Algorithm is used by Third Party Auditor which Takes an input Parameters such as File Copy Number and Tag and Challenge value for Verification.
8) *Verify(Pk,P,D):* This Function is used by the Verifier on the basis of Input Request. Third Party Auditor check the Integrity of the receiver and Decrypt the Requested block and on the basis of identity copies are accessed and decrypted using Pubic Key.

IV. RESULT ANALYSIS

The table shown below is the analysis of length of Ciphertext for various multi-owner data sharing techniques in Cloud. The analysis is done for various number of members and accordingly length of Ciphertext is computed. The Proposed methodology shows efficient ciphertext length which reduces computational cost of the system.

	Length of ciphertext (byte)				
No. of Members	KACS	Mona	EEAC	SSEM	Proposed
20	15000	40000	100000	30000	10000
40	40000	75000	200000	55000	35000
60	60000	120000	300000	95000	50000
80	75000	150000	400000	130000	65000

Table 1. Analysis of length of Ciphertext

The Figure shown below is the analysis and comparison of various multi-owner Data Sharing techniques for the time consuming during Encryption. The Proposed methodology implemented provides less Computation Cost for the Encryption.

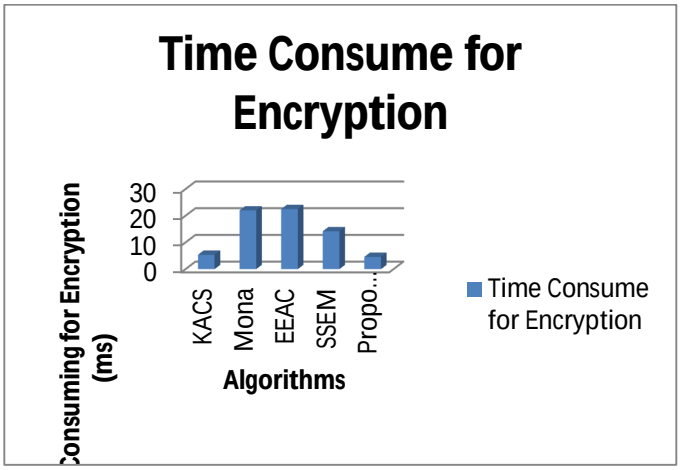

Figure 1. Comparison of Encryption Cost of Users

The Figure shown below is the analysis and comparison of various multi-owner Data Sharing techniques for the time consuming during Decryption. The Proposed methodology implemented provides less Computation Cost for the Decryption.

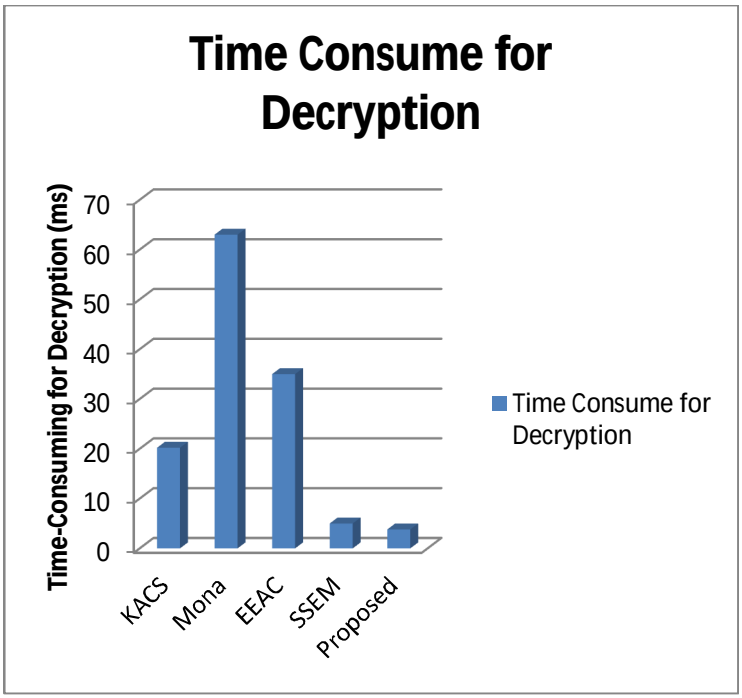

Figure 2. Comparison of Decryption Cost of Users

The Figure shown below is the analysis and comparison of various multi-owner Data Sharing techniques for the time consuming during Key Generation. The Proposed methodology implemented provides less Computation Cost for the Key Generation.

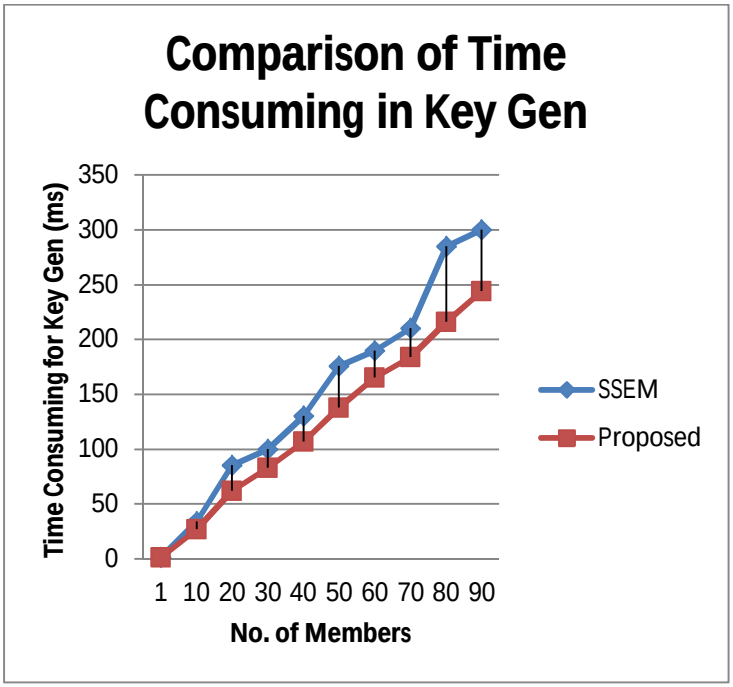

Figure 3. Comparison of Computational Cost for Key Generation

The table shown below is the analysis and comparison of various multi-owner Data Sharing techniques for the time consuming during Key Generation. The Proposed methodology implemented provides less Computation Cost for the Key Generation

	Computational Cost for Key Generation	
No. of Members	SSEM	Proposed
1	1	1
10	34	27
20	85	62
30	100	83
40	130	107
50	176	138
60	190	165
70	210	184
80	285	216
90	300	244

Table 2. Analysis of Computational Cost for Key Generation

V. CONCLUSION

Cloud computing enables various users to share or access resources over internet, but during the data sharing or storage in cloud security plays a vital role and hence various auditing protocols are implemented for the security of these cloud data and also provides privacy preservation between users.

The Existing methodology implemented [1] for the Multi Owner Data Sharing over Cloud provides efficient and scalable Data Sharing but takes high computational Cost of time and Encryption and decryption cost can be minimized. Hence in this paper a more scalable and efficient Multi Owner Data Sharing technique is implemented which contains a) the model is fully trusted where Data Owner and Key Generation Center and Server and Receiver is used. B) it delivers an indication to the clienteles that the CSP is not cheating by storing fewer copies. C) it rope subcontracting of active information, i.e., it provisions block-level processes, such as chunk alteration, supplement, obliteration, and attach. D) it permits official operators to flawlessly admittance the folder reproductions stowed by the CSP. E) Data owner → encrypts file → (key generation center & TPA) generate key and re-encrypt the data → stored on multiple servers → access by the authorized users. F) Includes Message Verification also.

REFERENCES

[1] Shungan Zhou, Ruiying Du, Jing Chen, Hua Deng, Jian Shen, Huanguo Zhang," SSEM: Secure, Scalable and Efficient Multi-Owner Data Sharing in Clouds", China Communication, 2016.

[2] 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.

[3] Shucheng Yu, Cong Wang, Kui Ren, Wenjing Lou," Achieving secure, scalable, and fine-grained data access control in cloud computing", INFOCOM'10 Proceedings of the 29th conference on Information communications Pages 534-542, 2010.

[4] Boyang Wang, Hui Li and Ming Li, "Privacy Preserving Public Auditing for Shared Cloud Data Supporting Group Dynamics", IEEE Communication and Information Systems Security 2013.

[5] Ning Cao, Cong Wang, Ming Li, Wenjing Lou," Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data", IEEE Transaction on Parallel and Distributed Systems, Volume 25, Issue 1, Jan 2014.

[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] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing,"Proc.IEEEINFOCOM,pp. 534-542, 2010.

[8] Shani Raj1, Dr. Varghese Paul2, Nishana Rahim3, "Multi-Owner Data Sharing in Cloud Storage Using Policy Based Encryption", International Journal of Emerging Technology and Advanced Engineering, Volume 4, Issue 5, May 2014.

[9] J. Li, N. Li, and W. H. Winsborough, "Automated trust negotiation using cryptographic credentials," in Proc. of CCS'05, 2005.

[10] M. Blaze. A cryptographic file system for UNIX. In CCS, 1993.

[11] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing remote untrusted storage," in Proc. of NDSS'03, 2003.

[12] 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.

[13] 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.

[14] Lu, R., Lin, X., Zhu, H., and Shen, X. Spark: a new vanet-based smart parking scheme for large parking lots. In The 28th Conference on Computer Communications (INFOCOM 2009) (Rio de Janeiro, Brazil, April 2009