Volume 6, Issue 7, July 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Survey of Different Searching Techniques for Encrypted Data Sharing on Cloud

Ajay Gawade*
Student M.E. (Computer Engineering)
Terna Engineering College, Nerul, University of Mumbai, Mumbai, Maharashtra, India

Rashmi Dhumal
Asst. Professor (Computer Engineering)
Terna Engineering College, Nerul, University of Mumbai, Mumbai, Maharashtra, India

Abstract- Usage of cloud storage has been increased tremendously by the use of internet technology. Billion of users are sharing their private data with their friends through a social network application using cloud storage. Organization and Business users also employ cloud storage due to better utilization of resources, low cost and easy access to data anywhere, anytime with the help of internet. The efficiency of sharing encrypted data with many users via public cloud storage may greatly reduce the security concerns over potential data leaks in the cloud. But, sharing group of selected files with the many users requires separate encryption keys to be used for encryption as well as for search. However, the data owner needs to distribute a large number of keys to users and user need to submit a large number of keyword trapdoors to perform searches over the shared files. This requires an efficient management of encryption keys for shared data access. Hence, we propose an efficient Aggregate Key Searchable Encryption scheme to share multiple files with single key. The data owner distributes a single key for sharing large number of files to the user and user submits a single trapdoor for querying the shared files.

Keywords- Cloud Storage, Searchable Encryption, Data Sharing, Trapdoor, Data Privacy.

I. INTRODUCTION

Cloud storage is gaining popularity in the recent years. Most of the online services are based on the cloud storage and anyone can access the online services from anywhere and anytime. Cloud storage provides an efficient, convenient, and on-demand access to large amounts of data shared over the Internet. Today, billions of users are sharing personal data, such as photos and videos, with their friends through social network applications using cloud storage. Business users or organization also uses cloud storage for better utilization of resources, low cost, pay per use and easy to access data anywhere in the world. For example, Google offers an online service known as Google Drive that enables a user to upload and download files from Google Drive at anytime and anywhere. Data privacy is a basic component of an organization in order to keep the information safe from various competitors. It helps to ensure the privacy of user personal information from others. Secured and timely transmission of data has been always an important aspect of an organization. Strong encryption algorithms and efficient key management techniques always help in achieving authentication, confidentiality and integrity of data and also reduce the overhead of the system.

Data sharing is an important functionality in cloud storage. For example, Social network application can allow their friends to view subset of their private photo, video, etc. An organization may allow his/her client access rights to particular data. But, the challenging problem is that how to efficiently share encrypted data via public cloud storage which greatly reduces potential data leakage in the cloud. First of all, user needs to be downloading the encrypted data from the cloud storage and decrypt them, and then send them to other people for sharing. But this approach requires secure communication storage and computational complexity which is clearly inefficient and impractical. Since, it loses the purpose of cloud storage. To overcome this problem, user should be able to give the access rights of the sharing data to other people so that they can access these data from the cloud storage directly. However, while enjoying the benefits of sharing data using public cloud storage, users are also increasing security related problem. For Example, data access by unauthorized users or misbehaving by the cloud operator can cause serious breaches of private data or business secrets. To address this problem, the data owner encrypts all the data before uploading them to the cloud. Then, the user will be retrieved encrypted data and decrypt it using decryption keys. Such system is called cryptographic cloud storage. But, encrypted data make it difficult for users to search and then selectively retrieve only the data containing given keywords. To overcome this problem, a general solution is using a searchable encryption scheme. Searchable encryption allows a user to encrypt its data in such a way that this data can be searched. In this scheme, a collection of files and an index of these files are encrypted with the potential keywords and upload them to the cloud. For retrieving data with a matching keyword, the user will send the corresponding keyword trapdoor to the cloud for performing search over the encrypted data.

Searchable encryption scheme with cryptographic cloud storage can accomplish the essential security requisite of the cloud storage. However, implementing such system for large scale application involving millions of users and billions of files may still practical issues for efficient management of encryption keys. Generally, sharing data with the different

users usually demand different encryption keys to be used for each file. For example, sharing a business files with certain employee or sharing a photo with certain friends in a social networking application on a cloud drive. However, this implies that the large number of keys that need to be securely distributed to users both for keyword searches and decryption of those files. In such scenario, a large number of keys not only distributed to users through secure channel, but also be securely stored as well as managed by the user in their devices. In addition, Users need to generate a large number of trapdoors and submitted to the cloud in order to perform a keyword search over many files. To address this problem, we require an efficient management of encryption keys for sharing data access. Hence, we propose an Efficient Aggregate Key Searchable Encryption Scheme. In this scheme, the data owner distributes a single key for sharing a large number of files to the users and user submits a single trapdoor for perform search over the files.

II. LITERATURE SURVEY

Over the years, the problem of searching over encrypted data has become an important issue in security and cryptography. Users outsource their data to un-trusted server. But user doesn't have trust on it as he loses control over the data. Hence the confidentiality of data is a major concern in cloud security. To achieve the confidentiality, data is stored in encrypted form on cloud storage. But, it is difficult to perform any operation on encrypted data. User need to download the encrypted data, and then decrypt it and performs operation on it. The resulting data need to be encrypted and uploaded on the cloud. Searching on encrypted data is a major issue in cloud. There are several solutions present in literature for searching over encrypted data are listed below.

2.1 Practical technique for search over encrypted data

D. Song et al. [1] were the first to propose a solution to store and retrieve data without any leak of information. They have used pseudo random function, pseudo random generator and sequential scan. Its supports controlled searching, query isolation and hidden searches.

The main objective of their work is:

1. Un-trusted server cannot able to search for arbitrary keywords without the user authorization.
2. Un-trusted server cannot learn anything more about the plaintext than the search result.
3. It supports hidden queries so that user may ask the un-trusted server to search for a secret word without revealing the word to the server.

There scheme is efficient, secure and practical and has no communication overhead. However, the sequential scan is not efficient if data size is large and also too slow in searching for a large number of documents. If we search frequently, server may be learned some information.

2.2 Searchable Symmetric Encryption

Although previous scheme provides search over encrypted data but it is too slow for searching as well as server learn some information. To address above problem, an efficient searchable symmetric encryption scheme was propose by R. Curtmola et al. [2]. Searchable symmetric encryption scheme use symmetric encryption which provides searching capabilities over encrypted data. An approach to provisioning symmetric encryption with search capabilities is called as secured index. An index is a data structure that stores document collections. Given a keyword, the indexes return a pointer to the document that contains it. User indexes and encrypts its document collection and sends the secure index together combine with encrypted data to the server. To search for a keyword, user generates and sends a trapdoor for that keyword to the server. The server uses the trapdoor to perform the search operation and recover pointers to appropriate document.

The main objective of their work is :

1. Symmetric searchable encryption can be achieved in its full generality.
2. Any types of queries can be performing over encrypted data.

This technique is more efficient and guarantees more security than previous schemes. But, this scheme not suitable for large scale data and support only single user.

2.3 Public Key Encryption with a keyword search

There are many applications that need more than a symmetric searchable encryption. Suppose a person wants to retrieve data for an existing encrypted document that he/she didn't store themselves. This situation would be impossible to solve unless he/she has a common secret key with the person who encrypted these data. To address this problem, Boneh et al. [3] proposed public key encryption with keyword search. In this scheme, document is encrypted using public key by the people who wants to store it in the un-trusted server. But the valid user can generate keyword trapdoor with the help of his private key can search over those encrypted data. For example, there are three entities which are data owner, user and server. A server is a cloud system who stores data and conducts searches, while data owner can be anyone who wants to share encrypted data under an authorized user's public key. Authorized user can query the server with keywords to perform search over the data owner documents. Anytime the server wants to conduct a keyword search, it needs the authorized user sends a trapdoor which includes his/her private key and keywords information. With the help of trapdoor, the server can test whether the keyword is present in the document or not. If the keyword matches, then returns desire document to user.

The proposed scheme provides better security, but increases the number of keys which makes a system complex. However, this scheme needs to support for trapdoor to identify keyword alone and it supports only single user.

2.4 Multi-user searchable encryption

The previous schemes provide search over encrypted data but it supports only single user. That is the only user who stored the encrypted data can perform a search and retrieve data. But in cloud storage environment, keyword search under multi-tenancy is a general scenario. In such scenario, where data owner would like to share a document to valid users and each user gives access right can perform keyword search over shared data. Tremendous work has been done on a multi user searchable encryption uses single key combined with access control [4], [5], [22]. Although they all adopt single-key combined with access control to achieve the goal. In MUSE schemes are constructed by sharing the document's searchable encryption key with all users who can access it, and broadcast encryption is used to achieve coarse-grained access control. In Broadcast encryption scheme, a broadcaster encrypts a message for some subsets S of users who are listening on a broadcast channel. Any user in S can use his/her private key to decrypt the message. This scheme allows multiple users can search over encrypted data. The main problem in MUSE is, how to control which users can access which documents and how to reduce the number of shared keys and trapdoors is not considered.

2.5 Multi-key searchable encryption

A multi-user searchable encryption scheme allows data owner to share encrypted data to users with access right to search over those encrypted data. In a multi user application, the number of trapdoors is proportional to number of documents. That is, user requires submitting a multiple trapdoor in order to perform search on shared encrypted data. To overcome this problem, Popa et al. [6] proposed a Multi-key searchable encryption scheme allows a user to search over shared data using a single keyword trapdoor instead of a multiple keyword trapdoor in the server.

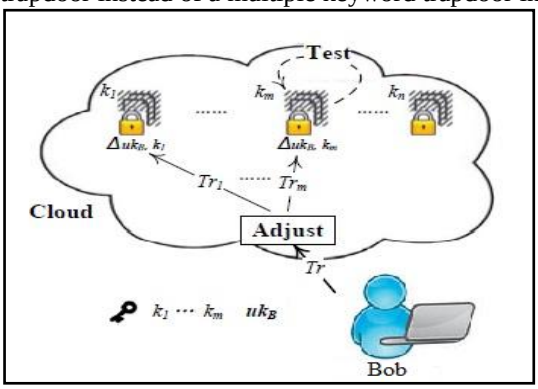

Fig. 1 Multi key searchable encryption

Suppose, Bob with key uk_B has m encrypted files on cloud server and each file encrypted under a key k_j for j = {1,…., m}.To allow the cloud server to adjust the trapdoor for each file with index j, Bob stores a public information called delta (Δ_{uk_B,k_j}) on the cloud server which is corresponding to both uk_B and k_j . As shown in Fig.1, when Bob wants to search for a particular keyword over all the files, he will use uk_B to compute a trapdoor for the keyword and submit it to the cloud server. The cloud server can use Δ_{uk_B,k_j} to convert a keyword trapdoor under key uk_B to a keyword trapdoor under k_j .This process known as adjust. In such a way, the cloud server can get the trapdoors for given keyword under $k_1 \ldots\ldots k_m$ while only receiving single trapdoor from Bob. Then he performs a traditional single key search with the new trapdoor. However, this approach still requires submitting multiple keys to user to generate the single trapdoor.

2.6 Key aggregate encryption for data sharing

In traditional approach, to share group of documents with different encryption keys with the same user, data owner require to distribute all the keys to user. To address this problem, Chow et al. [7] proposed a key aggregate cryptosystem to reduce the number of distributed data encryption keys.

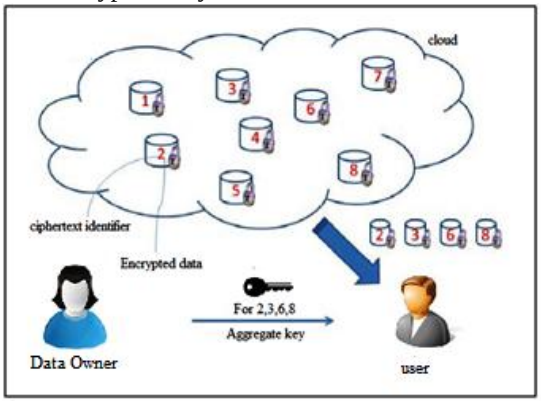

Fig. 2 KAC for data sharing in cloud storage

In this scheme, data owner generates an aggregate key for the user to decrypt all the documents. To allow a set of documents encrypted by different keys to be decrypted with a single aggregate key, user could encrypt a message not only under a public-key, but also under the identifier of each document. The data owner holds a master-secret key, which can be used to extract secret keys for different classes. The extracted key can be an aggregate key which is as compact as a secret key for a single class, but aggregates the power of many such keys. That is the decryption power for any subset of cipher-text classes. As shown in Fig 2, data owner shares files with identifiers 2, 3, 6 and 8 with the user by sending a single aggregate key. User can download the encrypted files from data owner's cloud storage and then use this aggregate key to decrypt them. This scheme is inspired by the broadcast encryption scheme [8].Although this scheme allows efficiently delegating the decryption right to other users, but this method doesn't not support search over encrypted data.

III. PROPOSED SYSTEM

The existing techniques for searching over encrypted data uses single key to search the document either for single user or multiple users. Still there is a need of an efficient searching technique in which data owner provides a single key for sharing a group of documents and authorized users submit a single trapdoor in order to perform keyword search over the shared documents. Hence, we propose an Efficient Aggregate Key Searchable Encryption scheme to search multiple documents using single aggregate key.

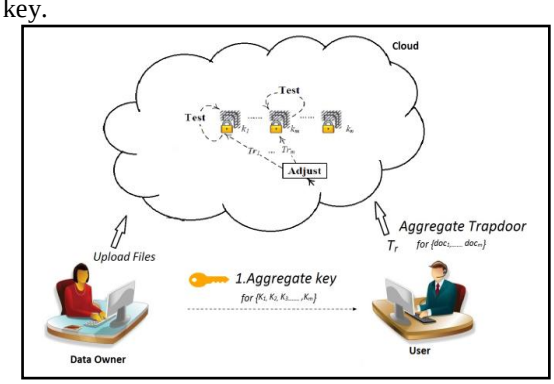

Fig. 3 Architecture of proposed system

In a proposed system, data owner distribute a single aggregate key for sharing m documents with user and user submits a single aggregate trapdoor to the cloud server. The cloud server can use this aggregate trapdoor and some public information to perform keyword search and return the result to user. Therefore, the delegation of keyword search right can be achieved by sharing the single aggregate key in our system. Our proposed system consists of seven algorithms. First of all, the cloud server would generate public parameters of the system through the *Setup* algorithm, and these parameters can be reused by various data owners to share their files. For each data owner, he/she should generate a public key and master-secret key pair using the *Keygen* algorithm. Keywords of each file can be encrypted via the *Encrypt* algorithm with the searchable encryption key. Then, the data owner can use the master-secret key to generate an aggregate searchable encryption key for selected files via the *Extract* algorithm. The aggregate key can be distributes via secure e-mails to authorized users to access those files. After that, as shown in Fig.3, an authorized user can generates a keyword trapdoor via the *Trapdoor* algorithm using this aggregate key, and submit the trapdoor to the cloud sever. After receiving the trapdoor, to perform the keyword search over shared files, the cloud server will run the *Adjust* algorithm to generate the right trapdoor for each file, and then run the *Test* algorithm to test whether the file contains the given keywords.

3.1. Advantages of Proposed System

1. Preserving data privacy and confidentiality.
2. Data is stored secure in cloud server.
3. Data owner distributes a single key for sharing large number of documents to the user.
4. User needs to submit a single trapdoor for performing keyword search over number of shared files.

IV. METHODOLOGY

An Efficient Aggregate Key Searchable Encryption Scheme as follows:

1) Setup (1^{λ}, *n*): this algorithm used by the *cloud server* to initialize system parameters as follows.

- Generate a bilinear map group system B= (p, G, G_1, e (·, ·)), where p is the order of G and $2^{\lambda} \leq p \leq {}^{2\lambda+1}$.
- Set n as the maximum possible number of documents which belongs to a data owner.
- Pick a random generator $g \in G$ and a random $\alpha \in Z_p$, and computes $g_i = g^{(\alpha^i)} \in G$ for $i = \{1, 2, \cdots, n, n+2, \cdots, 2n\}$.
- Select a one-way hash function H: $\{0, 1\}^* \rightarrow G$.

Finally, *cloud server* publishes the system parameters *params* = (*B*, *PubK*, *H*),
Where $PubK = (g, g_1, \ldots\ldots, g_n, g_{n+2}, \ldots, g_{2n}) \in G^{2n+1}$

2) Keygen : data owner uses this algorithm to generate his/her key pair. It picks a random $\gamma \in Z_p$, and outputs:

$$pk = v = g^{\gamma}, msk = \gamma \quad \text{..... (1)}$$

3) Encrypt (*pk, i*): data owner uses this algorithm to encrypt data and generate its keyword cipher-texts when uploading the *i*-th document. o generate the keyword cipher-texts, this algorithm takes as input the file index $i \in \{1,..., n\}$, and:

- Randomly picks a $t \in Z_p$, as the searchable encryption key k_i of this document.
- generates a delta Δ_i for k_i by computing:

$$c_1 = g^t, c_2 = (v \cdot g_i)^t \quad \text{..... (2)}$$

- for a keyword w, outputs its ciphertext c_w as:

$$c_w = e(g, H(w))^t / e(g_1, g_n)^t \quad \text{..... (3)}$$

Where c_1, c_2 are public and can be stored in the cloud server.

4) Extract (*msk, S*): data owner uses this algorithm to generate an aggregate searchable encryption key. For any subset $S \subseteq \{1,\ldots,n\}$ which contains the indices of documents, this algorithm takes as input the owner's master-secret key *msk* and outputs the aggregate key k_{agg}.
By computing:

$$k_{agg} = \prod_{j \in S} g_{n+1-j}^{\gamma} \quad \text{..... (4)}$$

To delegate the keyword search right to a user, data owner will send k_{agg} and the set *S* to the user.

5) Trapdoor (k_{agg} , *w*): the user uses this algorithm to generate the trapdoor to perform keyword search. For all documents which are relevant to the aggregate key k_{agg} , this algorithm generates the only one trapdoor Tr for the keyword w by computing: Then, the user sends (Tr, S) to the cloud server.
By computing:

$$Tr = k_{agg} \cdot H(w) \quad \text{..... (5)}$$

Then, the user sends (Tr, S) to the cloud server.

6) Adjust (*params, i, S, Tr*): the cloud server uses this algorithm to produce the right trapdoor. For each document in the set *S*, this algorithm takes as input the system public parameters *params*, the document index $i \in$ S and the aggregate trapdoor *Tr*, outputs the right trapdoor Tr_i by computing:

$$Tr_i = Tr. \prod_{j \in s, j \neq i} g_{n+1-j+i} \quad \text{..... (6)}$$

Then, the cloud server will use **Test** algorithm to complete the keyword search.

7) Test (Tr_i , *i*): the cloud server uses this algorithm to perform keyword search over the *i*-th document. For the *i*-th document, this algorithm takes as input the adjusted trapdoor Tr_i ,the $\Delta_i = (c_1, c_2)$ relevant to its searchable encryption k_i and the subset *S*, outputs true or false by judging:

$$c_w \stackrel{?}{=} e(Tr_i , c_1) / e(pub, c_2) \quad \text{..... (7)}$$

Where $pub = \prod_{j \in s} g_{n+1-j}$. Note that for efficiency consideration, the pub for the set *S* can be computed only once.

Table 1. Comparasion of Various Techniques

Techniques	User	Encryption type	Number of key	Trapdoor
Practical technique for search over encrypted data [1]	Single user	Symmetric	Single key per document	Not used
Symmetric searchable encryption [2]	Single user	Symmetric	Single key per document	Multiple Trapdoor
Public key encryption with keyword search [3]	Single user	Asymmetric	Two key per document	Multiple Trapdoor
Multi user searchable encryption [4] [5] [22]	Multi user	Asymmetric	Single key per document	Not used
Multi key searchable encryption [6]	Multi user	Asymmetric	Single key per document	Single Trapdoor
Key aggregate cryptosystem for group data sharing on cloud [7]	Multi user	Asymmetric	Single key per group of documents	Not used

V. CONCLUSION

Nowadays, everyone uses cloud services to outsource their private data via public cloud storage because of many advantages over traditional method. To ensure privacy of outsourced data, user encrypts their private data before uploading to cloud so that server doesn't know anything about the data. But, how to efficiently share selectively encrypted data with valid users and how the server can perform search operation without knowing the user data. With the traditional approach, data owner need to be securely distributing such large number of keys to user both for encryption and search and valid users require to store them securely and generate keyword trapdoor in order to perform searches over shared data. However, such system required secure channel, storage and increases computational overhead.

From the above table 1, Practical technique for search over encrypted data scheme allows efficient search over encrypted data. But, the sequential scan is inefficient if data size is large and also takes more time for searching large number of documents. Symmetric searchable encryption and public key encryption with keyword methods provide an efficient search over encrypted data with multiple keys for single user. In MUSE scheme, multiple users can search over shared data with multiple keys for multiple users. In MKSE scheme, multiple users can search over shared data using a single trapdoor. The major issues in MKSE is, data owner requires distribute multiple keys to the user to generate single trapdoor. Key aggregate cryptosystem allows decrypting the group of document using single aggregate key. But, this method doesn't not support search over encrypted data. Hence, we propose an efficient Aggregate Key Searchable Encryption scheme to share multiple files with single key. The data owner distributes a single key for sharing large number of files to the user and user submits a single trapdoor for querying the shared files.

REFERENCES

[1] X. Song, D. Wagner, A. Perrig. "Practical techniques for searches on Encrypted data", IEEE Symposium on Security and Privacy, IEEE Press, pp. 44C55,2000

[2] R.curtmola, J. Garay, S. Kamara, R.Ostrovsky. "Searchable symmetric encryption: improved definitions and efficient constructions", in: Proceedings of the 13th ACM conference on Computer and Communications Security, ACM Press, pp.

[3] D. Boneh, C. G, R. Ostrovsky, G. Persiano. "Public Key Encryption with Keyword Search", Eurocrypt2004, pp.506C522, 2004

[4] G 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.

[5] F. Zhao, 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.

[6] R. A. Popa, N. Zeldovich. "Multi-key searchable encryption" cryptology ePrint Archive, Report 2013/508, 2013.

[7] C.Chu, S.Chow, W.Tzeng, et al. "Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage", IEEE Transactions on Parallel and Distributed Systems, 2014.

[8] D. Boneh, C. Gentry, B. Waters. "Collusion resistant broadcast encryption with short Cipher-texts and private keys", Advances in Cryptology CCRYPTO 2005.

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

[10] 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.

[11] NIST Privacy and Security guidelines (2012) NIST. Source: http://csrc.nist.gov/publications/nistpubs/800-144/SP800-144.pdf. Accessed on Oct 2012

[12] B. Lynn, "The pairing-based cryptography library," http://crypto.stanford.edu/pbc/

[13] A. D. Caro and V. Iovino, "JPBC: Java Pairing Based Cryptography," 2011.

[14] P. Van,S. Sedghi, JM. Doumen. "Computationally efficient searchable symmetric encryption", Secure Data Management, pp. 87-100, 2010.

[15] 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.

[16] M. Chase and S.S.M. Chow, "Improving Privacy and Security in Multi-Authority Attribute-Based Encryption," Proc. ACM Conf. Computer and Comm. Security, pp. 121-130. 2009

[17] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Advances in Cryptology (CRYPTO '01), vol. 2139, pp. 213-229, 2001.

[18] P. Van,S. Sedghi, JM. Doumen. "Computationally efficient searchable symmetric encryption", Secure Data Management, pp. 87-100, 2010.

[19] 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.

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

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

[22] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data,"Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06),pp. 89-98, 2006

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

