International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 11 | Nov -2016 www.irjet.net p-ISSN: 2395-0072

Efficient and secure data storage on cloud using KASE (Key Aggregation Searchable Encryption)

Maithili Khuspe, Suraj Motegaonkar, Prajakta Divekar, Bhakti Chavare

Student ,Department of Computer Engineering ,SKN Sinhgad Institute of Technology &Science, Lonavala ,SPPU, Pune, Maharashtra,India.

Abstract -*These These Days cloud storage has become an emerging solution for storing large amount of data. It becomes easy for user to access, manipulate, store or retrieve data. Cloud storage is on convenient, on demand and very easy manipulation of data provided. Thousands of users shared their data through some applications which is stored in cloud storage. Cryptographic approach is used for data sharing is an important functionality in cloud storage. By the common approach data owner encrypts all the data before uploading it to the cloud so that later the encrypted data may be retrieved and decrypted by those who have the decryption keys. In traditional system there are number of encryption keys for sharing any group of selected documents with any group of users demands. efficient management of encryption keys is a key challenge here. This also implies the need of securely distributing a large number of keys to users for encryption. Also user will have to securely store the received key and submit an equally large number of keyword trapdoors to cloud for searching over the shared data. The practical problem for document owners is to distribute a large number of keys and equally large number of keyword trapdoors to users to enable them to access users documents. To avoid this problem we need a mechanism that provides good and efficient security system for the same a new encryption method has been devised called KASE. This mechanism provides secure way of distributing a single aggregate key among users for encryption , searching ,sharing documents among multiple users. KASE uses single aggregate trapdoor to the cloud for querying the shared documents in the cloud through this we can achieve security and efficiency.*

Key Words: **searchable encryption, data sharing, cloud storage, data privacy, key aggregate encryption.**

1. INTRODUCTION

With developing reliance on web for globalization, cost for owning IT Infrastructure, assets have expanded. Distributed computing is another idea that for the most part is an on request renting administration for web applications and IT assets. Today, various clients are chiefly sharing countless sorts of archives, which are thought to be under different classes like photographs, recordings and reports by means of different long range interpersonal communication construct applications with respect to consistent schedule. There are tremendous advantages of utilizing distributed storage like lower cost, more prominent readiness and better asset use has include more fascination from bounty number of business clients toward utilizing the distributed storage.

1.1 Three Layer Security

This project overcome privacy preserving system problem .in existing system data owner send each key for each file ,it create problem on user side for maintaining number of key for receiving and decrypting file in our kase project sender only send one key i.e aggregate key to the user for getting file This project provide 3 layer security to file

1. User uploaded file on cloud in encrypted form using privatekey.
2. Aggregate key is also encrypted with public key.

3.One more concept trapdoor .For receiving file from cloud user need to submit trapdoor to the cloud

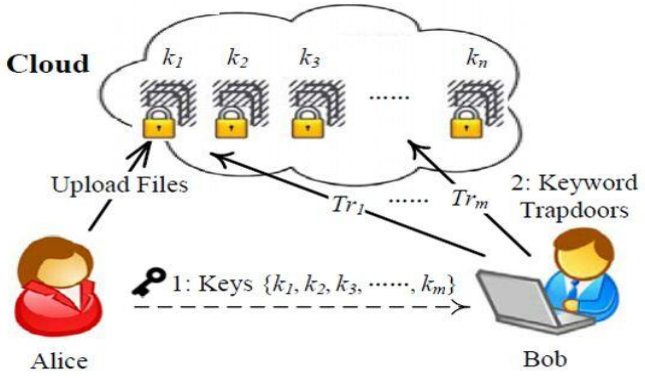
2. The KASE Framework

The KASE structure is made out of seven calculations. In particular, to set up the plan, the cloud server would create open parameters of the framework through the Setup calculation, and these open parameters can be reused by various information proprietors to share their documents. For every information proprietor, he/she ought to create an open/ace mystery key combine through the Keygen calculation. Catchphrases of every report can be scrambled by means of the Encrypt calculation with the one of a kind searchable encryption key. At that point, the information proprietor can utilize the ace mystery key to produce a total searchable encryption key for a gathering of chose reports by means of the Extract calculation. The total key can be dispersed safely (e.g., by means of secure messages on the other hand secure gadgets) to approved clients who need to get to those reports. After that, as appeared in Fig.2, an approved client can create a watchword trapdoor through the Trapdoor calculation utilizing this total key, also, present the trapdoor to the cloud. In the wake of accepting the trapdoor, to play out the catchphrase look over the indicated set of reports, the cloud server will run the Adjust algorithm to generate the right trapdoor for each document, and then

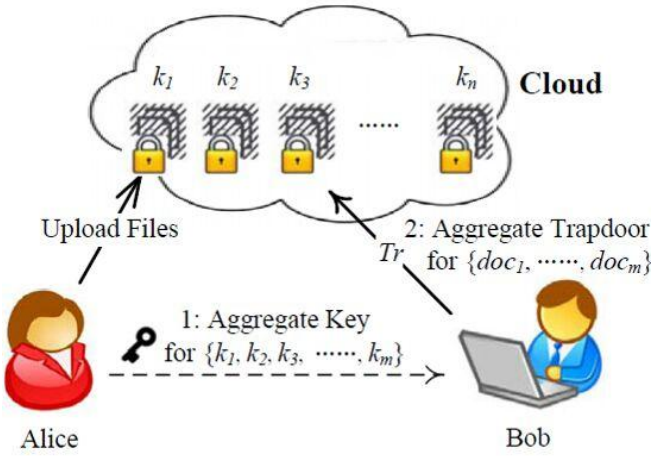
run the Test algorithm to test whether the document contains the keyword. This framework is summarized in the following.

Setup(1 λ , n): this calculation is controlled by the cloud benefit supplier to set up the plan. On contribution of a security parameter 1 λ and the most extreme conceivable number n of records which has a place with an information proprietor, it yields the general population framework parameter params.

• **Keygen**: this calculation is controlled by the information proprietor to create an irregular key combine (pk,msk).



(a) Traditional approach



(b) Key-Aggregate Searchable Encryption

• **Encrypt(pk,** i): this calculation is controlled by the information proprietor to scramble the i-th archive and create its catchphrases' ciphertexts. For every archive,this calculation will make a delta Δi for its searchable encryption key ki . On contribution of the proprietor's open key pk and the record file i, this calculation yields information ciphertext and catchphrase ciphertexts 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 msk and a set S which contains the indices of documents, then outputs the aggregate key kagg.

• **Trapdoor(kagg, w):** this calculation is controlled by the client who has the total key to play out a look. It takes as information the total searchable encryption key kagg and a catchphrase w, then yields just a single trapdoor T r.

• **Adjust(params, i, S, T r):** this calculation is run by cloud server to change the total trapdoor to create the privilege trapdoor for each extraordinary record. It takes as information the framework open parameters params, the set S of records' lists, the list i of target record and the total trapdoor T r, then yields each trapdoor T ri for the i-th target archive in S.

• **Test(T ri , i):** this calculation is controlled by the cloud server to perform watchword look over a scrambled archive. It takes as info the trapdoor T ri and the archive file i, then yields genuine on the other hand false to mean whether the archive doci contains the watchword w.

3. Requirements for Designing KASE Schemes

The KASE structure presented in the past segment gives general direction to planning a KASE plot. Be that as it may, a substantial KASE conspire must additionally fulfill a few utilitarian and security prerequisites, as expressed in the accompanying. A KASE plan ought to fulfill three utilitarian prerequisites as takes after.

• Compactness. This prerequisite requests a KASE plan to guarantee the measure of the total key to be autonomous of the quantity of documents to be shared. Formally, for an arrangement of keys {ki}i∈S, it requires that kagg←Extract(msk, S). The most effective method to total the arrangement of keys into a solitary key without nullifying later strides is a key test in outlining KASE plans.

• **Searchability.** This prerequisite is vital to all KASE plans since it empowers clients to produce coveted trapdoors for

any given catchphrase for looking scrambled records. In another word, lessening the quantity of keys ought to safeguard the look ability. Formally, for every record containing the watchword w with list i ∈ S, the searchability requires that if (T r=Trapdoor(kagg, w) and T ri←Adjust(params, i, S, T r)), then Test(T ri , i)=true.

• **Delegation.** The primary objective of KASE is to designate the catchphrase look appropriate to a client through an total key. To guarantee any client with the appointed key can perform watchword seek, this prerequisite requires that the contributions of the alteration calculation must not be open, i.e., these sources of info ought not depend on any client's private data. This is the second enter challenge in outlining KASE plans.

Moreover, any KASE plan ought to likewise fulfill two security prerequisites as takes after.

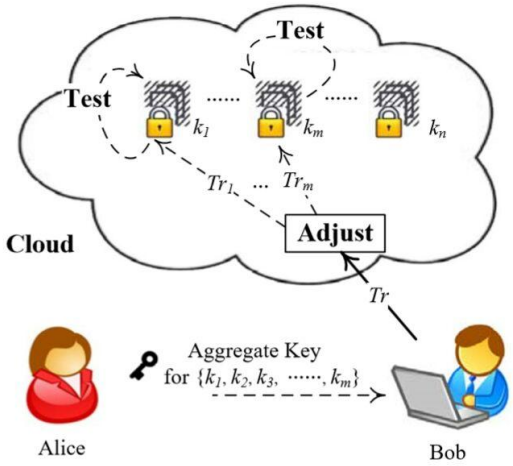
• **Controlled looking**. Implying that the assailants can't hunt down a subjective word without the information proprietor's approval. That is, the assailant can't perform catchphrase look over the records which are not applicable to the known total key, and he/she can't create new total searchable encryption keys for other set of archives from the known keys.

• **Query protection**. Implying that the assailants can't decide the catchphrase utilized as a part of a question, separated from the data that can be gained by means of perception and the data got from it. That is, the client may ask an untrusted cloud server to hunt down a touchy word without uncovering the word to the server.

4. Related Work

4.1.Multi-user Searchable Encryption

There is a rich writing on searchable encryption, counting SSE plans and PEKS plans. As opposed to those current work, in the setting of distributed storage, catchphrase seek under the multi-occupancy setting is a more regular situation. In such a situation, the information proprietor might want to share an archive with a gathering of approved clients, and every client who has the get to right can give a trapdoor to play out the watchword seek over the shared archive, in particular, the "multi-client searchable encryption" (MUSE) situation. MUSE situation, in spite of the fact that they all embrace single-key consolidated with get to control to accomplish the objective.MUSE plans are built by sharing the report's searchable encryption key with all clients who can get to it, and communicate encryption is utilized to accomplish coarse-grained get to control. Property based encryption (ABE) is connected to accomplish fine-grained get to control mindful watchword look. Thus, in MUSE, the principle issue is step by step instructions to control which clients can get to which archives, though how to decrease the quantity of shared keys and trapdoors is not considered. Key total searchable encryption can give the answer for the last mentioned, and it can make MUSE more effective and commonsense.



4.2. Multi-Key Searchable Encryption

On account of a multi-client application, considering that the quantity of trapdoors is corresponding to the number of archives to look over (if the client gives to the server a watchword trapdoor under every key with which a coordinating archive may be encoded), Popa [28] firstly presents the idea of multi-key searchable encryption (MKSE) and puts forward the main attainable plan in 2013. MKSE permits a client to give a solitary watchword trapdoor to the server, yet at the same time permits the server to seek for that trapdoor's watchword in records encoded with various keys. This may sound fundamentally the same as the objective of KASE, yet these are in reality two totally distinctive ideas. The objective of KASE is to appoint the catchphrase seek ideal to any client by dispersing the total key to him/her in a gathering information sharing framework, while the objective of MKSE is to guarantee the cloud server can perform watchword look with one trapdoor over various records inferable from a client. All the more particularly, indicate by uki the key of client i. Assume a client, say Bob (with key ukB), has m scrambled records on the cloud server, and each is encoded under a key kj for j

= {1, ·, m}. To permit the cloud server to alter the trapdoor for each archive with file j, Bob stores on the cloud server an open data called delta (signified as ΔukB, kj) which is important to both ukB and kj . As appeared in Fig. when Bob

needs to look for a word w over all the reports, he will utilize ukB to process a trapdoor for the word w and submit it to the cloud server. The cloud server can utilize ΔukB,kj to change over a watchword trapdoor under key ukB to a watchword trapdoor under kj ; this procedure is called conform. In such a way, the cloud server can acquire trapdoors for word w under k1, ⋯, km while just getting one trapdoor from Bob,and after that play out a customary single-key inquiry with the new trapdoors.

This approach of MKSE rouses us to concentrate on the issue of watchword hunt over a gathering of shared reports from a similar client in the multiuser applications, and the conform procedure in MKSE likewise gives a general way to deal with perform watchword seek over a gathering of reports with just a single trapdoor. Be that as it may, the conform procedure of MKSE needs a delta created from both client's critical and SE key of the record, so it doesn't straightforwardly apply to the outline of a solid KASE conspire.

4.3.Key-total Encryption for Data Sharing

Specifically, consider how to lessen the number of disseminated information encryption keys. To share a few reports with various encryption keys with the same client, the information proprietor should disperse all such keys to him/her in a conventional approaches which is normally unfeasible. Going for this test, a key aggregate Encryption (KAE) conspire for information sharing is proposed to produce a total key for the client to unscramble every one of the reports. To permit an arrangement of archives encoded by various keys to be unscrambled with a solitary total key, client could encode a message not just under an open key, in any case, likewise under the identifier of every archive. The development is propelled by the communicate encryption conspire. In this development, the information proprietor can be viewed as the telecaster, who has open key pk and ace mystery key msk; every record with identifier i can be viewed as a recipient listening to the communicate channel, and an open data utilized in unscrambling is intended to be applicable to both the proprietor's msk and the encryption key; the message encryption process is like information encryption utilizing symmetric encryption in BE, yet the key accumulation also, information decoding can be basically viewed as the promote scientific change of BE.Encrypt calculation and BE.Decrypt calculation individually.The plan permits effectively appointing the unscrambling rights to different clients, and is the principle motivation of our concentrate, yet it doesn't bolster any look over the encoded information. In the cloud environment, to accomplish the objective of security saving information sharing, watchword pursuit is a vital necessity. Luckily, the KAE gives bits of knowledge to the plan of a KASE plot, despite the fact that our plan will require a more intricate numerical change to bolster catchphrase ciphertext encryption, trapdoor era and watchword coordinating.

5. Advantages of using KASE

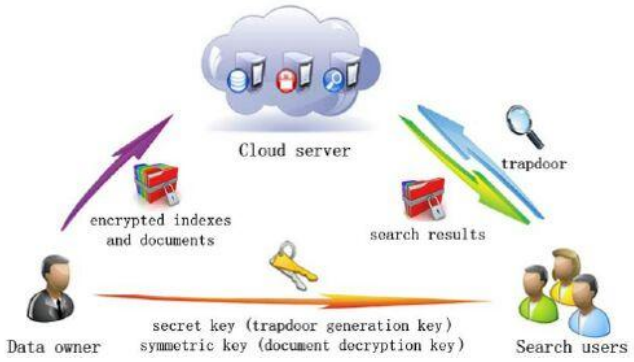
1) Effectiveness

Regarding proficiency, our plan plainly accomplishes consistent size watchword ciphertext, trapdoor and total keys.

2)Security

To dissect the security of our plan, and in specific demonstrate that the plan fulfills the security prerequisites given in Section 3.3, we accept that people in general cloud is "straightforward yet inquisitive". That is, the cloud server will just give genuine administrations as indicated by pre-characterized plans, in spite of the fact that it might attempt to recuperate mystery data in view of its learning. We likewise expect that the approved clients may attempt to get to information either inside or out of the extents of their benefits. Also, correspondence channels including people in general cloud are thought to be unreliable. In view of the above contemplations, we will demonstrate the security of our plan as far as controlled seeking and question protection.

6. PROPOSED WORK



Traditional key cryptosystems lack the enhanced security techniques as the keys are generated by the existing random key generation

Proposed system said to have aggregate key cryptosystem in which key generated by means of various derivations of cipher text class properties of data and its associated keys

7. CONCLUSIONS

To share information adaptably is essential thing in distributed computing .Clients want to transfer there information on cloud and among various clients .Outsourcing of information to server may prompt to release the private information of client to everybody. Encryption is a one arrangement which gives to impart chose information to fancied applicant. Sharing of decoding keys in secure way assumes essential part .Open key crypto systems gives appointment of mystery keys to various cipher text classes in distributed storage .The delegate gets safely a total key of steady size It is required to keep enough number of figure writings classes as they increment quick and the cipher text classes are limited that is the constraint

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. Cryptographic techniques,Re-Encryption tecnique

[2] R. Lu, X. Lin, X. Liang, and X. Shen,"Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing", Computer and Comm.Security,pp. 282-292, 2010.12 Cloud Computing, Data Forensics, Secure Provenance

[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 Distribute Systems, 2013, 24(6): 1182-1191. Cloud computing, Data sharing, Multiowner, access control, dynamic groups

[4] C. Chu, S. Chow, W. Tzeng, et al. "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage", IEEE Transactions on Parallel and DistributedSystems, 2014, 25(2): 468-477. Cloud storage, data sharing, key-aggregate encryption

[5] J. Li, K. Kim. "Hidden attribute-based signatures without anonymity revocation",Information Sciences, 180(9): 1681-1689, Elsevier, 2010. Signature, Attribute-based, Anonymity, Computational Diffie–Hellman assumption

[6] Baojiang Cui, Zheli Liu and LingyuvWang "Key-Aggregation Searchable Encryption (KASE) for Group Data Sharing via Cloud Storage".