Blockchain-Based Decentralized Cloud Storage

N.M.K. Ramalingamsakthivelan[1], B. Ramya[2]

[1]Associate Professor, [2]ME-II [ND] Year

Department of Computer Science and Engineering, Paavai Engineering College, Pachal, Namakkal.

Abstract — Public auditing schemes for cloud storage systems have been extensively explored with the growing importance of data integrity. A third-party auditor (TPA) is introduced in public auditing schemes to verify the integrity of outsourced data on behalf of users. In order to resist malicious TPAs, many blockchain-based public authentication schemes have been proposed. However, existing audit schemes rely on a centralized TPA and are vulnerable to greedy auditors who can collude with malicious blockchain miners to produce biased audit results. In this paper, we propose a blockchain-based Decentralized Public Audit (BDPA) scheme using a decentralized blockchain network to assume the responsibility of a centralized TPA as well as mitigate the influence of greedy auditors and malicious blockchain miners by leveraging the concept of a decentralized autonomous organization (DAO). A detailed security analysis shows that BDPA can maintain data integrity against greedy auditors and malicious blockchain miners. A comprehensive performance evaluation shows that BDPA is feasible and scalable.

Index Terms— Cloud storage, public integrity auditing, identity-based cryptography, blockchain.

1. INTRODUCTION

Cloud storage has attracted extensive attention from both academic and industrial research communities for its enormous advantages in cost, performance, and management [1]–[3]. Cloud users can reduce expenses for software, hardware and services by storing data on public cloud servers. And in the meantime, they can access outsourced data efficiently and remotely over the Internet without having to stay near their computers. Nowadays, more and more users choose to migrate their local data to cloud storage managed by professional cloud service providers (CSPs), such as Amazon cloud and Google cloud [4], [5]. Although cloud storage brings many benefits to cloud users, data outsourcing has also created many critical security issues [6]. One of the most important security issues is data integrity.

In fact, outsourced data can be corrupted because cloud servers can suffer from external competitive attacks and internal hardware or software failures [7], [8]. In addition, the cloud server is an independent and untrusted administrative unit and may delete some cloud data that users have never accessed to save storage space or hide data loss events to maintain its reputation [9], [10]. Unfortunately, most cloud users often delete locally stored backup data after uploading data to the cloud server. Due to the above factors, it is important that users regularly check the integrity of their outsourced data. Various cloud storage auditing techniques are popularly used to ensure the integrity of outsourced data [11]–[16]. In traditional public audit schemes, users often authorize a TPA to periodically publicly audit their data obtained from external sources [17], [18]. A trusted TPA can also provide users with reliable audit results and reduce users' communication and computational burden [19]. However, in traditional public audit schemes, the TPA is assumed to be honest and reliable, which is actually a strong assumption because it is quite possible for the auditor to be corrupt .

For example, the responsible auditor can always produce a valid audit report without performing the verification process to save computation costs [21]. To thwart malicious auditors, many blockchain-based public auditing schemes have been proposed [20]–[23]. In most existing blockchain-based schemes, blockchain technology [24]–[26] is used as a secure source of time-dependent pseudo-randomness and undeniable proofs of storage. Specifically, auditors typically extract block values such as Nonce and Block Hash from the blockchain to generate random challenge messages that contain the indexes of selected data blocks. They also generate log files for audit procedures and store hash values of log entries in the blockchain. However, most existing blockchain-based schemes suffer from tempting auditors and have a centralization problem.

2 RELATED WORKS

2.1 Traditional public auditing

To ensure the integrity of outsourced data, Juels et al. [28] first proposed a “proofs of retrievability” (POR) scheme that relies on indistinguishable blocks hidden between file blocks to act as a sentinel to detect cloud data corruption. Then Ateniese et al. [9] proposed a "provable data ownership" (PDP) technique, a variant of POR that can support an unlimited number of challenges. Considering the large volumes of data and the limitations of communication resources, cloud users usually adopt public auditing schemes to help them regularly check the integrity of their outsourced data. In 2013, Shacham et al. [29] proposed a compact POR scheme that uses homomorphic authenticators to obtain compact proofs and also support public authentication. Following this work, many public audit schemes have been proposed [6], [30]. However, these public audit protocols are mainly based on public key infrastructure (PKI) systems, which are inefficient and cumbersome. To avoid user certificate management, identity-based public auditing schemes have been proposed [31], [32]. In these schemes, a publicly known string representing an individual or organization is used as the public key. A trusted third party, also called a private key generator (PKG), generates private keys for all cloud users [33]. However, identity-based schemes have a key escrow problem and are vulnerable to a malicious PKG that checks the private key of all users [34]. Therefore, certificate-free public audit schemes have been proposed [21], [23], where the key generation process is split between the PKG and the user.

2.2 Blockchain-based public auditing

Recently, investigating the credibility of TPA has attracted the attention of researchers. In most traditional schemes, the auditor is assumed to be honest and reliable. This is a strong assumption because auditor corruption can occur in practice [20]. With the popularity of blockchain technology, many blockchain-based schemes have been designed to resist malicious auditors [20]–[23]. In 2014, Armknecht et al. [22] proposed a blockchain-based auditing scheme that uses random values in Bitcoin [24]. Then Zhang et al. [23] also proposed a certificate-free public audit scheme in cyber-physical-social systems. In 2019, Xue et al. [20] proposed an identity-based public auditing scheme that used Bitcoin as a random source and stored the hash values of the protocol in transactions on the Bitcoin blockchain. Later, Zhang et al. [21] also proposed a blockchain-based certificateless auditing scheme against procrastinating auditors using the Ethereum platform and provided a security analysis for malicious blockchain miners. Recently, Zhang et al. [35] proposed a conditional identity privacy-preserving public audit mechanism for cloud-based WBANs and integrated the Ethereum blockchain into this scheme.

3. EXISTING SYSTEM

Every existing system uses rewards to encourage peers to participate honestly in the deposit process. We assume that leaders act rationally and broadcast new blocks as soon as they are ready. This is a common behavior found in most existing blockchains. On the other hand, in the enabled setting, nodes must be authorized and sk allows them to authenticate in the system. Exposing the sk would allow an attacker to act maliciously on its behalf and would violate any existing contractual arrangements. Naturally, as already described, Audita can be implemented by modifying the protocol of an existing blockchain. The emergence of blockchain technology provides a new research idea to solve the problem of mutual trust. It uses cryptography rather than a centralized architecture to build trust in partners to protect their interactions.

If it is not equal to the hash value stored in the body of the block, the file is considered corrupted. This shows that as file sizes grow, the impact of network I/O on the system is greater than waiting to wrap into a block. Suppose a malicious user changes the data of the previous block, it will inevitably cause the hash of that block to change, which will further lead to an inconsistency between the hash of the previous block and the prehash of the current block.

4. PROPOSED SYSTEM

In this paper, we proposed a public audit scheme using blockchain technology to combat malicious auditors. The proposed plan introduced above also has the same problem. Customers need access to full data backups. However, as mentioned above, it is clearly not suitable in practice. In practical applications, the task of integrity checking is performed by the TPA, and most of the later proposed schemes support public auditing. Through the above analysis, we find that the proposed scheme has the following drawbacks: The security of these schemes relies on a trusted third party, TPA.

We created a prototype on the Ethereum platform that uses Aliyun as a data storage service and tested the performance of uploading and downloading files of different sizes. Blockchain limits the capacity of blocks, so only very sensitive security information is stored in blocks. Otherwise, the system performance will be unacceptable.

However, analyzing user operations from a large number of system logs is inefficient. The chain metadata information can be used later to verify the authenticity of the data. Chain operation records can be used later to track file access. Metadata information can be used later for integrity auditing. Auditing operations can be done by analyzing the operations log.

5. SYSTEM ARCHITECTURE

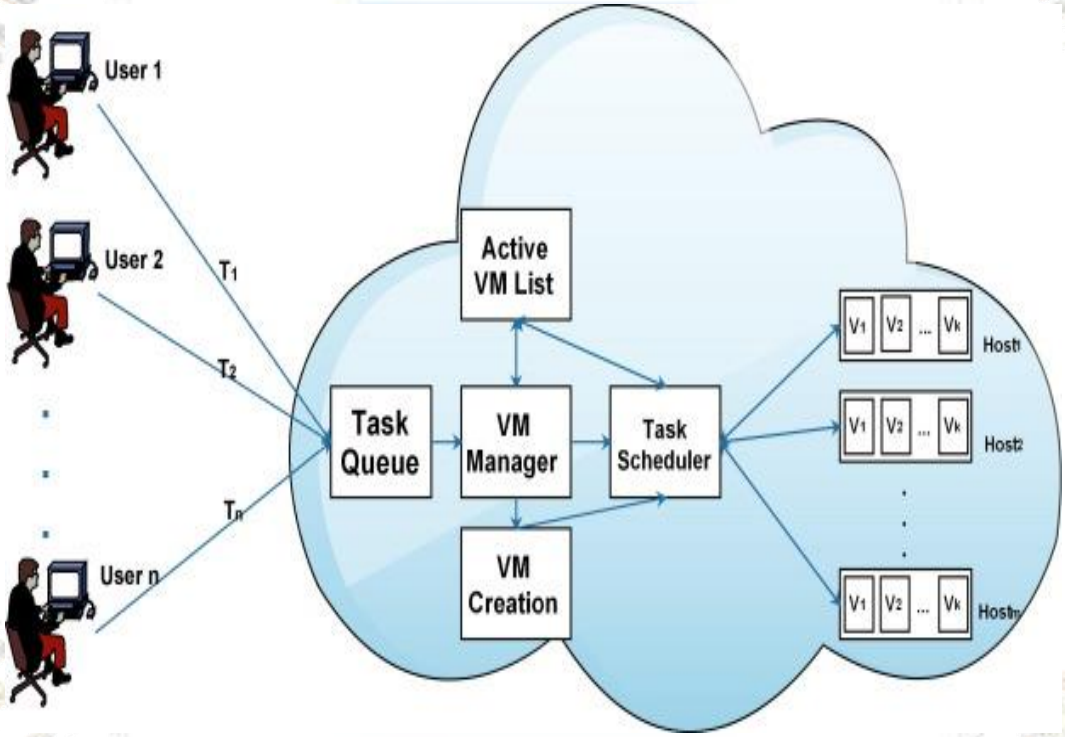


Fig 3. Blockchain-Based Decentralized Cloud System

5.1 TESTING

Implementation is the stage of the project when the theoretical design is turned into a working system. This is the final and important phase in the system life cycle It is actually the process of converting the new system into a operational one.

5.1.1 Unit Testing

Unit testing comprises the set of tests performed by an individual programmer prior to integration of the unit into a larger system. The module interface is tested to ensure that information properly flows into and out of the program unit. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary

conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing. All independent paths through the control structure are tested. All error-handling paths are tested.

5.1.2 Block Box Testing

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. It is sometimes referred to as specification-based testing.

5.2 SYSTEM IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned into a working system. This is the final and important phase in the system life cycle It is actually the process of converting the new system into a operational one.

6. RESULT AND DISCUSSION

The performance evaluation of the proposed framework starts with investigation of accuracy of online-prediction system. Its effectiveness can be seen where predicted CPU usage have almost overlapped actual resource usage for GCD, PL and BB workloads. The error score of proposed prediction approach for prediction interval of 5 minutes on three workloads.

RMSE for different workloads

Interval	GCD (CPU)	GCD (Memory)	PL (CPU)	BB (CPU)
5 min	0.0014	0.0035	0.0005	0.0031

The reason behind such an accuracy is that proposed neural network based predictor periodically learns and retrains itself according to changes in live and historical workload. In addition, the application of AADE algorithm works on N number of solutions which explores the search space in multiple directions and allows efficient learning of patterns and correlations from live data which is responsible for near accurate prediction of resources.

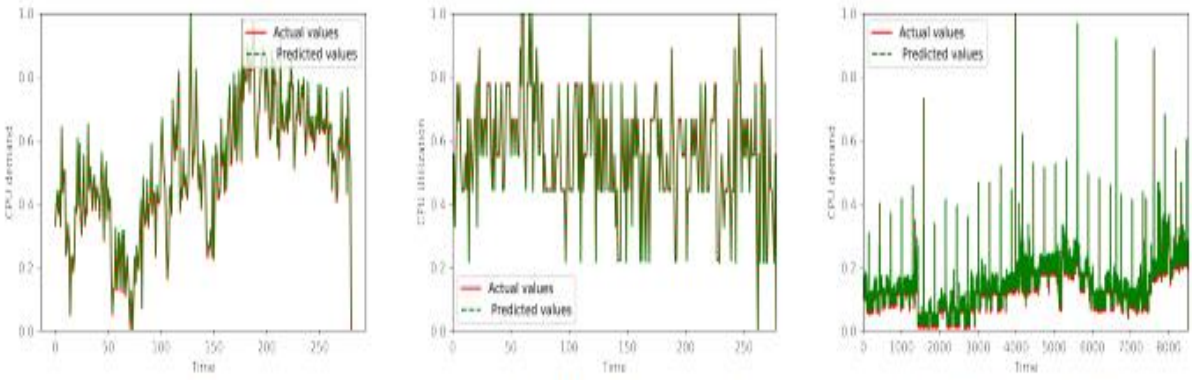
To analyze the performance of multi-objective load balancing, numerous experiments were conducted based on different combination of VMs and servers. The experiments are executed with the ratio of number of VMs and servers as 1:1.

It is mentioned that resource utilization (RU) percentages per VM extracted from the three workloads are used to compute the actual resource usage of VMs under simulation. For instance, if real workload trace shows 67.3% of CPU usage and VM has 500 MIPS of CPU capacity then actual CPU usage of the VM is given by the product of 67.3% and 500.

The number of users are not mentioned in the original datasets, therefore, we created random set of users, who requested different number and type of VMs to compute communication cost (Com) based on the location of inter-dependent VMs. The number of users are taken as 60% of size of the data center, where each user can hold VMs in the range between 0 and 5 with a constraint that at any instance, the total number of VM requests must not exceed total number of VMs of data center.

Each experiment was executed for 12-15 times and a mean of the obtained results are reported. The values of minimum and maximum threshold of CPU usage were 10% and 89% respectively for these experiments. The resource utilization for each workload is more than 63%, which varies between 63-64.8% for GCD, 68-69.8% for PL and 63-65.6% for BB workloads. The power consumption (PW) has increased with respect to the size of data center. The power consumption shows various trends for the three different datasets, depending upon the number of busy and idle servers.

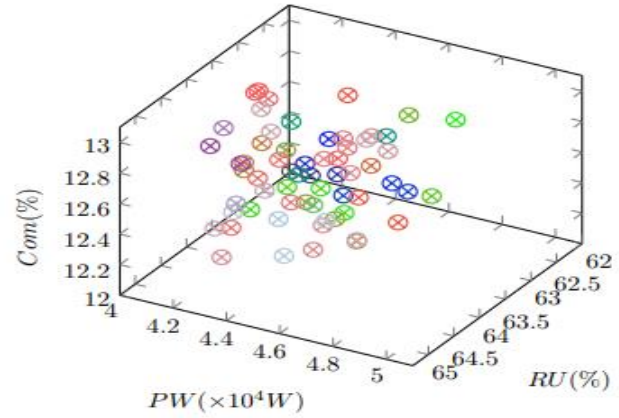
Since the communication cost depends upon the placement of inter-dependent VMs of active users, each workload shows similar values in the range 12-16%. However, power consumption and VM migration costs are increasing with respect to the size of the data center. The values for SLA compliance are varying according to the availability/non-availability of servers.



(a) Google Cluster CPU (duration=24 Hrs) (b) PlanetLab VM CPU (duration=24 Hrs) (c) Bitbrains CPU (duration=30 days)

Predicted vs actual workload

The pareto-front or non-dominated solutions for 1400 VMs placement is that depicts the contradictory behavior of three optimization variables viz. maximization of RU, minimization of PW and Com. The predicted and unpredicted overloads for all three datasets. It is noted that there is an increase in the number of correctly predicted overloads on the respective servers with increase in size of data center for all the three workloads. However, the number of unpredicted overloads are either lesser or equal to 0.07% independent of size of the data center for each experiment of every dataset. This is due to the efficiency of prediction system that accurately forecasts the future resource requirement. The overload prediction accuracy is around 99.94% for GCD during the period of 24 hours. Multi-objective Pareto Front for 1400 VMs with GCD correctly forecasted for PL and BB respectively.



Multi-objective Pareto Front for 1400 VMs with GCD

As a result, the number of VM migrations and SLA violations get reduced during actual VMs allocation which can be observed. Since SLA violations depends upon the availability of the server, they are indirectly varying with respect to the number of unpredicted overloads for different size of the data center. The number of unpredicted overloads prompt VM migration and unavailability of server and hence account for SLA violation.

7. CONCLUSION

This paper builds a blockchain-based behavioral audit framework that uses blockchain to store files' metadata information and users' behavior information. The framework implements operations such as auditing the integrity of files and auditing users' behaviors. Compared to the traditional logging-based audit method, the security of the audited data is guaranteed. Although proxy nodes are used to speed up operations querying on blocks, due to the packing delay problem in blockchain systems, file records may be packed into blocks for a long time, resulting in long waiting times. User to ensure that the operation is recorded in the log. Meanwhile, it takes a long time to wait for files to be packed into a block when they are stored, causing users to successfully upload files but not immediately query their own files. Through testing we found that when the file size is increased, the total time to pack records into a block gradually decreases.

8. FUTURE WORK

In terms of future work, we will investigate further How to select nodes in blockchain to verify proof information. Additionally, we will explore how to integrate some Existing blockchain-based bilinear outsourcing schemes for Handle calculation of TPA. In addition, we will design an efficient public auditing scheme and how it also checks Centralized PKG removal in our future work.

9. REFERENCES

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," Futu. Gen. Comput. Syst., vol. 25, no. 6, pp. 599–616, 2009.

[2] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: format+ schema," Google Inc., White Paper, pp. 1–14, 2011.

[3] E. K. Lee, H. Viswanathan, and D. Pompili, "Proactive thermal-aware resource management in virtualized hpc cloud datacenters," IEEE Trans. on Cloud Comput., vol. 5, no. 2, pp. 234–248, 2015.

[4] M. Dabbagh, B. Hamdaoui, M. Guizani, and A. Rayes, "An energyefficient vm prediction and migration framework for overcommitted clouds," IEEE Trans. on Cloud Comput., no. 4, pp. 955–966, 2018.

[5] C. Systems., "Forecast and methodology,," white paper,, 2016–2021,.

[6] C. Mastroianni, M. Meo, and G. Papuzzo, "Probabilistic consolidation of virtual machines in self-organizing cloud data centers," IEEE Trans. on Cloud Comput., vol. 1, no. 2, pp. 215–228, 2013.

[7] B. Jennings and R. Stadler, "Resource management in clouds: Survey and research challenges," J. of Netw. and Systems Mgmt., vol. 23, no. 3, pp. 567–619, 2015.

[8] D. Agarwal, S. Jain et al., "Efficient optimal algorithm of task scheduling in cloud computing environment," arXiv preprint arXiv:1404.2076, 2014.

[9] D. Saxena, R. Chauhan, and R. Kait, "Dynamic fair priority optimization task scheduling algorithm in cloud computing: concepts and implementations," Int'l J. of Comput. Netw. and Info. Security, vol. 8, no. 2, p. 41, 2016.

[10] K. Karthikeyan, R. Sunder, K. Shankar, S. Lakshmanaprabu, V. Vijayakumar, M. Elhoseny, and G. Manogaran, "Energy consumption analysis of virtual machine migration in cloud using hybrid swarm optimization (abc–ba)," J. of Supercomp., pp. 1–17, 2018.