International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 05, May-2015

Efficient Keyword based File Retrieval in Cloud

Nivedita R H
M.Tech,
Computer Science & Engineering
P. A College of Engineering
Mangalore -574153

John Prakash Viegas
Asst. Professor CSE Department
P. A College of Engineering
Mangalore-574153

Amaresh Patil
Asst. Professor CSE Department
AGMR College of Engineering & Tech
Varur, Hubli- 581207

***Abstract*— An organization dumps all files into the cloud so that it can be easily available to all its employees, but retrieving a particular file from plenty of files is bit difficult task so, to make it easier one can apply efficient keyword based file retrieval in cloud mechanism. A data owner collects all the files and encrypts each file using homomorphic encryption method and these encrypted files are stored in a bucket of the cloud securely. The keywords are extracted from every file and a dictionary is formed, this dictionary is used to generate an index file. An index file is created for every file, which is further used for keyword based file retrieval.**

Keywords— ***Index file, homomorphic encryption, keyword based file retrieval.***

I. INTRODUCTION

Cloud computing is a technological advancement that focuses on the way we design computing systems, develop applications, and leverage existing services for building software. There are three kinds of clouds, public cloud, private cloud and hybrid cloud. And cloud provides three services, IaaS(Infrastructure as a Service), SaaS(Software as a Service) and PaaS(Platform as a Service). Because of the advantages of cloud like flexibility, cost-effectiveness and scalability, many of the enterprises try to share the data with the cloud. The paper concentrates on private cloud, which is dedicated to particular organization, in an organization most used service is storage which is associated with IaaS of the cloud. For example, if an organisation is associated with private cloud. Thus, the members of the organisation are authorised to share data with the cloud. While sharing, each file is associated with a group of keywords and members can retrieve files by requesting the cloud with the use of keyword.

As organizations store common shared data in their private cloud such that the users of organization can access the shared data in cloud as there level of authority. so there are two issues to be concerned .

1. Security: the data should be stored in a cloud without losing its confidentiality and integrity.
2. Time efficiency: the time taken to retrieve a file from cloud based on the level of authority in an organization is a major concern. As in existing system file retrieval is based on query that is query for requesting file is processed and return corresponding result set to access the files. This query processing time is a additional time to retrieval time, so this paper presents the efficient approach of key word based file retrieval which will reduce the query processing time.

II. LITERATURE REVIEW

The main intention of this paper is to provide efficient keyword based search services while preserving information confidentiality in the cloud. Some of the existing works are:

[1] The authors Rajkumar Buyya, Christian Vecchila and S. Thamarai Selvi, introduces the fundamental principles of cloud computing and its related paradigms in the book "Mastering Cloud Computing Foundations and Applications Programming". The book discusses the concepts of virtualization technologies along with the architectural models of cloud computing. It presents prominent cloud computing technologies that are available in the marketplace, including the Aneka Cloud Application Platform. The book contains chapters dedicated to discussion of concurrent, high-throughput, and data-intensive computing paradigms and their use in programming cloud applications [5] Private searching scheme or Ostrovsky schema is proposed by Ostrovsky where a file is retrieved from an unsecured network, without leaking the information but this scheme consumes more computational time and cost because it has to compute on every individual file stored in the cloud which leads to the performance bottleneck. Reference [2] proposes one of the secure encryption technique called as Homomorphic encryption. The homomorphic encryption method is able to perform operations on encrypted data without decrypting them, using this method the files can be securely stored in the cloud without any leakage of information.[4] Amazon web service (AWS) provides the IaaS for the storage purpose for the valid users of the cloud where an organization can upload multiple variety of files to cloud.

III. ARCHITECTURE

The architecture provides the overview of the system behaviour and the process it follows. The organization files are securely loaded to the storage space in the cloud, as organization loads multiple number of files it is very difficult to retrieve a particular file which is required so, to make it simpler one can adapt the keyword based file retrieval, where the system takes the raw files as input and find the respective index file for each input file based on dictionary and encrypts all input files before storing to cloud. Only the index file and encrypted input files are stored in the cloud.

The user of the organization can retrieve the files based on keyword based file retrieval mechanism rather than using a query based file retrieval. In keyword based file retrieval mechanism user passes a search keyword which is contained in any of the files uploaded in cloud. The search keyword is taken as the input and identifies the files that contain the search keyword using index file and retrieve the matched encrypted files from cloud and the same files are decrypted and forwarded to the user.

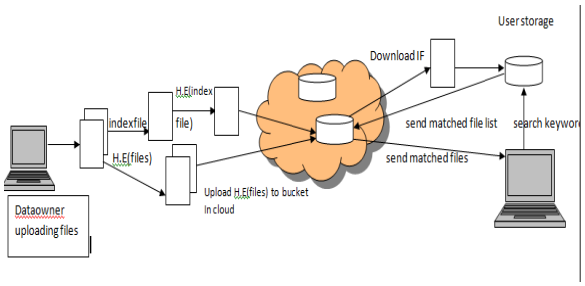

Figure 1 Architectural diagram.

IV. METHODOLOGY

The system is divided into two modules 1. Uploading files 2. Retrieval of files.

1. Uploading of files: Data owner collects all the text files and encrypts each text file using public key of homomorphic encryption. An index file is created for every text file, which is further used for keyword based file retrieval. Both index file and encrypted files are stored in the storage unit of the cloud.

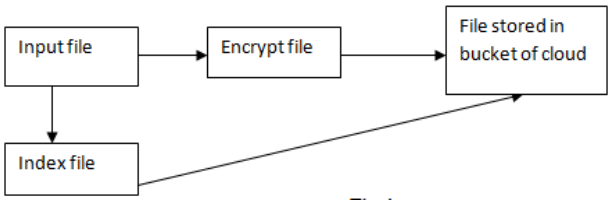

Figure 2. Uploading of files

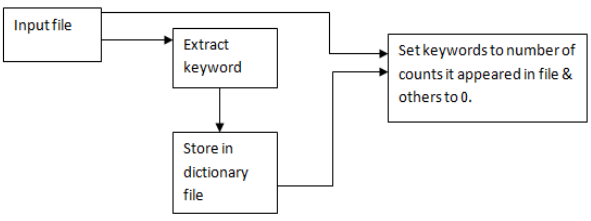

Figure 3. Index file creation

2. Retrieval of file: the index file is downloaded from the cloud. User gives a search keyword, with the help of which he can retrieve the files containing that search keyword. An Index Buffer is created for a given keyword with respect to dictionary file. Further we map the Index Buffer with the downloaded index file, by which we get the list of files containing the keyword. Now the user can download mapped encrypted files from cloud and decrypt it.

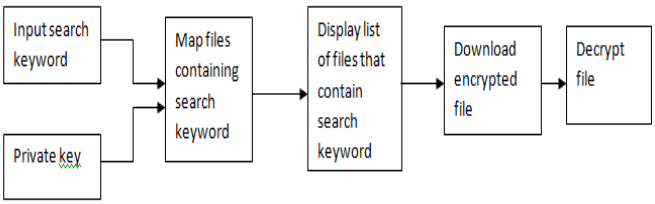

Figure 4 Retrieval of file

V. IMPLEMENTATION

This concept is implemented using JAVA programming language, because its object oriented and required API's are in JAVA." Amazon S3" [4] cloud is been used for the storage purpose by the mean of the API "aws-java-sdk-1.4.3.jar" to store encrypted file and keyword based index file of input file. Here text(.txt) files and dictionary file are used as input parameters. For every text file being uploaded to cloud a dictionary is created where it will collect all keywords and stores them in a different file. When the search keyword is given, if that keyword is present in the dictionary then, further the files containing that keyword is mapped using index file and returned. The list of files returned is based on the number of times that keyword is present in that particular file, if the file has more counts of that keyword then it will given higher priority. This approach is called keyword based file retrieval. For efficient way of file retrieval we use index file rather than using string matching approach. The index file is created for every input file, if a dictionary word or keyword is present in that file then the count of that word means, number of times the keyword is present in a file is written in index file and other words which are not dictionary words are made 0.

Every authorised user of [4] Amazon S3 creates the bucket to store the text files. The given input text file is encrypted using Homomorphic encryption algorithm [2] and this encrypted file and keyword based index file is been uploaded in allocated bucket in cloud. The procedure for creating dictionary and index file is explained below.

// Algorithm: Compute dictionary (input files)
Start
Step1: for each input files do step 2 and 3.
Step 2: split files keywords and store it in array.
Step 3: create dictionary by using union of all keywords array which is calculated above.
End
The above algorithm provides the dictionary for all input text files.
//Algorithm: index file (files, dictionary)
Start
Step 1: for all keywords in dictionary do step 2 to 3.
Step 2: if the keyword in a dictionary is present in input file then count++.
Step 3: concatenate present index with count.
Step 4: return index file.
End
An index file is created for all input files and uploaded to respective bucket.

The uploaded files are downloaded by user based on the search keyword. For the given search keyword the system will calculate the buffer index file, by mapping buffer index file with uploaded index file it will find which are the files that contain this search keyword and those respective encrypted files are been downloaded and decrypted, this decrypted file is returned to user. The buffer index file is created as below procedure.

//Algorithm: buffer index file(search keyword, dictionary)
Start
Step 1: for each keywords in dictionary do step 2 to 4.
Step 2: if search keyword match with dictionary's keyword then do step 3 else step 4.
Step 3: concatenate buffer index with 1.
Step 4: concatenate buffer index with 0.
Step 5: return buffer index.
End

VI. RESULT ANALYSIS

i. Security: the files which are uploaded to cloud should be secure so, homomorphic encryption is adapted. The files are encrypted before they are uploaded to cloud and while downloading the file, it will be in the encrypted form, the client should decrypt it using private key.

ii. Efficiency: to increase the speed of file matching, every text file will have keywords and all keywords of every text file form dictionary, with this dictionary an index file is created to every file, the index file increase the speed of keyword search and find files matched containing that keyword.

The above table shows the result of the demo session where the time consumed in milliseconds for uploading the number of text files. So this concludes number of files and upload time are directly proportional.

No of files	Uploading time in ms
3 files	18081
4 files	45350
5 files	76846

The downloading time includes the search time, number of files matched and time taken to download one of the matched file and the size of the file. In the demo session for e.g. if a search keyword "private" is passed and search time taken is 108ms, it matches one file and time taken to download that file is 16660ms.

VII. CONCLUSION

In this paper, keyword based file retrieval is been proposed. By using this schema user can retrieve the matched files from cloud by specifying a keyword. Here the files are protected by using homomorphic encryption. This schema is efficient because index file is created for every file being uploaded to cloud, this index file helps in mapping the matched files more faster than query based file retrieval.

REFERENCES

[1] Rajkumar Buyya, Christian Vecchila,S. Thamarai Selvi "Mastering Cloud Computing Foundations and Applications Programming", MK publication.
[2] Maha Tebaa, Said El Hajji, Abdellatif El Ghazi "Homomorphic Encryption Applied to Cloud computing Security", Proceedings of the world Congress on Engineering vol 1 WCE 2012.
[3] Peter Mell, Timothy Grance, "The NIST Definition of Cloud Computing", Special Publication 800-145, September 2011.
[4] http://www.aws.amazon.com
[5] R. Ostrovsky and W. Skeith, "Private Searching on Streaming Data", J. Cryptol, vol. 20, no. 4, pp. 397-430, Oct. 2007.
[6] Shashi Mehrotra Seth, Rajan Mishra, Comparative Analysis Of Encryption Algorithms For Data Communication, IJCST Vol. 2, Issue 2, June 2011.
[7] Q. Liu, C.C. Tan, J. Wu, and G. Wang, ''Efficient Information Retrieval for Ranked Queries in Cost-Effective Cloud Environments,'' in Proc. IEEE INFOCOM, 2012, pp. 2581-2585.
[8] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.
[9] (2002) The IEEE website. [Online]. Available: http://www.ieee.org/
[10] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/
[11] *FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.
[12] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.
[13] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
[14] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.
[15] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.