Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 4, April : 2023

Verification Protocols to Improve Cloud Server Data Availability and Security

[1]D. Lakshmi Narayana Reddy
[1]Assistant Professor.
Srinivasa Ramanujan Institute of Technology(Autonomous), Rotarypuram, Anantapuramu, A. P, India.
Department of Computer Science and Engineering
Mail id: lakshmi1217@gmail.com.
Mobile no: 9866025260

Abstract

Distributed computing networks have a prominent part in modern computer architecture since they enable the processing, analysis, and storage of sensitive data. The subject matter has generated considerable attention within the field of researching different encryption algorithms. There exists an urgent need to build a reliable and tested security methodology that improves the efficiency of data processing in Cloud servers, since ongoing cryptographic research indicates the susceptibility of server-side data security to possible intrusions. This study aims to provide a comprehensive understanding of the implementation process of a cloud-centric cloud storage architecture. Additionally, it introduces a novel security schema that verifies the integrity of encrypted secret data storage. The authors of this study provide an innovative computational security approach to address the challenge posed by internal nodes. They use efficient security algorithms to effectively reduce computational overhead. The findings presented in this study provide empirical support for the superior efficacy of the proposed mechanism when compared to other widely used security measures.

1. INTODUCTION

In the last century, the progress in computer technology has played a significant role in enabling the implementation of automation in several sectors. There exists a prevailing view among persons that the utilization of cloud technology and cloud computing has the capacity to engender a substantial paradigm shift in the realm of computing within the forthcoming years. The phrase "cloud computing" was first used at the 2006 SES (Search Engine Strategies) global conference hosted in San Jose. Following that, the clarification of NIST standards [1] has been provided. The National Institute of Standards and Technology (NIST) is a federal agency that is responsible for developing and promoting measurement standards, technology, and innovation in the United States. Cloud architecture has always had a substantial impact on corporate economic strategies with regard to storage and service provision, owing to its inherent authenticity and flexibility. Scholars from many global areas have been actively engaged in efforts to improve the

cost-effectiveness and efficiency of the fundamental architectural model of the Cloud, specifically with regards to data storage and provisioning. The dominance of storage as a cloud service is widely acknowledged, although the presence of competing options. The significant increase in the quantity and speed of data accumulation over the last few years may be ascribed to the fast progress of networks and the implementation of pivotal technologies such as the Internet of Things (IoT) and Industrial Internet of Things (IIoT). These technologies are of utmost importance in the collection of different data from various sources, making them important resources. This problem occurs when the calculation of large amounts of data becomes challenging, leading to a situation where the storage capacity available on local servers or workstations is inadequate to meet the user's needs. Cloud computing has become the prevailing framework for the management of virtual servers and the provision of storage capacity, with the objective of offering optimal storage capabilities. Cloud computing has shown its ability to do computational activities and manage a significant number of storage devices, facilitating their seamless collaboration via the use of orchestration principles. The achievement of this objective is facilitated by the use of the Distributed File System (DFS) and the application of virtualization principles. A number of well-known organizations, including Google, Apple, and Microsoft, have developed their own cloud storage systems, such as Dropbox, Google Drive, and iCloud, respectively, with the aim of offering these services to its customers. These companies are enabling the development and use of cloud-based services, which need substantial storage capacity.

Numerous academic investigations [2] have shown the presence of notable security vulnerabilities inside the cloud storage service. Furthermore, there is a prevalent belief within the technological domain that the Internet of Things (IoT) will smoothly assimilate into our everyday lives via a broad array of applications. The prevalence of various devices inside the Internet of Things (IoT), such as smartphones and smartwatches, may be linked to their significant level of portability. These gadgets consistently gather data via diverse techniques, which is dependent on the cloud storage framework permitted by distinct service providers. The need for cloud storage is expected to see significant rise due to the rising proliferation of networked devices. The outcome of this will result in heightened network congestion and diminished response durations, hence adversely affecting the Quality of Experience (QoE) for end users. If there is a gradual decline in the quality of experience (QoE), it is probable that the performance of the Internet of Things (IoT) device will also decline, leading to user discontent. In the year 2012, Cisco proposed the notion of cloud computing as a strategic approach to tackle the many issues and concerns related with this particular computing paradigm. The fundamental concept behind cloud computing is the virtualization and localization of distant cloud servers, with the objective of positioning them in closer proximity to end users. This technique enables the migration of these procedures to a decentralized network. The utilization of cloud computing in the domain of data storage endeavors to mitigate existing security apprehensions and enhance

the quality of experience (QoE). Contemporary technological research indicates that cloud computing is extensively acknowledged as a spatial-temporal network that places emphasis on end-users, effectively enhancing the caliber of their experiences. In addition, the network undergoes constant monitoring by cloud service providers on a global level.

The schematic shown in Figure 1 illustrates the data transfers that take place inside a traditional cloud and cloud computing setup. At the outset, clients will entrust their personal information to cloud servers that are vulnerable to security breaches and located at a considerable distance. Cloud service providers possess the capacity to access and scrutinize a substantial proportion of data that is kept inside the cloud. In contemporary times, the integrity of data privacy is being undermined by malevolent actors often known as hackers. Multiple research have been conducted to explore a wide array of cryptographic techniques [3] with the objective of safeguarding sensitive information stored in cloud settings. Previous research has shown the effectiveness of cryptographic techniques in guaranteeing the secrecy and integrity of data. Nevertheless, it is evident that their ability to identify and mitigate internal attacks is compromised. The utilization of Cloud nodes with the objective of guaranteeing anonymity has been thoroughly investigated in several scholarly articles [4]. Nevertheless, research has shown that even with the use of sophisticated algorithms like the Reed Solomon code and the MD5 algorithms, occurrences of data loss persist.

The main contribution of this study is the creation of a cloud-centric architecture that improves the effectiveness of data processing on cloud servers. This architectural solution aims to tackle the prominent challenge of verifying data ownership on cloud servers. The fundamental goal of demonstrated data possession is to verify if the data stored on the server has been modified. This is achieved by frequent and effective evaluations of the data's integrity after it has been sent by the client.

.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 4, April : 2023

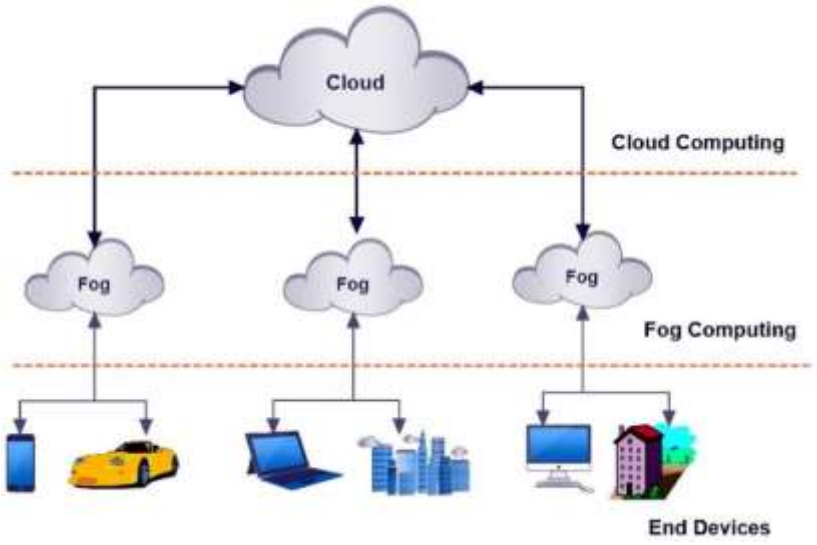


Figure 1. Generalized Cloud-Cloud Storage architecture

2. Literature Review

In recent years, there has been a discernible surge in the level of interest directed towards the notion of distributed computing. A considerable number of individuals have dedicated their time and efforts towards enhancing the practicality and efficacy of the subject, as seen by the academic references provided [7–11]. Mist processing is a nascent system innovation that seeks to overcome the constraints inherent in conventional distributed computing approaches. This methodology has generated considerable attention from scholars in academia as well as professionals in the industry. There is a scarcity of research that specifically examine the pertinent security aspects and system design [12–15], which is not surprising considering the novelty of the haze processing idea. The majority of existing technical literature primarily focuses on providing a comprehensive understanding of the complex structure and significant performance indicators associated with Cisco's innovative proposition for cloud computing architecture. This emphasis is due to the early stage of investigation in the domain of cloud computing [17]. The architectural design has three discernible layers: a fabricated cloud structure, an intermediate nebulous layer positioned between Internet of Things (IoT) devices and cloud servers, and a centralized network of IoT devices. The authors of References [13, 15] put forth an additional component for the allocation of residual tasks that effectively addresses power consumption, administrative idleness, CO2 emissions, and other pertinent factors, alongside their discussion on the technical complexities of mist registration technology. The authors have conducted a comprehensive analysis of the problems and security issues related to haze figuring, as shown in References [7, 16, 12]. In their research, Alrawais et al. (2016) used

fuzzy processing to do a contextual analysis of the issue of confirmation disavowal within the framework of the Internet of Things (IoT).

However, previous studies have not yet examined the full scope of the data review issue inside a loosely-defined distributed storage environment. To the extent of our current understanding, the examination of information evaluation has not been investigated within the framework of Cloudcloud capacity, despite the presence of significant findings related to distributed storage. Previous research have explored a range of approaches for assessing distributed storage systems (Refs. [9–16]) up till recently. Nevertheless, the current review protocols exhibit some constraints, such as a restriction on the overall quantity of reviews and the potential risk of data exposure by a Third-Party Auditor (TPA). The scholarly essay named "Yang and 32" was published in the Journal of Science and Technology at Tsinghua University, Volume 25, Issue 1, 2020. This study delves into several elements pertaining to the subject matter. The item may be located between the range of pages 28 to 43. Jia undertook an inquiry into probable conspiratorial factors related to the excess capacity and communication expenses involved with the current research. The study's results indicate that the accumulation of data necessary for the functioning of the frameworks would progressively grow overwhelming, a common obstacle seen in comparable initiatives. Concurrently, due to the need of in-person interactions between third-party administrators (TPA) and clients, these programs include notable expenses linked to communication. The Merkle Hash Tree (MHT), a well recognized and used data structure, has been extensively employed in several research projects pertaining to the auditing of networked storage [9, 2, 4, 6]. The aforementioned studies provide an enhanced methodology for evaluating distributed storage systems, which demonstrates reduced metadata-related capacity overhead and correspondence overhead in comparison to earlier methodologies.

Liu et al. (2019, 2016) have introduced a multi-copy Merkle Hash Tree (MHT) methodology for individual data blocks, including a mechanism to notify the distributed storage administrator in case of any replica failures. As a consequence, the importance of confirmation communication is reduced, leading to more flexibility in client information in the event of a capacity management error. The issue is exacerbated by the Multi-Replica Dynamic Public Auditing (MuR-DPA) method, as described in Reference [9], which requires the client to manage and create a separate unique Merkle Hash Tree (MHT) for each replica. This results in significant communication costs. However, the use of traditional review approaches for distributed storage in the context of haze distributed storage presents difficulties due to the distinctive attributes of haze distributed storage, including its specific operating protocols throughout successive working sessions. The vulnerability of current review plans that rely on Merkle Hash Trees (MHT) to replay attacks arises from the capacity administrator's ability to reuse a processed hash value for the sake of verifying information integrity, even in cases where the content has been modified or removed.

3. PROPOSED WORK

This section largely discusses an improved security approach for efficiently storing data at Cloud nodes, which are subject to frequent and rapid alterations. The primary purpose of this strategy is to effectively manage dynamically updated data at nodes that are centered on the Cloud, using established principles of data ownership. Multiple analyses have shown that the speed of demonstrating verifiable data ownership is associated with the quantity of disk input/output (I/O) operations, rather than the cryptographic analysis itself. The approach used in our study involves the utilization of B+ trees for data storage, specifically designed to accommodate composite keys. The suggested methodology offers enhanced value via the use of a composite key that amalgamates the index value and timestamp of the block. B+ trees are highly favored because to their self-balancing topologies, which enable efficient computation of results by effectively accepting dynamic changes originating from Cloud nodes. A crucial enhancement to the proposed methodology is the initial segmentation of the file intended for storage on the server into discrete blocks. These blocks may then be subjected to processing in order to generate metadata. The metadata is then saved in the server throughout the indexing process and retained for a prolonged duration. The use of this methodology primarily capitalizes on the quantum channel.

3.1 Quantum channel for Cloud

The architecture of the Quantum Key Distribution (QKD) device involves two primary steps: the initialization of contact via the Quantum channel and the subsequent transmission of information through the conventional channel. The output of the gadget is analyzed based on the given parameters.

a. Secured Main Rate (Skr)

b. Error rate of Qubit (Qer)

The Protected Key Rate (Skr) is stated to be Skr = vBP..... (1)

Where 'vBP' is deemed from the source as pulses per second and 'BP' is the bit rate per pulse.

3.2 Analysis to calculate the Qubit error rate

The detections at Bob's end are used to estimate the mean observed signal per pulse and the bit rate per pulse with the help of protocol inherent efficiency Ni to first calculate the Qubit error rate.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 4, April : 2023

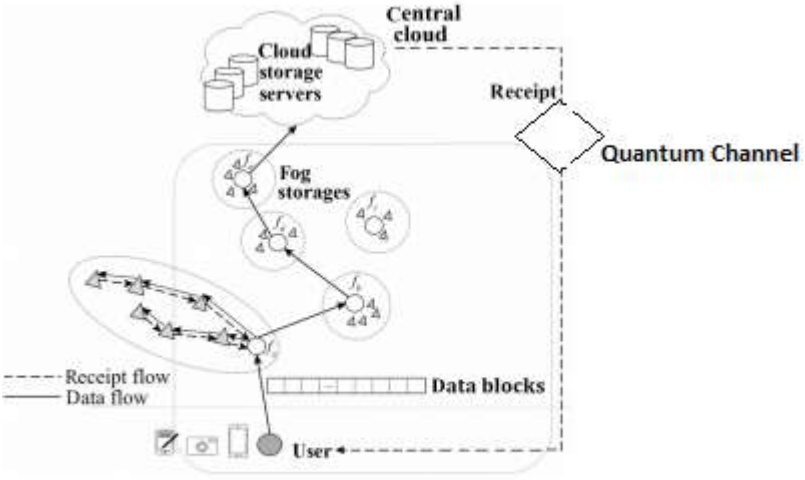


Figure 2: Proposed Quantum Cloud Storage Architecture

3.3.3 Computational Notations Utilized

Table 3.1: Notations used

Notation used	Attribute
F_s	Denotes the file to be stored at server in terms of blocks
O_d	Data Ownner
SOD	Outsourced data that is to be stored
T_s	Timestamp variable
CI_b	Counter variable to track Index over the file blocks
M_k , N_k	Pair of public key and private key
H(x)	Cryptographic hash fuction
P_{key}	Encryption mechanism used to encrypt the generated tags
P_{key}^{-1}	Decryption mechanism

The following elucidation presents the fundamental concept behind the operational mechanism. Prior to transmission to the server, the file Fs undergoes a process of fragmentation into many chunks. Once the preprocessing of these blocks is completed, metadata is generated and then sent to the server. In the preprocessing step, each block is assigned a randomly generated token, which serves the purpose of verifying the presence of the data on the server at a later stage. The encryption of the randomly produced tokens for each block is performed using the private key,

denoted as Pkey. Subsequently, the encrypted tokens are sent and associated with the corresponding file. The file storage technique is implemented using a block-by-block approach, which is governed by the B+ tree storage mechanism. The block index variable serves as the fundamental basis for this strategy.

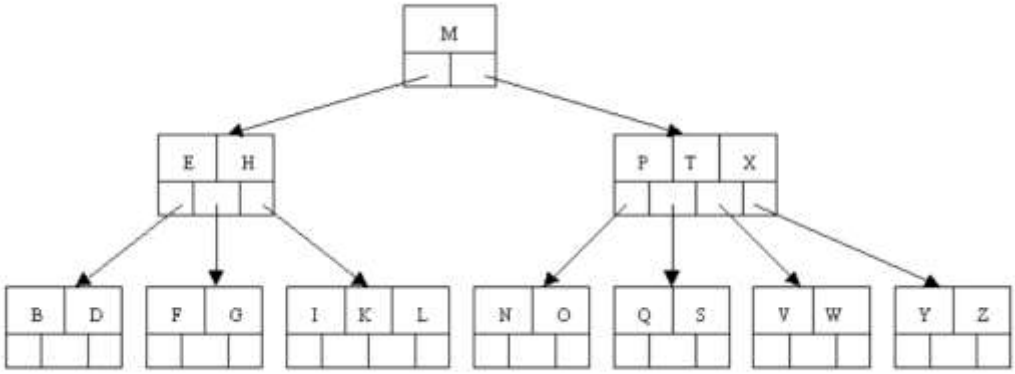


Figure 3. Block management using B+ Tree along

The terminal nodes of the B+ tree are the exclusive locations where data is stored. To locate the corresponding data block, the internal nodes of the tree are linked to index values that are associated with a timestamp.

3.3.4 Initial setup

Initially the given file F_n is classified into 'n' blocks such as $F_n[1]$, $F_n[2]$, $F_n[3]$……. $F_n[n]$ with the time stamp of $\mathbf{T_s}$ for each file block where these blocks are stored in the Cloud server based on their index values.

Algorithm 3.1: Initial Setup

Begin

$CI_b = 0$

While $(CI_b <= \mathbf{n})$

Begin

Compute $\mathbf{L_i} = \mathbf{H}[T_{si}, \mathbf{F_n[i]}]$

$\mathbf{L'_i} = P_{key}(\mathbf{L_i})$

$CI_b = CI_b + 1$

Transmit to SOD $(\mathbf{F_n[i]}, T_{si}, \mathbf{L_i})$

End

End

Utilizing the aforementioned procedure, a file undergoes first segmentation into distinct blocks, then stored on a Cloud server. The data is augmented with associated tags, which are systematically organized upon assignment.

3.3.5 Verification Module

During this phase, the data proprietor may verify the accuracy of the information stored on the Cloud server. This process involves the random selection of an index value, which is then sent to the Cloud server along with the corresponding time stamp value.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 4, April : 2023

Algorithm 3.2: Verification Module

Begin

O_d transmits the data to SOD $\{j, T_{sj}\}$

SOD evaluates $A = H(T_{sj}, F_n[j])$ on locating the index values.

SOD reclaim L'_i and transmit back the verifier the testimony as (A, L_i)

Verifier validates the testimony as :

$L_i = P_{key}^{-1}(L'_i)$

If $(A = L_i)$ then accept else reject

End

Upon reaching the Cloud server, the data undergoes a validation process against the specified tag. Subsequently, a newly generated tag, which is associated with the existing tag in the metadata, is sent. Through the implementation of dual-factor authentication, the process enables the verification of the authenticator.

4 Block Management at the Cloud layer

In this analysis, we will examine a B+ Tree of order n, where the tree has n key values and n+1 pointer keys. Our objective is to assess the efficiency of block operations inside this tree structure. The keys' values serve the purpose of identifying the pointer node Np that corresponds to a certain node inside the B+ tree. In the B+ tree data structure, each leaf node has a pointer that references the data page where all the leaf nodes are connected. This document presents a comprehensive analysis of the block operations, namely Insert, Delete, and Update.

4.1 Insertion

The process of inserting into the proposed system involves validating the entry each time the leaf node is accessed in the future. The fundamental concept behind the operation of this approach is the recursive insertion of a record by invoking the insertion algorithm on a designated child node. The process of recursion continues until it reaches the terminal node associated with its designated position, at which juncture it reverts back to the root node.

The proposed methodology involves designating the pointer node Z as the primary node, assigning a time stamp Ts, and introducing a new pointer as an initial null value until a split operation occurs. The variables Z and Ts denote the index and time stamp of a particular node, respectively.

Algorithm 3: Insertion of the record in to block (Pointer node (Z ,T_s), new)

Begin

Let N_p = Pointer to the node

If (pointer node is not a leaf node)

Locate s as if $Z_s = Z$ and $(Z<Z_{i+1})$

Insert (Z ,T_s), new

Return

elseif (N_p include a space)

Put(new N_p)

New = null

Return

else

Split N_p in the process of creating new node

New = points the smallest value in the node

Create new node

4.2 Delete a record in the Cloud layer

The process of inserting into the proposed system involves validating the entry each time the leaf node is accessed in the future. The fundamental concept behind the operation of this approach is the recursive insertion of a record by invoking the insertion algorithm on a designated child node. The process of recursion continues until it reaches the terminal node associated with its designated position, at which juncture it reverts back to the root node.

The proposed methodology involves designating the pointer node Z as the primary node, assigning a time stamp Ts, and introducing a new pointer as an initial null value until a split operation occurs. The variables Z and Ts denote the index and time stamp of a particular node, respectively.

Algorithm 4: Delete a record in the block (Pointer node (Z ,T_s), old)

```
Begin
    Let Np = Pointer to the node
    If (pointer node is not a leaf node)
            Locate s as if  Zs = Z and (Z<Zi+1)
            Delete  (Z ,Ts ), old
            Return
      If  (old = = null)
            Return
      Else
            Remove the old node from null
            If(Np has amd entry > n/2)
           Set old to null
      Endif
    Endif
End
```

4.3 Handling dynamic updates

During the process of block updating, the tokens and the current block are simultaneously changed by the overwrite operation done on the current block. When a request for a specific block update is received by the Cloud server, it proceeds to search for the requested block. Once the server has located the block, it proceeds to update both the block itself and its associated token.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 4, April : 2023

Algorithm 5: Update the block

Begin

Generate the request to SOD: update m, T_{sm}

At the SOD:

explore m, T_{sm}

transmit to O_d $(F_n[m], L'_m, T_{sm})$

At the O_d :

$F_n[m] = F_n[m]'$

$T_{sm} = T_{sm}'$

Evaluate a new tag $V_s = H(F_n[m], m, T_{sm})$

$V_s' = P_{key}(V_s)$

Transmit to SOD : $(F_n[m], L'_m, T_{sm})$

END

5. Performance analysis

This section of the study involves a comparative analysis between the proposed approach and the previous research conducted by Wang et al. [13], using a series of test examinations. In order to establish a more intricate association, we conducted modifications to many conditional variables, such as block size and communication speed, among others. In the context of information security, the proposed approach involves the use of Xor-Combination, while Wang et al. employ Reed-Solomon coding as their chosen technique. In essence, the proposed approach and the method used by Wang et al. both utilize tailored iterations of the CRH (Customized Robust Hashing) and MMD (Maliciously Manipulated Detection) algorithms in order to discern hazardous modifications. Therefore, the suggested plot is contrasted and shown as follows. Table 2 displays the anticipated time complexity of the proposed setup and verification modules. This study compares the proposed mechanism with the RS coding technique and the Xor combination algorithm in terms of data processing, communication expenses, and update detection costs.

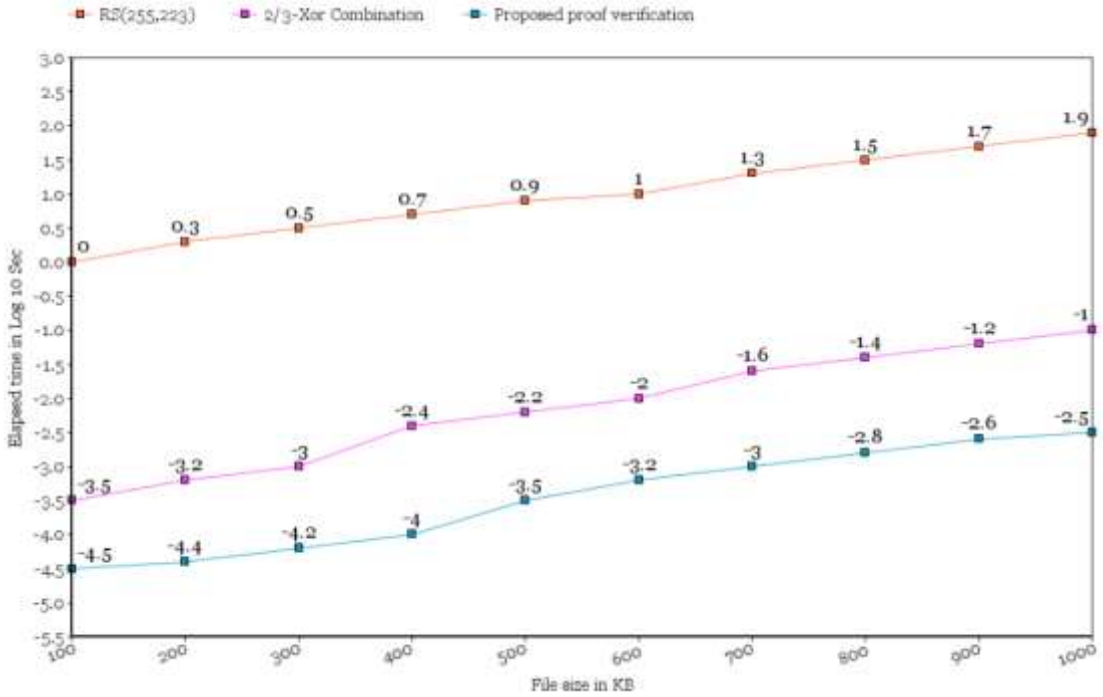
Table .2: Complexity analysis of the proposed algorithm

Phase	**Space Complexity**	**Communication complexity**	**Time Complexity**
Set-up Phase	O(n(k+m+p)	K+m+p bits	O(n)
Verification Phase	O(k+m)	M+q bits	$O(\log_b n)$

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 52, Issue 4, April : 2023

A. Data Processing

In the given context, a dataset consisting of blocks of varying sizes between 100KB and 1MB is systematically processed. The evaluation of a proof verification approach is conducted by using RS code and the 2/3-XoR combination method with parameters (255, 223). All of the algorithms maintain the integrity of the data partitions, ensuring that each iteration processes a single data block. Presented below is the comparative analysis of the execution times that has been generated.



6. Conclusion

This paper presents a novel security protocol inside a cloud-centric cloud storage architecture, which serves the purpose of authenticating the storage of sensitive data. The present study introduces a complete computational security mechanism aimed at mitigating the issue of internal nodes. This mechanism achieves a reduction in computational cost via the use of efficient security algorithms. The proposed approach seeks to enhance the processing data efficiency of the Cloud server. Moreover, it effectively minimizes the use of network capacity while facilitating instantaneous updates of data. Moreover, the efficacy of this technique may be enhanced by taking into account the distinctive characteristics of the edge network.

7.References

[1] P. Mell and T. Grance, “The nist definition of cloud computing,” 2010.

[2] D. Chen and H. Zhao, “Data security and privacy protection issues in cloud computing,” in Computer Science and Electronics Engineering (ICCSEE), International Conference on, vol. 1, pp. 647–651, IEEE, 2012.

[3] https://en.wikipedia.org/wiki/Clou_computing.

[4] L. A. Maghrabi, “The threats of data security over the cloud as perceived by experts and university students,” in Computer Applications & Research (WSCAR), 2014 World Symposium on, pp. 1–6, IEEE, 2014.

[5] S. Debnath, S. Neog, S. C. Sahana, and B. Bhuyan, “Study on Secrecy and Privacy Preserving Approaches over Cloud Data Outsourcing,” in International Conference on Computing and Communication System(I3CS), vol. II, pp. 135– 142, 2015.

[6] C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-preserving public auditing for data storage security in cloud computing,” in Infocom, 2010 proceedings ieee, pp. 1–9, IEEE, 2010.

[7] D. J. Abadi, “Data management in the cloud: Limitations and opportunities.,” IEEE Data Eng. Bull., vol. 32, no. 1, pp. 3–12, 2009.

[8] A. Sharma et al., “Privacy and security issues in cloud computing,” International Journal of Global Research in Computer Science (UGC Approved Journal), vol. 4, no. 9, pp. 15–17, 2013.

[9] S. Ruj, M. Stojmenovic, and A. Nayak, “Decentralized access control with anonymous authentication of data stored in clouds,” IEEE transactions on parallel and distributed systems, vol. 25, no. 2, pp. 384–394, 2014.

[10] S. Debnath and B. Bhuyan, “An Expressive Access Control Mechanism with user Revocation for Cloud Data Outsourcing,” in Proceedings of the International Conference on Electrical, Electronics,Computers, Communication, Machanical and Computing(EECCMC), pp. 250–256, IEEE, 2018.

[11] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward secure and dependable storage services in cloud computing," IEEE transactions on Services Computing, vol. 5, no. 2, pp. 220–232, 2012.

[12] Y. A. Younis, K. Kifayat, and M. Merabti, "An access control model for cloud computing," Journal of Information Security and Applications, vol. 19, no. 1, pp. 45–60, 2014.

[13] D. Servos and S. L. Osborn, "Current research and open problems in attributebased access control," ACM Computing Surveys (CSUR), vol. 49, no. 4, p. 65, 2017.

[14] D. R. Kuhn, E. J. Coyne, and T. R. Weil, "Adding attributes to role-based access control," Computer, vol. 43, no. 6, pp. 79–81, 2010.

[15] J. Hur, "Improving security and efficiency in attribute-based data sharing," IEEE transactions on knowledge and data engineering, vol. 25, no. 10, pp. 2271–2282, 2013.

[16] K. Yang and X. Jia, "Expressive, efficient, and revocable data access control for multi-authority cloud storage," IEEE transactions on parallel and distributed systems, vol. 25, no. 7, pp. 1735–1744, 2014.

[17] J. Han, W. Susilo, Y. Mu, J. Zhou, and M. H. A. Au, "Improving privacy and security in decentralized ciphertext-policy attribute-based encryption," IEEE Transactions on Information Forensics and Security, vol. 10, no. 3, pp. 665– 678, 2015.

[18] A. Shamir, "Identity-based cryptosystems and signature schemes," in Workshop on the theory and application of cryptographic techniques, pp. 47–53, Springer, 1984.

[19] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in Annual international cryptology conference, pp. 213–229, Springer, 2001.

[20] A. Sahai, B. Waters, et al., "Fuzzy identity-based encryption.," in Eurocrypt, vol. 3494, pp. 457–473, Springer, 2005.

[21] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure attribute-based systems," Journal of Computer Security, vol. 18, no. 5, pp. 799–837, 2010.

[22] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization.," in Public Key Cryptography, vol. 6571, pp. 53–70, Springer, 2011.

[23] M. S. Kiraz and O. Uzunkol, "Still wrong use of pairings in cryptography," arXiv preprint arXiv:1603.02826, 2016.

[24] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in Security and Privacy, 2007. SP'07. IEEE Symposium on, pp. 321–334, IEEE, 2007.

[25] K. E. Fu, Group sharing and random access in cryptographic storage file systems. PhD thesis, Massachusetts Institute of Technology, 1999.