UNIQUE CLOUD STORAGE FOR PRIVACY PRESERVING USING MD5

[1]V. Lakshmana Rao, [2] A. Sowmya, [3] J. Priyanka, [4]A.Divya Sai, [5] K.V.S.Swetha Devi
[1]Assistant Professor, [2]Student, [3]Student, [4]Student, [5]Student
Department of Computer Science and Engineering,
Gayatri Vidya Parishad College of Engineering for Women, Visakhapatnam, India.

ABSTRACT

In current days, the cloud technology gets a rapid increase of user attention by several small scale and large scale companies which include Software, Business Process Outsourcing (BPO), Healthcare, educational institutions and a lot more fields. It became popular to mange personal data for the economic savings. But for the privacy, the sensitive data in the cloud is must be encrypted before uploading it into the cloud server. Moreover the cloud server is dishonest and it cannot give accurate search results for user who wants to access the actual files. To solve this problem, we proposed and analysed the importance of bloom filter over encrypted data in cloud to provide the accurate searching of data for the cloud users. Here we designed a mechanism called fine-grained query results verification. In this mechanism, over a set of given encrypted query results set, the query user verifies the quality of each data file in the set and authorize the data by using Message Digest (MD5) algorithm. A short signature key is generated in this MD5 algorithm which is used to verify the data authentication.

Keywords: Bloom filter, cloud server, fine-grained query result verification mechanism, data owner, data user.

I. INTRODUCTION

The demand for cloud storage is rapidly increasing these days, the current clouds storage is almost centralized along with details of data owners and data users. The data is clearly visible by the cloud server. All the data can be viewed and accessed by anyone who is having an account within the cloud. So the data is not secured, that is anyone can modify the data. If any user tries to modify the cloud data, then the person cannot be identified in cloud server. The problem is that, generally we use MD5 algorithm for the integration of the data in the cloud but in the existing clouds there is no concept like applying this MD5 algorithm for uploading data in the cloud.

They used stream cipher and block cipher operations. They even provided searching techniques to find the encrypted data in cloud for security but they haven't authenticated whether their file is modified

or not when it reached to the user [3]. Even wildcard-based technique is used to construct the storage efficient fuzzy keyword sets by exploiting a significant observation on the similarity metric of edit distance. These fuzzy keywords used to search encrypted data [4]. Some tried to authenticate the file to check whether there are any modifications by using vector space model combined with the TF X IDF rule and a cosine similarity is used to measure the similarity of files. The files are uploaded file and requested file to download [1].

In our work we proposed a secure search techniques for encrypted data in cloud to allow an authorized user to query data files in cloud with certain keyword of the person's interest to download a file which is done while uploading a file into cloud for faster retrieval of file. We use a secure index is a data structure that allows a query with a trapdoor for a word x to test in O (1) time [6]. The encrypted query keyword is received by the user through mail. The goal is achieved by obtaining secure hash code (i.e, verification object) for files in the cloud. Here we proposed a hash code generation technique for authentication of verification object generated while uploading and retrieving of files. Hash code is generated using message digest algorithm.

In our paper work we used DriveHQ as live cloud to secure the data. The DriveHQ is the live cloud we have used to store these data files securely through File Transfer Protocol (FTP) connection. This cloud also supports the FTP for the data file storage in the cloud. This DriveHQ cloud is offering reliable cloud file storage. The file manager of this cloud can manage, share, sync, collaborate and publish local or cloud files with unusual reliability and ease [2].

The paper is organised as follows: In Section2 first we describe the proposed method architecture, bloom filter and then the algorithms that we mainly used Advanced Encryption Standard (AES) for encryption and decryption and MD5 for hash code generation. Then in the same section we describe the significance of the three modules in our proposed system. Next we explain the detailed process steps of our proposed system mechanism. In Section 3 we explained the results of outputs we got and analysed the effectiveness of the data even if it gets modified. Our conclusions and future enhancement are presented in Section 4 and Section 5 respectively.

II. PROPOSED SYSTEM

There are several architectures that combine recent and non-cryptographic primitives [5] but in our paper work we want to secure our data from unknown user who tries to modify the data in cloud.

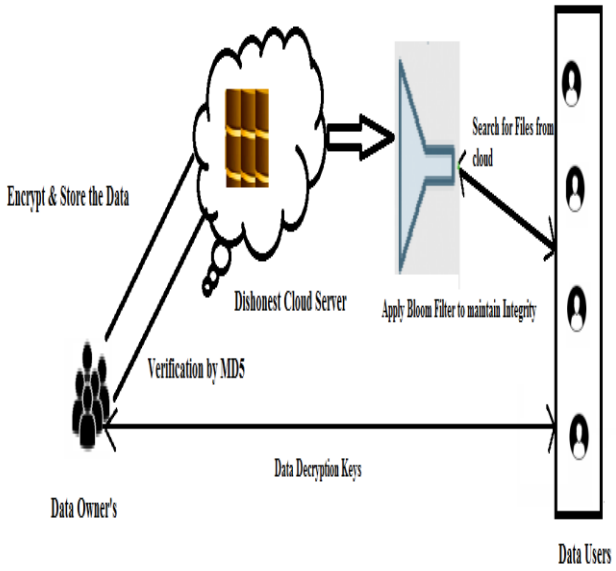

Figure 1: Proposed system architecture

Bloom Filter is a searching technique used to search file in cloud by data user using keyword to search a file. Owner uploads the file into cloud with certain keyword. User searches the file with keyword. The main concept is user must receive the same file as owner uploaded. For security user can view or download the file after getting permission from owner because data is in encrypted manner.

AES(Advanced Encryption Standard): AES is a block cipher which operates on block size of different key sizes to encrypt and decrypt the data. To encrypt it repeats four major functions in each round to produce cipher text there are: Sub Bytes, Shift rows, Mix Columns and Add key. Number of rounds depends upon the key size. If key size is 128 then 10 rounds are performed. Similarly for 192-12 rounds and 256-14 rounds. In our technique we encrypt the data before storing it in cloud and we used 128 key size.

MD5 algorithm (Message Digest 5): It accepts input without considering its length and produce fixed length output. The output is the hash code it is used for authentication of data. In our work we authenticate whether we received the file with modifications or not. The output contains for 32 hexadecimal digits.

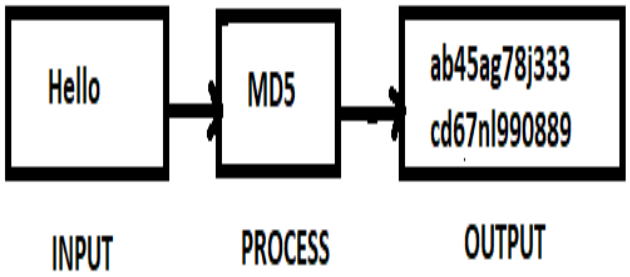

Figure 2: MD5 Hashing

Our Proposed System has 3 modules –

A. Data Owner
B. Data User
C. Cloud server

A. Data Owner

1. They can upload the file into the cloud.
2. They can also view upload files, download files.

3. These gives approval to user who want to download the file by sending verification object to user mail.

B. Data User

1. They can login into the site with secret key received through mail after registering into the site.
2. They can request the owner to download file by searching all the files which are uploaded by the owners.
3. They search the file which they required with keyword.
4. They can download the file only after getting the approval from the data owners.

C. Cloud Server

1. The cloud server is able to view all the details of file.
2. Data owner can edit or modify the files in cloud server.
3. The cloud server contains all downloaded history.

The detailed process of our proposed system steps are as follows:-

1. The data owner wants to upload file we have to connect this to the cloud using FTP connection to the cloud.
2. Upload file: The file which is uploaded into cloud contains encryption data. The encryption is done using Advanced Encryption Standard (AES) algorithm. The secret key is generated by using key generator function.
3. The data user who tries to login will receive a secret key like One Time Password (OTP).
4. The data user searches the file with the search keyword (related to file the person wants to download).
5. In order to download a file the data user must select a file and request to the data owner.
6. The data owner approves the request and sends verification object to the user through mail .We generated the mail using javax.mail.
7. The verification object contains information like trapdoor key, decryption key, and verification object.
8. The verification object is the hash code of the file generated by using message digest MD5 algorithm.
9. The data user enters the trapdoor key and decryption key to download the file. This person can also verify whether the file is modified or not by verification object.
10. Although the file is modified, data user is able to download the file same as uploaded by owner.

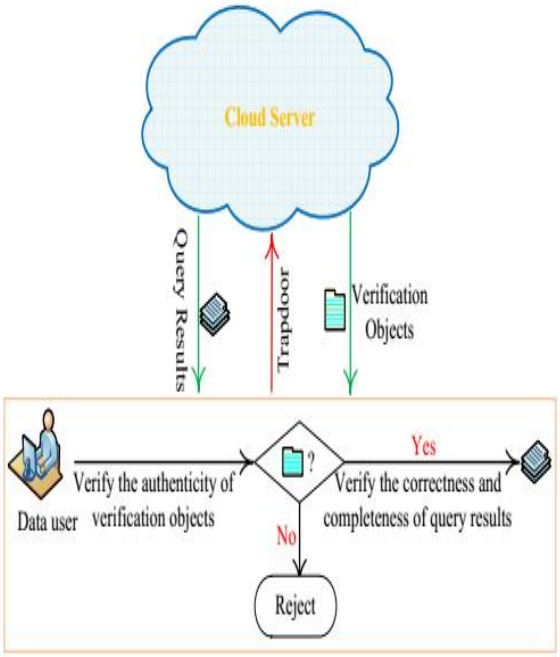

Figure 3: Procedure for File Download

III. RESULTS AND ANALYSIS

We have created web pages using JavaScript programming language. This JavaScript allows the web pages to become interactive. JavaScript works on client side as well as server side.

The first step in our process is data owner has to register into our site to upload any files into the cloud. Data owner must register into the site by giving up all the details in order to store data in cloud. Then owner has to login into the site so that he is able to find the upload button. The data owner has to upload the file by giving the keyword to the file so that data user can easily find the file. The file is stored in DriveHQ cloud as it is a free cloud. When we open the file in cloud it is in encrypted manner. For security we are encrypting the file.

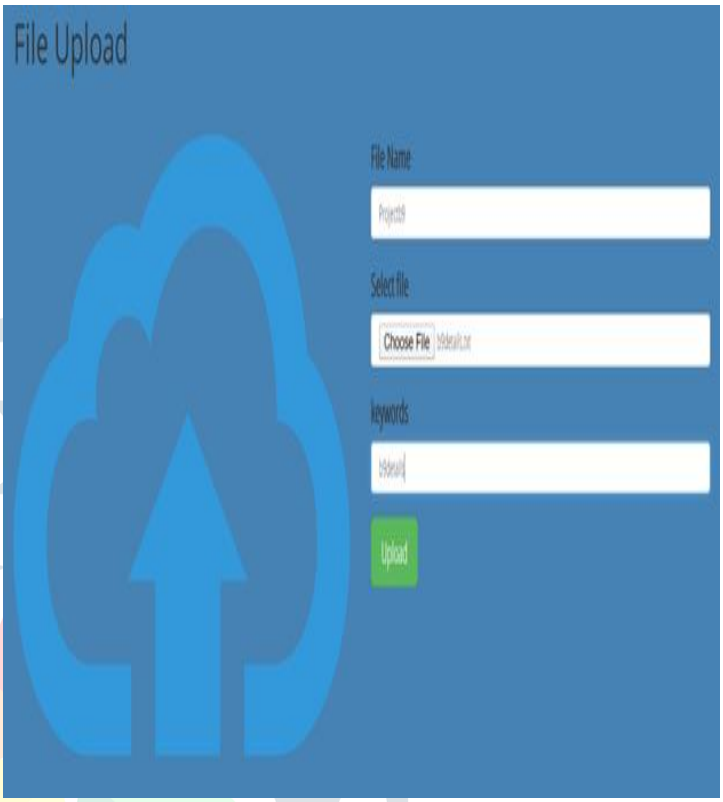

Figure 4: Data Owner uploading the file

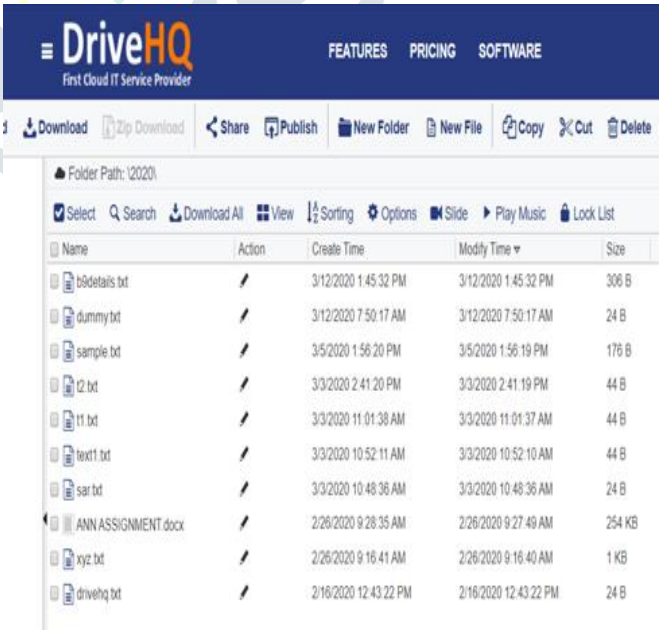

Figure 5: File Storage in DriveHQ Cloud

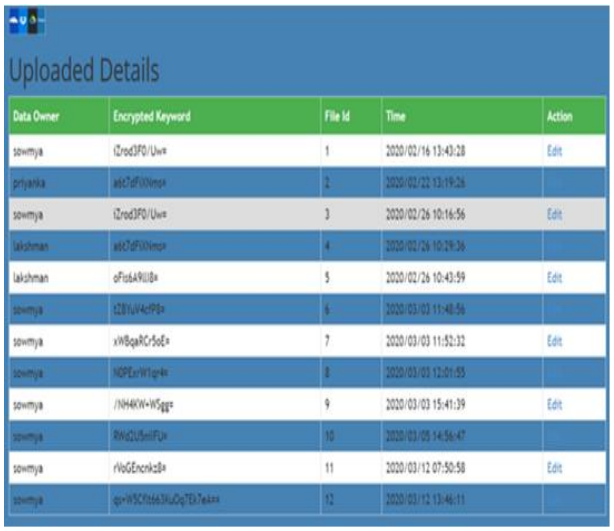

Figure 6: List of all downloaded files and uploaded files in the cloud

Next step is the data user register into site which we have like name, email and other details. Then the data user login to the site with secret key every time like one time password. The secret key is sent to the data user through mail.

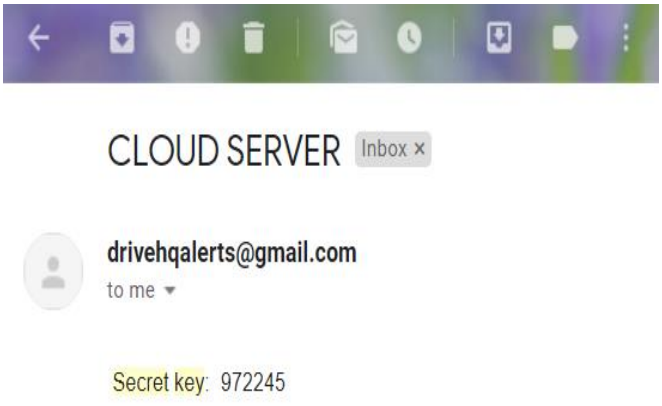

Figure 7: Secret key received through mail

After login to the site data user finds search files button. When data user click that button, the person receive a web page for searching the file.

Then data user asks permission to the data owner by clicking request button in action column. Then in data owner site in requested files web page data owner receives file name and file access granted in action column this means that somebody asks permission to owner for file download . Immediately data owner accepts the request asked by data user. Then user gets Approved in action column. Then the data user receives the mail which contains information like decryption key, verification object and trapdoor key to download the file.

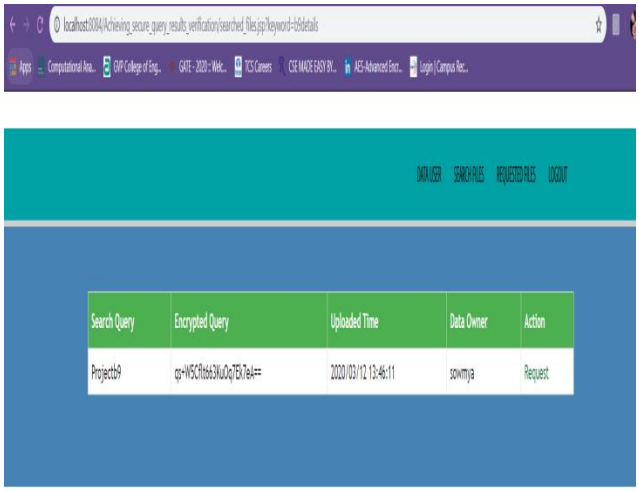

Figure 8: Data user requesting a file from the Data Owner

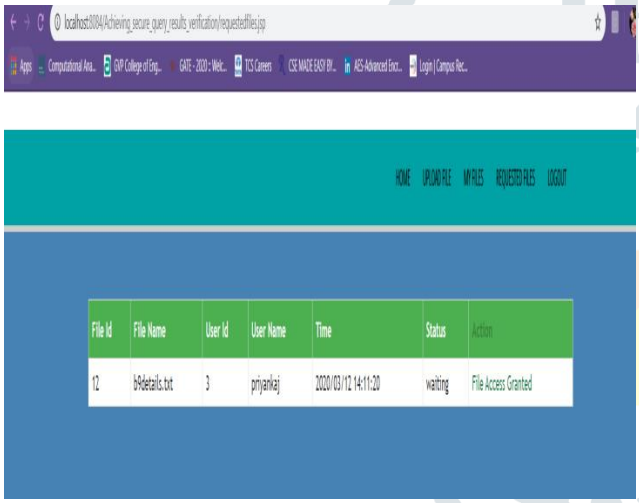

Figure 9: Data Owner granting access to the Data User

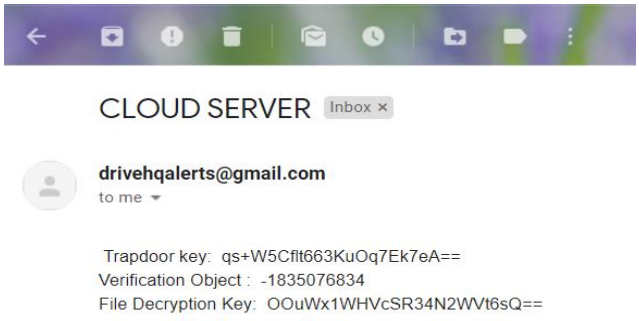

Figure 10: Data User getting mail from the Cloud to access the file

In order to download the file from cloud the data user has to enter the trapdoor key which is nothing but the hash code of file name generated by using MD5 algorithm. If the data user correctly enters trapdoor key then he is able to download the file or else not.

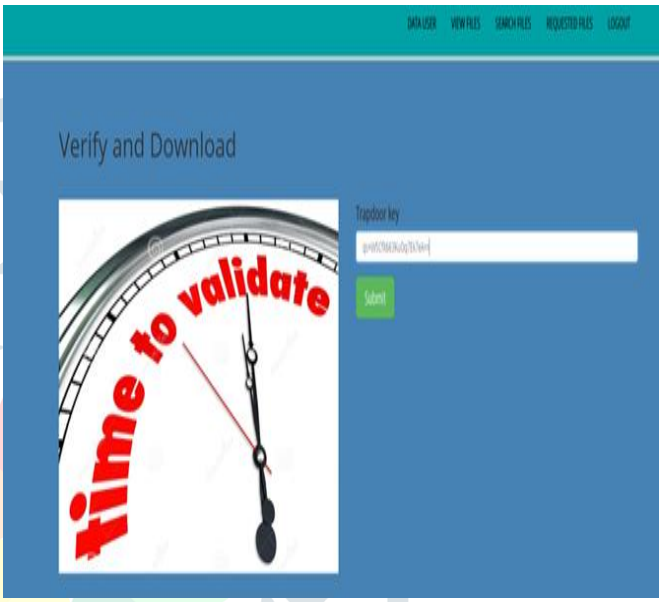

Figure 11: Verification by the Data User - Trapdoor key

After correctly entering the trapdoor key data user enters into download page where the person checks the verification object to know whether anybody tries to modify the file. If uploaded verification object and received file verification object are same then data user understood that nobody tries to modify the data in file.

If verification objects of uploaded file and received file are different then it

means that somebody tries to modify the data.

Figure 12: If nobody modifies the data

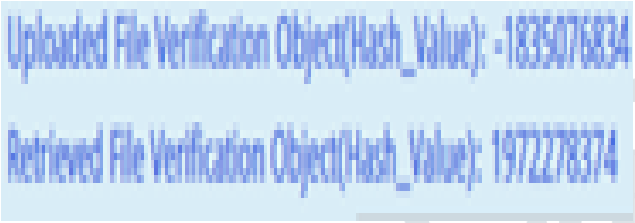

Figure 13: If anybody modified the data

In text area the data is in encrypted format. If data user correctly enters decryption key then the person can download the file or else data user is unable to download the file.

After correctly entering the decryption key then data user is able to see the decrypted data of the file that is original content of the file.

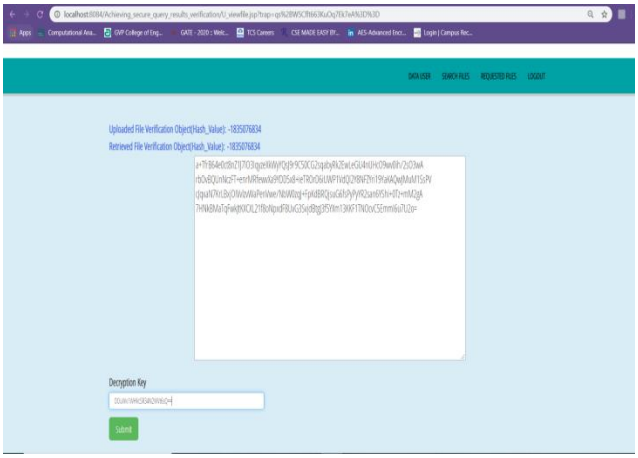

Figure 14: Entering the decryption key to download the file

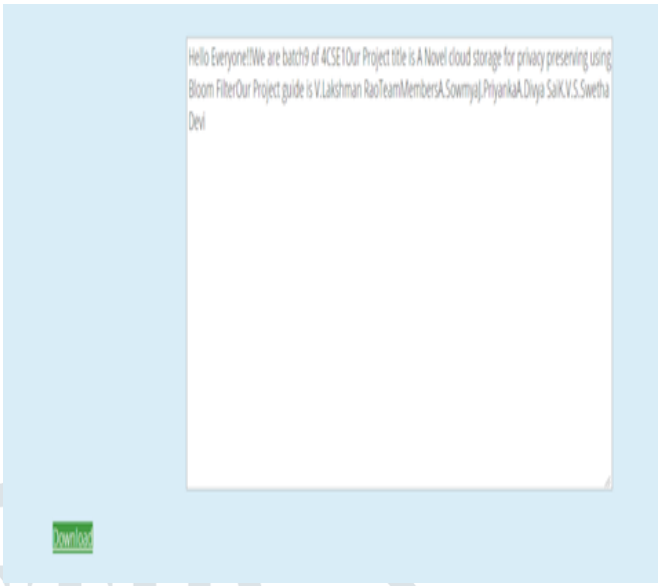

Figure 15: The data in the file getting displayed and also to download the file

In the above figure the original content of the file viewed in text area of correctly entering the decryption key by data user which the person had received through mail from data owner.

The data user can view the file in text area or we can download the file and save for future reference. To download there is download button in the website. If data user clicks the download button then the file is downloaded into the download folder of the user local machine where the person can see the file.

IV. CONCLUSION

Hence we proposed a secure, easily integrated fine-grained query results

verification mechanism for searching in a secure manner over encrypted cloud data. Our method will verify whether we received the file same as the original file uploaded by the data owner or anybody tries to modify the data in between the time of uploaded and downloading the file. We store the data in the cloud in encrypted format. A hash code generation technique is used to for authentication of verification object generated while uploading and retrieving of files. Hash code is generated using message digest algorithm.

V. FUTURE ENHANCEMENT

This mechanism can be used in the third party systems to run the online bank transactions securely. Also this can be used in health care sector for the patients retrieving the data securely. This can further enhanced by storing the other type of format files like images, audios, videos into the cloud and performing the encryption, decryption and also the data authentication on these type files.

VI. REFERENCES

[1] Cheng guo, Ruhan zhuang, Chin-chen chang and QiongQiong yuan, "Dynamic Multi-Keyword Ranked Search Based on Bloom Filter Over Encrypted Cloud Data" IEEE Access, March 13, 2019.

[2] The data files storage in the Live Cloud https://www.drivehq.com/

[3] D. X. Song, D. Wagner, and A. Perrig, ''Practical techniques for searches on encrypted data,'' in Proc. IEEE Symp. Secur. Privacy, May 2000, pp. 44–55.

[4] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, ''Fuzzy keyword search over encrypted data in cloud computing,'' in Proc. IEEE INFOCOM, San Diego, CA, USA, Mar. 2010, pp. 1–5.

[5] S. Kamara and K. Lauter, "Cryptographic cloud storage" in Springer RLCPS, January 2010.

[6] E.-J.Goh, "Secure indexes" IACR ePrint Cryptography Archive, http://eprint.iacr.org/2003/216, Tech. Rep., 2003.