Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 3, March -2017

Secure Group Data Sharing In Cloud Storage Using Key Aggregate Searchable Encryption

Kolhe Jayshree Jalindar.[1], Lonkar Bhagyashri D.[2],Pansare Suvarna B.[3], Paymode Ashwini B.[4]

[1]*Computer Dept., SGOI COE, Belhe*
[2]*Computer Dept., SGOI COE, Belhe*
[3]*Computer Dept., SGOI COE, Belhe*
[4]*Computer Dept., SGOI COE, Belhe*

Abstract — *Now a day cloud storage is very legendary. The main goal this work is to safe and efficient data sharing. Good reliability & accessibility, powerful protection, unpleasant consequence recovery and low cost are provide by cloud storage. This work done a through of making the secure data sharing and versatile release. On public cloud storage ability of sharing selected individual data or group of individual data with group of users. They able to change easy group of people by a constant size encryption key using plan of particular result which is key aggregate searchable encryption. To search over share data owner have to assign to users large number of key for encryption search and data users will have to securely store the received key and report an equally large number of word trapdoors to the cloud. Owner upload files on cloud by tagging files and use only one aggregate key to share large number of documents. Data user send only one aggregate trapdoor to cloud for searching documents..*

Keywords- *Cloud storage, Aggregate key, Data privacy, Encryption, Decryption, Data sharing*.**.**

I. INTRODUCTION

Cloud computing is a common term for anything that involves scalable services, delivering hosted services like accessing, data sharing, etc., over the web on demand basis. Cloud computing is known as an alternative to traditional technology due to its low-maintenance and better resource-sharing capabilities. The main goal of cloud computing is to provide high performance energy of computing for various field like military and research organization for performing billions of computations. It is also used for core technology behind many online services for personal applications.

With current technology user can access almost all of their files or emails by mobile phone or computer from any corner of the world. Everyday users are also sharing personal data, such as video and photos with others through social network applications based on cloud. In the cloud storage efficient public key encryption strategy which support flexible delegation in the sense that any subset of the cipher texts is decrypt able by a constant - size decryption key. The cloud computing is enforce on distributed computing service oriented architecture utility and parallel computing. The essential security requirement can be attained by combining both the cryptographic cloud storage along with searchable encryption scheme. Also key management is a serious problem. Generally user share various types of documents through cloud storage networking application like Drop box, Cloud me, Google drive, Citrix.

The common approach is for the data owner to encrypted all the data before uploading to the cloud then the encrypted data may be retrieved and decrypted by those users who have the decryption keys. Such a cloud storage is obtain called the cryptographic cloud. Key-aggregate searchable encryption (KASE) to adjust the problem of preserving privacy in public cloud storage in which data owner required to distribute large number of a keys to other user to enable the access to their data.

Searchable encryption scheme is a cryptographic technique that allows search of specific information in an encrypted content. One of the important paper on searchable encryption by Let see this concept with an intuitive example. Assume that Alice want to redirect the mail to Bob containing the term “office” as she is away in a holiday. Now how would the service provider know which mail to send to Bob if it is not able to read the data ? Again it is a security concern to transmit the data without end to end encryption. A solution of this problem is to use searchable encryption**.** Here Alice chose to encryption the message and create a collection of tags or search keyword that can uniquely identify the message for a specific search. Now there tags are appended to the encrypted message so that the service provided can perform the task of forwarding the mail without knowing the details of the mail. In cloud system overall cost of data storage is less as it does not require managing and maintaining expensive hardware. To address data leak problem in cloud cryptographic and storage system in a referred. In which data owner firstly encrypt all data before storing on a cloud in such way that only user whom having decryption keys can be decrypt or fetch the data. In propose scheme, Tagging a file. The tag includes the user that can access the file. Because of this tagging, only those file can be downloaded by secondary user which he/she is permitted to download(tag in that file). Such large number of a keys cannot securely manage and store in cloud system. Therefore such system implies as inefficient and impractical for a storage, complexities and communication.

Multi-user Searchable Encryption : Including PEKS as well as SSE schemes, on searchable encryption there is a rich literature. The keyword search under the multi-tenancy setting is a more common scenario in the context of cloud storage in contrast to those existing work. In such a scenario, to share a document with a group of authorized users the data owner would like, and over the "multi-user searchable encryption" (MUSE) scenario, each user can provide a trapdoor who has the access right to perform the keyword search. To such a MUSE scenario some recent work focus, although to achieve the goal with access control they all adopt single-key combined. With all users by sharing the document's searchable encryption key who can access it, MUSE schemes are constructed, and to achieve coarse-grained access control broadcast encryption is used. To achieve fine-grained access control aware keyword search attribute based encryption (ABE) is applied. As a result, in MUSE, how to control which users can access which documents is main problem, whereas There is not considered how to reduce trapdoors and shared the number of keys. The solution for the latter can provide by key aggregate searchable encryption, and it can be make more practical and efficient for MUSE.

Multi-key Searchable Encryption : In the case of application which has a multi-user, to search over considering that there is proportional the number of trapdoors to the number of documents, The concept of multi-key searchable encryption (MKSE) was introduced by Popa.. This algorithm is explained as a data user to give a single trapdoor which consists of a single keyword to the cloud server. But on other hand, the cloud server gives provision to search over the keyword trapdoor by using different keys. It consists of a Multi-Key Searchable Encryption (MKSE) which shows that a data user is submitting his/her generated trapdoor(Tr) to cloud server and the cloud server performing the adjust and test algorithm on the document collection. The main goals of both i.e. KASE and MKSE are completely different ideas. Goal of MKSE: when keyword search is performed by the cloud server with only one trapdoor on different types of user owned documents, that of KASE: by mainly providing the generated single aggregate key to data users in a group sharing based system. Speaking more about the MKSE, data user can store public data information which is known as Delta on cloud server. This public information is relevant to data user key and used encryption key. Data user can perform searching for a word on all the documents, for doing this he/she needs the data user key to calculate the trapdoor for the word and directly submit this generated trapdoor value to cloud server. Cloud server uses this information to convert received keyword trapdoor on the key available with the data user. This process is known as adjust. By doing so, cloud server can perform traditional searching by means of single-key with the newly generated trapdoor. In MKSE the adjust process is an approach to perform searching on the group of documents shared by means of single trapdoor. This adjust process can't be applied directly to the development of KASE scheme.

II. LITERATURE SURVEY

Baojiang Cui, Zheli Liu_ and Lingyu Wang [1] gives the information about characteristic of low maintenance, cloud computing provides financially and efficient solution for sharing data group resource among cloud users, The scheme is also very flexible, it can be simply extended to support more advanced searching query. we conclude that this provide a tremendous building block for the construction of secure services in the cloud storage which are not trusted by user. As we will share only single key the storage space required will become less and more efficient.

C. Chu, S. Chow, W. Tzeng, et al [2] proposed multi-key search to protect the users data privacy is a central question of cloud storage. More mathematical tools and cryptographic scheme are getting more versatile and often multiple keys for a single application. In this article we consider to "compress" secrete key in public key cryptosystems which support delegation of secrete key for different cipher-text classes in cloud storage. No matter which one among the power set of classes the delegate can always get constant size an aggregate key. The approach is holders share a similar set of privileges and more flexible than hierarchical keys assignment which can only save spaces for all key.

R. A. Popa ,N. Zeldovich [3] Introduces system that provide stronger security technique in that they construct a searchable encryption strategy that are enables keyword search over data encrypted with different keys. The scheme is practical designed to included in a new system for protecting data secrete in client - server applications against attacks on the server.

C. Wang, Q. Wang, K. Ren, and W. Lou [4] authors share data and privacy preserving auditing scheme with large groups in the cloud. They are utilize group signature to compute verification information on shared data. That is the TPA those able to audit correctness of shared data but cannot reveal the identity of the signers on each block. The original user can efficiently add new users to the group and close the identities of signers on all blocks.

S. Yu, C. Wang, K. Ren, and W. Lou [5] key search is introduced cryptographic storage system that enables secure file and sharing on un trusted servers, named Plautus. Dividing files into file group and encrypting each file group with a unique file block key. The data owner can share the file groups with others through delivering the corresponding lockbox key, here the lockbox key is used to encrypt the file block keys.

D. Boneh, C. G, R. Ostrovsky, G. Persiano [6] proposed security introduced the concept of decoding searchable encryption and to implement this new primitive using one IDKEM, KEM and hash functions. In that provided a IND CCA security in which relate precise security to the properties of the inner primitives.

III. PROBLEM STATEMENT

Consider a scenario where two employees of a company would like to share some confidential business data using a public cloud storage service. For instance, Alice is a primary user wants to upload a large collection of financial documents to the storage, which are meant for the directors of different departments to review. Suppose those documents contain highly sensitive information that should only be accessed by authorized users, and Bob is secondary user is one of the directors and is thus authorized to view documents related to his department. Due to concerns about potential data leakage in the cloud, Alice encrypts these documents with different keys and generate keyword ciphertext based on department names, before uploading to the cloud storage. Alice then uploads and shares those documents with the directors using the sharing functionality of the cloud storage. In order of Bob to view the documents related to his department, Alice must delegate to Bob the rights both for keyword search over those documents and for decryption of documents related to Bob's department. With a traditional approach, Alice must securely send all the searchable encryption key to Bob. After receiving these keys, Bob must store them securely and then he must generate all the keyword trapdoors using these keys in order to perform a keyword search. Alice is assumed to have a private document set $fdoc_i$ i=1, and for each document doc_i, a searchable encryption key k_i is used. Without loss of generality, we suppose Alice wants to share m documents $fdoc_i$ i=1 with Bob. In this case, Alice must send all the searchable encryption key fk_i i=1 to Bob. Then, when Bob wants to retrieve documents doc_i with key k_i and submit all the trapdoors fTr_i i=1 to the cloud server. When m is sufficiently large, the key distribution and storage as well as the trapdoor generation may become too expensive for Bob's client side device, which basically defies the purpose of using cloud storage..

IV. KASE FRAMEWORK

The KASE framework is composed of seven algorithms. Specifically, to set up the scheme, the cloud server would generate public parameters of the system.

1.Setup(1^λ, n):- algorithm, and these public parameters can be reused by different data owners to share their files. For each data owner, he/she should produce a public/master-secret key pair. 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.

2.Keygen:- algorithm. Keywords of each document can be encrypted. This algorithm is run by the data owner to generate a random key pair (pk,msk).

3.Encrypt(pk, i):- algorithm with the unique searchable encryption key. Then, the data owner can use the master-secret key to generate an aggregate searchable encryption key for a group of selected documents. 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 i for its searchable encryption key ki. On input of the owner's public key pk and the file index i, this algorithm outputs data ciphertext and keyword ciphertexts Ci.

4.Extract(msk, S):- The aggregate key can be distributed securely (e.g., via secure e-mails or secure devices) to authorized users who need to access those documents. an authorized user can produce a keyword trapdoor. 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.

5.Trapdoor(kagg, w):- Algorithm using this aggregate key, and submit the trapdoor to the cloud. After receiving the trapdoor, to perform the keyword search over. 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 kagg and a keyword w, then outputs only one trapdoor Tr.

6.Adjust(params, i, S, Tr):- Algorithm to generate the right trapdoor for each document, and then run. 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.

7.Test(Tri, i): - Algorithm to test whether the document contains the keyword. 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 doci contains the keyword w.

V. PROPOSED SYSTEM

Now a day's cloud storage is known as a promising solution for providing convenient, universal, and on demand access to greater amounts of information shared on the internet. Today, billions of users are sharing private data such as photos, videos, confidential documents with their friends using social networking applications based on the cloud storage. Business users are also getting attracted by cloud storage due to its numerous advantages, including lower price, greater agility and better resource utilization capabilities[1]. But there is the practical problem of preserving privacy of data sharing system based on public cloud storage which need a primary user to distribute a huge number of keys to

secondary users to enable them to access their documents, they are proposing first time the concept of key - aggregate searchable encryption (KASE) and built a concrete KASE scheme. We are discussed about the limitations, that how to reduce a number of shared keys with single aggregate key for all documents.

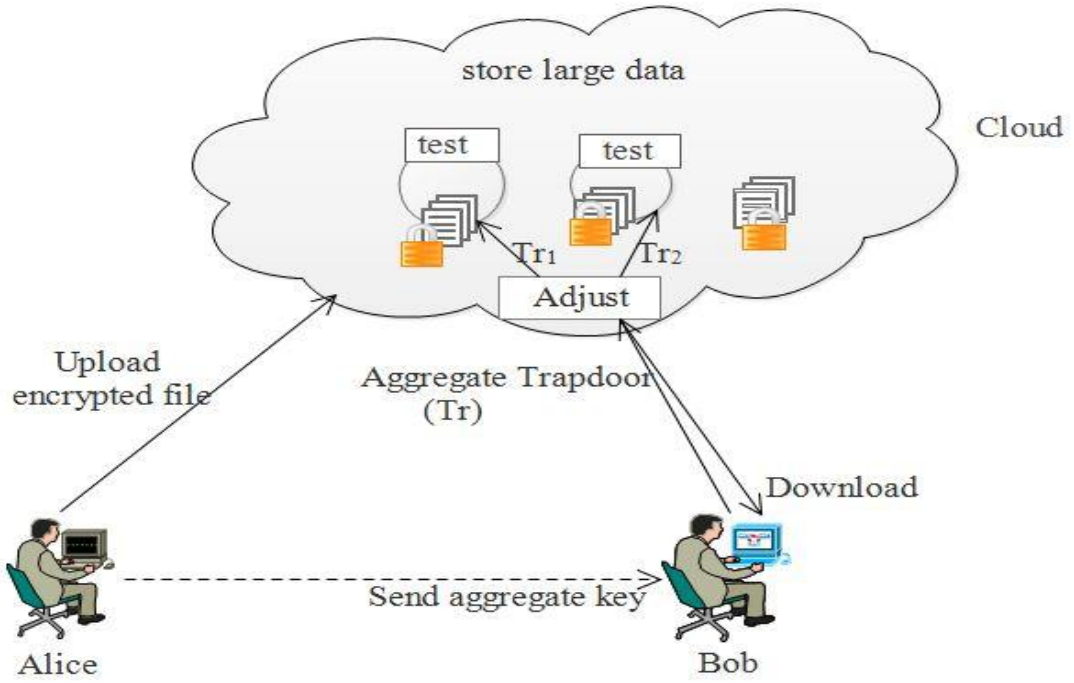

Fig.Architecture of key aggregate searchable encryption

VI. CONSTRUCTION OF THE KEY AGGREGATE SEARCHABLE ENCRYPTION

In this paper, we propose the new concept of key aggregate searchable encryption to improve the solution, in KASE primary user needs to issue single aggregate key, instead of sharing number of documents with secondary user and secondary user need to issue single aggregate trapdoor, instead of number of trapdoor to the cloud server. The cloud server can use this aggregate trapdoor and some public data to do keyword search and revisit the result to secondary user. In KASE, the delegation of keyword search right can be achieved by sharing the single aggregate key. To building a key aggregate searchable encryption method under which any subset of the keyword cipher-text from any set of documents is searchable with a constant size trapdoor produce by a constant size aggregate key.
KASE system structure was described in the above section, this KASE system consists of seven algorithms:

1. **SETUP(1^{λ}, n):** The cloud server will use this algorithm to initialize system parameters. In that input is security level parameter 1^{λ} and number of ciphertext classes n, public system parameter param is the output.
2. **KEYGEN:** This algorithm is run by the primary user to randomly generate a random key pair (pk, msk). For document encryption which will be used by the encrypt algorithm. In this stage, we have public key and master secrete key along with the generated key pair.
3. **ENCRYPT(pk, i):** This algorithm is run by primary user to executes data encryption and also generate corresponding ciphertext for all the document which will be uploaded. In that input is the file index i and owner public key pk and outputs keyword ciphertext ci and data ciphertext.
4. **EXTRACT(msk, S):** This algorithm is run by data owner and return an aggregate searchable encryption key and this key is send to all authorized users using secure communication channel. It take as input the owner's msk is a master-secret key and s is a set which enclose the directory of document and output is the aggregate key kagg.
5. **TRAPDOOR(kagg, x):** This algorithm is run by primary user and do keyword searching by generating trapdoor. It takes as input the aggregate searchable encryption key kagg and a keyword and a keyword w and output is only one trapdoor.
6. **ADJUST(params, i, S, Trd):** This algorithm is run by cloud server and creating trapdoor. It takes as input the system public parameters params, the set S of document's indices, the index i of target document and the aggregate trapdoor Tr, and output is each trapdoor tri for the i-th target document in S.
7. **TEST(Tri, i) :**This algorithm is run cloud server. It take as input the trapdoor Tri and the document index i and output will be binary i.e. true or false values after performing various c

VII. CONCRETE GROUP DATA SHARING SYSTEM

When constructing a practical group data sharing system, it is important to reduce the number of keys belonging to a user. In this subsection, we will introduce how to build such a system based on the KASE and KAE schemes with the same public parameters. We consider a group data sharing system without using any private cloud, but instead based on widely available public cloud services, such as Dropbox or citrix. Based on such a consideration, we assume a group manager (e.g., the HR director of an organization) with an authorized account to act in the role of "manager" who will be responsible for management of the system including maintaining the public system parameters stored in the cloud.

Table Definitions

We assume the cloud uses databases to manage the necessary information and four database (traditional or NOSQL databases) tables are thus defined as follows:

- Table **group**<groupID, groupName, parameters> is to store the system parameters.
- Table **member**<memberID, membeName, password, publicKey> is to store members' information including their public key.
- Table **docs**<docID, docName, OwnerID, EncKey, SEKey, filePath> is to store the uploaded document of an owner with identity ownerID.
- Table **sharedDocs**<SID, memberID, OwnerID, docIDSet> is to store the documents of a member with identity memberID shared by the owner with identity OwnerID. Field docIDSet is for all the indices of documents.

Note that the owner can encrypt the encryption key and SE key by his/her private key and store the ciphertexts in the fields EncKey and SEKey. In this way, he can reduce the cost of key management and only focus on how to safely store his/her private key.

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 groupID for this organization and insert a record into table **company**. Moreover, it assigns an administrator account for the manager. Then, the group data sharing system will work under the control of manager. To generate the system parameters params, manager runs the algorithm KASE **Setup** and updates the field parameters in table **company**.

User registration: When adding a new member, the manager assigns memberID, membeName, password and a key pair generated by any public key encryption (PKE) scheme for him, then stores the necessary information into the table **member**. A user's private key should be distributed through a secure channel.

User login: Like most popular data sharing products (e.g., Dropbox and citrix), our system relies on password verification for authenticating users. To further improve the security, multi-factor authentication or digital signatures may be used when available.

Data uploading: upload a document, the owner runs KAE **Encrypt** to encrypt the data and KASE **Encrypt** to encrypt the keyword ciphertexts, then uploads them to the cloud. The cloud assigns a docID for this document and stores the encrypted data in the path file Path, then inserts a record into the table **docs**. In addition, the owner can encrypt the keys using his/her private key and store them into the table **docs**.

Data sharing: To share a group of documents with a target member, the owner runs KAE **Extract** and KASE **Extract** to generate the aggregate keys, and distributes them to this member, then inserts/updates a record in table **sharedDocs**. If the shared documents for this member are changed, the owner must reextract the keys and update the field docIDSet in table **sharedDocs**.

Keyword Search: To retrieve the documents containing an expected keyword, a member runs KASE **Trapdoor** to generate the keyword trapdoor for documents shared by each owner, then submits each trapdoor and the related owner's identity OwnerID to the cloud. After receiving the request, for each trapdoor, the cloud will run KASE **Adjust** the trapdoor for each document in the docIDSet and run KASE **Test** to perform keyword search. Then, the cloud will return the encrypted documents which contains the expected keyword to the member.

Data retrieving: After receiving the encrypted document, the member will run KAE **Decrypt** to decrypt the document using the aggregate key distributed by the document's owner.

VIII. IMPLEMENTATION DETAIL

In our implementation, two source libraries about pairing computation are used:

1) pbc library is used to implement cryptographic operations running in computers. Because the aggregate key and trapdoor contains one G element, and the keyword ciphertexts only contain two G1 elements, we can select the Type-A pairing. Type-A pairing is the fastest (symmetric) pairing among all types of curves, which is constructed on the curve $Y^2 = X^3 + X$ over the field F_p for some prime $p=3 \bmod 4$. Each cryptographic operation involved in our system will be evaluated in two different platforms: in Java on Computer of Intel(R) Core(TM)i5-3337U CPU @ 1.80GHZ with Windows7 0S.

IX. CONCLUSION

Considering the practical problem of privacy preserving data sharing system based on public cloud storage which requires a data owner to distribute a large number of keys to user to enable them to access his/her documents, we for the first time propose the concept of key-aggregate searchable encryption (KASE) and construct a concrete KASE scheme. Both analysis and evaluation results confirm that our work can provide an effective solution to building practical data sharing system based on public cloud storage. In a KASE scheme, the owner only needs to distribute a single key to a user when sharing lots of document with the user, and the user only needs to submit a single trapdoor when he queries overall documents shared by same owner. However, if a user wants to query over documents shared by multiple owner, he must generate multiple trapdoors to the cloud. How to reduce the number of trapdoors under multi-owners setting is a future work. Moreover, federated clouds have a attracted a lot of attention nowadays, but our KASE cannot be applied in this case directly. It is also a future work to provide the solution for KASE in the case of federated clouds.

X. REFERANCES

[1] D. Boneh, C. G, R. Ostrovsky, G. Persiano., "Public Key Encryption with Keyword Search", EUROCRYPT 2004, pp. 506C522, 2004.
[2] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing",Proc. IEEEINFOCOM, pp.534-542, 2010.
[3] D. Boneh, C. Gentry, B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys", Advances in CryptologyCCrypto 2005, pp.258-275, 2005.
[4] K. Ren, C.Wang, Q.Wang et al., "Security challenges for the public cloud", IEEE Internet Computing, volume. 16, no. 1, pp.6973, 2012.
[5] 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.9198.
[6] 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.
[7] 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.
[8] P. Van, S. Sedghi, JM. Doumen., "Computationally efficient searchable symmetric encryption", Secure Data Management, pp.87-100, 2010.
[9] 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.
[10] X.F. Chen, J. Li, X.Y. Huang, J.W. Li, Y. Xiang. "Secure Outsourced Attribute-based Signatures", IEEE Transactions on Parallel and Distributed Systems. DOI.ieeecomputersociety.org/10.1109/TPDS.2013.180, 2013.
[11] 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.
[12] Cheng-Kang Chu et.al, "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage", IEEE Transactions on Parallel and Distributed System, Volume:25, Issue: 2,Year: 2014.
[13] M. J. Atallah et.al, "Dynamic and Efficient Key Management for Access Hierarchies," ACM Transactions on Information and System Security (TISSEC), vol. 12, no. 3, 2009.
[14] J. Benaloh et.al, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," in Proceedings of ACM Workshop on Cloud Computing Security (CCSW '09). ACM, 2009, pp. 103– 114.
[15] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy- Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362–375, 2013.
[16] F. Guo, Y. Mu, and Z. Chen, "Identity-Based Encryption: How to Decrypt Multiple Cipher-texts Using a Single Decryption Key," in Proceedings of Pairing-Based Cryptography (Pairing '07), ser. LNCS, vol. 4575. Springer, 2007, pp. 392–406.
[17] D. Boneh, B. Lynn, H. Shacham. "Short signatures from the Weil pairing", Advances in Cryptology ASIACRYPT 2001, pp.514-532, 2001.
[18] L. B. Oliveira, D. F. Aranha, E. Morais, et l. "Tinytate: Computing the tate pairing in resource-constrained sensor nodes", IEEE Sixth IEEE International Symposium on Network Computing and Applications, pp. 318-323, 2007.
[19] C. Bosch, R. Brinkma, P. Hartel."Conjunctive wildcard search over encrypted data", Secure Data Management. LNCS, pp. 114-127, 2011.
[20] C. Dong, G. Russello, N. Dulay. "Shared and searchable encrypted data for untrusted servers", Journal of Computer Security, pp. 367-397, 2011.
[21] 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.

