Open access Journal International Journal of Emerging Trends in Science and Technology
Impact Factor: 2.838 DOI: http://dx.doi.org/10.18535/ijetst/v3i03.08

The Efficient key Management Scheme for Data Sharing in Cloud Server

Authors
R.Latha MCA. M.E[1], N.Murugan[2], S.Suresh[3]
[1]Asst Prof, Veltech hightech Dr.Rangarajan Dr.Sakunthala Engineering College, Chennai, India
[2]Dept of MCA, Veltech hightech Dr.Rangarajan Dr.Sakunthala Engineering College, Chennai, India
Email: *murugan09051994@gmail.com*
[3]Dept of MCA, Veltech hightech Dr.Rangarajan Dr.Sakunthala Engineering College, Chennai, India
Email: *sureshsivan4121@gmail.com*

Abstract
The capability of selectively division encrypted data with different users via public cloud storage may greatly ease security concerns over involuntary data leaks in the cloud. A key test to design such encryption idea lays in the well-organized organization encryption keys. The preferred flexibility of allocating any group documents with any group of users by attaining weight age different encryption keys to be used for different documents. However, this also implies the necessity of securely dispensing to users a large number of keys for both encryption and search, and those users will have to steadily store the received keys, and submit an equally large number of keyword doorways to the cloud in order to perform search over the shared data.A novel concept of key aggregate searchable encryption (KASE) is to disseminate data to a large number of users by cloud service provider is found to be an effective approach. Hence the proposed system has an efficient key management scheme, key reuse to handle key distribution with regard to complex subscription(querying) options ,Time duration and user activities.
Keywords- *data division, data confidentiality, cloud storing, KASE*

INTRODUCTION

Nowadays the storing in the cloud has materialized as a capable answer for seemly and on-demand accesses to enormous amounts of information shared over the Internet. Business users are being paying attention by cloud storing due to its several benefits, including worse cost, better quickness, and improved resource utilization. Everyday users are also division private data, such as photos and videos, with their friends through social network applications based on cloud [2]. On the other hand, while benefiting from the expediency of sharing data through cloud storage, users are also increasingly worried about unintentional data reveal by the cloud. However, while enjoying the accessibility of sharing data via cloud storage, users are also gradually concerned about accidental data betrayals in the cloud. Such data leaks, caused by a malicious adversary misbehaving cloud operator, can usually lead to serious breaches of personal privacy or business secrets(e.g., the recent high profile happening of celebrity photos being leaked in iCloud). Even though integration a searchable encryption Scheme with cryptographic cloud storing can accomplish the essential safety needs of a cloud storage, executing such a system for large measure application relating huge number of users and large number of files may still be delayed by accurate issues relating the well-organized management of encryption keys, which, to the finest of our knowledge.Primarily, the want for selectively distribution encrypted data with different users usually demands unlike encryption keys to be used for unlike files. On the other indicator, this involves the amount of keys that

need to be spread to users, both for them to search over the encrypted records and to decrypt the files, will be relative to the number of such files. Such a large numeral of keys must not only be spread to users via protected channels, but also be securely stored and handled by the users in their devices.[5]The implicit requirement for protected communication, storage, and computational difficulty may cause system ineptness. A common solution is to engagement a searchable encryption (SE) scheme in which the data owner is required to encrypt feasible keywords and upload them to the cloud together with converted data, such that, for repossessing data matching a keyword, the user will send the equivalent keyword trapdoor to the cloud for execution search over the encoded data. Although merging a searchable encryption arrangement with cryptographic cloud storing can achieve the basic security requirements of cloud storage, implementing such a system for large measure applications involving millions of users and billions of records may still be hindered by concrete issues involving the efficient management of encryption keys, which, to the best of our knowledge, are largely ignored in the nonfiction. In this paper, we address this challenge by suggesting the novel impression of key-aggregate searchable encryption (KASE), and instantiating the concept through a concrete KASE scheme [7]. The proposed KASE scheme applies to any cloud storing that supports the searchable group data division functionality, which means several user may selectively share a group of selected files with a group of selected users, while allowing the latter to complete keyword quest over the former. Today, millions of users are division personal data, such as prints and videos, with their friends through social complex applications based on cloud storing on a daily basis. To support searchable group data sharing the main necessities for efficient key management are twofold. First, a data proprietor only needs to allot a single aggregate key (instead of a group of keys) to a user for sharing any number of accounts. Second, the user only needs to acquiesce a single aggregate trapdoor (instead of a group of trapdoors) to the cloud for execution keyword examination over any number of shared files. To the best of our knowledge, the KASE pattern planned in this paper is the first known scheme that can satisfy both necessities (the key-aggregate cryptosystem [4], which has motivated our work, can gratify the first requirement but not the second). Such a cloud storage is often called the cryptographic cloud storing [6].Such data revealing, will be performed by malevolent opponent or a impish cloud operator, can normally direct to severe desecration of private data or confidential data regarding business. However, the encryption of data makes it perplexing for users to search and then selectively retrieve only the data containing given keywords.

Contributions are as follows.

1. We first define a broad-spectrum agenda of key aggregate searchable encryption (KASE) serene of seven polynomial algorithms for sanctuary parameter setup, key group, encryption, key extraction, trapdoor generation, trapdoor regulation, and trapdoor testing. We then designate both functional and security requirements for conniving a valid KASE arrangement.
2. We then instantiate the KASE agenda by designing a concrete KASE scheme. After providing detailed structures for the seven algorithms, we analyze the productivity of the scheme, and establish its safety through detailed analysis.
3. We discuss various concrete issues in building an actual group data division system based on the proposed KASE scheme, and appraise its performance. The appraisal confirms our system can meet the performance necessities of practical applications.

PRELIMINARIES

In this section, we review some basic conventions and cryptology notions which will be needed later in this paper. In the rest of our reflections, let G andG1 be two cyclic groups of chief order p, and

g beak creator of G. Moreover, let doc be the article to be scrambled, k the searchable encryption key, and Tr the trapdoor for keyword exploration.

Broadcast Encryption

In a announcement encryption (BE) scheme, a broadcaster encrypts a memorandum for some subset S of users who are eavesdropping on a broadcast channel. Any user in Examination uses his/her private key to decrypt the recording. A BE scheme can be pronounced as a tuple of three polynomial-time algorithms BE = (Setup, Encrypt, Decrypt)as follows:

Setup: this algorithm is course by the systemto set up the scheme. It takes as input a safety parameter and the quantity of receivers n, outputs n private keys and a communal key pk.

Encrypt (pk; S): this algorithm is outing by the broadcasterto encrypt a missive for a subset of users. It takes as input a communal key pk and a subset of users crops a pair (Hdr,K),where Hdr is called the header and K is a memorandum encryption key which is compressed inHdr. We will often refer to Hdr as the recording cipher text. For a concrete dispatch, it will be encrypted by K and disseminated to the users inS.

Decrypt(pk; S; i; di;Hdr): this algorithm is outing by the user to decrypt the received memos. It takes as input a public key pk, a subdivision of users a user id i 2 f1;the private keydi for consumers and a header Hdr, outputs the missive encryption key K or the failure symbol?. The K will be recycled to decrypt the customary messages. To ensure the system to be accurate, it is required that, for all S ; ng and completely 2 S, if **Setup**(1, n) and (R **Encrypt**(pk; S)),formerly **Decrypt**(pk; S; i; di;Hdr)=K.

Searchable Encryption

Generally discourse, searchable encryption schemes tumble into two categories, i.e., searchable symmetric encryption (SSE) and municipal key encryption with keyword examine (PEKS). Both SSE and PEKS can be designated as the tuple SE= (Setup, Translate, Trapdoor, Test):

- **Setup**: this procedure is run by the owner to set up the order. It takes as input a safetyparameter1_, and outputs the compulsory keys.
- **Encrypt** (k;m): this algorithm is route by the owner to encrypt the data and spawn its keyword cipher texts. It takes as input the data m, owner's necessary keys counting searchable encryption keyk and data encryption key, productions data cipher text and keyword cryptograph texts Cm.
- **Trpdr**(k;w): this algorithm is route by a user to produce a doorways for a keyword w using deliberate.
- **Test** (Tr, Cm): this algorithm is course by the cloud attendant to perform a keyword search over scrambled data. It takes as participation trapdoor Tr and the keyword cipher texts Cm, crops whether Cm contains the identified keyword.

THE KEYAGGREGATE SEARCHABLE ENCRYPTION(KASE) FRAMEWORK

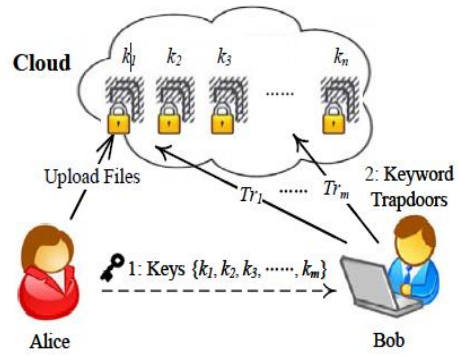
In this section, we first describe the universal problem, and then define a common framework for key comprehensive searchable encryption (KASE) and provide necessities for designing a valid KASE scheme.

Problem Statement

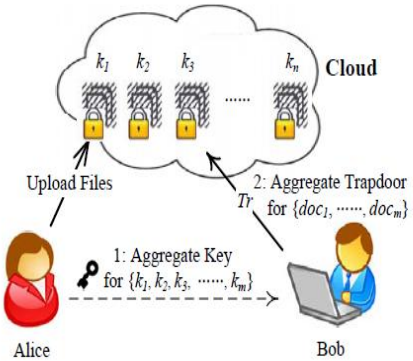
Consider a consequence where two employees of attend would like to share some confidential professional data spending a public cloud storage service (e.g. Drop box or simplicity). For illustration, Alice wants to upload a large assemblage of financial documents to the cloud storing, which are meant for the executives of different departments to review. Suppose those brochures contain highly sensitive information that should only be opened by authorized users, and Bob is one of the boards and is thus sanctioned to view documents related to his department. Due to concerns about impending

data leakage in the cloud, Alice translates these documents with different keys, and engenders keyword cipher texts based on subdivision names, before uploading to the cloud storing. Alice then uploads and parts those documents with the executives using the sharing functionality of the cloud storing. In order for Bob to view the pamphlets related to his department, Alice must deputy to Bob the rights both for keyword search over those brochures, and for decryption of brochures related to Bob's department.

In this paper, we propose the novel method of key-aggregate searchable encryption (KASE) by way of a improved solution, as showed in Fig.1 (b). , in KASE, Alice only needs to dispense a single aggregate key, instead of for allocation documents with Bob, and Bob only needs to acquiesce a single aggregate trapdoor, in its place of, to the cloud server. When m is appropriately large, the key circulation and storage as well as the trapdoor generation may become too exclusive for Bob's client-side device, which principally defies the persistence of using cloud storage.



(a) Traditional approach



(b) Key-Aggregate Searchable Encryption

Fig. 1. keyword search in group data partaking system.

To proposal a key-aggregate searchable encryption method under which any subdivision of the keyword cipher texts from any set of brochures is searchable with a constant-size doorway generated by a continual size aggregate key.

Therefore, in KASE, the assignment of keyword search right can be reached by sharing the single aggregate key. We note that the designation of decryption rights can be accomplished using the key-aggregate encryption attitude recently proposed in [4], but it vestiges an open problem to delegate the keyword search rights unruffled with the decryption rights, which is the focus topic of this paper. To summarize, the problem of manufacturing a KASE structure can be stated as:

"To proposal a key-aggregate searchable encryption scheme under which any subcategory of the keyword encryption texts (produced by the SE. Encrypt algorithm to be announced in Section5) from any set of brochures is searchable (performed by the SE.Test procedure) with a constant-size doorway(produced by SE.Trpdr algorithm) generated by a continual size aggregate key."

The KASE Framework

The KASE framework is unruffled of seven algorithms. Specifically, to set up the organization, the cloud server would create public parameters of the system concluded the **Setup** algorithm, and these public restrictions can be reused by different data owners to segment their files. For each data owner, he/she should harvest a public/master-secret key pair concluded the **Keygen** algorithm. Keywords of each document can be scrambled via the **Encrypt** algorithm with the inimitable searchable encryption key. Then, the data titleholder can use the master-secret key to cause an aggregate searchable encryption key for a group of nominated documents via the **Extract** algorithm. The cumulative key can be circulated securely (e.g., via protected e-mails or secure devices) to sanctioned users who need to access those brochures. After that, as shown in Fig.2,an authorized user can harvest a keyword

doorway via the **Trapdoor** algorithm using this aggregate key, and acquiesce the trapdoor to the cloud.

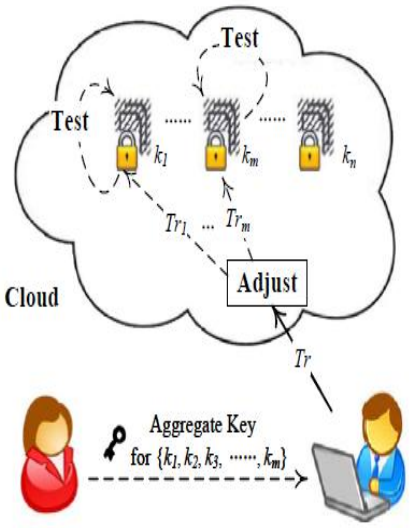


Fig. 2. Structure of key-aggregate searchable encryption

After that, as shown in Fig.2, an specialized user can create a keyword doorway via the **Trapdoor** algorithm using this aggregate key, and acquiesce the trapdoor to the cloud. After attainment the trapdoor, to carry out the keyword rifle over the particular set of brochures, the cloud server will run the **Adjust** algorithm to harvest the right trapdoor for each document, and then route the **Test** algorithm to test whether the article contains the keyword.

This construction is summarized in the following.

1. Setup(1λ, n): This algorithm is route by the cloud service benefactor to set up the scheme. On input of a security parameter 1λ and the supreme possible number n of brochures which belongs to a data owner, it outputs the public organisation parameter pushchairs.

2. **Keygen**: This algorithm is route by the data owner to generate a arbitrary key pair (pk,msk).

3.**Encrypt**(pk, i): This algorithm is route by the data owner to encrypt the i-th article and generate its keywords' cipher texts. For each article, this algorithm will fashion a delta Дi for its searchable encryption key ki. On input of the possessor's public key pk and the organiser index i, this algorithm harvests data cipher text and keyword cipher texts Ci.

3. **Extract** (msk, S): This algorithm is route by the data possessor to generate an aggregate searchable encryption key for indicator over the keyword search right for a confident set of documents to other users. It takes as participation the owner's master-secret key msk and a set S which encompass the directory of brochures, and then outputs the aggregate crucial kagg.

4. **Trapdoor**(kagg, x): This algorithm is route by the consumer who has the aggregate key to perform
search. It takes as contribution the aggregate searchable encryption strategic kagg and a keyword w, then productions only one trapdoor Trd.

5. **Adjust** (params, i, S, Trd): this algorithm is route by cloud attendant to adjust the aggregate trapdoor to generate the right doorway for each different document. It takes as input the coordination public constraints params, the set S of documents' indices, the index i of target article and the aggregate trapdoor Tr, then harvests each trapdoor Tri for the i-th target article in S.

6.**Test**(Tri, i): this algorithm is route by the cloud server to perform keyword examine over an encrypted document. It takes as involvement the trapdoor Tri and the article index i, then outputs true or false to denote whether the article doci contains the keyword w.

Requirements for Designing KASE Schemes

The KASE agenda introduced in the previous section provides general supervision to designing aKASE scheme. However, a binding KASE scheme must also mollify several functional and security requirements, as itemized in the following.

A KASE scheme should placate three functional necessities as follows.

1.Compactness. This requirement anxieties aKASE scheme to confirm the size of the aggregate key to be autonomous of the number of

files to be shared. Formally, for a set of secretsfkigi2S,it requires that kagg**Citation**(msk, S). How to aggregate the set of explanations into a single key without nullifying later steps is a key challenge in deceitfulASE schemes.

2.Searchability. This requirement is essential to all
KASE schemes since it empowers users to generate desired doorways for any given keyword for searching encrypted brochures. In another word, reducing the number of keys should reservethe search capability. Formally, for each article inclosing the keyword w with index i 2 S, the search ability necessitates that if (Tr=**Trapdoor**(kagg,w) and Tri**Amend**(prams, i, S, Tr)), then**Test**(Tri, i)=true.

3. **Delegation**. The focal goal of KASE is to delegate the keyword search precise to a user through an aggregate key. To ensure any employer with the delegated key can accomplish keyword search, this requirement involves that the inputs of the adjustment algorithm must not be municipal, i.e., these participations should not rely on any user's private information. This is the succeeding key challenge in deceitful KASE schemes.

4. Controlled searching. Meaning that the muggers cannot search for an capricious word without the data owner's permission. That is, the attacker cannot perform keyword search over the pamphlets which are not germane to the known aggregate key, and he/she cannot generate new comprehensive searchable encryption secrets for other setoff brochures from the known keys.

5.Query privacy. Meaning that the invaders cannot determine the keyword used in a interrogation, apart from the information that can be conquered via reflection and the information derived from it. That is, the user may ask an unimportant cloud server to search for a sensitive word without tight fitting the discussion to the server.

RELATED WORK

Before we announce our KASE scheme, this section first assessments several categories of existing solutions and explain their affiliations to our work.

Multi-user Searchable Encryption

There is a rich nonfiction on searchable encryption, counting SSE schemes [5]–[8] and PEKS schemes[9]–[15]. In distinction to those existing work, in the context of cloud storing, keyword search under thematic-tenancy venue is a more common situation. In such a scenario, the data owner would like to share a certificate with a group of authorized users, and each employer who has the access right can provide a doorway to perform the keyword search over the shared manuscript, namely, the "multi-user searchable encryption" (MUSE) consequence.

Multi-Key Searchable Encryption

The goal of KASE is to ambassador the keyword search right to any manager by allotting the aggregate key to him/her in a group data division system, whereas the goal of MKSE is to confirm the cloud server can accomplish keyword search with one trapdoor over unlike documents owing to a user. MKSE allows user to afford a single keyword doorway to the server, but still allows the server to search for that doorway's keyword in documents encrypted with different explanations. This might sound very analogous tothe goal of KASE, but these are in fact two completely different perceptions.

Key-aggregate Encryption for Data Sharing

Data sharing schemes based on cloud storage have Attracted much devotion recently [1]–[4]. In particular, Chu et al. [4] cogitate how to reduce the number of scattered data encryption keys. To share several brochures with different encryption keys with the same user, the data possessor will need to distribute all such keys to him/her in a outmoded approach which is habitually impractical. Aiming at this experiment, a key aggregate Encryption (KAE) scheme for data allotment is proposed to generate an aggregate

strategic for the user to decrypt all the brochures. To allow a set of documents translated by different Sources to be decrypted with a single aggregate key, user could encrypt a memo not only under a public-key, but also beneath the identifier of each document. The construction is encouraged by the recording encryption scheme [27]. In this construction, the data owner can be observed as the broadcaster, who has free keypk and master-secret key msk; each manuscript with identifier i can be regarded as a receiver attending to the broadcast channel, and a public evidence used in decryption is designed to be appropriate to both the owner's msk and the encryption key; the missive encryption process is analogous to data encryption using symmetric encryption in BE, but the key accumulation and data decryption can be simply regarded as the further scientific conversion of BE. **Encrypt**practice and BE. **Decrypt** algorithm respectively.

PERFORMANCE EVALUATION

Considering that: 1) in a energetic data sharing system based on cloud storage, the user can retrieve data by any possible maneuver and the mobile devices are frequently used now; 2) the performance is highly conditional on on the basic cryptographic operations especially in the coupling computation, we study whether the cryptographic maneuvers based on combining computation can be efficiently executed using both workstations and mobile devices.

Pairing Computation

About pairing computation, some investigate results have been published. In mobile expedients, Oliveiraet al. [25] show that it needs 5 succeeding in 2007, but now it will be faster; In feeler nodes and personal numeral assistant (PDA), Li et al. [26] shows that it only needs1.5 instant and 0.5 second individually in 2010.

TABLE 1

Execution times of type A pairing computation (ms)

	Pairing	pow(in $\mathcal{G}$)	pow(in $\mathcal{G}_1$)	pow(in Z_p)
Mobile Devices	485	243	74	0.8
Computer	10.2	13.3	1.7	0.05

Evaluation of KASE Algorithms

Considering that the processes including KASE.**Setup**, KASE.**Adjust** besides KASE.**Test** are only route in the cloud server, only the completing times in workstation are tested. As shown in Fig.4, we can see that:

1) The execution while of KASE.**Setup** is linear in

the maximum number of brochures belonging to one owner, and when the determined number grows up to 20000, it is practical thatKASE.**Setup** algorithm only needs 259 instant.

2) The finishing time of KASE.**Encrypt** is rectilinear in the numeral of keywords, and when the number cultivates up to 10000, KASE.**Encrypt** algorithm only requests 206 second in computers, but 10018second in mobile procedures. Therefore, we can draw two decisions; one is that it is not feasible to upload manuscript with lots of keywords using a mobile receiver; the other is that the keyword search with combining computation can be executed quickly in workstations now.

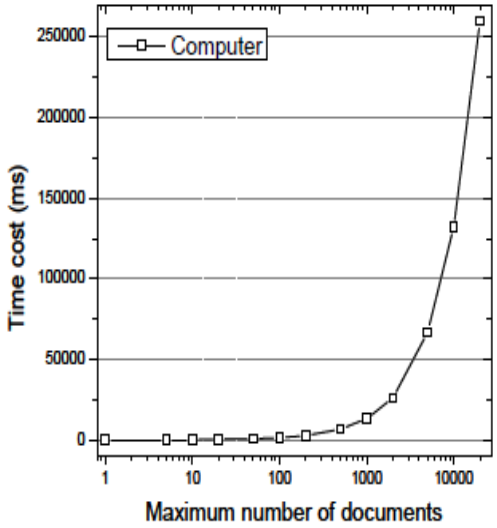
3) The effecting time of KASE.**Extract** is linear in the figure of shared documents, and when the number raises up to 10000, KASE. **Extract** algorithm only desires 132 second in computer, but 2430 instant in mobile devices. For the KASE. **Extract** always scores along with the KASE. **Encrypt**, it is not proposed to be executed in the mobile campaigns

4) The execution stretch of KASE.**Trapdoor** is a constant,i.e., 0.01 succeeding in computer and 0.25 second in movable devices. In fact, the

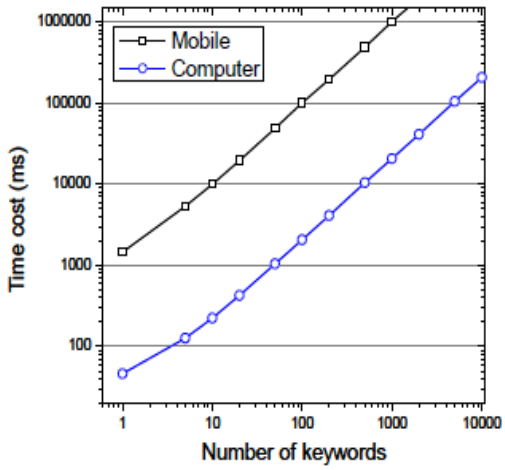
mathematical Maneuver in KASE.**Trapdoor** is the once duplication in G, so that the keyword search can be performed resourcefully in both mobile devices and computer. Compared with other patterns, there is a weighty improvement in our scheme.

5) The execution while of KASE.**Adjust** is linear in the number of brochures. In fact, it can be improved in the useful application, and the details are shown in section 6.4.

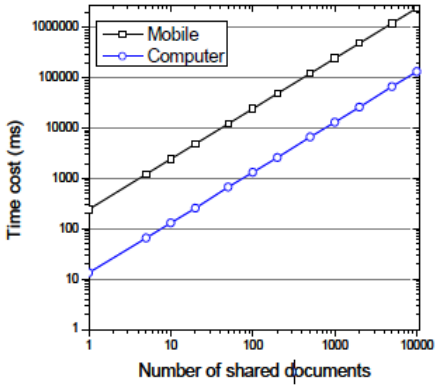
6) The execution stretch of KASE.**Test** is linear inthe quantity of keyword cipher texts. In fact, the mathematical procedure in KASE.**Test** is twice as much as the mixture computations. When the number raises up to 20000, it will take 467second.



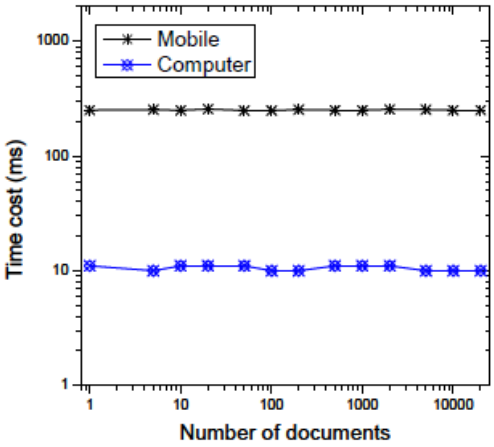
(a) Time cost of Setup



(b) Time cost of Encrypt



(c) Time cost of Extract



(d) Time cost of Trapdoor

Fig. 4. Time cost of KASE algorithms.

Evaluation of the Group Data Sharing System

Considering that the structure's performances most Unfavorably depend on the KASE algorithms, we consider employing storing techniques in the group data sharing system to further progress the efficiency of the **keyword search** practice. After receiving an cumulative trapdoor, the cloud server will run KASE. **Adjust**and KASE. **Test** to texture the keyword search. Fig.4(e) displays that the execution spell of KASE.**Adjust** is linear in the number of brochures. the performance, the cloud server can reserve the computation result of (S, i, P), where P is theartefact_j2s;j6=ign+1□j+i. Because the input and calculation development are the same for all users, this will critically save the calculation time.

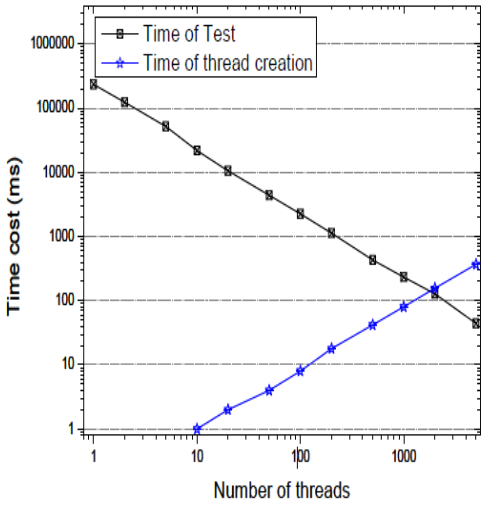


Fig. 5. Time cost of keyword search.

When the user demands for the second time, or the same S has seemed in the past queries, KASE. **Adjust** can route quickly by using the pre-computed result. Fig.4(f) displays that the execution time of KASE.**Test**is linear in the integer of keyword cipher texts. To progress the efficiency, some parallel totaling and distributed computing techniques maybe pragmatic, such as multi-thread, hadoop, etc. The multi-thread practice is adopted in our experiment. When the quantity grows up to 200, it only needs 1 second to appearance the keyword search over 10000 keyword cryptograph texts. We also see that when the cipher of threads is large, it would take more time to produce these threads. When the quantity grows up to 1000, the time of thread creation will convert 80 millisecond. So, the multithread procedure can provide the help for improving performance, but the quantity of threads should be selected judiciously in the practical applications.

CONCLUSION & FUTURE ENHANCEMENT

Taking into consideration of the genuine problem of privacy protective data sharing scheme based on public cloud storing which is need a data owner to allocate a large number of keys to consumers to certification them to access the documents, In this proposed perception of key-aggregate searchable encryption (KASE) and construct a material KASE scheme. It can provide an proficient solution to building concrete data sharing system based on public cloud storage. In a KASE scheme, the holder needs to distribute a single strategic to a user when contributing a lot of brochures with the user, and the user needs to submit a single doorway when they queries over all documents collective by the same owner. On the other hand, if a user requests to question over documents joint by multiple owners, that user must produce multiple entrances to the cloud. The future heightening for this proposed work is to find out how to shrinkage the number of trapdoors under multi-owners situation by attaining the safety.

REFERENCES

1. 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.
2. R. Lu, X. Lin, X. Liang, and X. Shen, “Secure Provenance: The Essential of Bread and Butter of Data Forensics in CloudComputing”, Proc. ACM Symp. Information, Computer and Comm. Security, pp. 282-292, 2010.
3. 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.
4. C. Chu, S. Chow,W. Tzeng, et al. “Key-Aggregate Cryptosystemfor Scalable Data Sharing in Cloud Storage”, IEEE Transactionson Parallel and Distributed Systems, 2014, 25(2): 468-477.
5. X. Song, D.Wagner, A. Perrig. “Practical techniques for searcheson encrypted data”, IEEE Symposium on Security and Privacy,IEEE Press, pp. 44C55, 2000.
6. R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky. “Searchable symmetric encryption: improved definitions and efficient constructions”,In: Proceedings of

the 13th ACM conference on Computer and Communications Security, ACM Press, pp. 79-88, 2006.

7. P. Van,S. Sedghi, JM. Doumen. "Computationally efficient searchable symmetric encryption", Secure Data Management, pp.87-100, 2010.
8. S. Kamara, C. Papamanthou, T. Roeder. "Dynamic searchable symmetric encryption", Proceedings of the 2012 ACM conference on Computer and communications security (CCS), ACM, pp. 965-976, 2012.
9. D. Boneh, C. G, R. Ostrovsky, G. Persiano. "Public Key Encryption with Keyword Search", EUROCRYPT 2004, pp. 506C522,2004.
10. Y. Hwang, P. Lee. "Public Key Encryption with ConjunctiveKeyword Search and Its Extension to a Multi-user System", In:Pairing-Based Cryptography C Pairing 2007, LNCS, pp. 2-22, 2007.
11. J. Li, Q. Wang, C. Wang. "Fuzzy keyword search over encrypteddata in cloud computing", Proc. IEEE INFOCOM, pp. 1-5,2010.
12. C. Bosch, R. Brinkma, P. Hartel. "Conjunctive wildcard searchover encrypted data", Secure Data Management. LNCS, pp. 114-127, 2011.
13. C. Dong, G. Russello, N. Dulay. "Shared and searchable encrypted data for untrusted servers",Journal of Computer Security,pp. 367-397, 2011.
14. F. Zhao, T. Nishide, K. Sakurai. Multi-User Keyword SearchScheme for Secure Data Sharing with Fine-Grained Access Control. Information Security and Cryptology, LNCS, pp. 406-418,2012.
15. 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.