Decentralized Web-Based File Storage & Sharing System on Ethereum

Vinay Kiran Polisetti
IIIT NAYA RAIPUR
Rajahmundry, India
polisetti21100@iiitnr.edu.in

Anirban Bhattacharjee
IIIT NAYA RAIPUR
Korba, India
anirban21100@iiitnr.edu.in

Amrit Gupta
IIIT NAYA RAIPUR
Ramagundam, India
amrit21100@iiitnr.edu.in

***Abstract*— Blockchain technology has been widely explored as a decentralized solution for recording transactions, but it has limitations when it comes to storing large files or documents. A decentralized storage system called InterPlanetary File System (IPFS) has been developed to overcome this challenge. Although there have been attempts to combine IPFS and blockchain, sharing data through this approach has proven to be inefficient. To address this issue, a secure file-sharing system is proposed that utilizes a Decentralized Application (dAapp) for distributed access control and group key management. The dApp takes charge of control policies, while the combination of the IPFS server and blockchain network enables a secure file-sharing system where members can create or join groups based on their preferences. Although the IPFS server and blockchain network lack an access control mechanism, the secure file-sharing system manages access control policies so that members can access only the files they are authorized to.**

Keywords*— *Ethereum, Decentralization, File Sharing, IPFS, Blockchain, MetaMask, Infura, Keys

I. INTRODUCTION

Digital privacy is becoming increasingly important in the modern age as more personal information is shared and stored online. In centralized systems, privacy cannot be fully ensured because the user's personal data and information are stored and controlled by a central authority or organization. This creates a single point of failure and vulnerability to cyber attacks or data breaches, where a hacker can gain access to all user data at once. Additionally, the central authority may have access to or share user data with third parties without the user's knowledge or consent, compromising their privacy. The problem addressed is the lack of privacy and security in traditional centralized file-sharing systems, and the need for a decentralized and secure alternative for file sharing on a global and peer-to-peer network along with team collaboration. The use of Blockchain such as Ethereum and InterPlanetary File System (IPFS) in decentralized applications (dApps) provides an opportunity to address this problem but the privacy aspect of dApps based on Blockchain and IPFS has not been fully explored.

A. Literature Survey

Paper Title	Methodology
Enhancing the Security of the Blockchain and the File Contents [1]	Combining Blockchain and IPFS to store files. Using only AES to encrypt file contents.
Secure Distributed Cloud Storage based on the Blockchain Technology and Smart Contracts [2]	Using a Ethereum Blockchain to store & transfer files using smart contracts. Using only RSA for encryption.
Decentralized File Storage (IPFS System) using Blockchain [3]	Uses Blockchain and IPFS to transfer files. Only peer-to-peer transfer is possible .No groups or global space.
A Secure File Sharing System Based on IPFS and Blockchain [4]	Employs secure file sharing with distributed access control and group key management, without a global space.

Table.1 *Summary of related research papers*

B. Research Gaps

After the literature survey as mentioned in Table 1

1. Blockchain networks can be used for two purposes. The integrity of the hash data collected from cloud collection to the blockchain network is protected and stored in a distributed manner to ensure stability.
2. To secure the data record, the cloud server is obliged to request block data from the blockchain network as permanent proof of data integrity.
3. Additionally, data analytics data analysis and control system and server analysis can help determine the first stage of denying a distribution of service attacks.
4. In addition, each response from the cloud server and website access will be recorded in a block series for further review or investigation.
5. Not only will the data record be kept permanently, but a data block will also be generated to validate the data.
6. Each data collected from the cloud collection is structured as meta-ancestor time data. After sending a meta-ancestor to the control system, the control system transfers hash data to the blockchain network and sends the original data to the cloud database.
7. At the same time, the control system will revert back to the cloud collection of some command data, these commands will also be converted to time, command format, and recorded in the blockchain.

Our research proposal intends to implement a global file-sharing space that allows users to obtain files from other users without the need for a file hash or a group to which the file owner belongs. Also, our implementation uses hybrid cryptography for file encryption that first encapsulates file contents using AES, then encrypts the AES key using the Rivest-Shamir-Adleman algorithm (RSA), providing dual-layer protection in case of a compromise. This is done in this order since AES is significantly faster when encrypting a large amount of data as compared to RSA, which is a

computationally heavy algorithm used to securely transfer the AES key.

C. Objectives

The objectives include showing the advantages and trouble-free nature of using decentralized cloud and Blockchain over centralized file-sharing applications, suggesting a model for file-sharing with enhanced privacy including a planet-wide efficient zero-redundancy cloud, and building a prototype website that demonstrates the proposal by sharing a few files live globally and peer-to-peer followed by downloading respectively.

The dApp allows users to upload and share files of any size through a fully decentralized system, which is powered by IPFS. IPFS is a peer-to-peer file-sharing protocol that allows for the distribution of files across a network of nodes without needing a centralized server. IPFS allows the Dapp faster file access, reduced bandwidth consumption, improved scalability and Automatic Login via Metamask compared to traditional centralized file-sharing systems [5]. Moreover, the files shared through IPFS are encrypted via Advanced Encryption Standard (AES) to provide total control over who can access them. This ensures that files are only accessible by authorized users and that the privacy of the users is protected.

D. Metamask

Metamask is a popular browser extension and mobile wallet that allows users to interact with decentralized applications (dApps) on the Ethereum blockchain. As a wallet, it allows users to securely store and manage their Ethereum and ERC-20 tokens. As a browser extension, it enables users to easily access and interact with dApps, such as decentralized exchanges, NFT marketplaces [6], and more. Fig.1 displays the various details generated for a transaction on metamask. One of the main problems with including blockchain is the latency associated with it which can be resolved by using Geth or Parity to create a custom exclusive client application like sepolia on blockchain which can be supported by metamask in just a few clicks.

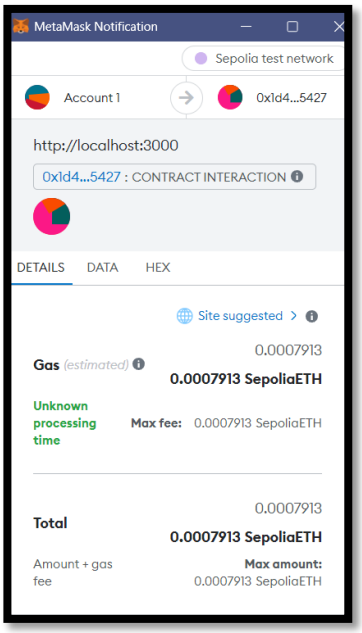

Fig.1 *Metamask transaction interface*

II. Proposed Solutions

When a new user wants to sign up to the website, they will have to enter a username (and a full name field). It is assumed that the user has a MetaMask account and is and is already logged into it and is accessing the website through the same browser window that has MetaMask installed providing the user with a metamask address as shown in Fig.2. A user's public and private key is generated using RSA. The public and private keys along with the username are sent to the node that has a contract named main deployed on it. The main contract will now take all three inputs (username, user public key, and user private key) and create a new user on the Ethereum blockchain. The block is then deployed using a transaction facilitated by the MetaMask wallet. In our solution, upon initiating a transaction, a signed transaction reaches an Infura endpoint which then redirects it to an address on the blockchain where the block is to be deployed.

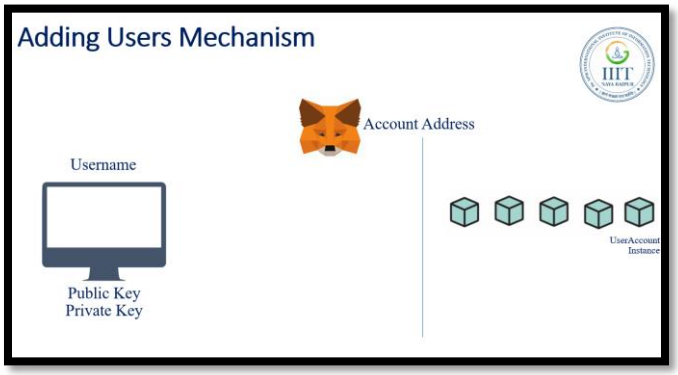

Fig.2 *Illustration of the mechanism of adding users*

After a user is logged in, in this case, we call the user a client, the client is given the freedom to create a group, to which it can add other clients as many as it wants/required. The non-owner clients get a notification/request in their dashboard regarding the group joins invitation. They can either accept or reject the group join request. In the same dashboard, the users can search for all the registered users on the website using usernames which on a successful fetch returns the user's MetaMask address and a list of common groups.

When creating a group, the user enters a group name. A group public-private key pair is created and sent to the user account instance already deployed on the blockchain. The instance hosts a function to create a group instance. The group instance is then created and deployed after a transaction through the MetaMask wallet. The group instance consists of a custom self-implemented data structure called FolderTree. Each node in the tree can have either a file or a folder as a child node. Every node is indexed as an SHA-3 Hash that takes the immediate parent address, owner MetaMask address, & Block No. as input so that the nodes are harder to compromise in case of an attack more precisely represented in Fig.3.

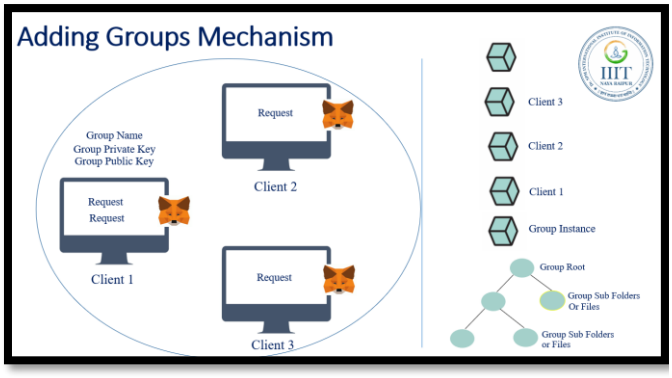

Fig.3 *Illustration of the mechanism of adding groups*

To add a file to the folder tree, we first convert the file to a file buffer (stream of bytes). The file before encryption is converted to a file buffer and is encoded. We then use AES to encrypt it since file sizes can be quite large and AES is a low-time complexity algorithm. The file is then uploaded on IPFS by calling an add function on the file. The file if successfully uploaded to IPFS, will return a hash that contains tabular information about what parts of the file are uploaded where on the IPFS. The AES key is encrypted using RSA. Now to the group instance where this file is shared, a write operation is initiated on the group instance block that adds the file directory relative to the Folder Tree, The IPFS hash, the filename, and the RSA encrypted key. Since this operation involves updating the block, a transaction is needed to complete this operation. On the website, a table created to store information about files uploaded in the group is called to add a row corresponding to the newly updated file

To download a file, from the group instance, the IPFS hash and the encrypted AES key (from the FolderTree), and the group private key are first sent to the client where the key is decrypted using the group private key. The IPFS hash is sent to the IPFS to retrieve a file buffer which will then be decrypted using the decrypted AES key, resulting in a decrypted file buffer. The file buffer can then be decoded before downloading the file in the comfort of your home.

To share a file, we first decrypt the encrypted AES key, and get the file IPFS hash from the FolderTree, get the recipient user instance, get their public key, decrypt the AES key of the file buffer using the public key as shown in Fig.4. Two transactions occur here: One is for adding a row that a file was sent from the sender and the second is for adding a row that a file was received by the recipient.

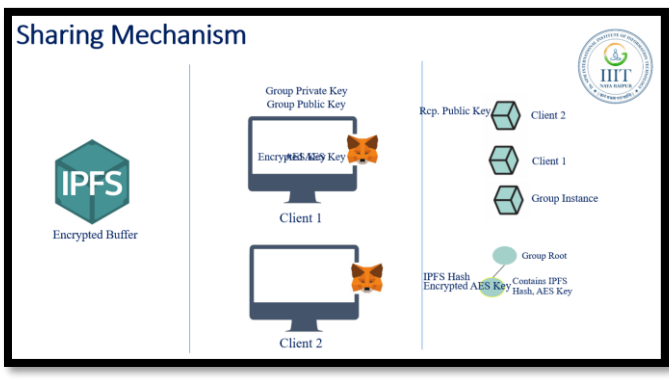

Fig.4 Pictorial Representation of the sharing mechanism

The user can upload, share or download files to the IPFS directly through the website without relying on any third party. The website keeps track of all the file activities of the client in a tabular format. The files here are encrypted using the user public key which was generated during the sign-up.

While sharing a file in the groups, the file key is encrypted using the group public key as the group private key is common and available to all group members during decryption. In peer-to-peer mode, the file key is encrypted using the recipient's public key.

In the global space that we implemented, the user has the choice to upload the files either anonymously or by using their username. The files are sorted by their extensions. The files can be directly opened on the website through IPFS since they are not encrypted. A real-time chat option is also available to all users so that they can request a required file from all the other users using their MetaMask addresses in the global space. Users can use the chat option as a message forum or use it for file sharing. One Transaction each is required to send a chat message and to add a folder.

When a file is deleted, the pointer variable corresponding to the file node marks the entire branch as invalid indicating not to search the branch for file retrieval in future requests, in order to account for a folder deletion.

III. Results

We have achieved the successful development of a decentralized web-based system that facilitates various functionalities, including account creation, automated login, messaging and chat-based transactions, folder creation, and group management with the ability to add members.

Automatic login system through metamask account address: This feature allows users to log in to the Dapp automatically using their metamask account address, which provides a secure and convenient way to access the Dapp.

Decentralized file sharing: The Dapp allows users to upload and share files of any size through a fully decentralized system, which is powered by IPFS. Files shared through IPFS are encrypted via Advanced Encryption Standard (AES) to provide total control over who can access them.

Team collaboration: The Dapp allows users to start a new project and invite other users to join, creating a decentralized collaborative workspace where team members can share, access, and manage files efficiently and securely.

Global and peer-to-peer file sharing: The Dapp allows users to share files globally and peer-to-peer, protected by Public and Private Keys of the users using RSA encryption[8]. IPFS leads to improved user experience in multiple cases. IPFS allows for distributed storage of data that is immune to altering and forgery[9].

Advantages of decentralized cloud and Blockchain: Decentralized cloud and Blockchain technology provide transparency, security, efficiency, and traceability, which are advantages over centralized file-sharing applications like Google Drive and Dropbox[10].

Comparison with centralized file-sharing applications: Centralized file-sharing applications have a restrictive nature and are vulnerable to data leaks. The Dapp is based on IPFS, which is a content-addressed system that lowers bandwidth consumption and improves file access speed[11].

Advantages in smart education: The Dapp allows students to access educational files like video tutorials and PDFs globally, and create groups for studying together and posting Q&A.

Security: Using a decentralized system to store files provides users with more security as files are at risk if they are in the hands of a single individual.

Global Space: The Dapp provides a global space where anyone can post any type of file (pdf, video, image, text)[12] to share with other users. Users can also request a particular content file in the global space by posting a request.

IV. Conclusion

In this report, we have discussed the privacy aspect of dApps based on Blockchain and IPFS and the advantages that they offer over centralized file-sharing applications. The use of Blockchain and IPFS in dApps allows for decentralized file sharing, team collaboration, and global and peer-to-peer file sharing, all while ensuring the security and privacy of users. Additionally, the use of dApps in the smart education and the creation of a global space for file sharing can lead to increased access to educational resources and opportunities for collaborative learning and a more open and decentralized repository of files.

Acknowledgment

First of all, we would like to thank the Almighty for giving us strength, wisdom, and courage and providing us the possibility to complete this report. This project is dedicated to our family for believing in us and for their love and support. We would like to express our most sincere gratitude to our Supervisor Dr. Anirban Bhowal for giving us the opportunity to be involved with such a subject that is interesting and modern and for his guidance. Special thanks to IIIT NR for providing the necessary facilities to implement the project successfully.

“We’ll never be able to give you back what you gave us”.

References

[1] Hsiao-Shan Huang, Tian-Sheuan Chang, and Jhih-Yi Wu, "A Secure File Sharing System Based on IPFS and Blockchain," in Proceedings of the 2nd International Electronics Communication Conference (IECC '20), New York, NY, USA, 2020, pp. 96-100, doi: 10.1145/3409934.3409948

[2] G. Solonas, Y. C. S., C. H., H. A., and N. K., "Secure Distributed Cloud Storage based on the Blockchain Technology and Smart Contracts," *iJournalse*, 2023. [Online]. Available: https://doi.org/10.28991/ESJ-2023-07-02-012

[3] K. Anusree, V. Jagan Sathiaseelan, R. Dev, and A. Abhinav, "Decentralized File Transfer System Blockchain-based File Transfer," *International Journal of Engineering Research & Technology (IJERT)*, vol. 11, no. 05, May 2022.

[4] N. Wadile, J. Shamdasani, S. Deshmukh, M. Sayyed, and S. Khandare, "Decentralized File Storage (Interplanetary File System) using Blockchain," *International Journal of Engineering Research & Technology (IJERT)*, vol. 12, issue March, 2023.

[5] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," *arXiv preprint arXiv:1407.3561*, Jul. 2014.

[6] Y. Niu, C. Li, and Y. Zhang, "Decentralized multimedia content distribution with blockchain," *Multimedia Tools and Applications*, vol. 77, no. 23, pp. 29785-29807, Dec. 2018.

[7] M. Zhou, Q. Xu, Y. Li, and Y. Liu, "A blockchain-based secure and privacy-preserving multimedia content delivery system," *Future Generation Computer Systems*, vol. 98, pp. 764-777, Nov. 2019.

[8] C. Liu, X. Zhang, Q. Wang, and J. Lu, "Blockchain-based trust management for multimedia sharing in IoT," *Sensors*, vol. 19, article no. 4268, Oct. 2019.

[9] H. Li, Y. Ren, and X. Jia, "A blockchain-based decentralized multimedia sharing system," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 5, pp. 27-40, May 2020.

[10] X. Zhang, X. Liu, J. Chen, and Q. Wang, "Blockchain-based secure multimedia transfer system," *Multimedia Tools and Applications*, vol. 78, no. 24, pp. 29689-29706, Dec. 2019.

[11] D. Tapscott and A. Tapscott, *Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World*, 1st ed. New York: Portfolio, 2016.

[12] H. Jin, et al., "BlockMedia: A Blockchain-Based Multimedia Content Delivery System," *IEEE Access*, vol. 7, pp. 19395-19405, 2019.