International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 23 Issue 9 – DECEMBER 2016.

KEY AGGREGATE CRYPTOSYSTEM FOR EFFICIENT WAY DATA SHARING IN CLOUD ENVIRONMENT

M. Madhuri[#1] and K. Kishore Raju[*2]

[#] *M.Tech. Student, Information Technology, SRKR Engineering College, Bhimavaram India*

[*] *Assistant Professor, Information Technology, SRKR Engineering College, Bhimavaram India*

***Abstract*— An efficient cryptographic approach for data sharing among a group of users is an important functionality in cloud storage. To securely and efficiently sharing selective data with others in cloud storage, we incorporated new novel concept of Key Aggregate Searchable Encryption (KASE) through development of a concrete encryption framework scheme.**

In this scheme data owner only needs to generate and distribute a single aggregate key to data user for sharing large number of documents. User also only needs to submit a single aggregate trapdoor to the cloud server, so that query processing over the shared documents can be done.

***Index Terms*— Searchable encryption, data sharing, cloud storage, data privacy**

I. INTRODUCTION

Cloud storage is becoming more popular nowadays. In enterprise settings, we see the rise in demand for data outsourcing, which benefits in the field of corporate data and its management. It is also useful as a core technology for different online technologies for individual applications. Cloud computing is known as an alternative to traditional technology due to its better resource-sharing and low-maintenance capabilities. The main aim of cloud computing is to provide high performance energy of computing for various field like military and research organization for performing billions of computations at each second. It is also used in customer oriented areas like portfolios to transfer confidential information.

In cloud computing, the cloud service providers, like Amazon, are able to provide various services to users with the help of powerful data servers. Moving the local data management systems into cloud servers, users can take advantage of high-quality services and store important investments on their local infrastructures. However, while sharing data through cloud storage, users are simultaneously aware about the data leakages in the cloud. One of the most fundamental services delivered by cloud service providers is data storage. Consider a data application. There is a company which permits its staffs in the same group or department to store and share documents or files in the cloud. Using the cloud, the staffs can be fully released from the local data storage and maintenance. However, it also creates a significant risk to the confidentiality of those stored documents. Specifically, the cloud servers controlled by cloud providers are not fully believed by users while the documents stored in the cloud may be s confidential, such as business ideas. Identification of privacy is most important problem for wide development of cloud computing. Without the proof of identity privacy users are not ready to utilize the cloud services because they don't want to expose their real identity. To maintain data privacy, a basic idea is to encrypt files, and then upload the encrypted data into the cloud. In this paper, we demonstrate cryptographic scenarios for the problem of searching on encrypted data and provide result of security for the resulting crypto systems.

The storage in the cloud has materialized as a capable answer for suitable and on-demand accesses to huge amounts of information shared over the Internet. Business users are being paying attention by cloud storage due to its several benefits, including lower cost, better agility, and improved resource utilization. Everyday users are also sharing private data, such as photos and videos, with their friends through social network applications based on cloud. On the other hand, while benefiting from the expediency of sharing data through cloud storage, users are also gradually worried about accidental data reveal by the cloud. Such data revealing, will be performed by malicious opponent or a mischievous cloud operator, can habitually direct to severe violation of private data or confidential data regarding business.

In this paper, we propose the novel concept of key-aggregate searchable encryption (KASE), and instantiating the concept through a concrete KASE method. The proposed KASE scheme relates to any cloud storage that supports the searchable group data sharing feature, which means any user may prefer to distribute a group of files which are selective with a group of selected users, while permitting the final to carry out keyword search above the earlier. To maintain searchable group data sharing the main needs for efficient key management are double. Primarily, a data owner wants to allocate a single aggregate key (instead of a group of keys) to a user for sharing any number of files. Subsequent, the user needs to submit a single aggregate trapdoor to the

cloud for performing keyword search over any quantity of shared files. KASE scheme can assure both requests.

II. PROBLEM STATEMENT

Suppose that Client 1 uploads all her private pictures and videos on Dropbox, and she does not want to see her photos by everyone. Due to various data leakages in cloud there may be possibility that client 1 cannot feel satisfied by just relying on the privacy protection provided by Dropbox, so she encrypts all the pictures using her own keys before uploading. One day, Client 1's friend, say client 2, asks her to share her pictures taken during all these years which client 2 appeared in. client 1 then uses the share function of Dropbox, but the problem is how to delegate the decryption rights for these pictures to client 2. A possible option client 1 can choose is to securely send client 2 the secret keys included .Therefore there are two ways for her under the traditional encryption paradigm:

A. client 1 encrypts all files with a single encryption key and gives client 2 the corresponding secret key directly.

B. client 1encrypts files with distinct keys and sends client 2 the corresponding secret keys surely, the first technique is inadequate since all data which is not yet chosen may be also leaked to client 2. For the second method, there are practical concerns on efficiency. The number of keys is equivalent to the number of the shared photos, say, a thousand. Sending these secret keys requires a more secure channel, and storage of these keys requires expensive secure storage.

III. LITERATURE SURVEY

Baojiang Cui, Zheli Liu and Lingyu Wang, proposed key-aggregate searchable encryption to address the problem of privacy preserving in public cloud storage in which data owner required to distribute huge number of keys to other users to enable the access to their data. This scheme can be implies on any cloud system which supports the functionality of searchable group data sharing. In searchable group data sharing scheme, data owner can share group of files with the selected group of users. For that data owner needs to distribute single key to the user for sharing the group. And instead of group of trapdoors user only needs to submit single aggregate trapdoor to perform keyword searching over the group of any number of files.

S. Yu, C. Wang, K. Ren, and W. Lou, This system provides the solution for the problem of fine-grainedness, scalability, and data confidentiality of access control in cloud storage. To address these problems access policies are created based on data attributes. This paper proposed attribute-based encryption (ABE), proxy re-encryption, and lazy re-encryption techniques to achieve their goal.

R. Lu, X. Lin, X. Liang, and X. Shen, in this secure provenance *SP* scheme based on the bilinear pairings in cloud computing model. This scheme is used provide security and trusted evidences for data forensics in cloud computing. Provable security techniques are used to check the validity of the security. Trusted evidences for data forensics are provided by the secure provenance *SP* scheme.

X. Song, D.Wagner, A. Perrig, Paper proposed the proofs of security with the help proposed cryptographic scheme. It supports searching functionality without losing the confidentiality of the data. This technique is secure for encryption as it provides control searching over the data. This system handles the hidden searches as well as query isolation over the cloud data. This system also supports random-access decryption in which the length of each word also needs to be stored with the word. For Searching process encrypted Index is used when data size is large.

R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky, This paper stronger security technique that is Searchable Symmetric Encryption (SSE). In this technique user can store data on remote server and can access it privately. To extend the searching ability authors were also proposed multi-user SSE. In this system user least the data from large dataset, Single-database PIR used to retrieve data from a server containing unencrypted data. For the secure modifications new documents can be added to the previous document collection.

S. Kamara, C. Papamanthou, T. Roeder, This paper proposed stronger security technique that is Searchable Symmetric Encryption (SSE). In this technique user can store data on remote server and can access it privately. To extend the searching ability authors were also proposed multi-user SSE. SSE is adaptive security than chosen-keyword attacks (CKA2). This system uses inverted index approach. SSE has capability to describe leakage of a database which contains two tables over word and file identifiers.

D. Boneh, C. G, R. Ostrovsky, G. Persiano, In this author refers mechanism called *Public Key Encryption with keyword Search.* In this user sends the key to server to identify that all messages are containing some specific keyword without learning extra information. This system is based on IBE construction.This approach is for users who own their data and they wish to upload that data to a third-party database in which they may not trust. The system is based on a variant of the Computational Diffie-Hellman problem.

C. Dong, G. Russello, N. Dulay, In this system user has its own key which is used to encrypt and decrypt the data. Therefore it does not require any trusted server for accessing the data. This encryption system is based on proxy cryptography in which users share data via an un-trusted data storage server. In this server is hosted by a third party. Proxy cryptography is build upon the El Gamal encryption scheme. To securely encrypt keywords, keyword encryption scheme is also obtained by proxy encryption scheme. This scheme allows user revocation straightforwardly.

IV. PROPOSED METHODOLOGIES

A. MULTIUSER SEARCHABLE ENCRYPTION

A rich literature has been available on searchable encryption. Including SSE schemes and PEKS schemes. Contradictory to those existing work, in the control of cloud storage keyword search under the multi tenancy setting is a more common scenario. In such a scenario, the data owner would like to share a document with a group of authorized users and the user who has access right can provide a trapdoor to perform the keyword search over the shared document namely the "Multi user searchable encryption" scenario. Some recent work, focus to such a MUSE scenario. Though they all adopt single key combined with access control to achieve the goal. In, Muse scheme are constructed by sharing the document's searchable encryption key with all users who can access its and broadcast encrypting is used to achieve coarse joined access control. In, attributes based encryption is applied to achieve line grained access control aware keyword search. The main problem in MUSE has been to control users who can access documents, In order to reduce the number of shared keys and trapdoors are not considered. Key aggregate searchable encryption can provide the solution for the latter and it can make MUSE more efficient and practical.

B. MULTI KEY SEARCHABLE ENCRYPTIONS

In multi user application the number of trapdoors are proportional to the number of documents to search over different provides to the server a keyword trapdoor under each key which have to be matched and document can be encrypted firstly introduces the concept of multi key searchable encryption (MKSE) and places forward the first feasible scheme in 2013 MUSE enables 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. KASE is altogether different from MKSE. KASE delegates the keyword search right to any user by distributing the aggregate key to him/her in a group data sharing system while the goal of MKSE is to ensure the cloud server can perform keyword search with one trapdoor over different documents owing to a user.

C. KEY AGGREGATE ENCRYPTION FOR DATA SHARING

Data sharing system based on closed storage has much priority now days. In particular, how to reduce the number of distributed data encryption keys sharing different document with different encryption keys with the same user the data owner will need to distribute all such keys to him/her in a traditional approach which is usually impractical. In order to resolve this problem key aggregate encryption (KAE) scheme for data sharing is proposed to generate an aggregate key for the user to decrypt all the documents. A set of documents encrypted by different keys to be decrypted with a single aggregate key so that user can encrypt a message both under a public key and under the identifier of each documents The construction is inspired by the broadcast encryption key The data owner can be regarded as the broadcaster who has public key pk and master key MSK Every document with identifier's can be regarded as a receiver listening to the broadcast channel and a public information used in decryption is designed to be relevant to both the owner's MSK and the encryption key the message encryption process has resemblance with data encryption using symmetric encryption in BE but the key aggregation and data encryption are regarded as mathematical transformation of BR Encrypt algorithm and BE Decrypt algorithm respectively.

D. ALGORITHMS

• Setup(1^{λ}, n): this algorithm is run by the cloud serviceprovider 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.

(1^{λ}, n) → params

• Keygen: this algorithm is run by the data owner to generate a random key pair (pk,msk).

Key gen→ (pk,msk)

• Encrypt(pk, i): this algorithm is run by the data owner to encrypt the i-th document and generate its keywords' ciphertexts. For each document, this algorithm will create a delta Δ_i for its searchable encryption key k_i. On input of the owner's public key pk and the file index i, this algorithm outputs data ciphertext and keyword ciphertexts C_i.

E(pk,i) → C_i

• 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 k_{agg}.

Ex(msk,S) → K_{agg}

• 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.

Tr(K_{agg},w) → Tr

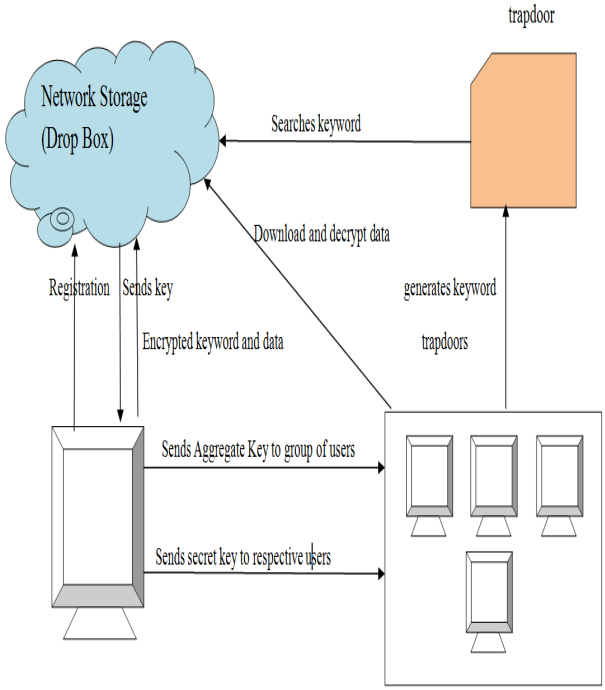
• Adjust(params, i, S, Tr): this algorithm is run by cloud server to adjust the aggregate trapdoor to generate the right trapdoor for each different document. It takes as input the system public parameters params, the set S of documents' indices, the index i of target document and the aggregate trapdoor Tr, then outputs each trapdoor Tr_i for the i-th target document in S.

Adjust(params,i,S,Tr) → Tr_i

• Test(Tr_i, i): this algorithm is run by the cloud server to perform keyword search over an encrypted document. It takes as input the trapdoor Tr_i and the document index i, then outputs true or false to denote whether the document doc_i contains the keyword w.

Test(Tr_i,i) → true/false

E. SYSTEM ARCITECTURE



V. MODULES

A. Data Owner

B. Network Storage

C. Trapdoor Generation

D. File User

1) MODULES DESCRIPTION:

a) Data Owner:

Data owner uploads selective documents to cloud server which are shared with the data user. Generally, each document is encrypted with a separate key. The key produced is send to the data user via a secure communication channel by the data owner. Data owner view the list of users available for accessing the files from drop box cloud. Than after performing all these actions, data user can perform searching over the shared documents by generating keyword trapdoors. If a match is obtained, the cloud server returns the original files which were shared by the data owner to corresponding requested data user.

b) Network Storage (Drop box):

With our solution, Alice can simply send Bob a single aggregate key via a secure e-mail. Bob can download the encrypted photos from Alice's Dropbox space and then use this aggregate key to decrypt these encrypted photos. In this Network Storage is untrusted third party server or dropbox.

c) Trapdoor generation

Trapdoor generation is run by the user who has the aggregate key to perform a search.

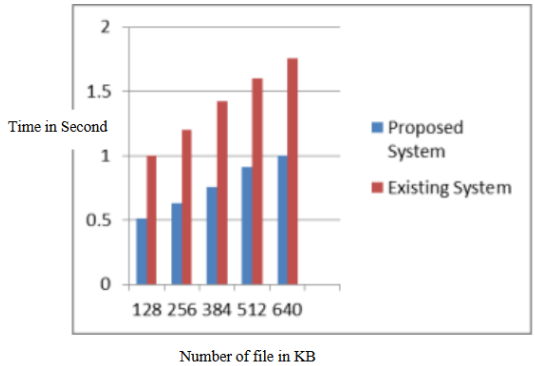
d) File User:

The generated keys can be passed to delegates securely (via secure e-mails or secure devices) finally; any user with the Trapdoor keyword generation process can decrypt any cipher text provided.

VI. EXPERIMENTAL RESULTS:

In this ,we provide a concept of key aggregate searchable encryption ,The data creator needs to distribute only a single aggregate key of his all document ,the user needs to submit a single trapdoor when he needs to download a document shared by the same creator. our proposed system provides greater time efficiency than the existing system.

Data in KB	Time complexity for decrypt [Proposed System] in second	Time complexity for decrypt [Existing System] in second
128	0.51	1
256	0.63	1.2
384	0.76	1.42
512	0.91	1.6
640	1	1.76



VII. CONCLUSION AND FUTURE WORK

Taking into consideration of the realistic problem of privacy preserving data sharing system based on public cloud storage which is need a data owner to allocate a large number of keys to users to permit them to access the documents, In this proposed concept of key-aggregate searchable encryption and construct a concrete KASE scheme. It can provide an efficient solution to building practical data sharing system based on

public cloud storage. In a KASE scheme, the owner needs to distribute a single key to a user when contributing a lot of documents with the user, and the user needs to submit a single trapdoor when they queries over all documents shared by the same owner. On the other hand, if a user wants to question over documents shared by multiple owners, that user must produce multiple trapdoors to the cloud. The future enhancement for this proposed work is to find out how to decrease the number of trapdoors under multi-owners setting by attaining the security.

REFERENCES

[1] S. Yu, C. Wang, K. Ran, 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. Sheen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing", Proc. ACM Sump. 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. 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] P. Van's. Sergei, JM. Doormen. "Computationally efficient searchable symmetric encryption", Secure Data Management, pp. 87-100, 2010.

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

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

[8] F. Zhao, T. Nishide, K. Sakurai. "Multi-User Keyword Search Scheme for Secure Data Sharing with Fine-Grained Access Control". Information Security and Cryptology, LNCS, pp. 406-418, 2012.

[9] 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.

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

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

[12] J.Li, X.F. Chen, M.Q. Li, J.W. Li, P. Lee, Wenjing Lou. "Secure Deduplication with Efficient and Reliable Convergent Key Management", IEEE Transactions on Parallel and Distributed Systems, 25(6): 1615-1625, 2014.

[13] Z. Liu, Z. Wang, X. Cheng, et al. "Multi-user Searchable Encryption with Coarser-Grained Access Control in Hybrid Cloud", Fourth International Conference on Emerging Intelligent Data and Web Technologies (EIDWT), IEEE, pp. 249-255, 2013.

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

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

[16] D. H. Phan, D. Pointcheval, S. F. Shahandashti, et al. "Adaptive CCA broadcast encryption with constant-size secret keys and ciphertexts", International journal of information security, 12(4): 251-265, 2013.

[17] L. B. Oliveira, D. F. Aranha, E. Morais, et al. "Tinytate: Computing the tate pairing in resource-constrained sensor nodes", IEEE Sixth IEEE International Symposium on Network Computing and Applications, pp. 318-323, 2007.