Journal of Harbin Engineering University
ISSN: 1006-7043
Vol 44 No. 7
July 2023

An Efficient Public Data Integrity for Big Data Processing System In Multiple Cloud Storage

Kanigiri Suresh[1] , Dr. Manoj Eknath Patil[2]

Research Scholar
Department of Computer Science & Engineering, Dr. A.P.J. Abdul Kalam University, Indore (M.P.) – 452010
Corresponding Author : - Kanigiri Sures

ABSTRACT: Big data processing is progressively becoming essential for everyone to extract the meaningful information from their large volume of data irrespective of types of users and their application areas. Big data processing is a broad term and includes several operations such as the storage, cleaning, organization, modelling, analysis and presentation of data at a scale and efficiency. For ordinary users, the significant challenges are the requirement of the powerful data processing system and its provisioning, installation of complex big data analytics and difficulty in their usage. However, this form of storage introduces new security challenges, such as unreliable service providers. Data storage correctness is another challenge that should be addressed before this modern storage model can be extensively applied. Hence, a new scheme is introduced for securing data integrity via a multiple third party auditors based mutual authentication to overcome the aforementioned limitations and ensure high-level security. This system is an inexpensive and user-friendly framework for everyone who has the knowledge of basic IT skills. The performance of this analysis is calculated on different aspects such as Accuracy, Precision, and security. Though the outcomes of various methods are different, efficient public data integrity for big data processing system in multiple cloud storage will be the best results in this analysis**.**

KEYWORDS: Docker Swarm, Big Data Processing System, Cloud computing, Data integrity

I. INTRODUCTION

Big data is high-volume, high-velocity and/or high-variety information assets that cannot be handled and processed by using the traditional IT infrastructure and tools [8]. Earlier, it was the requirement of major businesses and organisations but due to the rapid growth of data, ordinary users are looking to use big data processing options for their large volume of data, which cannot be processed by using traditional IT infrastructure [2]. For ordinary users, the significant challenges are the requirement of the powerful data processing system and its provisioning, installation of complex big data analytics and difficulty in their usage. Therefore, they require an economical, user-friendly, easy to design and develop data processing system.

Cloud-based big data processing systems are the most efficient and established infrastructure to fulfil the big data analysis requirements. Now, most businesses and users are shifting towards multi-cloud infrastructure for reducing their vendor dependent risk and achieving best services and resources for performance optimisation [3]. Virtualization is one of the key technologies of cloud computing, and most cloud-based systems are based on virtualization. However, the requirement of significant and redundant resources, issues of interoperability and deployment, load balancing and migration complexities make it unattractive for various types of big data analyses for ordinary users [5]. Docker is a container based virtualization technology and it has recently introduced Docker Swarm for the development of various types of multicloud distributed systems, which can be helpful in solving all above problems related to big data analysis for ordinary users

[4]. However, Docker is predominantly used in the software development industry, and less focus is given to the data processing aspect of this container-based technology.

Cloud computing offers users several benefits, such as high flexibility when using the cloud and the elimination of the need for expensive computing hardware and software. This approach introduces services that are highly important to users. Data storage, which allows users to outsource their data to the cloud without maintaining a local copy, is one of the key services of cloud computing. Cloud storage provides users the advantages of economy of scale and simultaneous reduction of the communication and computation costs of several applications. However, users still encounter threats that directly affect their data, including threats to confidentiality, data integrity, and access control. Several proposed methods have attempted to address these security challenges but have insufficiently fulfilled all user requirements.

Some approaches employ a third party auditor (TPA) to help a user verify the data with the cloud server provider (CSP) because the TPA has the ability and experience that the user may not have. Furthermore, when users do not have time to perform certain operations, users delegate tasks to the TPA, who then accomplishes the tasks on their behalf. However, a single third party auditor (STPA) may become a bottleneck in the system and may diminish system performance [5]. Thus, we propose a new scheme that employs *multiple third party auditors* (MTPAs) to overcome these risks.

However, once the data is uploaded on cloud server, the data owner loses the direct control of the data. How to guarantee security and privacy is a challenging problem in cloud storage. Although cloud server promises that the data is guaranteed well, some unreliable hardware or software may generate exceptions to destroy the data integrity. Furthermore, the cloud server is not completely trusted either. To save storage resource, the cloud server may delete the data accessed rarely without notifying the data owner. It is very possible that the data is tampered but the data owner doesn't know that. So clients need an effective method to verify the integrity of their data.

Provable data possession (PDP) is an important technique for the data owner to check whether the data is correctly maintained on remote cloud server without downloading the data. Although some different PDP schemes for different circumstance have been proposed, the core idea of them is almost the same. Specifically, to verify data integrity, the verifier launches a challenge to cloud server. Upon receiving the challenge information, the cloud server calculates an integrity proof by the data and the corresponding metadata. If the proof passes the verifier's verification, the data is proved to be intact. To make the checking result fair, the Third Party Auditor (TPA) trusted by both the cloud server and the data owner often conducts the data integrity auditing.

PDP can help the data owner check the data integrity.
Once the data is destroyed, the data owner also loses the data forever although he knows the truth. To improve the durability and availability of the data, data owners can generate multiple copies and store all copies on cloud server. If one copy is tampered, the data owner needs to recover the data from other copies. To further decrease risk, the data owner delivers multiple copies to different cloud storage servers. Even if all copies on one cloud storage server are broken, the data owner can get the data from other cloud storage servers. In this case, the PDP scheme must be extended so that all copies on distributed cloud servers can be checked together. To achieve this goal, special attentions should be paid to four challenges carefully. All copies should be checked by one challenge-response interaction. If the copy is verified one by one, the efficiency is very low and it is meaningless to construct PDP scheme for multi-copy. Data copies should be different from each other. If all copies are the same, the cloud servers can collude to cheat the data owner as they share only one copy but claim all copies are stored. The scheme should support public verification.

Currently, the network and society environment is becoming open, so public verification is an important and attractive feature of PDP, which needs to be realized. All cloud servers can work cooperatively to store and maintain all data copies for the data owner. In fact, it is an easy thing, because cloud computing is designed with open architectures and interfaces. Unfortunately, most of existing schemes only consider checking single data's integrity. For multiple copies, they must be run *N* times to check the integrity of *N* copies. It is very inefficient. To overcome this problem, some PDP schemes for multi-copy have been presented, most of which can verify all copies by only one challenge-response interaction. Although these protocols improve the efficiency greatly, they just deal with the simple case that all copies are stored on one cloud storage server. They are not suitable for the setting of multiple cloud servers. Furthermore, all of them are based on the traditional PKI technique. As we known, PKI is widely used in many applications but it brings big burden for certificate management. Compared with PKI, identity based cryptography (IBC) doesn't have such problems. IBC uses the user's unique identity as the public key, it doesn't need the certificate to ensure the validity of the public key. As a result, data integrity scheme avoids the heavy cost of certificate management. Therefore, it is important to design a data integrity scheme scheme for big data processing system in multi-copy and multi-cloud storage servers.

II. LITERATURE SURVEY

G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song et al. [13] analyzed the concept of PDP, which realizes the "spot-checking" technique to efficiently check the data on cloud server. In PDP, all data is split into numbers of blocks. By randomly checking parts of data, it can get the integrity status of the entire data with high probability. To support data dynamic,

C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia et al. [11] presented their PDP schemes based on authenticated skip list and the hardness of large integers factoring respectively. All these schemes are private verification, namely, only the data owner can check the data integrity. To achieve public verification,

Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li et al. [9] provided a dynamic PDP scheme with public audition. The advantage of public verification is that anyone can check the data on cloud server, which improves the flexibility for the scheme greatly.

R. Curtmola, O. Khan, R. Burns, and G. Ateniese et al. [12] first presented Multiple-replica provable data possession (MR-PDP) scheme for integrity checking of multiple replicas in cloud servers. However, MR-PDP scheme is not efficient because the replicas need to be checked one by one. Moreover, MR-PDP scheme only supports private verification. To overcome these problems,

Z. Hao and N. Yu et al. [10] provided a new public PDP protocol of the multiple replicas with privacy preservation. Following these works, many PDP schemes for multiple copies are presented.

M. Yi, L. Wang and J. Wei et al. [1] proposed a dynamic distributed PDP scheme with multi-copies in CSP, which designs a algorithm like 'binary search' to retrieve the corrupted data block. In these multi-copy schemes, all copies are stored on one cloud server. They cannot solve the integrity checking problem when data copies are distributed on multiple cloud servers. Furthermore, most of them are based on PKI technique. They need to bear the heavy cost of certificate management. In order to improve the efficiency.

Trnka et.al [6] analyzed the big data analysis can be based on traditional statistical methods or enhanced computational models and is used to analyse unstructured and unclean data of massive amount. A big data analytic is not a single tool/technology but a combination of multiple tools/technologies that are combined as a system/platform/framework and used to perform various operations in the entire big data analysis process such as data collection,

data cleaning, data modelling and visual interpretation of data.

H. Shacham and B. Waters et al [7] proposed a remote data-storage-correctness checking scheme based on HLA and an ECDSA signature to support public verifiability. This method uses only low computation resources because of the implemented algorithms. The support for public verifiability makes this scheme extremely flexible because the TPA can check the data on behalf of the users. Public verifiability allows anyone (not only the client) to challenge the CSP on data storage correctness without keeping private information. The TPA monitors the stored data in the remote server and notifies the client regarding data security.

III. An Efficient Public Data Integrity For Big Data Processing System In Multiple Cloud Storage

The block diagram of efficient public data integrity for big data processing system in multiple cloud storage is represented in fig.1.

A data volume is a specially designated directory within one or more containers that bypasses the Union File System (UFS). Data volumes are designed to persist data, independent of the container's life cycle. Therefore, Docker never automatically deletes these volumes when the user removes a container, nor the garbage collector removes volumes that are no longer referenced by a container. Here, two data volume containers are created for the previous Docker Swarm cluster, which can be shared among all the nodes of the cluster. Swarm worker is developed as a cluster of five Swarm Nodes by creating five lightweight VMs in VirtualBox on the same host computer (Mac OS X).

Swarm Managers are created on manager1, manager2 and manager3 Docker Machines. The manager1 is the primary manager (leader) as it is created first but this can be easily changed and reassigned. When the node is assigned the responsibility of a manager, it joins a *RAFT Consensus group* to share information and perform leadership election. The leader is the primary manager that maintains the state, which includes lists of nodes, services and tasks across the swarm in addition to making scheduling decisions. This state is circulated across the each manager node

through a built-in RAFT store. Consequently, managers have no dependency on an external key-value.

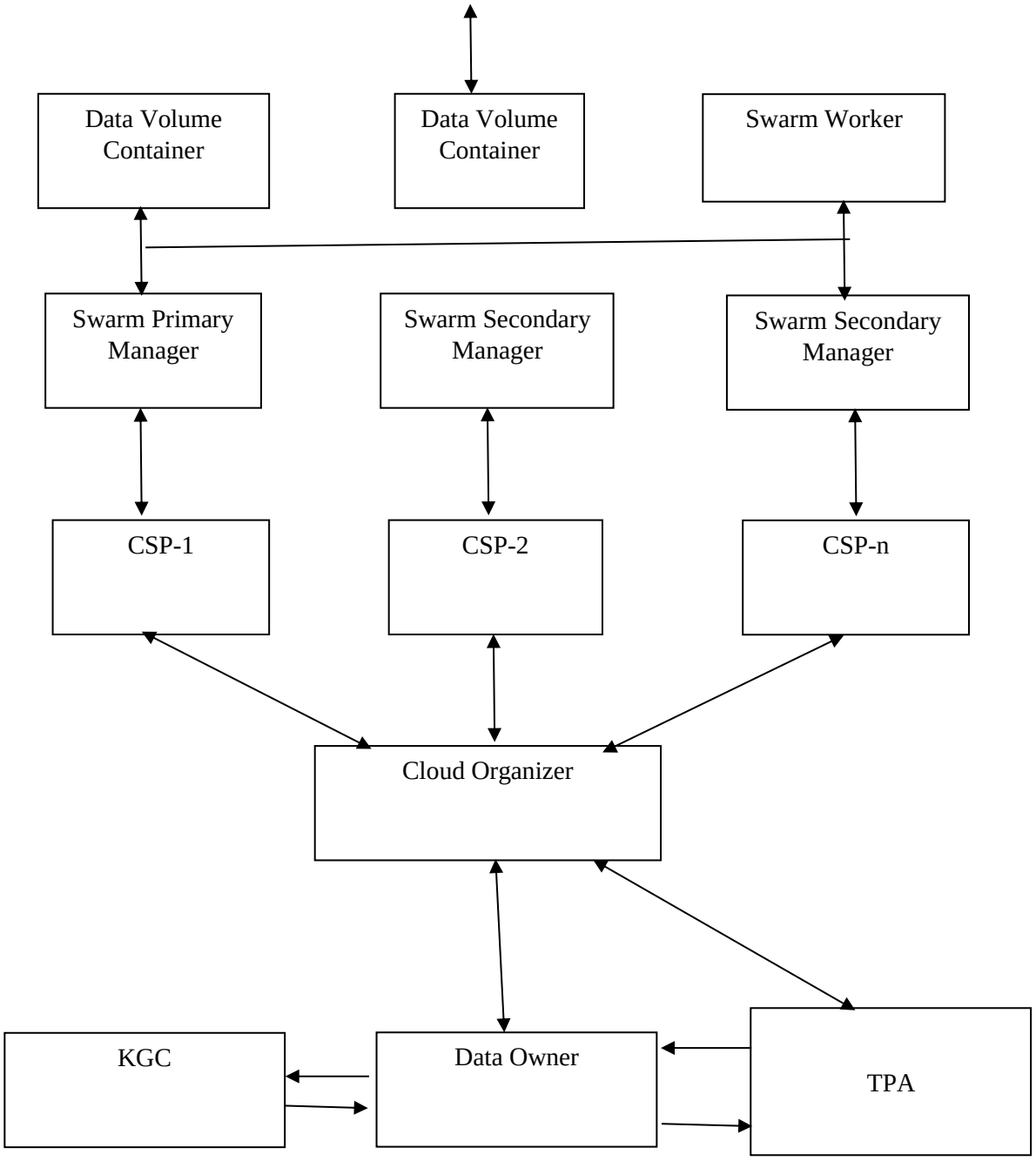

Fig.1: Block Diagram Of Efficient Public Data Integrity For Big Data Processing System In Multiple Cloud Storage

A primary manager (leader) is the main point of contact within the Docker Swarm cluster. In Docker Swarm, there could be one primary manager (leader) and multiple secondary managers (reachable managers) in case the primary manager fails. Primary manager works as a leader of the system and all the secondary managers contact with it regarding services and information. It is also possible to talk to secondary managers (replica instances) that will act as backups. However, all requests issued on a secondary manager are automatically proxied to the primary manager. If the primary manager fails, a secondary manager takes away the lead. Therefore, it facilitates a highly available and reliable cluster.

Cloud Service Provider (CSP) -1, CSP-2 and so on to CSP-n. Cloud organizer (CO) is the combiner of CSS. When storing file copies, the data owner first sends copies to CO. CO distributes different copies to the target CSS according to user's request. When challenging the file integrity, TPA first sends the challenge request to CO. CO distributes the challenge to the corresponding CSS. Upon receiving all the distinct proofs from CSS, CO aggregates them to be the complete proof and sends it to Third Party Auditor (TPA). In real-life, CO is supported by TPA, they can be bound as one service.

Key generation center (KGC) generates private keys for users. It uses the user's identity to calculate the private key and returns it to the user by secure channel.

Data owner rents the cloud storage service and stores massive data on multiple cloud servers. It generates many different copies of the outsourced file and stores the file copies to different cloud servers. It can be the organization consumer or the individual consumer.

TPA verifies the integrity of all outsourced copies on behalf of the data owner. Both the data owner and CSS trust that the TPA has capability and knowledge to honestly perform the verification work. TPA is assumed to be trustful, who is able to honestly perform the data integrity verification and returns the real result to the data owner.

IV. RESULT ANALYSIS

In this section result analysis is observed for efficient public data integrity for big data processing system in multiple cloud storage. The parameters are observed in terms of accuracy, precision and security.

Table.1 Performance Analysis

Performance Metrics	Data Integrity In Multiple Cloud Storage	Data Integrity In Single Cloud Storage
Accuracy	98	91
Precision	95	89
Security	99	87

The above table shows that an integrated approach to improve efficient public data integrity for big data processing system in multiple cloud storage gives the high accuracy, precision and security which are used to improve the public data security.

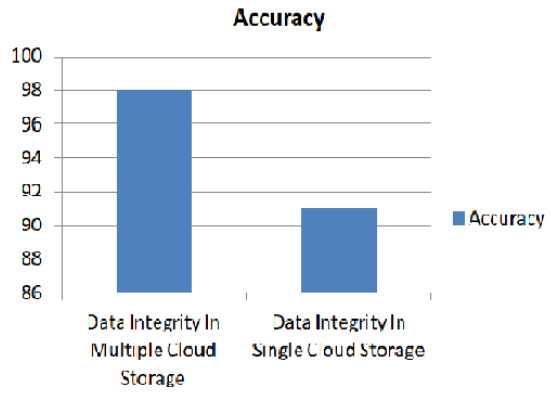

Fig.2: Accuracy Comparison Graph

Therefore the efficient public data integrity for big data processing system in multiple cloud storage has better accuracy.

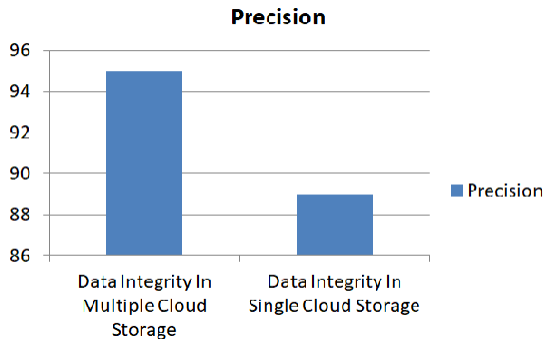

Fig.3: Precision Comparison Graph

In this comparison, the above graph shows that an efficient public data integrity for big data processing system in multiple cloud storage has higher precision.

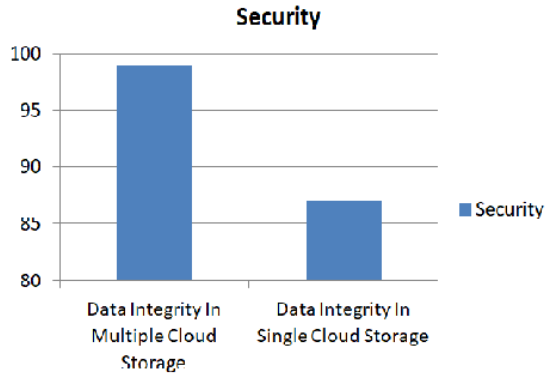

Fig.4: Security Comparison Graph

In the above comparison graph shows higher security when compared with other methods.

V. CONCLUSION

Big data processing system in multiple clouds explored another potential dimension. This efficient public data integrity for big data processing system is an inexpensive and user-friendly framework for everyone who has the knowledge of basic IT skills. Additionally, it can be easily developed on a multiple machines or multiple clouds. This framework showed that it has a potential to develop a big data processing system for everyone. This framework is designed to verify the integrity of multiple copies on multiple cloud servers within one challenge-response interaction. Moreover, the structural advantage of identity-based cryptography makes our scheme more efficient and secure. Hence this analysis achieved higher accuracy, precision and security. The outcomes of various methods are different, efficient public data integrity for big data processing system in multiple cloud storage achieved the best results in this analysis.

VI. REFERENCES

[1] M. Yi, L. Wang and J. Wei, "Distributed data possession provable in cloud," *Distributed and Parallel Databases*, vol. 35, no. 1, pp. 1-21, 2017.

[2] N. Naik, P. Jenkins, N. Savage, and V. Katos, "Big data security analysis approach using computational intelligence techniques in R for desktop users," in *IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2016.

[3] N. Naik, "Building a virtual system of systems using Docker Swarm in multiple clouds," in *IEEE International Symposium on Systems Engineering (ISSE)*. IEEE, 2016

[4] C. Anderson, "Docker [Software Engineering]," *IEEE Software*, no. 3, pp. 102–c3, 2015.

[5] K. Selvamani and S. Jayanthi, "A review on cloud data security and its mitigation techniques," *Procedia Computer Science*, Elseveir, vol. 48, pp. 347 – 352, 2015.

[6] Trnka, "Big data analysis," *European Journal of Science and Theology*, vol. 10, no. 1, pp. 143–148, 2014.

[7] H. Shacham and B. Waters, "Compact proofs of retrievability," *J. Cryptol.*, Springer-Verlag New York, vol. 26, no. 3, pp. 442–483, July 2013.

[8] D. Fisher, R. DeLine, M. Czerwinski, and S. Drucker, "Interactions with big data analytics," *Interactions*, vol. 19, no. 3, pp. 50–59, 2012.

[9] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 5, pp. 847-859, May, 2011.

[10] Z. Hao and N. Yu, "A multiple-replica remote data possession checking protocol with public verifiability," in *Proc. 2th Int'l Symp. Data, Privacy, E-Comm. (ISDPE)*, 2010, pp. 84-89.

[11] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in *Proc. 16th ACM Conf. on Comput. and Commun. Security (CCS), 2009*, pp. 213-222

[12] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR PDP: Multiple-replica provable

data possession,'' in *Proc. 28th IEEE Conf. on Distrib. Comput. Syst. (ICDCS)*, 2008, pp. 411-420.
[13] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, ''Provable data possession at untrusted stores,'' in *Proc. 14th ACM Conf. on Comput. and Commun. Security (CCS)*, 2007, pp. 598-609.
[14] M. S. Agarwal and M. Kwiatkowski, "Thrift: Scalable Cross- Language Services Implementation," *Facebook, Tech. Rep.*, 2007
[15] Y. Deswarte, J.-J. Quisquater, and A. Sadane, "Remote integrity checking," *Proceedings of the Sixth Working Conference on Integrity and Internal Control in Information Systems*, Springer, USA, pp. 1–11, 2004.
[16] D. Boneh, H. Shacham, and B. Lynn, ''Short signatures from the weil pairing,'' *J. Cryptol.*, vol. 17, no. 4, pp. 297-319, Sept. 2004.
[17] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Proc. CRYPTO*, vol. 2139. 2001, pp. 213– 229.