IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 3, March 2016

Enabling Public Auditability and Data Dynamics for Storage Security and Data Integrity in Cloud Computing

Adith .A. Varghese[1], M.M. Ashwin[2], R. Harsshath Prabu[3], T. Praveen[4], Mr. B. Karthikeyan[5]

Student, Department of Information Technology, Panimalar Engineering College, Chennai, India[1, 2, 3, 4]

Assistant Professor (Grade 1), Department of Information Technology, Panimalar Engineering College, Chennai, India[5]

Abstract: In the Present Scenario, all the software processes use web based application which enables multiple clients to request access to the same file simultaneously. This might lead to server overload or even trigger a deadlock occurrence. Keeping this in mind, the concept of dynamic proof through captcha or OTP has been introduced here with other features including the likes of Key Generation and Cryptography. There are three major actors namely Admin, Audit and Client each with respective roles. The Client performs registration; log-in activities and also the file upload operations. The Admin monitors and controls the user registration approval whereas the Audit analyses and approves the content of the file being uploaded. The data owners can provide file access to clients requesting it by means of transferring a key through mail. The clients can perform editing and updating processes in the respective file with the help of the key and await verification cum uploading of the file by the data owner. The maintenance and storage of these files in a cloud computing server and the usage of key generation and captcha helps reduce server overload and improve data security.

Keywords: File-Upload, Cloud Storage, Audit Verification, File Request, File Access Block, Recovery Stage.

I. INTRODUCTION

This project introduces the concept of dynamic proof during multiple requests by the clients to the server which might cause it to overload. Through the introduction of dynamic proof techniques like captcha or OTP the above said can be prevented. Whenever a client performs an operation in cloud like upload, file request or download a dynamic proof will be generated for this process. Also the technique of content matching is implemented whenever the client modifies the data. The data owner will match the content of original file and modified file and then will either upload the file or delete the changes performed. A number of solutions have been proposed to solve the verification of cloud data integrity and retrievability in cloud storage systems. Through the use of efficient dynamics proof and content matching, the main problem of overload occurrence on server and file recovery when client modifies the data may be overcome.

II. OBJECTIVE

The main aim of the project is to minimize the server overload and dead lock. So the multiple users can access the file in pipe lining systems. This system also provides the full security and cryptography systems. So this project can perform the reliable and secure file sharing and file access in cloud.

A. Problem Definition

A number of solutions have been proposed to solve the verification of cloud data integrity and retrievability in cloud storage systems. Most of the existing system, doesn't support efficient dynamics proof and content matching. The main problem occurs whenever overload occurs on the server and in file recovery when user modifies the data. The problem when user changes the content of file is that it's difficult for data owner to recover the original file.

B. Existing System

The Existing system triggers server overload in online file sharing in cloud and has very less security features. It is also easily hacked or blocked, permits miss-use of the data owner files and sports less reliability and poor performance.

I) Disadvantage

- Does not support efficient data dynamics.
- Suffers from security vulnerabilities when involving dynamic data operations.
- Less reliability and low performance.

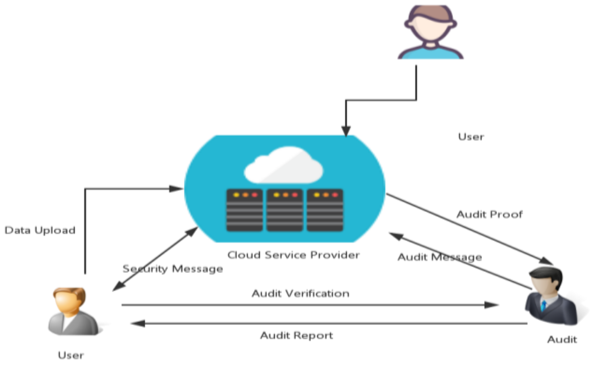

Fig 1: Existing System Architecture

DOI 10.17148/IJARCCE.2016.53233

C. Proposed System

The support of data dynamics allows the data owner to modify, delete the existing data blocks and insert new blocks. This project can generate dynamic proof for reducing more requests to the server by keeping captcha or OTP code. An additional layer of security is introduced where the user gets blocked on the event of three successive modifications to the data.

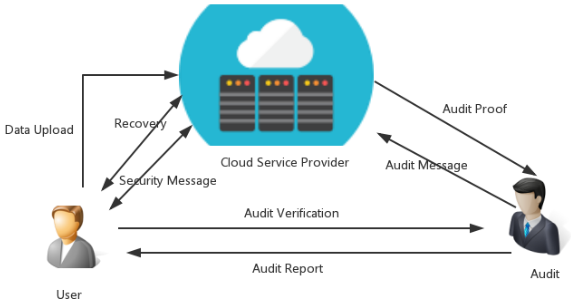

Fig 2: Proposed System Architecture

I) Advantage

- □ The verification of remote data integrity and retrieval in cloud.
- □ Proposed system is defending against pollution attacks during data recovery.
- □ Higher level of data integrity with the usage of editor tab for content matching.
- □ User blocking on unwanted modifications to the data; like more than or equal to 3 times simultaneously.

III. FEASIBILTY STUDY

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations involved in the feasibility analysis are

- Economical Feasibility
- Technical Feasibility
- Social Feasibility

I. Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

II. Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

III. Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

IV.MODULES

I) File Upload

In this module, the Data owner uploads the file to the server. This file is then encrypted, split and stored with the use of two keys namely a private key and a public key. This encrypted and fragmented file is stored in a cloud server. The public key is provided to a third party for verification purposes and the private key is maintained by the data owner for sharing purposes.

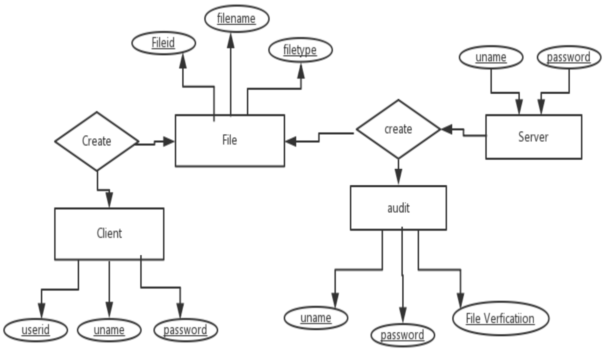

Fig 3: ER – Diagram for Proposed System

II) Cloud Storage

The encoding is split into two further codes namely outer code and the inner code. The outer code is the encrypted file itself whereas the inner code is the actual uploaded file. The actual file is also partitioned into three blocks. With the help of the inner code, the complete file can be recovered in case of any file corruption.

III) Audit Verification

The Audit or the auditor performs verification activity on the file being uploaded by the data owner; on the inner code i.e. the encrypted file and also the total number of blocks. This verification is performed on the basis of File id and Filename.

IV) File Sharing

In this module, the client can request for file access, the cloud will intimate the date owner on this request and the data owner may choose to share the key for file access and

DOI 10.17148/IJARCCE.2016.53233

sharing. With this key, the client may access the file and perform edition process. The encrypted file that's stored in the cloud can be decrypted only with this key provided by the data owner.

V) File Access Block
In this module, if a client tries to perform file modification, an alert will be sent to the data owner. If the same client tries to modify the file more than three times, the specified client may be blocked through a request sent by the data owner to the admin. A blocked client can access a file only after the admin approves the action.

VI) Recovery Stage
In this module, if a client has corrupted a file, the data owner may delete this corrupted file and upload the original file. To facilitate this replication approach, the original data is completely copied to each of the cloud servers thereby is one server is corrupted, the data may still be recovered from the other servers by simply copying it from the same.

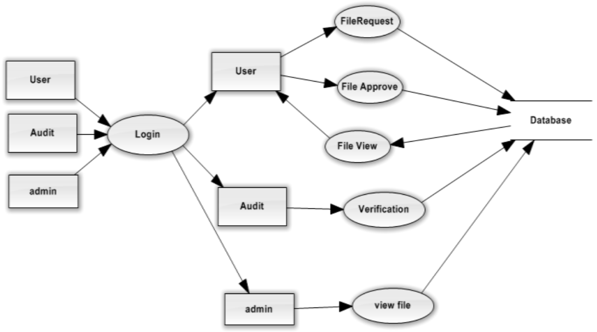

Fig 4: Data Flow Diagram for Proposed System

V. CONCLUSION

At last this project concludes a new dynamic proof of retrievability scheme for coded cloud storage systems. To reduce the overload on server by using dynamic proof and this project is implementing captcha for reducing overload. This project is implementing recovery process for data by content matching.

REFERENCES

[1.] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing," Proc. 14th European Symp. Research in Computer Security (ESORICS'09), pp. 355-370, 2009.

[2.] A. Juels and B.S. Kaliski, "PORs: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS'07), pp. 584-597, 2007.

[3.] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," Proc. 29th Annual IEEE Int'l Conf. Computer Comm. (INFOCOM'10), pp. 525-533, 2010.

[4.] Y. Zhu, H. Wang, Z. Hu, G.J. Ahn, H, Hu, and S.S. Yau, "Cooperative Provable Data Possession," Report 2010/234, Cryptology ePrint Archive, 2010.

[5.] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, and D. Song, "Remote Data Checking Using Provable Data Possession," ACM Transactions on Information and System Security, vol.14, no.1, pp.12-34, 2011.

[6.] H. Wang, "Proxy Provable Data Possession in Public Clouds," IEEE Transactions on Services Computing, vol.6, no.4, pp.55559, 2013.

[7.] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'08), pp.90- 107, 2008.

[8.] K.D. Bowers, A. Jules, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Proc. 16th ACM Conf. Computer and Comm. Security (CCS'09), pp.187-198, 2009.

[9.] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote Data Checking for Network Coding-Based Distributed Storage Systems," Proc. 2010 ACM Workshop on Cloud Computing Security Workshop (CCSW'10), pp.31-42, 2010.

[10.] Y. Zhu, H. Wang, Z. Hu, G.J. Ahn, and H. Hu, "Zero-knowledge Proofs of Retrievability,", Science China: Information Sciences, vol.54, no.8, pp.1608-1617, 2011.