International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 03 | Mar -2017 www.irjet.net p-ISSN: 2395-0072

A NOVEL ADDITIVE ORDER PROTOCOL IN CLOUD STORAGE AND AVOIDING THE TRAPDOORS

K.Vanitha[1], Mohammed Nasar Ahamed Siddiqui[2], K.Yasudha[3]

[1]Assistant Professor, GITAM institute of science, GITAM University, Visakhapatnam 530045, India
[2]P.G Student, GITAM institute of science, GITAM University, Visakhapatnam 530045, India
[3]Assistant Professor, GITAM institute of science, GITAM University, Visakhapatnam 530045, India

---

Abstract - *The popularity of the cloud had brought a convenience for outsourcing the owner's data. A cloud server is formed by connecting a various no of systems all together to a centralized remote server which is hosted on internet to store files but not on local machines. In present there was more demand for cloud computing in all domains and many users started outsourcing the files but in cloud there are some draw backs i.e. the owner's data will be in a plaintext format not in an encryption format. The search which is done will be off single keyword not of multi keyword. In this paper, we implement a new concept like double encryption for data before storing in to the cloud server and for the data on local host will be encrypted to avoid trapdoors and hacking. As an extension we proposed a new scheme like advanced authorization of cloud users, where the cloud user registration either owner or user need to get activation permission from cloud server. The user or owner who got activate permission will receive the login password for their registered mail id, with that only the owner or user can login and set their desired password otherwise they fail to login. And also we proposed a Novel Additive Order and a Privacy Preserving Ranked Multi-Keyword Search for Multiple Data Owner Model (PRMSMM), to give rank for the search files. So, finally this is the first time such functions, which gives high level of security for data during the insertion and retrieval compared to various primitive clouds.*

Key Words: Cloud server, Multi Keyword, Additive order, Rank search, Trapdoors.

1. INTRODUCTION

Cloud computing has more demand in existing world. In cloud computing total information is usually processed in remote machines [1], so the users cannot own it and operate it on their local machines. On cloud server there are two types of user's like data owners and users, the data owners will upload the file in cloud server and the users will download the file from cloud server. In cloud server the owners data is not having high level security due to lack of secured encryption format. At present, it is stored in plaintext format [2]. In this paper we proposed privacy preserving ranked multi keyword search protocol in a multi owner cloud model to enable cloud server to perform secure search. Also the multi keyword search over encrypted file would be encrypted and re-encrypted with different keys for different owners. In this scheme a new data owner allows to enter into the system without affecting other data owners and other data users. In this, the only authenticated data users can perform correct searches.

2. ABOUT MULTI KEYWORD SEARCH

In cloud servers, there was no facility like multi keyword search over an encrypted cloud data[9]. Now almost all cloud servers are using single keyword search with only one keyword like filename and there is no concept like searching the same file with multiple attributes[5]. If any user who wishes to download any file from the cloud server then he needs to get permission from the cloud owner, he needs to give the exact name of the file correctly in the search bar, then only he will be able to search that file[3]. If the user forgets that file name during his search then he can't download the exact file from the owner[6]. So in this paper we have implemented a keyword search like multi keyword search, in which the data owner while uploading the data in the current cloud enters a file name along with a keyword and then he browse the file from the desired location. In this way he is giving multi inputs while he is uploading the data[7]. If the user who wants to download the file from the cloud server then he can enter file name or keyword[4]. In this way multi keyword is implemented to retrieve the files from cloud server [8].

3. ABOUT RANKING

Ranking tells us the similarity among the set of items such that, for any two items, the first item is ranked higher than second item, or first item is ranked lower than second item or the first item is ranked equal to the second item. We can't say that all the objects should have different behavior, sometimes two objects may contain same rank during the comparison. In this paper we proposed the ranking method in order to differentiate each file compared with other files among the set. The file which has more rank will come to the top of list and the file which has very low rank will be displayed at bottom.

About Additive Order and Privacy Preserving Function

In additive order and Privacy Preserving Function we find the ranking for the documents which was uploaded into cloud server by data owner. Generally in theoretical, an **additive function** is defined as an arithmetic function termed as *f*(*x*) of the positive integer say '*x*' such that whenever *a1* and *a2* are prime, the additive function is defined as the summation of all the co prime values. This is represented as follows:

$$F(a1a2) = f(a1) + f(a2).$$

In this paper we use the sum of the relevance scores as the metric to rank search results. Here we introduced various encoded strategies for ranking the relevance scores. Initially, the cloud server computes

$$Vi,j = \Sigma_{t \in Wf} \; Vi,j,t$$

Now to find the ranking for the search values, the sum of all relevance scores between the j^{th} file and matched keywords for O_i, and the auxiliary value

$$Ti,j\,(y) = \Sigma\, t \in Wf\; Ti,j,t(y)$$

The relevance score between a keyword (W) and a document (F) represents the frequency or count in which that the keyword appears in the document. It can be used in searchable encryption for returning ranked results. A prevalent metric for evaluating the relevance score is **TF × IDF**, where TF (term frequency) represents the frequency of a given keyword in a document and IDF (inverse document frequency) represents the importance of keyword within the whole document collection. Without loss of generality, we select a widely used expression in [11] to evaluate the relevance score as

$$Score(\widetilde{W}, F_j) = \sum_{w \in \widetilde{W}} \frac{1}{|F_j|} \cdot (1 + \ln f_{j,w}) \cdot \ln(1 + \frac{N}{f_w})$$

Where

F j;w denotes the TF of keyword *w* in document *Fj* ;
Fw denotes the number of documents contain keyword *w*;
N denotes the number of documents in the collection; and
|Fj | denotes the length of *Fj* , obtained by counting the number of indexed keywords.

4. PROPOSED SYSTEM & ITS PRELIMINARIES

For many types of cloud applications, there will be several cloud service providers. But every cloud application should contain these four entities.

Data Owner Entity
Admin Entity
Cloud Server Entity
Data User/Search User Entity

The initial entity is data owner entity who wishes to outsource a collection of documents Z = (Z1, Z2, . . . , Zx) in encrypted form S = (S1, S2, . . . , Sx) to the cloud server and still preserve the search functionality on outsourced data. Here Z is assumed as document if there are many documents then it is represented as Z1, Z2 and so on. S is assumed as encrypted document which is stored in cloud server if there are many encrypted files then it is represented as S1, S2 and so on. Initially the file 'Z' will be encrypted by the data owner at his level before out sourcing into the Admin and also the file which is located on local machine will be encrypted to avoid hacking and trapdoors. The admin entity is one who login into his account and receives the data request which is send by data owner after an initial encryption. This admin will now receive all the files which were uploaded by various

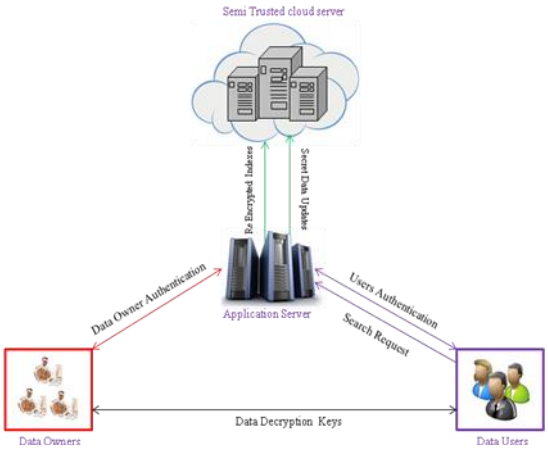

Fig -1: *Architecture Multi keyword search to avoiding the un authenticating in cloud storage*

data owners and then admin will re-encrypt with an inclusion of file name, file details, time and so on at his level and send to the cloud server. This double encryption or re-encrypt gives a high level of security for our proposed application compared with various primitive cloud service providers. Now the cloud server entity is an important entity among the four as this is the only entity which has privileges to store the encrypted documents into its storage area. When a search user entity try to download any file, search user will

send the input as either filename or keyword so that immediately the file request will be identified by the data base records and if the input keyword is matched the file will be downloaded otherwise search user cant download the file and he will be treated as an unauthenticated person. Cloud server entity which is a live cloud used to store the data in encrypted manner.

5. EVALUATION

Implementation is nothing but the conversion of theoretical design to programming manner. In this stage the proposed system will divide the application into different modules. The modules are

System Model
Data User Authentication
Illegal search detection
Search over multi owner

The proposed concept is implemented on java programming language with a front end HTML, Net beans and backend as a My SQL along with a real cloud called as DRIVEHQ.

5.1 System Model

We develop System Model to implement our proposed system. Our System model consists of Admin, Search users, data owners, and Cloud Servers. Data owners have a collection of files *F*. To enable efficient search operations on these files which will be encrypted, data owners first build a secure searchable index *I* on the keyword set *W* extracted from *F*, and then they submit *I* to the administration server. Finally, data owners encrypt their files *F* and outsource the corresponding encrypted files *C* to the cloud server. Upon receiving *I*, the administration server re-encrypts *I* for the authenticated data owners and outsources the re-encrypted index to the cloud server. Once a data user wants to search *t* keywords over these encrypted files stored on the cloud server, he first computes the corresponding trapdoors and submits them to the administration server. Once the data user is authenticated by the administration server, the administration server will further re-encrypt the trapdoors and submit them to the cloud server. Upon receiving the trapdoor *T*, the cloud server searches the encrypted index *I* of each data owner and returns the corresponding set of encrypted files. To improve the file retrieval accuracy and save communication cost, a data user would tell the cloud server a parameter *k* and cloud server would return the top-*k* relevant files to the data user. Once the data user receives the top-*k* encrypted files from the cloud server, he will decrypt these returned files [12][13].

5.2 Data User Authentication

To prevent attackers from pretending to be legal data users performing searches and launching statistical attacks based on the search result, data users must be authenticated before the administration server re-encrypts trapdoors for data users. Traditional authentication methods often follow three steps. First, data requester and data authenticator share a secret key, say, $k0$. Second, the requester encrypts his personally identifiable information $d0$ using $k0$ and sends the encrypted data $(d0)_{k0}$ to the authenticator. Third, the authenticator decrypts the received data with $k0$ and authenticates the decrypted data [12][13].

5.3 Illegal Search Detection

In our scheme, the authentication process is protected by the dynamic secret key and the historical information. We assume that an attacker has successfully eavesdropped the secret key $k_{0,\ j}$ of U_j. Then he has to construct the authentication data; if the attacker has not successfully eavesdropped the historical data, e.g., the request counter, the last request time, he cannot construct the correct authentication data. Therefore this illegal action will soon be detected by the administration server. Further, if the attacker has successfully eavesdropped all data of U_j, the attacker can correctly construct the authentication data and pretend himself to be *Uj* without being detected by the administration server [13].

5.4 Search Over Multi Owner

The proposed scheme should allow multi-keyword search over encrypted files which would be encrypted with different keys for different data owners. It also needs to allow the cloud server to rank the search results among different data owners and return the top-*k* results [13].

6. Conclusion

In this paper first time we implement a secure ranked search over multi keyword for multiple data owners and multiple search users in cloud computing environment. In present cloud services, there was no concept like encryption and ranked search before storing it into the cloud i.e., the data is just stored in plain text format. Since there was no encryption the data owner's data is not having a high level of security and due to lack of ranked search it is time taken process. In this paper for high level of security we proposed a double encryption concept which also avoids hacking the file from local machine. So trapdoors can't fetch the data

from local machines or remote machine. In this paper we also proposed a ranking search, so with in less time search user can get the file. For encryption and ranking search, an additive order and privacy preserving function is used. Finally this proposed mechanism gives a high level of security and getting a file within less time.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communication of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[2] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy preserving public auditing for secure cloud storage," *Computers, IEEE Transactions on*, vol. 62, no. 2, pp. 362–375, 2013.

[3] D.Song, D.Wagner, and A.Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE International Symposium on Security and Privacy (S&P'00)*, Nagoya, Japan, Jan. 2000, pp. 44–55.

[4] E. Goh. (2003) Secure indexes. [Online]. Available: http://eprint.iacr.org/

[5] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient constructions," in *Proc. ACM CCS'06*, VA, USA, Oct. 2006, pp. 79–88.

[6] D. B. et al., "Public key encryption with keyword search secure against keyword guessing attacks without random oracle," *EUROCRYPT*, vol. 43, pp. 506–522, 2004.

[7] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing," IEEE Trans. Services Computing, vol. 5, no. 2, pp. 220-232, Apr.-June 2012.

[8] [8] Y.-C. Chang and M. Mitzenmacher, "Privacy Preserving Keyword Searches on Remote Encrypted Data," Proc. Third Int'l Conf. Applied Cryptography and Network Security, 2005.

[9] [9] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in *Proc. Applied Cryptography and Network Security (ACNS'04)*, Yellow Mountain, China, Jun. 2004, pp. 31–45.

[10] Z. Xu, W. Kang, R. Li, K. Yow, and C. Xu, "Efficient multikeyword ranked query on encrypted data in the cloud," in *Proc IEEE Parallel and Distributed Systems (ICPADS'12)*, Singapore, Dec. 2012, pp. 244–251.

[11] Erdös, P., and M. Kac. On the Gaussian Law of Errors in the Theory of Additive Functions. Proc Natl Acad Sci USA. 1939 April; 25(4): 206–207.

[12] T. Jung, X. Y. Li, Z. Wan, and M. Wan, "Privacy preserving cloud data access with multi-authorities," in *Proc. IEEE INFOCOM' 13*, Turin, Italy, Apr. 2013, pp. 2625–2633.

[13] Wei Zhang, Yaping Lin, Sheng Xiao, Jie Wu and Siwang Zhou, "Privacy Preserving Ranked Multi-Keyword Search for Multiple Data Owners in Cloud Computing" January 2015.

BIOGRAPHY

Mohammed Nasar Ahamed Siddiqui pursuing Master of Computer Applications, GIS, GITAM University, Visakhapatnam. His area of interest in cloud computing.

K Vanitha is currently working as Assistant Professor in the Department of Computer Science, GIS, GITAM University. Her main areas of research includes Cloud Computing and Data Mining.

K Yasudha is currently working as Assistant Professor in the Department of Computer Science, GIS, GITAM University. Her main areas of research includes Cloud Computing.