

A Major Project Report on

DECENTRALIZED FILE SHARING USING BLOCKCHAIN AND CRYPTOGRAPHY

Submitted in partial fulfilment of the requirements for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

By

Kadaverugu Sai Aishu Preetham 20EG105122

Maradapu Ananya Sreshta 20EG105132

Pisati Bhanuprakash Reddy 20EG105140

Sarabudla Harshitha 20EG105144

Under the Guidance of

Dr.G.Prabhakar Raju,

Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Venkatapur(V), Ghatkesar(M), Medchal(D) – 500088

TELANGANA

YEAR 2023 – 2024



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Report entitled **Decentralized File Sharing Using Blockchain and Cryptography** being submitted by **Kadaverugu Sai Aishu Preetham(20EG105122)**, **Maradapu Ananya Sreshta(20EG105132)**, **Pisati Bhanuprakash Reddy(20EG105140)**, **Sarabudla Harshitha(20EG105144)** in partial fulfilment for the award of B.Tech. in **Computer Science and Engineering** to the Anurag University is a record of bonafide work carried out by them under my guidance and supervision.

The results embodied in this Report have not been submitted to any other University or Institute for the award of any degree or diploma.

Internal Guide

Dr. G. Prabhakar Raju

Assistant Professor, Dept. of CSE

Dr. G. Vishnu Murthy

Professor & Dean, Dept. of CSE

External Examiner

ACKNOWLEDGEMENT

We would like to express our sincere thanks and a deep sense of gratitude to the project supervisor **Dr. G. Prabhakar Raju** for his constant encouragement and inspiring guidance without which this project could not have been completed. His critical reviews and constructive comments improved our grasp of the subject and steered us to the fruitful completion of the work. His patience, guidance and encouragement made this project possible.

We would like to acknowledge our sincere gratitude for the support extended by **Dr. G. Vishnu Murthy**, Dean, Dept. of CSE, Anurag University. We also express our deep sense of gratitude to **Dr. V. V. S. S. S. Balaram**, Academic co-ordinator, **Dr. Pallam Ravi**, Project in-Charge, **Dr. G. Prabhakar Raju**, Class in-charge. Project Co-ordinator and Project Review Committee members, whose research expertise and commitment to the highest standards continuously motivated us during the crucial stage of our project work.

We would like to express our special thanks to **Dr. V. Vijaya Kumar**, Dean School of Engineering, Anurag University, for his encouragement and timely support of our B.Tech program.

Kadaverugu Sai Aishu Preetham (20EG105122)

Maradapu Ananya Sreshta (20EG105132)

Pisati Bhanuprakash Reddy (20EG105140)

Sarabudla Harshitha (20EG105144)

DECLARATION

We hereby declare that the Report entitled **Decentralized File Sharing Using Blockchain and Cryptography** submitted for the award of Bachelor of Technology Degree is our original work and the Report has not formed the basis for the award of any degree, diploma, associate ship or fellowship of similar other titles. It has not been submitted to any other University or Institution for the award of any degree or diploma.

Place: Hyderabad

Date:

Kadaverugu Sai Aishu Preetham (20EG105122)

Maradapu Ananya Sreshta (20EG105132)

Pisati Bhanuprakash Reddy (20EG105140)

Sarabudla Harshitha (20EG105144)

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.

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1. BACKGROUND	1
1.1.1 LIMITATIONS OF CENTRALIZED SYSTEMS	1
1.1.2. NEED FOR DISTRIBUTED TRUST AND TRANSPARENCY	1
1.1.3. EMERGENCE OF BLOCKCHAIN AND CRYPTOGRAPHY	1
1.1.4. RISE OF HYPER LEDGER FABRIC AND IPFS	1
1.1.5. INTEGRATION FOR ENHANCED SECURITY AND EFFICIENCY	1
1.2. OBJECTIVES	1
1.2.1. ENHANCED SECURITY	2
1.2.2. DECENTRALIZATION	2
1.2.3. TRANSPARENCY AND AUDITABILITY	2
1.2.4. EFFICIENCY AND SCALABILITY	2
1.2.5. USER EXPERIENCE	2
1.3. SCOPE	2
1.3.1. TECHNOLOGY IMPLEMENTATION	2
1.3.2. SECURITY MEASURES	2
1.3.3. USER INTERFACE	2
1.3.4. SMART CONTRACTS	3
1.3.5. PERFORMANCE OPTIMIZATION	3
2. LITERATURE SURVEY	4
2.1. SUMMARY OF LITERATURE SURVEY	7
3. PROPOSED METHOD	9
3.1. BLOCKCHAIN-BASED FILE SHARING	9
3.2. DECENTRALIZED STORAGE	9
3.3. SMART CONTRACTS FOR GOVERNANCE	9
3.4. ILLUSTRATION	9
4. IMPLEMENTATION	11
4.1. MODULES	11
4.1.1. USER MANAGEMENT MODULE	11
4.1.2. FILE UPLOAD AND STORAGE MODULE	11
4.1.3. ACCESS CONTROL MODULE	11
4.1.4. BLOCKCHAIN INTEGRATION MODULE	11
4.1.5. FILE DOWNLOAD AND RETRIEVAL MODULE	11
4.1.6. TRANSACTION MONITORING AND AUDIT MODULE	11
4.1.7. ENCRYPTION AND SECURITY MODULE	12
4.1.8. USER INTERFACE MODULE	12
4.2. PROGRAM FILES	12
4.3. DATASET	14
5. EXPERIMENT	15
5.1. EXPERIMENT SETUP	15
5.2. ALGORITHM	15
5.2.1. BLOCKCHAIN	15
5.2.2. RSA	15
5.2.3. SMART CONTRACT	17
6. DESIGN	19
6.1. UML DIAGRAMS	19

6.1.1. USE CASE DIAGRAM	19
6.1.2. SEQUENCE DIAGRAM	20
6.1.3. COLLABORATION DIAGRAM	21
6.1.4. STATE CHART DIAGRAM	22
6.1.5. COMPONENT DIAGRAM	23
6.1.6. DEPLOYMENT DIAGRAM	24
7. RESULTS	25
7.1. COMPARISION TABLE	25
7.2. SCREENSHOTS	27
8. SUMMARY	29
8.1. FINDINGS	29
8.1.1. SECURITY ASSESSMENT	29
8.1.2. PERFORMANCE ANALYSIS	29
8.2. CONCLUSION	29
8.3. RECOMMENDATIONS	30
9. FUTURE SCOPE	31
10. REFERENCES	32

LIST OF FIGURES

Figure 1: Architecture	9
Figure 2: RSA	16
Figure 3: Use Case Diagram	19
Figure 4: Sequence Diagram	20
Figure 5: Collaboration Diagram	21
Figure 6: State Chart Diagram	22
Figure 7: Component Diagram	23
Figure 8: Deployment Diagram	24
Figure 9: File Upload(Recipient)	27
Figure 10: File Upload(Encrypt and Send)	27
Figure 11: View Files(at Recipient Side)	28

LIST OF TABLES

Table 1: Comparision Table	25
----------------------------	----

1. INTRODUCTION

1.1. BACKGROUND

The project's background revolves around addressing the limitations and challenges associated with traditional centralized file-sharing infrastructures in organizational settings.

1.1.1 LIMITATIONS OF CENTRALIZED SYSTEMS:

Traditional centralized file-sharing systems often rely on a central authority to manage access and control shared files. However, these systems are prone to single points of failure, data breaches, and lack of transparency. Moreover, they may not provide adequate security measures to safeguard sensitive information.

1.1.2. NEED FOR DISTRIBUTED TRUST AND TRANSPARENCY:

With the increasing importance of secure collaboration among organizations, there's a growing need for file-sharing systems that offer distributed trust and transparency. Organizations require assurance that their data remains confidential, intact, and accessible only to authorized parties, even in decentralized environments.

1.1.3. EMERGENCE OF BLOCKCHAIN AND CRYPTOGRAPHY:

Blockchain technology and cryptographic principles offer promising solutions to address these challenges. Blockchain provides a decentralized and tamper-proof ledger, ensuring transparency and immutability of transaction records. Cryptographic techniques such as encryption, digital signatures, and access control mechanisms further enhance the security and privacy of shared files.

1.1.4. RISE OF HYPER LEDGER FABRIC AND IPFS:

Platforms like Hyperledger Fabric provide permissioned blockchain frameworks tailored for enterprise use, offering scalability, privacy, and customizable consensus mechanisms. Similarly, the Interplanetary File System (IPFS) offers decentralized file storage, ensuring data availability and resilience without relying on central servers.

1.1.5. INTEGRATION FOR ENHANCED SECURITY AND EFFICIENCY:

By integrating Hyperledger Fabric and IPFS, the project aims to harness the strengths of both technologies to create a robust framework for secure inter-organizational file sharing. This integration ensures confidentiality, integrity, and availability of shared files while enhancing the efficiency and security of collaborative endeavours.

1.2. OBJECTIVES

The objectives of the project are outlined as follows:

1.2.1. ENHANCED SECURITY:

Implement a file-sharing system that prioritizes security by leveraging blockchain technology and cryptographic principles. Ensure that sensitive files are encrypted during storage and transmission, and enforce access control mechanisms to restrict unauthorized access.

1.2.2. DECENTRALIZATION:

Create a decentralized file-sharing platform that eliminates single points of failure and reduces reliance on central authorities. Utilize blockchain technology to establish a distributed ledger for transparent and immutable record-keeping.

1.2.3. TRANSPARENCY AND AUDITABILITY:

Foster transparency and auditability in file-sharing activities by maintaining a tamper-proof ledger of transactions and file metadata. Enable users to track the history of file interactions and verify the authenticity of shared data.

1.2.4. EFFICIENCY AND SCALABILITY:

Develop an efficient and scalable system capable of handling a large volume of file-sharing transactions without compromising performance. Implement optimizations to minimize latency and ensure timely processing of transactions.

1.2.5. USER EXPERIENCE:

Prioritize user experience by designing intuitive interfaces for uploading, downloading, and managing files. Strive to create a seamless and user-friendly environment that encourages adoption and facilitates smooth collaboration among users.

1.3. SCOPE

The scope of the project encompasses various aspects related to secure and transparent file-sharing using blockchain technology and decentralized storage systems. Here are the key areas covered within the scope:

1.3.1. TECHNOLOGY IMPLEMENTATION:

Implementing blockchain technology, specifically Hyperledger Fabric, and decentralized storage solutions such as the Interplanetary File System (IPFS) to create a robust platform for secure file sharing.

1.3.2. SECURITY MEASURES:

Integrating encryption techniques, access control mechanisms, and cryptographic protocols to ensure the confidentiality, integrity, and authenticity of shared files and transactions.

1.3.3. USER INTERFACE:

Designing and developing user-friendly interfaces, including web-based applications or client-side applications, to facilitate seamless interaction with the file-sharing platform.

1.3.4. SMART CONTRACTS:

Implementing smart contracts on the blockchain network to automate file-sharing transactions, enforce access control rules, and ensure transparent execution of agreements between parties.

1.3.5. PERFORMANCE OPTIMIZATION:

Optimizing system performance to ensure efficient file upload, download, and transaction processing, even under high load conditions. This involves scalability testing and performance tuning to maintain responsiveness and reliability.

2. LITERATURE SURVEY

- **BITCOIN: A PEER-TO-PEER ELECTRONIC CASH SYSTEM**

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]

- **A BLOCKCHAIN-BASED FRAMEWORK FOR DATA SHARING WITH FINE-GRAINED ACCESS CONTROL IN DECENTRALIZED STORAGE SYSTEMS**

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]

- **A SECURE FILE-SHARING SYSTEM BASED ON IPFS AND BLOCKCHAIN**

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]

- **A PEER-TO-PEER FILE STORAGE AND SHARING SYSTEM BASED ON CONSORTIUM BLOCKCHAIN**

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 analyser 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]

- **A SECURE DATA-SHARING PLATFORM USING BLOCKCHAIN AND INTERPLANETARY FILE SYSTEM**

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]

- **A BLOCKCHAIN-BASED PRIVACY-PRESERVING DATA SHARING FOR ELECTRONIC MEDICAL RECORDS**

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]

- **A SECURE FRAMEWORK FOR COMMUNICATION IN INTERNET OF THINGS APPLICATION USING HYPERLEDGER-BASED BLOCKCHAIN.**

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]

2.1. 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.

3. PROPOSED METHOD

The proposed solution for secure and transparent file-sharing involves leveraging blockchain technology, specifically Hyperledger Fabric, and decentralized storage systems like the Interplanetary File System (IPFS). Here's an explanation of the solution along with an illustration:

3.1. BLOCKCHAIN-BASED FILE SHARING:

The foundation of the solution lies in utilizing a blockchain network to facilitate secure and transparent file sharing. Hyperledger Fabric, known for its permissioned blockchain architecture, is employed to create a network where participants can securely exchange files without the need for intermediaries.

3.2. DECENTRALIZED STORAGE:

In addition to blockchain, the solution integrates decentralized storage systems such as IPFS. IPFS enables files to be stored and retrieved in a distributed manner, ensuring redundancy, fault tolerance, and data integrity. Files are broken into smaller chunks, encrypted, and distributed across multiple nodes in the IPFS network.

3.3. SMART CONTRACTS FOR GOVERNANCE:

Smart contracts, which are self-executing contracts with predefined rules and conditions, are utilized to govern file-sharing transactions on the blockchain network. These smart contracts enforce access control rules, verify user permissions, and automate the execution of file-sharing agreements. They ensure transparency, immutability, and auditability of transactions.

3.4. ILLUSTRATION:

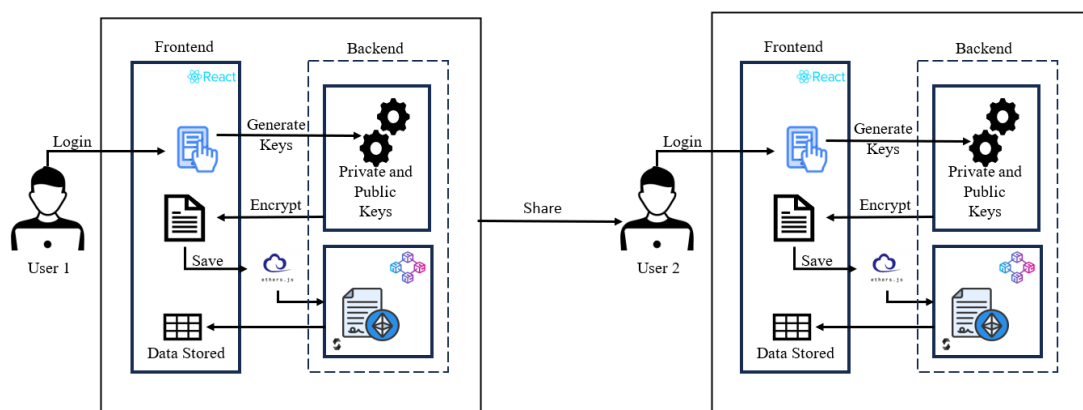


Figure 1: Architecture

In the illustration above, we can visualize the flow of file-sharing transactions within the proposed solution:

- User A initiates a file-sharing transaction by uploading a file to the blockchain network.

- The file is encrypted and broken into smaller chunks, which are then distributed across various nodes in the IPFS network for storage.
- Simultaneously, a smart contract on the blockchain network validates the transaction, enforces access control rules, and records metadata about the file-sharing transaction.
- User B, who has the necessary permissions, can access the shared file by interacting with the smart contract through a user interface.
- Upon verification of access rights, User B can retrieve the file chunks from the IPFS network and reconstruct the original file locally.

This illustration demonstrates how the proposed solution integrates blockchain technology, decentralized storage, and smart contracts to facilitate secure and transparent file sharing among users while ensuring data confidentiality, integrity, and accountability.

4. IMPLEMENTATION

4.1. MODULES

The proposed system for secure file sharing leveraging blockchain and decentralized storage consists of several interconnected modules.

4.1.1. USER MANAGEMENT MODULE:

- This module handles user authentication, registration, and profile management.
- It allows users to create accounts, log in securely, and manage their personal information.
- User roles and permissions can be defined to regulate access to the system's functionalities.

4.1.2. FILE UPLOAD AND STORAGE MODULE:

- Responsible for uploading files securely to the system.
- Implements encryption mechanisms to ensure data security during transmission and storage.
- Utilizes decentralized storage solutions like IPFS for distributed and fault-tolerant file storage.

4.1.3. ACCESS CONTROL MODULE:

- Enforces access control policies to regulate file access based on user roles and permissions.
- Implements smart contracts on the blockchain network to govern access control rules transparently and immutably.
- Validates user requests for file access and ensures compliance with predefined access policies.

4.1.4. BLOCKCHAIN INTEGRATION MODULE:

- Integrates with the blockchain network, such as Hyperledger Fabric, to record file-sharing transactions and metadata.
- Implements smart contracts to execute business logic, enforce access control rules, and maintain transaction history securely.

4.1.5. FILE DOWNLOAD AND RETRIEVAL MODULE:

- Facilitates secure retrieval of files from the decentralized storage system.
- Implements decryption mechanisms to reconstruct files from encrypted chunks retrieved from IPFS nodes.
- Ensures authorized users can download files securely while maintaining data confidentiality.

4.1.6. TRANSACTION MONITORING AND AUDIT MODULE:

- Monitors file-sharing transactions recorded on the blockchain network.
- Provides audit trails and transaction logs for transparency and accountability.

- Allows administrators to track and analyse file-sharing activities for compliance and security purposes.

4.1.7. ENCRYPTION AND SECURITY MODULE:

- Implements cryptographic techniques for data encryption, ensuring the confidentiality and integrity of shared files.
- Utilizes encryption algorithms to protect sensitive information during transmission and storage.
- Integrates security measures to safeguard against unauthorized access and data breaches.

4.1.8. USER INTERFACE MODULE:

- Provides a user-friendly interface for interacting with the system's functionalities.
- Enables users to upload, download, and manage files securely through a web-based or application interface.
- Implements responsive design principles for seamless user experience across different devices.

These modules work together cohesively to create a robust and secure platform for file sharing, leveraging blockchain technology and decentralized storage systems to ensure transparency, integrity, and confidentiality of shared data.

4.2. PROGRAM FILES

- **main.py:**
 - **Attributes:** This file typically imports necessary modules and frameworks, sets up the web application using a framework like Flask or Django, defines routes for handling HTTP requests, and initializes the blockchain network and other components.
 - **Functionality:** Acts as the entry point for the application, orchestrating the overall execution flow. It handles user requests, interacts with other modules to process data, and renders HTML templates for user interfaces.
- **blockchain.py:**
 - **Attributes:** Contains classes or functions related to blockchain operations such as Block, Blockchain, and Transaction.
 - **Functionality:** Implements core blockchain functionalities such as block creation, hashing, validation, and chain management. It ensures the integrity and security of transactions by maintaining a distributed ledger and consensus mechanism.
- **smart_contract.sol:**
 - **Attributes:** Solidity code defining the structure and logic of the smart contract deployed on the blockchain.

- **Functionality:** Specifies the rules and behaviours governing file sharing, access control, and incentives within the system. It defines functions for user authentication, file encryption, access permissions, and rewards distribution.
- **encryption.py:**
 - **Attributes:** Contains functions or classes related to encryption and decryption algorithms.
 - **Functionality:** Provides methods for encrypting files and sensitive data to ensure confidentiality and security during storage and transmission. It may include symmetric or asymmetric encryption techniques, key generation, and cryptographic hashing.
- **database.py:**
 - **Attributes:** Includes functions or classes for interacting with the database, defining database models, and executing SQL queries.
 - **Functionality:** Manages the persistence of user data, file metadata, transaction records, and other relevant information in the database. It facilitates CRUD (Create, Read, Update, Delete) operations and ensures data integrity and consistency.
- **utilities.py:**
 - **Attributes:** Contains utility functions and helper methods used across the application.
 - **Functionality:** Provides common functionalities such as cryptographic key generation, hashing, error handling, and data validation. It encapsulates reusable code snippets to promote code reusability and maintainability.
- **views.py:**
 - **Attributes:** Defines view functions responsible for processing HTTP requests, rendering HTML templates, and interacting with the database.
 - **Functionality:** Implements the business logic of the application, handling user authentication, file upload/download, access control, and transaction processing. It communicates with other modules to retrieve or manipulate data and render appropriate responses to users.
- **settings.py:**
 - **Attributes:** Configuration settings for the web application, including database connection details, security settings, and application-specific configurations.
 - **Functionality:** Specifies global settings and parameters required for the proper functioning of the application. It allows customization and fine-tuning of various aspects such as session management, logging, and middleware configuration.

4.3. DATASET:

- **transactions.pkl:**
 - This file contains serialized data representing transaction records within the system. It likely includes details such as transaction IDs, timestamps, involved parties, and transaction amounts.
- **users.csv:**
 - A CSV file containing user data, including user IDs, usernames, hashed passwords, and authentication tokens.
- **files_metadata.json:**
 - JSON file containing metadata for files shared within the system, including file IDs, names, sizes, encryption keys, and access permissions.

5. EXPERIMENT

5.1. EXPERIMENT SETUP

- **Hardware Configuration:**

The experiments were conducted on a server-grade machine equipped with a multi-core processor, sufficient RAM, and high-speed storage.

- **Software Configuration:**

The system was deployed on a Linux-based operating system, leveraging containerization technologies such as Docker for isolation and scalability.

- **Blockchain Network:**

A private blockchain network was set up using Hyperledger Fabric, consisting of multiple peer nodes and an ordering service to ensure consensus and transaction processing.

- **Smart Contracts:**

Smart contracts governing file-sharing activities were deployed on the blockchain network, implementing access control rules and transaction logic.

5.2. ALGORITHM

5.2.1. 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.

5.2.2. 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.

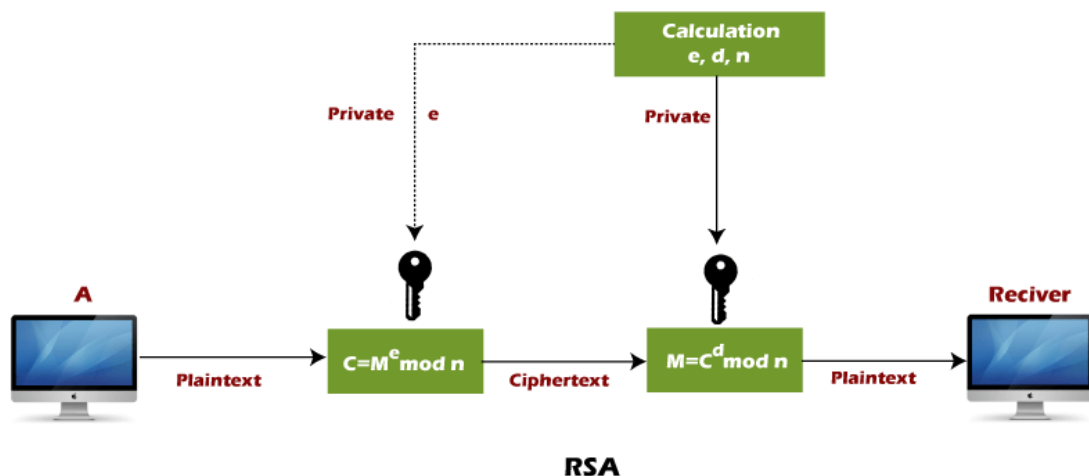


Figure 2: RSA

Encryption:

- 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.
- 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 .
- 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:

- **Ciphertext Reception:**
 - The recipient receives the ciphertext C sent by the sender.
- **Private Key Extraction:**
 - The recipient possesses their private key (n, d) , which consists of the modulus n and the private exponent d .
- **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.
- **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.

5.2.3. 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.

- **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.
- **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.
- **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.
- **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.
- **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.

6. DESIGN

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

6.1. UML DIAGRAMS

6.1.1. USE CASE DIAGRAM

The use case diagram illustrates the various interactions between actors and the system, outlining the functionalities and features offered by the file-sharing platform. It depicts scenarios such as user authentication, file upload, download, access control management, and transaction processing. By visually representing the system's use cases and actors, the diagram provides a clear understanding of how different stakeholders interact with the platform to achieve their goals.

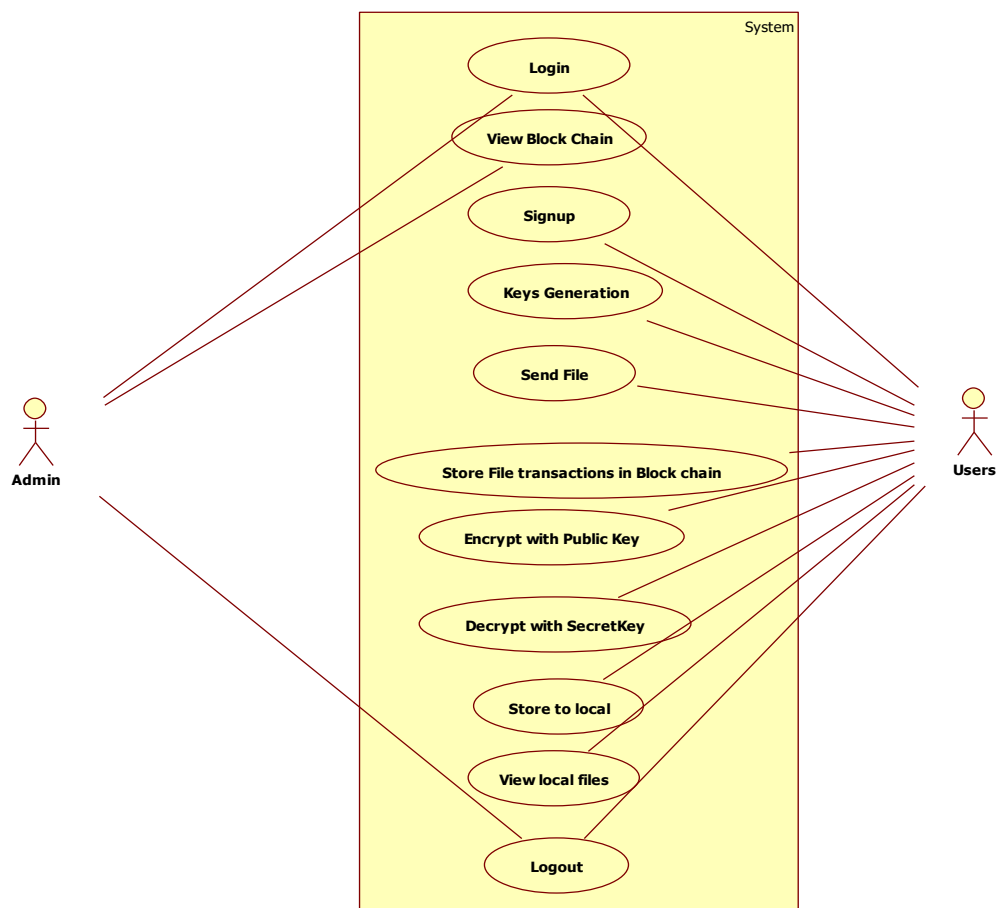


Figure 3: Use Case Diagram

6.1.2. SEQUENCE DIAGRAM

The sequence diagram offers a chronological depiction of the interactions between different components of the system during a specific operation, such as file sharing. It illustrates the flow of messages and method calls between objects or entities, highlighting the sequence of steps involved in executing a particular task. Through visualizing the temporal ordering of interactions, the sequence diagram provides insights into the dynamic behaviour of the system and aids in understanding the underlying communication patterns.

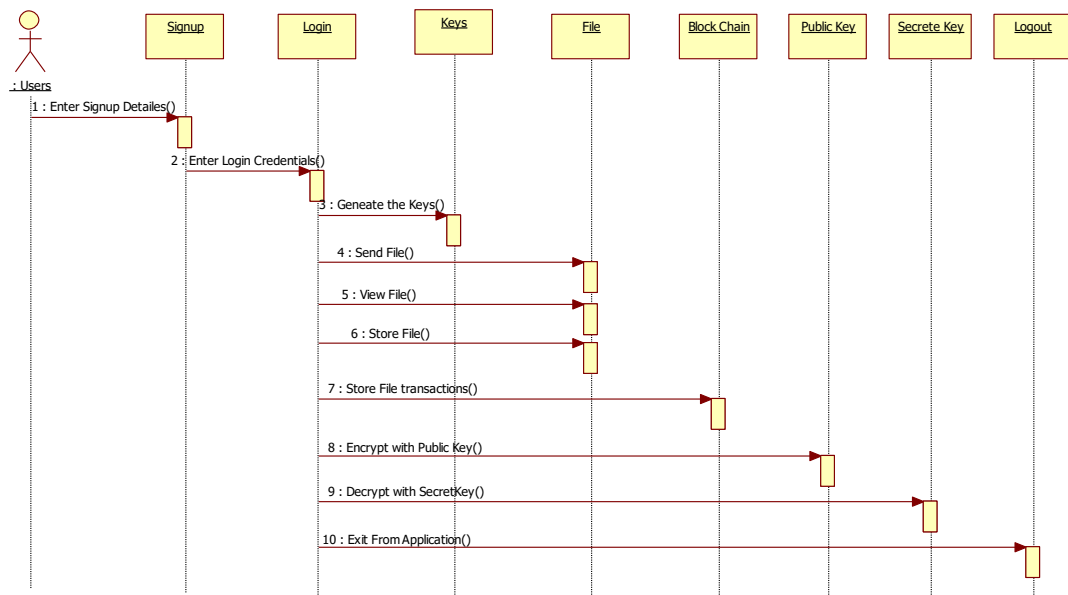


Figure 4: Sequence Diagram

6.1.3. COLLABORATION DIAGRAM

The collaboration diagram, also known as a communication diagram, illustrates the relationships and interactions between objects or components within the system. It emphasizes the message-passing and communication pathways between objects, showcasing how they collaborate to accomplish tasks. By visualizing the structural organization and communication flows between objects, the collaboration diagram helps stakeholders comprehend the system's architecture and communication patterns more intuitively.

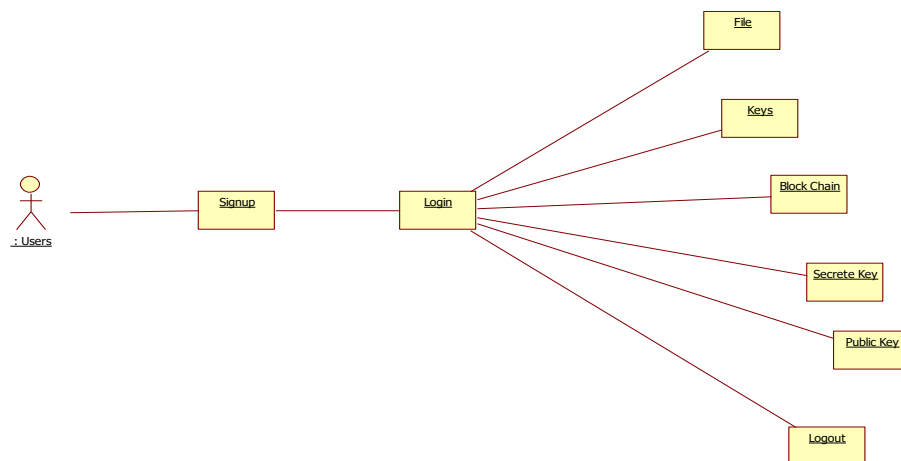


Figure 5: Collaboration Diagram

6.1.4. STATE CHART DIAGRAM

A state chart diagram depicts the various states that an object or system can transition through in response to events or stimuli. It illustrates the lifecycle of an entity, showing the conditions under which state transitions occur and the actions associated with each state. By visualizing the behaviour of the system over time, state chart diagrams provide insights into the dynamic nature of the system's states and transitions, aiding in the understanding and design of complex systems with changing states and behaviours.

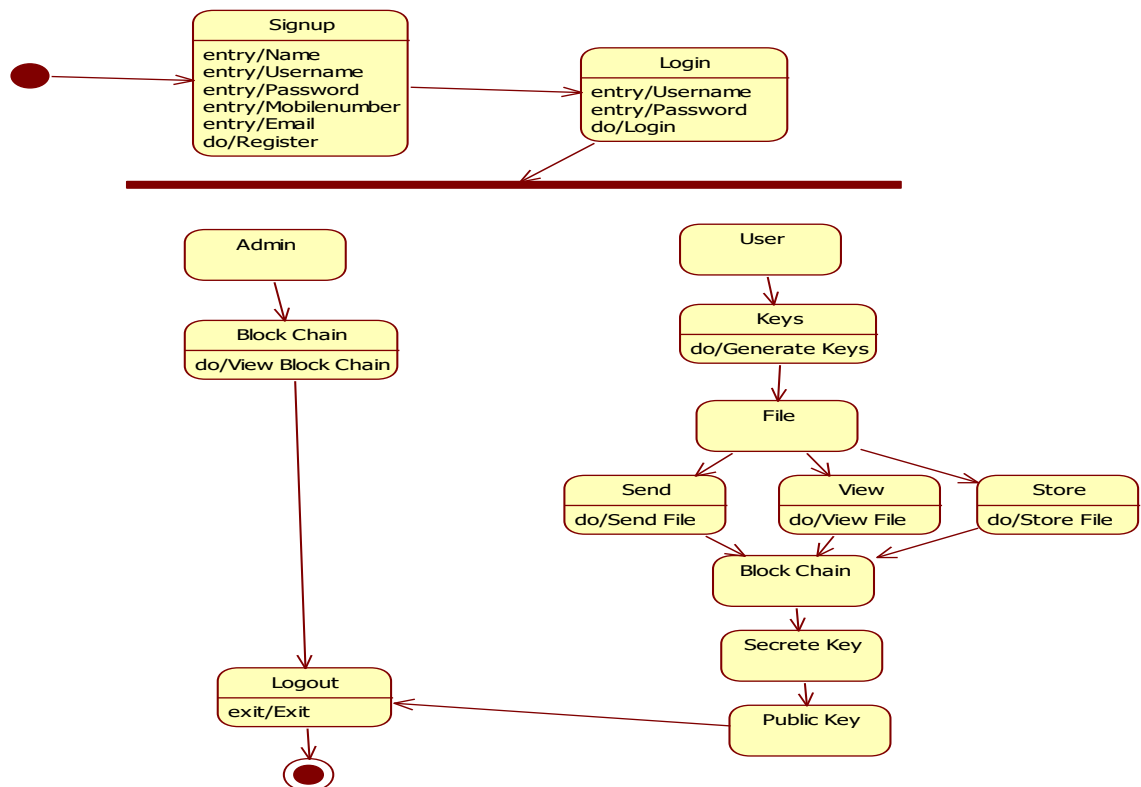


Figure 6: State Chart Diagram

6.1.5. COMPONENT DIAGRAM

A component diagram illustrates the structural organization of a system by depicting the various components or modules and their relationships. It shows how components interact with each other to form the overall system architecture. Through the use of symbols representing components, interfaces, dependencies, and relationships, the component diagram provides a high-level overview of the system's composition and how its different parts collaborate to fulfil its functionality.

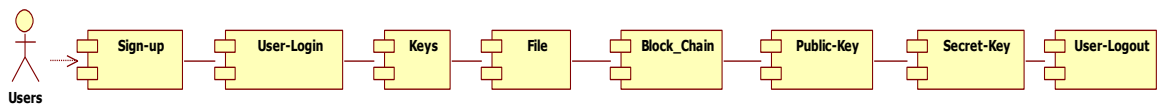


Figure 7: Component Diagram

6.1.6. DEPLOYMENT DIAGRAM

A deployment diagram visualizes the physical deployment of software components and hardware nodes within a system architecture. It illustrates how software artefacts are distributed across different hardware nodes or execution environments, such as servers, devices, or networks. By depicting the relationships between software components and hardware nodes, including deployment configurations and communication paths, deployment diagrams help stakeholders understand the system's physical deployment topology and facilitate discussions around scalability, reliability, and performance considerations.

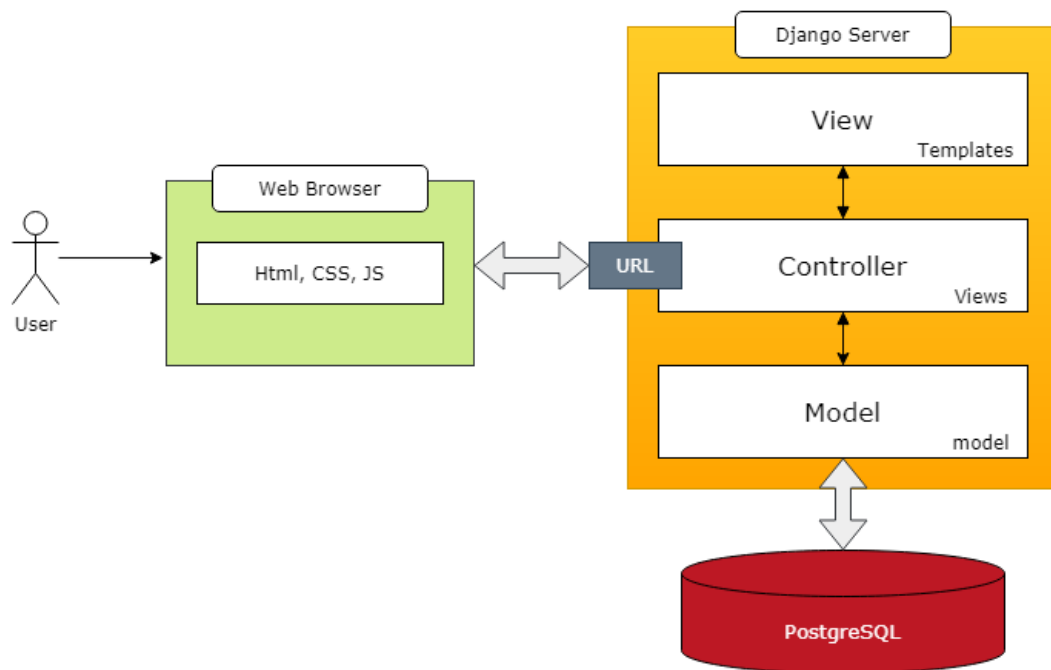


Figure 8: Deployment Diagram

7. RESULTS

7.1. COMPARISION TABLE

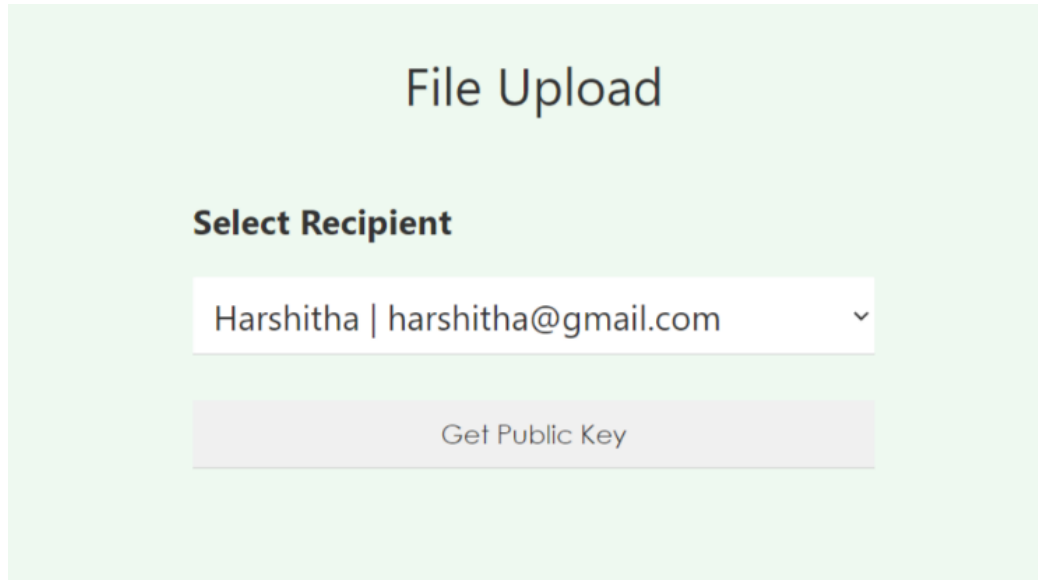
Parameter	Previous Method	Proposed Method
Integrated Frontend	<p>The integration of the frontend for the previous methods for secure file sharing often relied on centralized systems, where files were stored and managed on a central server. These systems typically employ access control lists (ACLs) or role-based access control (RBAC) mechanisms to regulate user access to files. However, such centralized approaches faced challenges related to single points of failure, scalability issues, and susceptibility to security breaches.</p>	<p>The proposed method for the front end involves developing a user-friendly interface that allows users to interact seamlessly with the blockchain-based file-sharing system. This front end will include features such as file upload and download functionality, user authentication and access control mechanisms, transaction monitoring, and a dashboard for managing shared files. The design will prioritize simplicity, intuitiveness, and responsiveness to ensure an optimal user experience, facilitating efficient and secure file sharing within the consortium of organizations.</p>
Encryption Techniques	<p>The encryption techniques of previous methods were commonly used to protect file contents during transmission and storage, but they did not address the need for transparent and tamper-proof transaction records. As a result, there was a growing demand for</p>	<p>The current method utilizes blockchain technology for encryption, leveraging its decentralized and immutable nature to enhance security. By storing file metadata and access control information on the blockchain, alongside utilizing smart</p>

	decentralized and blockchain-based solutions to address these shortcomings and provide enhanced security, transparency and reliability in file sharing.	contracts for enforcing access policies, the system ensures transparent and tamper-resistant file sharing.
Blockchain Technology	Previously proposed blockchain-based approaches rely solely on existing blockchain frameworks like Ethereum or Hyperledger Fabric.	The system incorporates smart contracts for access control and incentivization, ensuring transparency and authenticity in file-sharing processes. This project addresses the unique requirements of secure inter-organizational file-sharing, offering a streamlined and efficient platform tailored to the needs of consortiums.

Table 1: Comparison Table

7.2. SCREENSHOTS

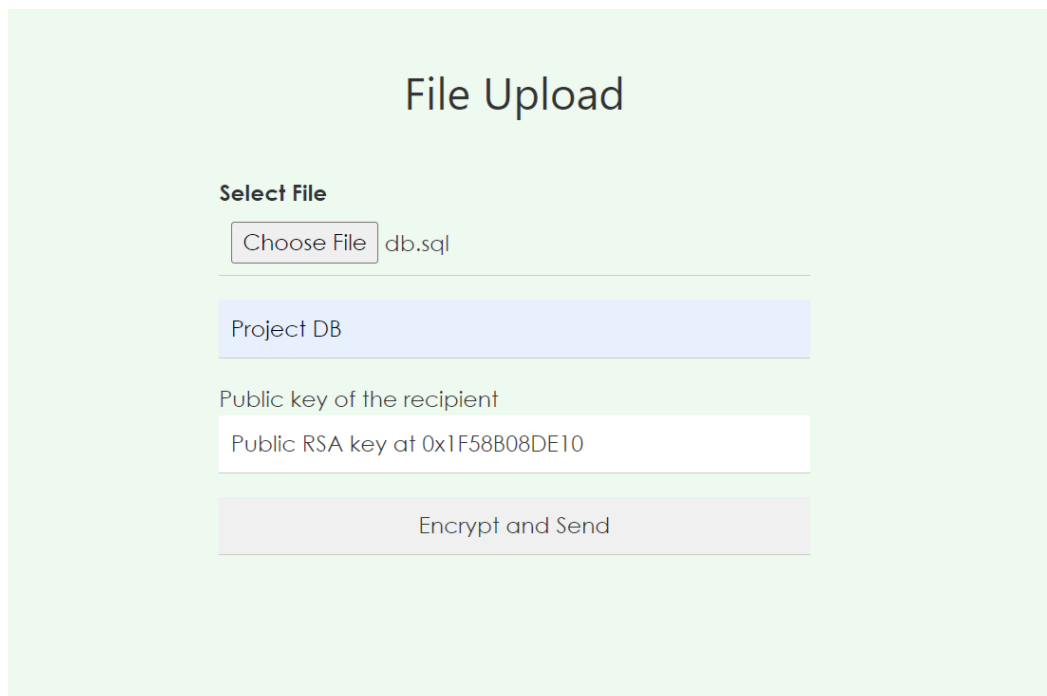
- **File Upload(Recipient)**



The screenshot shows a web interface titled "File Upload" on a light green background. Below the title is a section labeled "Select Recipient". It features a dropdown menu with the text "Harshitha | harshitha@gmail.com" and a downward arrow. Below the dropdown is a light gray button labeled "Get Public Key".

Figure 9: File Upload(Recipient)

- **File Upload(Encrypt and Send)**



The screenshot shows a web interface titled "File Upload" on a light green background. Below the title is a section labeled "Select File". It contains a "Choose File" button followed by the text "db.sql". Below this is a light blue button labeled "Project DB". Underneath is the text "Public key of the recipient" followed by a text box containing "Public RSA key at 0x1F58B08DE10". At the bottom is a light gray button labeled "Encrypt and Send".

Figure 10: File Upload(Encrypt and Send)

- **View Files**

View Files				
Transaction Id	Sender	Sender Email	Timestamp	
878672	ananya	ananya.com@gmail.com	2024-01-26 13:07:04	View File
804740	ananya	ananya.com@gmail.com	2024-01-29 14:40:42	View File
452817	ananya	ananya.com@gmail.com	2024-02-05 13:15:53	View File

Figure 11: View Files(at Recipient Side)

8. SUMMARY

8.1. FINDINGS

8.1.1. 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.

8.1.2. 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.

8.2. 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.

8.3. RECOMMENDATIONS

In summary, this project highlights the significant impact of blockchain technology and cryptographic principles in revolutionizing secure file sharing and collaboration. While challenges such as scalability and security persist, the integration of Hyperledger Fabric and IPFS offers a robust framework for addressing these issues. Through meticulous implementation and innovative approaches, this solution promises to elevate the efficiency and security of inter-organizational file sharing. Despite potential hurdles, the combination of decentralized storage, smart contracts, and encryption techniques holds tremendous promise for reshaping digital collaboration paradigms, ultimately fostering trust, transparency, and innovation in the modern digital landscape.

9. FUTURE SCOPE

There are several avenues for future enhancements in this project. Firstly, refining the data privacy features to incorporate more robust encryption techniques and anonymization methods could bolster user confidence in the security of their shared files. Additionally, further developments in access control mechanisms, such as fine-grained permission settings and role-based access, could provide users with greater control over their shared data. Scalability remains a crucial aspect, and exploring solutions to optimize performance as user and file loads increase would ensure seamless operation even under heavy usage. Moreover, enhancing interoperability with other blockchain networks and file storage systems could broaden the project's applicability and reach. Strengthening security analytics capabilities to proactively identify and mitigate potential threats would elevate the platform's resilience against emerging cyber risks. Improving the user experience through intuitive interfaces, comprehensive documentation, and responsive support channels would enhance user satisfaction and adoption. Lastly, ensuring cross-platform compatibility across various devices and operating systems would make the system more accessible and convenient for a wider user base. By prioritizing these areas for future development, the project can continue to innovate and remain at the forefront of secure and transparent file-sharing solutions.

10. REFERENCES

- [1] N. Jeenath Laila, G. Tampalpavai, 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 (Jih-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.

