International Journal of Advanced Development in Science and Technology

Volume : 3 Issue : 03 April 2021 www.ijadst.com e-ISSN: 2582-1059

CREATING SECURE CLOUDS BY CONTINUOUS AUDITING USING TERNARY HASH TREE

[1]J. Karthikeyan, [2]S. Maheswaran, [3]R. Reena , [4] R.K.Kapilavani,

[1,2] Students, [3,4] Faculty,

[1,2,3,4] Prince Shri Venkateshwara Padmavathy Engineering College, Ponmar,

[1] monkkarthi99@gmail.com

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, it is significant to build a secure and efficient data auditing framework for increasing and maintaining cloud users trust with CSP. Researchers suggested introducing Third Party Auditor (TPA) on behalf of cloud user for verifying the outsourced data integrity, which may reduce the computation overhead of cloud users. In this work, we proposed a novel integrity verification framework for securing cloud storage based on Ternary Hash Tree (THT) and Replica based Ternary Hash Tree (R-THT), which will be used by TPA to perform data auditing.

We further extend our framework to support error localization with data correctness, dynamic updates with block update, insert and delete operation in the cloud. The structure of THT and R-THT will reduce the computation cost and provide efficiency in data updates compared to the existing schemes. The security analysis of the proposed public auditing framework indicates the achievement of desired properties and performance has been evaluated with the detailed experiment set. The results show that the proposed secure cloud auditing framework is highly secure and efficient in storage, communication and computation costs.

Keywords – Cloud computing, data storage, public auditing, Ternary hash tree, Replica based Ternary Hash Tree, dynamic updates, error localization, data correctness

1. 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 the cloud services for their data storage, availability, and integrity. Even though cloud computing infrastructures and service providers are much reliable and powerful, a wide range of threats for data integrity exists. Incidents of data loss and outages of significant cloud services appear day to day. On the other hand, 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. Unless the third party auditor (TPA) or the Data Owner (DO) verifies the integrity of the data, the level of data confidentiality of the outsourced data is unknown. The major concern with the cloud data access is to ensure data integrity at untrusted cloud servers. Many researchers [5-23] have proposed various schemes to address the data integrity verification issues in the cloud. In existing schemes, the role of verifier falls either in private data auditing or in public data auditing. In private data auditing, DO takes the responsibility to verify the correctness of the stored data in CSP. In public data auditing, DO rely on trusted third-party auditor for challenging CSP against data integrity verification. However, data physical possession in the cloud is not known to DO. So, both types of auditing are performed without knowing the contents of actual data files. In the cloud, remotely stored data may be accessed to perform read and write operations. In order to perform a write operation, the data need to be available for dynamic access.

2. RELATED WORK

The data integrity verification of the outsourced data in the cloud is verified through auditing techniques. Initially, researchers performed static data verification through the error correcting codes, and then the need for dynamic data aroused. Researchers proposed various data auditing techniques like usage of sentinels, verification through data blocks, and signatures scheme. Recently many schemes have been practiced for data integrity verification. Wang proposed a public auditing mechanism using homomorphic linear authenticator which generates proof through the aggregation of individual data block authenticators. It supports dynamicupdates but causes versioning of the updates and maintaining them but does not contribute towards dynamicupdates performed by the Data Owner as well as CSP. Zhu proposed an interactive proof system (IPS) asa dynamic audit service with the zero-knowledge property for integrity verification. It also supports timely anomaly detection and dynamic data operations with random sampling, fragment structure, and File Allocation Table (FAT). The public auditing for the shared data in the cloud is preserved with user's data privacy through a ring signature scheme proposed by Wang. It supports dynamic data updates operations through index hashtables that avoid re-computation of signatures on blocks but limits public verifiability over the newly inserted ormodified blocks. Liu proposed a scheme of public auditing with verifiable fine-grained updates. It utilizes ranked Merkle hash tree algorithm to compute the verifiable metadata and authenticator values. Here the auditing process is made through a challenge-response protocol, which is verified by FAT. Li proposed ascheme involving offline and online tags that were used to authenticate and verify the integrity of the data blocks.

3. SYSTEM MODEL

Four different entities in the proposed architecture are as follows:

- Client - Data Owner (DO): an entity, who creates and store data in the cloud.
- Client - Data User (DU): an entity, who access data stored by DO.
- Cloud Storage Server (CS): an entity, which is supervised by Cloud Service Provider (CSP) to provide storage and computation.
- Third Party Auditor (TPA): an entity, who have capabilities to assess and interpret the risks in services offered by CS on the behalf of DO upon request.

In Cloud Storage Server (CS), data files in large will be stored by DO. Since the data is not stored locally, clients need to ensure the correctness of the data stored and maintained in the cloud. It is not feasible for DO to verify all stored data periodically with time and resource constraints. So, the task can be entrusted to Third Party Auditor (TPA). TPA is impartial when CS is not trusted. DO will access CS to store, retrieve, or update data files. The data integrity threats on the road to DO data files can be as internal and external attacks in CS. The attacks may be bugs in either software or hardware or network design, hacking, malicious activities, etc... Further, CS may be self-centered. CS may even decide to hide the flaws or loss of data maintenance, in order to maintain a reputation among DO's. To gain trust in the cloud, TPA service can be used to perform frequent auditing. In the proposed model, revealing hosted data to TPA or external parties is in no use, since the data will be in encrypted form and in storage order. CS and TPA even jointly cannot retrieve the actual data stored in the cloud, with the metadata information available. Because the proposed system utilizes different block ordering for storage and auditing. DO can issue TPA's public key to CS for authorizing the audits performed by TPA.

A. SERVER 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.. User willupload file to Cloud. This file is split into blocks using Dynamic Block generation Algorithm. The Blocks arestored 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.

B. DATA UPLOAD AND BLOCK SPLIT

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 upload the data to different cloud by the time it is split 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.

C. Data Integrity Checking and Update

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. The auditing processes have a flow, first the parent block checking. If the parent block have 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.

D. File Recovery

Attacker can corrupt data in any one of the cloud servers. On Data Integrity Checking done by the Verifier, the 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 join in the cloud ++ they will read the certificate and then they can create an account in the cloud.

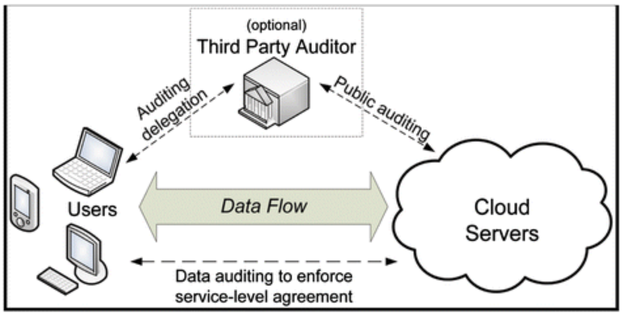
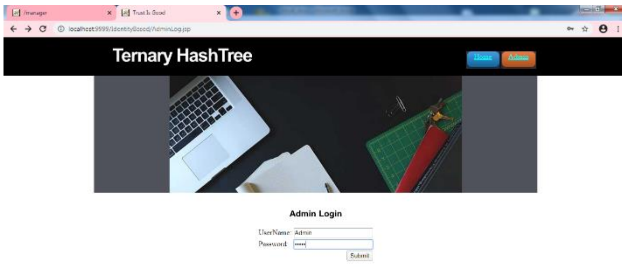


Fig 1: Interconnection of TPA, the user system and Cloud data storage

4. RESULTS

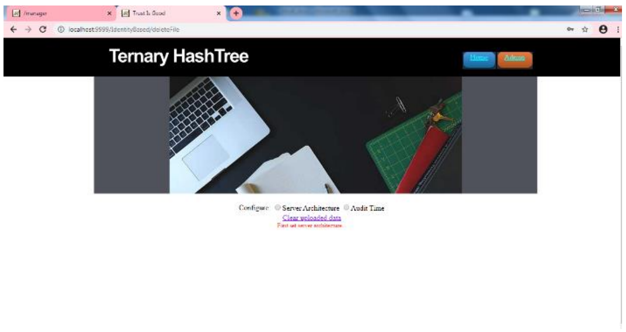
The system model explained as above is implemented in real time and gained immense responses from wide variety of localized businesses. It gained positive feedbacks and took the suggestions from real time users as requirements. The snapshots from the user interfaces are presented below for a clear view.



Admin Login

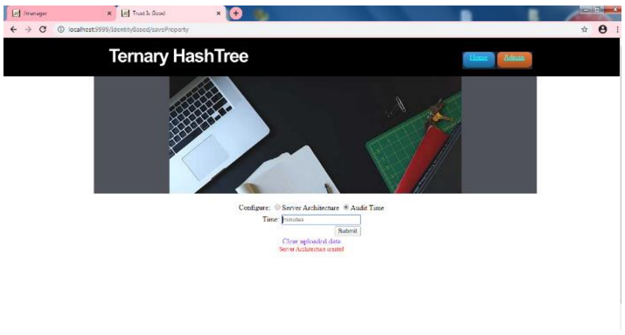
UserName: Admin
Password: •••••
Submit

Fig2:Admin Login



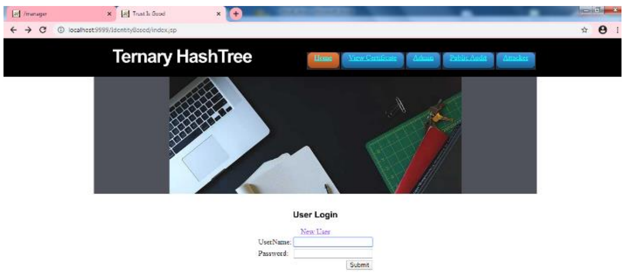
Configure: Server Architecture Audit Time
Clear uploaded data
First set server architecture

Fig 3: Server Setup



Configure: Server Architecture [✓ Audit Time]
Time: minutes
Submit
Clear uploaded data
Server Architecture created

Fig 4: Configure Audit Time



User Login
New User
UserName:
Password:
Submit

Fig 5: User Login

5. 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 isperformed maintaining public verifiability on the same with reduced complexity which was better than the existing schemes.

6. REFERENCES

[1] Kan Yang, Student Member, IEEE, Xiaohua Jia, Senior Member, IEEE, An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing

[2] Mehul A. Shah, Mary Baker, Jeffrey C. Mogul, Ram Swaminathan (HP Labs, Palo Alto), Auditing to Keep Online Storage Services Honest

[3] Yan Zhu1,2, Huaixi Wang3, Zexing Hu1, Gail-Joon Ahn4, Hongxin Hu4, Stephen S. Yau4, Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds

[4] Qian Wang, Student Member, IEEE, Cong Wang, Student Member, IEEE, Kui Ren, Member, IEEE,Wenjing Lou, Senior Member, IEEE, and Jin Li, Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing

[5] Cong Wang, Student Member, IEEE, Qian Wang, Student Member, IEEE, Kui Ren, Member, IEEE, Ning Cao, Student Member, IEEE, and Wenjing Lou, Senior Member, IEEE, Towards Secure and Dependable Storage Services in Cloud Computing

[6] Boyang Wang, Baochun Li, Member, IEEE, and Hui Li, Member, IEEE, Oruta: Privacy-

Preserving Public Auditing for Shared Data in the Cloud

[7] Mehul A. Shah, Mary Baker, Jeffrey C. Mogul, Ram Swaminathan (HP Labs, Palo Alto), Privacy-Preserving Audit and Extraction of Digital Contents

[8] Cong Wang, Student Member, IEEE, Sherman S.-M. Chow, Qian Wang, Student Member, IEEE, Kui Ren, Member, IEEE, and Wenjing Lou, Member, IEEE, Privacy-Preserving Public Auditing for Secure Cloud Storage