ISSN 2395-1621

Load Balancing With Third Party Auditor

[#1]Priya Tayde, [#2]Priyanka Shelke, [#3]Poonam Pawar, [#4]Pooja Take, [#5]Ashwini Sutar

[1]priyatayde22@gmail.com
[2]priyankashelke0015@gmail.com
[3]pypawar.sae@sinhgad.edu
[4]poojatake09@gmail.com
[5]ashwinisutar14@gmail.com

[#12345]Department of Information Technology
Sinhgad Academy of Engineering,Pune.

ABSTRACT

Cloud storage is provided to user through internet to secure their data on cloud. Cloud facility is freely provided to users to store their data to secure it. Organizations use private cloud for storing and securing data which is very costly because they have to buy space for data storage. The basic concept of this paper is to balance the data on cloud to avoid overload on single node and also to secure data using Third Party Auditor (TPA) which is cheaper than private cloud. Proposed system can be used for load balancing purpose and it can be done by distributing the load on one cloud upon other nodes and also to check data tampering occurred or not using TPA.

Keywords: **Private cloud storage, Load balancing, Third Party Auditor (TPA).**

ARTICLE INFO

Article History

Received: 12th May 2016

Received in revised form :

13th May 2016

Accepted: 18th May 2016

Published online :

19th May 2016

I. INTRODUCTION

It is very critical to protect the outsourced data in cloud storage against corruption, adding fault tolerance to cloud storage together with data integrity checking and failure reparation. Recently, regenerating codes have gained popularity as they have lower repair bandwidth while providing fault tolerance. Existing remote checking methods are only provide the private auditing, requiring data owners to all the times stay online and handle auditing, repairing which is sometimes impractical.

In our proposed system, we are using Third party Auditing (TPA) system for checking duplication and regenerating-code-based cloud storage.

To solve the regenerating problem of failed authentication when the data owners are offline, we are using a web service which regenerates the authentication. Moreover we design Third Party Authenticator (generated by a couple of keys and can be regenerated using partial Key), because of that we are completely release data owners from online burden.

Load balancing system is used to distribute the workload across two or more computers ,networks, CPU's or other resources to achieve the optimal resource utilization, maximum throughput and minimum response time and avoid the overload of any of the resources .To increase the Reliability through redundancy maximum components are used.

II. LITERATURE SURVEY

[1] M. Armbrust et al., "Above the clouds: A Berkeley view of cloud computing," Dept. Elect. Eng. Computer.Sci. Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-28, 2009.

Cloud Computing, the long-held dream of computing as a utility, has the potential to transform a large part of the IT industry, making software even more attractive as a service and shaping the way IT hardware is designed and purchased. Developers with innovative ideas for new Internet services no longer require the large capital outlays in hardware to deploy their service or the human expense to operate it. Moreover, companies with large batch-oriented tasks can get results as quickly as their programs can scale, since using 1000 servers for one hour costs no more than using one server for 1000 hours. This elasticity of resources, without paying a premium for large scale, is unprecedented in the history of IT. Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. The services themselves have long been referred to as Software as a Service (SaaS).

[2] G. Ateniese et al., "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun.Secur.(CCS), New York, NY, USA, 2007, pp. 598–609.

He introduce a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage systems. We present two provably-secure PDP schemes that are more efficient than previous solutions, even when compared with schemes that achieve weaker guarantees. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data. Experiments using our implementation verify the practicality of PDP and reveal that the performance of PDP is bounded by disk I/O and not by cryptographic computation.

[3] A. Juels and B. S. Kaliski, Jr., "PORs: Proofs of retrievability for large files," in Proc. 14th ACM Conf. Comput. Commun.Secur. 2007, pp. 584–597.

In this paper, we define and explore proofs of retrievability (PORs). A POR scheme enables an archive or back-up service (prover) to produce a concise proof that a user (verifier) can retrieve a target file F, that is, that the archive retains and reliably transmits file data sufficient for the user to recover F in its entirety. A POR may be viewed as a kind of cryptographic proof of knowledge (POK), but one specially designed to handle a large file (or bit-string) F. We explore POR protocols here in which the communication costs, number of memory accesses for the prover, and storage requirements of the user (verifier) are small parameters essentially independent of the length of F. In addition to proposing new, practical POR constructions, we explore implementation considerations and optimizations that bear on previously explored, related schemes. PORs give rise to a new and unusual security definition whose formulation is another contribution of our work. The goal of a POR is to accomplish these checks without users having to download the files themselves. A POR can also provide quality-of-service guarantees, i.e., show that a file is retrievable within a certain time bound.

III. PROPOSED SYSTEM

1. DISADVANTAGES OF EXISTING SYSTEMS

1. User need to be online always.
2. It increases the number of work force.
3. There is no guarantee that the data stored has no change.

2. INTRODUCTION TO PROPOSED SYSTEM

In proposed system we are going to divide the file which we have uploaded which helps in load balancing and then TPA will check uploaded files for tampering. File is downloaded after merging of three parts and mail is send to user whose file got tampered.

ALGORITHMS

1. AES ALGORITHM

AES algorithm is symmetric key cryptographic algorithm which uses same key for encryption and decryption of data contained in files.

2. SHA-1 ALGORITHM

SHA-1 is used for generating unique hash key for the data files. It is efficient than MD5 algorithm for hash key generation.

3. SPLIT AND MERGE ALGORITHM

In split algorithm, the file which we are going to upload is splitted into three parts and when we wand the uploaded file, that is, at the time of download the splitted parts gets combined using merge algorithm and then provided to user.

3. ADVANTAGES OF PROPOSED SYSTEM

1. The encryption of data, for safe keeping.
2. The uploaded file will be checked for tamper by third party auditor.
3. Implementing de- duplication to give efficient bandwidth utilization.
4. If any damage to uploaded file, then file will be repaired.

IV. PROPOSED SYSTEM ARCHITECTURE

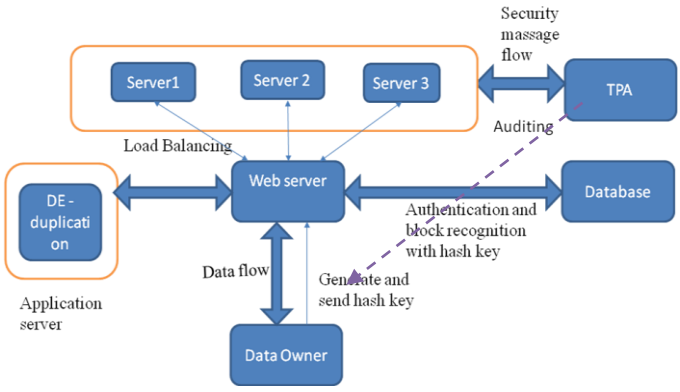


Fig 1. System architecture

Architecture diagram explains how the load is balanced within three nodes and how TPA detects the tampering and restores the tampered file.

V. PROPOSED SYSTEM ALGORITHMS

AES algorithm is an encryption algorithm for securing sensitive information. It is a symmetric algorithm, that is same key is used for encryption and decryption. Initially, in proposed system the user has to register him and then when he sends data to cloud the data will go in encrypted form which will secure the data from hackers as it is not easily readable. So we used AES algorithm for encryption:

- AES algorithm consists of 6 steps:

1. Derive the set of round keys from the cipher key.

2. Initialize the state array with the block data (plaintext).

3. Add the initial round key to the starting state array.

4. Perform nine rounds of state manipulation.

5. Perform the tenth and final round of state manipulation.

6. Copy the final state array out as the encrypted data (cipher text).

For generating hash key SHA1 algorithm is used.SHA1 algorithm is used to generate hash key for data packets. TPA uses this hash key for checking the tampering of data by matching the original hash key and the hash key generated by TPA after downloading the uploaded file.

- SHA1 algorithm consists of 6 steps:

1. Appending Padding Bits. The original message is "padded" (extended) so that its length (in bits) is congruent to 448, modulo 512.

2. Appending Length. 64 bits are appended to the end of the padded message to indicate the length of the original message in bytes.

3. Preparing Processing Function.

4. Preparing processing constants. It consists of 80 processing constants.

5. Initializing Buffers. It has 5 buffers with values:
H0=0x67452301
H1=0xEFCDAB89
H2=0x98BADCFE
H3=0x10325476
H4=0xC3D2E1F0

6. Processing Message in 512-bit Blocks. This is the task of SHA1 Algorithm, which loops through the padded and appended message in blocks of 512 bit each.

VI. SYSTEM MODEL

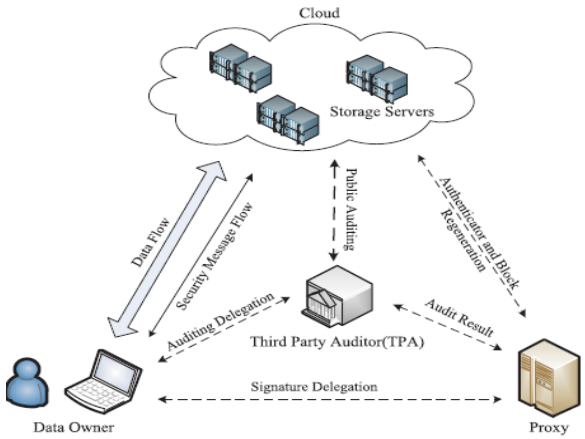


Fig. 2. The system model.

Fig 2. System model

The system model consist of following four entities:

Data owner who uploads the large amount of data files to the cloud,as they will be stored on the cloud. These files are managed by cloud service provider.cloud service provider, provides the storage service and have significant computational resources, then after that the Third Party Auditor(TPA) is a trusted and its audit result is used by both data owners and cloud servers and proxy agents proxy agent is a semi-trusted agent and used to regenerate authenticators and data blocks on the failed servers during the repair procedures.

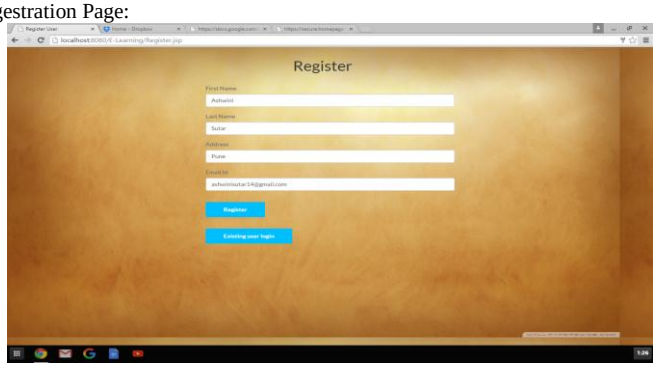
EXPERIMENT WORK:

In our experiment we have uploaded the document and then splitted it, stored on cloud. TPA download the document and check for tampering of data in it. While downloading Merging of splitted document is done and then provided to user as per their requirement.

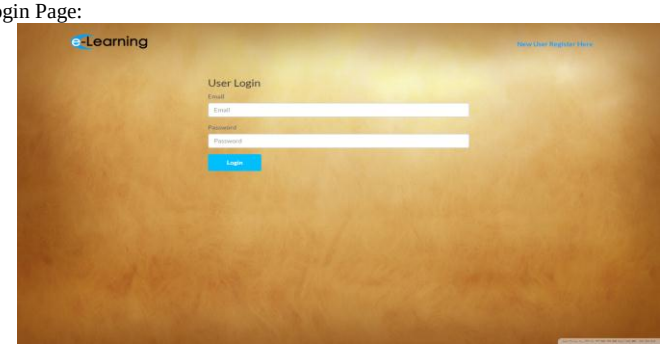
If tampering occurred, that is, if data in document changed then TPA tells that tampering occurred and replaces the file automatically.

VII. IMPLEMENTATION RESULT

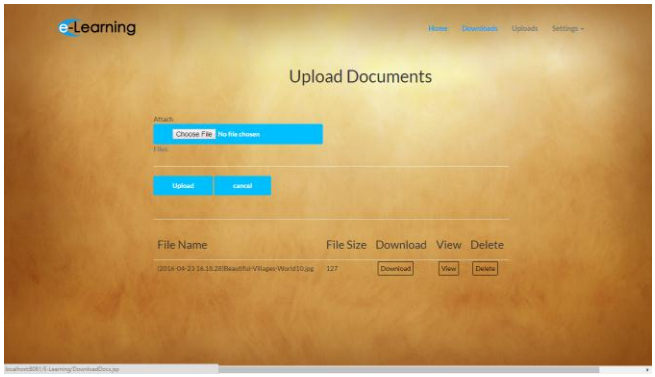
Regestration Page:



Login Page:

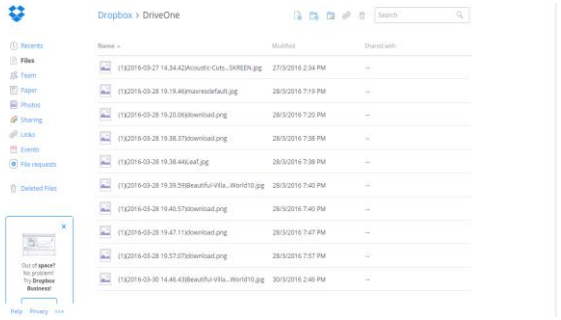


Upload document:

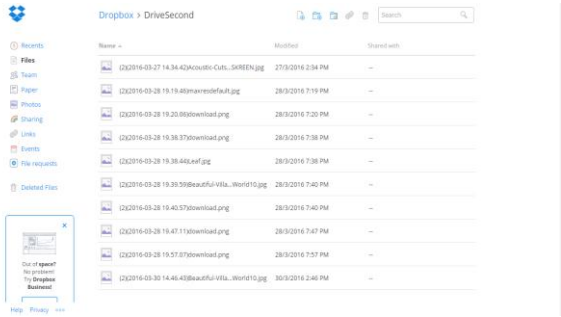


Splited files:

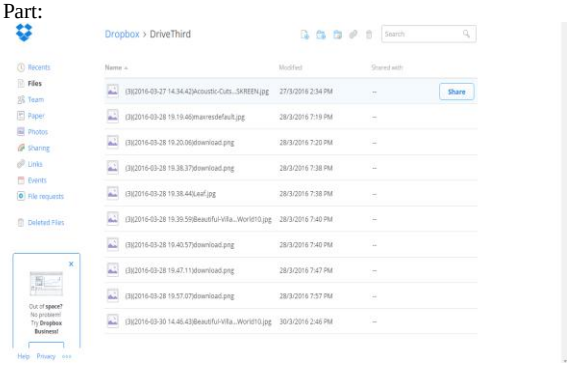
First Part:



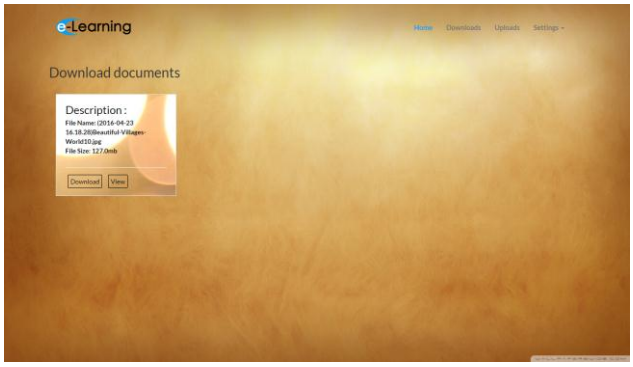
Second Part:



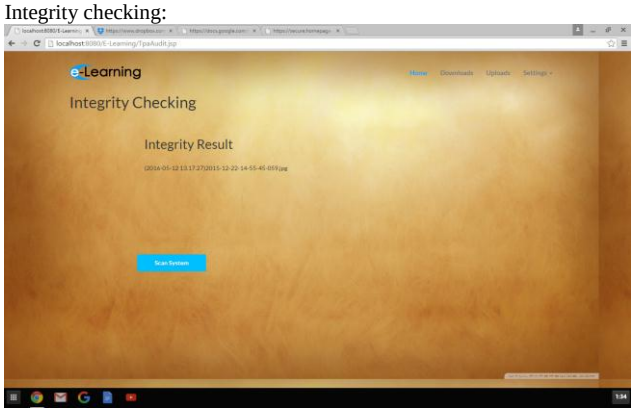
Third Part:



Downloaded Documents:



Integrity checking:



VIII. **FUTURE SCOPE**

We can work on this system in future also. In future we can balance the load based on the confidentiality of data required. In private cloud the data which is confidential is stored and on public cloud tae data which is not that much confidential is stored.

IX.CONCLUSION

The Problem was in earlier system there was no security, user need to be stay online and data lost may occurs. In this project we have developed, web system where anyone can store the data and checking load balancing with third party auditing. Thus, we have learnt about different types of cloud and how the data is secured on clouds. We are implementing and coming up with new system which will provide high security to cloud data using Third Party Audit. It will reduce the overhead of user by auto-uploading the packet if any packet is damaged even if user is offline.

REFRENCES

[1] A Load Balancing Algorithm For Private Cloud Storage ,4th ICCCNT 2013 ,July 4-6, 2013, Tiruchengode, India.

[2] Privacy-Preserving Public Auditing for Regenerating-Code-Based Cloud Storage IEEE transactions on information forensics and security, vol. 10, NO. 7, JULY 2015 Jian Liu, Kun Huang, Hong Rong, Huimei Wang, and Ming Xian.

[3]S. Yu, C. Wang, K. Ren, and W. Lou, Attribute based data sharing with attribute revocation, in Proc. 5th ACM Symposium on Information, Computer and Communications Security, Beijing, China, 2010, pp. 261– 270

[4]S. Ananthi, M.S. Sendil, and S. Karthik, Privacy preserving keyword search over encrypted cloud data, in Proc. 1st Advances in Computing and Communications, Kochi, India, 2011, pp. 480–487.

[5]L. Wang, L. Wang, M. Mambo, and E. Okamoto, New identity-based proxy re-encryption schemes to prevent collusion attacks, in Proc. 4th Int. Conf. Pairing-Based Crypto.

[6] W. Zeng, Y. Li, J. Wu, Q. Zhong and Q. Zhang. ”Load Rebalancing in Large-Scale Distributed File System.”IEEE 1st International Conference on Information Science and Engineering (ICISE), 2009, pp. 265-269.

[7] M. Armbrust et al., “Above the clouds: A Berkeley view of cloud computing,” Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-28, 2009.

[8] G. Ateniese et al., “Provable data possession at untrusted stores,” in Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS), New York, NY, USA, 2007, pp. 598–609.

[9] A. Juels and B. S. Kaliski, Jr., "PORs: Proofs of retrievability for large files," in Proc. 14th ACM Conf. Comput. Commun. Secur. 2007, pp. 584–597.

[10] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-replica provable data possession," in Proc. 28th Int. Conf. Distrib. Comput. Syst. (ICDCS), Jun. 2008, pp. 411–420.

[11] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A high-availability and integrity layer for cloud storage," in Proc. 16th ACM Conf. Comput. Commun. Secur., 2009, pp. 187–198.

[12] J. He, Y. Zhang, G. Huang, Y. Shi, and J. Cao, "Distributed data possession checking for securing multiple replicas in geographically dispersed clouds," J. Comput. Syst. Sci., vol. 78, no. 5, pp. 1345–1358, 2012.

[13] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote data checking for network coding-based distributed storage systems," in Proc. ACM Workshop Cloud Comput. Secur. Workshop, 2010, pp. 31–42.

[14] H. C. H. Chen and P. P. C. Lee, "Enabling data integrity protection in regenerating-coding-based cloud storage: Theory and implementation," IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 2, pp. 407–416, Feb. 2014.

[15] K. Yang and X. Jia, "An efficient and secure dynamic auditing protocol for data storage in cloud computing," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 9, pp. 1717–1726, Sep. 2013.

[16] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231–2244, Dec. 2012.

