Decentralized File Sharing: Blockchain and Cryptography

Dr. G. Prabhakar Raju, Kadaverugu Sai Aishiu Preetham, Maradapu Ananya Sreshta, Pisati Bhanuprakash Reddy, Sarabudla Harshitha,
Department of Computer Science and Engineering, Anurag University, Hyderabad, Telangana

Abstract- In the contemporary landscape of organizational synergy, the seamless exchange of information plays a pivotal role. However, conventional centralized file-sharing infrastructures often fail to provide the requisite distributed trust and transparency necessary for secure collaboration. This paper proposes an innovative solution utilizing blockchain technology and cryptographic principles to address these challenges. By leveraging Hyperledger Fabric and the Inter Planetary File System (IPFS), the proposed system offers a robust framework for secure inter-organizational file sharing. Through meticulous implementation, it ensures confidentiality, integrity, and availability, thus significantly enhancing the efficiency and security of collaborative endeavours. This paper delineates the intricacies of the proposed methodology, providing a comprehensive roadmap for organizations seeking to augment their file-sharing capabilities in an increasingly digitized world.

INTRODUCTION

Implementing a permissioned blockchain system, like Hyperledger Fabric, helps solve two big problems when it comes to sharing files securely in organizations: making sure it's private and making sure it can handle lots of users and files without slowing down.

In a permissioned blockchain, only certain people or groups can join the network and see what's happening. This means your files stay private and only authorized people can access them. With tools like Hyperledger Fabric, you can even create secret channels for sharing files so that only the intended recipients can see them.

The issue of scalability is all about handling lots of users and files efficiently. Permissioned blockchains split up the work so that it doesn't get overwhelmed when lots of people are using it at once. With Hyperledger Fabric, you can set up different channels for different groups or types of files, so the network stays fast and responsive even when a lot is going on.

By using a permissioned blockchain like Hyperledger Fabric, organizations can share files securely, keeping them private and handling lots of users and files without any problems.

LITERATURE SURVEY

A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers.[1]

The network itself requires minimal structure. Messages are broadcast on a best-effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone. We have proposed a system for electronic transactions without relying on trust. We started with the usual framework of coins made from digital signatures, which provides strong control of ownership but is incomplete without a way to prevent double-spending. To solve this, we proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power. The network is robust in its unstructured

simplicity. Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best-effort basis. Nodes can leave and rejoin the network at will, accepting the proof-of-work chain as proof of what happened while they were gone. They vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them. Any needed rules and incentives can be enforced with this consensus mechanism.[2]

In a research community, data sharing is an essential step to gaining maximum knowledge from prior work. Existing data-sharing platforms depend on trusted third parties (TTP). Due to the involvement of TTP, such systems lack trust, transparency, security, and immutability. To overcome these issues, this paper proposed a blockchain-based secure data-sharing platform by leveraging the benefits of the interplanetary file system (IPFS). Metadata is uploaded to the IPFS server by the owner and then divided into secret shares. The proposed scheme achieves security and access control by executing the access roles written in a smart contract by the owner.[3]

Users are first authenticated through RSA signatures and then submit the requested amount as a price of digital content. After the successful delivery of data, the user is encouraged to register reviews about the data. These reviews are validated through the Watson analyzer to filter out fake reviews. Customers registering valid reviews are given incentives. In this way, maximum reviews are submitted against every file. In this scenario, decentralized storage, Ethereum blockchain, encryption, and incentive mechanisms are combined. To implement the proposed scenario, smart contracts are written in solidity and deployed on the local Ethereum test network. The proposed scheme achieves transparency, security, access control, authenticity of owner, and quality of data. In simulation results, an analysis is performed on gas consumption and actual cost required in terms of USD, so that a good price estimate can be done while deploying the implemented scenario in a real set-up. Moreover, the computational time for different encryption schemes is plotted to represent the performance of the implemented scheme, which is Shamir secret sharing (SSS). Results show that SSS shows the least computational time as compared to advanced encryption standard (AES) 128 and 256.[4]

Electronic Medical Records (EMR) have emerged as a pivotal component of healthcare data, capturing significant attention due to their potential to enhance healthcare service quality and reduce medical costs. Despite its importance, the fragmented distribution of EMRs across decentralized hospitals poses challenges to effective data sharing and raises concerns about the privacy of patients.
In response to these challenges, we introduce a novel solution named BPDS (Blockchain-based Privacy-preserving Data Sharing) for EMRs. In the BPDS framework, the original EMRs find secure storage in the cloud, while their corresponding indexes are securely retained within a tamper-proof consortium blockchain. This dual-layered approach not only significantly reduces the risk of medical data leakage but also ensures the integrity of EMRs by preventing arbitrary modifications. The secure sharing of data is facilitated automatically based on predefined access permissions through the utilization of smart contracts in the blockchain.[5]

BPDS incorporates a joint design featuring a CP-ABE (Ciphertext-Policy Attribute-Based Encryption) based access control mechanism and a content extraction signature scheme. This innovative combination enhances privacy preservation during data sharing. Rigorous security analysis confirms that BPDS stands as a secure and effective solution to facilitate data sharing for EMRs, addressing the complexities associated with decentralized hospital systems and safeguarding patient privacy.[6]

In the era of the Internet of Things (IoT), smart devices are interconnected through wired or wireless means. These IoT devices possess the capability to sense their environment and transmit relevant information to the next level. Various application areas such as Smart Cities, Smart Transportation, Healthcare, Agriculture, and Environmental Monitoring rely on extensive information sharing among different devices. However, this information-sharing system presents numerous security and privacy challenges, including data leakage, data modification, and device identity

concerns. The first part of this paper focuses on identifying the communication protocols used in IoT applications and elucidates their working principles.
In response to the challenges identified in IoT applications, the authors propose a blockchain-based solution in the second part of the paper. The authors highlight the vulnerabilities present in the current IoT communication landscape, where data from various devices is stored in a centralized database, exposing it to potential security breaches. Additionally, issues related to the proper verification of the sender's authenticity make the system susceptible to security threats. To address these concerns, the authors introduce a secure architecture based on an open Blockchain, specifically Hyperledger, for IoT applications. The proposed solution aims to enhance security measures, including non-repudiation, privacy, and scalability in the context of IoT applications. The integration of Hyperledger facilitates easy detection of malicious actors, as every node within the network is aware of all other nodes in the Hyperledger network, ultimately ensuring robust security for IoT communication.[7]

SUMMARY OF LITERATURE SURVEY

- *Peer-to-Peer Electronic Cash System: Utilizes digital signatures and proof-of-work to ensure secure online transactions without the need for centralized financial institutions, enhancing trust and decentralization in the digital economy. By preventing double-spending and verifying transactions through a distributed network, this system enables peer-to-peer electronic cash transfers, offering users greater financial autonomy and privacy. Its reliance on cryptographic techniques and decentralised consensus mechanisms fosters resilience against fraud and censorship, laying the foundation for a more inclusive and transparent financial system.*
- *Blockchain-based Data-Sharing Platform: Leverages IPFS for decentralised file storage and smart contracts for access control, fostering transparency and security in data-sharing environments. Through incentivizing genuine reviews and enforcing access permissions via smart contracts, this platform ensures authenticity and integrity in shared data, mitigating the risk of manipulation or unauthorized access. By combining blockchain technology with IPFS, it provides a robust and tamper-resistant infrastructure for data exchange, empowering users with greater control over their digital assets and fostering trust among participants.*
- *BPDS for Secure Medical Records Sharing: Employs a dual-layered approach with blockchain for index storage and smart contracts for automated, secure data sharing, safeguarding patient privacy and data integrity in decentralized hospital systems. By automating access control and ensuring adherence to predefined permissions through smart contracts, BPDS mitigates the risk of unauthorised data access or tampering, enhancing trust among healthcare stakeholders. Its use of blockchain technology enables transparent and auditable record-keeping, facilitating secure and seamless sharing of electronic medical records while maintaining compliance with privacy regulations and standards.*
- *Enhanced Security for IoT Applications: Utilizes Hyperledger to address vulnerabilities in current IoT communication systems, ensuring non-repudiation, privacy, and scalability in data sharing among interconnected smart devices. By leveraging blockchain's immutable ledger and consensus mechanisms, it provides a tamper-resistant platform for secure IoT data exchange, reducing the risk of unauthorized access or manipulation. Through the detection of malicious actors and implementation of robust security measures, this solution safeguards sensitive IoT data, fostering trust and reliability in IoT ecosystems.*
- *Digital Signatures and Proof-of-Work: Serve as critical components of the peer-to-peer electronic cash system, providing cryptographic security measures to prevent fraudulent transactions and ensure transaction integrity. Digital signatures authenticate the identity of transaction participants, while a proof-of-work consensus mechanism validates and secures transactions on the blockchain network. Together, these cryptographic techniques enable trustless and decentralized transaction validation, enhancing*

the security and reliability of digital cash transfers without the need for intermediaries.

- *Smart Contracts for Access Control: Used in blockchain-based data-sharing platforms and secure medical records sharing, smart contracts automate access control policies and permissions, ensuring transparent and auditable data-sharing processes. By enforcing predefined rules and conditions, smart contracts enhance security and transparency, reducing the risk of unauthorized data access or manipulation. Their decentralized nature and self-executing capabilities foster trust among participants, streamlining data-sharing workflows and ensuring compliance with regulatory requirements.*
- *Decentralization and Transparency: Common themes across all solutions, decentralization eliminates single points of failure and reliance on central authorities, fostering resilience and inclusivity in digital ecosystems. By providing transparency through immutable transaction records and auditable smart contracts, decentralization enhances trust among participants, enabling greater accountability and integrity in online transactions, data sharing, and IoT applications. Together, decentralization and transparency pave the way for a more secure, resilient, and trustworthy digital infrastructure, empowering users with greater control over their digital assets and fostering innovation in various domains.*

METHODOLOGY

Algorithm

Blockchain

Blockchain is a shared, immutable ledger that facilitates the process of recording transactions and tracking assets in a business network. An asset can be tangible (a house, car, cash, land) or intangible (intellectual property, patents, copyrights, branding). Virtually anything of value can be tracked and traded on a blockchain network, reducing risk and cutting costs for all involved.

Business runs on information. The faster it's received and the more accurate it is, the better. Blockchain is ideal for delivering that information because it provides immediate, shared and completely transparent information stored on an immutable ledger that can be accessed only by permissioned network members. A blockchain network can track orders, payments, accounts, production and much more. And because members share a single view of the truth, you can see all the details of a transaction end to end, giving you greater confidence, as well as new efficiencies and opportunities.

RSA

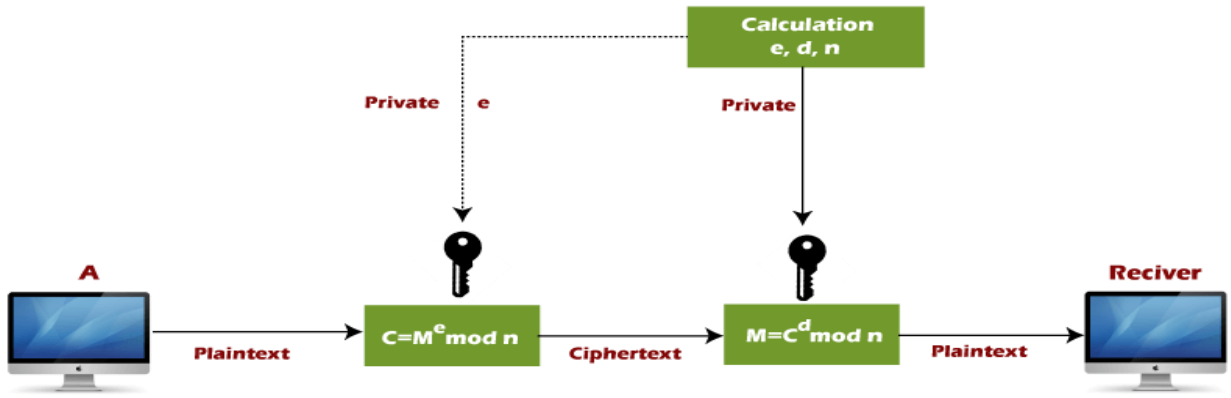
RSA algorithm is an asymmetric cryptography algorithm. Asymmetric means that it works on two different keys i.e. Public Key and Private Key. As the name describes the Public Key is given to everyone and the Private key is kept private.

An example of asymmetric cryptography:

- A client (for example browser) sends its public key to the server and requests some data.
- The server encrypts the data using the client's public key and sends the encrypted data.
- The client receives this data and decrypts it.

Since this is asymmetric, nobody else except the browser can decrypt the data even if a third party has the public key of the browser.

The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is a multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. So, if somebody can factorize the large number, the private key is compromised. Therefore encryption strength lies in the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long.



RSA

Encryption:

1. Key Generation:

- Choose two large prime numbers, p and q.
- Compute their product, $n = p \times q$, which will be the modulus for public and private keys.
- Compute Euler's totient function, $\phi(n) = (p-1) \times (q-1)$, which is used to ensure the security of RSA.
- Choose an integer e such that $1 < e < \phi(n)$ and e is coprime with $\phi(n)$. This e will be the public exponent.
- Calculate the modular multiplicative inverse of e modulo $\phi(n)$, denoted as d. This d will be the private exponent.

2. Public and Private Keys:

- The public key consists of the modulus n and the public exponent e.
- The private key consists of the modulus n and the private exponent d.

3. Encryption:

- To encrypt a message M, the sender uses the recipient's public key (n, e).
- The sender computes $C \equiv M^e \bmod n$, where C is the ciphertext.

The security of RSA relies on the difficulty of factoring large integers n into its prime factors p and q. As long as the private key remains secret and the prime factors of n are sufficiently large, RSA encryption is considered secure.

Decryption:

1. Ciphertext Reception:

- The recipient receives the ciphertext C sent by the sender.

2. Private Key Extraction:

- The recipient possesses their private key (n, d), which consists of the modulus n and the private exponent d.

3. Decryption:

- To decrypt the ciphertext C, the recipient computes $M \equiv C^d \bmod n$, where:
- M is the original plaintext message.
- d is the private exponent.
- n is the modulus.

4. Message Recovery:

- Once the recipient has calculated M, they obtain the original plaintext message sent by the sender.

It's critical to note that the security of RSA depends on keeping the private key mystery. As long as the private key remains private and the numerical properties of the RSA calculation are protected, the beneficiary can precisely unscramble the ciphertext to recoup the initial message.

Smart Contract

A smart contract serves as an automated, self-executing agreement that facilitates and enforces the rules governing file-sharing activities on the blockchain network.

1. Purpose:

- Facilitate Secure Transactions: The smart contract ensures secure and transparent file-sharing transactions between parties within the blockchain network.
- Enforce Access Control: It enforces access control policies, allowing only authorized users to upload, download, and share files based on predefined permissions.

- Maintain Transaction Records: The smart contract maintains an immutable record of file-sharing transactions on the blockchain ledger, ensuring transparency and auditability.

2. Key Features:

- Access Control Rules: Define access control rules within the smart contract to specify which users or entities have permission to perform various file-sharing actions.
- File Metadata Management: Store and manage metadata related to shared files, including ownership details, access permissions, timestamps, and transaction history.
- Transaction Execution: Automatically execute file-sharing transactions based on predefined conditions and triggers encoded within the smart contract code.
- Event Logging: Log important events and actions related to file-sharing activities on the blockchain, providing a transparent and auditable trail of transaction history.

3. Components:

- Data Structures: Define data structures within the smart contract to represent files, users, access permissions, and transaction records.
- Functions: Implement functions within the smart contract code to handle file-sharing operations such as file upload, download, access control validation, and transaction processing.
- Events: Emit events within the smart contract code to notify external applications or users about significant file-sharing events, such as successful uploads, downloads, or access requests.

4. Security Measures:

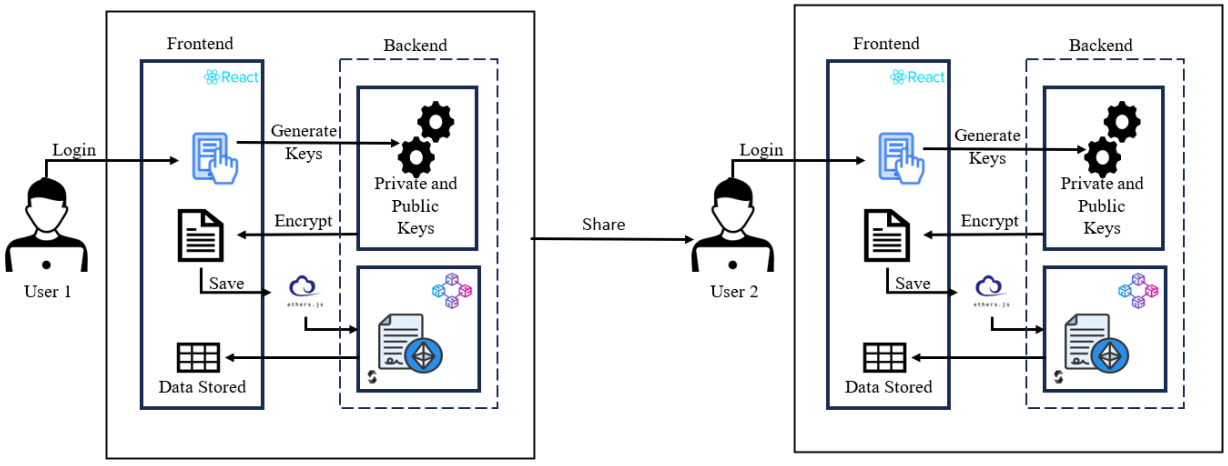
- Role-Based Access Control: Implement role-based access control mechanisms to ensure that only authorized users with the necessary permissions can interact with the smart contract.
- Encryption: Utilize encryption techniques to protect sensitive data stored within the smart contract, such as file metadata and access control rules.
- Error Handling: Implement robust error handling mechanisms within the smart contract code to prevent unauthorized access, data breaches, or other security vulnerabilities.

5. Integration:

- Integration with Blockchain Network: Deploy the smart contract onto the chosen blockchain platform, ensuring seamless integration with the underlying blockchain network infrastructure.
- Interaction with Frontend Applications: Enable frontend applications to interact with the smart contract through designated interfaces, allowing users to initiate and manage file-sharing transactions.

The smart contract plays a crucial role in automating and securing file-sharing activities on the blockchain network, ensuring trust, transparency, and integrity in the exchange of digital assets.

Architecture



The architecture of the project involves several interconnected components working together to facilitate secure file sharing using blockchain technology. Given is the Overview of the above Architecture:

1. Client Interface:
- Users interact with the system through a web-based user interface (UI) or application interface.
- The client interface allows users to upload, download, and manage files securely.

2. Web Application Layer:
- The web application layer handles user requests and responses, serving as the front end of the system.
- It includes components such as routers, controllers, and views responsible for processing HTTP requests and rendering dynamic web pages.

3. Backend Services:
- Backend services manage the core functionality of the system, including blockchain interaction, file storage, and smart contract execution.
- These services are responsible for implementing business logic, data processing, and communication with external systems.

4. Blockchain Network:
- The blockchain network serves as the underlying infrastructure for secure file sharing.
- It consists of interconnected nodes maintaining a distributed ledger containing transaction records and file metadata.
- The blockchain network implements consensus mechanisms to validate and add new blocks to the chain, ensuring data integrity and immutability.

5. Smart Contracts:
- Smart contracts are self-executing contracts with predefined rules and conditions encoded on the blockchain.
- They govern file access control, ownership, and transaction logic, ensuring secure and transparent file sharing.
- Smart contracts enforce access permissions, verify user identities, and facilitate file transfer and storage.

6. File Storage Layer:
- The file storage layer handles the actual storage and retrieval of files shared within the system.
- It utilizes distributed file storage solutions such as InterPlanetary File System (IPFS) for decentralized and resilient file storage.
- The file storage layer ensures data redundancy, availability, and fault tolerance to prevent data loss and ensure high availability.

7. Security and Encryption:
- Security and encryption mechanisms are integrated throughout the system to protect sensitive data and ensure privacy.
- Cryptographic techniques such as encryption, hashing, and digital signatures are employed to secure files, transactions, and communications.
- Access control mechanisms restrict unauthorized access to files and enforce data confidentiality and integrity.

8. Database and Persistence Layer:
- The database and persistence layer stores and manages system data, including user accounts, transaction history, and file metadata.
- It utilizes relational or NoSQL databases for efficient data storage and retrieval.
- The persistence layer ensures data consistency, durability, and scalability to support system operations.

9. External Integrations:
- The system may integrate with external services and APIs for additional functionality, such as identity verification, payment processing, or analytics.
- External integrations extend the capabilities of the system and enhance user experience by providing seamless access to external resources.

The architecture of the project combines blockchain technology, distributed file storage, smart contracts, and encryption techniques to create a secure and transparent file-sharing platform with decentralized control and data integrity.

Flow of Execution

The flow of execution for the System

1. Project Initialization:
- Begin by organising project directories and initialising essential files like configuration files.
- Install required dependencies and packages using package managers.
- Set up version control systems for effective management and tracking of project changes.

2. Blockchain Network Setup:
- Configure the genesis block and define network parameters such as block size and consensus mechanisms.
- Deploy network nodes and establish peer-to-peer communication channels.
- Implement security measures to safeguard against potential vulnerabilities and ensure network integrity.

3. Cryptographic Key Generation:
- Generate cryptographic key pairs, including public and private keys.
- Distribute public keys to network participants for secure communication.
- Employ secure storage mechanisms for private keys to maintain data confidentiality.

4. Blockchain Data Loading:
- Retrieve blockchain data from storage sources to synchronize local nodes with the network.
- Validate block hashes and transaction signatures for data integrity.
- Implement strategies for handling blockchain forks and chain reorganizations.

5. Database Management:
- Set up database schemas and configure tables for storing user data, transaction records, and other relevant information.
- Implement CRUD operations to manage database entities efficiently.
- Optimize database performance through indexing, query optimization, and caching mechanisms.

6. Web Application Configuration:
- Configure application settings to customize behaviour, including database connections and middleware settings.
- Define URL patterns and routes for mapping requests to corresponding view functions.
- Configure deployment settings for deploying the web application to production servers.

7. Model Definitions:
- Define models representing data entities such as users, files, transactions, and blocks.
- Specify model attributes, constraints, and validations for data integrity.
- Generate database migrations to propagate model changes and maintain synchronization.

8. View and Controller Logic:
- Implement view functions to handle HTTP requests, execute business logic, and render responses.
- Incorporate controller logic for user authentication, authorization, and session management.
- Implement error handling mechanisms for graceful error recovery.

9. HTML Templates and Static Files:
- Develop HTML templates for defining page structure and layout.
- Manage static files such as CSS stylesheets, JavaScript scripts, and images for visual appeal.
- Utilize template tags and filters for dynamic content generation.

10. Migration Management:
- Generate and apply database migrations to update schema changes.
- Execute migration commands to synchronise the database with model definitions.
- Perform thorough testing of migrations to ensure data consistency.

11. Testing and Debugging:
- Develop unit tests and integration tests for validating functionality.
- Utilize debugging tools and logging frameworks for error identification.
- Implement continuous testing practices for code quality assurance.

RESULTS

Security Assessment

In the security evaluation phase, the effectiveness of encryption methods employed for securing files during storage and transmission is thoroughly analysed. This involves assessing the robustness of encryption techniques in safeguarding the confidentiality and integrity of sensitive data. Additionally, the system's capability to enforce access control rules is scrutinized to ensure that only authorized users can access designated files. This involves validating the implementation of access control mechanisms and assessing their effectiveness in preventing unauthorized access to sensitive information. Furthermore, the immutability of blockchain records is verified to ensure the integrity of transaction data and file metadata. This involves confirming that the

blockchain ledger remains tamper-proof, thereby preventing any unauthorized alterations to stored information.

Performance Analysis

In the performance analysis phase, various aspects of the system's functionality are evaluated to ensure optimal performance and efficiency. This includes measuring the speed of file operations such as uploading and downloading, to ensure efficient file transfer functionality. Additionally, the speed of transaction processing on the blockchain network is assessed to guarantee the timely execution of file-sharing activities. This involves evaluating the responsiveness of the system in processing transactions and updating the blockchain ledger accordingly. Moreover, the system's scalability is evaluated to determine its ability to accommodate increasing user and file loads without compromising performance. This involves assessing how well the system can handle growing demands while maintaining optimal performance levels and responsiveness.

Conclusion

In conclusion, the project presents a comprehensive solution for secure and transparent file-sharing leveraging blockchain technology and decentralized storage systems. By integrating Hyperledger Fabric and IPFS, the system ensures confidentiality, integrity, and availability of shared files, addressing key challenges in centralized file-sharing systems. Through end-to-end encryption, tamper-resistant storage, and automated access control mechanisms, the project offers a robust and scalable platform for secure collaboration among consortiums of organizations. The successful implementation of the proposed solution demonstrates its potential to redefine standards for secure data exchange, fostering trust, transparency, and innovation in digital ecosystems.

REFERENCE

[1] N. Jeenath Laila, G. Tamilpavai, S. Saravana Kumar, "File Sharing Using Blockchain," Assistant Professor, Department Of CSE, GCE, Tirunelveli-7, India. Professor, Department Of CSE, GCE, Tirunelveli-7, India. Student, Department Of CSE, GCE, Tirunelveli-7, India. DOI:https://www.doi.org/10.56726/IRJMETS41190.

[2] T. Wu, W. Wang, C. Zhang, W. Zhang, L. Zhu, K. Gai, and H. Wang, "Blockchain-Based Anonymous Data Sharing with Accountability for Internet of Things," Tong Wu - Member, IEEE. Weijie Wang - Chuan Zhang - Member, IEEE. Weiting Zhang - Member, IEEE. Liehuang Zhu - Senior Member, IEEE. Keke Gai - Senior Member, IEEE. Haotian Wang.

[3] U. Satapathy, B. K. Mohanta, S. S. Panda, S. Sobhanayak, and D. Jena, "A Secure Framework for Communication in Internet of Things Application using Hyperledger based Blockchain," Department of Computer Science and Engineering, IIT Bhubaneswar, Odisha, India, 751003. Emails: A117010@iiit-bh.ac.in (Utkalika Satapathy), C116004@iiit-bh.ac.in (Bhabendu Ku. Mohanta), C117011@iiit-bh.ac.in (Soumyashree S Panda), srichandan@iiit-bh.ac.in (Srichandan Sobhanayak), debasish@iiit-bh.ac.in (Debashis Jena).

[4] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System."

[5] H.-S. Huang, T.-S. Chang, J.-Y. Wu, "A Secure File Sharing System Based on IPFS and Blockchain," Department of Electronics Engineering, National Chiao-Tung University, Telecommunication Laboratories, Chunghwa Telecom Co., Ltd., Hsinchu, Taiwan. Emails: tschang@mail.nctu.edu.tw (Tian-Sheuan Chang), ian_wu@cht.com.tw (Jhih-Yi Wu), phm@cht.com.tw (Hsiao-Shan Huang).

[6] S. Pradhan, S. Tripathy, and S. Nandi, "Blockchain-based Security Framework for P2P Filesharing System," Department of Computer Science & Engineering, Indian Institute of Technology Patna, India. Email: srikanta.pcs16@iitp.ac.in (Srikanta Pradhan), som@iitp.ac.in (Somanath Tripathy), sukumar@iitg.ernet.in (Sukumar Nandi).

[7] S. Peng, W. Bao, H. Liu, X. Xiao, J. Shang, L. Han, S. Wang, X. Xie, and Y. Xu, "A Peer-to-Peer File Storage and Sharing System Based on Consortium Blockchain," College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China. The State Key Laboratory of Chemo/Biosensing and Chemometrics, Hunan University, Changsha 410082, China. National Supercomputing Center in Zhengzhou, Zhengzhou University, Zhengzhou

450001, China. Faculty of Arts and Humanities, University of Macau, Macau 999078, Macao Special Administrative Region of China. Institute of Collaborative Innovation, University of Macau, Macau 999078, Macao Special Administrative Region of China. College of Information Science and Engineering, Guilin University of Technology, Guilin 541004, China.