EasyChair Preprint

№ 12593

Ternary Hash Tree Integrity Verifier to Secure Cloud Data

Teja Venkata Satyanarayana Sayyapureddy,
Bharath Naidu Kommineni, Jaya Krishna Javvadi,
Shiva Bhaskar Reddy Balapalapalli and Bhavika Darji

March 19, 2024

TERNARY HASH TREE INTEGRITY VERIFIER TO SECURE CLOUD DATA

SAYYAPUREDDY TEJA VENKATA SATYANARAYANA
Computer Science Engineering (CSE),
Parul University, Gujarat, India
teja.sayyapureddy@gmail.com

BHARATH NAIDU KOMMINENI
Computer Science Engineering (CSE),
Parul University, Gujarat, India
bharath224923@gmail.com

JAVVADI JAYA KRISHNA
Computer Science Engineering (CSE),
Parul University, Gujarat, India
jayakrishnajavvadi@gmail.com

BALAPALAPALLI SHIVA BHASKAR REDDY
Computer Science Engineering (CSE),
Parul University, Gujarat, India
shivashivabhaskar282@gmail.com

Prof. BHAVIKA DARJI, Assistant Professor
Dept. Of Computer Science Engineering
Parul University, Gujarat, India
bhavikaben.darji19384@paruluniversity.ac.in

***Abstract*—Cloud computing has transformed the landscape of data access, services, and applications, providing users with remote access to configurable computing resources. Despite these advancements, ensuring compliance with data security regulations by Cloud Service Providers (CSPs) remains a pressing concern for users, impacting their trust in these providers. To address this challenge, there is a critical need for a robust data auditing framework to reinforce users' confidence in CSPs. This study proposes integrating Third Party Auditors (TPAs) to independently verify the integrity of outsourced data, thereby alleviating the computational burden on cloud users.**

This paper presents an innovative integrity-checking framework to provide security for cloud data, built upon the principles of the Replica-based Ternary Hash Tree and ternary hash tree. TPAs utilize this framework for data auditing, distinguishing it from existing methods. Notably, our framework supports Node-level and document-level auditing with tree Node ordering and storage Node ordering to ensure data Security and availability in the cloud. Moreover, the framework facilitates error detection, ensures data Integrity, and accommodates updates, including Node updates, inserts, and deletions within a cloud environment. Compared to existing schemes, the THT and R-THT structures offer reduced computational costs and enhanced data update efficiency.

A comprehensive security analysis of the proposed public auditing framework confirms its ability to achieve desired properties. We further evaluate its performance through rigorous experimentation, demonstrating that the proposed secure cloud auditing framework excels in security and efficiency while minimizing storage, communication, and computation costs.

I. INTRODUCTION

In this research endeavor, we present the concept of "Enhancing Data Integrity Verification in Secure Cloud Storage through Ternary Hash Trees." Diverse industries, such as education and healthcare, have embraced cloud environments as integral components for their data storage and operational processes. Cloud services provide an extensive reservoir of computational capabilities and storage capacity, thereby liberating users from the responsibilities associated with local data management. Consequently, the reliance on cloud services for storage, data availability, and data security has grown significantly.

Despite the robustness and capabilities of the cloud and the service provider, the threat landscape surrounding data integrity remains a concern. Incidents of data loss and service outages, even in major cloud platforms, continue to be reported. Furthermore, Cloud Service Providers (CSPs) may encounter challenges like Byzantine failures, periodic file removals, and the withholding of data loss incidents from Data Owners (DOs) for reputation preservation. In the absence of third-party auditing or direct validation by Data Owners, the confidentiality and integrity of outsourced data remain uncertain. Consequently, the paramount concern when accessing data in cloud storage is ensuring its security, particularly in the context of untrusted cloud servers.

A. PROBLEM STATEMENT

In a Cloud, ensuring the security of data is paramount to safeguarding user data. When a user uploads a file to the cloud, it undergoes a process of segmentation using a

Dynamic Nodes generation Algorithm, resulting in Nodes stored in a THT format. These blocks are organized with parent and child nodes. The auditor plays a crucial role in this system, agreeing to check logs regularly generated during checking operations by the service provider to have certification adherence. if an attacker attempts to corrupt data in the Cloud setup, a continuous auditing process is initiated, enabling the verifier to conduct Node and Document level checks for remote data Security Checking using a Verifiable Data Security Checking Algorithm.

The auditing process follows a defined flow: initially, the parent Node undergoes checking. If any corruption is detected within the parent Node, subsequent auditing moves to the child nodes. If any corrupted files are found within the child nodes, the Verifier automatically initiates File recovery procedures. Users are also empowered to file complaints with the cloud service provider for file recovery assistance, ensuring the integrity and security of their data throughout the auditing process.

B. SCOPE OF THE PROJECT

The scope of the project encompasses the development of a robust system for remote data integrity checking within a MultiCloud environment to ensure the security of user data. This involves implementing a Dynamic Block generation Algorithm to segment uploaded files into blocks, which are then organized and stored in a Ternary Hash Tree (THT) format with parent and child nodes. The File Allocation Table (FAT) File System is utilized to maintain proper meta-data and indexing for the various chunks of Cloud data.

The project also involves the integration of an auditing mechanism, where auditors routinely inspect logs generated by service providers during monitoring operations to assess adherence to certification standards. In the event of data corruption by an attacker within the MultiCloud environment, a continuous auditing process is initiated to facilitate Node and Document level checks for remote data Security using a Verifiable Data Security Checking Algorithm.

The auditing process follows a structured flow, starting with the examination of parent blocks. If any corruption is detected within the parent block, the auditing process proceeds to inspect the child nodes. Should any corrupted files be found within the child nodes, the Verifier automatically initiates File recovery procedures. Additionally, users have the option to file complaints with the cloud service provider for assistance with file recovery, ensuring accountability and data integrity throughout the auditing process.

C. OBJECTIVE OF THE PROJECT

The main objective of this project is to develop and implement a robust integrity verification framework for secure cloud storage, ensuring the integrity, confidentiality, and availability of user data. Key goals include enhancing data integrity through dynamic block management and error localization, enabling seamless auditing by integrating Third Party Auditors (TPAs), and preserving user privacy during integrity checks. The framework aims to be scalable, efficient, and user-friendly, facilitating easy interaction and monitoring of data integrity status. Additionally, the project seeks to conduct comprehensive security analyses and identify opportunities for future enhancements and integration to further improve the security, efficiency, and usability of cloud storage systems. Ultimately, the project aims to address critical challenges in cloud data security, fostering user trust and confidence in cloud computing technologies across various industries and applications.

II. MOTIVATION

A. Background and Related Work

The motivation behind this project stems from the growing reliance on cloud computing services across diverse industries, coupled with the persistent concerns regarding data integrity and security in cloud storage environments. As organizations increasingly shift towards cloud-based solutions for data storage and processing, ensuring the confidentiality, integrity, and availability of data becomes paramount. However, incidents of data breaches, unauthorized access, and data corruption continue to pose significant challenges, eroding user trust in cloud service providers. This project aims to address these challenges by developing a robust integrity verification framework that can effectively detect and mitigate data integrity threats in multi-cloud environments. By enhancing data integrity assurance through dynamic block management, third-party auditing, and privacy preservation mechanisms, the project seeks to instill confidence in cloud computing technologies and promote their widespread adoption across industries. Ultimately, the goal is to empower organizations with secure and reliable cloud storage solutions, enabling them to harness the benefits of cloud computing while safeguarding their valuable data assets.

III. LITERATURE REVIEW

Cloud computing is one of the transformative technologies which is a way to access data and services remotely and transformed traditional computing infrastructures. Nonetheless, securing data located in cloud environments remains a major concern. Ateniese et al. (2007) came up with an innovative idea on provable data possession (PDP) that allows clients to check their data integrity stored on untrusted servers without retrieving all datasets, among others. And these PDP models are designed for generating probabilistic proofs of possession by using random subsets of nodes, which significantly reduces I/O costs and network communication. There have been recent advances in this area such as Wang et al.'s (2011) and Hao et al.'s (2011), PDP models that have refined them more improving data integrity in distributed storage systems as well as cultivating trust into cloud environments.

Further, Wang et al. (2013) proposed privacy-preserving auditing enabled secure cloud data storage system with an aim of ensuring user privacy as well as data integrity. Such methods involve the use of homomorphic linear authenticators and random masking to keep auditors from knowing what is inside any stored data.

Moreover, resilient and efficient cloud storage systems are currently realized through advancing dynamic auditing protocols and multi-replica based verification schemes. In their paper, Yang and Jia (2013) proposed a secure dynamic auditing protocol for cloud data storage that was both efficient and adaptable to the dynamic nature of cloud storage. Their protocol integrates evolving nature in terms of fine-grained auditing mechanisms and dynamic updates aimed at maintaining the availability and integrity of cloud data in a changing environment. Similarly, for big data storage on the cloud, Liu et al. (2015) presented a top-down levelled multi-replica Merkle hash tree-based secure public auditing scheme that targets dynamic environments. Such actions will not only help in addressing the issues to do with unauthorized access to information or tampering but also prevent loss of data reliability or lack of accessibility.
Privacy-preserving auditing protocols emerge as a crucial area of research, aiming to address concerns surrounding data confidentiality and privacy in cloud storage systems. Wang et al. (2013) proposed a secure cloud data storage system equipped with privacy-preserving auditing capabilities, allowing users to remotely store their data while ensuring its integrity without compromising privacy. These protocols leverage homomorphic linear authenticators and random masking techniques to prevent third-party auditors from accessing the contents of stored data, thus preserving user privacy. Additionally, advancements in dynamic auditing protocols and multi-replica-based verification schemes contribute to the resilience and efficiency of cloud storage systems, offering robust solutions for ensuring data integrity and availability. Overall, the literature underscores the ongoing efforts to fortify cloud storage security while simultaneously addressing privacy concerns, paving the way for more reliable and trustworthy cloud storage services.

Moreover, Wang (2015) proposed a system where data possession across multiple cloud platforms is assured through an identity-based distributed approach. This enhances security and reliability in multi-cloud storage scenarios. Wang's method employs identity-based cryptography to streamline data possession verification, ensuring efficiency and scalability while keeping computational costs low. Additionally, Yu et al. (2015) introduced a mechanism designed to enable cloud storage auditing with resistance to key exposure. This serves to mitigate the risks linked to the exposure of cryptographic keys, thereby fortifying cloud storage systems against insider threats and unauthorized access, ultimately bolstering overall security.

To summarize, ongoing research endeavors are aimed at reinforcing the security of cloud storage systems through innovative strategies like provable data possession, privacy-preserving auditing protocols, and dynamic verification schemes. These advancements not only tackle issues related to data integrity, availability, and privacy but also improve the transparency and accountability of cloud storage services. By leveraging cutting-edge cryptographic techniques and principles of distributed computing, researchers are continuously laying the groundwork for more dependable and trustworthy cloud storage solutions. This signifies a significant step forward in ensuring secure data management within cloud environments.

IV. IMPLEMENTATION

In a Multi-Cloud setting, ensuring the integrity of user data is imperative. When a user uploads a file to the cloud, the file undergoes a segmentation process using a Dynamic Block Generation Algorithm. These resulting blocks are organized and stored in a Ternary Hash Tree structure, where each block has parent and child nodes. The File Allocation Table File System is utilized for robust indexing and metadata management for various chunks of the Cloud Storage.
In this context, an auditor collaborates to examine the logs routinely generated during monitoring activities by service providers to assess compliance with certification requirements. In the event of data corruption by an attacker within the Multi-Cloud environment, continuous auditing plays a crucial role in allowing the verifier to perform data integrity checks at both the Block and File levels using a Verifiable Data Integrity Checking Algorithm.

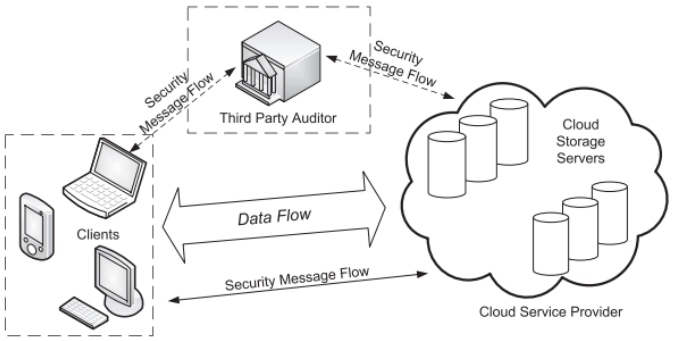

Fig. 1. System Architecture

The auditing process follows a structured sequence. Initially, it commences with a check of the parent block. If any corruption is detected within the parent block, a subsequent audit of the child nodes is triggered. In case any corruption is identified within the child nodes, the Verifier is equipped to initiate automatic file recovery procedures to rectify data corruption that may occur during the audit. This process ensures that users have a mechanism to request file recovery from the cloud provider if necessary.

A. *System Architecture and Working*

The system architecture of the proposed data integrity verification framework for secure cloud storage comprises multiple components designed to ensure robustness and reliability. At its core, the architecture involves a multi-cloud setup, where user data is segmented into blocks using a dynamic block generation algorithm before being organized and stored in a ternary hash tree (THT) structure. This architecture facilitates efficient indexing and metadata management using a File Allocation Table (FAT) File System. The system also integrates an auditing module, which collaborates with auditors to routinely monitor the cloud environment, ensuring compliance with certification requirements and facilitating data integrity checks at both block and file levels. In the event of data corruption, the system employs automatic recovery procedures, triggered by the verifier, to rectify any anomalies. Furthermore, user data access confidentiality is enhanced through dynamic block reallocation mechanisms, which update the FAT File System upon each access to maintain data privacy within the cloud environment. Overall, the architecture emphasizes proactive measures for data integrity verification, continuous monitoring, and swift recovery processes to mitigate potential risks and ensure the security and reliability of data stored in the cloud.

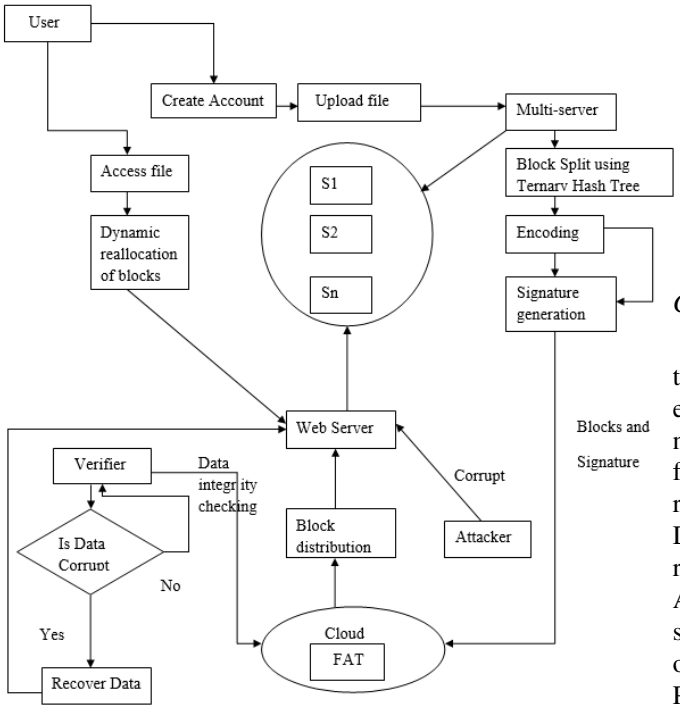

Fig. 2. System Architecture

B. *TECHNOLOGIES USED*

Algorithms

- **Base64 Algorithm:** Utilized for encoding binary data into ASCII characters to ensure safe transmission over text-based systems such as email or HTML forms. This algorithm converts binary data into a base64 character set consisting of letters, numbers, and symbols, enabling data to be represented and transmitted reliably across different platforms.
- **Message Digest (MD5) Dynamic block generation:** Employed for generating checksums or hash values from input data, providing a unique identifier for each block of data. This algorithm computes a fixed-size hash value from variable-sized input, ensuring data integrity and facilitating efficient data verification.

Software Requirements

- Operating System: Windows XP and Above
- Java Development Kit (JDK): Version 1.7
- Web Server: Apache Tomcat 6.0 and Tomcat 7.0
- Database Management System: MySQL 5.0

Hardware Requirements

- Hard Disk: Minimum 200GB storage capacity
- RAM: At least 2GB memory
- Processor: Intel Core i3 or equivalent processor

Technology Used

- J2EE eclipse for server-side development, incorporating JavaServer Pages (JSP) and Servlets for dynamic web content generation, tomcat server for running environment.
- Front-end development technologies including JavaScript, Ajax, HTML, CSS, and jQuery for creating interactive and user-friendly web interfaces and we use jstl to write backend logic along with web interface.
- Integration of Web Services using JAX-WS (Java API for XML Web Services) and JSON (JavaScript Object Notation) for efficient data exchange between heterogeneous systems.

C. *RESULT*

In summary, this study provides a comprehensive solution to the pressing issue of data integrity in cloud storage, employing a multifaceted auditing approach covering file, node, and replica levels. The selective auditing of blocks for common tasks optimizes computational efficiency, while replica-level auditing ensures consistency across cloud copies. Detecting and rectifying corrupted blocks, especially for real-time applications, reinforces robust data maintenance. Additionally, the adoption of public auditing techniques safeguards user data privacy by concealing random node ordering from Third Party Auditors and Cloud Service Providers. The incorporation of dynamic data updates, while maintaining public verification, streamlines processes and reduces complexity, surpassing existing methods like message authenticators which are very slow when compared to this tree data structure. In the broader context, cloud computing's increasing popularity stems from its convenience, offering computing resources without local storage and software management hassles. Yet, persistent data security concerns necessitate a dependable auditing framework to build and reinforce trust in CSPs, meeting legal expectations. The

proposed integrity verification framework, utilizing Tree data strucuture and Replica based Ternary Hash Tree data structure, empowers TPAs for thorough data audits, simultaneously performing Node-level, Document-level, and Replica-level audits. It ensures data integrity and availability through organized block and storage ordering. This framework excels in security, efficiency, and cost-effectiveness, meeting desired security standards. In conclusion, it offers a robust solution for cloud data security, instilling confidence in CSPs. Future directions may involve further enhancements and integration with complex data structures and existing cloud systems.

V. CONCLUSION AND FUTURE WORK

In conclusion, this study provides a comprehensive solution to the pressing issue of data integrity in cloud storage, employing a multifaceted auditing approach covering file, block, and replica levels. The selective auditing of blocks for common tasks optimizes computational efficiency, while replica-level auditing ensures consistency across cloud copies. Detecting and rectifying corrupted blocks, especially for real-time applications, reinforces robust data maintenance.

Looking ahead, enhancing the framework's resilience against potential attacks remains a priority. Implementing mechanisms to detect and counter data corruption attempts by malicious actors within cloud servers is crucial. Additionally, exploring proactive measures to preemptively safeguard against corruption incidents and bolster the recovery process will further fortify data integrity. Addressing user concerns about corrupted files is vital, necessitating an efficient reporting and resolution system. Developing a user-friendly interface that allows seamless reporting of corrupted files to the cloud, and implementing automated processes to rectify such issues will enhance user satisfaction and confidence in the system. Continuously optimizing access confidentiality is another avenue for future work. Dynamic reallocation of blocks upon user file access can be refined for further efficiency and security. Additionally, investigating techniques to minimize the impact of block reallocation on system performance is essential for a seamless user experience. While the ternary hash tree structure has a higher security level than binary hash trees, there is room for more research or study on additional security mechanisms. This could mean incorporating encryption methods, digital signatures or other cryptographic protocols to enhance overall system's safety. The project can be widened to accommodate complex cloud storage architectures such as object storage systems or distributed file systems. This would necessitate modifying the ternary hash tree-based integrity verification system so that it becomes interoperable with these architectures and finding out how it could work with existing ones; such as erasure coding or replication.

The role of auditors in ensuring cloud performance and security cannot be understated. Future efforts should focus on enhancing auditing techniques and methodologies to provide more accurate and timely evaluations. This may involve leveraging advanced analytics and monitoring tools to deliver comprehensive assessments and certifications based on cloud performance. Lastly, streamlining the onboarding process for new users through certificate-based account creation is a key area for improvement. Developing an intuitive and secure system for users to seamlessly access the cloud based on the provided certificates will optimize user experience and foster wider adoption

VI. CONCLUSION

A project on the same has been very successful in addressing this robustness of data integrity issues in cloud environment: ensuring that information once placed there does not get compromised. This is achievable through the application of ternary hash tree structure thus making it possible to verify whether the data saved in cloud is genuine.

The project also aimed at optimizing data retrieval and verification procedure. Through caching mechanisms and fine-tuning traversal algorithms, this system reduces the time required for verification operations. As a result, latency is minimized, overall performance improved thereby suitable for real-time storage of clouds.

This makes the system more secure against attacks like collision attack or manipulation of hash values; by including three different types of hash values at each level in a tree.

REFERENCES

[1] Xiaohong Wang, " Algorithm for Risk Assessment and Intervention of Mother to Child Transmission, 2011.

[2] Madelon L. Geurtsen, "High maternal early-pregnancy blood glucose levels are associated with altered fetal growth, 2020.

[3] J. Leigh Mills, J. Larry Simpson, S. Gerald Driscoll, et al., "Incidence of Spontaneous Abortion among Normal Women and Insulin-Dependent Diabetic Women Whose Pregnancies Were Identified within 21 Days of Conception," New England Journal of Medicine, 1988.

[4] Jill R. Weiss, "Can prepregnancy care of diabetic women reduce the risk of abnormal babies," Journal of the American Medical Association, 2018.

[5] Md. Tanvir Rahman, Md. Nurul Absar, Mohammad Hasan Imam, "Ensemble learning-based feature engineering to analyze maternal health during pregnancy and health risk prediction," PLOS ONE, 2023.

[6] Brinda Hansraj Sampat, Bala Prabhakar, "Privacy and Security Issues in Mobile Health Applications," Journal of Information and Technology Management, 2022.

[7] Stephanie L. Gaw, et al., "Digital Health Interventions for Pregnancy-Related Issues: A Systematic Review and Meta-Analysis," JAMA Internal Medicine, 2023.

[8] Aravindh Selvaraj, S. Suganya, L. Suguna, "Personalized Nutrition for Pregnant Women Using Machine Learning Techniques," Frontiers in Nutrition, 2023.

[9] Michael J. Roberts, Susan B. Roberts, Christopher J. Gardner, "Machine Learning-Based Dietary Assessment and Counseling in Clinical Care Settings," Annals of Family Medicine, 2023.

[10] Barua A, Kurata G, Finkelstein J, Chui K, "Personalized nutrition recommendations: By analyzing data on a woman's diet and nutritional status, machine learning algorithms can recommend specific foods and supplements that can help support a healthy pregnancy," 2019.

[11] Sara De Bruyne, Koenraad Cuypers, Jeroen Van den Bergh, "Maternal blood pressure in pregnancy, birth weight, and perinatal mortality in first births: prospective study," BMC Medicine, 2023.

[12] Christine M. Palmer, Stephanie L. Gaw, Kathryn D. Jhaveri, "Predictive modeling for adverse pregnancy outcomes using electronic health records and machine learning: a systematic review," Nutrients, 2023.

[13] Ruben A. Smith PhD, Lee Warner Ph.D., "The Pregnancy Risk Assessment Monitoring System (PRAMS): Overview of Design and Methodology," September 12, 2018.

[14] Egemen Ertugrul, Varol Topcu, ”Fetal health status prediction based on maternal clinical history, September 2018.
[15] Jyoti Kiran Gaikwad, Vaishali Taksande, ”Web Base App on Maternal and Neonatal Outcome Among Pregnant Adolescents:,” 2022.
[16] Aguiar, E., Zhang, Y., Blanton, M., 2014. An overview of issues and recent developments in cloud computing and storage
[17] GL Dunietz, KL Strutz, C Holzman, Y Tian, D Todem, BL Bullen, JM Catov, ”Moderately elevated blood pressure during pregnancy and odds of hypertension later in life: ,” 11 January 2017.
[18] Interactive anomaly-based DDoS attack detection method in cloud computing environments using a third party auditor 2023, Journal of Parallel and Distributed Computing
[19] Catalina Valencia, Leona C. Y. Poon, Nikos A. Kametas, Kypros H. Nicolaides, ”Hypertensive Disorders in Pregnancy: Screening by Systolic Diastolic and Mean Arterial Pressure at 11–13 Weeks,” 06 Sep 2010.
[20] Jørn Olsen, Yuelian Sun, ”Prenatal exposure to elevated maternal body temperature and risk of epilepsy in childhood: a population-based pregnancy cohort study,” 07 December 2010.