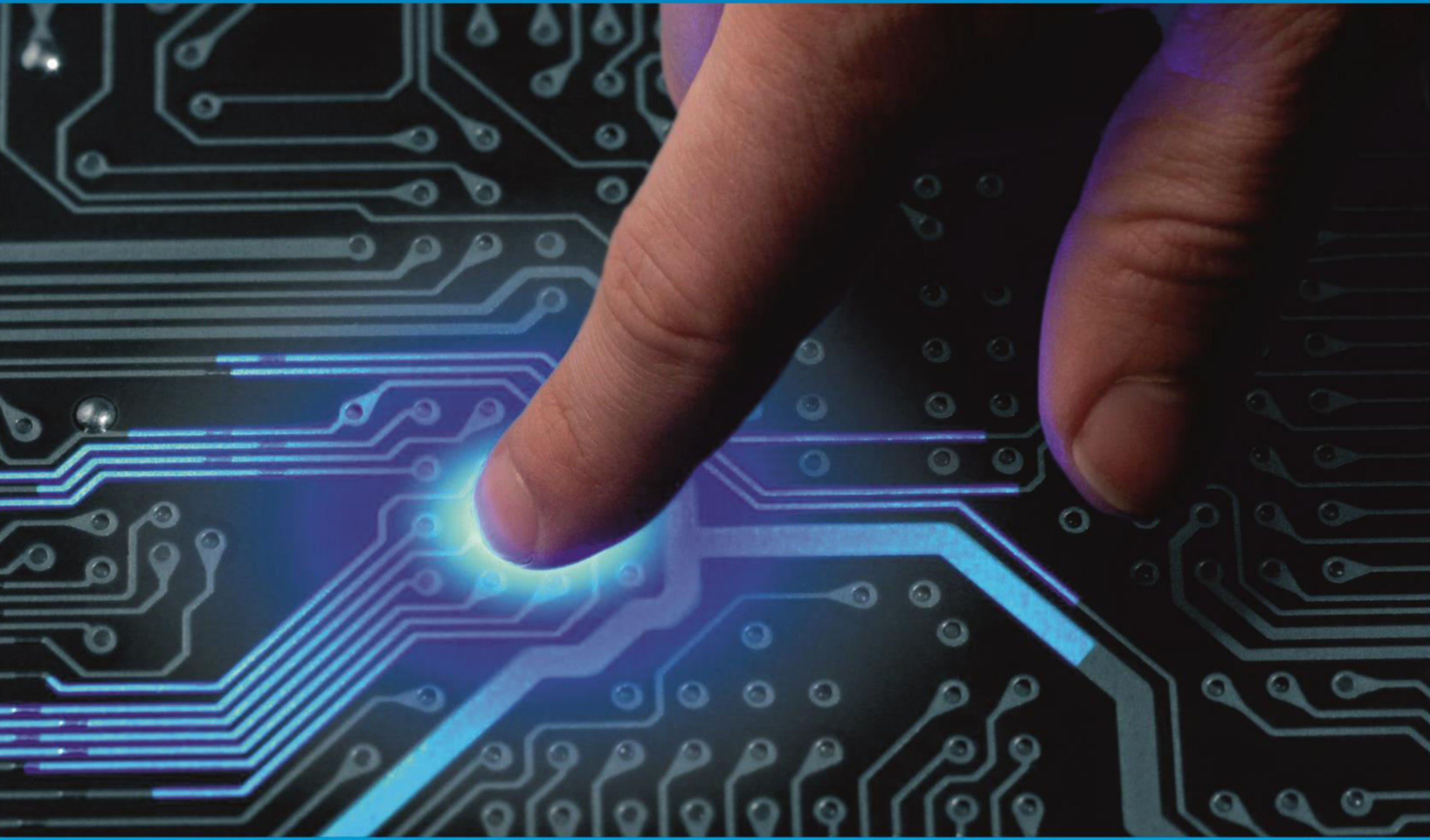
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 3, March 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.488

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903159 |

Enabling Secure Clouds by Continuous Auditing using Ternary Hash Tree

Maneesh K, Sanjay Doss K, Srinath K, Dr.K.Anbazhagan, M.Tech., Ph.D.

U.G. Student, Department of Computer Science and Engineering, Velammal Institute of Technology, Chennai, Tamil Nadu, India

Assistant Professor, Department of Computer Science and Engineering, Velammal Institute of Technology, Chennai, Tamil Nadu, India

ABSTRACT: Cloud Computing enables the remote users to access data, services, and applications in on-demand from the shared pool of configurable computing resources, without the consideration of storage, hardware, and software management. On the other hand, it is not easy for cloud users to identify whether Cloud Service Provider's (CSP) tag along with the data security legal expectations. So, cloud users could not rely on CSP's in terms of trust. So, we will be using the auditing process with the help of ternary hash tree. This process will be automatic and recovers the corrupted file at the requested time. The encrypted data of the uploaded files will split into blocks. Hashes will get generated for each separate block. The content inside the blocks will be encrypted and decrypted using MD5 (Message Digest) algorithm. The ternary hash tree and root hash will be formed using the SHA-1 algorithm. If the file is corrupted by the attacker or the third party, data will get easily recovered by using the hashes and the root hash in the file allocation table (FAT). So, the user can easily download or access the files they uploaded at the required time. The results show that the proposed secure cloud auditing framework is highly secure and efficient in storage, communication, and computation costs.

KEYWORDS: File Allocation Table, Ternary Hash Tree, Root Hash

I. INTRODUCTION

Cloud computing has rapidly become a widely adopted model for delivering services over the internet. As we know, cloud computing technology is used by both small and large organizations to store the information in cloud and access it from anywhere at any time using the internet connection. Cloud computing architecture is a combination of service-oriented architecture and event-driven architecture. Cloud computing architecture is divided into the following two parts Front End & Back End The below diagram shows the architecture ofcloud computing

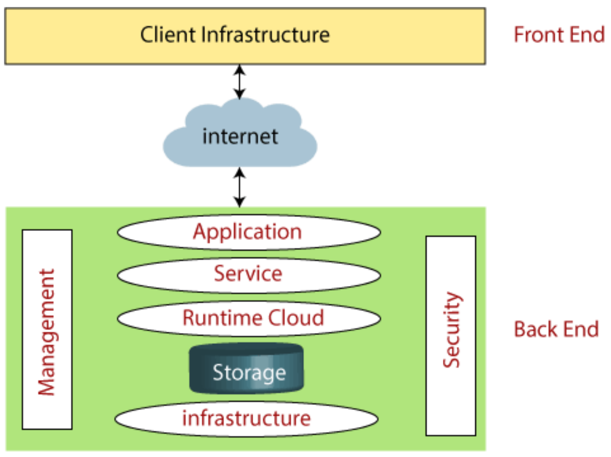


Fig.1 Architecture of Cloud Computing

The front end is used by the client. It contains client-side interfaces and applications that are required to access the cloud computing platforms. The front end includes web servers (including Chrome, Firefox, internet explorer, etc.), thin & fat clients, tablets, and mobile devices. The back end is used by the service provider. It manages all the resources that are required to provide cloud computing services. It includes a huge amount of data storage, security mechanism, virtual machines, deploying models, servers, traffic control mechanisms, etc.

II. LITERATURE SURVEY

In the year 2014, Mikael Asplund, Simin Nadjm-Tehrani, proposed a system with cloud storage services, it is commonplace for data to be not only stored in the cloud, but also shared across multiple users. However, public auditing for such shared data while preserving identity privacy remains to be an open challenge. In this paper, we propose the first privacy-preserving mechanism that allows public auditing on shared data stored in the cloud. In particular, we exploit ring signatures to compute the verification information needed to audit the integrity of shared data.

In another study at year 2007, Giuseppe Ateniese Randal Burns Reza Curtmola Joseph Herring† Lea Kissner Zachary Peterson Dawn Song proposed 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. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data.

In the year of 2010, Mehul A. Shah Ram Swaminathan Mary Baker HP Labs proposed a growing number of online services, such as Google, Yahoo!, and Amazon, are starting to charge users for their storage. Customers often use these services to store valuable data such as email, family photos and videos, and disk backups. Today, a customer must entirely trust such external services to maintain the integrity of hosted data and return it intact. Unfortunately, no service is infallible. To make storage services accountable for data loss, we present protocols that allow a third party auditor to periodically verify the data stored by a service and assist in returning the data intact to the customer. Most importantly, our protocols are privacy-preserving, in that they never reveal the data contents to the auditor

In the year of 2010 Cong Wang, Qian Wang, Kui Ren, Ning Cao, Wenjing Lou proposed that Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications without the burden of local hardware and software management. Though the benefits are clear, such a service is also relinquishing users' physical possession of their outsourced data, which inevitably poses new security risks towards the correctness of the data in cloud. In order to address this new problem and further achieve a secure and dependable cloud storage service, we propose in this paper a flexible distributed storage integrity auditing mechanism, utilizing the homomorphic token and distributed erasure-coded data. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost. The auditing result not only ensures strong cloud storage correctness guarantee, but also simultaneously achieves fast data error localization, i.e., the identification of misbehaving server. Considering the cloud data are dynamic in nature, the proposed design further supports secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append. Analysis shows the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

III. PROPOSED METHODOLOGY & DISCUSSION

In cloud computing, remote data integrity checking is an important security problem. The client's massive data is outside his control. The malicious cloud server may corrupt the client's data to gain more benefits. However, cloud services are part of an ever- changing environment, resulting from fast technology life cycles and inherent cloud computing (CC) characteristics, like on-demand provisioning and entangled supply chains. Hence, such long validity periods may put in doubt reliability of issued certifications. And cloud service customers do not longer possess their data locally, assuring that their data is being correctly stored, and integrity is maintained in cloud environments is of critical importance. Data integrity may be threatened by, for example, malicious insiders, data loss, technical failures, and by external attackers.

Limitations:

- Long validity periods may put in doubt reliability of issued certifications,

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903159 |

- Data Integrity
- Procrastinating third-party auditors.

In MultiCloud environment, remote data integrity checking is required to secure user's data. User will upload file to Cloud. This file is split into blocks using Dynamic Block generation Algorithm. The Blocks are stored in Ternary Hash Tree (THT) format. The blocks have a parents node and child node. File Allocation Table (FAT) File System has proper Indexing and Metadata's for the different Chunks of the Cloud Storage. Here the auditor agrees to inspect logs, which are routinely created during monitoring operations by services providers to assess certification adherence. If Attacker corrupts data in MultiCloud, the continuous auditing process helps the verifier to perform Block and File level checking for remote data Integrity Checking using Verifiable Data Integrity Checking Algorithm. The auditing processes have a flow, first the parent block checking. If the parent block has any corrupted file then the child node auditing. If the child nodes have any corrupted file, the File recovery is done by the Verifier automatically if the data gets corrupted during checking.Users can complaint cloud for file recovery.

Advantages:

- The proposed framework performs Block-level, File-level and Replica-level auditing with tree block ordering.
- Our framework to support error localization with data correctness, dynamic updates with block update, insert and delete operations in the cloud.

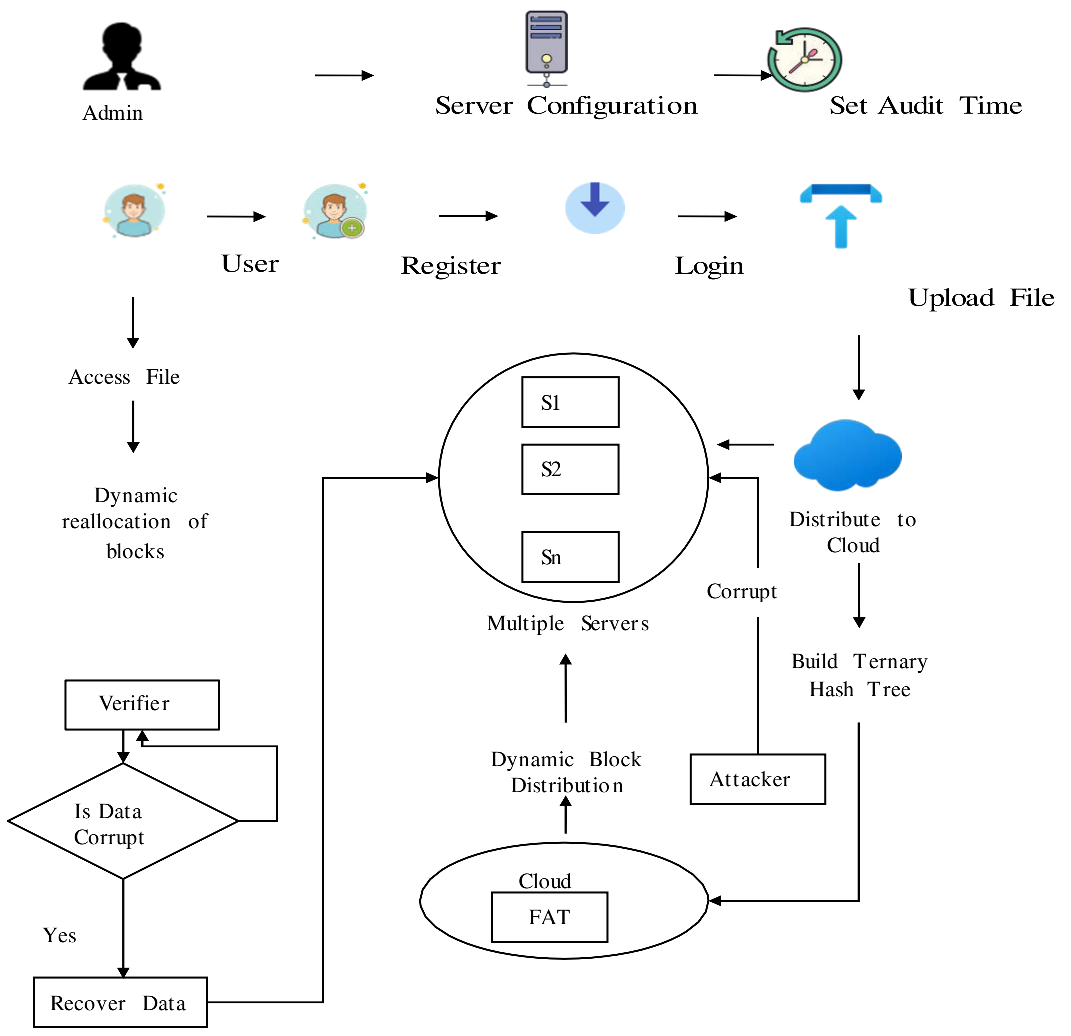


Fig.2 Architecture of Proposed System

Admin Configuration

Admin configure Multi-Cloud server setup. Server IP Address and Port number is given by the admin for each Cloud. Now a Server Architecture is created for Multi-Cloud Storage. If the admin has to reconfigure the old Multi-Cloud server setup, it can be done. For old server setup, FAT file can be modified or remain same. Audit time will be set by the admin for Data Integrity checking process.

Formation of Ternary Hash Tree

User has an initial level Registration Process at the web end. The users provide their own personal information for this process. The server in turn stores the information in its database. After Registration, user can upload files to the server. Uploaded files will be stored in a Server. When the user uploads the data to different cloud by the time it is splitted into different blocks using dynamic block generation Algorithm and each block will be appended with Signatures before storing the data in FATFS. Signature generated using MD5 Algorithm.Also, the data gets encoded using for Base64 Algorithm.

File Status Checking (Public Auditing)

FATFS has proper Indexing and Metadata's for the different Chunks of the Data that is being uploaded by User. Verifier performs Remote Integrity Checking on Cloud Data. Cloud allocates random combination of all the blocks to the Verifier, instead of the whole file is retrieved during integrity checking. This is to protect user privacy from a third party (Verifier). Verifiable Data Integrity Checking Algorithm is done in two steps: Block Checking and File Checking. In Block Checking step: Three signatures are generated for Block level Checking.

- A Signature of a block retrieved from a FATFS
- A new signature is generated for block to be checked
- A Signature is retrieved from the block appended with the signature which is stored in the cloud

The above three signatures are cross checked for Block level Integrity Checking. And the block contents are appended to verify with File level Integrity Checking.

File Attack and Recovery

Attacker can corrupt data in any one of the cloud servers. On Data Integrity Checking done by the Verifier, Verifier informs Corrupted blocks to the Cloud. Recovery Process will be done by the verifier automatically when data gets corrupted. User can complaint to the Cloud if the user file get corrupted (Verifier doesn't perform checking on this file).Whenever user access file, Blocks will be reallocated dynamically to provide access confidentiality in cloud and FAT File System will get updated. Auditor will monitor the cloud continuously and they provide the certificate based on the cloud performance. When new user joins in the cloud they will read the certificate and then they can create an account in the cloud.

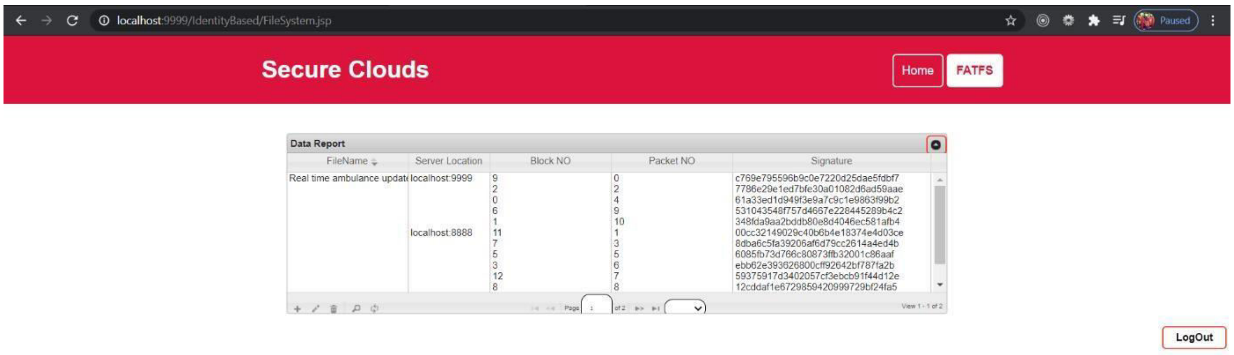
| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0903159 |

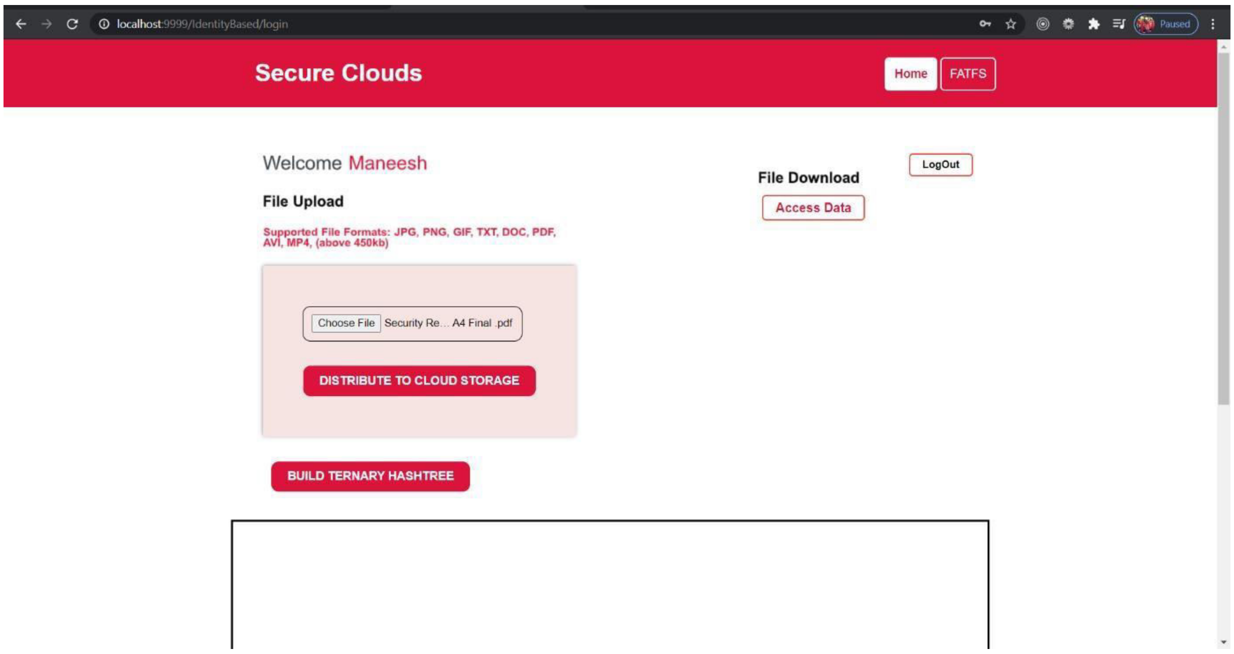
IV. EXPERIMENTAL RESULTS

The following figures shows enabling secure clouds by continuous auditing using ternary hash tree.



FileName	Server Location	Block NO	Packet NO	Signature
Real time ambulance updat	localhost:9999	9	0	c769e795596b9c0e7220d25dae5fdbf7
		2	2	7786e29e1ed7bfe30a01082d6ad59aae
		0	4	61a33ed1d949f3e9a7c9c1e9863f99b2
		6	9	531043548f757d4667e228445289b4c2
		1	10	348fda9aa2bddb80e8d4046ec581afb4
	localhost:8888	11	1	00cc32149029c40b6b4e18374e4d03ce
		7	3	8dba6c5fa39206af6d79cc2614a4ed4b
		5	5	6085fb73d766c80873ffb32001c86aaf
		3	6	ebb62e393626800cff92642bf787fa2b
		12	7	59375917d3402057cf3ebcb91f44d12e
		8	8	12cddaf1e672985942099729bf24fa5

Fig 3. Shows the FATFS table which has filename, Server location, Block & Packet No, Signature.



Welcome Maneesh

Supported File Formats: JPG, PNG, GIF, TXT, DOC, PDF, AVI, MP4, (above 450kb)

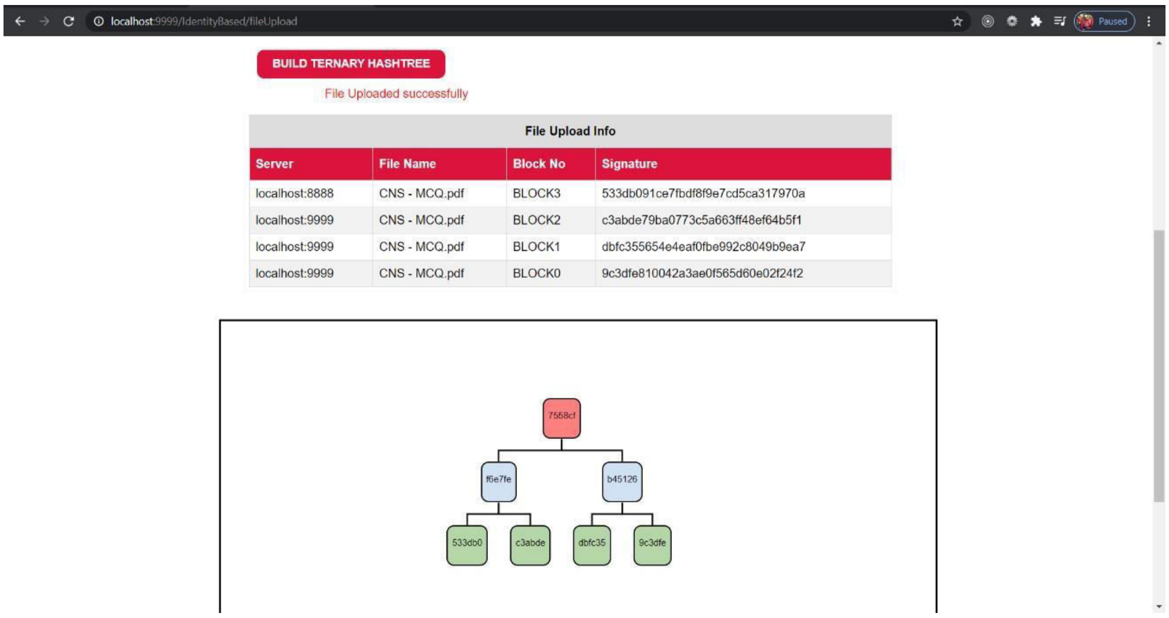
Security Re... A4 Final .pdf

Fig 4. File Upload Page

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.488 |

|| Volume 9, Issue 3, March 2021 ||

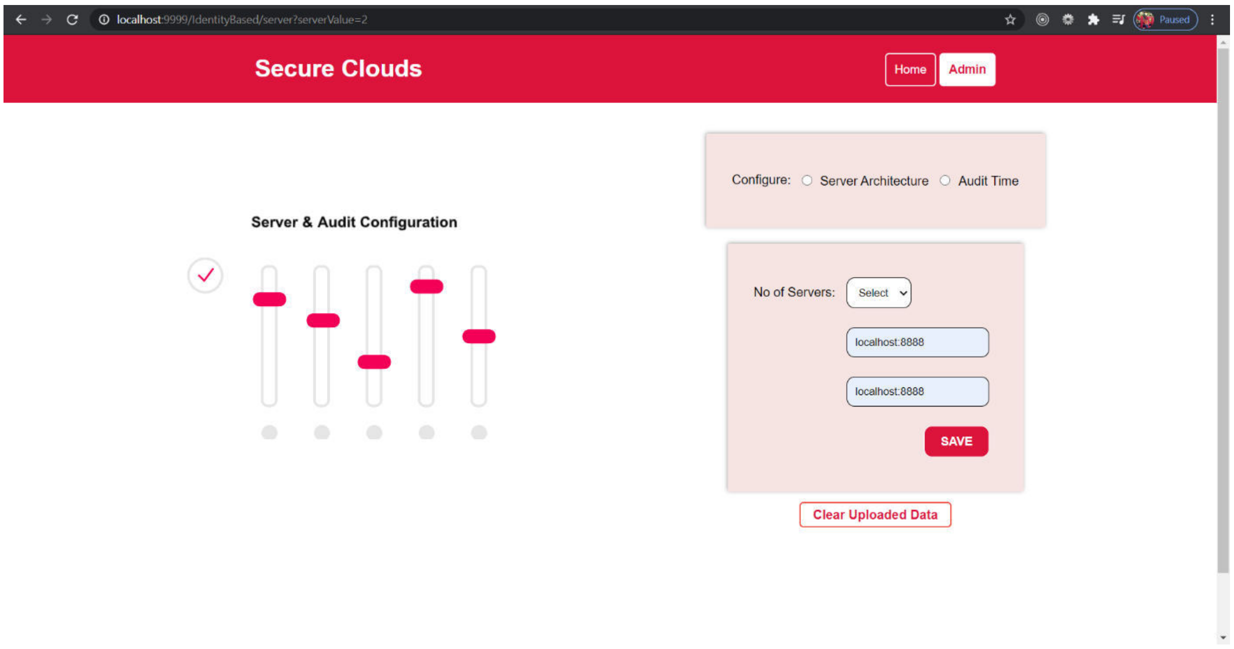
| DOI: 10.15680/IJIRCCE.2021.0903159 |



File Uploaded successfully

File Upload Info			
Server	File Name	Block No	Signature
localhost:8888	CNS - MCQ.pdf	BLOCK3	533db091ce7fbdf8f9e7cd5ca317970a
localhost:9999	CNS - MCQ.pdf	BLOCK2	c3abde79ba0773c5a663ff48ef64b5f1
localhost:9999	CNS - MCQ.pdf	BLOCK1	dbfc355654e4eaf0fbe992c8049b9ea7
localhost:9999	CNS - MCQ.pdf	BLOCK0	9c3dfe810042a3ae0f565d60e02f24f2

Fig 5. Shows the File upload info & Ternary Hash tree built from the uploaded file.


localhost:9999/IdentityBased/server?serverValue=2
Secure Clouds
Home
Admin
Server & Audit Configuration
SAVE
Clear Uploaded Data

Configure: ○ Server Architecture ○ Audit Time

No of Servers: Select

localhost:8888

localhost:8888

Fig 6. Admin Config Page

V. CONCLUSION

The data auditing in file level, block level, and replica level are achieved for verifying data integrity of the entire file. Few blocks for frequent audit tasks making it computationally efficient and replica level auditing to ensure data consistency across all the replicas in the cloud respectively. Further, the corrupted blocks identified during auditing is localized and corrected to suit the need for real-time applications. Moreover, public auditing preserves the privacy of user data from TPA through the random ordering of the blocks being unknown to TPA and CS. Further data dynamics is performed maintaining public verifiability on the same with reduced complexity which was better than the existing schemes.

VI. FUTURE ENHANCEMENT

The future work is to extend the framework for data sharing where data integrity is verified over the shared data across the user groups and to automate the server configuration for each groups to get rid of mundane configurations among the groups.

REFERENCES

[1] Thangavel M, Varalakshmi P "Enabling Ternary Hash Tree based Integrity Verification for Secure Cloud Data Storage", IEEE Transaction on Knowledge & Data Engineering, Dec 2020

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

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

[4] 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.

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

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

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

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

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

[10] 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.

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

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

[13] Zhu Y, 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.