CREATING SECURED MULTI-CLOUDS BY CONTINUOUS AUTOMATED AUDITING USING TERNARY HASH TREE IN AWS

[1]Jayalakshmi. S, [2]Sai Veena. K, [3] Harini. A,[4] Kavitha Subramani

[1]UG Scholar, [2] UG Scholar, [3]UG Scholar,[4]Associate Professor

[1]Department of Computer Science & Engineering,
[1]Panimalar Engineering College, Poonamallee, Chennai, India.

Abstract: There are countless choices for the way to store and access your data. There's the disc drive on your portable computer, the external disc drive you utilize for backing up and transferring information, network file shares, USB drives, and a lot of. With numerous storage choices accessible, what makes cloud storage unique? what makes cloud storage more captivating is that files are accessed and altered with ease. All you would require is a web affiliation and you'll access your files from any device, anywhere. Most of the organizations are moving to cloud storage to acquire advantage of the larger information availability, vital value savings, and information redundancy compared to conventional infrastructure. An important element of cloud data security is data integrity — preventing unauthorized modification or deletion and guaranteeing that knowledge remains unchanged since originally uploaded. This can be sometimes monitored by Third-party auditors appointed by the organizations. However, it can lead to high risks associated related to knowledge integrity which incorporates human errors(procrastination), corporate executive threats (the corruption of files), malicious intruders, compromised hardware, and configuration error, etc. Our resolution aims at resolving the preceding issues. During this work, we tend to plan a unique integrity verification framework in a multi-cloud atmosphere for securing cloud storage with the help of Ternary Hash Tree (THT) and Replica-based Ternary Hash Tree (R-THT). This helps to perform continuous data auditing at fixed time intervals which will be set by the admin. Differing from existing work, the auditing occurs at 3 levels-Block- level, File-level, and Replica-level with tree block ordering, storage block ordering for preserving data-integrity and guaranteeing data availableness within the cloud. The auditing processes have a flow, initial the parent block checking. If the parent block has any corrupted file, then the child node auditing happens. If the child nodes have any corrupted files, then the File recovery process will be carried out by the checker automatically if the information gets corrupted while checking. Users will cavil the cloud for file recovery. The outcomes depict that the planned secure cloud auditing framework is very secure and economical in storage, communication, and computation prices.

Index Terms **– Data-integrity, Automated auditing, Dynamic block split, File recovery, Block redistribution.**

I. INTRODUCTION

Nowadays, it has become a common practice for education, healthcare, and other sectors to utilize the cloud environment for storing and processing the business data, for acting as a backup repository to personal data such as photographs, mail archives, contact details, and so on. Cloud services provide a massive amount of computing resources and storage space, which makes the users eliminate the accountability of maintaining data at local machines. So, users completely depend on cloud services for their data storage, availability, and integrity [30]. Even though cloud computing infrastructures and service providers are much reliable and powerful, a wide range of threats to data integrity exists. Incidents of data loss and outages of significant cloud services appear day to day. Contrarily, the Cloud Service Provider (CSP) may face Byzantine failure problems occasionally, may remove hardly ever accessed files, and may decide not to showcase the data loss errors to DO for maintaining the reputation [1][2][3]. Unless the third-party auditor (TPA) or the Data Owner (DO) checks the coherence of the stored data, the extent of data confidentiality of the outsourced data is unknown. The major concern with the access of information in the cloud is to preserve data integrity at disloyal cloud servers. Many researchers [5-23] have proposed various schemes to address data integrity verification problems in the cloud data storage. In existing schemes, the role of verifiers falls either in private data auditing or in public data auditing. In private data auditing, DO takes the responsibility to monitor the precision of files saved in CSP. In public data auditing, DO rely on trusted third-party auditor for challenging CSP against integrity verification. Both types of auditing are performed without having

knowledge of actual data files. The inconsistent data blocks are capable of being found through data auditing techniques, but the methodology to localize the error blocks and correcting them across the replica blocks are left as an open issue.

II. RELATED WORK

This work has its foundation from previous works where the data of the client has been stored in the cloud and accessed with ease without the need to download the entire data in order to check the integrity of the data [1]. The system mentioned above relays on a third- party auditor, who continuously checks the integrity of the data [2]. Auditing protocols put forward are reliable and coherent, especially it curbs the charge of auditor. All the protocols used are privacy-preserving which doesn't reveal the data to the auditor [5]. It removes the burden of verification from the customer and eliminates the fear of data leakage [5]. It involves dividing the data into blocks and storing it into multiple servers. Of all these advantages, the disadvantages including procrastination of auditing by the auditor which may lead to data corruption and data loss. He may also get compromised by some other dealers which may lead to the trade of our private data. It also involves higher costs for the dedicated protocols (security related) used in the existing systems [1]. Moreover, the system does not immediately offer any guarantee on data integrity and availability and also insufficient to detect data corruption [6] [10].

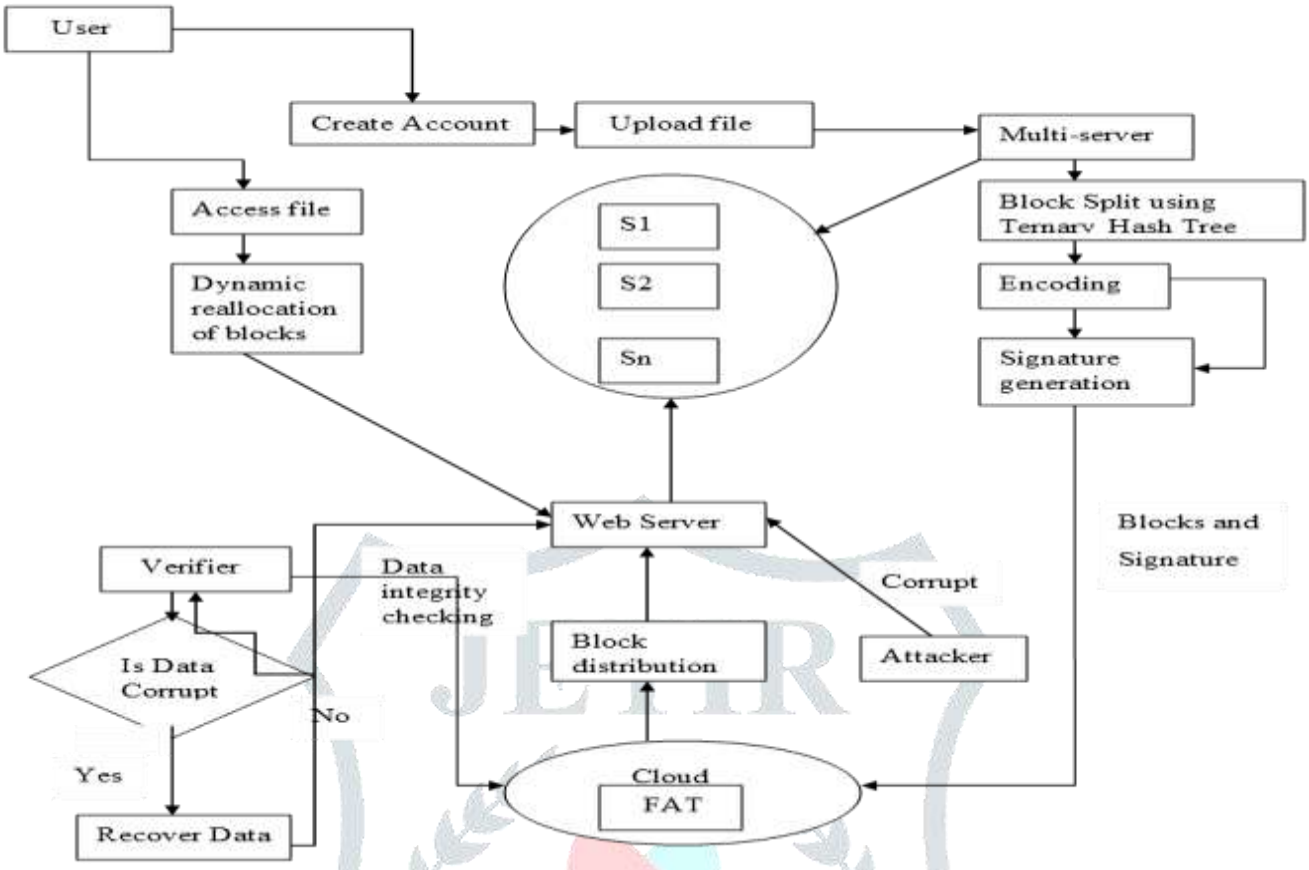
III. PROPOSED WORK

In our solution, the file which is uploaded by the user will get split into blocks. This is done by Dynamic Block Generation algorithm. These blocks will be stored in multiple clouds in Ternary Hash Tree format. In this tree, the root hash value is used to check data integrity of the data blocks. FAT (FILE ALLOCATION TABLE FILE SYSTEM) has all the metadata with proper indexing for various chunks of stored file. Time interval will be fixed by the admin. The continuous auditing process will be carried out based on the fixed time interval. This helps the verifier to perform Block and File level checking. If the attacker corrupts the data, it will be notified in FAT file system as "File corrupted" and then the file will be recovered. The continuous auditing system has a hierarchy. Firstly, parent block will be checked. If it has any corrupted file, then child node auditing will be done. If it also has any corrupted file, then file recovery is done. If the corrupted block is not checked during the particular time interval, then user can get their file recovered when they try to download it.

IV. MODULES

- Configuration of server
- Uploading data and Block Split
- File Integrity checking and update in FAT
- Automatic recovery of file

V. SYSTEM ARCHITECTURE



VI. CONFIGURATION OF SERVER

Admin has the access to configure multi-cloud server. For each cloud, server IP address and Port number will be assigned by the admin. Then a server architecture will be created after the number of servers selected by the admin. This setup can be re-configured by the admin if he wishes to do so. FAT file will be modified accordingly once the server is re-configured. Time interval for continuous auditing will be set by the admin. Based on the fixed time interval, data integrity checking process will take place.

VII. UPLOADING DATA AND BLOCK SPLIT

Initially, user need to register to use the web application. Personal details of the user are required to complete the registration process. The server in turn stores all these details in the database. Once the registration is done, the user can login using the username and password and can upload the files in the server. Once the file is uploaded by the user, it gets split into blocks using Dynamic block generation algorithm. Before storing the data in FATFS, each block will be appended with a signature using MD5 algorithm. After the data is stored in cloud, encoding is done by Base64 algorithm.

VIII. UPLOADING DATA AND BLOCK SPLIT

Initially, user need to register to use the web application. Personal details of the user is required to complete the registration process. The server in turn stores all these details in the database. Once the registration is done, the user can login using the username and password and can upload the files in the server. Once the file is uploaded by the user, it gets split into blocks using Dynamic block generation algorithm. Before storing the data in FATFS, each block will be appended with a signature using MD5 algorithm. After the data is stored in cloud, encoding is done by Base64 algorithm.

IX. FILE INTEGRITY CHECKING AND UPDATE IN FAT

For various chunks of data which is uploaded by the user, FATFS will have proper indexing and metadata. Remote Integrity Checking for these data will be performed by the verifier. During this checking process, only random blocks will be checked instead of checking the entire file. Since only random blocks are checked and not the entire file, it will protect the user privacy from third party verifier. The data-integrity checking is done in two steps namely Block level and File level.

BLOCK LEVEL CHECKING

In block level integrity checking, totally three signatures have been created.

- Signature of the block retrieved from the FATFS.
- New signature is appended for the block which is to be checked.
- Signature which is already stored in the cloud is appended with the signature fetched from the block

For block level Integrity checking, all of these above three signatures have been reviewed.

FILE LEVEL CHECKING

All these block contents are joined so as to perform File level verifying process.

X. AUTOMATED FILE RECOVERY

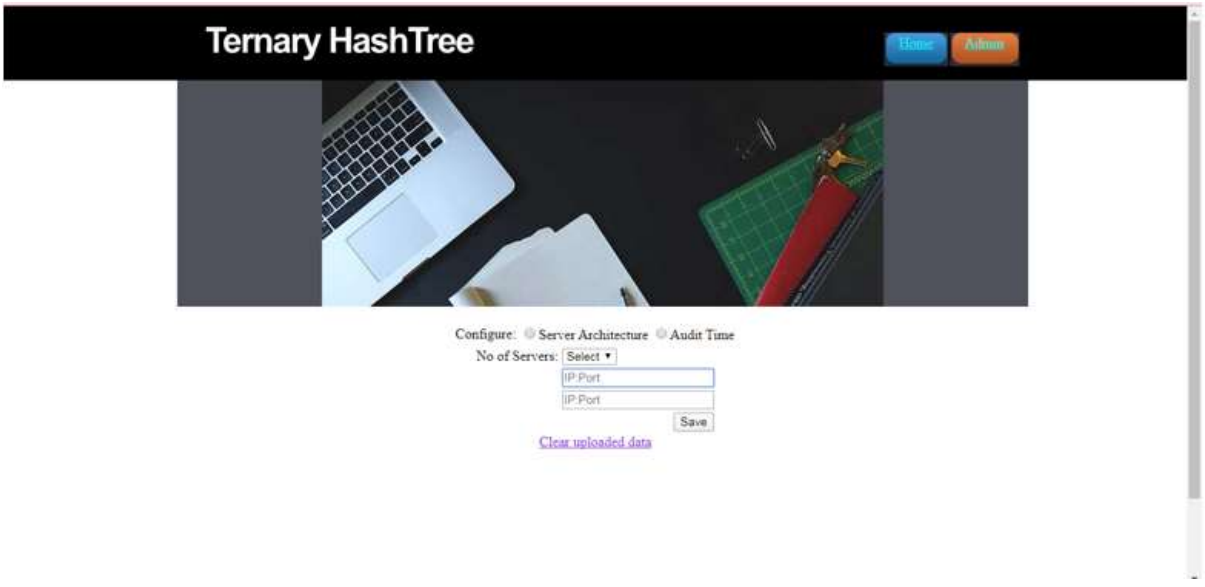
If any one of the blocks in the cloud server is corrupted by the attacker, it can be found during data integrity checking which is done by the verifier, Then the file recovery process is done automatically. If that particular corrupted block is not checked during random block checking, it can be found and notified when the user tries to download the file and then file recovery takes place. Every time when the user tries to access the file, it gets reallocated dynamically to ensure access confidentiality in the cloud and FAT file will get updated accordingly. Auditor will perform continuous auditing based on the time interval and certificate will be provided based on the cloud performance. When a new user wishes to register, they can view the certificate and then decide to create an account in the cloud.

XI. ADVANTAGES

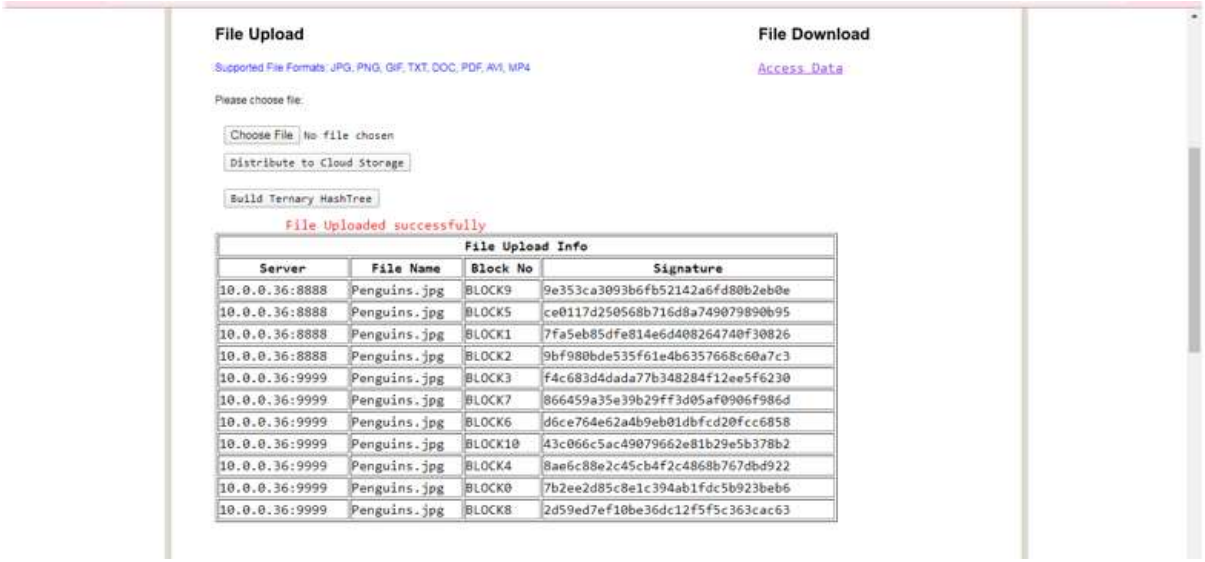
- Automatic file reclamation
- Dynamic block split
- Preserving Access Confidentiality: Every time when the user accesses the file, the blocks will be reallocated dynamically.
- Privacy protection.

XII. EXPERIMENTAL RESULTS

1. SERVER CONFIGURATION PAGE



2. FILE UPLOAD PAGE

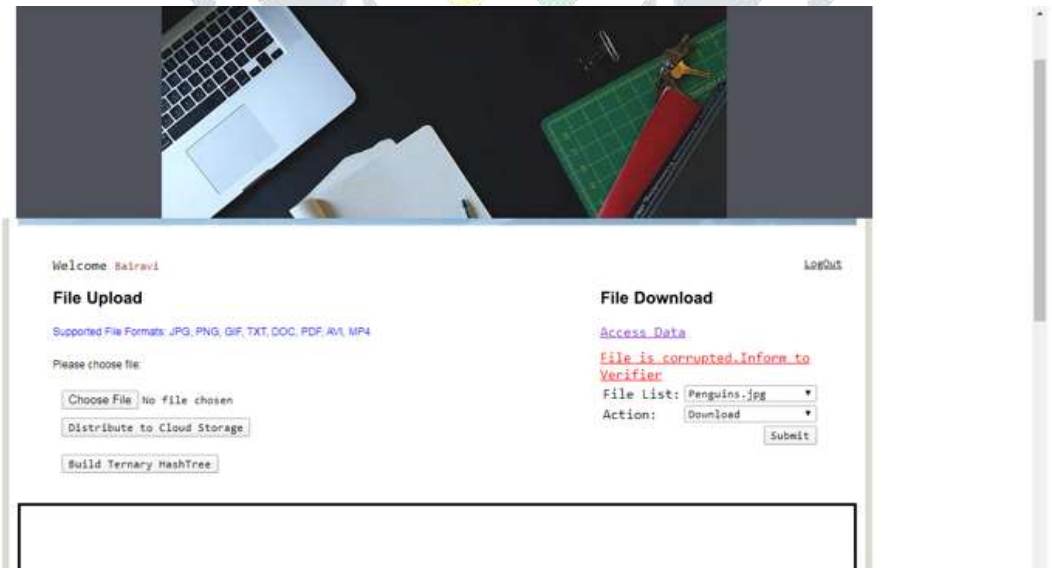


File Upload Info

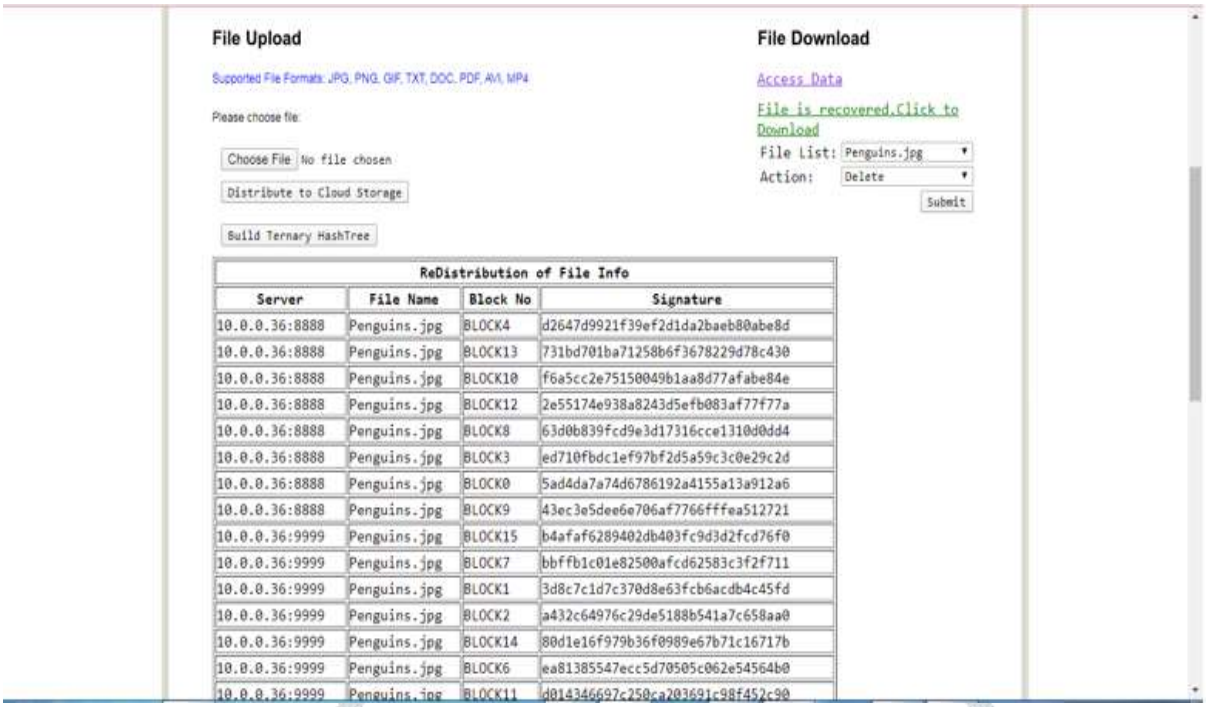
Server	File Name	Block No	Signature
10.0.0.36:8888	Penguins.jpg	BLOCK9	9e353ca3093b6fb52142a6fd80b2eb0e
10.0.0.36:8888	Penguins.jpg	BLOCK5	ce0117d250568b716d8a749079890b95
10.0.0.36:8888	Penguins.jpg	BLOCK1	7fa5eb85dfe814e6d408264740f30826
10.0.0.36:8888	Penguins.jpg	BLOCK2	9bf980bde535f61e4b6357668c60a7c3
10.0.0.36:9999	Penguins.jpg	BLOCK3	f4c683d4dada77b348284f12ee5f6230
10.0.0.36:9999	Penguins.jpg	BLOCK7	866459a35e39b29ff3d05af0906f986d
10.0.0.36:9999	Penguins.jpg	BLOCK6	d6ce764e62a4b9eb01dbfcd20fcc6858
10.0.0.36:9999	Penguins.jpg	BLOCK10	43c066c5ac49079662e81b29e5b378b2
10.0.0.36:9999	Penguins.jpg	BLOCK4	8ae6c88e2c45cb4f2c4868b767dbd922
10.0.0.36:9999	Penguins.jpg	BLOCK0	7b2ee2d85c8e1c394ab1fdc5b923beb6
10.0.0.36:9999	Penguins.jpg	BLOCK8	2d59ed7ef10be36dc12f5f5c363cac63

3. FAT FILE SYSTEM PAGE

4. WHEN USER TRIES TO DOWNLOAD A CORRUPTED PAGE, IT DISPLAYS “FILE IS CORRUPTED. INFORM TO VERIFIER”



5. FILE RECOVERY NOTIFICATION



ReDistribution of File Info

Server	File Name	Block No	Signature
10.0.0.36:8888	Penguins.jpg	BLOCK4	d2647d9921f39ef2d1da2baeb80abe8d
10.0.0.36:8888	Penguins.jpg	BLOCK13	731bd701ba71258b6f3678229d78c430
10.0.0.36:8888	Penguins.jpg	BLOCK10	f6a5cc2e75150049b1aa8d77afabe84e
10.0.0.36:8888	Penguins.jpg	BLOCK12	2e55174e938a8243d5efb083af77f77a
10.0.0.36:8888	Penguins.jpg	BLOCK8	63d0b839fcd9e3d17316cce1310d0dd4
10.0.0.36:8888	Penguins.jpg	BLOCK3	ed710fbdc1ef97bf2d5a59c3c0e29c2d
10.0.0.36:8888	Penguins.jpg	BLOCK0	5ad4da7a74d6786192a4155a13a912a6
10.0.0.36:8888	Penguins.jpg	BLOCK9	43ec3e5dee6e706af7766fffea512721
10.0.0.36:9999	Penguins.jpg	BLOCK15	b4afaf6289402db403fc9d3d2fcd76f0
10.0.0.36:9999	Penguins.jpg	BLOCK7	bbffb1c01e82500afcd62583c3f2f711
10.0.0.36:9999	Penguins.jpg	BLOCK1	3d8c7c1d7c370d8e63fcb6acdb4c45fd
10.0.0.36:9999	Penguins.jpg	BLOCK2	a432c64976c29de5188b541a7c658aa0
10.0.0.36:9999	Penguins.jpg	BLOCK14	80d1e16f979b36f0989e67b71c16717b
10.0.0.36:9999	Penguins.jpg	BLOCK6	ea81385547ecc5d70505c062e54564b0
10.0.0.36:9999	Penguins.jpg	BLOCK11	d014346697c250ca203691c98f452c90

XIII.CONCLUSION

The continuous File level, Block level and replica level auditing are performed for ensuring data integrity of the files which is uploaded. Since we are picking few random blocks to perform auditing, it helps in making sure the computational efficiency of our system. Data consistency is preserved with the help of replica level auditing in the cloud. When the corrupted blocks are localized during continuous auditing, it is corrected to fulfil the requirement of a real time application. Most importantly, privacy of the user data is preserved, since we choose blocks for checking in a random order. So that the TPA cannot access the complete data of the user. This helps our system to overcome the drawbacks of the existing work. Our future goal is to implement this application for data sharing and therefore to ensure the integrity of the data which is shared and accessed by the group of users.

XIV.REFERENCES

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z.Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS'07), pp. 598-609, Oct. 2007.

[2] M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop Hot Topics in Operating Systems (HotOS '07), pp. 1-6, 2007.

[3] M.A. Shah, R. Swaminathan, and M. Baker, "Privacy-Preserving Audit and Extraction of Digital Contents," Cryptology ePrint Archive, Report 2008/186, http://eprint.iacr.org, 2008.

[4] Vladimir A. Dobrushkin, "Methods in Algorithmic Analysis", CRC Press, 2009.

[5] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing", IEEE Transactions on Parallel and Distributed Systems, Vol. 22, No. 5, 847-859, 2011.

[6] Hao, Z., Zhong, S., and Yu, N, "A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability", IEEE Transactions on Knowledge and Data Engineering, Vol.23, 1432 – 1437, 2011.

[7] Y. Zhu, G. J. Ahn, and Z. Hu, "Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds", IEEE Transactions on Services Computing, Vol. 6, 227-238, 2011.

[8] Wang, C., Wang, Q., Ren, K., Cao, N., and Lou, W, "Toward secure and dependable storage services in cloud computing", IEEE Transactions on Services Computing, Vol.5, No.2, 220 – 232, 2012.

[9] Cong Wang, Sherman S.M. Chow, Qian Wang, Kui Ren, and Wenjing Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage", IEEE Transactions on Computers, Vol.62, No.2, 362-375, 2013.

[10] Yang, K., and Jia, X, "An efficient and secure dynamic auditing protocol for data storage in cloud computing", IEEE Transactions on Parallel and Distributed Systems, Vol.24, No.9, 1717 – 1726, 2013.

[11] H. Wang, "Proxy Provable Data Possession in Public Clouds", IEEE Transactions on Services Computing, Vol.6, No.4, 551 – 559, 2013.

[12] Boyang Wang, Baochun Li, and Hui Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud", IEEE Transactions on Cloud Computing, Vol.2, No.1, 2014.