A Proficient Aggregate Keyword Hunt Scheme Using MUSTKASE

[1] Mr.P.Krishnaraj, [2] Mr.T.Narendra Prasath, [3] Ms.P.Anantha Prabha

[1,2] UG Scholar, Department of CSE, Sri Krishna College of Technology, Coimbatore.

[3] Assistant Professor, Department of CSE, Sri Krishna College of Technology, Coimbatore.

Abstract— The ability of preferentially sharing encrypted data with unlike users through public cloud storage might really ease security distress, by possible data disclose in the cloud. A key test to design such encryption idea lies in the well-organized management encryption keys. Sharing the group of documents among of group of users with different set of keys needs to be sent in a secured way to the users. But it involves large number of complexities like encryption schemes ,storage space for received users. This paper proposes the concept of Multiple User Single Trapdoor Key Aggregate Searchable Encryption (MUSTKASE) and instantiates the idea through a real KASE scheme, in which a data owner wants to share out a single key to a user for distributing a large number of documents, and the user needs to present a single trapdoor to the cloud for questioning the shared documents.

Keywords—data sharing, Searchable encryption, data privacy, cloud storage, key aggregate, trapdoor

I. INTRODUCTION

Nowadays the storage in the cloud has materialized as a capable answer for suitable and on-demand accesses to huge amounts of information shared over the Internet. Business users are being paying attention by cloud storage due to its several benefits, including lower cost, better agility, and improved resource utilization. Everyday users are also sharing private data, such as photos and videos, with their friends through social network applications based on cloud. On the other hand, while benefiting from the expediency of sharing data through cloud storage, users are also gradually worried about accidental data reveal by the cloud. Such data revealing, will be performed by malicious opponent or a mischievous cloud operator, can habitually direct to severe violation of private data or confidential data regarding business. To speak about users anxiety over possible data reveal in cloud storage, a general approach is for the data owner to encrypt all the data before uploading them in to the cloud, such that presently the encrypted data may be get back and decrypted by individuals who contains the decryption keys. Such cloud storage is often called the cryptographic cloud storage .Though the encryption of data builds it demanding for users to search and then preferable retrieve only the data including the given keywords. A common solution is to employ a searchable encryption (SE) scheme in which the data owner is required to encrypt potential keywords and upload them to the cloud together with encrypted data, such that, for retrieving data matching a keyword, the user will send the

matching keyword to the cloud to react for the search over the encrypted data.

Even though merging a searchable encryption Scheme with cryptographic cloud storage can accomplish the essential security needs of a cloud storage, executing such a system for large scale application relating huge number of users and large number of files may still be delayed by realistic issues relating the well-organized management of encryption keys, which, to the finest of our knowledge. Primarily, the want for selectively sharing encrypted data with different users usually demands different encryption keys to be used for different files. On the other hand, this involves the number of keys that need to be spread to users, both for them to search over the encrypted files and to decrypt the files, will be relative to the number of such files. Such a large number of keys must not only be spread to users via secure channels, but also be securely stored and handled by the users in their devices. The implicit requirement for secure communication, storage, and computational difficulty may cause system ineffectiveness.

Here, we propose the novel concept of Multi User single Trapdoor key-aggregate searchable encryption (MUSTKASE), and instantiating the concept through a concrete KASE method. The proposed MUSTKASE scheme relates to any cloud storage that supports the searchable group data sharing feature, which means any user may prefer to distribute a group of files which are selective with a group of selected users, while permitting the final to carry out keyword search above the earlier. To maintain searchable group data sharing the main needs for efficient key management are double. Primarily, a data owner wants to allocate a single aggregate key (instead of a group of keys) to a user for sharing any number of files. Subsequently, the user needs to submit a single aggregate trapdoor to the cloud for performing keyword search over any quantity of shared files. KASE scheme can assure both requests.

II. RELATED WORK

This section reviews several categories of exixting solutions and explain their relationships to our work.

A. MULTI-USER SEARCHABLE ENCRYPTION (MUSE)

In existing mechanisms, the context of cloud storage, keyword search under the multi tenancy setting is a more common scenario . In such a case, the data owner would like to share a document with a authorized group of users, and each user with the access right can provide a trapdoor to search the keyword over the shared document[10].

In MUSE, the main difficulty is the right of users who can access the documents, and also reducing the number of shared keys and trapdoors is not discussed. MUSE can be with the help of Key aggregate searchable encryption .

B. *MULTI-KEY SEARCHABLE ENCRYPTION (MKSE)*

MKSE provides user a single keyword trapdoor to the server, but it use different encrypted keys to search for that trapdoor's keyword in documents.[9].

The problem of keyword search over a group of shared documents from the same user in the multi-user applications is discussed here and MKSE provides a solution to perform keyword search over a group of documents with only one trapdoor. In MKSE different trapdoors are used by user for different documents.

C. *PUBLIC KEY ENCRYPTION WITH KEYWORD SEARCH (PEKS)*

The data owner encrypts the potential keywords and upload them to the cloud together with encrypted data, such that, for retrieving data matching a keyword, the user will send the corresponding keyword trapdoor to the cloud for performing search over the encrypted data.

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, namely, the "multi-user searchable encryption" (MUSE) scenario.

MUSE schemes are constructed by sharing the document's searchable encryption key with all users who can access it, and broadcast encryption is used to achieve coarse-grained access control. In attribute based encryption (ABE) is applied to achieve fine-grained access control aware keyword search. As a result, in MUSE, the main problem is how to control which users can access which documents, whereas how to reduce the number of shared keys and trapdoors is not considered.

III. MULTI USER SINGLE TRAPDOOR KEY AGGREGATE SEARCHABLE ENCRYPTION (MUSTKASE)

The MUSTKASE scheme applies to any cloud storage that supports the searchable group data sharing functionality. Any user may selectively share a group of selected files with a group of selected users, while allowing the latter to perform keyword search over the former.

To support searchable group data sharing the main requirements for efficient key management are two:

(1) A data owner only needs to distribute a single aggregate key (instead of a group of keys) to a user for sharing any number of files.

(2) User only needs to submit a single aggregate trapdoor (instead of a group of trapdoors) to the cloud for performing keyword search over any number of shared files.

A. *MUSTKASE CONSTRUCTION*

Multi User Single Trapdoor Key-aggregate searchable encryption (MUSTKASE) is an enhanced solution, as where data owner needs

to issue a single aggregate key, instead of numerous keys for sharing 'm' documents with multiple user, and user needs to issue a single aggregate trapdoor, instead of multiple trapdoors to the cloud server. The cloud server can utilize this aggregate trapdoor and some public data to carry out keyword search and revisit the result to user. As a result, in MUSTKASE, the delegation of keyword search right can be achieved by sharing the single aggregatekey.

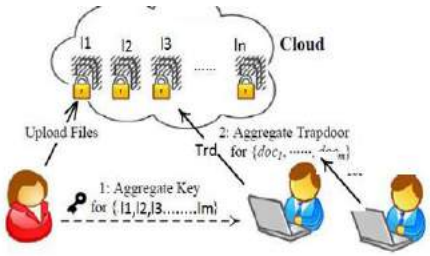

Fig 1: MUSTKASE Architecture

The MUSTKASE construction is composed of several algorithms. Specially, to set up the method, the cloud server would generate public parameters of the system during the **Setup** algorithm, and these public parameters can be reprocess by dissimilar data owners to distribute their files. For each data owner, they should produce a public/master-secret key pair through the **Keygen** algorithm. Keywords of each document can be encrypted through the **Encrypt** algorithm with the exclusive searchable encryption key. In that case, the data owner can apply the master-secret key to produce an aggregate searchable encryption key for a group of selected documents through the **Extract** algorithm. The aggregate key can be spread securely to approve users who need to access those documents. After that, as shown in Fig.1, an certified user can create a keyword trapdoor via the **Trapdoor** algorithm using this aggregate key, and submit the trapdoor to the cloud. After getting the trapdoor, to carry out the keyword search over the particular set of documents, the cloud server will run the **Adjust** algorithm to produce the right trapdoor for each document, and then run the **Test** algorithm to test whether the document contains the keyword.

This construction is summarized as following.

1. Setup(1^{λ}, n): This algorithm is run by the cloud service provider to set up the scheme. On input of a security parameter 1^{λ} and the maximum possible number n of documents which belongs to a data owner, it outputs the public system parameter params.

2. Keygen: This algorithm is run by the data owner to generate a random key pair (pk,msk) where pk is the public key and msk is master secret key.

3. Encrypt(pk, i): This algorithm is run by the data owner to encrypt the i-th document and

generate its keywords' ciphertexts. For each document, this algorithm will create a delta Δi for its searchable encryption key ki. On input of the owner's public key pk and the file index i, this algorithm outputs data cipher text and keyword ciphertexts Ci.

3. Extract(msk, S): This algorithm is run by the data owner to generate an aggregate searchable encryption key for hand over the keyword search right for a certain set of documents to other users. It takes as input the owner's master-secret key msk and a set S which enclose the directory of documents, and then outputs the aggregate key kagg.

4. Trapdoor(kagg, x): 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 kagg and a keyword x, then outputs only one trapdoor Trd.

5. Adjust(params, i, S, Trd): this algorithm is run by cloud server to adjust the aggregate trapdoor to generate the right trapdoor for each different document. It takes as input the system public parameters params, the set S of documents' indices, the index i of target document and the aggregate trapdoor Tr, then outputs each trapdoor Tr_i for the i-th target document in S.

6.Test(Tri, i): This algorithm is run by the cloud server to perform keyword search over an encrypted document. It takes as input the trapdoor Tr_i and the document index i, then outputs true or false to denote whether the document doc_i contains the keyword w.

IV. IMPLEMENTATION AND RESULTS

A. GROUP CREATION AND USER REGISTRATION: The group will be created among the data owner and the users for the purpose of sharing.Only the users who has permission to access the documents will be added in the group. After the registration, user obtains a private key which will be used for group signature generation and file decryption. The security parameter 1^{λ} used for setting up this scheme.

B. DATA UPLOADING & SHARING: Data (files) has been uploaded by the data owner to the cloud. Key is generated for each files using randomised key generation algorithm and it is aggregated. Data files uploading are done using randomly generated keys. For encryption Encrypt(pk, i) where pk is the public key and i^{th} document. For Key generation keygen algorithm is used with the parameters public key pk and master secret key msk

C. KEYWORD SEARCH: The key generation algorithm takes as input the master key msk and a set of attributes S that describe the key. Extract(msk, S) is used to owner to generate an aggregate searchable encryption key kagg. Trapdoor(kagg, x) is used by the the user who

has the aggregate key to perform a search for keyword x and outputs only one trapdoor Trd.

.

D. DATA RETRIEVING: The decryption algorithm takes as input the public parameters pk, a ciphertext C_i, which contains an access policy and a private key k_i for a set S of attributes. If the set S of attributes satisfied the access structure then the algorithm will decrypt the ciphertext. Adjust(params, i, S, Trd) is run by cloud server to adjust the aggregate trapdoor to generate the right trapdoor for each different document. Test(Tr_i, i) takes as input the trapdoor Tr_i and the document index i, then outputs the keyword x.

V. CONCLUSION

In this paper, a new method called MUSTKASE is projected. MUSTKASE can provide an efficient solution to building practical data sharing system based on public cloud storage. In a MUSTKASE scheme, the owner needs to distribute a single key to a user when contributing a lot of documents with the user, and the user needs to submit a single trapdoor when they queries over all documents shared by the same owner. On the other hand, if a user wants to question over documents shared by multiple owners, that user can use the single trapdoor to the cloud.

REFERENCES

[1] Baojiang Cui, Zheli Liu and Lingyu Wang : Key-Aggregate Searchable Encryption for Group Data Sharing via Cloud Storage, IEEE Transactions On Computers, Vol. 65, No.8, August 2016

[2] 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.

[3] X. Liu, Y. Zhang, B. Wang, and J. Yan. "Mona: secure multiowner data sharing for dynamic groups in the cloud", IEEE Transactions on Parallel and Distributed Systems, 2013, 24(6): 1182-1191.

[4] C. Chu, S. Chow,W. Tzeng, et al. "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage", IEEE Transactions on Parallel and Distributed Systems, 2014, 25(2): 468-477.

[5] X. Song, D.Wagner, A. Perrig. "Practical techniques for searches on encrypted data", IEEE Symposium on Security and Privacy, IEEE Press, pp. 44C55, 2000.

[6] R. Curtmola, J. Garay, S. Kamara, R.Ostrovsky. "Searchable symmetric encryption: improved definitions and efficient constructions", In: Proceedings of the 13th ACM conference on Computer and Communications Security, ACM Press, pp. 79-88, 2006.

[7] P.Van,S.Sedghi,JM.Doumen."Computationally efficient searchable symmetric encryption", Secure Data Management, pp.87-100, 2010.

[8] S. Kamara, C. Papamanthou, T. Roeder. "Dynamic searchable symmetric encryption", Proceedings of the 2012 ACM conference on Computer and communications security (CCS), ACM, pp. 965- 976, 2012.

[9] D. Boneh, C. G, R. Ostrovsky, G. Persiano "Public Key Encryption with Keyword Search", EUROCRYPT 2004, pp. 506C522,2004.

[10] Y. Hwang, P. Lee. "Public Key Encryption with Conjunctive Keyword Search and Its Extension to a Multi-user System", In: Pairing-Based Cryptography C Pairing 2007, LNCS, pp. 2-22, 2007.

[11] J. Li, Q. Wang, C. Wang. "Fuzzy keyword search over encrypted data in cloud computing", Proc. IEEE INFOCOM, pp. 1-5, 2010.

[12] C. Bosch, R. Brinkma, P. Hartel. "Conjunctive wildcard search over encrypted data", Secure Data Management. LNCS, pp. 114- 127, 2011.

[13] C. Dong, G. Russello, N. Dulay. "Shared and searchable encrypted data for untrusted servers",Journal of Computer Security