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**

# Secured Group Data Sharing Over Cloud by Using Aggregate Key and Searchable Encryption Techniques

Snehal A. Patil

M.E. Student, Department of Computer Engineering, RMD Sinhgad School of Engineering, Savitribai Phule University, Pune, India

**ABSTRACT:** Cloud computing has given the users the accessibility to deploy number of files to the centralized cloud and share those with number of users. The flexibility of cloud computing always comes with the hurdles of security concerns. The data owner always needs to encrypt the files before uploading and it must decrypt before end users. This system needs secure storage of keys, but as files gets increased in number keys management becomes complex. We have proposed the system called as (KASE). The system proposes aggregate key for file sharing in groups and searchable encryption. We have observed that to create trapdoors manually for specific files it becomes very tedious and hence we have applied the TF-IDF technique to avoid manual job**.**

**KEYWORDS:** Searchable encryption, Key aggregation, cloud computing, group data sharing

## I. INTRODUCTION

Cloud systems can be used to enable data sharing capabilities and this can provide an abundant of benefits to the user. There is currently a push for IT organizations to increase their data sharing efforts. In enterprise settings, demand for data outsourcing is increased today. Data outsourcing should be assists in the strategic management of corporate data. This scheme is also used as a core technology behind many online services. These online services used for online application. Currently this scheme was easy to apply for free accounts for mail, photograph album, sharing of file with storage size more than 25GB[5]. Together by using the current wireless technology, cloud users can access almost all of their files, directories and emails by a mobile phone in any corner of the world.

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. Any member within the group should be able to gain access to the data anytime, anywhere without the data owners 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 data owner 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. No member of the group should be allowed to revoke rights or join new users to the group. One trivial solution to achieving secure data sharing in the Cloud is for the data owner to encrypt his data before storing into the Cloud, and hence the data remain information-theoretically secure against the Cloud provider and other malicious users. When the data owner wants to share his data to a group, he sends the key used for data encryption to each member of the group. Any member of the group can then get the encrypted data from the Cloud and decrypt the data using the key and hence does not require the intervention of the data owner. However, the problem with this technique is that it is computationally inefficient and places too much burden on the data owner when considering factors such as user revocation. When the data owner revokes access rights to a member of the group, that member should not be able to gain access to the corresponding data. Since the member still has the data access key, the data owner has to re-encrypt the data with a new key, rendering the revoked member's key useless. When the data is re-encrypted, he must distribute the new key to the remaining users in the group and this is computationally inefficient and places too much burden on the data owner when considering large group sizes that could be in excess of millions of users. Hence this solution is impractical to be deployed in the real-world for very critical data such as business, government and medical related data. While considering data privacy, we cannot rely on traditional technique of

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**

authentication, because unexpected privilege escalation will expose all data. Solution is to encrypt data before uploading to the server with users own key. Data sharing is again important functionality of cloud storage, because user can share data from anywhere and anytime to anyone. For example, organization may grant permission to access part of sensitive data to their employees. But challenging task is that how to share encrypted data. Traditional way is user can download the encrypted data from storage, decrypt that data and send it to share with others, but it loses the importance of cloud storage.

## II. LITERATURE SURVEY

This section provides the purpose of the feasibility study[12,14], the background of the proposed system, the methodology used for performing the study, and any reference materials used in conducting the feasibility study for the project.

*A. Multi-User Searchable Encryption*

There is a rich literature on searchable encryption, including SSE schemes [6] and PEKS schemes [8]. In contrast to those existing work, in the context of cloud storage, keyword search under the multi-tenancy setting is a more common scenario. In such a scenario, the data owner would like to share a document with a group of authorized users, and each user who has the access right can provide a trapdoor to perform the keyword search over the shared document. In MUSE scenario, although they all adopt single-key combined with access control to achieve the goal[9], and schemes are constructed by sharing the documents searchable encryption key with all users who can access it, and broadcast encryption is used to achieve coarse-grained access control. Attribute based encryption (ABE) is applied to achieve fine-grained access control[2] aware keyword search[13]. Attribute-based encryption (ABE) [8] allows each ciphertext to be associated with an attribute, and the master-secret key holder can extract a secret key for a policy of these attributes so that a ciphertext can be decrypted by this key if its associated attribute conforms to the policy.

*B. Multi-Key Searchable Encryption*

In a multi-user application, consider that the number of trapdoors is proportional to the number of documents to search over (if the user Multi-user Searchable Encryption provides to the server a keyword trapdoor under each key with which a matching document might be encrypted), firstly introduces the concept of multi-key searchable encryption (MKSE) and puts forward the first feasible scheme in 2013[10]. MKSE allows a user to provide a single keyword trapdoor to the server, but still allows the server to search for that trapdoors keyword in documents encrypted with different keys[7].

## III. PROPOSED ALGORITHM

A. *Design Considerations:*

The proposed product is designed for secure authorized access to cloud database. In this system, hybrid cloud architecture is introduced. The intermediate proxy server keys for privileges will not be issued to users directly, which will be kept and may become single point of failure. User cannot trust third parties as value of data is high for user. In this way, the users cannot give control of data to entrusted parties or in hand of cloud service provider. The authorized access can guarantee data confidentiality and optimal complexity to access that data.

KASE scheme to solve the above mentioned problem in problem definition. There is need to concentrate on two techniques. The architecture basically build on entities like CSP, Data owner, End User and Components like encryptor, decryptor and key generator. The Data owner first uploads the file/files and generates the aggregate key based on the ciphertext classes which is known as personal document hierarchy.

The key generation is basically composed as the combination of node id, node name, and parent node id of the hierarchy tree[4]. Once the aggregate key is generated file will be encrypted under the Elgamal algorithm and trapdoors will be generated for the files. These trapdoors and encrypted file will be uploaded on the cloud. Meanwhile data owner will share the Aggregate key with the End User.

End user on the other hand, will search the document based on the trapdoors and will decrypt the original files on submission of decryption 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 6, June 2016**

B. *Description of the Proposed Algorithm:*
The below proposed algorithm is basically deals with complete data sharing model in group. The each step represent the as describe the below.

**Step 1:** This step basically initialize the basic set up model and generating the public keys. Public keys are generating through random alphanumeric characters also considering global number of users.

Initialize the public parameters:

**Pk = random_gen(setoff(characters, numbers), random_user_id)**

**Step 2:** Dataowner will upload the files and select the users to be able to access the file and encryption of files.

Upload the file (Dataowner):
**E'(F) = Encrypt(F)**

**Step 3:** After encryption of files, the dataowner should also generate the trapdoors.

Generate Trapdoors:
**Td = Trap(F, w')**
Where w' = set of keywords

**Step 4:** User must create the aggregate key based on the cipher text class and send this to end users.

Generate aggregate key:
**K = Combine(F, tag(level, Pk))**

**Step 5:** At last, encrypted file and trapdoor must be send to cloud for storage.

Upload to CSP:
**CSP(F) = E'(F) + Td**

**Step 6:** At the end user side, end user will decrypt the file specifying the aggregate key.

Decrypt at user end:
**F = Decrypt(E'(F))**

## IV. PSEUDO CODE

- KeyGen():
  In this step system initialization will be done and public and private key pair will be generated in order to perform the encryption. This key will be based on the nodes of binary tree hierarchy which will be defined by data owner. Each node represents the privacy level of Data owner interest known as ciphertext classes.

- ElGamal Encryption Algorithm

**INPUT:** A plaintext integer m satisfying $0 \leq m < p$, and recipient's public key $p$, $g$, and $h$.
**OUTPUT:** Pair of integers$(v,w)$

1. **function** ENCRYPT $(m)$
2. Select a pseudo-random integer $k$, such that $0<k<p-1$
3. Compute $v=g^k \bmod p$
4. Compute $w=m \cdot h^k \bmod p$
5. **return** $(v,w)$
6. **end function**

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

- ElGamal Decryption Algorithm

**INPUT:** A ciphertext integers *v* and *w*, and own private-key *a* and *p*.
**OUTPUT:** message m, which is an ElGamal decryption of (*v*,*w*).

1. **function** DECRYPT(*v*,*w*)
2. Compute $t=v^{p-1-a} \mod p$
3. Compute $m= t \cdot w \mod p$
4. **return** m
5. **end function**

## V. SIMULATION RESULTS

The simulation studies involve To perform this operation under live cloud having minimum 32 GB RAM and with 5.5 version of MySQL. The algorithm Elgamal is selected to give better results. The size of the public key is minimal in Elgamal and it definitely affects the size of cipher text. The size of cipher text is lower as compare to other asymmetric encryption algorithm. The trapdoor0 generation is another important aspect of our application. The technique based on the synonym of the word under the paragraph is expected to have efficient keywords. The result are expected to give better execution time as compare to base paper.

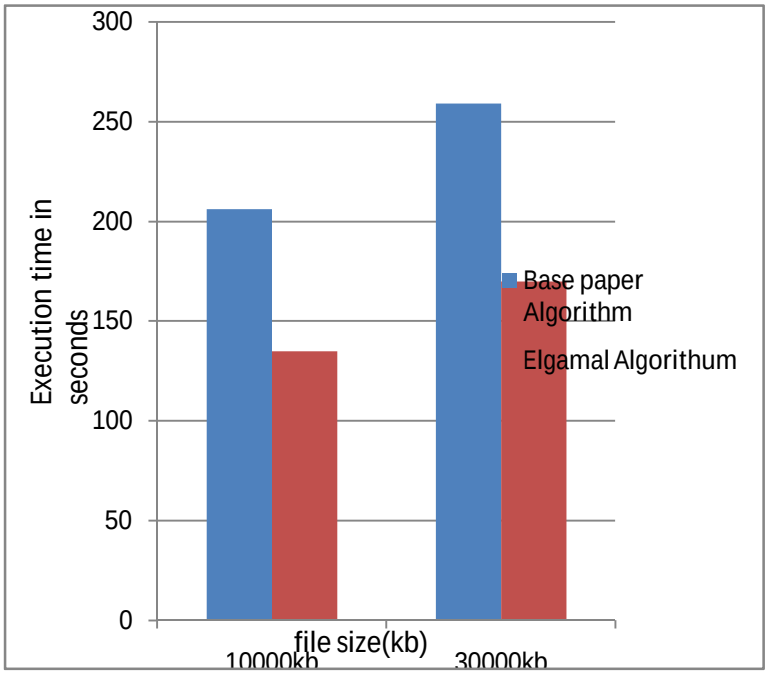


Fig. 1. Simulation result in terms of Execution time.
Fig. 2.

As shown in fig.1 we can see execution time required for our proposed solution is less than traditional. The base paper take almost 206 sec to encrypt the 10000kb files and Elgamal is expected to take approx 135 sec. The base paper take almost 259 sec to encrypt the 30000kb files and Elgamal is expected to take approx 170 sec.

## VI. CONCLUSION AND FUTURE WORK

The analysed scheme of Aggregate key along with searchable encryption in experiment. The aggregate key is basically composed from binary tree nodes from the hierarchy tree of data owner interest of privacy. The aggregate key allows the decryption of files at the child nodes of aggregate key node. The searchable encryption is basically composed of trapdoors concept. We have also studies the recent encryption algorithm ElGamal and synonym based keyword extraction scheme.

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. Baojiang Cui, Zheli Liu and Lingyu Wan "Key-Aggregate Searchable Encryption (KASE) for group data sharing via cloud storage", IEEE Transactions on Computers, Vol: pp no:99 year 2015.
2. V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted data," in Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS 06) ACM, 2006.
3. F. Guo, Y. Mu, Z. Chen, and L. Xu, "Multi-Identity Single-Key Decryption without Random Oracles", in Proceedings of Information Security and Cryptology, ser. LNCS,vol.4990.Springer,2007,pp.384-398.
4. M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, "Dynamic and Efficient Key Management for Access Hierarchies", ACM Transactions on Information and System Security (TISSEC), vol. 12, no. 3, 2009.
5. C. Chu, S. Chow,W. Tzeng., "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage", IEEE Transactions on Parallel and Distributed Systems, 2014, 25(2):468-477.
6. X. Song, D.Wagner, A. Perrig. "Practical techniques for searcheson encrypted data", IEEE Symposium on Security and Privacy, IEEE Press, pp. 44-55, 2000.
7. D. Boneh, C. G, R. Ostrovsky, G. Persiano, "Public Key Encryption with Keyword Search", EUROCRYPT 2004, pp. 506-522, 2004.
8. J. Li, Q. Wang, C. Wang, "Fuzzy keyword search over encrypted data in cloud computing", Proc. IEEE INFOCOM, pp. 1-5, 2010.
9. Z. Liu, Z. Wang, X. Cheng, et al, "Multi-user Searchable Encryption with Coarser-Grained Access Control in Hybrid Cloud", Fourth International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), IEEE, pp. 249-255,2013.
10. J.Li, X.F. Chen, M.Q. Li, J.W. Li, P. Lee, Wenjing Lou, "Secure Deduplication with Efficient and Reliable Convergent Key Management", IEEE Transactions on Parallel and Distributed Systems,25(6): 1615-1625, 2014.
11. . A. Popa ,N. Zeldovich. "Multi-key searchable encryption", Cryptology ePrint Archive, Report 2013/508, 2013.
12. 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.
13. P. Van,S. Sedghi, JM. Doumen. "Computationally efficient searchable symmetric encryption", Secure Data Management, pp. 87-100, 2010.
14. S. Kamara, C. Papamanthou, T. Roeder. "Dynamic searchable symmetric encryption", Proceedings of the2012 ACM conference on Computer and communication CCS), ACM, pp. 965-976, 2012