INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

CLOUD DATA STORAGE AND SHARING USING KEY AGGREGATE SEARCHABLE ENCRYPTION

PALLAVI VIJAY NICHAL[1], PROF. P. L. RAMTEKE[2]

1. Student of Master of Engineering (CS&IT), HVPM's college of Engineering and Technology Amravati, India.
2. Associate Professor & HOD (IT), HVPM's College of Engineering and Technology Amravati, India.

Accepted Date: 15/03/2016; Published Date: 01/05/2016

Abstract*:* Data sharing is an important functionality in cloud storage. Cloud storage has emerged as a promising solution for providing securely, efficiently, and on-demand accesses to large amounts of data shared over the Internet. In this paper, we address this practical problem, by proposing the novel concept of key aggregate searchable encryption (KASE) and instantiating the concept through a concrete KASE 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.

Keywords: Searchable Encryption, Data Sharing, Cloud Storage, Data Privacy.

Corresponding Author: MS. PALLAVI VIJAY NICHAL

Access Online On:

www.ijpret.com

How to Cite This Article:

Pallavi Vijay Nichal, IJPRET, 2016; Volume 4 (9): 932-937

PAPER-QR CODE

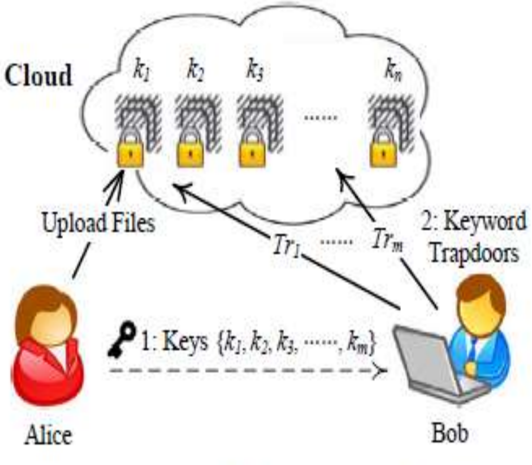
INTRODUCTION

Data sharing structures primarily based on cloud storage have attracted tons attention currently. Specially, Chu et al.[2] keep in mind the way to lessen the number of dispensed data encryption keys. To proportion several files with distinctive encryption keys with the same user, the facts owner will need to distribute all such keys to him/her in a traditional technique which is normally impractical. Aiming at this challenge, a key mixture Encryption (KAE) scheme for information sharing is proposed to generate an aggregate key for the user to decrypt all the documents. To permit a fixed of files encrypted through distinct keys to be decrypted with a single combination key, user may want to encrypt a message no longer best under a public-key, however also underneath the identifier of every record[2]. To address users' concerns over potential data leaks in cloud storage, a common approach is for the data owner to encrypt all the data before uploading them to the cloud, such that later the encrypted data may be retrieved and decrypted by those who have the decryption keys. Such a cloud storage is often called the cryptographic cloud storage [3].

In addition, a large number of trapdoors must be generated by users and submitted to the cloud in order to perform a keyword search over many files. The implied need for secure communication, storage, and computational complexity may render such a system inefficient and impractical. In this paper, we address this challenge by proposing the novel concept of key-aggregate searchable encryption (KASE), and instantiating the concept through a concrete KASE scheme. The proposed KASE scheme applies to any cloud storage that supports the searchable group data sharing functionality, which means 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 twofold. First, 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. Second, the 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. let G and G1 be two cyclic groups of prime order p, and g be a generator of G. Moreover, let doc be the document to be encrypted, k the searchable encryption key, and Tr the trapdoor for keyword search.

I. PROBLEM DEFINITION

Consider a scenario where two employees of a company would like to share some confidential business data using a public cloud storage service (e.g., dropbox or syncplicity). For instance, Alice wants to upload a large collection of financial documents to the cloud storage, which are meant for the directors of different departments to review. Suppose those documents contain highly sensitive information that should only be accessed by authorised users, and Bob is one of the directors and is thus authorized to view documents related to his department. Due to concerns about potential data leakage in the cloud, Alice encrypts these documents with different keys, and generates keyword ciphertexts based on department names, before uploading to the cloud storage. Alice then uploads and shares those documents with the directors using the sharing functionality of the cloud storage. In order for Bob to view the documents related to his department, Alice must delegate to Bob the rights both for keyword search over those documents, and for decryption of documents related to Bob’s department. With a traditional approach, Alice must securely send all the searchable encryption keys to Bob. After receiving these keys, Bob must store them securely, and then he must generate all the keyword trapdoors using these keys in order to perform a keyword search. As shown in Fig.1,

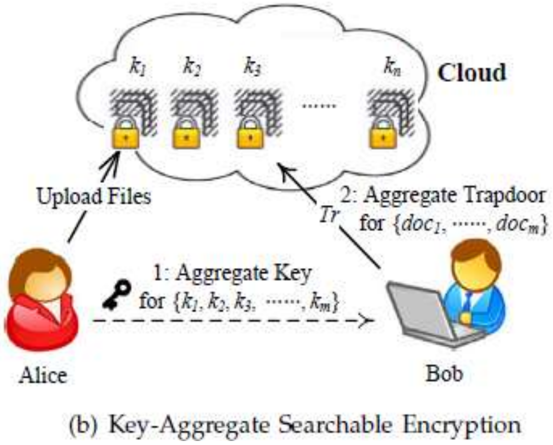


(a) Traditional approach

Fig.1. Keyword search in group data sharing system

In this paper, we propose the novel approach of key-aggregate searchable encryption (KASE) as a better solution, as depicted in Fig.2., in KASE, Alice only needs to distribute a single aggregate key, instead of fkigm i=1 for sharing m documents with Bob, and Bob only needs to submit a single aggregate trapdoor, instead of fTrigm i=1, to the cloud server. The cloud server can use this aggregate trapdoor and some public information to perform keyword search and return the result to Bob. Therefore, in KASE, the delegation of keyword search right can be achieved by sharing the single aggregate key. We note that the delegation of decryption rights can be achieved using the key-aggregate encryption approach recently proposed in [2], but it remains an open problem to delegate the keyword search rights together with the decryption rights, which is the subject topic of this paper. To summarize, the problem of constructing a KASE scheme can be stated as: "To design a key-aggregate searchable encryption scheme under which any subset of the keyword ciphertexts (produced by the SE.Encrypt algorithm to be introduced in Section 5) from any set of documents is searchable (performed by the SE.Test algorithm) with a constant-size trapdoor (produced by SE.Trpdr algorithm) generated by a constantsize aggregate key."

The KASE Framework


k_1
k_2
k_3
......
k_n
Cloud
Upload Files
Tr
2: Aggregate Trapdoor for $\{doc_1, \cdots\cdots, doc_m\}$
1: Aggregate Key for $\{k_1, k_2, k_3, \cdots\cdots, k_m\}$
Alice
Bob

(b) Key-Aggregate Searchable Encryption

Fig.2. Framework of key-aggregate searchable encryption.

Key-aggregate Encryption for Data Sharing

Data sharing structures primarily based on cloud storage have attracted tons attention currently. Specially, Chu et al.[2] keep in mind the way to lessen the number of dispensed data encryption keys. To proportion several files with distinctive encryption keys with the same user, the facts owner will need to distribute all such keys to him/her in a traditional technique which is normally impractical. Aiming at this challenge, a key mixture Encryption (KAE) scheme for information sharing is proposed to generate an aggregate key for the user to decrypt all the documents. To permit a fixed of files encrypted through distinct keys to be decrypted with a single combination key, user may want to encrypt a message no longer best under a public-key, however also underneath the identifier of every record[4].

II. THE PROPOSED SCHEME

The design of our KASE scheme draws its insights from both the multi-key searchable encryption scheme [5] and the key-aggregate data sharing scheme [2]. Specifically, in order to create an aggregate searchable encryption key instead of many independent keys, we adapt the idea presented in [2]. 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, we employ a similar process as in [5]. The cloud server can use this process to produce an adjusted trapdoor for every document.

To further describe this system in details, we describe its main work flows in this section. **System setup**. When an organization submits a request, the cloud will create a database containing above four tables, assign a group ID for this organization and insert a record into table **company**. Moreover, it assigns an administrator account for the manager. Then, the group data sharing system will work under the control of manager. To generate the system parameters params, manager runs the algorithm KASE. **Setup** and updates the field parameters in table **company**. **User registration**. When adding a new member, the manager assigns member ID, membe Name, password and a key pair generated by any public key encryption (PKE) scheme for him, then stores the necessary information into the table **member**. A user's private key should be distributed through a secure channel. **User login**. Like most popular data sharing products (e.g., Dropbox and citrix), our system relies on password verification for authenticating users. To further improve the security, multi-factor authentication or digital signatures may be used when available. **Data uploading**. To upload a document, the owner runs KAE. **Encrypt** to encrypt the data and KASE. **Encrypt** to encrypt the keyword cipher texts, then uploads them to the cloud. The cloud assigns a docID for this document and stores the encrypted data in the path file Path, then inserts a record into the table **docs**. In addition, the owner can encrypt the keys using his/her private key and store them into the table **docs**.

Data sharing. To share a group of documents with a target member, the owner runs KAE. **Extract** and KASE. **Extract** to generate the aggregate keys, and distributes them to this member, then inserts/updates a record in table **shared Docs**. If the shared documents for this member are changed, the owner must re extract the keys and update the field docID Set in table **shared Docs**. **Keyword Search**. To retrieve the documents containing an expected keyword, a member runs KASE. **Trapdoor** to generate the keyword trapdoor for documents shared by each owner, then submits each trapdoor and the related owner's identity Owner ID to the cloud. After receiving the request, for each trapdoor, the cloud will run KASE. **Adjust** the trapdoor for each document in the docID Set and run KASE. **Test** to perform keyword search. Then, the cloud will return the encrypted documents which contains the expected keyword to the member. **Data retrieving**. After receiving the encrypted document, the member will run KAE. **Decrypt** to decrypt the document using the aggregate key distributed by the document's owner.

III. CONCLUSION:

From the work flows above, we can see that the number of keys of a member is linear in the number of users who share documents with him, and the number of trapdoors in a keyword search is the same. Compared to traditional data sharing solutions, this system has better efficiency.

IV. REFERENCES:

1. S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing",vProc. IEEE INFOCOM, pp. 534-542, 2010.
2. 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.
3. 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.
4. F. Zhao, T. Nishide, K. Sakurai. Multi-User Keyword Search Scheme for Secure Data Sharing with Fine-Grained Access Control. Information Security and Cryptology, LNCS, pp. 406-418, 2012.
5. R. A. Popa, N. Zeldovich. "Multi-key searchable encryption". Cryptology ePrint Archive, Report 2013/508, 2013.
6. D. Boneh, C. Gentry and B. Waters. "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys", CRYPTO'05, pp. 258C275, 2005.
7. 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.
8. X. Song, D.Wagner, A. Perrig. "Practical techniques for searches on encrypted data", IEEE Symposium on Security and Privacy, IEEE Press, pp. 44C55, 2000.
9. P. Van s. Sedghi, JM . Doumen. "Computationally efficient searchable symmetric encryption", Secure Data Management, pp. 87-100, 2010.