International Journal of Computer Applications (0975 - 8887)
Volume 139 - No.2, April 2016

Review on Key-Aggregate Searchable Encryption (KASE) for Group Data Sharing

R. Rakesh
P G Scholar
Department of Computer Science & Engineering
College of Engineering, Perumon(CUSAT), Kerala, India

Anoop S.
Assistant Professor in IT
Department of Information Technology
College of Engineering, Perumon(CUSAT), Kerala, India

ABSTRACT

An efficient cryptographic approach for data sharing where data is shared among a group of users as Data sharing is an important functionality in cloud storage. How to securely and efficiently share a collection of data related to any subject areas with others in cloud storage. Development of new novel concept of Key-Aggregate Searchable Encryption (KASE). This concept is implemented through development of a concrete key-aggregate searchable encryption framework scheme. This scheme is described as where a data owner only needs to generate and distribute a single aggregate key to a data user for sharing a large number of documents and on the other side user only needs to submit a single aggregate trapdoor to the cloud server, so that he/she can query over the shared documents by the help of generated single aggregate trapdoor. This proposed scheme is perfectly more secure and practically efficient. It is an effective method which is considered as best solution to build a practical data sharing system based on public cloud storage. A detailed review of various methods used for data access controls and encryption is presented and a brief comparison among the discussed methods is given.

General Terms

Cloud storage, data privacy, data sharing, searchable encryption

Keywords

Cloud Storage Provider, Outsourcing, Attribute based Encryption, Key-Aggregate Cryptosystem

1. INTRODUCTION

Cloud storage is a solution for sharing and accessing large amounts of data, which is shared for various users by means of internet. Today, a number of users are mainly sharing a large number of various kinds of documents, which are considered to be under various categories like photos, videos and documents via various social networking based applications on daily basis. There are huge benefits of using cloud storage like lower cost, greater agility and better resource utilization has add more attraction from plenty number of business users toward using the cloud storage.

Cloud computing which is built on parallel, distributed computing, utility computing and service-oriented architecture. Generally, speaking about cloud storages, we all are enjoying the comfort of sharing all kinds of data. But all users are more bothered about the data leaks which usually happen in the cloud storage. Such type of data leaks occur due to reason like an untrusted cloud provider and by hackers who decrypt the files using various types of software. A common approach usually used is to encrypt all the types of data available with him/her. Which are to be uploaded to the cloud by the data owner. The encrypted data obtained shall be retrieved and then performing decryption by persons who have right set of access keys. This type of cloud storage is known as Cryptographic cloud storage.

However, there are two challenging tasks:

(1) How can a user perform searching over the documents shared?
(2) How to retrieve only the data which can be retrieved by a given keywords?

Above stated two challenges can be solved by the implementation of searchable encryption (SE) scheme. In this scheme, the data owner encrypts all the keywords which were used to encrypt the data and both the encrypted keyword and encrypted data are uploaded to the cloud together. To obtain the original data back, the user will need to send a keyword trapdoor which will be used to match a data with a keyword. If a match is obtained than the document belonging to a data user can be retrieved, otherwise the keyword based searching continues, until all the keyword trapdoor have been tested on the document collection available on the cloud server.

By combining both the cryptographic cloud storage along with the searchable encryption scheme, the essential basic security requirements can be attained. Also, management of keys is a serious problem. How to efficiently manage the encryption keys is generally neglected in case of survey based on literature. First requirement of a data owner is to share the selected set of data with types of different users. For example, sharing a photo and videos is a common fashion now with the help social network applications like Facebook, WhatsApp etc. Generally, users share various types of documents through cloud storage social networking application like Google drive, Dropbox, Citrix etc. Also Cloud service providers examples like Amazons EC2 and S3 [2], Google App Engine [3], and Microsoft Azure [4], these provide us all the resources required as per our needs. We can pay them as we use these services. Usually uploaded data is encrypted with a different encryption key. The

number of key generated will be proportional to the number of document files to be encrypted. Also, how to send these set of different keys among the various kind of users. So, has to perform the searching and decryption over the set of documents. These keys must be send to a user using a secure communication channel, also how can a user store and manage these keys in their devices like mobile phones, PCs, laptops, removable devices etc.

Speaking about the traditional method of data sharing through various cloud storage providers, in Fig.1 it consists of two types of users: Data owner and Data user. Data owner is uploading n numbers of documents to cloud server which are shared with the data user. Generally, each document is encrypted with a separate key, i.e. if n documents are to be encrypted than n keys are required to perform encryption using them. The key produced is send to the data user via a secure communication channel by the data owner. Than after performing all these actions, data user can perform searching over the shared documents by generating keyword trapdoors. If a match is obtained, the cloud server returns the original files which were shared by the data owner to corresponding requested data user.

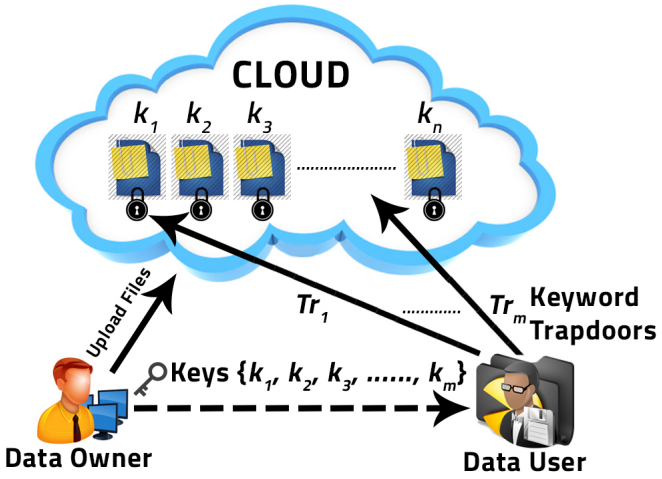

Fig. 1. Traditional Approach of data sharing

Various techniques have been proposed for data sharing via cloud storage, their efficiency is to be increased by means of development of new concepts and schemes. This paper is organized as follows: Section 2 illustrates some of the methodologies used for data sharing through cloud storage. Section 3 illustrating the solution to problems which are stated in the Section 2. Section 4 describes the comparison of various existing methods. Section 5 consists of experimental results and analysis which presents the performance evaluation of KASE scheme. Section 6 concludes the review of KASE scheme.

2. LITERATURE SURVEY

Cloud storage has grown to become popular and is adopted by many individuals and organizations. The widely adoption of cloud storage raised several security concerns about the outsourced data, such as confidentiality, integrity and access control of the data. Both academic and industrial world are making efforts to maintain the security of the outsourced data.

2.1 Access controls

Works have been done as to migrate and adapt the mature traditional authorization management to cloud computing. Besides that, a series of new access control schemes and solutions have been researched and devised for cloud environment based on the general access control solutions.

2.1.1 Identity-Based Encryption. Of all the access control architectures, Attribute-Based Encryption (ABE) schemes are the most popular ones due to its scalability and security. Unlike Access Control List(ACL) only defines which entities have the access right, ABE schemes encrypt the data under the access policy which only ensure the eligible entities to do decryption. A distinguished work Fuzzy Identity-Based Encryption(IBE) was introduced by Sahai and Waters in 2005. In Fuzzy IBE scheme, a private key for an identity set ω , can be used to decrypt a cipher-text encrypted with an slightly different identity set ω '. Fuzzy IBE realizes error tolerance by setting the threshold value of root node smaller than the size of identity set.

2.1.2 Keypolicy Attribute-Based Encryption. Based on Fuzzy IBE, Goyal et al. present Keypolicy-Attribute Based Encryption (KP-ABE) in which ciphertexts are labeled with sets of attributes and private keys are associated with access structures that control which ciphertexts a user can decrypt. In this scheme when a user made a secret request, the trusted authority determined which combination of attributes must appear in the ciphertext for the user to decrypt.

2.1.3 Ciphertext Policy Attribute Based Encryption. Bethencourt et al. introduced a complementary scheme to KP-ABE, called Ciphertext Policy-Attribute Based Encryption (CP-ABE). In this ciphertext policy attribute based encryption system, a user's private key is associated with a set of attributes and an encrypted cipher text will specify an access policy over attributes. A user will be able to decrypt if and only if his attributes satisfy the cipher text's policy. One of the most challenging issues in data sharing systems is the enforcement of access policies and the support of policies updates. CP-ABE is becoming a promising cryptographic solution to this issue. It enables data owners to define their own access policies over user attributes and enforce the policies on the data to be distributed.

2.2 Literature Survey on Related Works

2.2.1 Multi-User Searchable Encryption (MUSE). In description of cloud storage, a most common scenario is keyword search which is performed by various users and it is known as multiuser setting. In this MUSE, the data owner shares a document with a number of authorized users and each authorized user who has the right set of access rights can perform searching over the document using trapdoor mechanism. Most recent developed works included in [9], [16-18]. Speaking about [22] which mainly focus on MUSE, in this implementation is done by single key combined with various access controls.

In the construction of MUSE scheme [9] and [22], which is developed for first of all share the searchable encryption key which is used for document encryption to all the users. The users who have the keys can access these documents, also by using broadcast encryption. It achieves the access control for all the documents shared. In the description [16-21], by applying the attribute based encryption, it achieves more fine access control which is based on keyword searching.

But, in case of MUSE there are two major problems which is not considered are:

(1) How to check whether a user has the right to access the document?
(2) How to decrease the number of trapdoor generated and total number of shared keys?

2.2.2 Multi-Key Searchable Encryption (MKSE). Considering the multi user based applications, the ratio of number of trapdoors is directly equivalent to the number of searched documents. MKSE was developed and presented in the year 2013. This algorithm is explained as a data user to give a single trapdoor which consists of a single keyword to the cloud server. But on other hand, the cloud server gives provision to search over the keyword trapdoor by using different keys. In Fig.2, it consists of a Multi-Key Searchable Encryption (MKSE) which shows that a data user is submitting his/her generated trapdoor(Tr) to cloud server and the cloud server performing the adjust and test algorithm on the document collection.

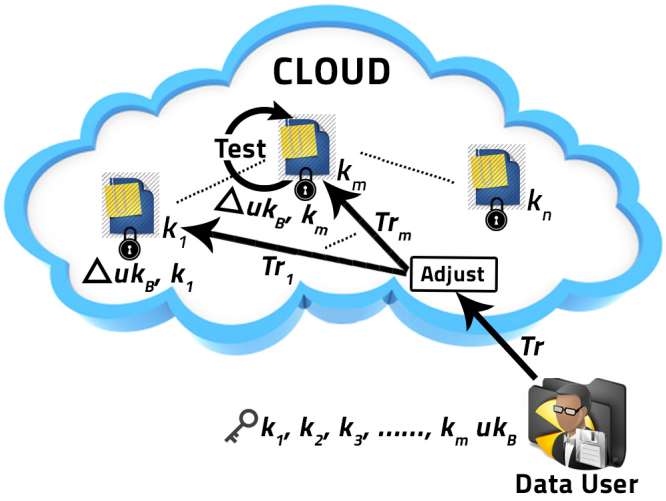

Fig. 2. Multi-Key Searchable Encryption

The main goals of both i.e. KASE and MKSE are completely different ideas. Goal of MKSE: when keyword search is performed by the cloud server with only one trapdoor on different types of user owned documents, that of KASE: by mainly providing the generated single aggregate key to data users in a group sharing based system. Speaking more about the MKSE, data user can store public data information which is known as Delta on cloud server. This public information is relevant to data user key and used encryption key. Data user can perform searching for a word on all the documents, for doing this he/she needs the data user key to calculate the trapdoor for the word and directly submit this generated trapdoor value to cloud server.

Cloud server uses this information to convert received keyword trapdoor on the key available with the data user. This process is known as adjust. By doing so, cloud server can perform traditional searching by means of single-key with the newly generated trapdoor. In MKSE the adjust process is an approach to perform searching on the group of documents shared by means of single trapdoor. This adjust process can't be applied directly to the development of KASE scheme.

2.2.3 Key-Aggregate Encryption (KAE). Recently more attention has been created around the cloud storage, which is based on data sharing systems [5]-[7]. By considering the paper [7] which points out that how to decrease the number of keys used for data encryption. In traditional approach, all used encryption keys must be distributed among the concerned authorized users. This challenge is solved by KAE, where it generates an aggregate key which will be used by the user to decrypt all the documents shared with him/her. Concept of KAE is to obtain the original document by decrypting with a single aggregate key, which was encrypted with different keys. To perform this data owner, not only needs the public key but also the identity of each document. This is concept is adapted from the broadcast encryption scheme [29].

In development of KAE scheme, the data owner is designed as broadcaster. Broadcaster will be having the public key and master secret key. Data user is designed as the receivers, who are listening to this secure broadcast channel. Generally, speaking about public information which consists of various relevant information like data owner's master secret key and encryption key. Here, data encryption is performed using the symmetric encryption in broadcast encryption. But the key aggregation and data decryption is done by the algorithms like BE.Encrypt and BE.Decrypt respectively. By using scheme [7], which delegates all the decryption rights to the data users. The problem with KAE, we can't perform searching over the encrypted documents. So, the development of new scheme is needed, which will provide us to perform keyword based searching, trapdoor generation and also more complex procedure to obtain keyword matching in more efficient way. So, KASE scheme was designed and developed by the researchers in the field of research and development.

3. KEY-AGGREGATE SEARCHABLE ENCRYPTION (KASE)

Development of KASE scheme ideas is adapted from papers like key-aggregate cryptosystem scheme [7] for scalable data sharing and Multi-key searchable encryption scheme [31]. This was done to generate a single aggregate encryption key in replacement of many numbers of individual independent keys for each documents uploaded by the data owner. Defining this scheme each key which is used for searching is connected with a particular index of uploaded document. Creation of aggregate key is done by using the data owner's master-secret key with product of his/her public keys used for encryption. Keyword based searching is performed by generation of aggregate trapdoor mechanism. This is implemented by adjusting process [31]. Than cloud server can use single adjusted aggregated trapdoor which was created for each set of document.

3.0.4 KASE Scheme Description. KASE Framework was described in the above section, this KASE scheme consists of seven algorithms:

(1) **Setup** : This algorithm is run by cloud server to setup all system parameters. Generate a bilinear mapping based group sharing system, set the maximum possible number of documents available with the data owner. Two operations are computed which are random generator calculation and selecting a one-way hash function. Cloud server broadcast the generated system parameter and public key.
(2) **Keygen** : This algorithm is run by data owner to generate his/her key pair which will be used for document encryption by the Encrypt algorithm. In this stage, we have public key and master secret key along with the generated key pair.

(3) **Encrypt** : This algorithm is run by data owner to perform data encryption and also generate corresponding ciphertexts for all the documents which will be uploaded. For the creating the keyword ciphertexts, it takes the document file index, randomly picks a searchable encryption key for each document and generates a delta information. It will produce a ciphertext for a keyword, this generated ciphertexts are stored under cloud server.

(4) **Extract** : This algorithm is run by data owner and generating an aggregate searchable encryption key and this key is send to all authorized users via a secure communication channel. This algorithm takes input as master secret key and generates an aggregate key as output. Data owner than send this aggregate key to data users, so that they can perform keyword searching over the shared documents.

(5) **Trapdoor** : This algorithm is run by data user and performs keyword searching by generating trapdoor. In the case of searching for matching relevant documents by use of single aggregate searchable key. Only one single aggregate trapdoor is generated for a single keyword which is used for searching. Than data user sends this generate single trapdoor and subset of matched documents.

(6) **Adjust** : This algorithm is run by cloud server and creating right set of trapdoor. It accepts input as system publicly available parameters, all documents index in the set and also single aggregate trapdoor. It performs adjusting process on the single aggregate trapdoor and output a new right single trapdoor. This produced trapdoor will be used for next Test algorithm for performing keyword search over the shared collection of documents.

(7) **Test** : This algorithm is run by the cloud server. Cloud server does a series of keyword searching by using the input, which is adjusted trapdoor and creates the delta information which is relevant to subset by using searchable encryption key. Output produced will be binary, i.e. true or false values after performing various computations.

Key-aggregate searchable encryption (KASE) method of data sharing, in Fig.3 it consists of two types of users: Data owner and Data user. Data owner is uploading n numbers of documents to cloud server which are shared with the data user. Generally, here documents is encrypted by a key pair, this obtained key pair is changed into single aggregate key by using data owner public key and master secret key. The single aggregate key produced is send to the data user via a secure communication channel. Data user can perform searching over the shared documents by generating single aggregate trapdoor. For each searched word, it can generate an aggregate trapdoor. If a match is obtained, the shared documents are unlocked and returned to respective authorized data user.

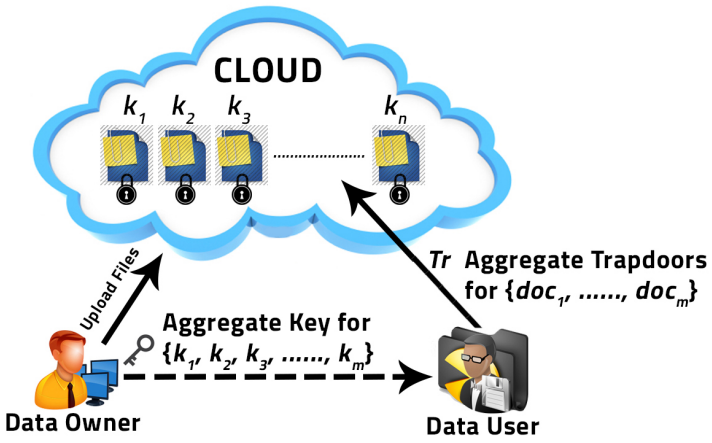

Fig. 3. Key-Aggregate Searchable Encryption

Framework of Key-aggregate searchable encryption (KASE), in Fig.4 it consists of a data owner generates a single aggregate key which was created by using data owner public key and master secret key for encrypting the shared documents. This single aggregate key produced is send to the data user through a secure communication channel. Then, data user can perform searching over the shared documents by generating single aggregate trapdoor, submitted this trapdoor to the cloud server.

Cloud server performs the adjusting algorithm/process by using the aggregate trapdoor over the collection of documents. Then, test algorithm is performed to ensure that the respective requester has the right to access them. If a match occurs, than cloud server will return all the shared documents to the respective data user.

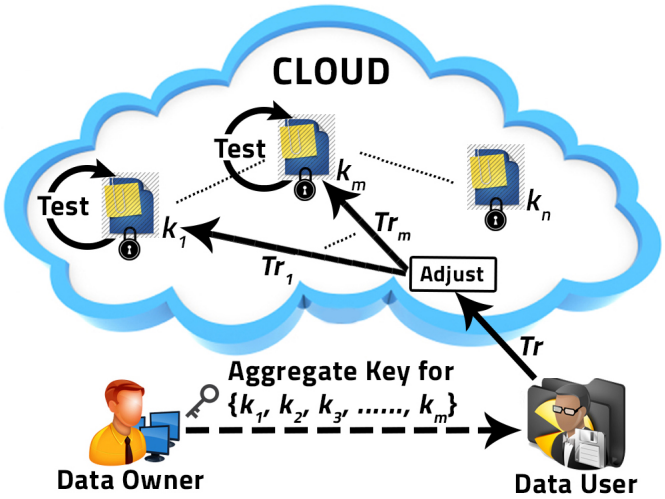

Fig. 4. Framework of key-aggregate searchable encryption

4. COMPARISON OF VARIOUS METHODS

Considering the case of data stored under the cloud storage, the serious issues like confidentiality, integrity and access control should be checked, whether they are meet or not. There are plenty of access control schemes like Attribute Based Encryption (ABE), Key Policy-Attribute Based Encryption (KP-ABE), Ciphertext Policy-Attribute Based Encryption (CP-ABE) and Key-Aggregate Searchable Encryption (KASE). Let us compare and analyze these access control schemes in detail. Comparison of all access control schemes is as follows in Table 1.

Sahai and Waters who introduced the Attribute Based Encryption (ABE) scheme, it is a public key based encryption which is giving more security and better access control. The main specialty of this scheme, it provides the encryption and decryption by means of their user attributes. Generation of ciphertext and secret keys depends on their attributes. If the attribute of secret key is different from the attribute of ciphertext, than decryption process is not possible. Considering the value of threshold as t. If atleast 't' numbers are matching, then performing decryption. Advantage of ABE is it has very complex access control and no need of list of users in this process, only required is the access policy. Disadvantage of ABE is that the data owner needs to use all the available set of users public keys, so as to perform the data encryption.

Table 1. COMPARISON OF ABE, KP-ABE, CP-ABE and KASE

Parameters	**ABE**	**KP-ABE**	**CP-ABE**	**KASE**
Efficiency	Average	Low	Average	Average
Data confidentiality	Present	Present	Present	Present
User accountability	Absent	Absent	Present	Present
Fine grained access control	Low	Low	Average	Average
Computational overhead	High	Low	Average	Average
Collusion resistant	Average	Good	Good	Excellent
Revoke users	Absent	Present	Present	Present

Speaking about the Key Policy-Attribute Based Encryption (KP-ABE), it is another type of ABE. It can perform one to many communications. In this KP-ABE scheme, each private key will be linked with an access tree structure. This type of access tree structure will explain the type of ciphertext which can be decrypted by using the key. Here, the ciphertext is represented with the set of attributes and the key is represented with the access structure, this scheme is called as KP-ABE. This scheme gives a fine grained access control and it can also provide better flexibility than ABE. Problem with KP-ABE is that who can decrypt the encrypted data decision can't be taken by the data owner.

The Ciphertext Policy Attribute Based Encryption (CP-ABE) runs in reverse order of KP-ABE. This will eliminate the main disadvantage of KP-ABE. In CP-ABE, the data owner will decide the policy about who can perform decryption on the encrypted data. Disadvantage of CP-ABE is how to manage the attributes of data users and their respective access policy.

The Key-Aggregate Searchable Encryption (KASE), it is a public key encryption scheme which is adapted from key-aggregate cryptosystem scheme [7] and Multi-key searchable encryption scheme [31]. Advantage is that in place of sharing the documents, data owner send the single aggregate key. By using this key, he/she can access all the documents will is meant for him/her. It eliminates the main disadvantage of KP-ABE, CP-ABE. In KASE, the data owner will generate a single aggregate key and transmit it to the user. Data user can submit generated single aggregate trapdoors to the cloud server. Cloud server than perform adjust and test algorithms to retrieve the relevant documents shared with him/her.

5. EXPERIMENTAL RESULTS AND ANALYSIS

5.1 PERFORMANCE EVALUATION

Considering the studies of various cryptographic operations based on pairing computation. Which can be efficiently executed and be tested on both computers(Intel(R) Core(TM)i5-3337U CPU @ 1.80GHZ with OS as Windows7) and mobile devices(Samsung G3502U phone) is shown as under in Table-2.

Table 2. Pairing based computation execution times

Tested on	Pairing	pow(in $\mathcal{G}$)	pow(in $\mathcal{G}_1$)	pow(in Z_p)
Samsung G3502U	485	243	74	0.8
Computer	10.2	13.3	1.7	0.05

Implementation of this system is done, by means of two libraries: jpbc (for mobile phones) and pbc library (for computer). In case of mobile devices, it takes about 5 seconds for pairing computations. But the sensor nodes and Personal Digital Assistant (PDA) requires only 1.5 and 0.5 seconds respectively. The above depicts the average time required by mobile device and computer for performing pairing based computations. Computers have faster average time for pairing as compared to mobile devices.

5.2 KASE ALGORITHM EVALUATION

Considering all the algorithms (Setup, Keygen, Encrypt, Extract, Trapdoor, Adjust, Test) which were present in KASE scheme and this scheme is evaluated on both mobile devices and computers.

(1) **KASE Setup:** Generally setup algorithm requires a linear execution time against the maximum number of documents which were belonging to a particular data owner. When the maximum number of documents reaches a value of 20000, the KASE Setup algorithms requires 259 seconds (computers).

(2) **KASE Encrypt:** Execution time of this is also linear against the number of keywords generated. Considering the case when the number of keywords reaches a value of 10000, the KASE Encrypt algorithms require 206 seconds in computers, whereas in mobile devices it takes 10018 seconds. By above values, two conclusions can be made: not to use mobile devices for uploading the documents associated with large number of keywords, keyword based searching can be executed more quickly in computers with the help of pairing based computation.

(3) **KASE Extract:** Execution time against the number of shared documents is also linear. When the number of keywords reaches a value of 10000, the KASE Extract algorithms require 132 seconds in computers, whereas in mobile devices it takes 2430 seconds. Considering the above values, it is not suggested to use mobile devices for this stage. Since, the KASE Extract runs along with the KASE Encrypt algorithm.

(4) **KASE Trapdoor:** Execution time is a constant value for both the mobile devices and computers. Considering the values such as 0.01 seconds in computers, whereas in mobile devices it takes 0.25 seconds. Considering the above values, keyword searching can be done more efficiently in both mobile devices and computers. Also comparing with other available schemes,

KASE scheme is having substantial improvements in trapdoor generation.

(5) **KASE Adjust:** It also provides a linear relation, when plotted execution time against the number of documents available to perform adjusting operation. It can be improved in practical applications more efficiently.

(6) **KASE Test:** Execution time cost against the number of keyword ciphertexts is also linear. Considering the execution of KASE Test algorithm is twice the execution of pairing based computations. When the number of keyword ciphertexts grows to a value of 20000, computers takes 467 seconds for execution.

5.3 GROUP DATA SHARING SYSTEM BASED EVALUATION

Speaking about the group data sharing system where performance directly depends on the KASE algorithms. To improve the existing system, the caching based improved technique need to be used to perform more efficient way of keyword searching. Processing of KASE algorithm: when an aggregate single trapdoor is received, the cloud server executes the KASE.Adjust and KASE.Test keyword searching can be finished.

Considering the time evaluation cost of Adjust algorithm is linear when plotted against the number of documents. In order to avoid the existing system problem such as the repeated number of calculation and improving the performance, the solution that a cloud server can provide is to do some cache computation of the results obtained. Since, the input and calculation processing are same for all set of users. This operation will eliminate the time used for calculation. Next consider the case when a user queries the documents collection for the second set of time, KASE.Adjust can run much faster because of available pre-calculated result.

KASE.Test execution time is a linear structured graph when plotted against the number of ciphertexts generated. To enhance and increase the efficiency, techniques like parallel and distributed computing, multi-thread, hadoop may be used in various scenarios whenever required. In our existing system case, multi-thread techniques are used to perform all the experiments. Next is perform the performance testing by setting the number of keyword ciphertexts to 10000. Execution time of KASE.Test will reduce when the number of threads increases.

Considering the number grows to a value of 200, KASE.Test requires only 1 second to completely performing the keyword based searching over the 10000 keyword ciphertexts. When the number of threads increases in large numbers, existing system will take more time to generate these threads. Considering next scenario when the number reaches a value of 1000, the time required to generate these threads will be equal to 80 milliseconds. So, this multi-thread technique improves the existing system performance to next levels. In case of deployment in practical applications, the number of required threads value should be selected with more care and precision. So, has to obtain the best results.

6. CONCLUSION

In this review paper, practical problems of sharing data among a set of users is considered, without data leaks which usually occurs in the cloud storage. Normal method performed is to share a large number of keys to all authorized data users from data owner through a secure communication channel, which gives the authorized user to access the relevant set of documents shared to him/her. Development of new concept involving the key-aggregate searchable encryption (KASE) and also constructing a KASE scheme. Results based on various comparison and analysis confirm that KASE work can give a better and more efficient solution for building a more secure data sharing system based on public cloud storage available on internet. Description of KASE scheme, the data owner generates a single aggregate key which will be used for encryption process and send this key to the entire authorized user. On the other end, data user creates and query through generated single aggregate trapdoor, this trapdoor produced is used to query over collection of documents shared by the same data owner. Comparison of various methodologies is done and performed pairing computation analysis on system and mobile phone. However, future work of this is concerned over the data shared under multiple owners and how to decrease the number of trapdoor generation.

7. REFERENCES

[1] Cloud-Storage, http://www.thetop10bestonlinebackup.com/cloud-storage.

[2] Amazon Web Services (AWS), http://aws.amazon.com.

[3] Google App Engine, http://code.google.com/appengine/.

[4] Microsoft Azure, http://www.microsoft.com/azure/.

[5] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing”, Proc. IEEE INFOCOM, pp. 534-542, 2010.

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

[7] 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.

[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] 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.

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

[11] 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.

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

[13] 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.

[14] J. Li, Q. Wang, C. Wang. “Fuzzy keyword search over encrypt-ed data in cloud computing”, Proc. IEEE INFOCOM, pp. 1-5, 2010.

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

[16] C. Dong, G. Russello, N. Dulay. "Shared and searchable encrypted data for untrusted servers", Journal of Computer Security, pp. 367-397, 2011.

[17] 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.

[18] 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.

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

[20] X.F. Chen, J. Li, X.Y. Huang, J.W. Li, Y. Xiang. "Secure Outsourced Attribute-based Signatures", IEEE Trans. on Parallel and Distributed Systems, DOI.ieeecomputersociety.org/10.1109/TPDS.2013.180, 2013.

[21] 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.

[22] 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.

[23] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing", Proc. IEEE INFOCOM, pp. 525-533, 2010.

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

[25] D. Boneh, C. Gentry, B. Waters. "Collusion resistant broadcast encryption with short ciphertexts and private keys", Advances in CryptologyCCRYPTO 2005, pp. 258-275, 2005.

[26] 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.

[27] D. Boneh, B. Lynn, H. Shacham. "Short signatures from the Weil pairing", Advances in Cryptology ASIACRYPT 2001, pp. 514-532, 2001

[28] 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.

[29] D. Boneh, C. Gentry and B. Waters. "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys", CRYPTO05, pp. 258C275, 2005.

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

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

[32] PBC library: The pairing-based cryptography library. http://crypto.stanford.edu/pbc/.

[33] K. Ren, C. Wang, Q. Wang et al., "Security challenges for the public cloud", IEEE Internet Computing, volume. 16, no. 1, pp. 6973, 2012.

[34] S. Ruj, A. Nayak, and I. Stojmenovic, "Dacc: Distributed access control in clouds", in Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on. IEEE, 2011, pp.9198.

[35] M. Chase, "Multi-authority attribute based encryption", in Theory of Cryptography. Springer, 2007, pp. 515534.

[36] T Parameswaran, S Vanitha, K S Arvind, "An Efficient Sharing of Personal Health Records Using DABE in Secure Cloud Environment" International Journal of Advanced Research in Computer Engineering and Technology (IJARCET) Volume 2, Issue 3, March 2013