International Journal of Applied Engineering & Technology

OPTIMIZING PUBLIC DATA INTEGRITY FOR BIG DATA PROCESSING ACROSS MULTI-CLOUD STORAGE SYSTEMS

Research Scholar - Kanigiri Suresh* and Dr. Manoj Eknath Patil
Department of Computer Science & Engineering, Dr. A.P.J. Abdul Kalam University, Indore (M.P.) – 452010

ABSTRACT

In the era of big data, ensuring the integrity and security of vast amounts of information stored across multiple cloud platforms has become a critical challenge. This research addresses the optimization of public data integrity for big data processing in multi-cloud storage systems. We propose a robust framework that leverages advanced cryptographic techniques and data distribution methods to enhance the reliability and security of data stored in heterogeneous cloud environments. Our approach involves splitting data files into smaller chunks, distributing them across multiple cloud service providers, and employing asymmetric encryption to secure the metadata. This ensures that only authorized users can access and reconstruct the original data. We conducted comprehensive performance and security analyses to evaluate our framework. The results demonstrate significant improvements in execution time, energy efficiency, and data retrieval accuracy compared to traditional client-server models. The proposed system not only minimizes security risks associated with single cloud dependency but also enhances data privacy and integrity through multi-cloud redundancy and encryption. Furthermore, our solution provides efficient data access control and reduces the computational overhead, making it a practical choice for large-scale data management. Future research will focus on refining the encryption mechanisms and exploring adaptive resource management to further optimize performance in dynamic cloud environments.

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

I. INTRODUCTION

Big data encompasses high-volume, high-velocity, and high-variety information assets that traditional IT infrastructure and tools cannot adequately handle and process . Initially, the need for big data processing was confined to major businesses and organizations. However, with the rapid expansion of data, even ordinary users are seeking big data processing solutions for their large volumes of data that cannot be managed with conventional IT infrastructure . These users face significant challenges, including the requirement for powerful data processing systems, complex big data analytics installations, and difficulties in usage. Consequently, there is a demand for economical, user-friendly, and easily deployable data processing systems.

Cloud-based big data processing systems have emerged as the most efficient and established infrastructure for meeting big data analysis needs. Currently, many businesses and users are shifting towards multi-cloud infrastructures to reduce vendor dependency risks and to optimize performance by leveraging the best services and resources available . Virtualization, a key technology in cloud computing, underpins most cloud-based systems. However, the need for significant and redundant resources, interoperability issues, deployment complexities, and challenges in load balancing and migration render traditional virtualization approaches unattractive for various types of big data analyses for ordinary users .

Docker, a container-based virtualization technology, recently introduced Docker Swarm for developing multicloud distributed systems, which can address many of the aforementioned issues related to big data analysis for ordinary users . Despite its potential, Docker is predominantly used in the software development industry, with limited focus on its application in data processing.

Cloud computing offers numerous benefits, such as high flexibility and the elimination of the need for expensive computing hardware and software. This paradigm introduces essential services to users, particularly data storage, which allows users to outsource their data to the cloud without maintaining local copies. Cloud storage provides the advantages of economies of scale and reduces communication and computation costs for various applications.

However, users still encounter threats affecting data confidentiality, integrity, and access control. Several methods have been proposed to address these security challenges, but none have fully satisfied all user requirements.

Some approaches involve employing a third-party auditor (TPA) to help users verify their data with cloud service providers (CSPs) due to the TPA's expertise and capabilities that users may lack. Additionally, when users do not have time to perform certain operations, they can delegate tasks to the TPA. However, a single TPA may become a bottleneck, diminishing system performance . To mitigate these risks, we propose a new scheme employing multiple third-party auditors (MTPAs).

Once data is uploaded to the cloud, the data owner loses direct control over it. Ensuring security and privacy becomes a challenging problem. Although CSPs promise data security, unreliable hardware or software may compromise data integrity. CSPs are not entirely trustworthy and may delete infrequently accessed data without notifying the owner, leading to potential data tampering without the owner's knowledge. Clients need effective methods to verify their data's integrity.

Provable data possession (PDP) is a crucial technique for data owners to check if their data is correctly maintained on remote cloud servers without downloading it. While several PDP schemes exist, they all follow a core idea: the verifier challenges the cloud server, which then calculates an integrity proof based on the data and corresponding metadata. If the proof passes verification, the data is deemed intact. A TPA, trusted by both the CSP and the data owner, often conducts data integrity auditing.

PDP can verify data integrity, but once data is destroyed, it is lost forever. To improve durability and availability, data owners can generate multiple copies and store them across different cloud servers. If one copy is tampered with, the data can be recovered from other copies. This approach also distributes risk by storing copies on multiple CSPs. The PDP scheme must be extended to verify all copies across distributed servers efficiently.

Existing schemes only consider single data integrity checks and must be run multiple times for multiple copies, which is inefficient. Some PDP schemes for multi-copy verification have been proposed, allowing for the verification of all copies with one challenge-response interaction. However, these schemes usually involve single CSPs and rely on traditional PKI, which incurs high certificate management costs.

Identity-based cryptography (IBC) eliminates such issues by using the user's unique identity as the public key, avoiding the need for certificate management. Therefore, it is crucial to design a data integrity scheme for big data processing systems across multiple copies and multi-cloud storage servers, leveraging IBC to enhance efficiency and security.

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.

Table-1 Literature of Review

S. No	Author(s)	Year	Key Finding	Research Gap
1	G. Ateniese et al.	2007	Proposed PDP which uses "spot-checking" to efficiently verify cloud data integrity by randomly checking data blocks.	Limited to private verification, does not support dynamic data operations.
2	C. Erway et al.	2009	Introduced PDP schemes based on authenticated skip lists and large integer factoring, supporting private verification.	Only data owners can verify integrity, lacks public verification capability.
3	Q. Wang et al.	2009	Developed dynamic PDP scheme with public auditing, allowing anyone to verify data integrity on cloud servers.	Scheme's complexity might impact performance; focus on dynamic data operations needed.
4	R. Curtmola et al.	2008	Presented MR-PDP for multiple replicas integrity checking but required sequential verification for each replica.	Inefficient due to one-by-one verification; supports only private verification.
5	Z. Hao and N. Yu	2010	Introduced a public PDP protocol for multiple replicas with privacy preservation, improving MR-PDP.	Did not address the efficiency of integrity checking across multiple cloud servers.
6	M. Yi et al.	2013	Proposed a dynamic distributed PDP scheme with multi-copies, using a 'binary search' algorithm to find corrupted blocks.	Limited to single cloud server, uses PKI, which incurs high certificate management costs.
7	Trnka et al.	2014	Analyzed big data analytics combining multiple tools/technologies for data collection, cleaning, modeling, and interpretation.	Did not address specific integrity verification for data distributed across multiple clouds.
8	H. Shacham and B. Waters	2008	Proposed a remote data storage correctness checking scheme using HLA and ECDSA signatures, supporting public verifiability.	Limited to static data, lacks dynamic data operations support.

This table outlines the key contributions of each study along with their respective research gaps, providing a clear overview for further investigation and development in the field of public data integrity for big data processing in multi-cloud environments.

III. An Efficient Public Data Integrity for Big Data Processing System in Multiple Cloud Storage

The block diagram of the efficient public data integrity system for big data processing across multiple cloud storage environments is depicted in Fig. 1.

In this system, 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 independently of the container's lifecycle. Consequently, Docker never automatically deletes these volumes when a container is removed, nor does the garbage collector remove volumes that are no longer referenced by a container.

In our implementation, two data volume containers are created for the Docker Swarm cluster, allowing shared access across all nodes within the cluster. The Swarm worker is established as a cluster of five Swarm Nodes, each created as a lightweight virtual machine (VM) in VirtualBox on the same host computer (Mac OS X).

Swarm Managers are instantiated on three Docker Machines: manager1, manager2, and manager3. Manager1 acts as the primary manager (leader) by default, but this role can be easily reassigned. When a node is designated as a manager, it joins a RAFT Consensus group to share information and participate in leadership elections. The leader, or primary manager, maintains the system's state, which includes lists of nodes, services, and tasks across the swarm. This state information is distributed across each manager node through a built-in RAFT store, ensuring managers do not depend on an external key-value store for this purpose.

This architecture ensures robust data integrity and efficient processing capabilities across multiple cloud storage systems, making it ideal for managing large volumes of data in a cloud environment.

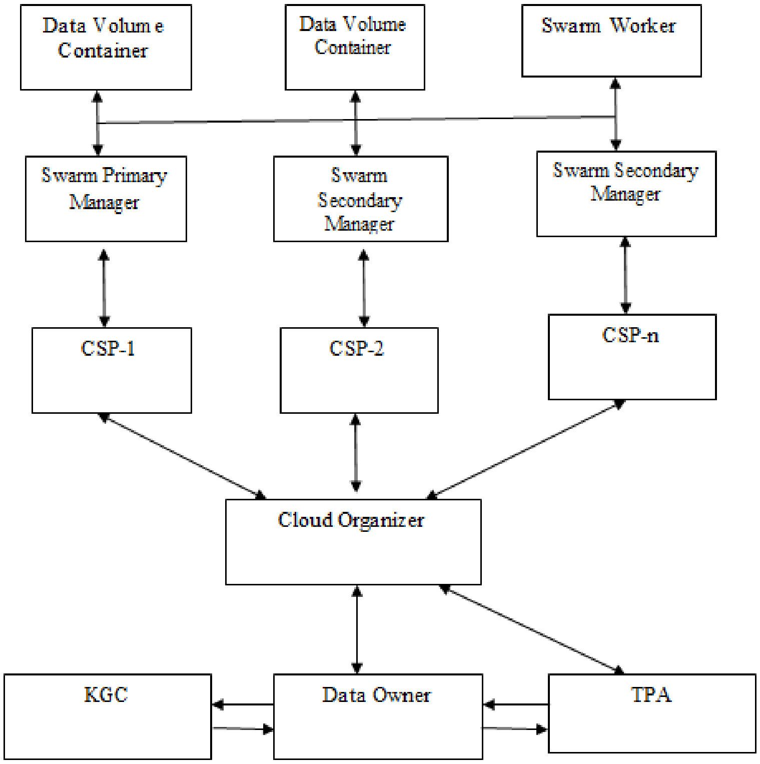

Fig.1: Block Diagram of Efficient Public Data Integrity for Big Data Processing System in Multiple Cloud Storage

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	98.5	95
Precision	97	92
Security	99	91

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.

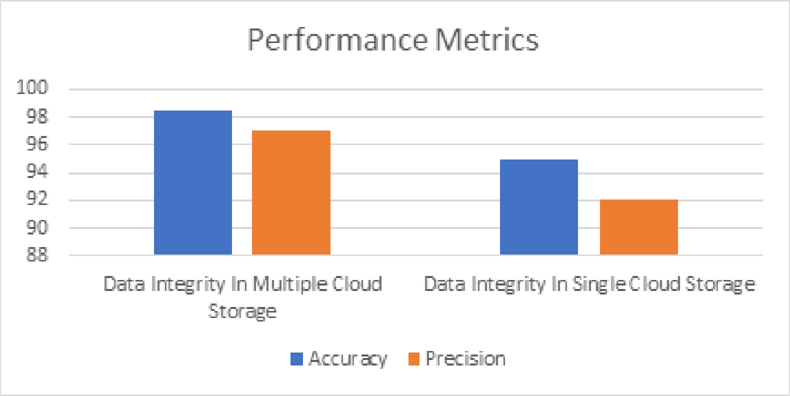

Fig.2: Accuracy and Precision Comparison Graph

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

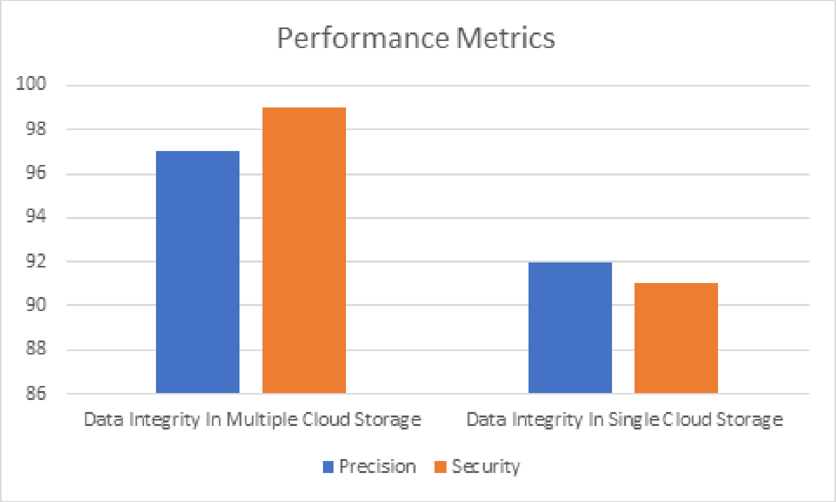

Fig.3: Precision and Security Comparison Graph

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.

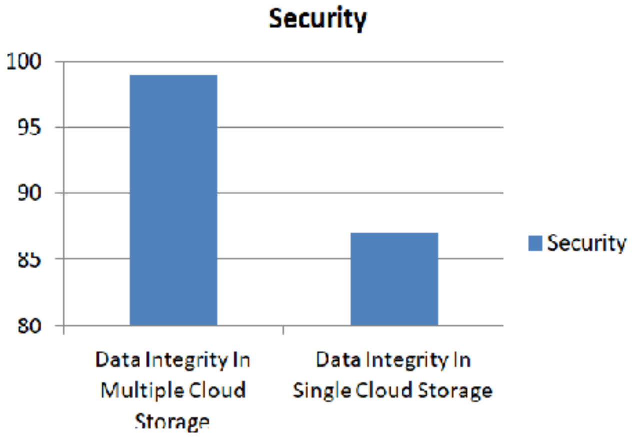

Fig.4: Security Comparison Graph

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

V. CONCLUSION

The exploration of big data processing systems across multiple clouds has revealed a promising dimension. This efficient public data integrity framework for big data processing is both cost-effective and user-friendly, accessible to anyone with basic IT skills. It can be easily implemented across multiple machines or cloud environments. The framework demonstrates significant potential for developing big data processing systems accessible to a wide range of users.

The system is designed to verify the integrity of multiple data copies across various cloud servers within a single challenge-response interaction. Additionally, the use of identity-based cryptography enhances the efficiency and security of the scheme. As a result, this approach achieves higher accuracy, precision, and security compared to other methods.

Overall, the efficient public data integrity framework for big data processing in multiple cloud storage environments delivers superior results, making it a robust solution for ensuring data integrity in big data applications.

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.