ANURAG UNIVERSITY

Decentralized File Sharing: Blockchain and Cryptography

Team Details

1. K. Preetham(20EG105122)
2. M. Ananya(20EG105132)
3. P. Bhanuprakash(20EG105140)
4. S. Harshitha(20EG105144)

Project Supervisor

Dr. G. Prabhakar Raju
Assistant Professor

Introduction

This project introduces a blockchain-based secure file-sharing system to facilitate confidential and transparent file exchanges among a consortium of organizations. It utilizes blockchain technology, smart contracts, and distributed file storage to ensure secure and auditable transactions.

The system requires a web-based interface for user interaction, backend services for blockchain interaction and file management, and integration with external systems for enhanced functionality.

Applications of this project include secure document sharing in corporate environments, decentralized file storage solutions, transparent data exchange in supply chain management, and secure electronic health record sharing in healthcare systems.

Problem Statement

In traditional file-sharing systems, centralized platforms pose significant security and privacy risks due to the reliance on single points of failure and the potential for unauthorized access or tampering. Additionally, existing solutions often lack transparency and may not adequately protect sensitive data during transmission and storage.

Challenges:

1. Security Risks
2. Privacy Concerns
3. Transparency Issues

Objectives:

1. Develop a Secure and Transparent Platform
2. Enable Secure Data Exchange
3. Enhance Trust and Reliability

Proposed Method

1. Blockchain-Based File Sharing Network:

- Imagine a network of interconnected computers, referred to as nodes, each running a blockchain protocol.
- Users within this network can upload files to be shared with others, similar to traditional file-sharing platforms.
- However, instead of relying on a central server to store and manage files, the files are distributed across the network of nodes.

2. Decentralized Storage with IPFS:

- When a user uploads a file, it is broken down into smaller chunks and encrypted for security.
- These encrypted chunks are then distributed across multiple nodes in the network using the InterPlanetary File System (IPFS).
- Each node stores only a portion of the file, ensuring redundancy and fault tolerance.

Proposed Method

3. Smart Contracts for Access Control:

- Smart contracts are deployed on the blockchain to manage access control to the shared files.
- Users can define access permissions for each file, specifying who can view, download, or modify it.
- Access permissions are enforced automatically by the smart contracts, ensuring that only authorized users can access the files.

4. Transaction Execution and Validation:

- When a user requests to download or access a file, the smart contract verifies their permissions and initiates the transaction.
- The transaction details, including access permissions and file metadata, are recorded on the blockchain as a new block.
- Other nodes in the network validate the transaction by consensus, ensuring its integrity and immutability.

Proposed Method

5. User Interaction:

- Users interact with the file-sharing platform through a user-friendly interface, similar to conventional file-sharing applications.
- They can upload, download, and manage files, as well as define access permissions using intuitive controls.
- Behind the scenes, blockchain technology and decentralized storage ensure the security, transparency, and resilience of the file-sharing process.

Experiment Environment

1. Blockchain Network:

- A blockchain network, such as Ethereum or Hyperledger Fabric, serves as the underlying infrastructure for the file-sharing platform.
- Nodes within the network run blockchain protocols and participate in consensus mechanisms to validate transactions and maintain the integrity of the ledger.

2. IPFS (InterPlanetary File System):

- IPFS is utilized for decentralized file storage, allowing files to be distributed across multiple nodes in the network.
- Nodes in the IPFS network store encrypted file chunks and facilitate file retrieval and sharing.

Experiment Environment

3. Smart Contracts:

- Smart contracts are deployed on the blockchain to govern access control and file-sharing rules.
- These smart contracts are written in languages like Solidity (for Ethereum) or Go (for Hyperledger Fabric) and are executed on the blockchain network.

4. User Interfaces:

- Web-based or desktop applications provide user interfaces for interacting with the file-sharing platform.
- Users can upload, download, and manage files through these interfaces and define access permissions using intuitive controls.

Experiment Environment

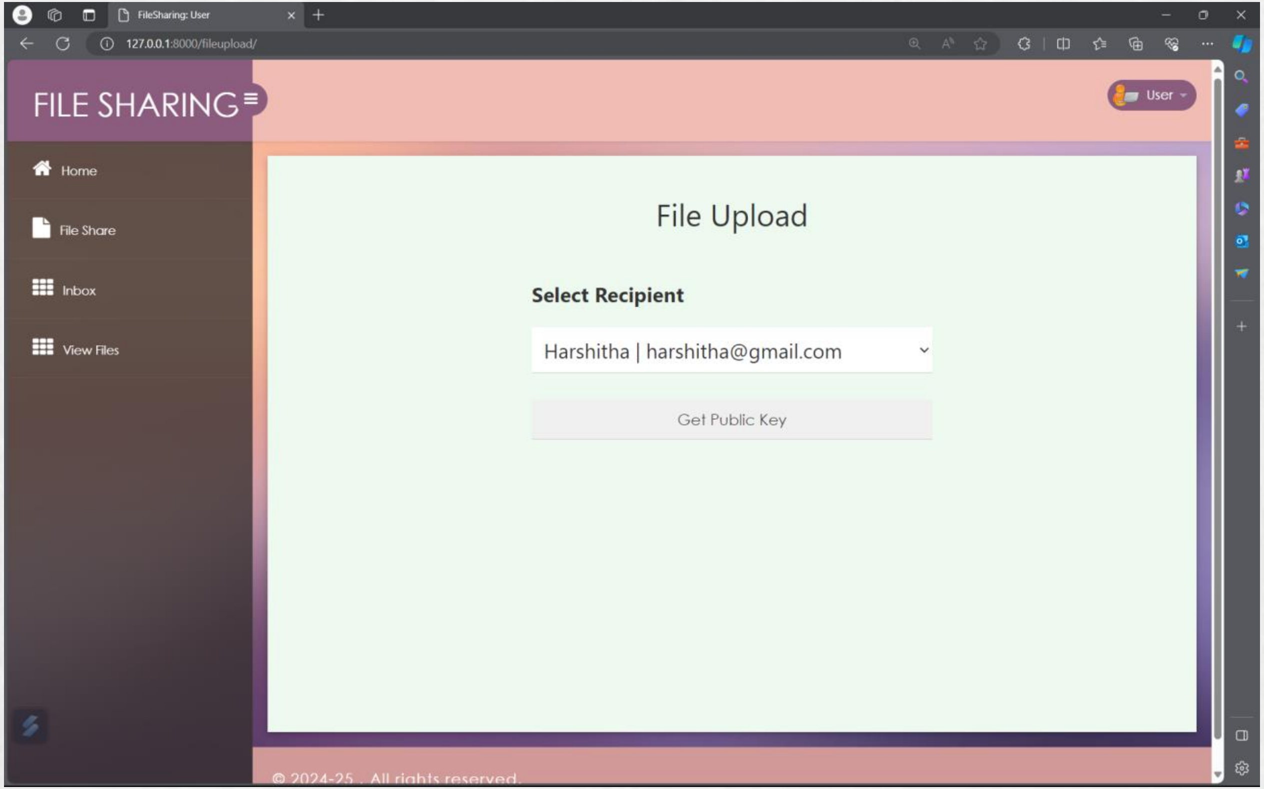
5. Testing Frameworks:

- Testing framework like Ganache are employed to conduct unit tests and integration tests for the file-sharing platform.
- These frameworks help ensure the correctness and robustness of the platform's functionality under different scenarios.

6. Security Measures:

- Various security measures, including encryption techniques, access control mechanisms, and authentication protocols, are implemented to protect sensitive data and ensure platform security.
- Tools like MetaMask or hardware wallets may be used for secure key management and transaction signing.

Experiment Screenshots



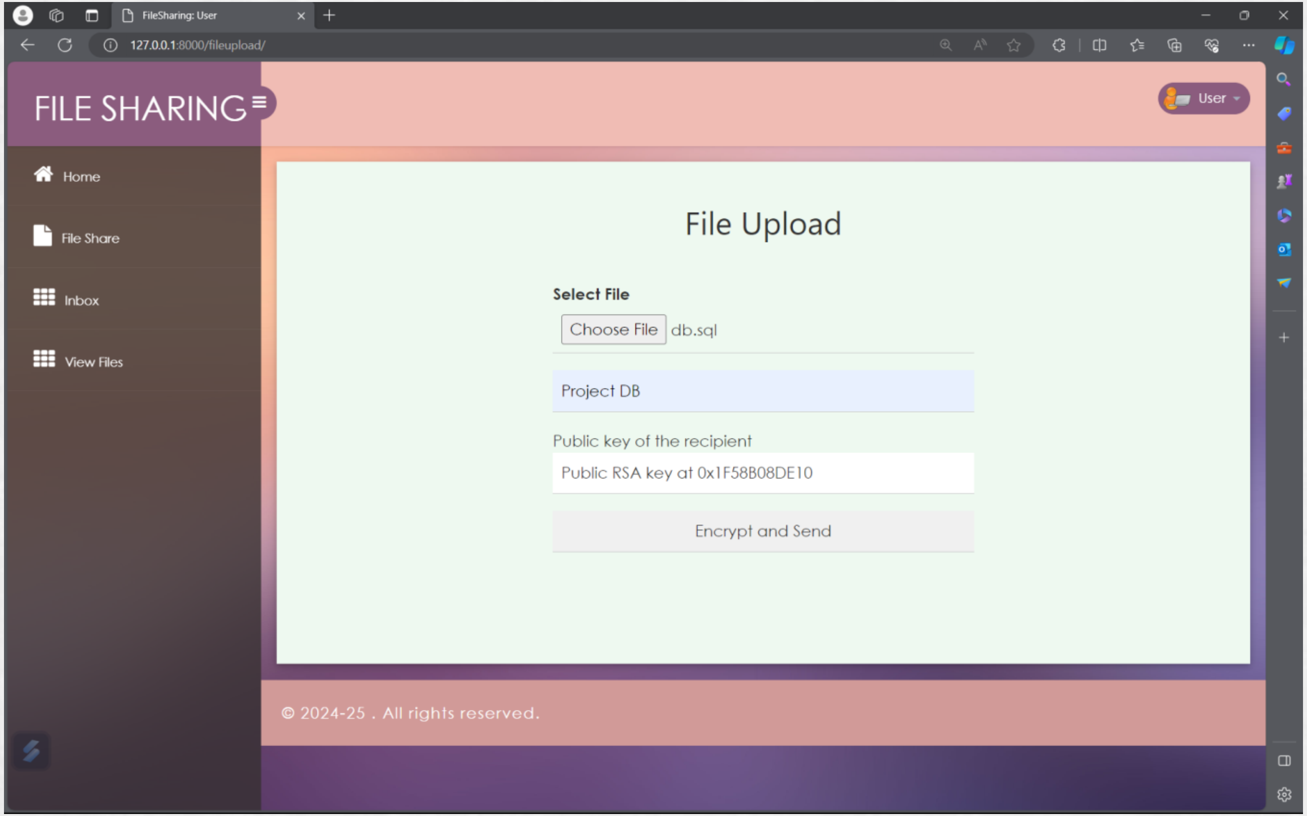
File Upload

Select Recipient

Harshitha | harshitha@gmail.com

© 2024-25 . All rights reserved.

Experiment Screenshots



File Upload

Select File

Choose File db.sql

Project DB

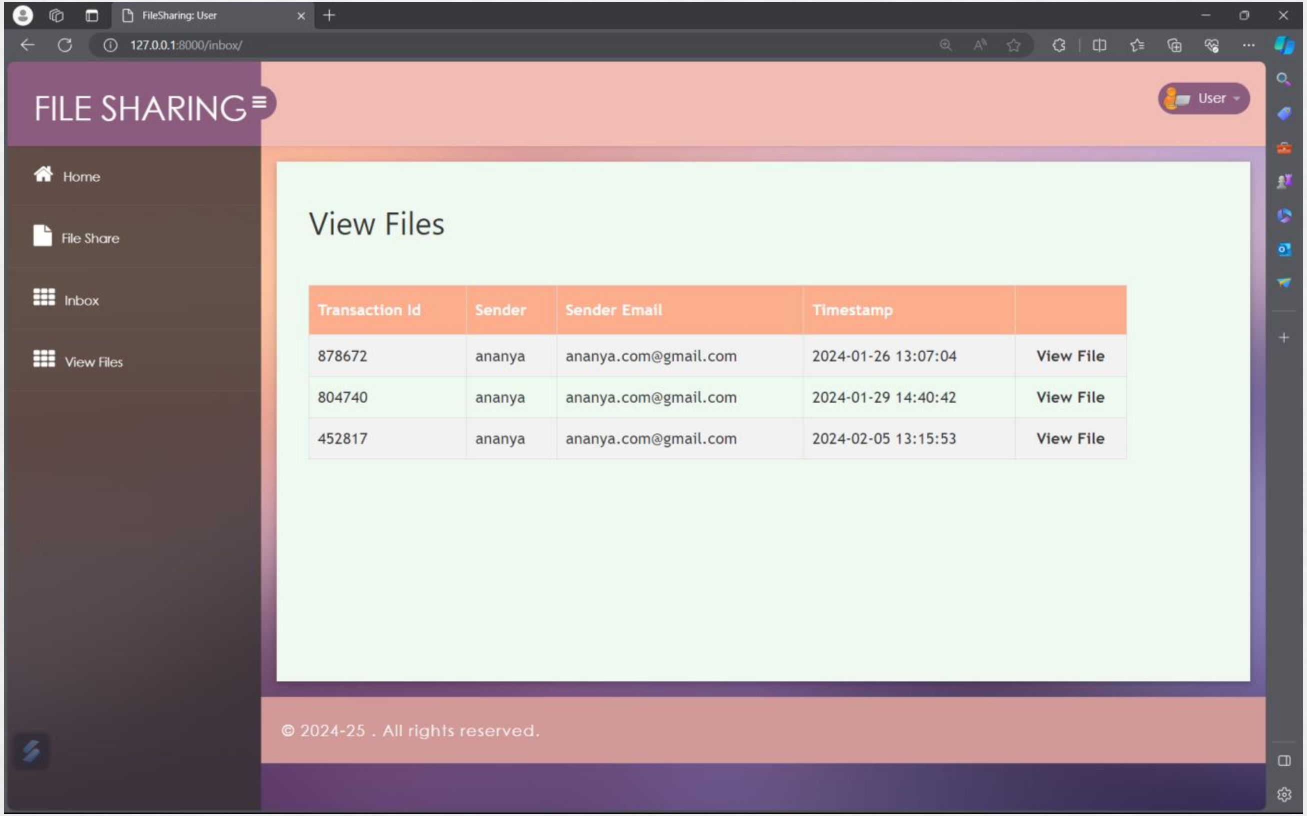
Public key of the recipient

Public RSA key at 0x1F58B08DE10

Encrypt and Send

© 2024-25 . All rights reserved.

Experiment Screenshots



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

© 2024-25 . All rights reserved.

Experiment Results

1. Security Assessment:

- Evaluation of encryption methods and access control mechanisms to ensure the confidentiality and integrity of shared files.
- Analysis of the immutability of blockchain records to verify the integrity and tamper-proof nature of transaction data and file metadata.
- Identification of potential vulnerabilities and weaknesses in the system's security architecture, along with recommendations for mitigation.

2. Performance Analysis:

- Measurement of the speed and efficiency of file operations, such as uploading and downloading, to assess the system's responsiveness.
- Evaluation of transaction processing speed on the blockchain network to ensure timely execution of file-sharing activities.
- Assessment of the system's scalability to determine its ability to handle increasing user and file loads while maintaining optimal performance levels.

Experiment Results

3. Usability and User Experience:

- Feedback from users regarding the ease of use, intuitiveness, and overall experience of interacting with the file-sharing platform.
- Identification of user pain points, obstacles, and areas for improvement in terms of user interface design and functionality.
- Iterative refinement of the user interface and user experience based on user feedback and usability testing.

4. Integration and Compatibility:

- Testing of external integrations with services and APIs for additional functionality, such as identity verification or payment processing.
- Verification of compatibility with different web browsers, operating systems, and devices to ensure broad accessibility and usability.
- Validation of interoperability with existing systems and technologies used by stakeholders or third-party services.

Findings

1. Security Enhancement:

- Implementation of blockchain technology and encryption techniques has significantly enhanced the security of the file-sharing platform.
- Encryption methods employed for securing files during storage and transmission have proved robust in safeguarding data confidentiality and integrity.
- Access control mechanisms, enforced through smart contracts, have effectively restricted unauthorized access to shared files, ensuring data privacy.

2. Improved Data Integrity:

- The immutability of blockchain records has been successfully verified, ensuring the integrity and tamper-proof nature of transaction data and file metadata.
- Blockchain-based data sharing has mitigated the risk of unauthorized alterations to stored information, providing a trustworthy platform for secure collaboration.

Findings

3. Enhanced Performance:

- File operations, including uploading and downloading, have demonstrated optimal speed and efficiency, ensuring seamless file transfer functionality.
- Transaction processing on the blockchain network has shown timely execution, with the system maintaining responsiveness even under increasing user and file loads.

Justification

Parameters Improved:

1. Security Parameters:

- **Confidentiality:** Encryption techniques have enhanced data confidentiality by securing files during storage and transmission.
- **Integrity:** Blockchain-based data sharing has ensured the integrity of transaction data and file metadata through immutability and tamper-proof records.
- **Access Control:** Smart contracts have enforced access control mechanisms, restricting unauthorized access to shared files and ensuring data privacy.

2. Performance Parameters:

- **Transaction Processing:** Transaction processing on the blockchain network has demonstrated timely execution, maintaining responsiveness under varying user and file loads.
- **Scalability:** The system has shown scalability in handling increasing demands, maintaining optimal performance levels even with growing user and file volumes.

Justification

Formulas Used:

Each user generates a key pair i.e. public and private key using the following steps:

- Each user selects two large primes at random - p, q
- Compute their system modulus n=p. q
- Calculate ø(n), where ø(n)=(p-1)(q-1)
- Selecting at random the encryption key e, where 1<e<ø(n),and gcd (e , ø(n))=1
- Solve following equation to find decryption key d: e. d=1 mod ø(n) and 0≤d≤n
- Publish their public encryption key: Ku={e, n}
- Keep secret private decryption key: KR={d, n}

Justification

Parameter values improved because:

1. Enhanced Security Measures:

- Implementation of robust encryption techniques, access control mechanisms, and authentication protocols may have improved security parameters such as confidentiality, integrity, and regulatory compliance.

2. Optimized Performance:

- Streamlined algorithms, efficient data structures, and optimized network configurations may have contributed to improved performance parameters such as speed, transaction processing, and scalability.

3. Improved User Experience:

- User-centric design, intuitive interfaces, and responsive feedback mechanisms may have enhanced usability parameters such as user experience, accessibility, and satisfaction.

Thank You