iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.81648

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

A Scalable Blockchain-Driven Storage Network Architecture

Prof. Pradnya Kasture[1], Aditya Shirsath[2], Vivek Biradar[3], Roshan Wazare[4], Akash Shinde[5]
Department-Computer Engineering, RMD Sinhgad School of Engineering

Abstract: ***In the modern digital era, centralized cloud storage platforms suffer from multiple limitations, including dependency on a single authority, vulnerability to cyberattacks, and unexpected service disruptions. To overcome these drawbacks, this paper introduces a Decentralized Storage Network (DSN) powered by blockchain technology.***
The proposed framework integrates a custom-built blockchain, peer-to-peer communication, and advanced encryption mechanisms to deliver a highly secure and resilient storage solution. Instead of storing files in a single location, data is encrypted and divided into smaller fragments, which are distributed across multiple nodes using an IPFS-inspired system.
All activities such as file storage, access permissions, and metadata handling are securely recorded on a blockchain ledger. This ensures data authenticity, confidentiality, and availability while eliminating reliance on centralized systems.
Keywords: Blockchain, Distributed Storage, IPFS, Smart Contracts, Peer-to-Peer Systems, Data Security.

I. INTRODUCTION

Conventional cloud storage architectures rely heavily on centralized infrastructure, which introduces several critical issues such as single points of failure, data privacy concerns, and lack of transparency. Users must trust third-party providers to manage and secure their sensitive data, which is not always reliable.
To address these challenges, this work aims to design a decentralized and user-centric storage platform. By utilizing blockchain and distributed networking, the system ensures improved trust, reliability, and resistance to failures.

A. Objectives of the Study

The primary objective of this study is to develop a distributed storage system that eliminates dependency on centralized servers by leveraging a peer-to-peer architecture, thereby improving reliability and fault tolerance. The system aims to ensure data confidentiality and integrity through the use of strong encryption and validation techniques, protecting sensitive information from unauthorized access and tampering. Additionally, it focuses on maintaining transparency and trust by recording all system operations on an immutable blockchain ledger, which provides secure and verifiable transaction history. This paper further presents a detailed explanation of the system design, overall architecture, and the operational modules that collectively contribute to the effective functioning of the proposed solution.

II. RELATED WORK

Blockchain-based decentralized storage has been extensively explored in recent research. Several studies have contributed significantly to this domain. S. Mann et al. proposed a decentralized file storage system that integrates Ethereum, IPFS, and encryption techniques to ensure secure and distributed data management. G. Richa Shalom and G. Rohit Nirogi focused on enhancing security and privacy in cloud storage environments by leveraging blockchain technology. Yan Zhu and colleagues investigated the use of unused global storage resources to build efficient decentralized storage systems. Furthermore, Lu Meng and Bin Sun combined Hyperledger Fabric with IPFS to improve data reliability and mitigate risks such as data loss and tampering. In addition, Ganesh J et al. developed a user-friendly decentralized storage framework that emphasizes security, accessibility, and ease of use.

A. Research Gap

Despite the progress made in this field, most existing systems rely either on public blockchain platforms, such as Ethereum, or on complex enterprise frameworks like Hyperledger. These approaches often introduce limitations related to scalability, cost, and system complexity. To address these challenges, the proposed approach introduces a custom lightweight blockchain that utilizes an efficient consensus mechanism, such as Proof-of-Authority (PoA). This design is specifically optimized for storage operations, providing improved performance, reduced overhead, and greater flexibility compared to existing solutions.

III. PROPOSED SYSTEM ARCHITECTURE

The proposed model combines blockchain technology with a distributed file storage system. The blockchain manages metadata and permissions, while the actual data is stored across a peer-to-peer network.
The overall system architecture is shown in Fig. 1

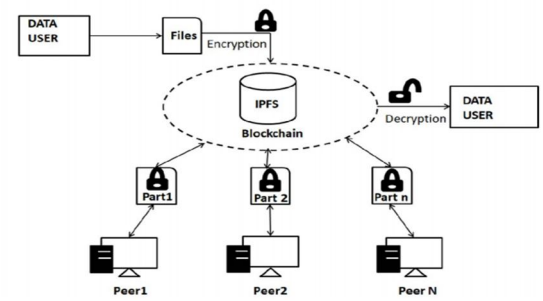


Fig. 1. Proposed System Architecture

A. Working Mechanism

The process, as shown in Fig. 1, follows a secure flow in which a user initially uploads a file into the system. Before transmission, the file is encrypted on the client side to ensure confidentiality. The encrypted data is then divided into multiple segments, which are distributed across different nodes in the network to enhance security and availability. The blockchain records essential information including file identifiers (hashes), metadata, and access permissions, thereby ensuring integrity and traceability. When a user requests a file, the system retrieves the corresponding segments from various nodes, reassembles them, and performs decryption locally to reconstruct the original file for authorized access.

B. System Modules

The architecture consists of several key components working together to provide a secure and efficient system. The User Interface Module handles authentication and file-related operations such as upload and download. The Encryption and Storage Module utilizes AES-256 encryption and manages distributed storage of data. The Blockchain Module maintains records of transactions and metadata, ensuring transparency and immutability. The Consensus Mechanism employs Proof-of-Authority to validate transactions efficiently with minimal overhead. The Access Control Module uses smart contracts to enforce permissions and ownership rules. The P2P Network Module is responsible for storing file fragments across distributed nodes, while the Data Retrieval Module collects, reconstructs, and decrypts data for authorized users.

C. Data Flow Description

The workflow of the system involves encrypting the data, splitting it into smaller chunks, storing these chunks in IPFS, and recording their corresponding hash references on the blockchain. In this architecture, the blockchain functions as a trust layer by maintaining secure records and enforcing access control, whereas IPFS ensures efficient and scalable distributed storage of data across the network.

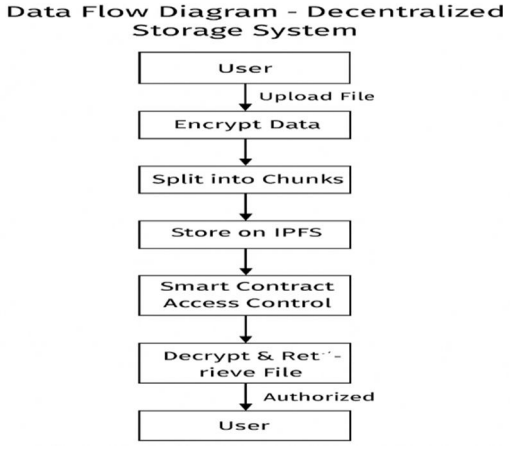


Fig. 2. Data Flow Diagram

The relationships between the system components are defined in the Class Diagram (Fig. 3). It shows the User, File, Encryption Module, IPFS Node, Blockchain, and Smart Contract, and their interactions, such as uploads, encrypts File, stores Chunks, and records Metadata.

D. System Class Design

The proposed blockchain-based decentralized storage system is structured using multiple interconnected components, each responsible for a specific functionality within the architecture. The system begins with the User entity, which represents an authenticated participant capable of uploading and requesting files. The user interacts with the system through operations such as file upload and retrieval requests.

When a file is uploaded, it is represented by the File entity, which contains attributes such as file identifier, file name, file hash, and encryption key. Before storage, the file is processed by the Encryption Module, which is responsible for securing the data through encryption and enabling decryption during retrieval. This ensures that sensitive information remains protected throughout its lifecycle.

Once encrypted, the file is divided into smaller chunks and distributed across the network using the IPFS Node component. The IPFS node handles the storage and retrieval of file chunks in a decentralized manner. It interacts with multiple nodes in the Peer-to-Peer (P2P) Network, which facilitates data sharing and node connectivity. This distributed approach enhances fault tolerance and data availability.

The Blockchain component plays a critical role in maintaining system integrity by recording metadata related to file storage, such as content identifiers and transaction details. It ensures immutability and transparency of all operations. The blockchain also interacts with the Consensus Module, which validates transactions and ensures agreement among network participants using consensus mechanisms such as Proof-of-Authority.

Access control within the system is managed through Smart Contracts, which define and enforce rules for granting and revoking access permissions. These smart contracts operate in conjunction with the Access Control module, which authenticates users and authorizes access based on predefined policies. This combination ensures secure and controlled data sharing among users.

Finally, the Retrieval Module is responsible for reconstructing the original file when requested by an authorized user. It fetches the encrypted chunks from the IPFS nodes, reassembles them, and passes them through the decryption process to restore the original content. This module ensures efficient and secure data retrieval while maintaining system integrity.

E. File Upload Process

The file upload process is detailed in the sequence diagram. The User initiates an "Upload File" action. The Encryption Service encrypts the file, stores the chunks on IPFS, and receives a file hash. This hash and metadata are then recorded on the Blockchain, which confirms the transaction, notifying the user of success.

The file upload process in the proposed decentralized storage system follows a structured sequence of interactions among the User, Encryption Service, IPFS network, and Blockchain components. Initially, the process is triggered when the user initiates a file upload request to the system. Upon receiving this request, the Encryption Service takes responsibility for securing the file by performing client-side encryption. This ensures that the data remains confidential before being transmitted to any external storage system.

After encryption, the file is divided into smaller chunks and forwarded to the IPFS network for decentralized storage. The IPFS component stores these encrypted chunks across distributed nodes and generates a unique content identifier, commonly referred to as a file hash. This hash acts as a reference for locating and retrieving the stored data in the future. The generated file hash is then returned to the Encryption Service.

Subsequently, the Encryption Service communicates with the Blockchain component to record essential metadata associated with the uploaded file. This metadata includes the file hash and other relevant transaction details. The blockchain ensures that this information is stored in an immutable and transparent manner, thereby guaranteeing data integrity and traceability.

Once the metadata is successfully recorded, the blockchain validates and confirms the transaction through its consensus mechanism. After confirmation, a success response is sent back through the system, ultimately notifying the user that the file upload process has been completed successfully.

This sequence ensures that the file is securely encrypted, reliably stored in a decentralized manner, and permanently recorded on the blockchain, thereby achieving data security, integrity, and availability within the system.

IV. METHODOLOGY

The proposed system follows a structured methodology to implement a secure and decentralized storage framework using blockchain technology. The process begins with user authentication, where only authorized users are allowed to interact with the system. Once authenticated, the user uploads a file, which is immediately processed by the client-side encryption module. The file is encrypted using a strong cryptographic algorithm such as AES-256 to ensure confidentiality before transmission.

After encryption, the file is divided into smaller chunks to enable distributed storage. These chunks are then transmitted to the InterPlanetary File System (IPFS), where they are stored across multiple peer nodes in a decentralized manner. Each stored chunk generates a unique content identifier (CID), which acts as a reference for retrieval.

The metadata associated with the file, including the hash and access permissions, is recorded on a custom blockchain. The blockchain ensures immutability and transparency of all transactions. A Proof-of-Authority (PoA) consensus mechanism is used to validate transactions efficiently, reducing computational overhead and improving system performance.

Smart contracts are deployed on the blockchain to manage access control. These contracts define the rules for granting and revoking access to stored files. When a user requests a file, the system verifies access permissions through the smart contract. Upon successful verification, the encrypted chunks are retrieved from IPFS, reassembled, and decrypted at the client side to reconstruct the original file.

This methodology ensures data security, fault tolerance, and transparency by combining encryption, decentralized storage, and blockchain-based validation.

V. SYSTEM REQUIREMENT

The proposed decentralized storage system requires a reliable computing environment capable of supporting blockchain operations, peer-to-peer networking, and cryptographic processing. The system must handle file encryption, chunking, distributed storage, and blockchain transaction validation efficiently. Additionally, it should support network communication between nodes and ensure secure interaction between system modules such as the blockchain, IPFS, and user interface. The system must also provide sufficient computational resources to execute consensus algorithms and smart contracts without significant latency. Furthermore, a stable internet connection is essential to maintain connectivity between distributed nodes and ensure seamless data transfer across the network.

A. Hardware Requirements

The hardware requirements for implementing the proposed system are moderate and can be fulfilled using standard computing devices. A system with a multi-core processor, such as an Intel Core i5 or equivalent, is sufficient to handle encryption, blockchain processing, and network operations. A minimum of 8 GB RAM is recommended to support concurrent processes, including node communication and file handling tasks. Additionally, adequate storage capacity is required to store temporary encrypted files, blockchain data, and system logs. Network hardware capable of maintaining stable and high-speed internet connectivity is also necessary to ensure efficient communication between distributed nodes in the peer-to-peer network.

B. Software Requirements

The software requirements include a development environment capable of supporting blockchain and distributed system implementation. The system is developed using .NET and C# for building the blockchain and application logic. IPFS is used as the decentralized storage platform for managing file chunks across peer nodes. A suitable operating system such as Windows or Linux is required to run the application and supporting services. Cryptographic libraries are necessary for implementing AES-256 encryption and decryption mechanisms. Additionally, tools for blockchain development and testing, such as local blockchain simulators or frameworks, are required to deploy and test smart contracts and consensus mechanisms. The system also requires networking libraries to enable peer-to-peer communication and APIs to integrate different modules such as the user interface, blockchain, and storage layer.

VI. EXPERIMENTAL RESULTS

To validate our proposed architecture, we developed a prototype and conducted several experiments to measure its performance.

A. Implementation Details

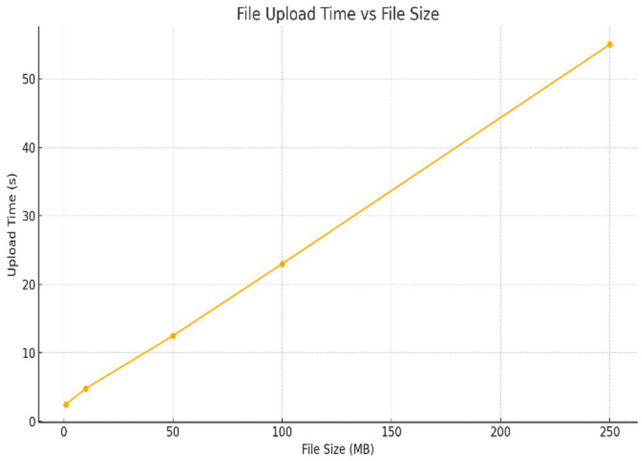
The prototype was developed on a system with the following specifications: (e.g., Intel Core i5, 8GB RAM). The custom blockchain and peer nodes were developed using .NET and C# and a Proof-of-Authority consensus mechanism. The client-side application handles AES-256 encryption, and data is stored on the IPFS network.

B. Performance Evaluation

We conducted experiments to measure the time taken for the two key operations of the system: file upload and file download. We tested with various file sizes (1 MB, 10 MB, 50 MB, 100 MB, and 250 MB) to observe the system's scalability. Each test was repeated five times, and the average time was recorded.

Fig. 5 illustrates the performance of the file upload process. This time includes client-side encryption, splitting the file into chunks, storing the chunks on the IPFS network, and finally, recording the file's metadata and hash in a transaction on our custom blockchain.

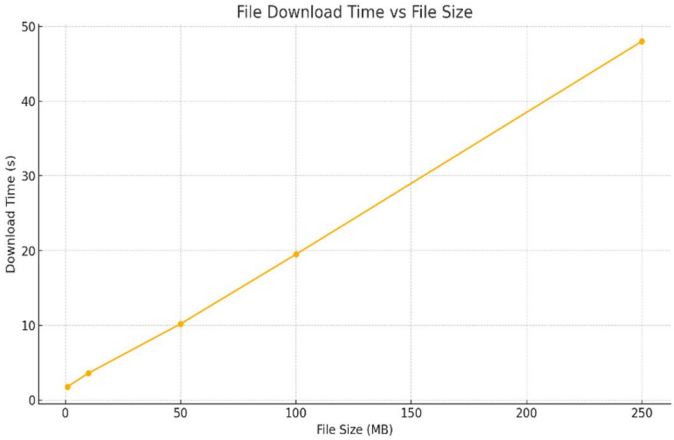
Fig. 5. File Upload Time vs File Size



As shown in Fig. 5, the upload time scales in a near-linear fashion with the increase in file size. This is an expected and desirable outcome, indicating that the system's performance does not degrade unexpectedly with larger files. We observed a small, consistent overhead of approximately **[e.g., 1.8 seconds]** on each transaction, which is attributed to the blockchain consensus and block confirmation time. This minor delay is an acceptable trade-off for the data integrity and immutability provided by the blockchain.

Fig. 6 illustrates the performance of the file download process. This time includes querying the smart contract to verify access, retrieving the file hash, fetching all encrypted chunks from the IPFS network, reassembling the file, and decrypting it on the client's machine.

Fig. 6. File Download Time vs File Size



Similar to the upload process, the download time shown in Fig. 6 also scales linearly with the file size. The query time to the smart contract for access verification was consistently low, averaging **[e.g., 0.6 seconds]**. The primary factor influencing download time is the P2P network speed of IPFS. The results confirm that our system provides robust security and access control with a minimal and predictable performance overhead.

VII. CONCLUSION

This paper presents a blockchain-based decentralized storage system designed to overcome the limitations of traditional cloud storage solutions. By integrating encryption, peer-to-peer networking, and a custom blockchain, the proposed system ensures data security, privacy, and transparency. The architecture eliminates reliance on centralized authorities and provides a robust mechanism for data storage and retrieval.

Experimental results demonstrate that the system achieves a balance between security and performance, with minimal overhead introduced by blockchain operations. The use of a custom blockchain and a lightweight consensus mechanism enhances efficiency and scalability.

Currently, the system utilizes a cloud-based infrastructure, specifically Microsoft Azure, for managing server-side operations and user credential storage. As part of future work, the system will be extended to incorporate a dedicated server and a custom database for securely managing user credentials. This enhancement aims to provide greater control, improved security, and reduced dependency on third-party cloud services.

Future research will also focus on optimizing network performance, improving scalability, and conducting comprehensive security analyses to further strengthen the system.

VIII. ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to all those who supported the completion of this work. We extend our thanks to our project guide and faculty members for their valuable guidance, encouragement, and continuous support throughout the development of this system. We also acknowledge our institution for providing the necessary resources and infrastructure required to carry out this research. Finally, we are grateful to our peers and colleagues for their constructive feedback and assistance, which contributed significantly to the successful completion of this project.

REFERENCES

[1] L. Meng and B. Sun, "Research on Decentralized Storage Based on a Blockchain," *Sustainability*, vol. 14, no. 20, p. 13060, Oct. 2022. [Online]. Available: https://doi.org/10.3390/su142013060

[2] G. J, P. R, S. K, and V. G, "A Secure Decentralized Data Storage Framework Using Blockchain Technology," *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*, vol. 2, no. 1, pp. 330-335, June 2022. [Online]. Available: https://doi.org/10.48175/IJARSCT-4600

[3] L. Jiang and X. Zhang, "BCOSN: A Blockchain-Based Decentralized Online Social Network," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 6, pp. 1454-1466, Dec. 2019. [Online]. Available: https://doi.org/10.1109/TCSS.2019.2941650

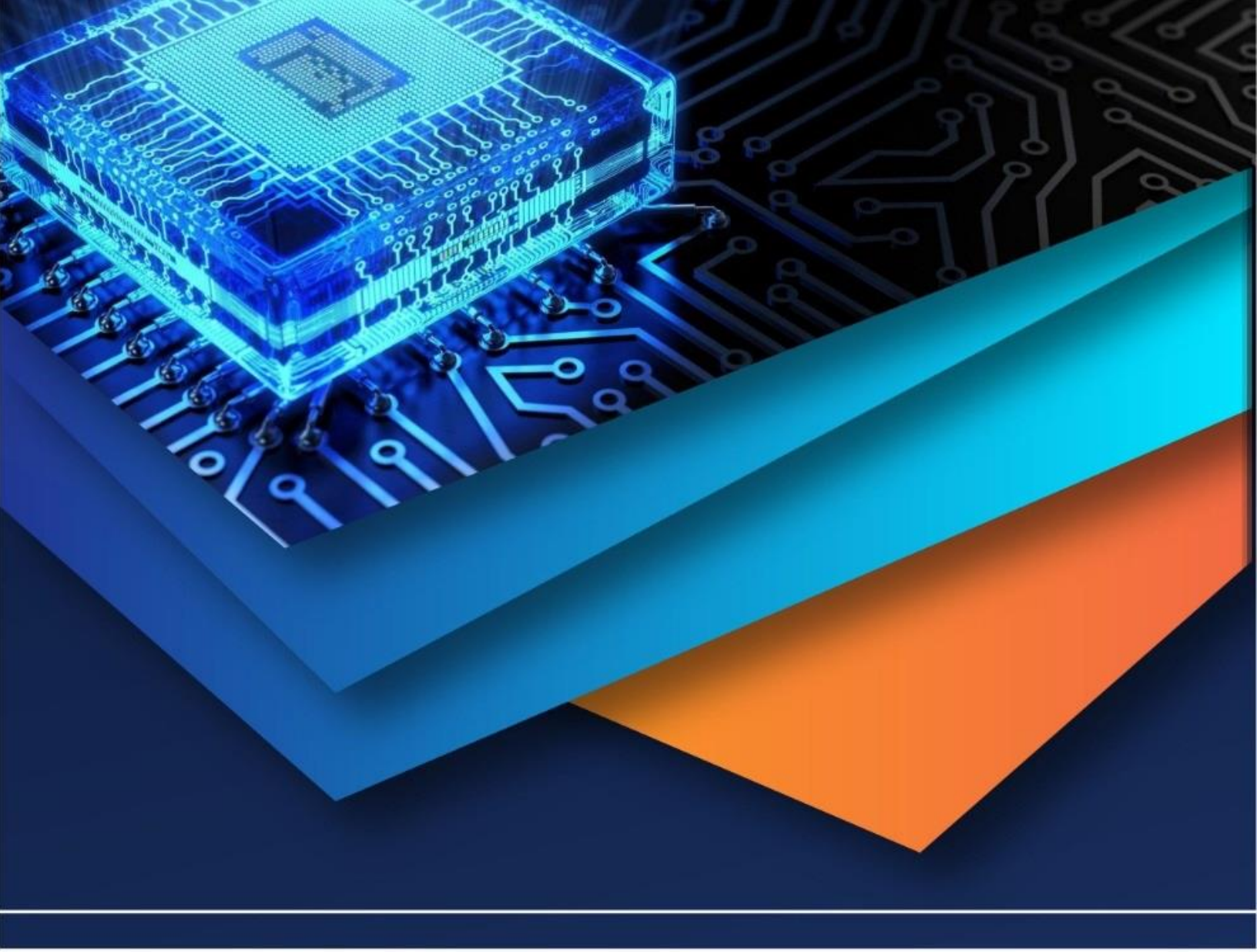
[4] Y. M. Gajmal and R. Udayakumar, "Blockchain-Based Access Control and Data Sharing Mechanism in Cloud Decentralized Storage System," *Journal of Web Engineering*, vol. 20, no. 5, pp. 1359–1388, 2021. [Online]. Available: https://doi.org/10.13052/jwe1540-9589.2054

[5] S. Mann, H. Chaudhary, A. khatri, R. Malik, and Y. Gupta, "A Peer-to-Peer File Storage System Using Blockchain and Interplanetary File System," *International Journal of Current Science Research and Review*, vol. 5, no. 2, pp. 582-589, Feb. 2022. [Online]. Available: https://doi.org/10.47191/ijcsrr/V5-i2-34

[6] G. R. Shalom and G. R. Nirogi, "Decentralized Cloud Storage Using Blockchain," *International Journal For Research in Applied Science and Engineering Technology (IJRASET)*, vol. 10, no. 9, pp. 1294-1300, Sep. 2022. [Online]. Available: https://doi.org/10.22214/ijraset.2022.46810

[7] M. I. Khalid et al., "A Comprehensive Survey on Blockchain-Based Decentralized Storage Networks," *IEEE Access*, vol. 11, pp. 10995-11015, 2023. [Online]. Available: https://doi.org/10.1109/ACCESS.2023.3240237

[8] H. Zang, H. Kim, and J. Kim, "Blockchain-Based Decentralized Storage Design for Data Confidence Over Cloud-Native Edge Infrastructure," *IEEE Access*, vol. 12, pp. 50083-50099, 2024. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3383010



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY