Impact Factor (SJIF): 5.301

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 5 , Issue 2, February-2018

Key Aggregated Multiuser File Sharing

Tanpure Trupti .[1], Tavandkar Shamal.[2],
Tope Monika .[3], Wadekar Dipali .[4]
And Guide Prof: V.N.Dhage

[1]Tanpure Trupti T., Computer Engineering/ SPPU, Pune (India)
[1]truptitanpure20@gmail.com
[2]Tavandkar Shamal J., Computer Engineering/ SPPU, Pune (India)
[2]shamaltavandkar2017@gmail.com
[3]Tope Monika M., Computer Engineering/ SPPU,
Pune (India)
[3]monikatope95@gmail.com
[4]Wadekar Dipali R., Computer Engineering/ SPPU,
Pune (India)
[4] dipaliwadekar95@gmail.com

I.Abstract

The important functionality in cloud is a sharing .To address user concerns over potential data leaks in cloud storage a common approach is for the data owner to encrypt all data before uploading them to the cloud, such that later the encrypted data may be retrieved and decrypted those who have the decryption keys. A key challenge to designing such encryption schemes lies in the sufficient management of encryption keys. This also implies the necessity of securely distributing to users a large number of keys for both encryption in search and user will have to securely store the received key and submit an same large number of keywords trapdoors to the cloud in order to perform search over the shared data. The practical problem of privacy preserving data sharing system based on public cloud storage which requires a data owner to distribute a large number of keys to users to enable them to access his/her documents. By addressing this practical problem which is hugely neglected in literature, we propose the novel concept of key aggregate Tagged File Searching (KATFS) in which data owner only need to distribute a single to user for sharing large number of data and user only needs to submit a single trapdoor to the cloud for querying the shared a large number of documents.

The ability of selectively distributing unreadable data with many users via public cloud storage may greatly ease security concerns over in advertent data leaks in the cloud. To designing the encryption schemes lies in the efficient management of encryption keys.The desired flexibility of sharing any group of selected documents with any group of users demands different encryption keys to be used for different data .The necessity of safely distributing to users a large number of keys for both encryption and search, and those users will have to securely store the received keys, and submit an equally huge number of keyword trapdoors to the cloud in order to perform search over the shared data. The required for secure communication, storage, and complexity clearly renders the approach impractical. The concept of Key Aggregate Tagged File Searching (KATFS) and instantiating the concept through a concrete KATFS scheme, in which a data owner only needs to distribute a single key to a user for sharing a large number of documents, and the user only needs to submit a single trapdoor to the cloud for querying the shared documents.The security analysis and performance evaluation both confirm that our pro-posed schemes are provably secure and practically efficient.

Keywords: Encryption, big-data, data sharing, cloud storage, Data Privacy, Tagging

II. Introduction

Cloud systems can be used to enable document distributing abilities 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. Together by using the present 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 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 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.

III. Literature survey

In [1], Kallahalla et al. proposed a cryptographic storage system that enables safe document sharing on untrusted servers, namedPlutus. By dividing files into file groups and encrypting each file group with a unique file block key, the data owner can share the file groups with others through delivering the corresponding lockbox key, where the lockbox key is used to encrypt the file block keys .However; it brings about a heavy key distribution over- head for large-scale file sharing. Additionally, the file-block key needs to be updated and distributed again for a user revocation.

In [2], files stored on the untrusted server include two parts: file metadata and file data. The file metadata implies the access control information including a series of encrypted key blocks, each of which is encrypted under the public key of trusted users. Thus, the size of the file data about data is proportional to the number of authorized users. The user revocation in the scheme is an intractable issue especially for large-scale sharing, since the file metadata needs to be updated.

In [3], the NNL construction is used for efficient key revocation. However, when a new user joins the group, the private key of each user in an NNL system needs to be recomputed, which may limit the application for dynamic groups. Another concern is that the computation overhead of unreadable linearly increases with the distributing scale.

[4] leveraged proxy re encryptions to secure distributed storage. Specifically, the data owner encrypts blocks of content with unique and symmetric content keys, which are further encrypted under a master public key. For access control, the server uses proxy cryptography to directly encrypt the correct content key(s) from the master public key to a granted user's public key. Unfortunately, a collusion attack between the untrusted server and any revoked malicious user can be launched, which enables them to learn the decryption keys of all the encrypted blocks. In Xuefeng Liu, Yuqing Zhang

IV. Existing system

With a traditional approach, primary user must securely send all the searchable encryption keys to secondary user. After receiving these keys,secondary user must store them securely, and then he must generate all the keyword trapdoors using these keys in order to perform a keyword search. Primary user is assumed to have a private document set {doc}ni=1, and for each document a searchable encryption key ki is used. Without loss of generality, we suppose primary user wants to share m documents {doc}mi=1

with secondary user. In this case, primary user must send all the searchable encryption keys {k}m=1 to secondary user. Then, when secondary user wants to retrieve documents containing a keyword w, he must generate keyword trapdoor Tri for each document doc with key k and submit all the trapdoors {Tri}m i=1 to the cloud server. When m is sufficiently large, the key distribution and storage as well as the trapdoor gener- ation may become too expensive for secondary user client-side device, which basically defies the purpose of using cloud storage.

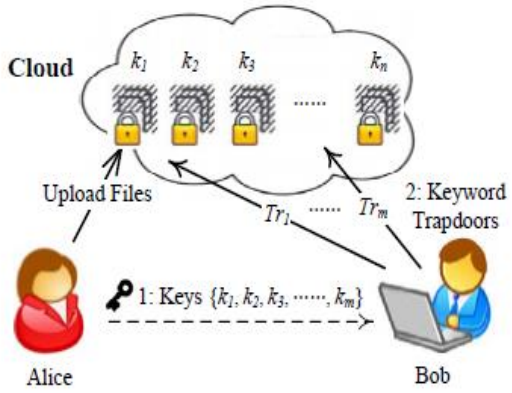

Figure 1 : Existing System

A) Disadvantages of existing system

1) Not more secure

2) Time requirement is more

3) Space complexity

v .Proposed system

The design of our proposed system draws its insights from both the multi-key searchable encryption scheme and the key-aggregate data sharing scheme. Specifically, in order to create an aggregate searchable encryption key instead of many independent keys, we adapt the idea presented in. Each searchable encryption key is associated with a particular index of document, and the aggregate key is created by embedding the owner's master-secret key into the product of public keys associated with the documents. In order to implement keyword search over different documents using the aggregate trapdoor, The cloud server can use this process to produce an adjusted trapdoor for every document.

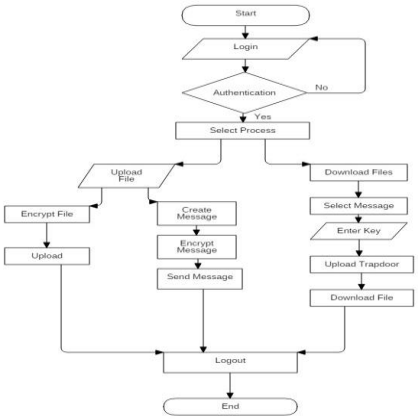

Figure 2 : Data Flow Diagram

B) Advantages of proposed system

1) More secure than existing system

2) Time requirement is less than existing system

3) Space Reduction

VI. System architecture

Authorized user can produce a keyword trapdoor via the Trapdoor algorithm using aggregate key, and submit the trapdoor to the cloud. After receiving the trapdoor, to perform the keyword search over the specified set of documents, the cloud server will run, this algorithm is run by the user who has the aggregate key to perform a search. It takes as input the aggregate searchable encryption key and keyword then out- puts only one trapdoor Tr.

One more important thing here using "Double Encryption", so system is more secure than existing system

Double encryption: A security device (with an additional public key or serial number) is still required. The encryption process is executed twice. First encrypt the plaintext corresponding to the public key or identity of the user. Then encrypt it again corresponding to the public key or serial number of the security device. For the decryption stage, the security device first decrypts once. The par- tially decrypted ciphertext is then passed to the computer which uses the user secret key to further decrypt it. Without either part (user secret key or security device) one cannot decrypt the ciphertext.

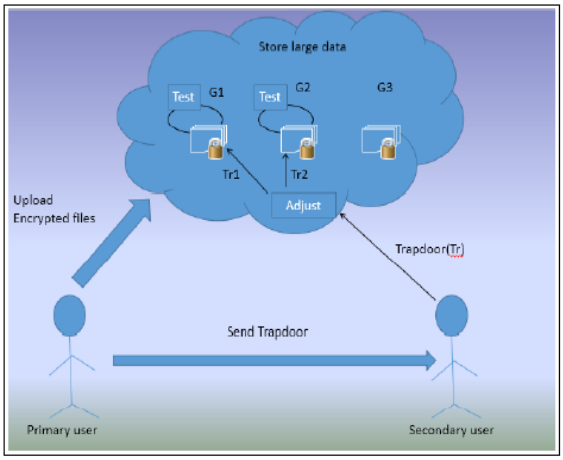

Figure 3 : Proposed System

• 1. System setup: When an organization submits a request,the cloud will create a database containing any information assign a ID for this organization and insert a record into table collage. Moreover, it assigns an administrator ac-count for the HOD. Then, the group data sharing system will work under the control of HOD. To generate the system pa- rameters .HOD runs the algorithm KATFS. Setup and updates the field parameters in table collage.

• 2. User registration: When adding a new member, the HOD assigns studID, studName, password and a key pair generated by any pub- lic key encryption (PKE) scheme for him, then stores the necessary information into the table member. A user's private key should be dis- tribute through a secure channel.

• 3. User login: our system relies on password verification for authenti- cating users. To further improve the security, multi-factor authentica- tion may be used when available.

• 4. Data uploading: To upload a document, the owner can encrypt the keys using his/her private key and store them into the table docs.

• 5. Data sharing: To share a group of documents with a target mem- ber, the owner runs KAE.Extract and KATFS.Extract to generate the aggregate keys, and distributes them to this member, then in- sets/updates a record in table shared Docs.

• 6. Keyword Search: Trapdoor to generate the keyword trapdoor for documents shared by each student, then submits each trapdoor and the related owners identity studID to the cloud.

• 7. Data retrieving: After receiving the encrypted document ,the mem- ber will run KAE.Decrypt to decrypt the document using the aggregate key dis-tributed by the documents student.

C) Applications:

1) Multi-user Searchable Encryption

2) Multi-Key Searchable Encryption

3) Key-aggregate Encryption for Data Sharing

VII. Conclusion

Considering the practical problem of privacy- preserving data sharing system based on public cloud storage which requires a data owner to distribute a large number of keys to users to enable them to access his/her documents, we for the first time propose the concept of key-aggregate searchable encryption (KASE) and construct a concrete KASE scheme. Both analysis and evaluation results confirm that our work can provide an effective solution to building practical data sharing system based on public cloud storage. In a KASE scheme, the owner only needs to dis- tribute a single key to a user when sharing lots of documents with the user and the user only needs to submit a single trapdoor when he queries over all documents shared by the same owner.

VIII. References

1. Kallahalla, E. Riedel, R. Swaminathan, Q.Wang, and K. Fu, Plutus: Scalable Secure document Sharing on Unauthorized Storage, Proc. USENIX Conf. File and Storage Technologies, pp. 29-42, 2003.

2. E. Goh, H. Shacham, N. Modadugu, and D. Boneh, Sirius: Securing Remote Untrusted Storage, Proc. Network and Distributed SystemsSecu- ritySymp.(NDSS), pp.131-145, 2003.

3. 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

4. G. Ateniese, K. Fu, M. Green, and S. Hohenberger, Improved Proxy Re- Encryption Schemes with App lications to Secure Distributed Storage, Proc. Network and Distributed Systems SecuritySymp. (NDSS), pp.29-43, 2005

5. S. Yu, C. Wang, K. Ren, and W. Lou, acquring Secure, Scalable, and Fine- Grained file Access maintain in Cloud Computing, Proc. IEEE IN- FOCOM, pp. 534-542, 2010.

6. In detail. [6] proposed a secure provenance scheme, which is built upon group signatures and cipher text-policy attribute- based encryption techniques. Particularly, the system in their scheme is set with a single attribute. Each user obtains two keys after the registration: a group signature key and an attribute key.Thus, any user is able to encrypt a data file using attribute- based encryption and others in the group can decrypt the encrypted data using their attribute keys. Meanwhile, the user signs encrypted data with her group signature key for privacy preserving and traceability. However, user revocation is not supported in their scheme Key aggregated Tagged File Searching.

7. In Xuefeng Liu, Yuqing Zhang [7] Mona: Secure Multi Ownership Data Sharing for Dynamic Groups in the Cloud In this paper, they propose a secure multi owner document sharing scheme, named Mona, for dynamic groups in the cloud. By leveraging group signature and dynamic broadcast encryption techniques, any cloud user can anonymously share data with others. Meanwhile, the storage overhead and encryption computation price of our scheme are does not dependent with the number of revoked users.

8. In Cong Wang KuiRen[8] Privacy Preserving Public Auditing for Secure Cloud Storage enabling public audit ability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced file and be worry without cost. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities toward user data privacy, and introduce no additional online burden to user. In this paper, a secure cloud storage system supporting security preserving public auditing. They further extend result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient.