EKASE: Enhanced KeyAggregate Searchable Encryption for Multi-owner Data Sharing via Cloud

S. Brindha [1], M. Raghini [2], R. Birundha [3], V. R. Hemalatha [4]
[1] Assistant Professor,[2] Assistant Professor(Sr.Gr), [3,4]B.E-Final Year
[1,2,3,4] Department Computer Science and Engineering, K.L.N College of Engineering
Sivagangai District, Tamilnadu, India

***Abstract* - The technique to share encrypted data with users through public cloud have security threats such as data leaks in the cloud. The practical problem of privacy-preserving in data sharing system based on public cloud storage requires a data owners to distribute a large number of user-keys to enable them to access the documents. Another issue is user needs to generate multiple trapdoors to access data shared by multiple owners . Efficient management of keys used in such encryption keys is the major challenge. This issue is addressed by the novel concept "Enhanced Key-aggregate searchable encryption (EKASE)", in which the user only needs to submit a single trapdoor for retrieving the documents shared by multiple owners audited by trusted authority. The security analysis and performance evaluation both confirm that the proposed scheme for data, shared by multi-owners accessed through single trapdoor are secured and practically efficient.**

Keywords—Trapdooor, searchable encryption, public cloud storage.

I. INTRODUCTION

Cloud storage is turning widespread in these days. Cloud is one of the easier method for sharing a huge amount of data through internet. Cloud system enable data sharing capabilities which can be provide abundant of benefits to the user. The benefit of data sharing increase productivity time and cost in a cloud system is much less compared to having a manual exchange. Cloud computing is recognized as an alternative to traditional technology due to its intrinsic resource sharing and low maintenance characteristics. To address user's concern over potential data leakage in cloud storage, a common approach is the data owner who encrypts all the data before uploading them to the cloud. Then the encrypted data may be retrieved and decrypted by those who have the decryption keys, this technique is termed as cryptographic storage in cloud [5].

Unfortunately, to design an efficient and secure data sharing scheme for groups in the cloud is not that much easier due to the following challenging issues. User needs to submit multiple keys [4], to perform a keyword search over documents of a single owner encrypted using a different key through trapdoor generation. However this problem is overcome by aggregate key technique [1]. But user needs to submit multiple trap doors to perform keyword search over documents submitted by multiple users. A new method is required for enabling a user to enable keyword search over documents shared by multiple owners by using single trapdoor.

In this paper, by using the novel concept of key-aggregate searchable encryption, we introduce an enhanced methodology "Enhanced Key Aggregate searchable Encryption (E-KASE) framework", based on the available key aggregate searchable encryption [1]. The proposed E-KASE scheme applies to any cloud storage that supports the searchable group data sharing functionality for multiple owners. Which means any owner may selectively share a group of selected files within a group of users who are selected, while allowing the latter to perform keyword search over the shared documents [7] through a single trapdoor. To support searchable group data sharing the main requirements for efficient key management are of three steps.

1) Multiple data owner needs to generate a single aggregate key (instead of group of keys) and send it to trusted authority located at the cloud, for sharing any number of files. For that, we define a general framework of enhanced key-aggregate searchable encryption composed of seven polynomial algorithms for security parameter system setup, key generation, encryption, extraction, generating trapdoor, adjusting trapdoor, testing trapdoor.

2) The trusted authority which interconnects the data owner and user needs to generate super aggregate key by combining all the aggregate keys received from multiple owners. Next, the trusted authority distributes super aggregate key (instead of group of aggregate keys) to the users to whom owners are willing to share their documents, and registered with the trusted authority.

3) The user only needs to generate a single trapdoor (instead of a group of trapdoors) to the cloud server for performing keyword search over documents shared by multiple owners, to retrieve the documents having the matching keywords.

The outline of the paper is as follows II. Related Work, III.EKASE Framework, IV.EKASE Algorithm, V. Security Analysis, VI. Performance Evaluation, VII. Conclusion and Future work.

II. RELATED WORK

In this section, we review some basic assumptions and cryptology which will be needed later in this paper. In the rest our discussions, 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* be the searchable encryption key, and *Tr* be the trapdoor for keyword search.

A. Broadcast encryption

In a broadcast encryption scheme [1], a broadcaster encrypts a message for *S* subset of users who are listening in the broadcast channel. Any user in *S* can use their private key to decrypt the broadcast. A broadcast encryption (*BE*) scheme can be described as *BE= (Setup, Encrypt, Decrypt).*

1) Setup (1^{λ}, n) algorithm:

- Inputs security parameters 1^{λ} and number of receivers n.
- Output private keys d^1, …., d^n and public keys pk.

2) Encrypt (pk, s) algorithm:

This algorithm is run by the broadcaster to encrypt a message for subset of users.

- It takes as input a public key *pk* and a subset of users *S* subset of {1,…,n}.
- Outputs a pair (*Hdr, k*), where *Hdr* is called the header and *K* is a message encryption key which is encapsulated in *Hdr*. We will often refer to *Hdr* as the broadcast cipher text. For a concrete message. It will be encrypted by *K* and broadcasted to the users in *S*.

3) Decrypt (pk, S, I, d_i, Hdr) algorithm:

This algorithm decrypts the received messages.

- Inputs a public key *pka* subset of users *S* subset of *{1,….,n}*. A user id *i<={1,…,n}*, the private key d_i for user *i* and a header *Hdr*.
- Outputs the message encryption key *K*. The *K* will be used to decrypt the received messages.

Searchable encryption

Searchable encryption scheme [1] falls into two categories, i.e., searchable symmetric encryption (SSE) and public key encryption with keyword search (PEKS). Both SSE and PEKS can be described *SE= (Setup, Encrypt, Trapdoor, Test).*

1) Setup(1^{λ}):

This algorithm is run by the owner to setup the scheme. It takes a security parameter 1^{λ} and outputs the necessary keys.

2) Encrypt(k, m):

This algorithm is run by the owner to encrypt the data and generate its keyword cipher texts.

- *S* input the data *m*, owner's necessary keys including key *k* and data encryption key.
- Outputs data cipher text and keyword cipher text C_m.

3) Trapdoor(k, w):

This algorithm is run by a user to generate a trapdoor *Tr* for a keyword *w* using a key *k.*

4) Test(Tr, C_m):

This algorithm is run by the cloud server to perform a keyword search over encrypted data.

- Input trapdoor *Tr* and the keyword cipher texts C_m.
- Outputs whether C_m contains the specified keyword.

B. Key aggregate searchable encryption

In this scheme, owners encrypt each of the documents using different keys. Sharing multiple keys with users makes it complex. So, the owner combines all the keys using master key, generates and shares an aggregate key to the user through mail. User submits this aggregate key along with the keyword through trapdoor to perform keyword search over different documents submitted by the same owner [2].

III. EKASE FRAMEWORK

In this section, we first describe the general problem in key aggregate searchable encryption framework and then define a generic framework for Enhanced Key Aggregate Searchable Encryption.

Consider a scenario where group of employees and multi-owners of the company would like to share some confidential business data using a public cloud storage service (e.g., drop-box). For instance, Alice and Jack (Owners) want to upload a large collection of financial documents to the cloud storage. Documents contain highly sensitive information that should only be accessed by authorized users. Bob (user) is one of the directors and is thus authorized to view and download the documents related to his departments. For providing more security, Alice and jack encrypt their documents with different keys and generate keyword cipher texts based on department names before uploading to the cloud storage [4]. Alice and Jack share files using the sharing functionality of the cloud storage. If Bob wants to retrieve the documents related to his department, Alice and Jack must delegate rights to Bob to perform keyword search and decryption over those documents.

In the traditional KASE (Fig.1), each owner shares an aggregate key (generated by combining all the keys used to encrypt different documents belonging to an individual owner) to the user for enabling access to all documents shared by them. A single trapdoor is generated which uses aggregate key and keyword. Now keyword search is performed in the cloud and the matched files are given to the users. If a user wants to access the documents shared by different owners, they must generate multiple trapdoors.

To overcome this problem, in this paper we propose an enhanced scheme known as Enhanced Key Aggregate Searchable Encryption (EKASE). In EKASE (Fig.2), the trusted authority is introduced, which combines multiple aggregate keys into a super aggregate key and it is shared by the trusted authority. At user side, single trapdoor *(Tr)* is

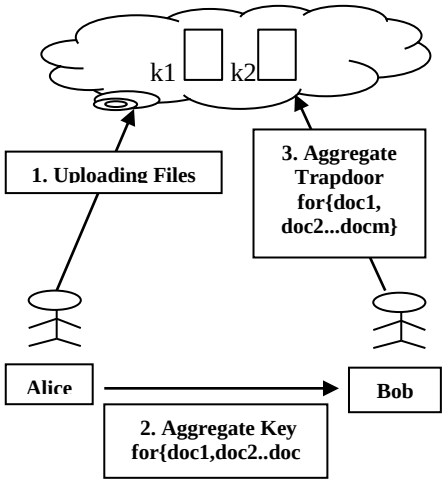

Fig. 1: Key Aggregate Searchable Encryption

generated which uses super aggregate key and keyword to retrieve the documents shared by multiple owners. By using this scheme, the generation of multiple trapdoors could be resolved.

Then using the submitted trapdoor *(Tr)*, the trusted authority generates the right trapdoor for owners and further the cloud server uses that trapdoor for each documents shared by an individual owner (Fig.3)

IV. EKASE ALGORITHM

1) System setup:

This algorithm is run by the cloud service provider to setup the scheme. On input of a security parameter 1^{λ}, the maximum possible number n of documents and number of users *N*, which belongs to a data owner. It outputs the public system parameter *params.*

Finally cloud service publishes the system parameters *params= {b, PubK, H}*, where $PubK= (g,g_1,\dots g_n,g_{n+2},\dots g_{2n})$.

2) Key generation:

Data owner uses this algorithm to generate key pair (public key and master key) and it is also used by trusted authority to generate super master key.

- It picks a random $\gamma \epsilon Z_P$ and outputs the key pair, $p_k=v=g^{\lambda}$.
- Trusted authority generates super master key S_{msk}.

3) Encryption:

Each data owner uses this algorithm to encrypt data and generate its keyword cipher texts when uploading the i^{th} document. For every document this algorithm will create an encryption key k_i which is generated by using the owner's public key and file index i. this algorithm generates the data and keyword cipher texts. The algorithm is,

- Input the file index $i\epsilon\{1,2,\dots n\}$.
- Randomly pick a $t \epsilon Z_P$, as the searchable encryption key k_i of this document.
- Generates a delta Δ_i for k_i by computing:

$$c_1= g^t,\ c_2= (v.g_i)^t$$

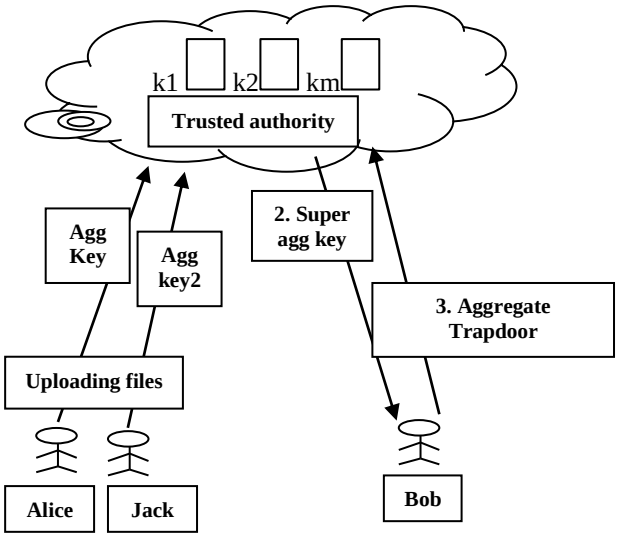

Fig. 2: Enhanced Key Aggregate Searchable Encryption

- For a keyword w, outputs the cipher texts c_w as:

$$c_w=e(g,(H(w)^t/e(g_1,g_2)^t)$$

c_1, c_2 are public and that can be stored in the cloud server, *w* is the keyword.

4) Extraction (msk, Omsk, O, S):

Data owner uses this algorithm to generate an aggregate searchable encryption key.

- For any subset $S\epsilon \{1,2,\dots n\}$ which contains the indices of documents. This algorithm takes input the master secret key *msk.*
- Outputs the super aggregate key k_{agg} by computing:

$$K_{agg}=\prod_{j\epsilon s} g^{\gamma}{}_{n+1-j}$$

The trusted authority receives the aggregate key from owners and generates super aggregate key using owner index and super master key.

- For any subset $O\epsilon \{1,2,\dots n\}$ which contains the indices of owners. This algorithm takes the trusted authority's super master secret key *Omsk.*
- Outputs the super aggregate key S_{agg}. To delegate a keyword search right to the user, the trusted authority will send S_{agg} to the user.

5) Trapdoor generation (k_{agg}, w):

The user uses this algorithm to generate the trapdoor to perform keyword search. For all documents which are relevant to the super aggregate key S_{agg}. It generates only one trapdoor *Tr* for the keyword *w* by computing:

$$Tr= S_{agg}. H(w)$$

6) Adjust (params, i, S, O, O_i, Tr):

The cloud server uses this algorithm to produce the right trapdoor for multiple owners and document.

- Trusted authority inputs the system public parameters O of owner indices, index O_i of the target owner and the aggregate trapdoor *Tr* to adjust right trapdoor for each owner and then outputs each trapdoor Tr_{oi} for i^{th} target owner in O.
- After identifying the owner, the cloud server adjusts the Tr_{oi} trapdoor to generate right trapdoor for each document using the system parameter, set *S* of

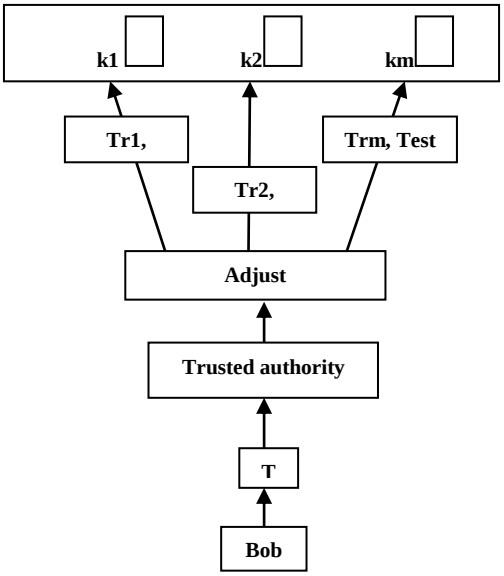

Fig. 3: Single Trapdoor Generation

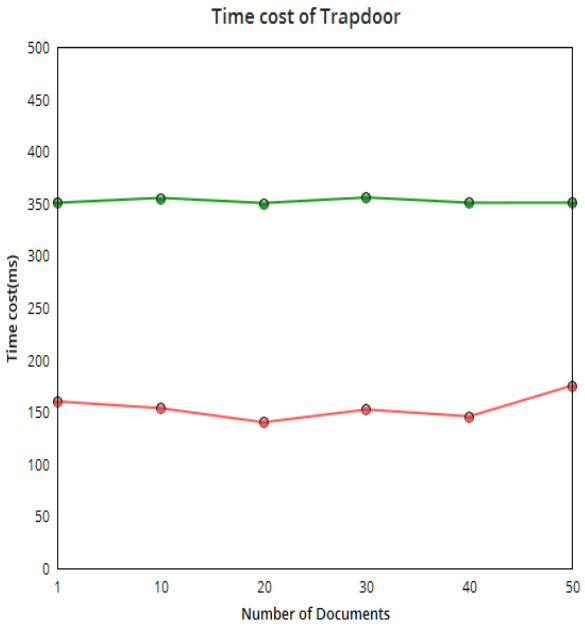

Fig. 4: Execution Time for Trapdoor Generation Algorithm

document's indices, index *i* of the target document and trapdoor Tr_{oi} then outputs trapdoor for each i^{th} document as Tr_i.

7) Test:

This algorithm is run by cloud server for performing keyword search over encrypted document. The input is trapdoor Tr_i, index of the document and output is true or false to indicate whether the document doc_i has a keyword *w*.

V. Security Analysis

To study the security of the proposed scheme, we assume that cloud server will only provide the services confirming to rules, according to already defined techniques and it may recover secret information based on its knowledge [3], [7].

Based on the above case, we will prove that the security of the proposed scheme in context of searching and query privacy.

- The proposed scheme supports controlled searching, because only user who has the aggregate key can perform a successful keyword search, even when the cloud server encounters malicious authorized user. They will not able to perform keyword search over any documents. Because the malicious user won't have knowledge about *S*, the document set.
- The proposed technique can achieve query privacy. Because the attacker won't be able to determine a keyword in the query from the submitted trapdoor.
- An attacker will not be able to determine a keyword in a document from the stored keyword cipher text and the related public information.

VI. Performance Evaluation

Considering that the practical data sharing system based on cloud storage, the user can retrieve data by any possible device and the mobile devices are widely used. The performance is highly dependent on the basic cryptographic operations especially in the pairing computation, we study whether the cryptographic operations based on pairing computation can be efficiently executed using both computers and mobile devices.

- The execution time of system setup grows linear depending on the number of documents and owners.
- The encryption time is linear to number of keywords.
- The execution time of Extract is linear to number of keywords.
- The execution time of Trapdoor is constant according to the number of documents (Fig. 4), while the number of documents is 50, the time cost for trap door generation using the proposed scheme is 150ms. Since cloud user can retrieve data by any possible device like the mobile and computer device, the execution time for trapdoor generation is tested and data are shown using the green and red lines respectively.
- The execution time of Adjust phase grows linear to the execution time of Test is linear to the number of keyword cipher texts.

Adjust and Test algorithm both confirms that the execution time is linear to the number of documents. To improve efficiency of Adjust and Test, parallel computing and distributed computing techniques are applied. By increasing the number of threads, the execution time of Test algorithm is reduced (Fig. 5). In Fig. 5 the green line indicated the time of test and the red lines indicates the time of thread creation. When the number of threads is 100, the execution time is 400ms to finish the keyword search over 10,000 keyword cipher texts. Multiple thread technique helps to improve performance, but selection of threads is an important case.

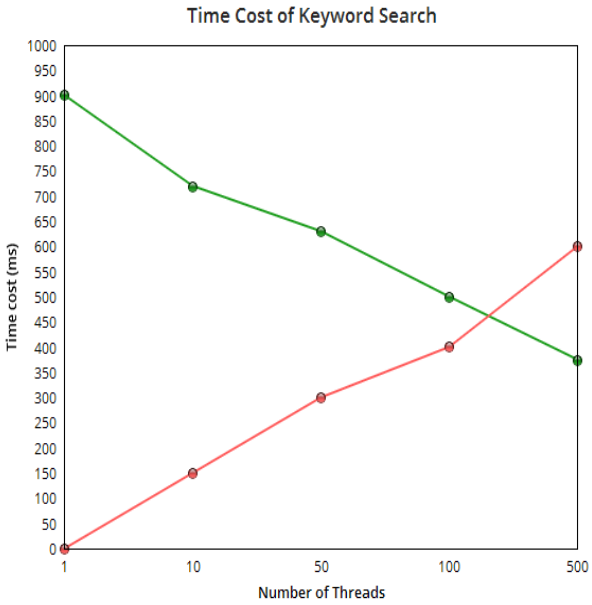

Fig. 4: Execution Time of Test Algorithm

VII. CONCLUSION AND FUTURE WORK

Taking into account the problem of privacy preserving data sharing in public cloud requiring multiple owners to distributed many aggregate key to users for enabling them to access the documents shared by multiple owners. We propose the concept of Enhanced Key Aggregate Searchable Encryption and constructed EKASE scheme. In EKASE scheme, the trusted authority generates super aggregate key and user submits only single trapdoor to access data shared by multiple owners. Both analysis and evaluation results confirm that our work provide good solution for data sharing system through public cloud.

The future work is towards key optimization since the proposed technique requires multiple keys to generate aggregate key and super aggregate key, leading to overhead in key usage so optimization helps in improving the efficiency of the EKASE technique.

REFERENCES

[1] Baojiang cui, Zheli liu, and lingyu Wang, "Key-Aggregate Searchabe Encryption (KASE) for group Data Sharing via Cloud Storage," IEEE Transactions on Computers, vol.65, No.8 August 2016.

[2] C. K. Chu, S.Chow, W. G. Tzeng, J. Y. Zhou, and R. Deng, "Key-aggregate cryptosystem for scalable data sharing in cloud storage," IEEE Transactions Parallel Distrib, Syst., vol 25, no. 2, pp. 468-477, Feb. 2014.

[3] D. Boneh, C. G, R. Ostrovsky, and G. Persiano, "Public Key encryption with keyword serach." In Proc, Int. Conf. Theory Appl. Cryptograph Techn., 2004, pp. 506-522.

[4] R. A. Popa and N. Zeldovich, "Multi-key searchable encryption," Cryptol ePrint Archive, Rep. 2013/508, 2013.

[5] S Yu, C Wang, K. Ren, and W Lou, "Achieving secure scalable, and fine-grained data access control in cloud computing," in Proc IEEE Conf Comput Commun, 2010, pp 534-542.

[6] X Liu, Y. Zhang, B. Wang, and J. Ya, "Mona: Secure multi-owner data sharing for dynamic groups in the cloud." IEEE Trans Parallel Distrib., vol 24, no. 6, pp. 1182-1191, Jun. 2013.

[7] Y. Hwang and P.Lee, " Public key encryption with conjuctive keyword search and its extension to a multi-user system." In Proc.