University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)

ISSN: 2714-3627

A Journal of the Department of Computer Science, University of Ibadan, Ibadan, Nigeria

Volume 14 No. 1, June, 2025

journals.ui.edu.ng/uijslictr
http://uijslictr.org.ng/

University of Ibadan
Journal of Science and Logics in ICT Research

Development of a Hybrid Blockchain-Based File Sharing System in a Cloud Environment

[1] ✉ Osuolale A. F., [2]Alimi O. D. and [3]Okeowo I. T.

[1,2,3]Department of Computer Science, The Federal University of Technology, Akure, Nigeria.
[1]aofestus@futa.edu.ng [2]davidalimi148@gmail.com [3]okeowoidris01@gmail.com

Abstract

Conventional file storage and sharing systems encounter issues with security vulnerabilities, transparency shortcomings, and ineffective distribution. This initiative introduces an innovative blockchain-based framework for secure file sharing, addressing these challenges through the utilization of blockchain, Interplanetary File System (IPFS), and Advanced Encryption System (AES) encryption. The system uses a blockchain network to record file metadata like ownership, timestamps, and access permissions, ensuring data transparency and integrity. AES encryption secures data confidentiality, while IPFS efficiently distributes file chunks across nodes, enhancing availability and reliability. This approach creates a secure, efficient, and transparent file sharing solution. Performance evaluations demonstrate that the developed model outperforms existing models in terms of response time and requests per second, highlighting its efficiency under load.

Keywords: Hybrid Blockchain, Interplanetary File System (IPFS), Advanced Encryption System (AES)

1. Introduction

In the era of digital technology, the widespread adoption of file storage and sharing brings convenience but is accompanied by challenges such as security vulnerabilities, transparency issues, and inefficient distribution. As data volumes continue to grow exponentially, limitations of traditional centralized systems persist. This presents an urgent need for innovative solutions that ensure efficiency alongside robust security. Groundbreaking technologies like blockchain and the InterPlanetary File System (IPFS) offer decentralized models that address the weaknesses of conventional systems. Blockchain ensures tamper-proof, transparent transactions without central control, as noted by Mann *et al*., [9].

IPFS revolutionizes content storage and sharing by distributing data across nodes. Integrating these technologies paves the way for the next generation of file sharing systems. This study puts forth a blockchain-based framework for secure file sharing in a cloud environment. The proposed system harnesses the immutability of blockchain ledgers to track metadata and the versatility of IPFS to efficiently distribute content in a decentralized manner. Advanced Encryption Standard (AES) further augments confidentiality protections. This blend of technologies aims to tackle persistent challenges like single points of failure, lack of transparency, inefficient data segmentation, and threats to intellectual property security.

2. Related Works

Shetty *et. al.*, [1] made a commendable proposal to integrate AES encryption and decryption techniques into a cloud file sharing system, aiming to enhance data security and privacy. It's crucial to acknowledge that while AES encryption addresses critical concerns related to data confidentiality, it doesn't provide a comprehensive solution to all the challenges faced by traditional cloud file sharing systems. Many inherent challenges, including data ownership, access control, latency, and cost considerations, still persist.

Furthermore, the proposed system may introduce complexities related to key management and user experience, which require

Osuolale A. F., Alimi O. D. and Okeowo I. T. (2025). Development of a Hybrid Blockchain-Based File Sharing System in a Cloud Environment. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 14 No. 1, pp. 1 – 10.

careful attention. Therefore, while AES encryption significantly contributes to data protection, it represents just one piece of the puzzle in creating a robust and holistic solution for modern distributed sharing systems.

Sumanth *et. al*., [3] proposed Secure File Sharing system places a strong emphasis on enhancing data security within cloud-based file sharing processes. It empowers users to safeguard their files by encrypting them prior to uploading them to the cloud server. Once stored, registered users can access files and pertinent information, including file origin, title, and size. Access to a file necessitates a user-initiated request to the uploader, followed by a cryptocurrency payment transaction. Successful payment grants the user a secret key via email, allowing decryption and access to the desired file. The system's security foundation relies on the AES encryption algorithm, ensuring data confidentiality and integrity. Users engage with the system through a web-based application that mandates registration and login, unlocking features like file viewing, uploading, and encryption. File sharing requires encryption with a private key, and recipients can access shared files by providing the corresponding secret key generated during encryption.

A notable innovation in this system is its integration of blockchain technology, meticulously recording file upload and download data. Cryptocurrency is instrumental in the unique "Pay to View" feature, facilitated through smart contracts to secure fund transfers and transparently document transaction details. While this blockchain-based approach significantly enhances data security and transparency, it's important to acknowledge the ongoing challenges such as latency, scalability, and user adoption that should be addressed for a comprehensive and robust solution.

Shafieinejad *et al.,* [5] introduced a collaborative scheme for secure cloud file sharing that utilizes blockchain and attribute-based encryption (ABE), presenting an inventive method for enhancing access control and ensuring data security. However, it's important to acknowledge that this solution may encounter performance challenges compared to traditional AES encryption due to the intricacies of ABE. Attribute-based encryption involves complex cryptographic processes that can introduce computational overhead, potentially impacting system performance, especially in scenarios with large-scale file sharing and numerous access requests. While the presented proposal introduces advanced security measures and access controls, potential performance challenges and efficient data distribution issues may arise when compared to alternative approaches such as AES encryption and IPFS integration.

In the work by Kang *et al*., [4], a combination of Named Data Network (NDN) technology, cloud blockchain, and Interplanetary File System (IPFS) is proposed, outlining a blockchain-oriented approach for knowledge file storage and sharing based on NDN principles. This method utilizes NDN for file content signature and encryption, effectively segregating the security and transmission processes of the files. Simultaneously, it employs a flexible NDN reverse path forwarding and routing strategy, incorporating an IPFS private storage network to enhance the security of encrypted data storage. Ultimately, the approach leverages consensus among participating nodes and employs a synchronized blockchain for file sharing, ensuring traceability. However, it suffers from a high response time.

Motivated by identified limitations in prior works, such as Shetty *et. al*. [1] focusing on data confidentiality without addressing broader challenges, and Sumanth *et. al*. [3] acknowledging issues like latency and scalability, this research aims to provide a comprehensive solution. Shafieinejad *et. al*. [5] introduces collaboration but may face performance challenges. Kang *et. al*. [4] proposes a blockchain-based method with a high response time. To address these gaps, our research integrates AES encryption, blockchain, and IPFS, creating a holistic framework for secure and efficient cloud-based file sharing, overcoming individual limitations in prior works.

Hybrid blockchains combine elements of both public and private blockchains. They offer flexibility in terms of access control and data visibility, making them suitable for a wide range of use cases. Different parts of the blockchain can have varying levels of access control, allowing for public and private segments within the same network. Hybrid blockchains are used when organizations require the benefits of public blockchains, like decentralization and transparency, while also needing some level of privacy and control over certain data or processes.

The InterPlanetary File System (IPFS), as emphasized by Athreya *et al.*, [2], strives to create a decentralized and revolutionary peer-to-peer distributed file system with the goal of integrating computing devices within a common file system. Developed by Joan Bennett in 2015 and managed by Protocol Labs, IPFS operates through client software on participant computers, functioning as nodes that store and retrieve various file types. Unlike HTTP, IPFS identifies data by content, generating a unique hash as a file's identifier. This hash ensures data integrity, allowing efficient retrieval and verification. IPFS divides files into hashed chunks distributed across nodes, forming a resilient and redundant network that enhances data availability and reduces the risk of loss.

Suman *et al.*, [10] emphasizes IPFS's content-based addressing, guaranteeing file integrity through cryptographic hashes and thwarting unauthorized modifications. IPFS's decentralized and efficient approach to file storage and sharing, based on content identification and distributed redundancy, positions it as a reliable protocol for the proposed scheme, contributing to overall robustness and fault tolerance.

AES, also known as Rijndael, is a widely adopted symmetric encryption algorithm used to protect the confidentiality and integrity of data. Established by the U.S. National Institute of Standards and Technology (NIST) in 2001, AES has evolved into a crucial tool for safeguarding sensitive information across diverse industries. This symmetric block cipher algorithm operates on 128-bit blocks, employing keys of 128, 192, and 256 bits to convert these individual blocks. Following encryption, the blocks are amalgamated to create the cipher text. AES is grounded in a substitution-permutation network, also referred to as an SP network, comprising interconnected operations such as input replacements with designated outputs (substitutions) and bit shuffling processes (permutations). Every cipher operates on data in 128-bit blocks, utilizing cryptographic keys of 128, 192, and 256 bits for encryption and decryption. Symmetric, or secret key, ciphers employ a shared key for both encrypting and decrypting data, requiring both the sender and the receiver to possess and utilize the identical secret key.

The system leverages blockchain-backed access controls, allowing users to assert ownership and fine-tuned access permissions for file sharing. This shift empowers users to dictate who can access their data and on what terms, reinforcing their authority over data ownership and control It also safeguard their files by encrypting them prior to uploading them to the cloud server.

3. Methodology

The system architecture as shown in Figure 1 is composed of four components namely: Hybrid blockchain technology, Advanced encryption standards (AES), and the InterPlanetary File System (IPFS) network.

The blockchain network serves as the cornerstone, leveraging mathematical consensus algorithms to validate and confirm transactions. The mining probability that quantifies the likelihood of successfully mining a block, ensuring decentralized transaction approval is stated below in equation 1:

$$P(\text{mining}) = \frac{MP}{TMP} \tag{1}$$

Where MP is the Mining Power and TMP is the Total Mining Power.
For each file file_i in the system, essential metadata is maintained within the blockchain. This metadata includes:

a. Owner(file_i): Representing the owner of the file.
b. Timestamp(file_i)**:** Denoting the time of file upload.
c. AccessPermissions(file_i)***:*** Specifying access privileges (e.g., read-only, read-write).
d. Hash(file_i)**:** Calculating the cryptographic hash of the file's content.
e. TransactionHistory(file_i): Tracking file-related transactions.

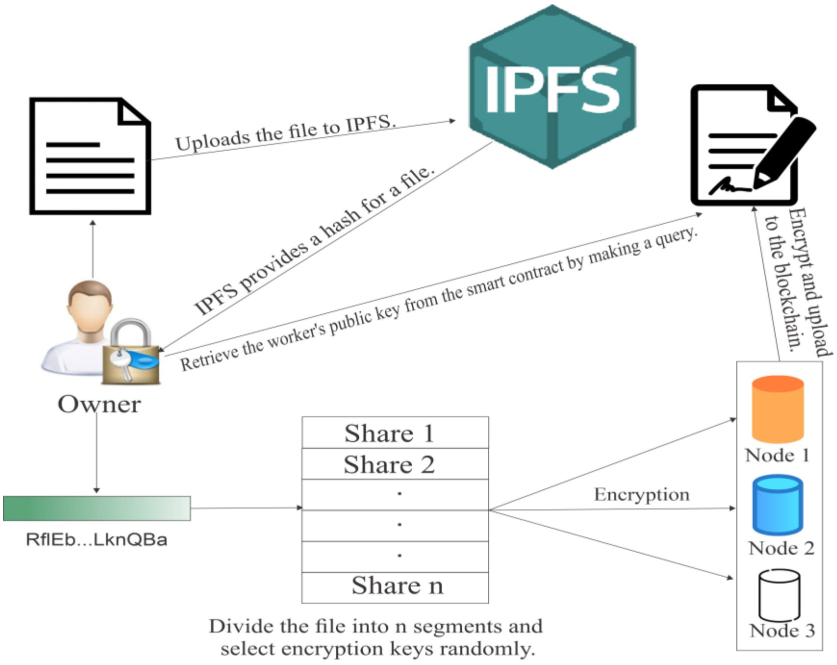

IPFS
Uploads the file to IPFS.
IPFS provides a hash for a file.
Retrieve the worker's public key from the smart contract by making a query.
Encrypt and upload to the blockchain.
Owner
Share 1
Share 2
.
.
.
Share n
RflEb...LknQBa
Encryption
Node 1
Node 2
Node 3
Divide the file into n segments and select encryption keys randomly.

Figure 1: Architecture of a blockchain file sharing system

These attributes are quantified as follows:

a. Owner(file_i) ∈ O
b. Timestamp(file_i) ∈ T
c. AccessPermissions(file_i) ∈ {read-only, read-write}
d. Hash(file_i) = HashFunction(Content(file_i))
e. TransactionHistory(file_i) = [transaction_1, transaction_2, ..., transaction_K]

In this context, the term "Owner(file_i)" refers to the owner of a specific file, denoted as "file_i," which is a part of the larger set of all owners, denoted as "O." Similarly, "Timestamp(file_i)" pertains to the timestamp associated with the file, drawn from the larger set of all timestamps, represented as "T." The concept of "AccessPermissions(file_i)" is limited to permissible access modes. The "Hash(file_i)" is derived by applying a dedicated hash function to the content of "file_i."

Lastly, the "TransactionHistory(file_i)" is depicted as a list of transactions involving the particular file. The system's security is fortified through the integration of an impregnable Advanced Encryption Standard (AES) encryption system. Files are encrypted using the AES encryption standard, with encryption and decryption functions formulated as:

$$\text{AES\_Encrypt}(\text{file\_i}, \text{Key}(\text{file\_i})) = \text{EncryptedContent}(\text{file\_i}) \quad (2)$$

$$\text{AES\_Decrypt}(\text{file\_i}, \text{Key}(\text{file\_i})) = \quad (3)$$

$$\text{DecryptedContent}(\text{EncryptedContent}(\text{file\_i})$$

Equation 2 represents the function that encrypts file_i using a unique encryption key and produces “EncryptedContent(file_i)” while equation 3 represents the second function that decrypts file_i using it’s unique encryption key thereby producing

“DecryptedContent(EncryptedContent(file_i))”. Complementing the blockchain's metadata capabilities, the InterPlanetary File System (IPFS) network ushers in a new era of file storage. Employing cryptographic hash functions, files are chunked, forming an integral facet of the system's mathematical framework. These content-addressable chunks, systematically distributed across a sprawling network of nodes, are rapidly retrieved through hash references, underpinned by sophisticated data structures. Advanced algorithms orchestrate the distribution and replication of chunks, ensuring heightened data availability and reliability. The IPFS distribution for a file file_i is determined by the algorithm presented in equation 4.

$$\text{IPFS\_Distribution}(\text{file\_i}) = \text{DistributionAlgorithm}(\text{file\_i}) \quad (4)$$

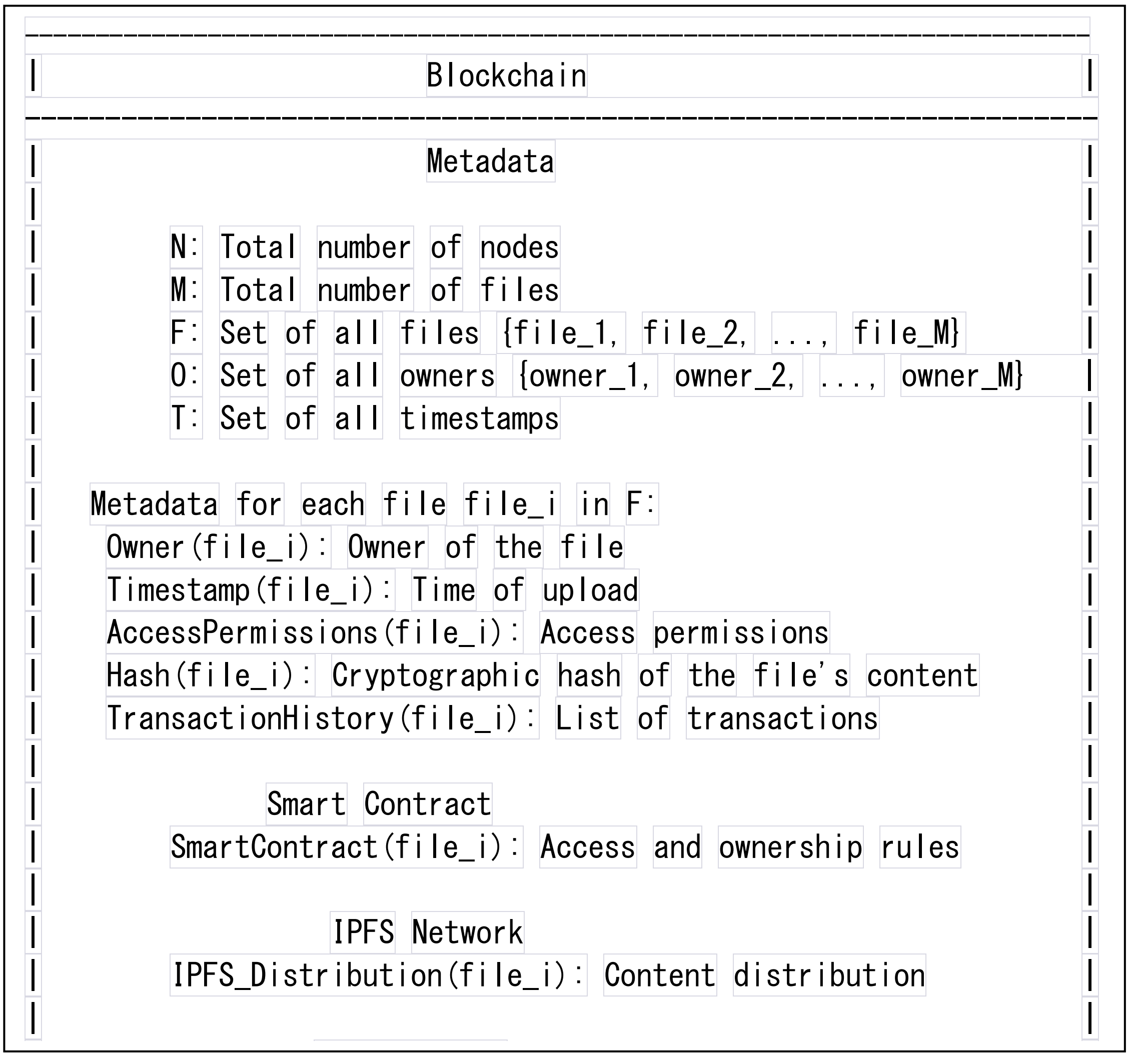
To assess the overall system security, cryptographic models are employed. These models encompass a collection of cryptographic algorithms and protocols, including hash functions, public-key cryptography, and other relevant security mechanisms. These cryptographic models are shown in equation 5.

$$\text{Cryptographic\_Models} = \{\text{hash\_functions}, \text{public\_key\_cryptography} \dots \tag{5}$$

Figure 2 presents a mathematical model of the system.

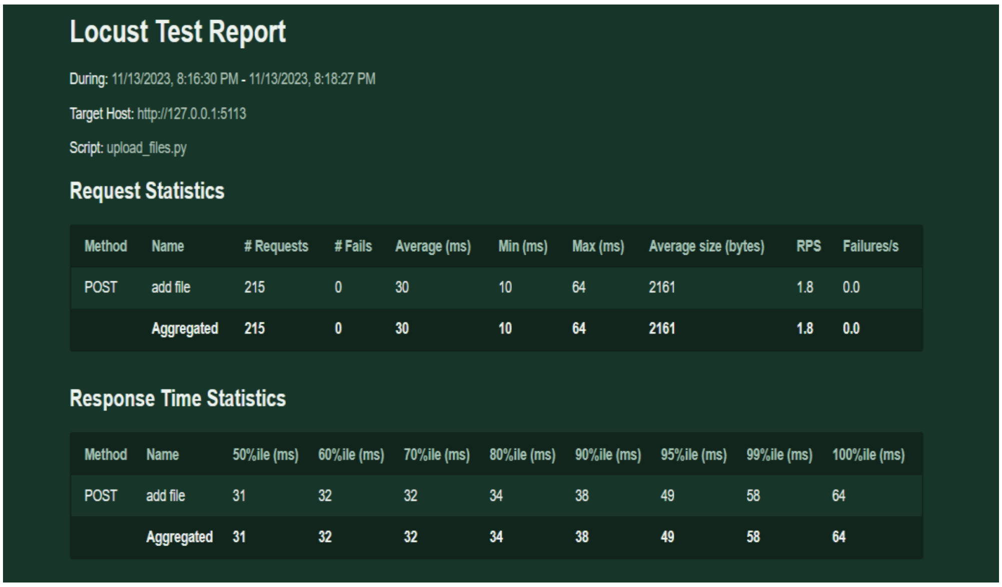
4. Results and Discussion

The results obtained from the simulation of file upload and download processes demonstrate the efficiency and robustness of the developed hybrid blockchain-based file-sharing system. The simulation carried out using the python locust package and ubuntu 23.04, reveals an impressive performance during the file upload process. With no failures among 215 concurrent upload activities, the server averaged 30ms response time. The slowest request responded within 64ms as shown in Figure 3, showcasing consistent and swift handling of incoming requests. Overall, this test shows a well optimized model ready to handle significant load.



```
-----------------------------------------------------------------------
|                          Blockchain                                 |
-----------------------------------------------------------------------
|                          Metadata                                   |
|                                                                     |
|        N: Total number of nodes                                     |
|        M: Total number of files                                     |
|        F: Set of all files {file_1, file_2, ..., file_M}            |
|        O: Set of all owners {owner_1, owner_2, ..., owner_M}        |
|        T: Set of all timestamps                                     |
|                                                                     |
|   Metadata for each file file_i in F:                               |
|    Owner(file_i): Owner of the file                                 |
|    Timestamp(file_i): Time of upload                                |
|    AccessPermissions(file_i): Access permissions                    |
|    Hash(file_i): Cryptographic hash of the file's content           |
|    TransactionHistory(file_i): List of transactions                 |
|                                                                     |
|              Smart Contract                                         |
|        SmartContract(file_i): Access and ownership rules            |
|                                                                     |
|                  IPFS Network                                       |
|        IPFS_Distribution(file_i): Content distribution              |
|                                                                     |
```

Figure 2: Mathematical Representation of the system



Locust Test Report

During: 11/13/2023, 8:16:30 PM - 11/13/2023, 8:18:27 PM

Target Host: http://127.0.0.1:5113

Script: upload_files.py

Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
POST	add file	215	0	30	10	64	2161	1.8	0.0
	Aggregated	**215**	**0**	**30**	**10**	**64**	**2161**	**1.8**	**0.0**

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
POST	add file	31	32	32	34	38	49	58	64
	Aggregated	**31**	**32**	**32**	**34**	**38**	**49**	**58**	**64**

Figure 3: Request and Response time statistic

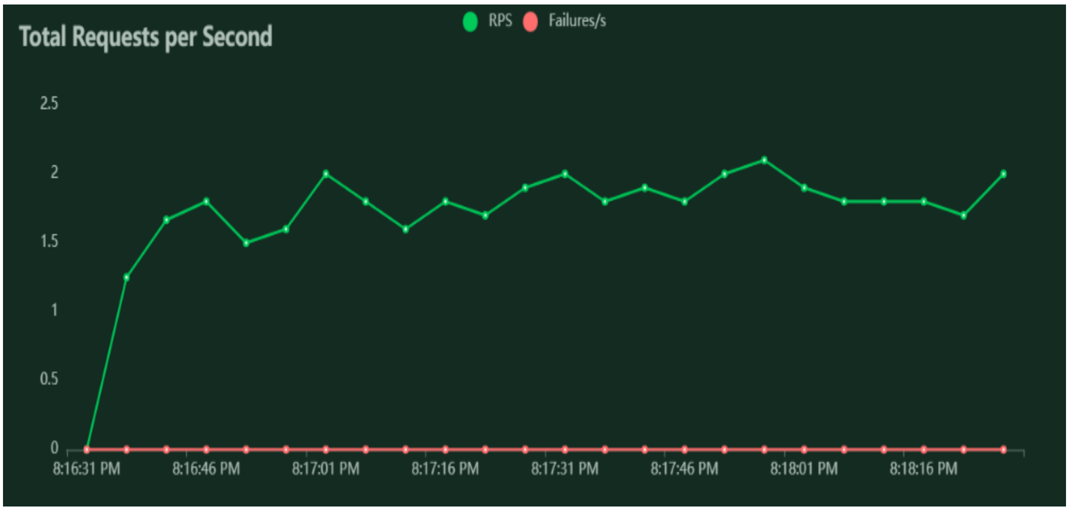


Figure 4: Total request per second chart

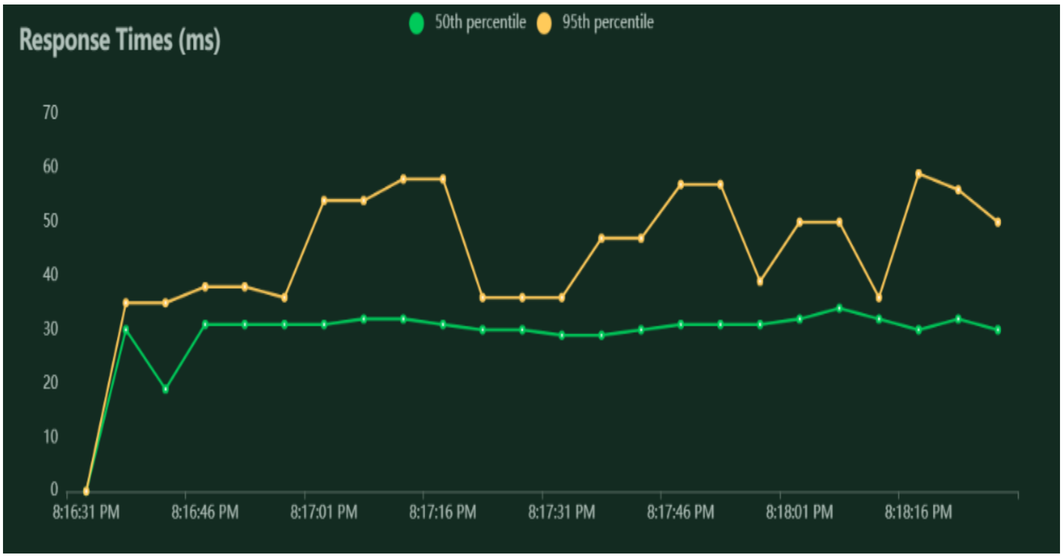
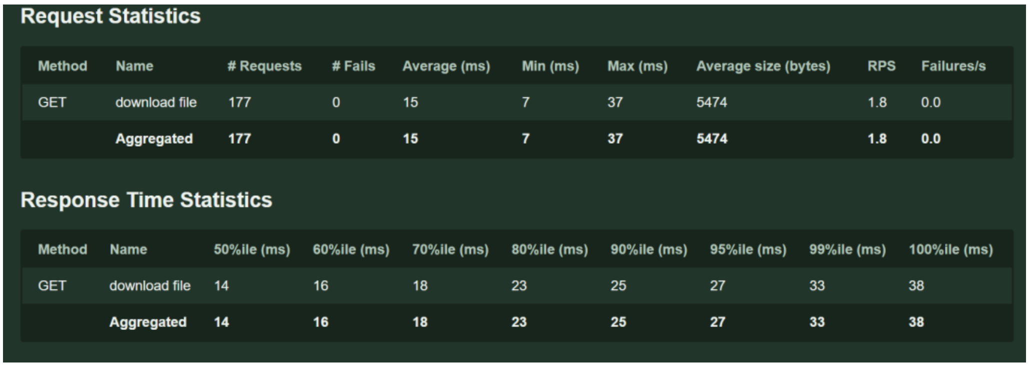


Figure 6: Total response time chart

Figure 3 illustrates the chart for the total requests per second made during this file upload process, with a maximum value capped at 2 RPS. Figure 4 shows the chart for the total response time. The response times are relatively stable throughout the graph, with the 50th percentile response time hovering around 30 milliseconds and the 95th percentile response time hovering around 60 milliseconds. There are a few spikes in the response times, but they are all relatively small and short-lived.

Another simulation was carried out during the file download process. Different file types and sizes were downloaded. Figure 5 shows that 177 download file requests were made, all of which were successful. The average response time was 15 milliseconds, with a minimum of 7 milliseconds and a maximum of 37 milliseconds. The average request size was 5474 bytes. There were 18 requests per second, with a failure rate of 0%.

Figure 6 illustrates the chart for the total requests per second made during this file download process, with a maximum value capped at 2 RPS.



Request Statistics

Method	Name	# Requests	# Fails	Average (ms)	Min (ms)	Max (ms)	Average size (bytes)	RPS	Failures/s
GET	download file	177	0	15	7	37	5474	1.8	0.0
	Aggregated	**177**	**0**	**15**	**7**	**37**	**5474**	**1.8**	**0.0**

Response Time Statistics

Method	Name	50%ile (ms)	60%ile (ms)	70%ile (ms)	80%ile (ms)	90%ile (ms)	95%ile (ms)	99%ile (ms)	100%ile (ms)
GET	download file	14	16	18	23	25	27	33	38
	Aggregated	**14**	**16**	**18**	**23**	**25**	**27**	**33**	**38**

Figure 6: Request and response time statistics for downloading

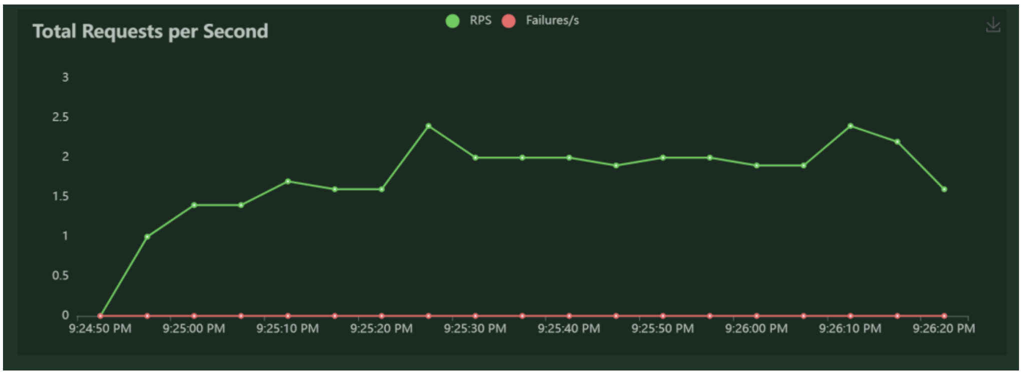


Figure 7: Total request per second chart

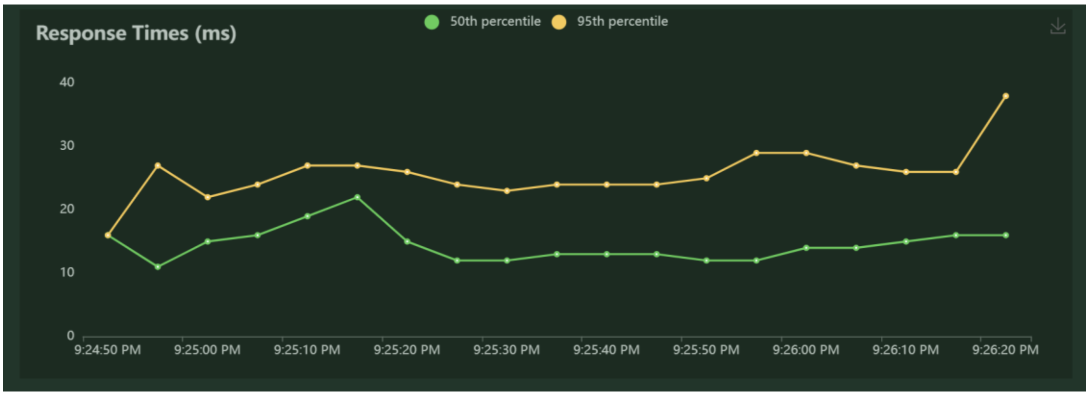


Figure 8: Total response time chart

Figure 7 shows the chart for the total response time made in this file download phase which shows a maximum response time of 40ms and minimum of 15ms. A total number of 10 users were available.

The integration of blockchain technology enhances system performance and security by ensuring data integrity, preventing unauthorized modifications, and enabling secure peer-to-peer file exchanges through a decentralized approach. Unlike traditional file-sharing systems vulnerable to single points of failure, blockchain mitigates these risks by distributing file verification across multiple nodes. The use of smart contracts further reduces latency and improves efficiency.

Academically, this work contributes to research on blockchain-based distributed systems, offering empirical evidence of improved file-sharing efficiency, security, and reliability. Societally, it benefits sectors like finance, healthcare, and law by providing a tamper-proof, scalable file-sharing model that ensures secure data exchange, which can also be extended to cloud storage services.

In order to assess the efficiency of the developed model in comparison to a conventional system, performance evaluation was conducted considering the approach presented by Kang *et al*. (2022), wherein the author advocated for an NDN Network without file encryption.

The information in Tables 1 and 2 show that the developed model has a lower average response time, min response time, and max response time than the existing model.

It also has a slightly higher Request per Second (RPS). This indicates that the developed model is performing better than the existing model under load.

Figure 8 to Figure 11 further show this information in graphs.

Tables 1 and 2 show the detailed comparison of the two models.

Table 1: Developed model versus Existing

Statistic	**Developed Model**	**Existing Model**
Average Response Time(ms)	30	35
Min Response Time (ms)	10	15
Max Response Time (ms)	64	70
RPS	2	2.7
Failures/s	0	0

Table 2: Developed model versus Existing Model Statistics during file download

Statistic	**Developed Model**	**Existing Model**
Average Response Time (ms)	15	35
Min Response Time (ms)	7	12
Max Response Time (ms)	37	65
RPS	2	2.4
Failures/s	0	0

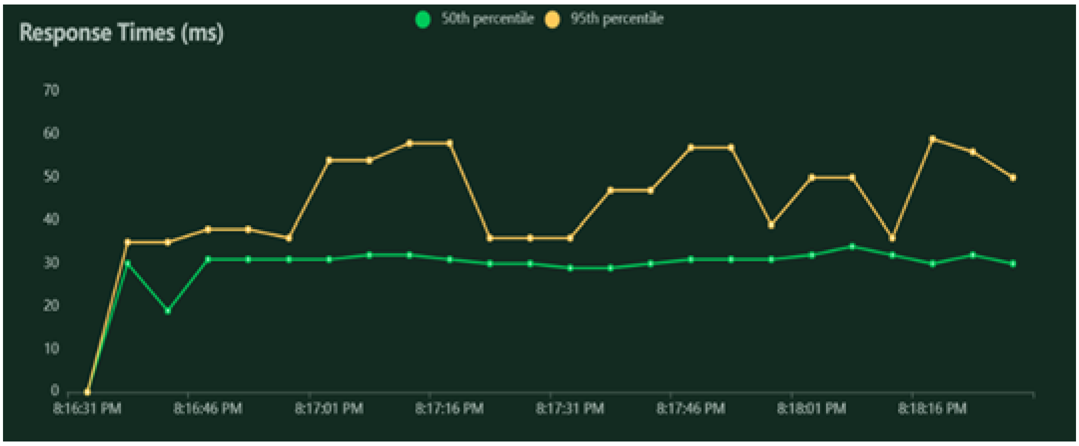


Figure 9: Response Time Chart for file upload in Developed Model

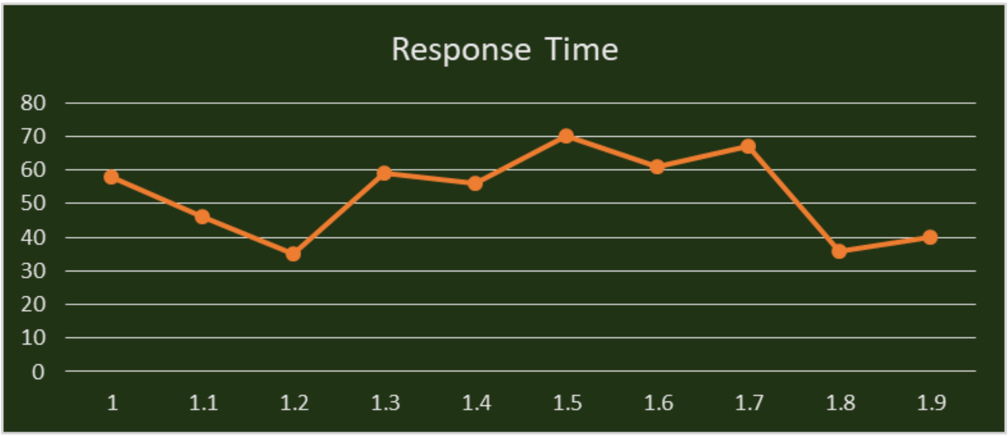


Figure 10: Response Time Chart for file upload in Existing Model

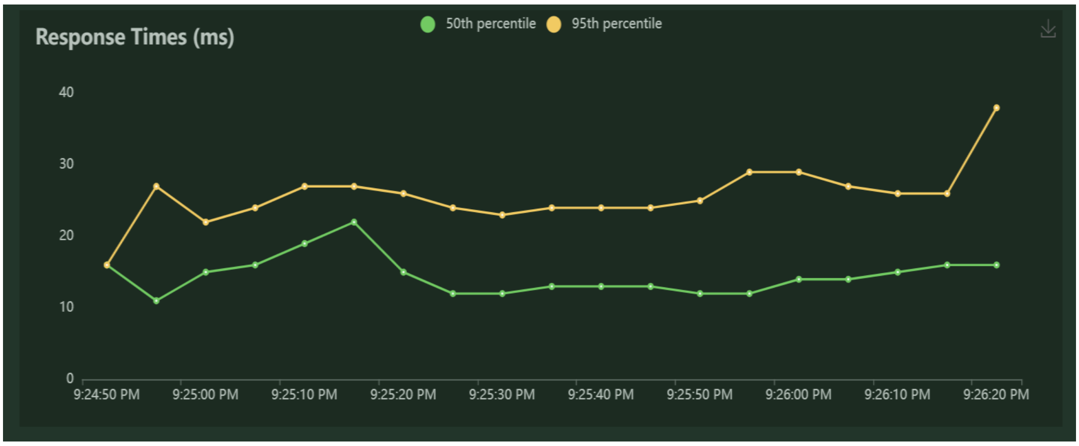


Figure 11: Response Time Chart for file download in Developed Model

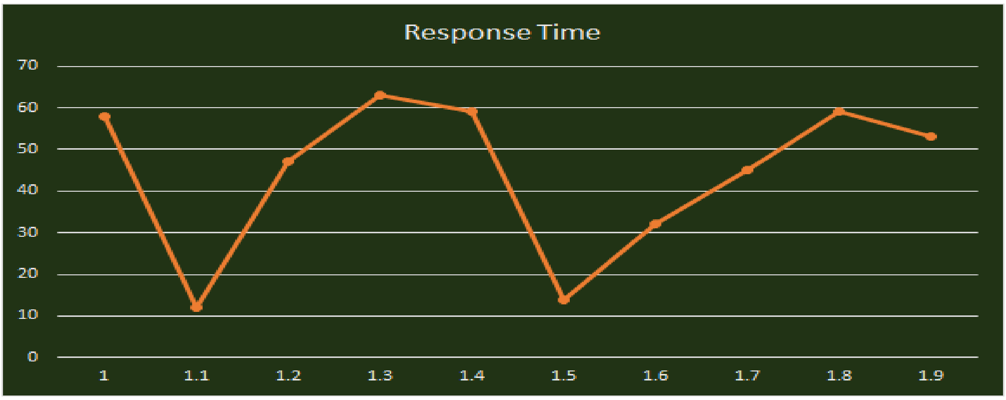


Figure 12: Response Time Chart for file upload in Existing Model

5. Conclusion

This study extensively explores the concepts of Blockchain, Interplanetary File System, and cloud file sharing. The integration of blockchain technology and the Interplanetary File System (IPFS) in a distributed file-sharing system signifies a notable advancement in addressing challenges related to security, transparency, and decentralization. The immutable ledger of the blockchain guarantees transaction and user interaction integrity, creating a trustless environment where participants can securely share files without depending on a central authority. The use of IPFS further enhances this system by providing a decentralized and peer-to-peer file storage mechanism, breaking away from the limitations of traditional centralized cloud storage solutions.

References

[1] Shetty, P., Sunanda, R., K S, S. S., N G., S. N., & Kumar, H. N. P. (2019). File Sharing in Cloud-Based Environments. International Journal of Engineering Research & Technology (IJERT), Volume 6, Issue 15, Special Issue - 2018, ICRTT - 2018 Conference Proceedings, ISSN: 2278- 0181.

[2] Athreya, A., Kumar, A., Nagarajath, S., H L, Gururaj, Kumar, V., D N, Sachin, & K R, Rakesh. (2021). Peer-to-Peer Distributed Storage Using InterPlanetary File System. In Proceedings of the 2021 International Conference on Computing and Communications (ICCC) (pp. 559-567). Springer. https://doi.org/10.1007/978-981-15-3514-7_54

[3] Sumanth, P., Chowdary, P.P., Bharani, P., Krishna, T.G., & Bojjagani, S. (2022). Blockchain-Aided Cloud File Sharing. EasyChair Preprint no. 8078.

[4] Kang, P., Yang, W., & Zheng, J. (2022). Blockchain-Based Private File Storage and Sharing Using IPFS. *Sensors*, *22*(14), 5100. https://doi.org/10.3390/s22145100. Accessed JuFebruary 2025.

[5] Shafieinejad, A., & Almasian, M. (2022). A Secure Framework for Cloud File Sharing Using Blockchain and Attribute-Based Encryption. SSRN. http://dx.doi.org/10.2139/ssrn.4252101. Accessed February, 2025.

[6] Labazova, O., Dehling, T., and Sunyaev, A. (2019). Navigating the Landscape: A Taxonomy of Blockchain Applications. In Proceedings of the 52nd Hawaii International Conference on System Science.

[7] Liu, Y., Gong, W., & Fan, W. (2018). Implementing AES and RSA Hybrid Algorithm for E-Mail Security. In 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS), 701–3. IEEE. doi.org/10.1109/ICIS.2018.8466380.

[8] Mohd Javaid, A. Haleem, R. P. Singh, R. Suman, and S. Khan (2022). A review of Blockchain Technology applications for financial services. BenchCouncil Transactions on Benchmarks, Standards and Evaluations, vol. 2, no. 3, p. 100073, 2022, ISSN: 2772-4859. https://doi.org/10.1016/j.tbench.2022.100073.

[9] Mann, S., Chaudhary, H., Khatri, A., Malik, R., & Gupta, Y. (2022). Decentralized Peer-to-Peer File Storage System Using Blockchain and Interplanetary File System. International Journal of Current Science Research and Review, 05(02), 582-589. DOI: 10.47191/ijcsrr/V5-i2-34.