Research Article

LSSA: A Protective Shared Data Communication Mechanism in Cloud Environment

N. PRAVEENA, MADHAVI LATHA PANDALA, MADHAVI KATAMANE

[1,2,3]Assistant Professor, Dept.of Information Technology, VR Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India.

Email: praveena.4u@gmail.com, chinnu065@gmail.com, itsmahavi12@gmail.com

Article History: Received: 11 January 2021; Accepted: 27 February 2021; Published online: 5 April 2021

Abstract - Users can access shared data warehouses using a cloud infrastructure. It is important to verify the data successfully to ensure mutual data integrity. The accuracy checking of the shared data is carried out by an examination system that encourages Group members to alter data, but this method leads to complicated estimates for the Group members. The monitoring method of the assigned agent estimates the group members lightly, but lacks the safety threats between the group members and their agents. With the implementation of Hash graph technology and the development of a management Third Party Medium (TPM) approach, a Lightweight Safe Cloud Storage Audition System (LSSA) is suggested, achieving group security protection and a lightweight group measurement. In the meantime, the TCP Sliding Fan Technology incorporates a simulated TPM pool with interconnected features to enhance support for the handler. We test our method in numerical analysis and tests, which prove that our system provides the groups with lightweight computing and ensures the safety data evaluation process.

Keywords: Shared data, virtual TPM pool, lightweight calculation, agent security

I. Introduction

Increment of customers' access to a shared pool of networks, software and infrastructures without even needing to demand them from distributed computing is the newest utility-oriented decentralised computing paradigm which has envisaged a massive IT transformation. Cloud computing is divided into three frameworks in the sense of deployment: I public, ii) private, iii) hybrid, (iv) community clouds that are described below:

Public Cloud: Through public cloud computing providers move different applications as a service and enable consumers via access to infrastructure, such as Amazon Web Services, Google App Engine, by concentrating distributed servers over the Internet.
Private Cloud: A success organization requires and manages programmes and structures entirely.
Community Cloud: A collection of organizations that are either supervised personally or by a trustworthy external entity distribute the resources and structure.
Hybrid Cloud: The Hybrid cloud pursues a combination of on-site, proprietary cloud and public cloud third-party providers, in a two-platform structure.

As for the reference architecture and taxonomy of three service models , i.e. PaaS, SaaS, IaaS, Liu and his colleagues[1] addressed the obstacles to select and improve distributed computing and classes of utility computing and explore their opportunities[2] for selecting and developing them. Buyya and his colleagues[3] suggested a market-oriented cloud asset management system. It offers cluster, grid and cloud features and knowledge of processes for market-driven asset management.

The PaaS system provides designers with runtime requirements, as their individual needs suggest. The PaaS offers the development, delivery and monitoring of the applications through programming framework, libraries and toolboxes. For trading clients, such as S3 (Simple Storage Service) and EC2 (Elastic Cloud Computing), the IaaS provides tracking, repositioning and systems management in a kind of scalable Virtual Machine (VM). Distributed storage offers a cloud storage as a service for the monitoring, monitoring, and remote backup of information that is available via a network to users (usually the Internet). The customer is concerned that the information contained inside the cloud is integral so unauthorized actors can target or change customer information. Therefore, in Cloud Computing a new principle called data auditing for the safe storage of information is implemented. The audit is a consumer information authentication procedure that may either be performed by the consumer itself (informational proprietor) or by a TPA (third party auditor). It helps hold the data stored in the cloud integrity.

The two sections of the role of the verifier are: firstly, private auditing, where the honesty of the data is only reviewed by the recipient or the information holders. No other party has the power to ask the server about the results. However, the average consumer check continues to improve. Secondly, public auditability encourages everyone to question the server, not just the client, and provides a review of the records by TPA. The TPA is a business that is used to work by the consumer. This provides all the required skills, intellect and experience required for the job of certifying honesty which thus reduces customers' overhead. The distributed database information without requiring a local copy of information must be effectively checked by TPA. The details contained on the distributed server should be known zero.

II. Existing Works

Ateniese et al. first suggested a Provable Custody of Data (PDP) in 2007 that would be able to validate cloud data ownership without all data being retrieved[5]. Then Juels et al . suggested to use the retrievability evidence framework to provide evidence that data can be recovered by the verifier[6] by a back-up or archive facilities [6].

The PDP framework that supports complex operations [7] was introduced by Ateniese et al. in a follow-up report. This ensures that a data up loader has complete control of any operation carried out in a cloud application, including block deletions, modifications and insertions. The authenticated table [8] was then introduced by Waters et al. to introduce a fully-dynamic PDP framework. In comparison to these works,[9] [14] is used to analyze the credibility of common data in the following structures. Users can modify and share data with the cloud providers as a collective in this case, where any member of the group can view and alter the shared data, and also share the variant they have updated with the others [11]. In the same way, user can modify and share information.

A BLS based signature scheme to support agile group management was introduced in 2016 by Mr Yang et al.[9]. In addition to the collusion attacks of the Cloud service provider and community participant, Jiang et al. have suggested data confidentiality based on the vector committing methodology [10]. Through integrating proxy encryption with encryption.

Luo et al . introduced a secure consumer revocation system in 2017[11]. Since then, Huang et al. has introduced an effective, logical hierarchy-bound key distribution among groups, thus maintaining the group's identity privacy[12]. Huang et al. subsequently suggested the removal of the main scrow to provide certificateless audits would further enhance the privacy of the user[13]. The groundbreaking studies preceded by Huang et al. Fu et al . proposed to carry out an audit method to restore accurate common blocks of data by modifying the group's binary tree tracking data[14].

Li et al. suggested a new cloud storage audit scheme with a server for cloud audit and cloud management [15]. Until uploading to the cloud storage system, the cloud audit service establishes authentication labels for customers. While this scheme can minimize user overhead computation, it will expose to the cloud audit system entirely the private keys and user details. Malicious cloud service providers will then go through the authentication process without storing customer data.

Guan et al. was using an analogous contradictory approach to creating a cloud storage audit scheme [16], which minimized the time taken for verification labels but expanded the time needed to validate cloud data integrity. Wang et al. incorporated agents to help community members in creating labels of authentication and auditing data integrity [17], thus reducing the strain on group members of computing.

Nevertheless, the community member must encrypt the data before submitting to the server, which ultimately raises the device workload, in order to guarantee data protection. Shen et al . suggested a minor audit scheme to replace members of the community with authentication labels by adding the Third party mechanism (called the agent) [18].

III. Problem Definition

A malicious cloud server is capable of discarding all data exchanged by reserving any intermediate outcomes or previous legitimate facts that we call a substitution attack or a re-playing attack, which may provide clear proof of data ownership. A malevolent group member may alter data of other members without being detected in that group. A malicious agent may work with unauthorised members of the community to harvest data from users and identities. The three above things we know continue to be open problems for the creation of a stable integrity audit scheme on customer side for common data with lightweight computing.

IV. Implementation Procedure

A lightweight, reliable cloud storage data auditing system (LSSA) was introduced here. Similar to the audit system in the cloud computing system[18], a third party medium (TPM) is used in the verification mark measurement instead of group members and the effects of audit data accuracy are easy for group members to measure. In relation to this schema, we divided the group members and the TPM into a group manager, to divide and rule the group members and the TPM and to remove their cooperation. With respect to the participants of the party. Our contributions to science can be summarized as follows:

(1) This paper offers the data protection and anonymity of the community participants by the use of an effective blind process. This paper removes the secret protection threats of community members by implementing a hash graph and at the same time makes user identification traceable.

(2) The concept for the TPM management was created and the Project Manager designed the interactive TPM pool. The technique guarantees agent protection (TPM) and contributes to light

estimates for the agent. The usage of the TPM to measure the mark of authentication and to audit the data integrity ensures that community members are able to determine lightweight.

(3) The scheme's security review suggests that the scheme is secure and can survive threats and replay attacks.

(4) The experimental review of the Scheme indicates that lightweight measurements can be done for community members and the TPM.

V. Model for Implementation

The model of the Structure contains four separate entities: the participants of the Group (M), the cloud, the Group Manager (GM), and the TPM. As shown in Figure 1, a group is comprised of several groups. After the data owner (the person or entity who controls the original data) generates and uploads the data file to the cloud, every member of the community may view it and change it. Notice that the original owner of the data will play the position of GM and in each category there is only one GM. The play two important roles: 1) blind data and 2) blind data collected and distributed via a Hashgraph within the community. The cloud offers data store services for group members (iCloud, OneDrive and Baidu Cloud) and offers group members with a forum to exchange data. The GM is playing three critical role models: 1) creating public-private TPM key pairs, 2) multi-purpose the TPM management approach and 3) generating a hidden seed for blinding community members' data and retrieving real cloud information. The TPM has two significant roles: 1) the development of a data 2) authentication symbol for the members of the Community, on behalf of the participants, the checking of cloud data 's credibility.

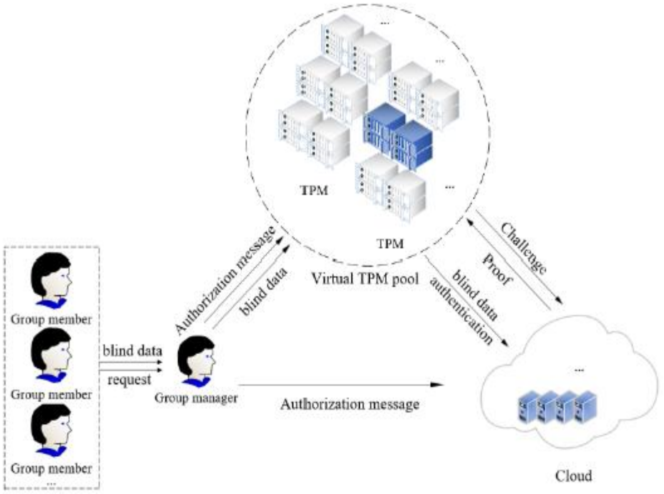

Fig1: Architecture model for implementation

The method of implementation is broken down into the data transfer and the audit phase. The data would then be blinded by the hidden seed and saved to the project manager before the group member asks to download the updated data to the server. Under TPM administration, the group management manager picked a TPM for authorization from the simulated TPM pool, and for those blinded data during the time of authorization the approved TPM calculates the corresponding authentication labels. The cloud will also obtain the blind data and authentication mark. The cloud will verify if the TPM permission is active at the present time before obtaining these messages. If it is, the verification mark verifies that it is right or not. If right, he will retrieve the actual data and determine their labels of authentication. Finally, these actual labels and verification data are stored in the cloud. The community manager picks a TPM and establishes the authorisation under the TPM compliance plan before conducting the auditing process. The approved TPM then sends challenge communications to the cloud. The cloud will verify if TPM permissions are correct before these messages are received. If so, the cloud may provide confirmation that the shared data was in custody. Finally, the TPM will verify the correctness of the facts by verifying the credibility of shared data in the cloud.

a. Design of the Implementation

Lightweight computing: This strategy means that community participants don't have to carry out time-consuming assessments during authentication labels or during mutual data assessments. In the estimation, multiple TPMs take part, which means only one TPM is measured gently.

Identity traceability: Moving data from unauthorized users may lead to conflicts between members of the community using the same shared data. This aim means that all unauthorized members of the company can be identified and expelled by GM and thus achieves company security management.

TPM management security: Each of the TPM operates independently to ensure the TPM 's legal presence. This aim means that the cloud embraces and preserves only data from GM-licensed TPMs and addresses the problems of the GM-authorized TPMs only.

Data privacy and identity privacy: Instead of community members, it is not possible to know the exact data block information when the TPM produces authentication marks. At the point of downloading data and auditing data, the TPM cannot access the identity information of community members.

Audit correctness and security: Through the use of the audit process the TPM will check the quality of the shared data. The audit cannot be done by replacing or replaying attacks by malicious cloud service providers.

b. HashGraph Approach

As seen in Figure 2 below, every circle of the figure represents a hah-value case. In the historical records the earlier vertices represent early occurrences and Mi represents the user i. In Gossip mode, the message is distributed over the Hashgraph network. When B happens, user M2 who created B will add its own signature, Sign M2, to this event and randomly transfer it to user M1. This message is received by User M1 and a new event A is generated. Event A comprises two event hashes (a historical event C and a coordinated event B user M2), and user M1 adds event A to his / her signature, Sign M1.

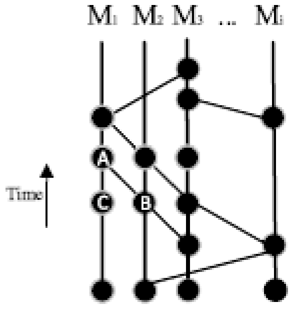

Figure 2: Hashgraph

Design of LSSA

The concept theory of community member management can be suggested using Hash graph technologies. The definition of the TPM management approach is indicated by comparison to the TCP sliding window and the interconnection feature.

Design of Group Member Management

The group manager creates an account arbitrarily as the identity mark of the member M of the group, whenever the account is used to determine the real identity of the group member. The project manager deletes the account if the group member is dropped. We specified the following notes for notational convenience.

M: les data are divided into n blocks (m1, m2. .mn), where mi is divided into slices (mi;1,mi;2. .mi;s) each is repeated as mi and each slice is recorded as mij.
mij: The mij-corresponding blind data block.
idi;j: the blind data block public identification information mij.
MOwner: Data owner's account (ID).
Block: Block, e.g. SHA-1, and SHA-256. Hash function
Sign: The name of the participants is marked.

The MOwner Data Owner sends the blind data block mij to the team manager who measures the hash(idi;j) value of idi;j as the transaction record (called the original transaction record) and adds the SignMOwner signature. To synchronise this with the original event, the group member or group manager is chosen arbitrarily, sending the event to the network nodes. The Group members have access to the original shared data and may change it, but the Mi Group members who have subsequently updated and modified mij must update their blind block identifier. Thus, the members calculate the hash value of *idij* as a modify/access record (called a transaction record) for a new event and attach the signature *SignMi* to spread it within the group.

The creative director produces the public-private key pair for TPM during the data upload process. He also produces a hidden seed and transfers it to members of the party and the server. As the group manager's port is the link point between the members of the group and the TPM, the group manager has the power to choose the sending and contact roles, to establish a TPM management plan authorization and then to give the authorization to TPM. Whenever the user decides to load data into the cloud the blindness element is first calculated to blind this data using the hidden seed, the blind data is then determined as a transaction record for a new occurrence and then distributed to the project manager by distributing them within the project. The community manager may validate before getting the messages whether the member's hash value is true or not. If so, the approval will be submitted to the TPM.

The TPM will then create the corresponding authentication labels for the blinded data and save them together in the cloud. The cloud tests if the TPM permission is current at the present time before retrieving those messages. If so, he can verify whether these marks are right or not. If right, the actual data is retrieved by using the element blindness and their authentication labels are computed. The cloud eventually holds the individual data and codes for authentication.

Experiment Evaluation

The main theme of this paper is to eliminate future safety risks using a better route. Team participants are most concerned with the issue of productivity when using data in the audit scheme with shared data. In this portion, the device overhead of the LSSA scheme is first measured, and then tested in the real operating environment. The final results show that the scheme will have limited weight for the members of the party and that LSSA is secure from related audit schemes.

VI. Conclusions

The proposal suggested an established pooled data ownership in cloud storage for a lightweight and secure audit process. By implementing a Hash graph the group membership will track, and Hash graph technology will avoid the illicit actions of the group members. Every community member and TPM, defining several TPMs for the measurement and management under the management approach, is different, which means that the mechanism of cloud data testing is safe and that the TPM is measured lightweight.

References:

[1]. C. Yang and J. Ye, “Secure and efficient fine-grained data access control scheme in cloud computing”, Journal of High Speed Networks, Vol.21, No.4, pp.259–271, 2015.

[2]. X. Chen, J. Li, J. Ma, et al., “New algorithms for secure outsourcing of modular exponentiations”, IEEE Transactions on Parallel and Distributed Systems, Vol.25, No.9, pp.2386–2396, 2014.

[3]. P. Li, J. Li, Z. Huang, et al., “Privacy-preserving outsourced classification in cloud computing”, Cluster Computing, Vol.21, No.1, pp.277–286, 2018.

[4]. B. Varghese and R. Buyya, “Next generation cloud computing: New trends and research directions”, Future Generation Computer Systems, Vol.79, pp.849–861, 2018.

[5]. W. Shen, J. Qin, J. Yu, et al., “Enabling identity-based integrity auditing and data sharing with sensitive information hiding for secure cloud storage”, IEEE Transactions on Information Forensics and Security, Vol.14, No.2, pp.331–346, 2019.

[6]. R. Kaur, I. Chana and J. Bhattacharya J, “Data deduplication techniques for efficient cloud storage management: A systematic review”, The Journal of Supercomputing, Vol.74, No.5, pp.2035–2085, 2018.

[7]. Cisco, “Cisco global cloud index: Forecast and methodology, 2014–2019”, available at: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/ white-paper-c11-738085.pdf, 2019-5-5.

[8]. Cloudsfer, “Migrate & backup your files from any cloud to any cloud”, available at: https://www.cloudsfer.com/, 2019-5-5.

[9]. Y. Liu, S. Xiao, H. Wang, et al., “New provable data transfer from provable data possession and deletion for secure cloud storage”, International Journal of Distributed Sensor Networks, Vol.15, No.4, pp.1–12, 2019.

[10]. Y. Wang, X. Tao, J. Ni, et al., “Data integrity checking with reliable data transfer for secure cloud storage”, International Journal of Web and Grid Services, Vol.14, No.1, pp.106–121, 2018.