Secured Document Storing Using Blockchain

[1]Sakshi Jha, [2]Govind Dhingra, [3]Gagan Mittal, [4]Harsh Vardan

[1]Assistant Professor, [2]Student, [3]Student, [4]Student
Department of Computer Science and Engineering,
Maharaja Agrasen Institute of Technology, Delhi, India

***Abstract*: With the rapid advancement of technology and the growing number of information records, there is a significant risk of data leakage and record tampering, posing a serious threat to the privacy and accuracy of research records. When this information is stored on a centralized server, security and reliability issues may arise. As a result, a distributed system that is both efficient and secure is required.**
Blockchain is the arising innovation which endeavors to tackle these issues by making carefully designed occasion of records in a distributed environment. So, we are proposing a secured decentralized document storing and sharing option in which we are using IPFS which enables us to store large files and put immutable, permanent links in transactions. Our solution uses Huffman compression for file size optimization and RSA encryption is used for data security purposes.

Index Terms: Blockchain, IPFS, RSA Encryption, Solidity, Ethereum Smart Contracts.

I. INTRODUCTION

Changes in the technology used to create and retain records are nothing new for information workers. Several such shifts have occurred over the last few decades, resulting in new types of records, such as web or social media records, as well as recordkeeping solutions, such as electronic records management systems and cloud-based software services. Blockchains, which can be defined as ledgers with entries organized in an append-only, sequential chain using cryptographic links and distributed and stored on a peer-to-peer computer network, are an emerging recordkeeping technology that is producing new types of records and new recordkeeping modalities with which records and information professionals will need to engage.

The most well-known use of blockchains is in cryptocurrency systems, where they serve as a secure and decentralized ledger of transactions. The blockchain's novelty is that it ensures the accuracy and security of a data record while also generating trust without the requirement for a trusted third party.

Because of blockchain's decentralized nature, all transactions may be transparently observed by utilizing a personal node or blockchain explorers, which let anybody to witness transactions in real time. Each node has its own copy of the chain, which is updated as new blocks are added and confirmed. This implies one could track it anywhere it went if wanted.

Motivation

The motivation was the existing issue of data breaches which results in losses of millions of dollars to develop system resistant to such breaches. The structure of data in a blockchain differs from that of a traditional database. A blockchain organizes data into groupings called blocks, each of which contains a collection of data. Blocks have specific storage capabilities, and when they are filled, they are closed and linked to the preceding block, producing a data chain known as the blockchain. All additional information added after that newly added block is compiled into a new block, which is then added to the chain after it is filled.

A database organizes data into tables, whereas a blockchain organizes data into chunks (blocks) that are strung together, as the name suggests. When implemented in a decentralized manner, this data structure creates an irreversible data time line. When a block is filled, it becomes permanent and part of the timeline. When each block is added to the chain, it is given a specific time stamp.

In a scenario where a hacker wants to change a blockchain and take cryptocurrency from everyone else, if they changed their single copy, it would no longer match the copy of everyone else. When everyone else compares their copies, they'll notice that this one stands out, and that hacker's version of the chain will be discarded as invalid.

The cost of accomplishing such a feat would almost certainly be impossible, given the scale of many cryptocurrency networks and how quickly they are developing. Not only would this be prohibitively expensive, but it would also be futile. Such behavior would not go unnoticed by network participants, who would notice the blockchain's significant changes.

Challenges

1. Integration of IPFS

Because IPFS is used, our blockchain is both lightweight and scalable. If data were stored directly on the blockchain, it would grow incredibly large and inefficient. We can take advantage of IPFS' decentralized storage capabilities while simultaneously boosting the security and accessibility of the blockchain by merging IPFS and blockchain. Instead of directly storing files on the blockchain, we use the IPFS network, and the blockchain only keeps the file's hash. IPFS employs the SHA-256 hashing algorithm,

which ensures that each file has a distinct hash. As a result, the file is kept and accessible via the blockchain in a secure manner. The hash generated by the file is easily accessible. IPFS eliminates the storage bottleneck as a result.

2. User Interface and User Experience

Creating a simple and effective user interface is one of the most difficult components of developing an app. A better user interface equals a better user experience. These days, simpler and more customer-centric web apps are in high demand. Small UI elements can have a huge impact on the user experience. If your website's navigation is simple, visitors will have a better experience. Intuitive navigation leads your viewers to the content they seek without a high learning curve. When navigation is simple, visitors can find information quickly, resulting in a flawless experience that keeps them from turning to competitors.

3. Implementation of file compression

One of the most crucial tasks and obstacles in the project was file compression. We needed to compress the data so that we could upload larger files. File compression also aids in the reduction of the blockchain network's Gas fee. The processing speed of compressed files has also risen. Huffman compression was used to achieve file compression. To build this feature, we used the js-string-compression node module, which has an internal method that offers Huffman compression. It also has a decompression function that is used to decompress the data.

Scope

The protection of personal information from those who should not have access to it, as well as the ability of individuals to control who has access to it, is referred to as data privacy.

Data privacy has become more important as the number of individuals utilizing the Internet has increased. Websites, software, and social media platforms commonly need to acquire and keep personal data about users in order to provide services. On the other side, some programs and platforms may go beyond users' expectations in terms of data collection and use, leaving them with less privacy than they intended.

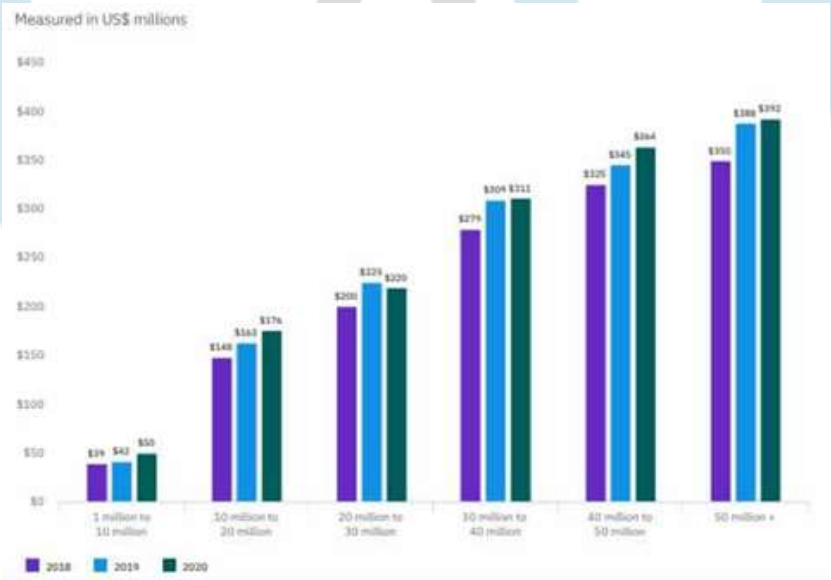

Fig 1: Data of revenue loss because of data breaches over the years

II. LITERATURE SURVEY

DD-Locker: Blockchain-based Decentralized Personal Document Locker [1]

In this paper, a blockchain-based solution for securely storing the personal document is proposed. The documents are stored without being misused or exploited by the vendor or the officials due to various issues like DoS attacks, information breaches and privacy concerns present in the centralized storage solutions like DigiLocker. The proposed system is makes possible to verify documents remotely which has become a requirement, especially during the pandemic times when the candidates could not be physically present during the registration process or interview. The solution provides confidentiality, data privacy, access control, and Integrity of the documents. The proposed solution uses the features of immutability and tamper-proof property of Ethereum blockchain, to maintain the transactions that store user details, meta-data about uploading the document, granting and sharing read permission, etc. The proposed system implementation uses blockchain which stores the event logs in a tamper-proof way to protect the system from repudiation attacks. The designed application may be extended with an additional privilege called as issuer who is responsible for issuing the documents to the resident. The application has three roles for resident- who uploads the document and has power to share, requester- who requests the documents for verification purposes and administrator- who has the complete control over the infrastructure. The application comprises of various features including the authentication, uploading and granting permission

for reading. Though the files are stored after encrypting them but the proposed system is serverless and depends on centralized servers for functioning. In the current application, Dropbox is used as the current cloud provider. The developers have proposed to use IPFS or multiple cloud services integrations. This is because centralized cloud storage solutions such as Dropbox, Google Drive, Microsoft OneDrive, etc. though have been increasingly popular in recent years to suit the data storage and sharing needs of businesses, organizations, and individuals but such systems are more likely to be attacked or have their services severely disrupted. Decentralized storage systems on the other hand provide several advantages over centralized systems, including over-coming the single-point failure problem. Furthermore, they ensure the complete removal of confidence in a third party which enhances privacy of users. This is need as the stored documents may contain a lot of personal information, and their leak due to a data breach can be misused to launch phishing attacks on the individuals. Storing on IPFS will also address the issue of scalability and cost.

Distributed Data Sharing System based on Smart contract and IPFS. [2]

We looked into numerous blockchain technologies, such as the Consensus algorithm, for this article. Blockchain, like other distributed systems, has the difficulty of efficiently reaching consensus.

We learned about smart contracts, which are special programs that may be run automatically on the blockchain. Its distinguishing feature is that the computer code and data are both stored on the chain, making it highly tamper-proof and decentralized. Transactions are used to design and execute smart contracts. Because the contract programe runs on all nodes in the distributed network, any node failure has no impact on the contract program's operation. Ethereum and Hyperledger Fabric are the two most popular blockchains for smart contracts right now.

IPFS (Inter Planetary File System) is a globally connected distributed file system that combines the benefits of distributed hash tables, quick switching, version control systems, and self-certified file systems into one system. It is addressable by content, non-tamperable, and decentralized. IPFS calculates the file fingerprint based on the file content while storing a file. When acquiring a file, IPFS will acquire it from the storage node using the file fingerprint and fix it before returning it to the user.

Private, secure, and censorship resistant document sharing [3]

We selected this literature for survey because its complement our research and it's based upon latest researches. Learning from this literature is related to Security assessment of related work, Cryptography, Distributed ledger, Decentralized data storage and file-sharing services like Sia, Storj, etc.

Got to know how much current archive(data) exchange frameworks (for example Dropbox) consent to the data security integrity, confidentiality, privacy, protection, realness of creators, non-renouncement, and responsibility; with the outcome that all examined frameworks need support for protection and obscurity. Primarily because of their incorporated plan, missing (meta)data encryption, and guidelines in which they work. In light of that investigation a decentralized idea for documents sharing from one user to other in a shared style using client-side encryption, the division of information and metadata, through distributed ledger technology for directory administration arrangement, was created.

Research about different Cryptography like Symmetric cryptography, Hashing, Asymmetric cryptography, Encryption. In symmetric cryptography the equivalent (secret-)key is utilized to perform both of two partner cryptographic tasks. Generally symmetric cryptography is utilized to scramble plaintext into cyphertext and to unscramble cyphertext back into plaintext.

A hash work H maps an inconsistent, variable-length bit string, s, into a fixed-length string, h=H(s), called hash. Asymmetric cryptography, additionally called public-key cryptography, utilizes a key-pair to perform two partner cryptographic activities. One key is utilized for the first activity and the other for its partner. Common applications are encryption, computerized marks, and key-understanding. Asymmetric cryptography is utilized for encryption the public key is utilized by an outsider to scramble content for the substance possessing the relating private key.

In decentralized information storage services lease capacity from one another. Because of their decentralized plan decentralized capacity arrangements don't depend on a solitary confided in outsider to be functional and therefore are more control safe than centralized arrangements. Gave benefit to take an interest in the framework, ways of managing coming up short or difficult to reach nodes, and the shift from server-side security to client-side security arrangement should be thought of.

Sia, Storj are secure decentralized information storage choices. In this information is encoded client-side before it is transferred. Information is sharded into lumps of fixed size to protect metadata security. Issue with Storj, Sia is that they just a data storage service and not a file sharing option, legitimacy of creators, responsibility and collaboration are not applicable.

III. METHODOLOGY

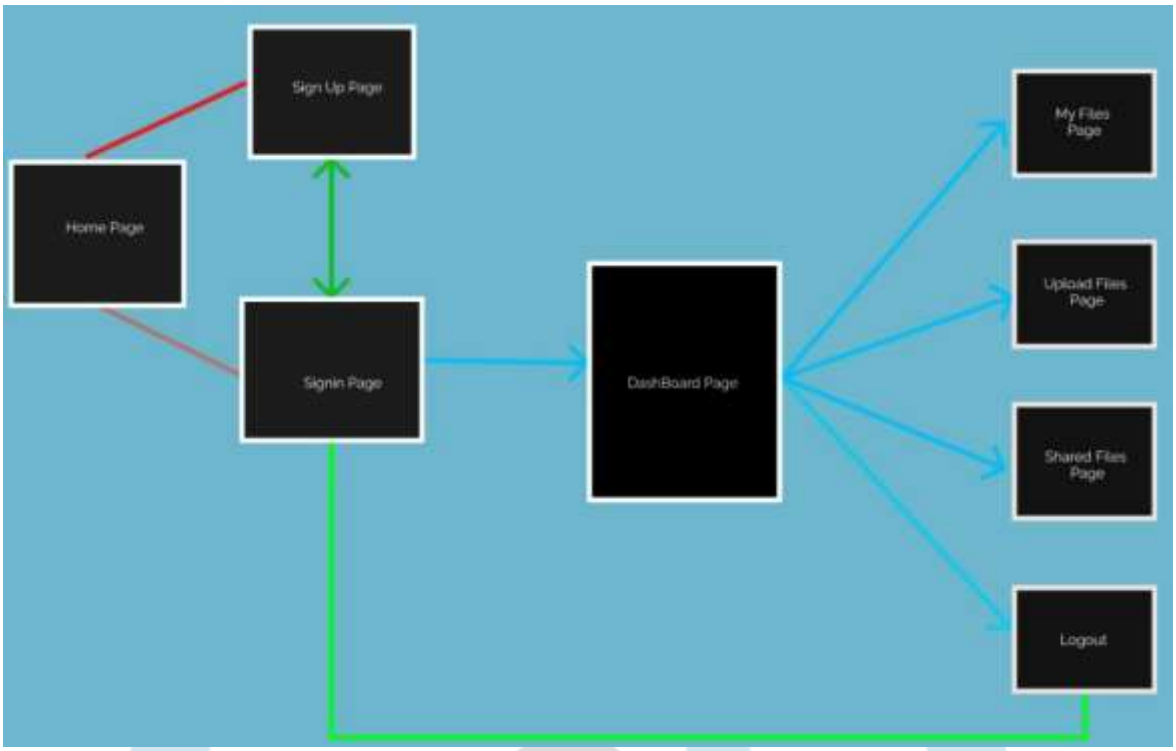

Fig 2: Flow chart of user's journey inside the application

The Block contains:

- **Block number** - Simply displays the index number of the block. Block 0 refers to the genesis block.
- **Timestamp** - This field indicates as to when the block was created and added to the blockchain.
- **Proof** - Also called a nonce, it stands for "number only used once," which is a number added to a hashed—or encrypted—block in a blockchain that, when rehashed, meets the difficulty level restrictions i.e by varying the proof we can vary the hash generated so that a new block can be created.
- **Previous hash** - This field represents the hash of the previous block. (In this case block index 2). The hash of the entire block is generated using the SHA-256 hashing algorithm. This field creates a chain of blocks and is the main element behind blockchain architecture's security.
- **Sender** - The person who uploads the file enters his identity proof or name when he uploads the file.
- **Receiver** - Displays who the intended receiver of the shall be.
- **Hash of the file shared** - The uploaded file is first encrypted with the file key given by the uploader using the AES encryption mechanism and subsequently using the SHA-256 hashing algorithm when it is uploaded to ipfs. The hash, then received from the IPFS after the encryption is the hash of the shared file which is added to the block.

- **Design of blockchain contract**

  Smart contracts are simply programs stored on a blockchain that run when predetermined conditions are met. They typically are used to automate the execution of an agreement so that all participants can be immediately certain of the outcome, without any intermediary's involvement or time loss. They can also automate a workflow, triggering the next action when conditions are met. So, we need to make a structure similar to class and design that system by analyzing the required member functions and structures.

- **Design of interface**:

  An interactive interface is needed for easy access and seamless usage. We explored different web frameworks which can be used to make a working interface for our project. After comparing some of the frameworks like React, Angular, Vue. etc., we decided to use react over other because of its advantages like

1. Speed
2. Flexibility
3. Performance
4. Usability

The sender and receiver of a shared file share a unique key/password to strengthen the security of the file(s) on the blockchain network.

The uploader who is eager to distribute the file should complete out the upload page. Before uploading the file to the IPFS network, the file key entered here will be used to encrypt it with AES encryption. The uploader must only share the key with the intended recipient(s) in order for them to download the file. The following file types can be uploaded:.pdf,.png,.jpeg, and.txt. The maximum file size that can be uploaded to the network is currently 16 Megabytes.

The receiver must fill out the download page if he or she has the sender's valid file key and wants to download the shared file from the blockchain to his or her local computer. The file key is used to unlock the AES encrypted file that was downloaded from the IPFS network so that it could be read. For a successful download, make sure you enter the right file key and hash.

IV. ARCHITECTURE

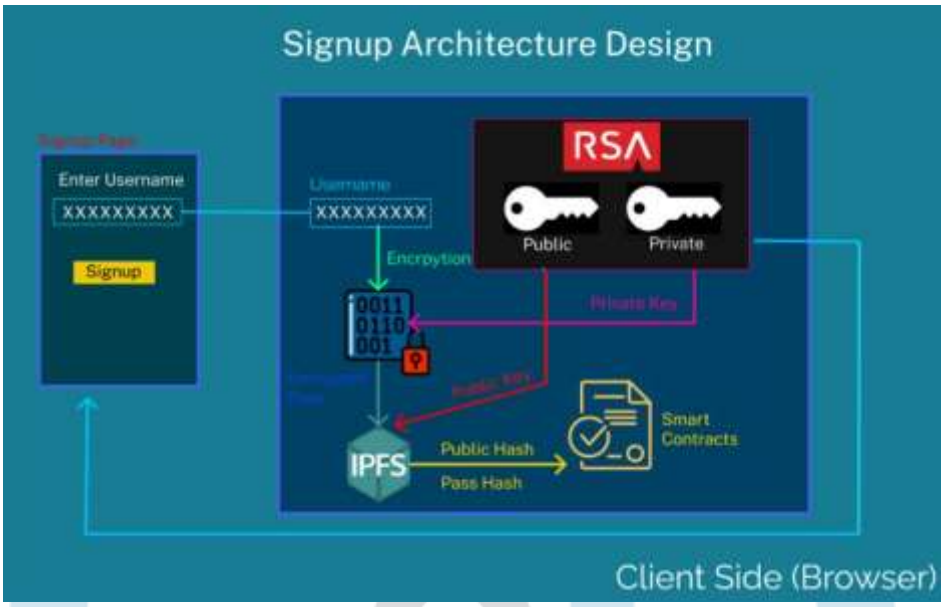

Fig 3: Background tasks execution when user signup

When user signup, backend code runs and generate 2 keys: public and private key (RSA). Public key is stored at IPFS of that user which create a public hash, which triggers the smart contract of backend and authenticate the user to proceed into the application.

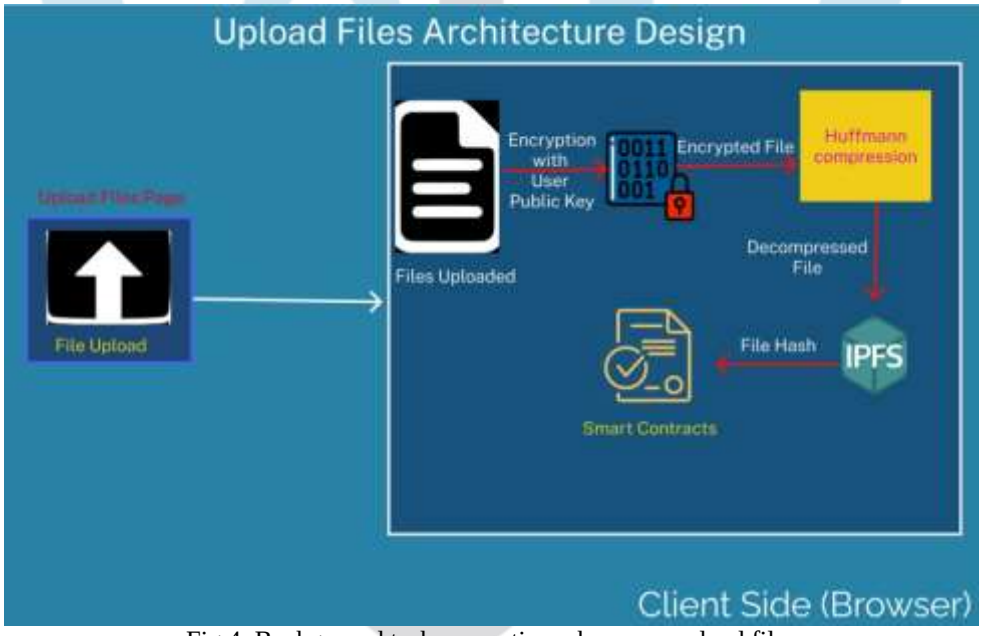

Fig 4: Background tasks execution when user upload files

To upload the file user will require its private key generated when user sign up. User will go onto upload section and select the file to upload, after which will upload its private key, then the backend code will run.

Uploaded file will get encrypted and will be compressed by using Huffman compression.

Then this compressed file will go through IPFS to generated the file hash which will trigger the smart contract and its contract will be made and its metadata is stored on blockchain.

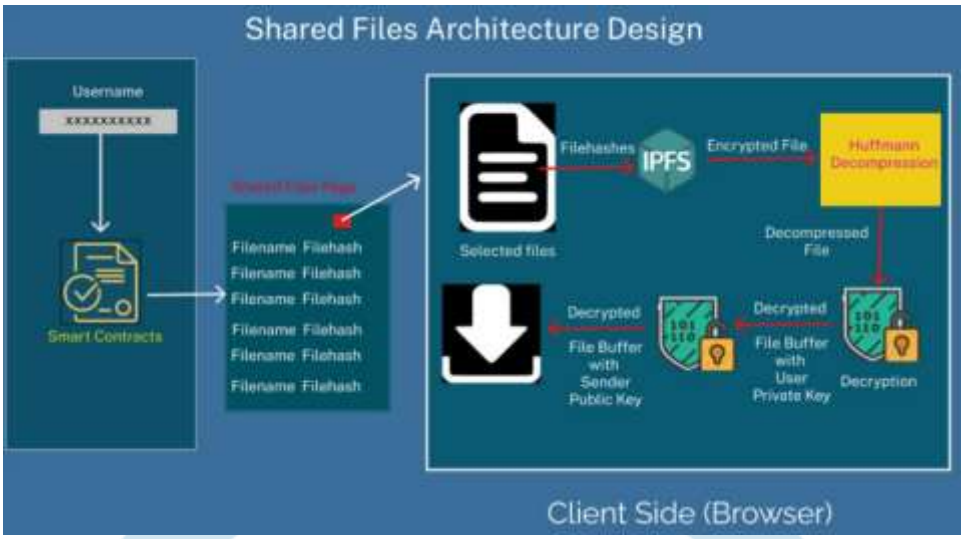

Fig 5: Background tasks execution when user share file

Upon entering the username and password, the user can access the smart contracts to access the shared files page.

To The selected file then passes through the IPFS.

IPFS uses three main functions:

1. Content addressing for Unique identification.
1. Directed Acyclic graphs (DAGs) for content linking.
1. Distributed hash tables (DHTs) for content discovery.

The encrypted file then passes through the Huffman decompression. The file is then decrypted using user private key which is then decrypted using sender public key. After this the file is available for download.

V. USING THE TEMPLATE

- **Node.js**

  Node.js is an open-source runtime environment and framework for executing web applications outside of the browser. Although it was designed with real-time, push-based architectures in mind, it's a server-side programming language that's usually used for non-blocking, event-driven servers like ordinary webpages and back-end API services. Node.js is one of the JS engines available in each browser. js is powered by Google Chrome's V8 JavaScript engine.

- **Ethereum Smart Contracts**

  A smart contract is a self-executing contract in which the conditions of the buyer-seller agreement are put directly into lines of code. The code and agreements it contains are disseminated across a decentralized blockchain network. Transactions are trackable and irreversible, and the programming regulates their execution.

- **Solidity**

  Solidity is a high-level object-oriented language for creating smart contracts. Smart contracts are programs that control how accounts behave in the Ethereum state. Solidity is written in curly brackets. It's inspired by C++, Python, and JavaScript, and it's built for the Ethereum Virtual Machine. Solidity is statically typed and, among other things, enables inheritance, libraries, and sophisticated user-defined types.

- **React**

  React is a JavaScript toolkit for creating user interfaces that makes creating interactive UIs a breeze. Create basic views for each of our application's states. Because React is solely concerned with state management and rendering that information to the DOM, building React apps frequently necessitates the usage of extra frameworks for routing and client-side functionality.

- **IPFS**

  The Interplanetary File System (IPFS) is a distributed file system protocol and peer-to-peer network for storing and sharing data. In a global namespace connecting all computing devices, IPFS uses content-addressing to uniquely identify each file.

- **Metamask**

  MetaMask is a browser extension that makes it easier to access Ethereum's Dapp ecosystem. It also functions as a wallet for ERC-20 tokens, allowing users can use the wallet to access network services.

VI. RESULT

Desired result of the project was achieved and make a robust document storing and sharing was developed with easy-to-use interface. Users only need the Metamask account to be able to perform the blockchain transactions to operate the application. A

significant note is that the private key generated at the time of sign up should be carefully preserved by the user. The application gives option to upload, share and download the files while preventing them from any attacks or breaches.

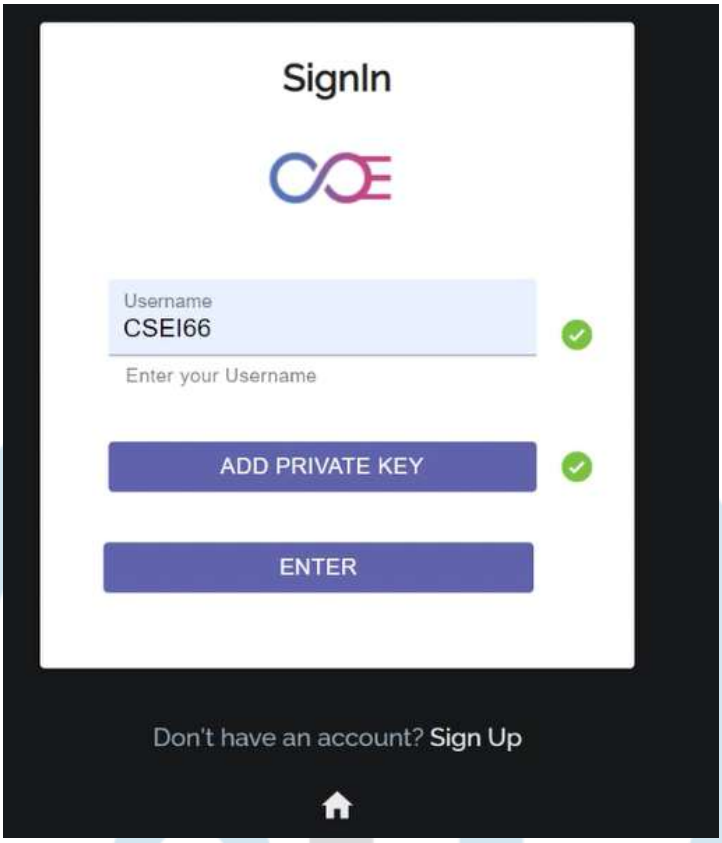

Fig 6: Sign In page

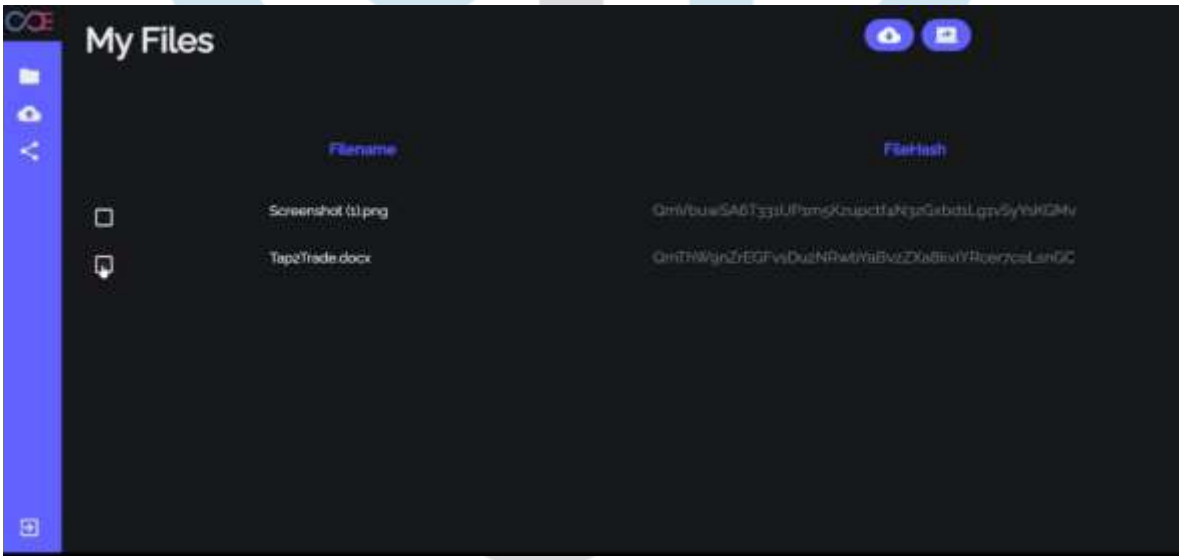

Fig 7: Dashboard of user uploaded files

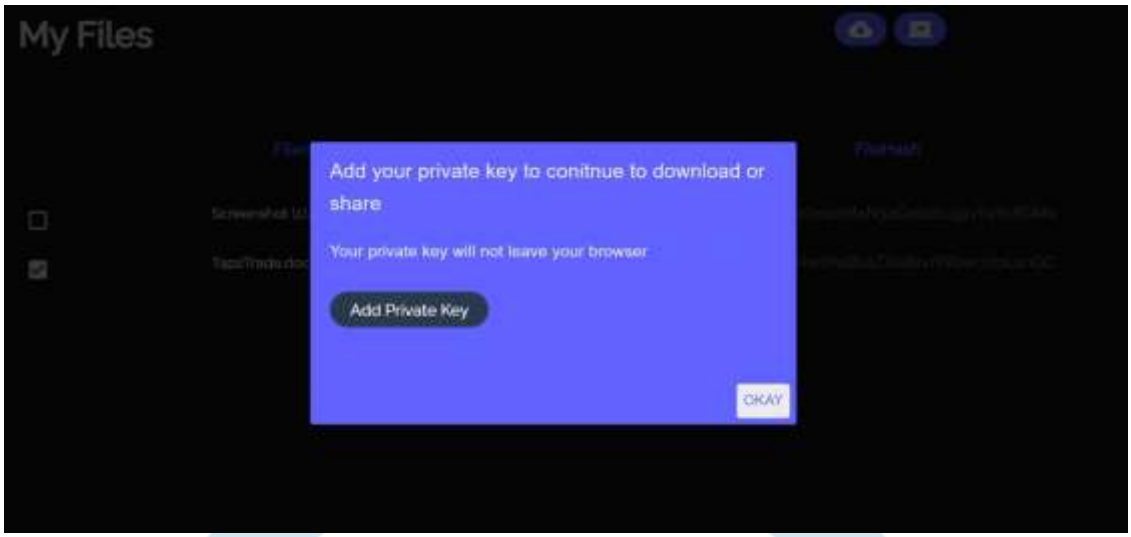

Fig 8: Private key required for download / sharing

VII. Conclusion

This paper proposes a framework for developing a secure, tamper-resistant paradigm for keeping research records in a distributed file system with no single point of control. The metadata information collected from the distributed file system is likewise stored on the blockchain. The blockchain provides an indelible record of events since it is a distributed ledger system that records all transactions and cannot be changed or altered. As a result, malicious alterations to the blockchain's metadata information are prevented.

The developed solution provides confidentiality, access control, data privacy, and document integrity. Using the immutability and tamper-proof capabilities of the Ethereum blockchain, the proposed system stores transactions that hold user details, meta-information about uploading the document, granting and distributing read permission, and so on. The issuer authorization, which is responsible for issuing documents to residents, can be added to this application to improve it.

.

References

[1] Jai Singhal, Ankit Singh Gautam, Ashutosh Bhatia, Ankit Agrawal " DD-Locker: Blockchain based Decentralized Personal Document Locker" International Conference on Information Networking (ICOIN), 68-73 Jan, 2022
[2] Enchang Sun, Kang Meng, Ruizhe Yang, Yanhua Zhang and Meng Li "Distributed Data Sharing System based on Internet of Things and Blockchain " Journal of Systems Science and Information Volume 119 No. 15 2018, 1437-1442.
[3] Jens Rowekamp," Private, secure, and censorship resistant document sharing for individuals and groups based on distributed ledger technology ", 2018.
[4] Ajay Kumar Shrestha, Julita Vassileva and Ralph Deters, "A Blockchain Platform for User Data Sharing Ensuring User Control and Incentives", October 2020.
[5] Enchang Sun, Kang Meng, Ruizhe Yang, Yanhua Zhang and Meng Li, " Research on Distributed Data Sharing System based on Internet of Things and Blockchain", Journal of Systems Science and Information, July 2021.
[6] Mihir Nevpurkar, Chetan Bandgar, Ranjeet Deshmukh, Jay Thombre, Rajashri Sadafule,
[7] Suhasini Bhat, "Decentralized File Storing and Sharing System using Blockchain and IPFS", International Research Journal of Engineering and Technology (IRJET), Volume: 07 Issue: 05 | May 2020.
[8] Ian Zhou, Imran Makhdoom, Mehran Abolhasa "A Blockchain-based File-sharing System for Academic Paper Review", Conference: International Conferences on Signal Processing and Communication, October 2019.
[9] Alevtina Dubovitskaya, Petr Novotny, Zhigang Xu, Fusheng Wange, " Applications of Blockchain Technology for Data-Sharing in Oncology: Results from a Systematic Literature Review", December 2019.