ISSN No: - 2456 - 2165

Secure Key Aggregate Searchable Encryption (KASE) and Efficient Data Sharing in Cloud

M/s. Pratiksha Gadekar
Department of Computer Engineering
SPCOE
Otur (Pune),Maharashtra
gpratiksha92@gmail.com

Prof. Ugale Pradip
Department of Computer Engineering
SPCOE
Otur (Pune),Maharashtra
ugalepr@gmail.com

***Abstract*— over casual information to be carried away in the cloud there might excessive vast security worries with individualize clients through open distributed storage due to the capacity of specifically scrambled information sharing. In the effective encryption keys management, to procreate such decode plans, falseness is a key test. For various records, with any gathering of clients requests for diverse encryption keys, the financed adaptability of sharing utilized by gathering of that documents. In any case, for both pursuit and encryption, to clients countless the necessity of disseminating safely and to safely store the got keys likewise suggests those clients will have and to perform seek over the information which is common submit to the cloud all together a just as vast number of countersign trapdoors. The methodology is not probable for the inferred demand for secure stockpiling, multifaceted nature and resemblance unmistakably renders. By idea instantiating through the scheme of KASE and proposing the conception of key-total searchable encryption (KASE), we address this down to terrene issue. In the writing this issue was generally dis-regarded, in which countless part to a client, there necessities just to correct a solitary key, knowledge proprietor, and the suitor requirements for questioning the reciprocal archives for presenting an unattended trapdoor to the cloud.**

Keywords- *Key Searchable encryption, data sharing, cloud computing, data privacy.*

I. INTRODUCTION

Over the Internet for giving advantage, Ubiquitous and for a lot of shared information on-interest gets to, there has developed as a promising preparation by spread storage. Taking into account distributed storage through informal organization applications, individual information, for example, photographs and recordings are imparted by a large number of users to their mate once a day. Because of its various less cost, better especial use and more prominent knack, by distributed storage the business clients are likewise being repelled in. In case of, worried of clients about coincidental information spills in the cloud additionally progressively by means of distributed storage while getting a charge out of the accommodation of sharing information.

There can as a rule lead to genuine breaks of individual protection or business mysteries because of such information spills. Over potential information spills in distributed storage to address clients' worries, all the information scrambled before transferring them to the cloud is the regular methodology for the information proprietor, such that later by the individuals who have the decoding keys, the encoded information might be recovered and unscrambled which is known as the cryptographic distributed storage. In any case, for clients to pursuit and after that specifically recover just the information containing given watchwords, the encryption of information makes it testing. To utilize a searchable encryption (SE) conspire, a typical arrangement is in which potential catchphrases are scramble by information proprietor and together with encoded information transfer them to the cloud, such that, for performing seek over the scrambled information, the client will send the relating watchword trapdoor to the cloud for recovering information coordinating a catchphrase. The fundamental security necessity of a distributed storage can accomplish by the distributed storage in spite of the fact that joining a searchable encryption plan with cryptographic, for extensive scale applications, executing such a framework including a great many clients and by functional issues including billions of documents might in any case be blocked the effective administration of encryption keys, which, are generally overlooked in the writing to the best of our insight. Most importantly, for various records which the requirement for specifically imparting scrambled information to various clients, there as a rule requests diverse encryption keys to be utilized. Such a substantial number of keys must be safely put away and oversaw and in addition circulated to clients by means of secure channels, by the clients in their gadgets. What's more, by the clients there must be created countless and keeping in mind the end goal to perform a catchphrase seek over numerous records submitted to the cloud. Such a framework wasteful and unfeasible the inferred requirement for secure computational multifaceted nature, capacity and correspondence might render. In this paper by proposing the novel scheme of KASE, we address this test and through a solid KASE plan instantiating the scheme. To any distributed storage there applies the proposed approach KASE plan which underpins the usefulness of searchable gathering information sharing, which implies that, any client might specifically impart the gathering of chose

documents to a chose clients gathering, to perform catchphrase look over the previous while permitting the last mentioned. For proficient key administration the primary prerequisites are twofold to support searchable gathering information sharing. To start with, for sharing any number of records, an information proprietor just needs to circulate a solitary total key to a client. Second, over any number of shared records for performing watchword look, there just needs to present the client to the cloud a solitary total trapdoor. To the best of our insight, in this paper the KASE plan proposed can fulfill both prerequisites.

Contributions: More particularly, takes after are our primary commitments.

1) For era of key, setup of security parameter, key extraction, encryption, era of trapdoor, modification of trapdoor, and testing of trapdoor, we first characterize a general KASE system which make seven polynomial algorithms. For outlining a substantial KASE plan we then portray the prerequisites of both utilitarian and additionally security.

2) After planning plan of a solid KASE, we then instantiate the KASE structure. For the seven algorithms in the wake of giving point by point developments, we build up its security through nitty gritty examination and investigate the proficiency of the plan.

3) Based on the proposed KASE plan, in building a real gath-ering information sharing framework we talk about different commonsense issues and assess its execution.

II. LITERATURE REVIEW

There is a rich literature survey on searchable encryption, including SSE schemes [5] [8] and PEKS schemes. In contrast to those existing, in the context of cloud storage, keyword search under the multitenancy setting is a very common scenario. In such a scenario, the data owner would like to share a document with a group of authorized users, and every user who has the access right can provide a trapdoor to perform the keyword search over the shared document, namely, the multi-user searchable encryption (MUSE) scenario. Some recent work [6] focus to such a MUSE scenario, although they all adopt single-key combined with access control to achieve the goal. In [6], MUSE schemes are constructed by sharing the documents searchable encryption key with all users who can access it, and broadcast encryption is used to achieve coarse-grained access control. In attribute based encryption is applied to get fine-grained access control aware keyword search. As a result, in MUSE, the main problem is how to control which users can access which documents, whereas how to reduce the no. of shared keys and trapdoors is not considered. Key aggregate searchable encryption can provide the solution for the latter, and it can make MUSE more efficient and practical. In the case of a multi user application, considering that the no. of trapdoors is proportional to the number of documents to search over (if the user provides to the server a keyword trapdoor under every 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 facilitate a single keyword trapdoor to the server, but still allows the server to search for that trapdoors keyword in documents encrypted with different keys. This might sound similar to the goal of KASE, but these are in fact two completely different concepts. The goal of KASE is to delegate the keyword search right to any user by distributing the aggregate key to her/him in a group data sharing system, whereas the goal of MKSE is to ensure the cloud server can perform keyword search with one trapdoor over different documents owing to a user. This approach of MKSE inspires us to focus on the problem of keyword search over a group of shared documents from the same user in the multiuser applications, and the adjust process in MKSE also facilitates a general approach to perform keyword search over a group of documents with only one trapdoor. However, the adjust process of MKSE needs a delta generated from both users key and SE key of the document, so it does not directly apply to the design of a concrete KASE scheme.

Data sharing systems based on cloud storage have attracted much attention recently [1][4]. In particular, Chu et al. [4] con-sider how to reduce the number of distributed data encryption keys. To share several documents with various encryption keys with the same user, the data owner will require distributing all such keys to him/her in a traditional approach which is usually impractical. Aiming at this challenge, a key aggregate Encryption scheme for data sharing is proposed to produce an aggregate key for the user to decrypt all the documents. To allow a set of documents encrypted by different keys to be decrypted with a single aggregate key, user could encrypt message not only under a public-key, but also under the identifier of each document. The construction is inspired by the broadcast encryption scheme. In this construction, the data owner can be regarded as the broadcaster, who has public key pk and master-secret key msk; each document with identifier i can be regarded as a receiver listening to the broadcast channel, and public information used in decryption is designed to be relevant to both the owners msk and the encryption key, the message encryption process is similar to data encryption using symmetric encryption in BE, but the key aggregation and data decryption can be simply regarded as the further mathematical transformation of BE. Encrypt algorithm and BE. Decryption algorithm respectively.

A. Multi-user Searchable Encryption

Counting PEKS and additionally SSE plans, on searchable encryption there is a rich writing. The watchword seek under the multi-tenure setting is a more basic situation in the connec-tion of distributed storage as opposed to those current work. In such a situation, to impart a record to a gathering of approved clients the information proprietor might want, and over the ”multi-client searchable encryption” (MUSE) situation, every

client can give a trapdoor who has the entrance right to perform the watchword look.

To such a MUSE situation some late work center, in spite of the fact that to accomplish the objective with access control they all receive single-key consolidated. With all clients by sharing the record's searchable encryption key who can get to it, MUSE plans are developed, and to accomplish coarse-grained access control telecast encryption is utilized. To accomplish fine-grained access control mindful catchphrase seek characteristic based encryption (ABE) is connected. Sub-sequently, in MUSE, how to control which clients can get to which documents is primary issue, though There is not considered how to minimize trapdoors and shared the quantity of keys. The answer for the last can give by key total searchable encryption, and it can be make more pragmatic and effective for MUSE.

B. Multi-Key Searchable Encryption

On account of use which has a multi-client, to look over considering that there is relative the quantity of trapdoors to the quantity of archives, the idea of multi-key searchable encryption (MKSE) was presented by Popa. In 2013 he advances the primary plausible plan. For giving a solitary

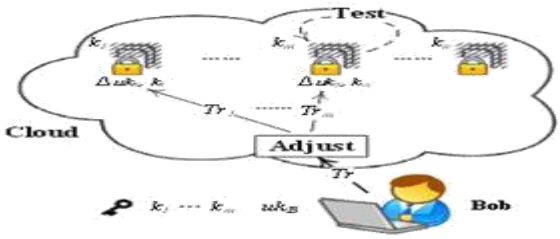

Fig. 1. Multi-Key Searchable Encryption

Catchphrase trapdoor to the server by a client permits by MKSE, yet in records encoded with various keys, to hunt down that trapdoor's watchword still permits the server. To the objective of KASE this may sound fundamentally the same, however these are indeed two unique ideas totally. From the same client in the multi user applications over a gathering of shared records, there spotlight on the issue of catchphrase inquiry by this methodology of MKSE which rouses us, and to perform watchword seek over a gathering of documents with one and only trapdoor a general approach additionally gives by the change process in MKSE. In any case, the MKSE's change handle needs. From both client's vital and SE key of the archive a delta created, so to the outline of a solid KASE plan it doesn't specifically apply.

III. IMPLEMENTATION DETAILS

The implementation details given as implementing modules as mention below:

A. USER MODULE:

In user module, users make his registration with the system. For registration he fill all the required details related to him in the system. After that he login with the system by entering the user name and password. If he wants to send the file to the end user,then he select the file for generating the aggregate key for that file. Then he encrypts this selected file. After file get encrypted successfully, user upload it to the cloud. For every file in the cloud, there is an aggregate key. Every files on the cloud are encrypted. User send the aggregate key to the receiver if he get the request from receiver about the aggregate key.

B. CRYPTOGRAPHIC MODULE:

The files selected by user are get encrypted. Files are encrypted using aggregate key. In cryptographic module, Files are encrypted using the aggregate keys and these encrypted files are uploaded on the cloud. This file decrypted only same aggregate key at receiver side.

C. EXTRACTION MODULE:

In extraction module, user download the file which uploaded by sender from cloud. For downloading file, user search it by its name or keywords which are provided. Then, receiver downloads it. For decrypting it receiver must have the aggregate key which was used for encrypting this file. Then, receiver sends the request for aggregate key to the sender. After getting the aggregate key, receiver decrypt this file and get the original data from it.

D. Software Requirements Specifications

Hardware Requirements

- Intel Pentium IV or above
- Minimum RAM 512 MB
- Minimum 500 MB Hard Disk

Software Requirements

- OS Requirements: Windows 7 onwards
- Netbeans 7.3.1
- JAVA JDK 1.7 and above
- Mysql Server 5.5
- Apache Tomcat 6.0 and above.

IV. MATHEMATICAL MODEL

- Input: File request
- Output: Aggregate key for requested files

  Success Conditions: File decrypted

- Failure Conditions: File not decrypted

Mathematical Formulation

Set S= I,P,R,O
Where I=set of all inputs given to the system.(user name, password, encryption key)
P= Set of process to generate the output.
R= Set of rules.
O= Set of Output.
I= α,β,ɣ,δ
α=login(id,password)

- Enter id and password
- Validate with database:
  Select * from user where userid='id' and passsword='password'
- if (userid==id and password==password) then
  login successful
- else login
  unsuccessful

β=login result

- if (userid==id and password==password) then
- login successful
- else login
  unsuccessful

ɣ=pk and mk
δ= Pf
Where pk=public key, mk= master key and Pf = file to be encrypted.
P= p0, p1, p2, p3
P0= Login to system
P1= Encrypt(Pf)
P2= Generate aggregate key
P3=Decrypt(Ef)

R= R0, R1
R0= Verify(id,password).
R1= Activation Status.

O=O1,O2,O3
O1= Ef encrypted file
O2= Ak aggregate key
pair (pk,msk)
where pk-public key and
msk-master key
O3= Df Decrypted file

V. ALGORITHM

The proposed system uses AES algorithm for encryption of the files which is available in javax.security packages.

The algorithm for Key generation and key extraction are based on string concatenation and string splitting, where the keys are considered to be strings and the operations are performed over these strings.

VI. RESULT ANALYSIS

A. Result table

Result Description	Expected Output	Actual Output
The proposed system must make minimum use for storing the keys over cloud. If the users requests Number of files for downloading, the key should be the aggregate key of all the files in such a way that the same key can be used for Decrypting the requested all files. This will also reduce the key com-Munication issue.	If multiple files are requested and if multiple keys are to be sent, key sending overhead Increases. The files being requested should only be decrypted by the received aggregate key. If the aggregate key is for file ids, f1,f3 , f5 only these files should be decrypted adn not f2,f4 etc.	The proposed system requires only one key for all requested files for decrypting the files, without any requirement of additional key.

B. Efficiency Calculation and Parameter Evaluation

1 Efficiency Calculation:

- In existing system, the files being requested require different keys for decrypting the files and thereby reduces the efficiency of the sys-tem. So if n files are being requested, the key required for decryting the n files will be:
  K_A = Extract(ff_1; f_2; :::; f_ng; K_M)
- The proposed system computes the aggregate key from the individual keys and the master key being formed based on file id and the in-dividual key combination. The existing system efficiency will be as follows:
  K_D = ff1; K1g; ff2; K2g; ff3; K3g

2 **Parameter Evaluation:** The proposed system can be evaluated based on the number of keys being required for decrypting the requested files and the efficiency of key storage over server.

C. Outcome and Success Definition

As per the Result analysis and efficiency evaluation, the Proposed system successfully achieves its goal of reducing the keys storage mechanism and reducing key communication

overhead by aggregating the keys of the requested files and creating only 1 key for the requested files.

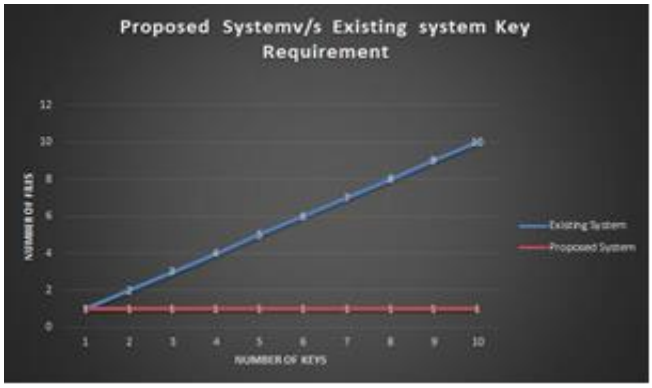

Fig. 2. Evaluation of Proposed system v/s Existing system

VII. CONCLUSION

The proposed paper shows that how the cloud storage utilization for key storage and data sharing over clouds can be efficiently managed. the key aggregation concept reduces the space required to store the keys and thereby make the space utilization optimal. Thus the proposed system can pro-vide maximum throughput as per the implementation of key aggregation over cloud is concerned.

ACKNOWLEDGMENT

All faith and honor to the GOD for his grace and inspiration. I would like to thank all my Friends and Family members they were always been there to support me. I sincerely thanks to my Department Head, PG coordinator and all other staff members to give me the guidelines for this paper.

REFERENCES

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

[2] X.Liang, R.Lu, X.Lin, 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] Y.Zhang, X.Liu, B.Wang, and J.Yan. “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, et al. “Key-Aggregate Cryptosystem for W.Tzeng, 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] R.Curtmola, J.Garay, S.Kamara, R. Ostrovsky. “Searchable symmetric encryption: improved definitions and efficient constructions”, In: Proceedings of the 13th ACM conference on 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] C.Papamanthou, S.Kamara, T.Roeder. “Dynamic searchable symmetric encryption”, 2012 ACM conference on Computer and communications security (CCS), ACM, pp. 965-976, 2012.