International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 3, Issue 2, February 2016

An Efficient Searchable Encryption By Hierarchical Key Aggregation

[1] R Rakesh, [2] Anoop S
[1] P.G Scholar, [2] Assistant Professor in IT,
College of Engineering Perumon(CUSAT), Kerala, India
[1] rrakesh093@gmail.com, [2] anoopsivasankar@yahoo.co.in

Abstract: **An efficient cryptographic cloud approach for data sharing system which is designed in such a way that data is shared among a large collection of different kinds of users. Since, Data sharing is an important functionality in cloud storage. Main concerned of cloud security is performing securely make all the transactions and efficiently share a collection of data related to any subject specializations areas with others users in cloud storage. Development of new novel concept of Hierarchical Key-Aggregate Searchable Encryption. This concept is implemented through development of a concrete Hierarchical key-aggregate searchable encryption framework scheme. This scheme is described as where a data owner only needs to produce and transmit a single aggregate key to a data user for sharing a large number of documents, also generated key is structured using access tree, data access policies are defined such as who can access the encryption data and data users are grouped under various hierarchical structuring. On the other side data user only needs to submit a single aggregate trapdoor to the cloud server, so that he/she can query over the shared set of documents with the help of generated single aggregate trapdoor. This proposed scheme is perfectly more secure and practically efficient. It is an effective and efficient method for public cloud storage, which is considered as a best solution to build a practical data sharing system.**

***Index Terms*—Cloud storage, data sharing, data privacy, searchable encryption.**

I. INTRODUCTION

Generally nowadays online Storage is a basic requirement of everyone person who has access to internet, So that he/she can access their stored files located in some server by means of Cloud storage. Development of cloud storage and cloud servers, if we need to have personal cloud storage or cloud storage for business purpose, it is very easy to backup all your important computer files online. This can be shared with various users by means of internet. Today, every user is sharing a number of various documents, which are considered to be under types like photos, videos and documents via various social networking based applications on daily basis. There are various benefits of using cloud storage like lower cost, increased agility and also best resource utilization has add more focus on cloud storages, a number of business users are using the cloud storage for business purposes.

We are more flexible in accessing all our files which are stored in cloud storages and can be retrieved from anywhere in the world, with the gain of knowing that all our files are securely stored and available to us. But in this dynamically change lifestyle, users are worried about the data loss or data leaks which happen usually in case of cloud storage. These types of data leak happen, due to an untrusted cloud provider or via hackers who decodes our documents files using various types of malicious software. A common approach which is normally used is to encrypt all the documents files. Which are expected to be uploaded to the cloud server by authorized data owner. The encrypted data can be retrieved and then performing decryption by user, who have right access keys.

This category of cloud storage is called as "**Cryptographic cloud storage**."

Considering an example, where sharing a photo and videos is a common fashion nowadays. It is performed with the help of social network based applications like Facebook, Facebook Messenger, WhatsApp, Viber, Line, KakaoTalk Messenger, Skype, LiveProfile, Groupme, Kik Messenger, ChatON, WeChat etc. By using these cloud storages, the number of hiring expert IT professionals to manage and perform maintenance can be reduced and also the saving the cost invested for IT infrastructure development. Generally, all types of users share various documents through cloud storage available like Google drive, Dropbox, Citrix etc. Also Cloud service providers examples like Amazons EC2 and S3 [2], Google App Engine [3], and Microsoft Azure [4], these provide us all the resources required as per our needs. We can perform payment of them as a plan i.e. pay for what you use fashion. These services and cloud based utilities provided are very low cost and easily accessible.

One of the major security concerned issues: data security and privacy of user data. Since, all the data uploaded by us are stored and managed by means of internet through cloud service providers. So, can't completely trust these companies. Major two challenging tasks which cloud storage faces are: how will a data user perform keyword

based searching over the set of documents shared and how can a user extract the required relevant data which can be retrieved by a given keywords. The above stated two challenging tasks can be efficiently solved by the use of Searchable Encryption (SE) scheme for development and implementation of new system.

Regarding the management of encryption and decryption keys is a serious issue of cloud storage.

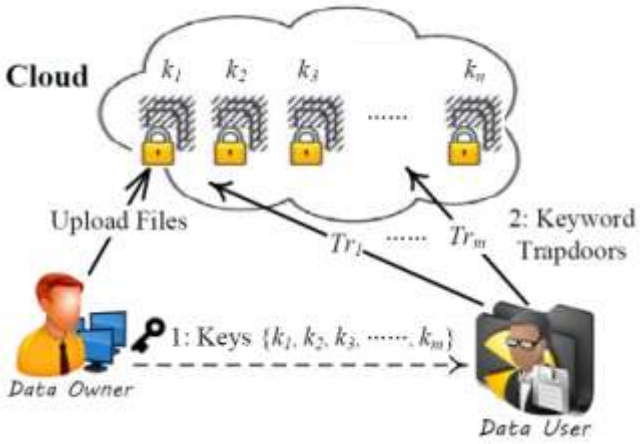

Fig: 1 Traditional method of data sharing

Describing about the traditional method of data sharing provider by various cloud storage providers [7], in Fig.1 shows that there are two types of users: Data owner and Data user. Data owner is uploading a numbers of documents to cloud server which are shared with the data user, which were encrypted with a separate key (i.e. First document encrypted with key1 (k1), second document encrypted with key2 (k2), and so on.). Number of documents to be encrypted should be equal to the number of keys used for encryption. Encryption key generated is transferred to the respective data user by means of a secure communication channel by the data owner. Than after validating all these actions, data user can perform keyword based searching over the shared documents by generating keyword trapdoors (Tr1, Tr2, Tr3, …, Trm) for each document. If a keyword based searching has a match, than the cloud server return back the original files which were shared by the data owner to the corresponding requested queried data user.

Various methods have been constructed for data sharing via cloud storage, their efficiency are to be increased by means of development of new concepts and schemes. This paper is organized as follows: Section 2 illustrates some of the methodologies and related work used for data sharing through cloud storage, Section 3 illustrating the proposed scheme i.e. Searchable Encryption by Hierarchical Key Aggregation, and Section 4 concludes the Searchable Encryption by Hierarchical Key Aggregation scheme.

II. RELATED WORKS

Cloud storage and cloud servers have become popular and are widely used by many individuals, IT professionals and organizations. The frequent using of the cloud storage has raised various security issues which are concerned about the outsourced data. Properties like confidentiality, integrity and data access control needed to be checked and maintained by cloud servers. Both academic and industrial world are making efforts to maintain the security of the outsourced data.

A. Identity Based Encryption

Considering the access control architectures, Attribute-Based Encryption (ABE) schemes is the most popular and well known ones due to its scalability and security. But Access Control List (ACL), which only defines that entities have the authorized access right, these schemes encrypt the data under the data access policy which only ensures the eligible entities to perform the decryption. A research work Fuzzy Identity-Based Encryption (IBE) was introduced by Sahai and Waters in the year 2005. Describing the Fuzzy IBE scheme, a private key for an identity set w, it can be used for decrypting the ciphertext encrypted with an slightly different or modified identity set w'. Fuzzy IBE detects the error tolerance by usually setting the threshold value for root node smaller than the identity set size.

B. Keypolicy Attribute-Based Encryption

Based on the Fuzzy IBE scheme, Goyal et al. presented the Key Policy-Attribute Based Encryption (KP-ABE), which is described as ciphertexts are labelled together with sets of attributes and private keys are linked with access structures that can set controls. So, that which ciphertexts a user can decrypt i.e. assigning the access and decryption rights to a user. In this scheme, when a user submits a secret request, the trusted authority determines which combination of attributes should appear in the ciphertext for the user to decrypt the document files.

C. Cipher text Policy-Attribute Based Encryption

A complementary scheme to KP-ABE was introduced by Bethencourt, called as Ciphertext Policy-Attribute Based Encryption (CP-ABE). In this ciphertext policy attribute based encryption system, described as: a user's private key is associated with attributes set and the encrypted ciphertext will explain the access policies defined over the attributes. A user will be able to decrypt if and only if his attributes satisfy the ciphertext's policy. One of the most challenging issues in data sharing systems is the enforcement of access policies and the support of policies changes or updates. CP-ABE is becoming a major promising cryptographic solution to this concerned issue. It enables the data owners to define their own access policies over the user attributes and

enforcing these policies on the data to be distributed among the data users.

D. Broadcast Encryption

In case of broadcast encryption (BE) scheme, a broadcaster (Data owner) encrypts a document files for some subset S of data users who are currently listening on a broadcast secure channel. Any data user who is in S can use his/her private key to decrypt the broadcasted document files. A BE scheme as a tuple of three polynomial time algorithms BE = (Setup, Encrypt, Decrypt). Setup algorithm is run by the system to set up the scheme. It has taken input a security parameter and the number of receivers n, outputs n private keys d1, d2,…, dn and a public key (pk). Encrypt algorithm is run by the broadcaster to encrypt a document file for a subset of users s. It takes as input a public key pk and a subset of users S, outputs a pair (Hdr, K), where Hdr is known as the header and K is a message encryption key and which is encapsulated in Hdr. We will often refer to Hdr as the broadcast ciphertext. For a concrete developed message, it will be encrypted by K and broadcasted to the users in S. Decrypt (pk;S;i;di;Hdr): this algorithm is run by the user to decrypt the received messages. It takes as input a public key pk, a subset of users S, a user id i, the private key di for user i and a header Hdr, outputs the message encryption key K or the failure symbol. The K will be used to decrypt the received messages.

E. 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) [9], [12 – 13]. Both SSE and PEKS can be described as the tuple SE= (Setup, Encrypt, Trapdoor, and Test): Setup algorithm is run by the owner to set up the scheme. It takes as input a security parameter, and outputs the necessary keys. Encrypt(k;m): this algorithm is run by the owner to encrypt the data and generate its keyword ciphertexts. It takes as input the data m, owner's necessary keys including searchable encryption key k and data encryption key, outputs data ciphertext and keyword ciphertexts Cm. Trpdr(k,w): this algorithm is run by a user to generate a trapdoor Tr 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 Tr and the keyword ciphertexts Cm, outputs whether Cm contains the specified keyword.

F. Key-Aggregate Encryption (KAE)

Recently more attention has been created around the cloud storage, which is based on data sharing systems [5]-[7]. By considering the paper [7] which points out that how to decrease the number of keys used for data encryption. In traditional approach, all used encryption keys must be distributed among the concerned authorized users. This challenge is solved by KAE, where it generates an aggregate key which will be used by the user to decrypt all the documents shared with him/her. Concept of KAE is to obtain the original document by decrypting with a single aggregate key, which was encrypted with different keys. To perform this data owner, not only needs the public key but also the identity of each document. This is concept is adapted from the broadcast encryption scheme [29].

In development of KAE scheme, the data owner is designed as broadcaster. Broadcaster will be having the public key and master secret key. Data user is designed as the receivers, who are listening to this secure broadcast channel. Generally, public information which consists of various relevant information like data owner's master secret key and encryption key. Here, data encryption is per-formed using the symmetric encryption in broadcast encryption. But the key aggregation and data decryption is done by the algorithms like BE.Encrypt and BE.Decrypt respectively. By using scheme [7], which delegates all the decryption rights to the data users. The problem with KAE, we can't perform searching over the encrypted documents. So, the development of new scheme is needed, which will provide us to perform keyword based searching, trapdoor generation and also more complex procedure to obtained a keyword matching in more efficient way. So, KASE scheme was designed and developed by the researchers in the field of research and development.

III. THE PROPOSED SCHEME

In this section, a new system is proposed for development of Searchable Encryption by Hierarchical Key Aggregation scheme ideas is adapted from papers like Multi-key searchable encryption scheme[31], the key-aggregate cryptosystem scheme[7] for scalable data sharing and Key-Aggregate Searchable Encryption (KASE) for Group Data Sharing via Cloud Storage[42]. This scheme consists of data owner, data user, a number of domain authorities and a trusted authority as shown in Fig.2.

2 System Model

Generally cloud service provider (CSP) manages cloud servers to provide data storage services. Data owners

encrypt their data files with help of key structure and store them in the cloud for sharing with data users. To access the shared data documents files, data user downloads the encrypted data files which was uploaded to the cloud by data owner and then decrypt them. Each data owner is administrated by a domain authority. Each domain authority is maintained and managed by its parent domain authority or the trusted authority. Data owners, data user, domain authorities, and the trusted authority are structured in a hierarchical manner.

The trusted authority is the root authority and who is responsible for managing top-level domain authorities. Each top-level domain authority responds to a top-level organization, such as a federated cloud based enterprise, while each lower-level domain authority responds directly to a lower level organization. For example, Data owners may respond to employees in an organization. Each domain authority is responsible in managing all the domain authorities at the next level or the data owners in its domain. In our designed new system, neither data owners nor data users will be always online. They usually come online only when necessary, while the three components of the system are always online are: cloud service provider, the trusted authority, and domain authorities.

A. Searchable Encryption by Hierarchical Key Aggregation

The newly designed system framework scheme is described in this section; this scheme consists of seven algorithms:

1. ***Setup***: This algorithm is run by cloud server to setup all system parameters. Generate a bilinear mapping based group sharing system, set the maximum possible number of documents available with the data owner. Two operations are computed which are random generator calculation and selecting a one way hash function. Cloud server broadcast the generated system parameter and public key. Here, the domain authority can manage and authorize the data uploaded by a user. If a user leaves a group than restructuring is done on the policy schemes.

2. ***Keygen:*** This algorithm is run by data owner to generate his/her key pair which will be used for document encryption by the Encrypt algorithm. In this stage, we have public key and master secret key along with the generated key pair. The cloud server performs the key structuring based on data owner role and set a hierarchical key structuring.

3. ***Encrypt:*** This algorithm is run by data owner to perform data encryption and also generate corresponding cipertexts for all the documents which will be uploaded. For the creating the keyword ciphertexts, it takes the document file index, randomly picks a searchable encryption key for each document and generates a delta information. It will produce a ciphertext for a keyword, this generated ciphertexts are stored under cloud server. Data owner sets the attribute based expiration time for the document file.

4. ***Extract***: This algorithm is run by data owner and generating an aggregate searchable encryption key and this key is send to all authorized users via a secure communication channel. This algorithm takes input as master secret key and generates an aggregate key as output. Data owner than send this aggregate key to data users, so that they can perform keyword searching over the shared documents.

5. ***Trapdoor:*** This algorithm is run by data user and performs keyword searching by generating trapdoor. Searching for matching relevant documents by use of single aggregate searchable key. Only one single aggregate trapdoor is generated based on the data owner who request for uploading data. A single keyword which is used for searching will be matched against each trapdoor submitted. Than data user sends this generated single trapdoor and subset of matched documents.

6. ***Adjust***: This algorithm is run by cloud server and creating right set of trapdoor. It accepts input as system publicly available parameters, all documents index in the set and also single aggregate trapdoor. It performs adjusting process on the single aggregate trapdoor and output a new right single trapdoor. This produced trapdoor will be used for next Test algorithm for performing keyword search over the shared collection of documents.

7. ***Test:*** This algorithm is run by the cloud server. Cloud server does a series of keyword searching by using the input, which is adjusted trapdoor and creates the delta information which is relevant to subset by using searchable encryption key. Output produced will be binary, i.e. true or false values after performing various computations.

IV. CONCLUSION

In this paper, a new system is developed, which is searchable encryption by hierarchical key aggregation and constructing a concrete scheme for this system. Results

based on various comparison and analysis confirm that this work can give a better and more efficient solution for building a more secure data sharing system based on public cloud storage available on internet. Since, system efficiency has increased, complexity will increased due to hierarchical key aggregation process. This scheme is described as where a data owner only need to generate a single aggregate key using hierarchical structuring by means of access tree and send this key to the data user. Data access policies are defined such that who can access the encrypted data and data users are grouped under various hierarchical structuring. Data user only needs to submit a single aggregate trapdoor to the cloud server, so that he/she can query over the shared set of documents with the help of generated single aggregate trapdoor. This proposed scheme is perfectly more secure and practically efficient for public cloud storage.

REFERENCES

1. Cloud-Storage,
2. http://www.thetop10bestonlinebackup.com/cloud-storage.
3. Amazon Web Services (AWS), http://aws.amazon.com
4. Google App Engine, http://code.google.com/appengine/
5. Microsoft Azure, http://www.microsoft.com/azure/
6. 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.
7. 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.
8. Chu, S. Chow, W. Tzeng, et al. "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage", IEEE Trans-actions on Parallel and Distributed Systems, 2014, 25(2): 468-477.
9. X. Song, D. Wagner, A. Perrig. "Practical techniques for searches on encrypted data", IEEE Symposium on Security and Privacy, IEEE Press, pp. 44C55, 2000.
10. R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky. "Searchable symmetric encryption: improved definitions and efficient con-structions", In: Proceedings of the 13th ACM conference on Computer and Communications Security, ACM Press, pp. 79-88, 2006.
11. P. Van,S. Sedghi, JM. Doumen. "Computationally efficient searchable symmetric encryption", Secure Data Management, pp. 87-100, 2010.
12. S. Kamara, C. Papamanthou, T. Roeder. "Dynamic searchable symmetric encryption", Proceedings of the 2012 ACM conference on Computer and communications security (CCS), ACM, pp. 965-976, 2012.
13. Boneh, C. G, R. Ostrovsky, G. Persiano. "Public Key Encryption with Keyword Search", EUROCRYPT 2004, pp. 506C522, 2004.
14. 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.
15. J. Li, Q. Wang, C. Wang. "Fuzzy keyword search over encrypted data in cloud computing", Proc. IEEE INFOCOM, pp. 1-5, 2010.
16. Bosch, R. Brinkma, P. Hartel. "Conjunctive wildcard search over encrypted data", Secure Data Management. LNCS, pp. 114-127, 2011
17. Dong, G. Russello, N. Dulay. "Shared and searchable encrypted data for untrusted servers", Journal of Computer Security, pp. 367-397, 2011.
18. 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.
19. 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.
20. J. Li, K. Kim. "Hidden attribute-based signatures without anonymity revocation", Information Sciences, 180(9): 1681-1689, Elsevier, 2010.
21. 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.
22. 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.

23. 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.

24. 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.

25. 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.

26. Boneh, C. Gentry, B. Waters. "Collusion resistant broad-cast encryption with short ciphertexts and private keys", Advances in Cryptology CCRYPTO 2005, pp. 258-275, 2005.

27. 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.

28. D. Boneh, B. Lynn, H. Shacham. "Short signatures from the Weil pairing", Advances in Cryptology ASIACRYPT 2001, pp. 514-532, 2001

29. 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.

30. D. Boneh, C. Gentry and B. Waters. "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys", CRYPTO05, pp. 258C275, 2005.

31. M. Li, W. Lou, K. Ren. "Data security and privacy in wire-less body area networks", Wireless Communications, IEEE, 17(1): 51-58, 2010.

32. R. A. Popa, N. Zeldovich. "Multi-key searchable encryption". Cryptology ePrint Archive, Report 2013/508, 2013.

33. T Parameswaran, S Vanitha, K S Arvind, "An Efficient Sharing of Personal Health Records Using DABE in Secure Cloud Environment" International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 3, March 2013

34. Mihir Bellare, Alexandra Boldyreva, Adam O'Neill, Deterministic and Efficiently Searchable Encryption, Advances in Cryptology - CRYP TO' 07 Proceedings, Vol. 4622, p p. 535–552, Springer, 2007.

35. B.Lynn, "The pairing-based cryptography library," 2006.

36. J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in Security and Privacy, 2007. SP'07. IEEE Symposium on. IEEE, 2007, pp. 321–334.

37. M. Chase, "Multi-authority attribute based encryption," in Theory of Cryptography. Springer, 2007, pp. 515–534.

38. Sahai and B. Waters, "Fuzzy identity-based encryption" in Advances in Cryptology-EUROCRYPT 2005. Springer, 2005, pp. 457–473.

39. J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," Parallel and Distributed Systems, IEEE Transactions on, vol. 22, no. 7, pp. 1214–1221, 2011.

40. S. Ruj, A. Nayak, and I. Stojmenovic, "Dacc: Distributed access control in clouds," in Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on. IEEE, 2011, pp.91–98.

41. K. Ren, C. Wang, Q. Wang et al., "Security challenges for the public cloud," IEEE Internet Computing, volume. 16, no. 1, pp. 69–73, 2012.

42. PBC library: The pairing-based cryptography library. http://crypto.stanford.edu/pbc/.

43. Baojiang Cui, Zheli Liu and Lingyu Wang, "Key-Aggregate Searchable Encryption (KASE) for Group Data Sharing via Cloud Storage", IEEE TRANSACTIONS ON COMPUTERS, VOL. 6, NO. 1, JANUARY 2014.