International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 11
July 2016

Cloud Data Allocation for Members Data Using KASE

Mr. B.TIRUPATHI KUMAR
Asst. Professor
Department of CSE

Ms. SUMA MALAKA
M.Tech in Computer Science
Department of CSE

Malla Reddy Institute of Technology, Maisammaguda, Dhulapally Post, Secunderabad and Telengana State, India.

Abstract:

The capability of selectively sharing encrypted facts with one kind of users via public cloud storage may additionally substantially ease safety issues over inadvertent statistics leaks inside the cloud. A key undertaking to designing such encryption schemes lies in the efficient management of encryption keys. The favored flexibility of sharing any institution of selected documents with any organization of customers demands different encryption keys to be used for unique files. But, this additionally implies the necessity of securely dispensing to users a massive variety of keys for both encryption and seeks, and people customers will have to securely shop the acquired keys, and submit an equally huge variety of keyword trapdoors to the cloud for you to carry out search over the shared statistics. The implied need for comfy communication, storage, and complexity genuinely renders the approach impractical. Here, we cope with this sensible trouble, which is essentially omitted inside the literature, through providing the radical concept of key aggregate searchable encryption (kase) and instantiating the idea via a concrete kase scheme, wherein a records owner simplest wishes to distribute a single key to a user for sharing a huge variety of documents, and the user best needs to post a single trapdoor to the cloud for querying the shared files. The security analysis and overall performance evaluation both verify that our proposed schemes are provably comfy and nearly green. **Index termss:** searchable encryption, records sharing, cloud storage, records privacy.

INTRODUCTION:

Cloud storage has emerged as a promising solution for providing ubiquitous, handy, and on-call for accesses to large quantities of records shared over the net. Today, millions of customers are sharing private statistics, including photos and films, with their pals via social network applications based totally on cloud garage on a day by day basis. Business users are also being attracted by cloud garage because of its numerous blessings, which include lower fee, extra agility, and higher resource utilization. However, even as enjoying the benefit of sharing statistics through cloud garage, customers also are more and more concerned about inadvertent records leaks in the cloud. Such records leaks, as a result of a malicious adversary or a misbehaving cloud operator, can generally result in extreme breaches of private privateness or commercial enterprise secrets and techniques (e.g., the recent high profile incident of celebrity pictures being leaked in icloud). To address customers' worries over ability records leaks in cloud storage, a common technique is for the statistics owner to encrypt all of the facts earlier than importing them to the cloud, such that later the encrypted records can be retrieved and decrypted by means of those who've the decryption keys. Such a cloud garage is regularly called the cryptographic cloud garage. However, the encryption of statistics makes it difficult for customers to go looking and then selectively retrieve most effective the statistics containing given key phrases. A not unusual answer is to employ a searchable encryption (SE) scheme wherein the facts owner is needed to encrypt capability key phrases and upload them to the cloud collectively with encrypted information, such that, for retrieving statistics matching a keyword, the person will ship the corresponding key-word trapdoor to the cloud for acting seek over the encrypted information.

Despite the fact that combining a searchable encryption scheme with cryptographic cloud garage can reap the simple security necessities of cloud storage, implementing such a gadget for massive scale applications related to thousands and thousands

of users and billions of files can also nonetheless be hindered through realistic issues concerning the green management of encryption keys, which, to the pleasant of our expertise, are in large part not noted inside the literature. Initially, the need for selectively sharing encrypted facts with one kind of customers (e.g., sharing a picture with certain pals in a social network utility, or sharing a business document with sure colleagues on a cloud force) commonly demands distinctive encryption keys to be used for different files. However, this means the quantity of keys that want to be disbursed to users, both for them to look over the encrypted documents and to decrypt the documents, will be proportional to the wide variety of such documents. This kind of massive number of keys should not only be distributed to users via secure channels, however also be securely saved and controlled by the customers of their gadgets. Similarly, a massive range of trapdoors need to be generated by way of users and submitted to the cloud with a purpose to carry out a key-word seek over many files. The implied need for cozy conversation, garage, and computational complexity may render this kind of device inefficient and impractical. Here, we deal with this venture by way of offering the radical idea of key-aggregate searchable encryption (KASE), and instantiating the concept through a concrete KASE scheme. The proposed KASE scheme applies to any cloud storage that helps the searchable institution information sharing functionality, which means that any consumer may also selectively proportion a group of decided on documents with a collection of decided on users, while allowing the latter to carry out key-word seek over the previous. To help searchable institution facts sharing the main requirements for green key control are twofold.

First, a records owner best desires to distribute a single mixture key (as opposed to a group of keys) to a user for sharing any variety of documents. 2nd, the user best wishes to put up a single aggregate trapdoor (in place of a collection of trapdoors) to the cloud for performing key-word seek over any variety of shared files. To the high-quality of our expertise, the KASE scheme proposed in this paper is the first regarded scheme which could satisfy both necessities (the key-combination cryptosystem, which has inspired our work, can fulfill the primary requirement however not the 2nd) contributions. More particularly, our essential contributions are as follows:

1) We first outline a trendy framework key aggregate searchable encryption (KASE) composed of seven polynomial algorithms for protection parameter setup, key generation, encryption, key extraction, trapdoor technology, trapdoor adjustment, and trapdoor checking out. We then describe each purposeful and safety necessities for designing a legitimate KASE scheme. 2) We then instantiate the KASE framework by means of designing a concrete KASE scheme. After imparting specific buildings for the seven algorithms, we analyze the performance of the scheme, and establish its safety through targeted evaluation.

3) We discuss diverse realistic problems in constructing an actual organization records sharing system based on the proposed KASE scheme, and compare its performance. The evaluation confirms our system can meet the performance requirements of practical packages.

2 PRELIMINARIES

In this segment, we evaluate some basic assumptions and cryptology ideas for you to be needed later. Within the rest of our discussions, permit g and g1 be the cyclic companies of higher order p, and g be a generator of g. furthermore, let document be the report to be encrypted, k the searchable encryption key, and t_r the trapdoor for key-word search.

Complexity Assumption:

Bilinear Map:

A bilinear map is a map e : G×G->G1 with the following properties:

1. Bilinearity: for all u; v € G and a; b Z_p^*, we have $e(u^a; v^b) = e(u; v)^{ab}$.

2. Non-degeneracy: e(g, g)≠1.

3. Computability: there is an efficient algorithm to

compute e(u; v) for any u; v € G.

Bilinear Diffie-Hellman Exponent Assumption:

The bilinear Diffie-Hellman exponent (BDHE) assumption has been widely used to prove the security of some broadcast encryption (BE) schemes.

Definition. 1. The (l,ε ")-BDHE assumption holds in G if no algorithm has advantage more than ε in solving the l- BDHE problem in G.

Broadcast Encryption: In a broadcast encryption (BE) scheme, a broadcaster encrypts a message for some

subset S of users who are listening on a broadcast channel. Any user in S can use his/her private key to decrypt the broadcast. A BE scheme can be described as a tuple of three polynomial-time algorithms BE = (Setup, Encrypt, Decrypt).

Searchable Encryption: Generally speaking, searchable encryption schemes fall into two categories, i.e., searchable symmetric encryption (SSE) and public key encryption with keyword search (PEKS). Both SSE and PEKS can be described as the tuple SE= (Setup, Encrypt, Trapdoor, Test):

- Setup (1^λ): this algorithm is run by the owner to set up the scheme. It takes as input a security parameter 1^λ, and outputs the necessary keys.
- Encrypt (k;m): this algorithm is run by the owner to encrypt the data and generate its keyword cipher-texts. It takes as input the data m, owner's necessary keys including searchable encryption key k and data encryption key, outputs data cipher-text and keyword cipher-texts Cm.
- Trpdr(k;w): This algorithm is run by a user to generate a trapdoor T_r for a keyword w using key k.
- Test (Tr, Cm): this algorithm is run by the cloud server to perform a keyword search over encrypted data. It takes as input trapdoor T_r and the keyword cipher-texts Cm, outputs whether Cm contains the specified keyword.

3 THE KEY-AGGREGATE SEARCHABLE ENCRYPTION (KASE) FRAMEWORK

In this section, we first describe the general problem, and then define a generic framework for key aggregate search-able encryption (KASE) and provide requirements for designing a valid KASE scheme.

Problem Statement: Remember a situation where in employees of a employer would love to share some exclusive commercial enterprise facts the use of a public cloud storage provider (e.g., drop box or simplicity). As an example, Alice desires to add a large collection of monetary files to the cloud storage that are intended for the directors of different departments to check. Think those documents comprise exceedingly sensitive information that must best be accessed by using permitted users, and Bob is one of the directors and is accordingly legal to view documents related to his department. Due to issues approximately capacity statistics leakage within the cloud, Alice encrypts those documents with special keys, and generates key-word cipher-texts based on branch names, earlier than uploading to the cloud storage. Alice then uploads and shares the ones documents with the directors the use of the sharing functionality of the cloud storage, so as for Bob to view the documents associated with his branch, Alice have to delegate to Bob the rights both for keyword search over those files, and for decryption of documents related to Bob's branch.

Fig.1. Keyword search in group data sharing system.

- **Setup** (1^λ, n): this algorithm is run by the cloud service provider to set up the scheme. On input of a security parameter 1^λ and the maximum possible number n of documents which belongs to a data owner, it outputs the public system parameter params.
- **Keygen**: this algorithm is run by the data owner to generate a random key pair (pk,msk).

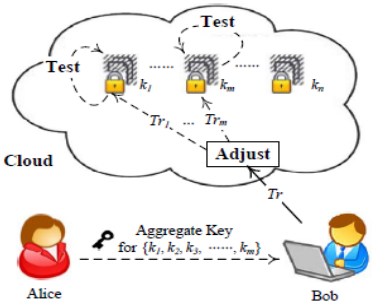
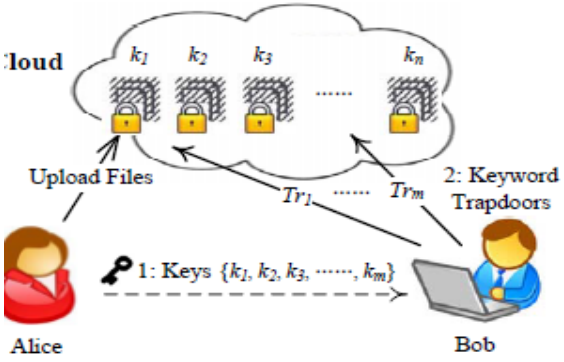
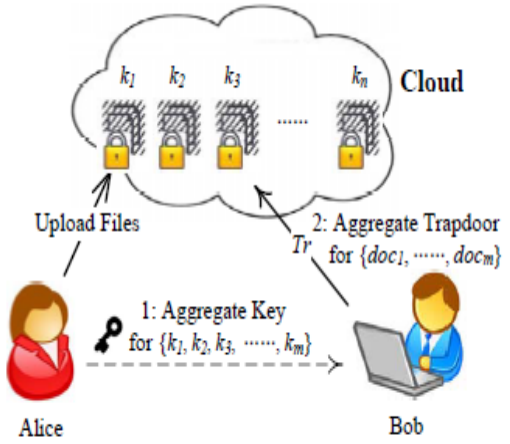


Fig. 2. Framework of key-aggregate searchable encryption.



(a) Traditional approach



(b) Key-Aggregate Searchable Encryption

- **Encrypt**(pk, i): this algorithm is run by the data owner to encrypt the i-th document and generate its keywords' cipher-texts. For each document, this algorithm will create a delta i for its searchable encryption key k_i. On input of the owner's public key p_k and the file index i, this algorithm outputs data cipher-text and keyword cipher-texts Ci.
- **Extract**(msk, S)**:** this algorithm is run by the data owner to generate an aggregate searchable encryption key for delegating the keyword search right for a certain set of documents to other users. It takes as input the owner's master-secret key. Personal use is permitted, but republication/redistribution requires IEEE permission. This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Msk and a set S which contains the indices of documents, then outputs the aggregate key kagg.
- **Trapdoor (k_{agg}, w):** This algorithm is run by the user who has the aggregate key to perform a search. It takes as input the aggregate searchable encryption key k_{agg} and a keyword w, then outputs only one trapdoor Tr.
- **Adjust (params, i, S, Tr):** This algorithm is run by cloud server to adjust the aggregate trapdoor to generate the right trapdoor for each different document. It takes as input the system public parameters params, the set S of documents' indices, the index i of target document and the aggregate trapdoor Tr, then outputs each trapdoor Tri for the i-th target document in S.
- **Test** (Tri, i): This algorithm is run by the cloud server to perform keyword search over an encrypted document. It takes as input the trapdoor Tri and the document index i, then outputs true or false to denote whether the document doc_i contains the keyword.

Requirements for Designing KASE Schemes:

- **Compactness**. This requirement demands a KASE scheme to ensure the size of the aggregate key to be independent of the number of files to be shared.
- **Searchability:** This require-ment is central to all KASE schemes since it enables users to generate desired trapdoors for any given keyword for the searching encrypted documents.
- **Delegation**. The main goal of KASE is to delegate the keyword search right to a user through an aggregate key.
- **Controlled searching**. Meaning that the attackers cannot search for an arbitrary word without the data owner's authorization.
- **Query privacy**. Meaning 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.

4. RELATED WORK:

Earlier than we introduce our kase scheme, this segment first critiques numerous categories of existing answers and give an explanation for their relationships to our paintings.

Multi-person Searchable encryption there's a wealthy literature on searchable encryption, which include SSE schemes and PEKS schemes. In assessment to those present paintings, in the context of cloud storage, key-word search underneath the multi-tenancy placing is an extra commonplace scenario. In the sort of state of affairs, the statistics owner would really like to share a document with a collection of legal customers, and each consumer who has the access right can offer a trapdoor to carry out the keyword seek over the shared document, particularly, the "multi-person searchable encryption" (MUSE) situation.

Multi key searchable encryption: Within the case of a multi-consumer utility, considering that the range of trapdoors is proportional to the quantity of files to look over (if the consumer provides to the server a keyword trapdoor under each key with which a

matching document might be encrypted), Popa firstly introduces the concept of multi-key searchable encryption (MKSE) and puts forward the first feasible scheme in 2013. MKSE allows a user to provide a single keyword trapdoor to the server, but still allows the server to search for that trapdoor's keyword in documents encrypted with different keys.

Key-aggregate Encryption for Data Sharing: Information sharing structures based on cloud storage

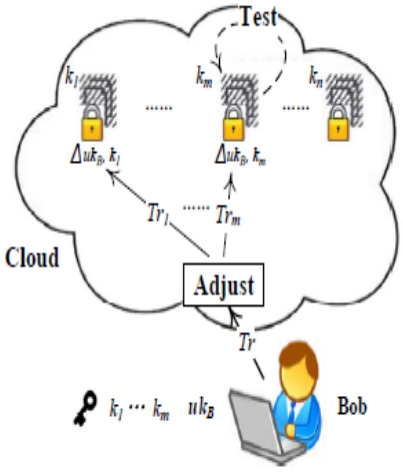


Fig. 3. Multi-Key Searchable Encryption.

have attracted tons interest these days. Mainly, Chu et al. consider a way to lessen the number of dispensed statistics encryption keys. To proportion numerous documents with special encryption keys with the equal person, the statistics proprietor will want to distribute all such keys to him/her in a traditional approach which is usually impractical. Aiming at this venture, a key aggregate Encryption (KAE) scheme for information sharing is proposed to generate an aggregate key for the user to decrypt all of the files.

5. THE PROPOSED SCHEME:

The design of our kase scheme draws its insights from each the multi-key searchable encryption scheme and the important thing-aggregate statistics sharing scheme. Specially, so that it will create an aggregate searchable encryption key as opposed to many independent keys, we adapt the idea provided. Each searchable encryption key is related to a particular index of file, and the aggregate secret is created via embedding the owner's grasp-secret key into the made from public keys related to the documents. A good way to put into effect key-word search over distinctive documents using the combination trapdoor, we hire a similar manner. The cloud server can use this technique to supply an adjusted trapdoor for each document.

Efficiency: In phrases of efficiency, our scheme certainly achieves steady-length keyword cipher-text, trapdoor and mixture keys. In addition, we ought to point out that:

1) The set S, which incorporates the indices of shared documents, has a linear size in the range of files related to the mixture key. However, this does not have an effect on the usefulness of the information sharing device, due to the fact the content of S may be appropriately saved in the cloud server (greater details might be furnished in section 5.5), such that there is no need to post them to the cloud server when submitting the trapdoor.

2) The public system parameters PubK is O(n) in size, that is linear in the most viable wide variety of files belonging to a records owner, however now not dependent on the range of files stored inside the cloud server, and therefore this could not have an effect on the system's practicality.

Security Analysis:

To analyze the security of our scheme, and in particular show that the scheme satisfies the security requirements given in Section 3.3, we assume that the public cloud is "honest-but-curious". That is, the cloud server will only provide legitimate services according to pre-defined schemes, although it may try to recover secret information based on its knowledge.

Theorem 1 requires that any user with the aggregate key can perform a keyword search over documents in the set S, but he cannot do it over the documents outside this set. He also cannot generate other aggregate searchable encryption keys for a new set so from the known one. Theorem 1 can be deduced from the following lemmas:

Lemma 1 is equivalent to the correctness of the proposed scheme. After receiving the submitted single trapdoor T_r, the cloud server can adjust T_r to generate a desired trapdoor Tri for the i-th document in the S, and then execute KASE. Test algorithm to perform keyword search. For correctness, we can see that:

International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 11
July 2016

1) Retrieve the value of t from the known c1 or c2. However, the discrete logarithm problem means A cannot compute the value of t in this case.

2) Compute the value of e $(g_1, g_n)^t$. Notice that A can get the value of e $(g;H(w))^t$ by computing e (c1;H(w)), so when he gets the value of e $(g_1; g_n)^t$, he will determine whether keyword w is in the c_w of the traget document. To obtain e $(g1; gn)^t$, A will compute e(c1; gn+1). However, because PubK is missing the term gn+1 = gan+1, the attacker A cannot finish this computation. In fact, this result is ensured by the assumption of the intractability of BDHE problem. As a result, an attacker cannot learn the content from the stored information.

A Concrete Group Data Sharing System:

- **Table-group:** <groupID, groupName, parameters> is to store the system parameters.
- **Table member:** <memberID, member-Name, password, publicKey> is to store members' information including their public key.
- Table **docs:** <doc-ID, doc-Name, Owner-ID, Enc-Key, SE-Key, file-Path> is to store the uploaded document of an owner with identity ownerID.
- **Table sharedDocs:** <SID, member-ID, Owner-ID, doc-ID-Set> is to store the documents of a member with identity member-ID shared by the owner with identity Owner ID. Field doc-ID-Set is for all the indices of documents.

$$
\begin{aligned}
& e(Tr_i, c_1)/e(pub, c_2) \\
= & \frac{e(k_{agg} \cdot \Pi_{j\in s, j\neq i} g_{n+1-j+i} \cdot H(w), g^t)}{e(\Pi_{j\in s} g_{n+1-j}, (v \cdot g_i)^t)} \\
= & \frac{e(k_{agg}, g^t) \cdot e(\Pi_{j\in s, j\neq i} g_{n+1-j+i} \cdot H(w), g^t)}{e(\Pi_{j\in s} g_{n+1-j}, (v \cdot g_i)^t)} \\
= & \frac{e(k_{agg}, g^t) \cdot e(\Pi_{j\in s, j\neq i} g_{n+1-j+i} \cdot H(w), g^t)}{e(\Pi_{j\in s} g_{n+1-j}, g^{\gamma t}) \cdot e(\Pi_{j\in s} g_{n+1-j}, (g_i)^t)} \\
= & \frac{e(\Pi_{j\in s, j\neq i} g_{n+1-j+i} \cdot H(w), g^t)}{e(\Pi_{j\in s} g_{n+1-j}, (g_i)^t)} \\
= & \frac{e(\Pi_{j\in s, j\neq i} g_{n+1-j+i}, g^t) \cdot e(H(w), g^t)}{e(\Pi_{j\in s} g_{n+1-j}, (g_i)^t)} \\
= & \frac{e(\Pi_{j\in s, j\neq i} g_{n+1-j+i}, g^t) \cdot e(H(w), g^t)}{e(\Pi_{j\in s} g_{n+1-j+i}, g^t)} \\
= & \frac{e(H(w), g^t) \cdot e(\Pi_{j\in s, j\neq i} g_{n+1-j+i}, g^t)}{e(\Pi_{j\in s} g_{n+1-j+i}, g^t)} \\
= & \frac{e(H(w), g^t) \cdot \frac{e(\Pi_{j\in s} g_{n+1-j+i}, g^t)}{e(g_{n+1}, g^t)}}{e(\Pi_{j\in s} g_{n+1-j+i}, g^t)} \\
= & \frac{e(H(w), g^t)}{e(g_{n+1}, g^t)} = \frac{e(H(w), g)^t}{e(g_1, g_n)^t} \\
= & \ c_w \qquad (1)
\end{aligned}
$$

Work Flows: To further describe this system in details, we describe its main work flows in this section. System -setup. When an organization submits a request, the cloud will create a database containing above four tables, assign a group-ID for this organization and insert a record into Table Company. Moreover, it assigns an administrator account for the manager.

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

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

6. Overall performance evaluation:

Thinking about that: 1) In a sensible information sharing system based on cloud garage, the consumer can retrieve records through any feasible tool and the cellular devices are extensively used now; 2) the overall performance is distinctly structured on the fundamental cryptographic operations specifically in the pairing computation, we look at whether the cryptographic operations based totally on pairing computation may be effectively performed using both computer systems and cell devices.

Implementation information: In our implementation, two source libraries about pairing computation are used: 1) jpbc library is used to implement cryptographic operations running in mobile smartphones; 2) p.c library is used to implement cryptographic operations jogging in computers.

Pairing Computation: About pairing computation, some experiment results have been published.

Evaluation of KASE Algorithms:

1) The execution time of KASE. Setup is linear in the most range of documents belonging to one proprietor, and when the most quantity grows up to 20000, it's far reasonable that KASE. Setup set of rules most effective needs 259 second.

2) The execution time of KASE.Encrypt is linear i the range of keywords, and whilst the quantity grows as much as ten thousand, KASE. Encrypt algorithm handiest needs 206 2nd in computer systems, however 10018 second in mobile gadgets. Consequently, we can draw conclusions:1) It is not possible to add document with plenty of key phrases the usage of a cellular phone; 2) The key-word search with pairing computation can be performed speedy in computers now. 3) The execution time of KASE. Extract is linear in the range of shared documents, and while the quantity grows up to 10000, KASE. Extract algorithm most effective desires 132 second in computer, however 2430 second in cell devices. Because the KASE. Extract continually runs along with the KASE. Encrypt, it isn't always counseled to be done in the cell devices. 4) The execution time of kase. Trapdoor is a constant, i.e., 0.01 second in laptop and 0.25 second in cellular gadgets. In fact,

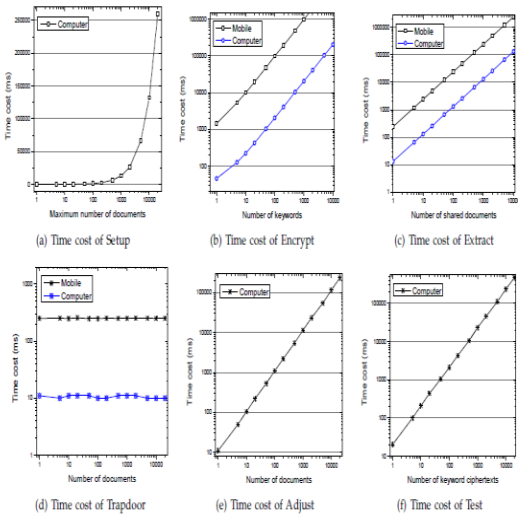


Fig. 4. Time cost of KASE algorithms.

this is mathematical operation in kase. Trapdoor is the as soon as multiplication in g, so that the key-word search may be finished efficaciously in both cellular gadgets and laptop. In comparison with other schemes, there is a considerable development in our scheme. 5) The execution time of kase. Regulate is linear within the range of files. In reality, it is able to be stepped forward within the sensible utility, and the information is proven in phase 6.4. 6) The execution time of kase. Take a look at is linear in the number of key-word cipher-texts. In reality, the mathematical operation is in kase. Take a look at is twice as much because the pairing computations. While the number grows up to 20000, it will take 467 seconds.

Evaluation of the Group Data Sharing System: Considering that the system's performances most critically depend on the KASE algorithms, we consider employing caching techniques in the group data sharing system to further improve the efficiency of the keyword search procedure. After receiving an aggregate trapdoor, the cloud server will run KASE. Adjust and KASE. Test to finish the keyword search.

7.CONCLUSION:

Thinking about the sensible problem of privacy preserving facts sharing system based on public cloud storage which calls for a information proprietor to distribute a massive range of keys to users to permit them to get entry to his/her files, we for the first time advocate the idea of key-mixture searchable encryption (KASE) and assemble a concrete KASE scheme. Each analysis and assessment effects verify that our paintings can provide an effective technique to building practical data sharing device based on public cloud garage. In a KASE scheme, the owner only desires to distribute an unmarried key to a person while sharing masses of documents with the consumer, and the user simplest wishes to post a unmarried trapdoor while he queries over all documents shared by way of the same owner. However, if a user desires to question over documents shared by way of more than one owner, he ought to generate a couple of trapdoors to the cloud. A way to reduce the number of trapdoors

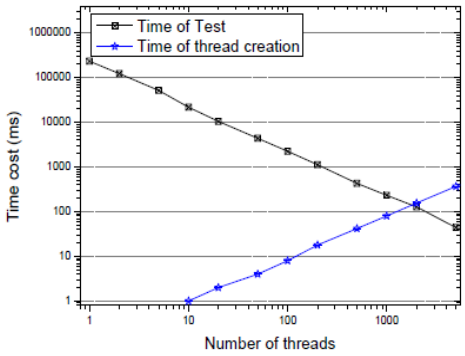


Fig. 5. Time cost of keyword search.

beneath multi-proprietors setting is a destiny painting. Moreover, federated clouds have attracted plenty of interest in recent times, but our KASE cannot be implemented on this case immediately. It's also a destiny work to provide the solution for KASE within the case of federated clouds.

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 Cloud Computing", Proc. ACM Symp. Information, Computer and Comm. Security, pp. 282-292, 2010.

[3] 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.

[4] 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.

[5] X. Song, D.Wagner, A. Perrig. "Practical techniques for searches on encrypted data", IEEE Symposium on Security and Privacy, IEEE Press, pp. 44C55, 2000.

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

[7] 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.

[8] J. Li, Q. Wang, C. Wang. "Fuzzy keyword search over encrypted data in cloud computing", Proc. IEEE INFOCOM, pp. 1-5, 2010.

Mr. B.TIRUPATHI KUMAR was born in India in the year of 1984. He received B.Tech degree in the year of 2007 & M.Tech PG in the year of 2010 from K.U. He was expert in Data Mining, Web Data Mining, Web Technologies subjects. He is currently working as An Associate Professor in CSE Department in Malla Reddy Institute of Technology, Maisammaguda, Dhulapally Post, Secunderabad, and Telengana State, India.

Mail id : tirupathi.kumar@gmail.com

Ms. SUMA MALAKA was born in India . She pursuing M.Tech degree in Computer Science & Engineering in CSE Department in Malla Reddy Institute of Technology, Maisammaguda, Dhulapally Post, Secunderabad and Telengana State, India.

Mail id: sumamalaka454@gmail.com