SECURING HYBRID DATA ON CLOUD COMPUTING

[1]Mrs. Shraddha.R.Khonde, [2]Tejaswini Mane, [3]Pranali Randhir , [4]Swati Jaybhaye
[5]Kirti Nangude
Dept of Computer Engg MESCOE, Pune.
Email: [1]shraddha.khonde@mescoepune.org, [2]tejaswinivmane@gmail.com,
[3]pranalirandhir@gmail.com. [4]Swatijaybhaye0911@gmail.com,
[5]kitkatnangude.20@gmail.com

Abstract- Cloud computing is a recently evolved computing metaphor based on utility and consumption of computing resources. Data Outsourcing to cloud storage server is raising trend among many organization & user owing to its economic advantages. This essentially means that owner (client) of the data moves its data to a third party storage server which is supposed to presumably for a free-faithfully store the data with it and provide it back to the owner whenever required. We provide a scheme which gives a proof of data integrity in the cloud which the customer can employ to check the correctness of his data in cloud. This proof can be agreed by both the cloud and customer and be incorporated by Service Level Agreement (SLA).

Keywords: cloud computing, data integrity, third party auditor, etc.

I. INTRODUCTION

The Cloud is just a simple term for a network or remote servers which can be accessed via an Internet connection store and manage information. It consists of hardware and software resources made available on the Internet as managed third-party services.

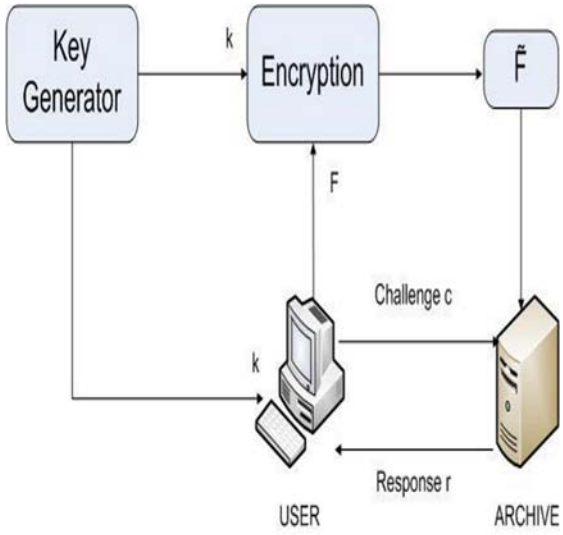
In cloud computing, a large number of clients or data owners stored their data on servers and it is provided back to them whenever needed. After data is fetched on the cloud, you may not have control over data where it is stored as well as how it is used. Numerous issues are associated with this situation. It is difficult for the user to store the entire data within the system; therefore clouds are formed to store the user data. These services typically provide access to advanced software applications and high-end networks of server computers. In cloud computing, data is moved to a remotely located cloud server. Cloud faithfully stores the data and return back to the owner whenever needed. But there is no guarantee that data stored in the cloud is secured and not altered by the cloud or Third Party Auditor (TPA).

As data generation is far outperforming data storage it proves costly for small organizations to frequently update their hardware whenever additional data is created. Also maintaining the storages can be a difficult task. Storage outsourcing of data to cloud storage helps such organizations by reducing the costs of storage, maintenance and workforces. It can also assure a reliable storage of important data by keeping multiple copies of the data thereby reducing the chance of losing data by hardware failures.

Storing of user data in the cloud in spite of its advantages has many interesting security concerns which need to be extensively investigated for making it a reliable solution to the problem of avoiding local storage of data. In this paper we deal with the problem of implementing a protocol for obtaining a proof of data possession in the cloud sometimes referred to as Proof of retrievability (POR).This problem tries to obtain and verify a proof that the data that is stored by a user at a remote data storage in the cloud. Not modified by the archive and thereby the integrity of the data is assured.

The Client store his data on a cloud which is to be clearly understood resolved and need to be addressed is to assure the customer about the integrity i.e. Correctness of his data on the cloud. Such verification systems prevent the cloud storage archives from twisting or modifying the data stored at it without the permission of the data owner by using frequent checks on the storage archives. Such checks must allow the data owner to efficiently, frequently, quickly and securely verify that the cloud archive is not cheating the owner. Cheating in this context, means that the storage archive might delete some of the data or may modify some of the data.

II. SYSTEM ARCHITECTURE



- **Key generator:**

A random key is generated

- **Encryption:**

A metadata of the file is generated and is encrypted using any suitable algorithm

- **Archive:**

It is the data centre where the data is outsourced

- **Challenge:**

Verification of data integrity by the user

- **Response:**

Proof of data integrity by the data centre.

III. PROJECT SCOPE

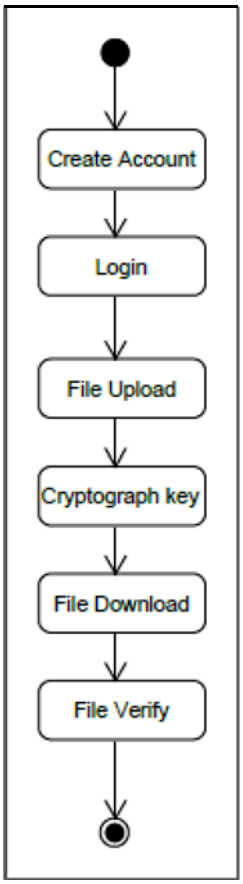
Cloud storing its data file F at the client should process it and create suitable meta data which is used in the later stage of verification the data integrity at the cloud storage. When checking for data integrity the client queries the cloud storage for suitable replies based on which it concludes the integrity of its data stored in the client. our data integrity protocol the verifier needs to store only a single cryptographic key - irrespective of the size of the data file F- and two functions which generate a random sequence. The verifier does not store any data with it. The verifier before storing the file at the archive, preprocesses the file and appends some meta data to the file and stores at the archive.

IV. PROJECT PURPOSE

Purpose of developing proofs for data possession at untrusted cloud storage servers we are often limited by the resources at the cloud server as well as at the client. Given that the data sizes are large and are stored at remote servers, accessing the entire file can be expensive in I/O costs to the storage server. Also transmitting the file across the network to the client can consume heavy bandwidths. Since growth in storage capacity has far outpaced the growth in data access as well as network bandwidth, accessing and transmitting the entire archive even occasionally greatly limits the scalability of the network resources. Furthermore, the I/O to establish the data proof interferes with the on

demand bandwidth of the server used for normal storage and retrieving purpose.

V. FLOWCHART



1. **Create Account**
   User will create an account on server.
2. **Login**
   After creating an account he will gets its respective username and password by which he can login.
3. **File Upload**
   User will upload his file on server using its login id.
4. **Generating Key**
   After uploading the file on server two keys are generated i.e. public key and private.
5. **Download**
   The selected file will be downloaded.
6. **Verify**
   Third Party Auditor will check for the correctness of data i.e. whether the integrity is maintained.

VI. CONCLUSION

In this paper we have worked to facilitate the client in getting a proof of integrity of the data which he wishes to store in the cloud storage servers with bare minimum costs and efforts. Our scheme was developed to reduce the computational and storage overhead of the client as well as to minimize the computational overhead of the cloud storage server. We also minimized the size of the proof of data integrity so as to reduce the network bandwidth consumption. Many of the schemes proposed earlier require the archive to perform tasks that need a lot of computational power to generate the proof of data integrity. But in our scheme the archive just need to fetch and send few bits of data to the client.

VII. ACKNOWLEDGEMENT

We thereby thank our college **MES College of Engineering (MESCOE),** Pune for the motivation. Also we would like to thank our guide **Mrs. Shraddha.R.Khonde** for her active support.

REFERENCES

1. E. Mykletun, M. Narasimha, and G. Tsudik, “Authentication and integrity in outsourced databases,” Trans. Storage, vol. 2, no. 2, pp. 107–138, 2006.
2. Data Communications and Networking, by *Behrouz A Forouzan*.
3. A.Adya, W.J.Bolosky, M.Castro, G.Cermak, R.Chaiken, J.R.Douceur, J.Howell, J.R.Lorch, M.Theimer, and R.Wattenhofer,” Farsite: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment” Proc. Fifth Symp. Operating System Design and Implementation (OSDI), pp. 1-14, 2002.

4. H.-Y. Lin and W.-G. Tzeng, "A Secure Decentralized Erasure Code for Distributed Network Storage", IEEE Trans. Parallel and Distributed Systems, vol. 21, no. 11, PP. 1586-1594, Nov. 2010.

5. M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus: Scalable Secure File Sharing on Untrusted Storage", Proc. Second USENIX Conf. File and Storage Technologies (FAST), PP. 29-42, 2003.