Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 3, March 2016, pg.587 – 593

SECURE FILE KEY SHARING OVER SINGLE IMAGE ON CLOUD COMPUTING

M.Aruna[1], J. Nulyn Punitha[2]

[1]B.E Student, Department of Computer Science and Engineering, IFET College of Engineering, Villupuram, India
Email ID: arunaifet@gmail.com

[2]Senior Assistant Professor, Department of Computer Science and Engineering, IFET College of Engineering, Villupuram, India
Email ID: mailnulyn@gmail.com

Abstract- Cloud computing is a model of internet based computing where the resources like a storage space, online software are provided. IAAS - Infrastructure as a Service is one of the cloud service to users for outsourcing and retrieve data from anywhere in the world. When a cloud user outsources the data on cloud security is provided for preventing data manipulation or data access by unauthorized users. The TEES is time consuming and also there is a chance the cloud service provider might access those files which stored in cloud server because both the encrypted file and correspondent key and file indexes are stored in cloud server. Avoid these problems this system introduces storage nodes for storing file indexes and encrypted files and cloud server stores files keys. When a cloud user upload file then file index is generated automatically and file encrypted by using AES algorithm with generated key. After the key is converted into image as key image and source images by using visual cryptography scheme. The encrypted file and the file indexes are stored in storage node, key source image is stored in cloud server and key image is passed to file owner. Whenever file owner or file users want to download or access files then perform search and then put key image as a input. If it is valid then original key is displayed then again enter this key, after that file is decrypted then downloaded. Thus the proposed project not only reduces the computation time but also provides high security for files on cloud.

Keywords – IAAS, TEES, Visual Cryptography

I. INTRODUCTION

In recent year cloud storage system is widely used for uploading the file into the cloud server and retrieving them from cloud server at anywhere in the world. In this cloud storage all users are used to store their own files that time unauthorized user access the file without file owner permissions. That is chances to miss use the cloud file. So here we are using the encrypted the upload file and store to the cloud server. This is also a not secure for store the file into the cloud server system. So here we learn about in this paper how to store the file into cloud server with multiple keys this is mainly used for the unauthorized user cant access file without these two keys. And also we are encrypted the key and store to the cloud sever. When user download the file from the cloud server that time user must want asking the key from the file owner when that owner give the file permission for the users after that only access the file from the user. Here we using the key store to the particular cloud storage and files are stored in different node. When owner give the permission to the user that time file owner send the key into image time. That image background only view for the file user. When that file request send to the file owner that time only view the file key to file user. After that file user file download from the cloud server with decryption.

II. RELATED WORKS

A few scientists have done similar searches to avoid security related issues. Following are some of them, [1]A. A. Moffat, T. C. Bell's Managing gigabytes: compressing and indexing documents and images in which they used memory-based inversion algorithm highly trusted to use a frequent stored index is of course more complex searchable processing for Boolean queries. It is most appropriate for the systems that only be required to support ranked queries and extra processing cost for Boolean queries.

D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano[21]have done similar works using Polynomial time algorithm in the paper named Public key encryption with keyword search but the drawback is that The public key length grows linearly with the total dictionary size If we have an upper-bound on the public key length grows linearly with the total dictionary size If we have an upper-bound on the Total number of key word trap doors that the user will release to the email gateway (though we do not need to know these key words a-priori) we can do much Better using cover-free Families and can allow key word dictionary to be of exponential size.

Symmetric encryption: improved definitions and efficient constructions by []R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky addressed both of these issues by proposing new in distinguish ability and simulation-based definitions that provide security for both indexes and trapdoors, and show their equivalence by using Notation and Preliminaries.

Public key computation algorithm had been also used by S. Kamara and K. Lauter in the paper Cryptographic cloud storage," in Financial Cryptography and Data Security which concentrates on the user needs to maintain an independent secure channel with the PKG for the retrieval of his private key. To avoid the leakage of the relative relevance order but not the relevance score and realize an "as-strong-as-possible" ranked searchable symmetric encryption C. Wang, N. Cao, J. Li, K. Ren, and W. Lou proposed the paper named Secure ranked keyword search over encrypted cloud data.

FILE RETRIEVAL IN CLOUD STORAGE

Traditional Encrypted Search over Cloud Data

Traditional cloud storage system architecture and gen- eral procedures are shown in Figure 1, which include: file/index encryption by the data owner, outsourcing the data to the cloud storage, and encrypted data search/retrieval procedure of the data users in cloud computing File/Index Encryption The data owner first executes the preprocessing and indexing work as shown on Figure 2. He should invert files, that are selected to store on the cloud, for text search engines Every word in these files undergoes stemming to retain the word stem. After this step, the data owner encrypts and hashes every term (word stem) to fix its entry in the index. The index is then created by the data owner. Finally, the data owner encrypts the index and stores it into the cloud server, together with the encrypted file set. Most of the previous schemes under this architecture use Order Preserving Encryption (OPE)to encrypt the file index. This file index is often a TF (Term

Frequency) table composed of TF values. The TF-IDF table could be used to determine word relevance in documents.

Data Search and Retrieval after Authentication

A data user can only access a file after being authenticated by the data owner. In the process of authentication, the data user sends his identity to the data owner. The data owner sends the encrypted keys back if the user is a legal user. In the process of search and retrieval, the cloud server helps the users to find the top-k relevant files for a given keyword.

1) An authenticated user stems the keyword to be required, encrypt entry in the index. Then the encrypted keyword is sent to the cloud server.

2) On receiving the encrypted keyword, the cloud server first searches[21]for it in the index. Then the index related to this keyword is sent back to the data user.

3) The data user calculates the relevance scores with the selected index to find the top-k relevant files and sends a follow-up request to the cloud server in order to retrieve the files.

4) The position of these files is selected and they are sent back to the data user from the cloud server.

5) The data user decrypts the files and recovers the original data. The related computational components for these steps are illustrated in Figure 3, which indicate the traditional two-round-trip scheme for a file search and retrieval process invoked by an authenticated user. We call this file retrieval scheme abbreviated as TRS (Two Round trip Search). This scheme provides privacy protection through a complicated file retrieval process compared to a simple PlainText Search scheme (PTS) where searching and retrieving a file is done in only one round without security service.

Security Challenges: According to the efficiency challenges in cloud storage mentioned before, we should then address the security challenges introduced by of- floading part of the calculation onto the cloud. We con- sider the scenario where an authorized data user wants to search for files stored on the cloud server. This data user needs to retrieve the most relevant files through the encrypted data without downloading all the files. So the index should be stored in the cloud, leading to potential threats for MCS in the following cases:

1) Statistics Information Leak. Attackers could get the terms by analysing the TF table, since an Order Preserving Encryption (OPE) [5] method encrypted TF table produces a peaky histogram of TF values. In other words, term frequency should be evenly distributed to avoid statistic information leak, oth- erwise a broken index can be introduced with serious information leaks.

2) Keywords-files Association Leak. An attacker could determine query terms by observing queries and results through a wireless channel: as the result of the retrieval is keyword specific, attackers may guess the queried keyword by only observing the keyword and the result of the retrieval. Thus, it should avoid the relation of this keywords-files association in data encryption, which will be elab- orated in Section 4.

3) Server Information Acquisition. The cloud server maybe honest-but-curious, and may try to learn the underlying plaintext of users data. The cloud server can infer and analyze the encrypted index and get additional information, and we need to minimize the information acquisitionof the curious cloud server

III. DESIGN PRINCIPLE

Modified Process of Search and Retrieval

During the preprocessing and indexing stages, the data owner gets a TF tables index and uses [5]Order Preserving encryption (OPE) to encrypt it. As a result, the cloud server is able to calculate the relevance scores and rank them without decrypting the index. This renders the offloadingofthecomputationalloadsecureandpossible. Thus, the modified search and retrieval processes of TEES shown in Figure 2.

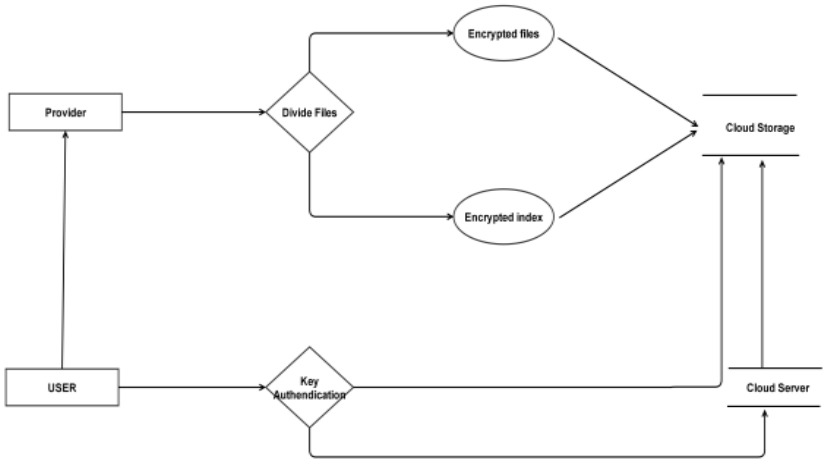

Fig 1: System Architecture

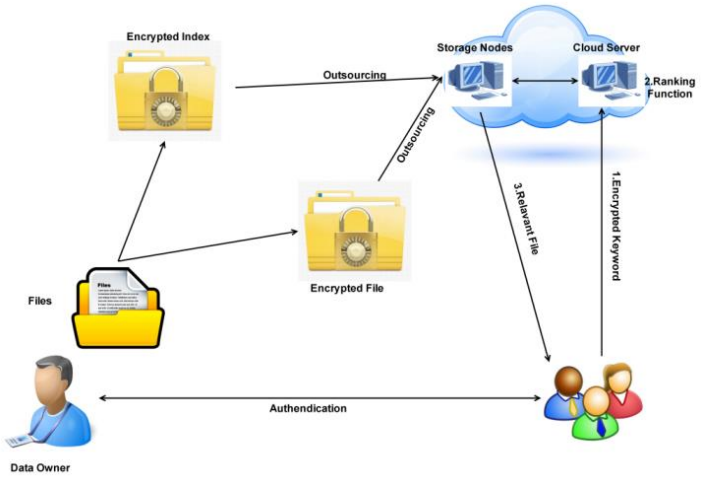

Fig 2: Process involved in TEES

Authentication:

In TEES, the data owner maintains a set of legal users ("legal set") and a set of users that will become invalid in after a defined delay ("overdue set"). When a user intends to access the file, he first sends his information to be authenticated by the data owner. In our design, we use our unified school authentication in TEES and transfer it through https for safety concern. The data owner sends the keys along with the hash table back if the user belongs to the legal set. This hash table will be used in the hash process .Then the data owner records the "International Mobile Equipment Identity"[14] of the user's mobile device and stores its encrypted version into the cloud. When the user's authority is overdue, his identity information is moved to the "overdue" set. The data owner will also notify the cloud of the changes. Note that the data owner should regularly update the hash table and the keys such that only users in the "legal set" will be notified. At the same time, this authentication process needs the data owner be online, but mature notification methods can be involved to push the authentication requests to the offline data owner.

IV. SYSTEM IMPLEMENTATION

The input of a system can be defined as the information that is provided to the system. This is used for future processing by the system to obtain meaningful information, which helps in decision-making. Input design is the process of converting user-oriented inputs to a computer-based format.

Input is a part of overall system design, which requires special attention. Inaccurate input data are the most common cause of errors in error processing. Input design can control errors entered by users. Entered data have to be checked for their accuracy and direction of errors. Appropriate error message have to be displayed. When an invalid data is entered, the user should not be allowed to type that data.

a)authentication implementation

The same procedure as followed in TEES for authentication is implemented primarily for authentication. When one user want to file from the cloud server that time first want two keys. So user send request to the file owner for access the first key. then when user receive the first key from the file owner, that time image key will be show to the user side. Finally user enter the both key into the cloud server that time file will be access from the cloud server.

b)building index and ranking

When file owner upload the file into the cloud server that time file are stored in one storage node keys are stored in other storage node. That time they are generate the two keys one is index key[10] that key only knows about file owner. When access the file that key only must will know about the index key. so every user can approve by the file owner permission. Cloud server calculates the relevance scores and return top-k relevant files according to the searching query from data user. The calculation scheme in is used in our scheme. Note that due to the order preserving index, any other relevance scores calculation method can also be employed.

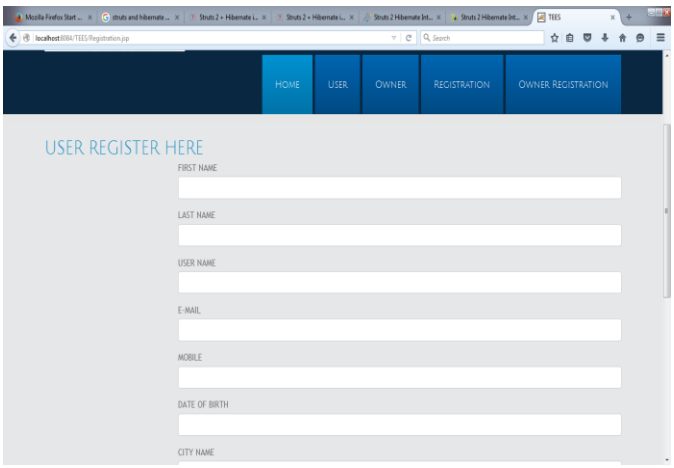

Fig 3: User registration for getting permission to access and downloading file

We assume that the cloud storage system provider will not collude with malicious users or intrude users' data intentionally. The cloud server can infer and analyze the encrypted index and get additional information, but it has no intention to modify any important data. We use the private cloud server from our school and assumed it as honest and perform important calculations here. This assumption is also used in most of the previous work.

Report generation module handled by the file owner. When file upload to the cloud server. Here report will display by the starting date and ending date wise. Because of how many files are we stored in cloud server easy to identify from the cloud server. And also duplicate file will avoid to store into the cloud sever.

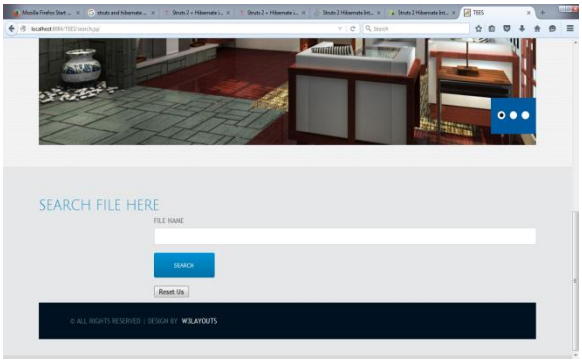

Fig 4:Process of searching the required file by giving the file key

V. CONCLUSION

In this existing system file owner stored the file into the cloud server. So here lot of file owners access permission in the same cloud server that time other file owner will access the other files. That owner will have the chances to miss use the other owner file. So here we are using the encryption technique and also two type of keys are generation. When you want access the file from the cloud server that time that two keys are must so that are security purpose of key generation. At the same time hacker will hack the key also. Hacker chances to hack the key also, so here in this proposed system we are also two key generation at the same time one key is hiding behind of the image. When file owner want file access from the cloud server that time must will enter the one key after that image key will be displayed this is mainly used for the hacker will not access the keys. When hacker hack the key that time image only will be displayed. This also reduces the computational time and enhances the security of the files that are uploaded into the cloud.

REFERENCES:

[1]A. Moffat, T. C. Bell et al., Managing gigabytes: compressing and indexing documents and images. Morgan Kaufmann Pub, 1999. [8] D. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Security and Privacy, 2000. S&P 2000. Proceedings. 2000 IEEE Symposium on. IEEE, 2000, pp. 44– 55.

[2]D. Huang, "Mobile cloud computing," IEEE COMSOC Multimedia Communications Technical Committee (MMTC) E-Letter, 2011.

[3] J. Oberheide, K. Veeraraghavan, E.Cooke, J. Flinn, and F. Jaha- nian, "Virtualized in-cloud security services for mobile devices," in Proceedings of the First Workshop on Virtualization in Mobile Computing. ACM, 2008, pp. 31–35.

[4] E. Han and G. Karypis, "Centroid-based document classification: Analysis and experimental results," Principles of Data Mining and Knowledge Discovery, pp. 116–123, 2000.

[5] A. Boldyreva, N. Chenette, Y. Lee, and A. O´ neill, "Order- preserving symmetric encryption," Advances in Cryptology- EUROCRYPT 2009, pp. 224–241, 2009. [35] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," Advances in Cryptology–EUROCRYPT 2010, pp. 24–43, 2010.

[6] C. Gentry and S. Halevi, "Implementing gentry´s fully- homomorphic encryption scheme," Advances in Cryptology– EUROCRYPT 2011, pp. 129–148, 2011.

[7] A. Swaminathan, Y. Mao, G. Su, H. Gou, A. Varna, S. He, M. Wu, and D. Oard, "Confidentiality-preserving rank-ordered search," in Proceedings of the 2007 ACM workshop on Storage security and survivability. ACM, 2007, pp. 7–12

[8] C. ¨Orencik and E. Savas¸, "Efficient and secure ranked multi- keyword search on encrypted cloud data," in Proceedings of the 2012 Joint EDBT/ICDT Workshops. ACM, 2012, pp. 186–195.

[9] K. Bowers, A. Juels, and A. Oprea, "Hail: a high-availability and integrity layer for cloud storage," in Proceedings of the 16th ACM conference on Computer and communications security. ACM, 2009, pp. 187–198.

[10] J. Zhang, B. Deng, and X. Li, "Additive order preserving encryption based encrypted documents ranking in secure cloud storage," Advances in Swarm Intelligence, pp. 58–65, 2012.

[11] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009. N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in INFOCOM, 2011 Proceedings IEEE. IEEE, 2011, pp. 829–837.

[12] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in Applied Cryptography and Network Security. Springer, 2004, pp. 31–45.

[13] A. Aizawa, "An information-theoretic perspective of tf-idf mea- sures," Information Processing and Management,vol. 39, pp. 45–65, 2003.

[14] G. Salton and M. J. McGill, "Introduction to modern information retrieval," McGraw-Hill, Inc.,New York, NY, 1986.

[15]N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," Parallel and Distributed Systems, IEEE Transactions on, vol. 25, no. 1, pp. 222–233, 2014.

[16] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," in INFOCOM, 2014 Proceedings IEEE.

[17] J. Zobel and A. Moffat, "Inverted files for text search engines," ACM Computing Surveys (CSUR), vol. 38, no. 2, p. 6, 2006.

[18] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in Proceedings of the 2004 ACM SIGMOD international conference on Management of data. ACM, 2004, pp. 563–574.

[19]D.M.Blei,A.Y.Ng,andM.I.Jordan,"Latentdirichletallocation," the Journal of machine Learning research, vol. 3, pp. 993–1022, 2003.

[20] L. Baker and A. McCallum, "Distributional clustering of words for text classification," in Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1998, pp. 96–103.

[21] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Advances in Cryptology- Eurocrypt 2004. Springer, 2004, pp. 506–522.

