International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 3, Issue 4, April 2016

Fuzzy Authorization on Cloud Computing By Using Merging Technique

[1] Rohit Thakur [2] Jitendra Korda [3]Datta Kadam [4]Pritesh Dhawre
[1][2][3][4] Department of Computer Engineering, JSPM's Rajarshi Shahu College Of Engineering Pune
[1]rohit.c.thakur@gmail.com,[2]jitendra96k@gmail.com, [3]askshreedatt@gmail.com,
[4]priteshdaware11@gmail.com

Abstract: **For storing the large amount of data from various users of different place the concept of cloud storage is used. But here we have issues like data missing, confidentiality and access control in cloud storage provider. In Cloud Storage, fuzzy authorization Scheme is used to overcome these issues. In our FA Scheme to enable an application which is registered to one cloud account can access the data files from his own cloud account. Our newly proposed System permits for the Fuzzy authorization for the improvement of scalability and flexibility of data file sharing by making the advantages of one to one correspondence between the Linear Secret Sharing System. In the Fuzzy authorization the security comes under the Diffe - Helman Statement. In the system we are proposing, the user can also register his profile detail in their own cloud, here now in this cloud having the application for the pdf merging, from different storage areas of field the files with OTP (one time password) authenticated to the third party account. Now after accessing these files from the third party merging is done of pdf file of own cloud to third party file into one file of pdf and then stored into own cloud space. In our proposed system, the authorization can realize fuzzy logic between the heterogeneous clouds with security and effectiveness.**

Keywords**: Access control, OTP authenticated, data missing, data storage, heterogeneous clouds, fuzzy authorization**

I. INTRODUCTION

Efficient search on encrypted data is also an important concern in clouds. The clouds should not know the query but should be able to return the records that satisfy the query. This is achieved by means of search able encryption. In this phase the users register and upload their file into cloud. For registration user has to fill all their personal details. Without user register into cloud they can't upload their files. Here we are using two types of cloud one is for cloud storage and other is for cloud storage, application provider. Once user registered their details into the cloud he/she can upload the file. Here PDF files are only uploaded into the cloud. In this phase user can view own profile and merging their uploaded PDF files. In that user account, cloud will show the user details and file which he/she uploaded. After viewing the account user will merge the files in the cloud. For merging file user select PDF files which the user has uploaded. Merged file can be displayed into the user page. Here merging and display PDF files are all done in single cloud. Here merging PDF files are done based on different cloud files. So far we are done merging only in a single cloud. But here user merge file from other cloud. For merging other cloud file, users enter into the cloud. Once particular user entered, cloud will generate OTP for that user. After he/she entered the OTP into the cloud, cloud verifies the OTP. If it is verified, cloud will list out the PDF files that he/she uploads. After user getting the list they can merge files. Owner of data encrypts the data with random symmetric key .

KE and it convert into CP-ABE form. Data owner can hide CP-ABE text into his application cloud. There is no need of storage in cloud storage.

II. DETAILS EXPERIMENTAL

2.1. Objective

Distributed access control of data stored in cloud so that only authorized users with valid attributes can access them. Authentication of users who store and modify their data on the cloud. The identity of the user is protected from the cloud during authentication. We propose a new secure authorization scheme for cloud storage providing file discrepancy tolerance, called fuzzy authorization. The security analysis shows that our FA scheme provides a thorough security of outsourced data, including confidentiality, integrity and secure access control.

There are four main entities in the system as shown in Fig. 2.2.1. Data owner: an entity who stores his/her data inside cloud storage and wishes to utilize cloud application services to process the data. A data owner must register with cloud storage provider and must be logged-in in order to upload access data or authorize.

2.2. System Architecture

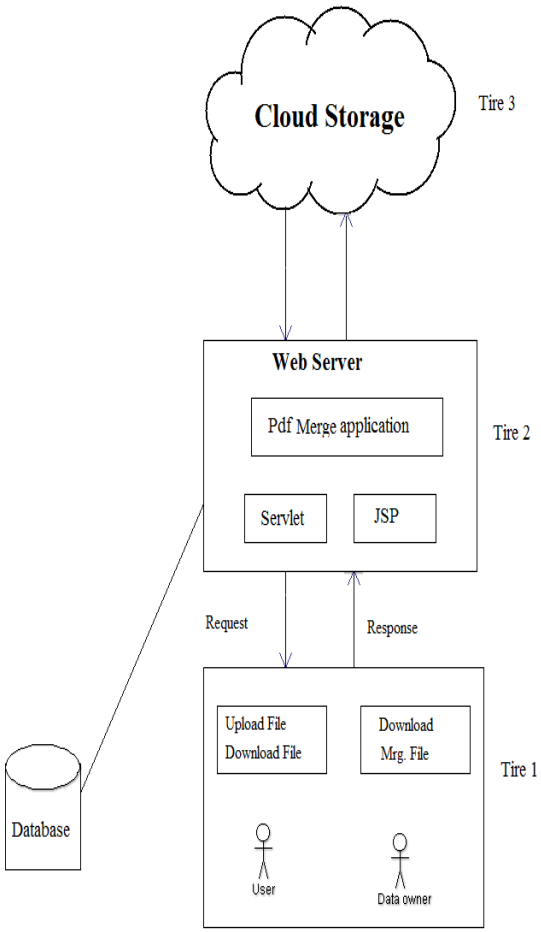

Fig. 2.2.1. System architecture

Application service provider (ASP): an entity to be authorized to access cloud storage data. It is an application software resides in vendor's system or cloud and can be accessed by users through a web browser or a special purpose client software. For example, PDFMerge is an online tool which can be used to merge several pdf files into one pdf file. With proper authorization, PDF Merge fetches the source pdf files from cloud storage. As a result, uploading files from data owner's local device is avoided. Cloud storage provider (CSP): an entity which supplies storage as a service to its clients and also provides access application programming interfaces (APIs) to ASP when ASP holds a valid access token. Dropbox and Just Cloud mentioned previously are examples of such entity. Application store (AS): an entity with which ASP must be registered to ensure itself integrity and authenticity. Google Chrome Web Store is a typical application store. Data owner encrypts his data with a random symmetric key KE and encrypts KE with our modified CP-ABE scheme. Owner encapsulates cipher text of KE and cipher textof data as an archive and stores the archive in the CSP. Format of the archive is similarly defined as AAuth archive. When owner needs to share data with ASP, he/she and CSP join together to issue ASP the indirect secret shares of file attributes while AS and owner collaborate to issue the indirect secret shares of application attributes. In this paper, an indirect share contains a genuine secret share as its exponent or a part of its exponent. For example, when ζ1 is a genuine secret share, g is a group element and r is random element, then gζ1r is an indirect secret share.

Since we emphasize the flexibility of multiple-file sharing, fuzziness is realized based on the file attributes. Once ASP gets all the indirect secret shares, it sends a request to CSP for formatted archive and then performs decryption of the archive header for KE. The main objective of this paper is to propose a secure and feasible way to address file sharing issue with high scalability and flexibility in cloud storage. The way owner accesses the archive is not discussed here. We assume that CSP, AS and ASP hold valid public-key certificates from Certificate Authorities and communications among the four parties are protected by SSL/TLS channels. We also assume that owner has both reading and writing permissions to cloud storage while ASP can be authorized with merely reading right.

III. RESULTS AND DISCUSSION

The Experimental results can be given as fuzziness of authorization among heterogeneous clouds can be achieved with the help of fuzzy authorization with security and efficiency.

a) Storage Consumption: In the access tree, each leaf node is associated with an attribute y. At the same time, two corresponding cipher components Cy = gPy(0) 2 , C_y = H(y)Py(0) must be added into the ciphertext. Assume the total number of leaf nodes of the access tree is n, and the number of F-subtree leaf nodes is n Let k be the number of archives that could be decrypted with the same KE and η be the distance that can be tolerated. In our simulation, n = 16 and k ranges from 1 to 10. Of all the attributes, FileName and FileLocation are most likely to be different and

cause the distance between two attribute sets. Hence two typical values of η, 1 and 2 are simulated. In Fuzzy IBE1, fuzziness of authorization canBe achieved by changing the threshold value of F-subtree. Polynomial Pf (x) and values of leaf nodes have to be recomputed. The extra cipher components for F-subtree leaf nodes must be updated and saved accordingly. At least 2 ∗ n 2 extra elements from group G are required. As to the Fuzzy IBE2, extra default nodes are added into F-subtree resulting in extra cipher components to be mounted in the ciphertext, i.e., 2η group elements from G. In addition, extra space for n 2 file attributes is needed. In FA, according to the property of MDS code, for distance adjustment ability of η, at least 2η checking nodes are required. The number of extra elements from group G is 4η. As the distance adjustment ability η and the number of authorized files k grow, the storage consumption of FA grows faster than Fuzzy IBE2. Even though Fuzzy IBE2 has to store at least n 2 file attributes, when k = 6, η = 2, FA has a higher storage consumption as Fig. 3 shows. For the same reason, FA exceeds Fuzzy IBE2 in storage consumption when k = 9, η = 1.

From Fig. 3, it suggests that extra storage consumption of FA is always less than Fuzzy IBE1. According to Fig. 3, when η = 1, k < 10 and η = 2, k < 6, FA has an advantage in storage consumption than Fuzzy IBE2.

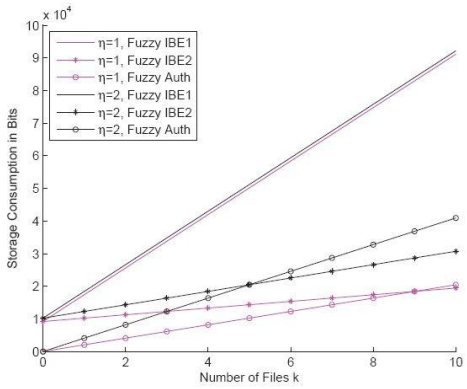

Fig. 3: Storage consumption of Fuzzy IBE and fuzzy authorization.

b) Revocation Efficiency: Currently, most authorization scheme utilizes manual revocation. As theBackground of owner varies greatly and for a less-cared owner, he/she may easily forget revocation. We assume that once owner remembers, he/she will revoke. Therefore, based on Ebbinghaus Forgetting Curve, the probability of revocation failure is demonstrated in Fig. 4. Assume owner updates the original data at time tchange, then in FA scheme, the non-revocation probability before tchange is 100% and after tchange is 0%. As manifested in Fig. 4, the uncertainty of human brain may result in higher failure while revocation in FA is more determinate. And in a long run, revocation in FA is advantageous.

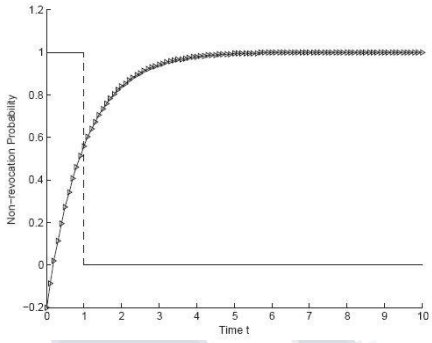

Fig. 2: Non-revocation probability of manually and fuzzy authorization.

IV. CONCLUSIONS

In this paper, we propose a new Fuzzy Authorization for cloud which handles out a smooth file-sharing between file owner who stores his data in one cloud party and other applications which are registered within another cloud party. The replication of FA technique proves that our scheme can successfully quickly correct the unmatched indirect secret shares, adjust the attribute distance, resoundingly recover the top secret and then efficiently perform the decryption for KE. Furthermore, the replication indicates that with the update of Time Slot attribute, FA protocol automatically invalidates the authorized reading right from ASP.A more accurate authentication is needed between data owner, ASP and AS, which makes the problem more challenging.

REFERENCES

1. Shasha Zhu and Guang Gong, Fellow, IEEE, "Fuzzy Authorization for Cloud Storage", Citation information: DOI.2014, IEEE Transactions on Cloud Computing.

2. SushmitaRuj‡, Milos Stojmenovic†, Amiya Nayak∗‡CSE, Indian Institute of Technology, "Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds", IEEE

TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS VOL:25 NO:2 YEAR 2014.

3. Xinyi Huang, Joseph K. Liu+, Shaohua Tang, Yang Xiang, Kaitai Liang, Li Xu, Jianying Zhou," Cost Effective Authentic and Anonymous Data Sharing with Forward Security", IEEE TRANSACTIONS ON COMPUTERS VOL: 64 NO: 6 YEAR 2015.

4. Gladwin A,AmarnathS,ArunMozhiS.,"Fuzzy Security in Cloud Storage", International Journal of Computer Science and Information Technology Research ISSN 2348-120X,Month: January - March 2015.

5. Fengyuan Xu, Member, IEEE, Xiaojun Zhu, Chiu C. Tan, Member, IEEE," SmartAssoc: Decentralized Access Point, Selection Algorithm to Improve Throughput", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS,NO. 12, DECEMBER 2013.