IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-3, Issue-9 E-ISSN: 2347-2693

Secured Group Data Bestow with Key-Agglomerative Searchable Encryption via Cloud Storage

Santhiya.C[1]*, Vanishree K.A.[2] and M.K. Chandrasekaran Ph.D.[3]

[1*,2,3]*Dept. Of Computer Science and Engineering, Anna University, India,*

www.ijcseonline.org

Received: Sep/02/2015 Revised: Sep/11/2015 Accepted: Sep /27/2015 Accepted: Sep/30/2015

Abstract-The capability of distributing selected encrypted data with different users by means of public cloud storage may greatly alleviate the protection concerns over inadvertent data leaks in the cloud. Efficient management of encryption keys solves this difficulty. The preferred flexibility of distributing any group of selected documents with any group of users hassle different encryption keys to be used for different credentials. However, this also implies the requirement of securely sharing to users a large amount of keys for both encryption and search, and those users will have to securely stock up the received encrypted keys, and submit an equally large amount of keyword trapdoors to the cloud in order to carry out the exploration over the shared data. The obscure need for secure communication, storage, and complexity noticeably explains that the approach is not appliable. In this paper, a novel concept called, key aggregate Searchable encryption (KASE) is estimated to resolve this matter-of-fact problem and instantiating the notion through a concrete KASE scheme, in which a data holder only needs to share a single key to a user for distributing a large amount of documents, and the user only needs to tender a single trapdoor to the cloud for querying the user shared documents. But by using single key for a group, it is easily misused by the group members. If moved to multiple-keys, information is accesses by Brute-force attack. Hence it should be enhanced in a way that reduced number of keys should be used. The security analysis and concert evaluation both confirm that our projected schemes are provably secure and practically efficient.

Keywords - *Searchable Group Data Sharing, Public Key Encryption, Trapdoor, Cryptographic Cloud System*

I. INTRODUCTION

Now-a-days millions of data are being uploaded to cloud such as photos, videos and confidential documents. Notably photos and videos are shared with friends through social network applications such as facebook, twitter, etc. In addition to this, business users are using cloud storage because of cloud's benefits such as lower cost, greater dexterity and better resource utilization.

The main advantage of cloud system is that cloud users can retrieve their files or documents from anywhere in the world without the need of their personal system. But the main issue is the data leakage. Such data leaks are done by malicious attackers or else by misbehaving cloud machinist. Hence the authorized users are in need to encrypt their confidential documents by using their public key before uploading them. Then the users who are in need of that document have to decrypt them with their private key. This storage system is known as cryptographic cloud system.

In case, if the owner of a document wants to made availability of only a particular files in a document, it is possible with searchable encryption (SE) technique. By using SE, data owner can encrypt potential keywords and upload them to the cloud with encrypted data. Then the data to be retrieved with a matching keyword, the user will send corresponding keyword trapdoor for performing search over encrypted data. This could be made possible through a novel concept of *Key-agglomerative searchable encryption (KASE).* KASE is applicable only to the cloud storage that supports the well known *searchable group data sharing* functionality. In a searchable group data sharing functionality, any user can share a group of files to only a selected group of users. There are two requirements for performing this.

1. *Data owner should only need to distribute a single aggregate key, instead of group of keys to a user.*
2. *User should submit a single aggregate trapdoor instead of group of trapdoors for searching a keyword.*

There are two ways to share the encrypted data:

1. Alice encrypts data with single unique secret key and shares that secret key directly with the Bob.

2. Alice can encrypt data with divergent keys and send Bob corresponding keys to Bob via secure channel.

In first approach, unwanted data also get depiction to the Bob, which is inadequate. In second approach, no. of keys is as many as no. of shared files. Hence, the discernment of KASE is to share the group of data by searchable encryption scheme by using a single unique key

alone. Also it is imperative that the owner of a data should delegate rights to all the users who have to access the document.

To make this attainable the KASE framework should be constructed containing seven algorithms namely, setup, keygen (key generation), encryption, key extraction, trapdoor generation, trapdoor adjustment and trapdoor testing. Finally, the KASE should be evaluated to meet its recital requirements.

II. SEARCHABLE ENCRYPTION

`Broadcast Encryption (BE) Scheme should be considered before considering searchable encryption (SE) scheme. Here, a broadcaster encrypts message only for particular users in a group to snoop on a broadcast channel. The only difference between a BE and SE is that BE is described by using a table containing only three algorithms BE = (Setup, Encrypt, Decrypt). In searchable encryption technique, client sends to server a searchable indication, which contains an encrypted matching document of a key. Then the server can decrypt the document via key. Cryptography technique can be applied in a two major ways- one is symmetric key and other is asymmetric key encryption. In first, same keys are used for encryption and decryption. By contrast, in second different keys are used, public key for encryption and private key for decryption. Two categories of SE are 1.Searchable Symmetric Encryption (SSE) and Public key Encryption with Keyword Search (PEKS). Both the categories are described using the single table SE= (Setup, Trapdoor, Encrypt, Test). The algorithm comparison of searchable encryption scheme is as follows.

S.NO:	Algorithm comparison			
	Name of Algorithm	***Run by***	***Input***	Output
1.	Setup	Owner of the scheme	1^{λ}	Necessary keys
2.	Encrypt	Data owner	m, necessary keys and data encryption keys.	Keyword cipher text (C_m)
3.	Trapdoor	User	K	T_r
4.	Test	Cloud server	Tr, C_m	C_m contains keyword or not

Table: 1 Comparison of algorithms in Searchable Encryption (SE)

Where,

1^{λ}- Input parameter
m- Message
C_m- Cipher Text
K- Keyword
T_r- Trapdoor

III. RELATED WORK

This section explains some existing solutions of KASE scheme, which are to be considered before knowing about KASE.

2.1. Multi-user searchable encryption:

In the framework of cloud storage, a keyword should be searched under the multi-tenancy setting, which is contrast to the concept of SE scheme and PEKS scheme. In Multi-tenancy setting, data owner would share his/her documents to the group of users and the user who has rights can use their trapdoor for exploration of the keyword. This is known as "Multi User Searchable Encryption" (MUSE).This setting is often used to for sharing resources cost efficiently and securely. Multi-tenancy is not same as multi-user. It is a key attribute of both public and private clouds. The main drawback in MUSE is that the following two things are not considered.

- How to control which user should access which documents?
- How to reduce the number of shared keys and trapdoors?

KASE resolves these two drawbacks and makes the MUSE technique more efficient.

2.2 Multi-key Searchable Encryption:

The Multi-key Searchable Encryption (MKSE) technique allows a user to provide a single keyword trapdoor to the server, but still allows the server to search for a fastidious trapdoor's keyword in documents which is encrypted with different keys. MKSE is similar to the KASE scheme. But, the only difference between MKSE and KASE is that, MKSE is to ensure that the cloud server can perform keyword search with one unique trapdoor over different documents owing to a user.

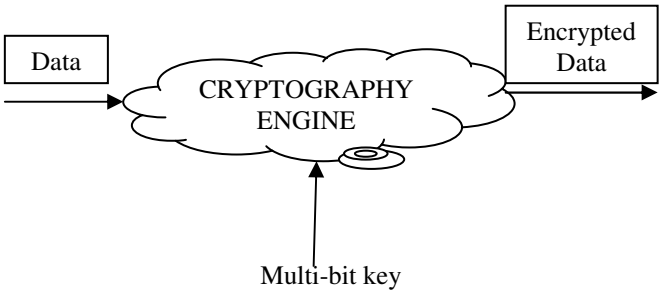


Fig: 1 Multi-key Searchable Encryption

IV. KEY AGGLOMERATIVE SEARCHABLE ENCRYPTION (KASE) FRAMEWORK

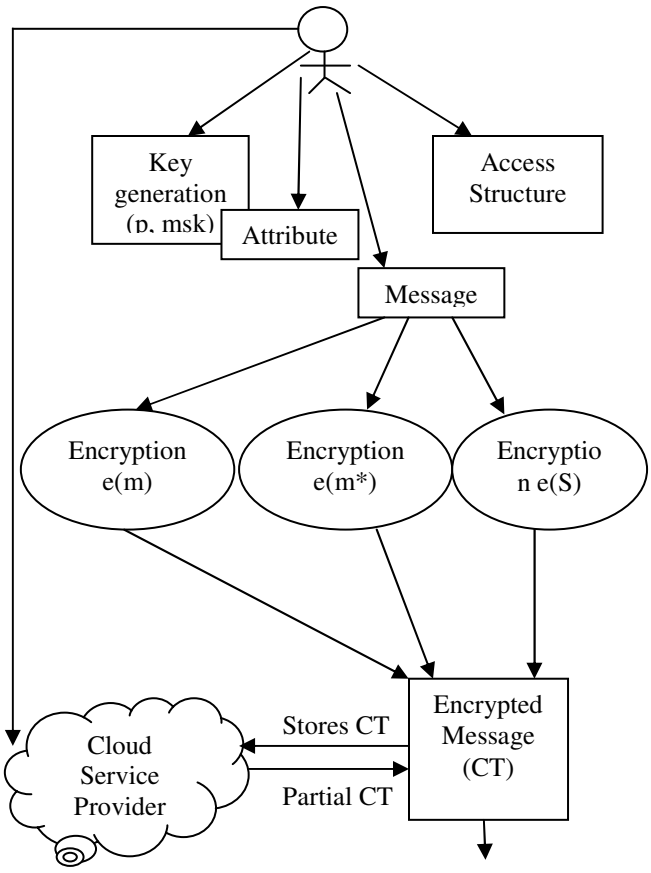
The KASE framework is used to,

1. describes general problem in KASE
2. defines framework for KASE
3. provides requirements for designing a KASE

A. Problem Statement

Let us consider, there are two employees in a company namely; employee1 and employee2.Employee1 send his confidential financial documents to employee2 via public cloud storage service. The confidential documents should only be accessed by directors of different departments. Hence employee1 is in need to encrypt the document with department name. If a third person, namely Employee3 who also have to access the document for searching essential documents should be delegated by employee1. Employee1 should entrust rights for 1) keyword search and 2) decryption of documents.

A diagrammatic representation of encryption and decryption amid a user and retriever is as follows, where the user should encrypt the message with his public key and the retriever should decrypt the unrestrained message using his own private key. Partially encrypted message i.e., the partial cipher text should be hoarded in a cloud overhaul contributor.



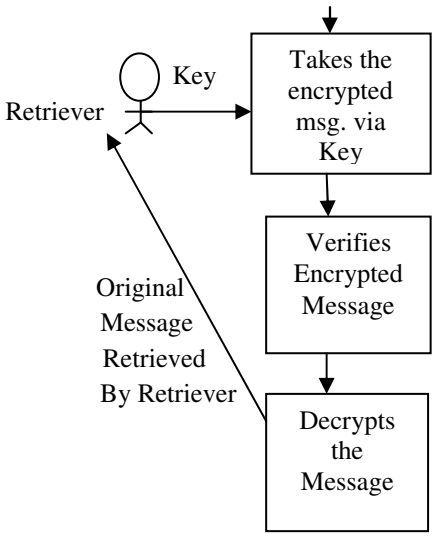


Fig: 2 Encryption and Decryption of a message

Here, user procreate a key and also provides an attribute and an access structure for a communicative message. After the communicative message is encrypted, the decoded plain text is reworded into a secured cipher text. This cipher text should again be converted to a plain text by the retriever for getting the inventive plaintext again, which is known as decryption. For that the retriever should takes the secured inventive encrypted message and use his private key to decrypt the message.

The decrypted message should be verified such that both the plaintext and cipher text have to be identical. Finally, the decrypted message will be send to the retriever.

B. Requirements for designing KASE scheme

A legitimate KASE scheme should satisfy some functional and security requirements. There are three functional (purposeful) requirements and two security (safe-keeping) requirements.

1) Functional Requirements:

The functional requirements for getting a legitimate KASE scheme are compactness, searchability and delegation.

1.1) Compactness:

It ensures that the volume of the K_{agg} should needs to be independent of the number of files to be shared i.e., for a set of keys $\{K_i\}_{i \in S}$, it requires K_{agg} <- Extract (msk, S). The main confront here is that the set of keys i.e., multi keys should be agglomerated into a single key.

1.2) Searchability:

The innermost part in designing a KASE framework is the Searchability requirement. It generates trapdoor for a keyword to investigate for the encrypted

S. NO:	Algorithm comparison			
	Name of Algorithm	***Run by***	***Input***	Output
1.	Setup	Cloud service provider	1^{λ} , n	params
2.	Keygen	Data owner	Pk, msk	Master secret key pair
3.	Encrypt	Data owner	Pk, i	Data cipher text & C_i
4.	Extract	Data owner	Msk, S	k_{agg}
5.	Trapdoor	User who has k_{agg} to search	K_{agg} & keyword w	Aggregate Trapdoor (T_r)
6.	Adjust	Cloud server	Params, S, i, aggregate trapdoor (T_r)	Trapdoor T_{ri} for i^{th} target document in S
7.	Test	Cloud server	Tri, i	True (or) False

documents. One advantage here is that, it is possible to defend search capability by reducing the amount of keys. Here for each document, it requires (T_r=**Trapdoor** (k_{agg}, w) & Tr_i <- **Adjust** (params, i, S, Tr)), then **Test** (Tr_i, i) = true.

1.3) Delegation:

The main concept behind KASE scheme is to entrust users who are also having some part of role in a document such as to access a file for their own purpose. But the inputs of Adjust algorithm must not be public to all the users. This is manifestly the second confront in a KASE scheme.

2) Security Requirements:

The security requirements for getting a legitimate KASE scheme are controlled probing and Query Privacy.

2.1) Controlled Probing:

It means that without the data holder's entrust, it is impossible for the attackers or hackers for searching an illogical key. So they can't perform keyword search which are not appropriate to $k_{agg.}$ It is also not possible for the cruel attackers to generate aggregate piercing encryption keys from the known keys.

2.2) Query privacy:

It means that the attackers cannot determine the keyword used in a query, apart from the information that can be acquired via observation and the information derived from it. That is, the user may ask an untrusted cloud server to search for a perceptive word without enlightening the word to the server.

C. Comparision of KASE scheme:

The KASE scheme contains seven algorithms, in which similarity is shown with Searchable Encryption scheme. The only difference is in KASE is that, supplementary to setup, trapdoor, encrypt and test in searchable encryption scheme, keygen, extract and adjust algorithms are secondary in KASE.

The seven algorithms in a KASE scheme is compared below which contains information such as the algorithm name, algorithm is run by whom and what are its input and output with parameter description.

Table: 2 Comparison of algorithms in key Aggregate Searchable Encryption (KASE)

Where,

1^{λ}-security parameters,
N- Number of documents which belongs to user,
pk, msk- random key pair
pk - Public key
msk - Master Secret Key
i- File index
S- Set S, which contains indices of documents
W- Keyword
Params- parameters
K_{agg}-Aggregate key

V. CONCRETE GROUP DATA SHARING SYSTEM

When constructing a practical matter-of-fact group data sharing System, it is important to reduce the amount of keys belonging to a user. Hence, a new scheme is provided to build such a system based on the KASE and KAE schemes with the same communal parameters. We regard as a group data sharing system without using any private cloud, but instead based on extensively available communal public cloud services, such as Dropbox or citrix. Based on such a consideration, we presuppose a group executive (e.g., the HR director of an Organization) with an authorized account to act in the role of the one who will be accountable for management of the system including perpetuate the public system parameters stored in the cloud.

A) Table Definition

For constructing a legitimate group data sharing system, it is necessary to consider the following things.

*1) Table **group**<groupID, groupName, parameters> is used to store the system parameters.*

*2) Table **member**<memberID, membeName, password, DOB, publicKey> is to store members' information including their public key.*

*3) Table **docs**<documentID, documentName, OwnerID, EncryptionKey, SEKey, filePath> is to store the uploaded document of an owner with identityi.e., ownerID.*

*4) Table **sharedDocs**<SID, memberID, OwnerID, documentIDSet> is to store the documents of a member with the only known identity memberID shared by the proprietor with identity OwnerID. Field documentIDSet is for all the indices of documents.*

VI. WORK FLOWS

The work flow section of a KASE scheme contains five steps. They are,

1) *System setup*
2) *User registration*
3) *User Login*
4) *Data Uploading*
5) *Data Sharing*
6) *Keyword Search*
7) *Data Retrieving*

1) System setup:

When an organization submits a request, the cloud will create a database containing above four tables, assign a groupID for this organization and insert a record into the database named, Table Company. Moreover, it assigns an administrator account for the manager. Then, the group data sharing system will toil under the control of manager. To generate the system parameters params, manager runs the algorithm KASE. **Setup** and updates the all the field parameters in Table Company.

2) User registration:

When adding a new member, the manager are in need to assign memberID, membeName, password and a key pair generated by any public key encryption (PKE) proposal and stores the obligatory information into the table ***member***. A user's private key should be dispersed through a secure channel.

3) User login*:*

For authenticating users, like most popular data sharing products (e.g., Dropbox and citrix), our system relies on password verification. To further improve the security, multi-factor endorsement or digital signatures may be used when available.

4) Data uploading:

To upload a document, the owner runs KAE.**Encrypt** the data and then encrypt the keyword ciphertexts, then uploads them to the cloud. The cloud assigns a documentID for this document and stores the encrypted data in the trail filePath, then inserts a record into the table **docs**. For further security, the owner can encrypt the keys using his/her private key and store them into the table **docs**.

5) Data sharing:

To bestow a group of documents with a target member, the owner runs KAE **Extract** and KASE.**Extract** to spawn the aggregate keys and distributes them to this member, then inserts/updates a record in table **sharedDocs**. If the shared documents are changed, the owner must re-extract the keys and update the field docIDSet in table **sharedDocs**.

6) Keyword Search:

To retrieve the documents containing an expected keyword, a member must runs KASE **Trapdoor** to generate the keyword trapdoor for credentials or documents to be shared by each owner, then indu;lge each trapdoor and the related owner's identity OwnerID to the cloud. After this, for each trapdoor, the cloud will run KASE **Adjust** trapdoor for each document in the documentIDSet and run KASE. Also run the KASE **Test** to perform keyword search. Then, the cloud will replace the encrypted credentials which contains the expected keyword to the member.

7) Data retrieving:

After receiving the encrypted document, the member will run KASE **Decrypt** to decrypt the document using the aggregate key scattered by the document's owner.

VI. EFFICIENCY OF KASE TABLE DEFINITION

The term efficiency in KASE means that the size of keyword cipher text, trapdoor and aggregate keys are constant. In addition the following two should be notable.

- The set S contains indices of shared documents with aggregate key of linear size. This won't affect the data sharing system, because contents of S are stored in cloud server.
- System is not affected since the public system parameters is O(n) size.

VI. ADVANTAGES OF KASE SCHEME

1) Only the users who are having the aggregate key can perform a successful keyword search.
2) Even when the cloud server colludes with a malevolent authorized user, It is not possible to perform a keyword search over any document not in the compass of the user's aggregate key.
3) An attacker is unable to bring out the new aggregate key for any new set of documents from the recognized aggregate key
4) An attacker is unable to determine a keyword in a query from the submitted trapdoor.

5) An attacker is incapable of understanding a keyword in a document from the stored keyword cipher texts and the allied public information.

CONCLUSION

In scrutiny of the matter-of-fact problem of privacy preserving data sharing system on public cloud storage should be triumph over by distributing only a single unique key, the KASE scheme is used. From the outcome, it is lucid that legitimate KASE provides a proficient solution for building data sharing system under public cloud storage.

In the KASE scheme, the owner should distribute only a single key to the user for the rationale of distributing documents with patron. Similarly user should submit only a single trapdoor to access a document. In accumulation to this, the data owner should entrust some additional rights to other users. However for accessing multiple documents there occurs a annoyance i.e., for giving out a data with multiple owners, user should generate multiple trapdoors. To steadfastness this is a future work. Also in KASE, federated blur in cloud is not accessed unswervingly. This is also a future work in KASE.

REFERENCES

[1] S Yu, C Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Com-puting", Proc. IEEE INFOCOM, pp. **534-542, 2010.**

[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] 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**.

[4] L B Oliveira, D F Aranha, E Morais, et al. "Tinytate: Computing the tate pairing in resource-constrained sensor nodes", IEEE Sixth IEEE International Symposium on Network Computing and Applications, pp. **318-323, 2007.**

[5] M. Li, W. Lou, K. Ren. "Data security and privacy in wireless body area networks", Wireless Communications, IEEE, 17(1): **51- 58, 2010.**

[6] B Wang, B Li, and H Li, "Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud", Proc. 10th Int'l Conf. Applied Cryptography and Network Security, pp. **507-525, 2012**

[7] R A Popa ,N Zeldovich. "Multi-key searchable encryption". Cryptology ePrint Archive, Report **2013/508, 2013**.

[8] 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.**

[9] D H Phan, D Pointcheval, S F Shahandashti, et al. "Adaptive CCA broadcast encryption with constant-size secret keys and ciphertexts", International journal of information security, 12(4): **251-265, 2013.**

[10] J Li, K Kim. "Hidden attribute-based signatures without anonymity revocation", Information Sciences, 180(9): **1681-1689,** Elsevier, **2010.**

[11] J W Li, J Li, X F Chen, et al. "Efficient Keyword Search over Encrypted Data with Fine-Grained Access Control in Hybrid Cloud", In: Network and System Security 2012, LNCS, pp. **490- 502, 2012.**

[12] 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.**