Proceedings of the Fourth International Conference on Trends in Electronics and Informatics (ICOEI 2020)
IEEE Xplore Part Number: CFP20J32-ART; ISBN: 978-1-7281-5518-0

Decentralized Cloud Storage Using Blockchain

Meet Shah
Department of Information Technology
St. Francis Institute of Technology
Mumbai, India
meetshah133@gmail.com

Mohammedhasan Shaikh
Department of Information Technology
St. Francis Institute of Technology
Mumbai, India
mdhasan.shaikh@yahoo.com

Vishwajeet Mishra
Department of Information Technology
St. Francis Institute of Technology
Mumbai, India
vishwajeetmishra1998@gmail.com

Grinal Tuscano
Department of Information Technology
St. Francis Institute of Technology
Mumbai, India
grinaltuscano@sfit.ac.in

***Abstract*—Cloud storage is one of the leading options to store massive data, however, the centralized storage approach of cloud computing is not secure. On the other hand, Blockchain is a decentralized cloud storage system that ensures data security. Any computing node connected to the internet can join and form peers network thereby maximizing resource utilization. Blockchain is a distributed peer to peer system where each node in the network stores a copy of blockchain thus making it immutable. In the proposed system, the user's file is encrypted and stored across multiple peers in the network using the IPFS (InterPlanetary File System) protocol. IPFS creates hash value. The hash value indicates the path of the file and is stored in the blockchain. This paper focuses on decentralized secure data storage, high availability of data, and efficient utilization of storage resources.**

Keywords— Blockchain, Data Security, IPFS, Encryption, Smart Contract, Cloud Storage.

I. INTRODUCTION

As per the Forbes article [1], 2.5 quintillion bytes of data are produced each day. Out of the total data in the world over 90 percent of data was produced in the last 2 years. With such a massive increase in the data, cloud storage is required to store the data. Much of the data currently available through the internet is quite centralized and is stored with a handful of technology companies that have the experience and capital to build massive data centers capable of handling this enormous data. The problem with this approach is the security of data. As this data is stored in a centralized manner, if an attacker can gain access to the server he can easily view and modify the data. Another problem with this approach is the privacy of user data. In many instances, this data is used by third parties for data analysis and marketing purposes. Also, the cost incurred in storing data in centralized servers is more and many times users have to pay for the entire plan which they have selected even if they have used only a fraction of storage portion thus it does not provide flexibility to the user to pay only for what they are using. Another issue is the scalability of the system, it is difficult to scale a centralized storage system to meet the increasing demand. With zero trust two parties can transact in Blockchain.

II. PROBLEM IDENTIFICATION

Data privacy and security are concerns when data resides third party storage. Storage can be created from the underutilized resources of peers. Data security, privacy, availability, and resource utilization are the areas handled by the proposed system.

III. LITERATURE SURVEY

Zhe, Diao, et al [2], discusses the increasing demand for cloud storage with associated security and privacy issues in centralized cloud storage. As per the discussion by encrypting the data and scattering the data across multiple nodes, a high level of data security can be achieved. Lee et al [3], has shown encryption enhances the security of user's data stored in cloud storage. Authors have used the AES encryption algorithm to enhance security with speed without impacting the system's performance. Nakamoto, Satoshi [4], uses the concept of bitcoin in blockchain technology to show transaction records. The transaction details are stored in the blocks and are chained to each other serially, using the concept of hashing. Every peer involved in the network has a copy of the blockchain to verify the credibility of the blockchain. The author claims that the transactions stored in the peer to peer network are tamperproof, cannot be altered by an attacker and the identity of all the parties involved in transactions is secure. The peer-to-peer network uses 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. Zyskind [5], raises the problems related to centralized cloud storage and suggest blockchain to

solve the issues. The proposed system allows two transactions viz, Taccess and Tdata, Taccess for access control permission which will be set by the respective user who owns the data, and Tdata are used for data storage purposes. The shared encryption key secures the data from third parties. Cachin, Christian et al [6], discusses the architecture of hyper ledger blockchain fabric, limitations of electronic coins, working of hyper ledger fabric, and proof of work consensus algorithm. Hyperledger Fabric is a permissioned blockchain network that allows only limited nodes that have permission to add new blocks in the blockchain. In paper [7], the author explains the architecture of Ethereum and the working of smart contracts. Bitcoin was used only for sending and receiving cryptocurrency but lacked to add business logic. Ethereum applications like decentralized file storage and decentralized autonomous systems are discussed. Ruj, Sushmita [8], proposes BlockStore, a decentralized framework using blockchain technology to enhance security, transparency in transactions between peers (Host and Renters). The system uses proof of storage and proof of work to verify that hosts do not meddle with data in Blockchain. The proposed system does not encrypt or decrypts data before uploading it to peers which creates a threat to confidentiality and privacy of user's data. Juan Benet et al. [9], introduces a new peer to peer file transfer protocol called IPFS (InterPlanetary File System). IPFS uses a content-based addressing scheme. As per the author, IPFS provides a high throughput content-addressed block storage model along with content-addressed hyperlinks. Li, Dagang [10], discusses how data sharing in blockchain-based applications differs from traditional applications. The author identifies that data-sharing in decentralized architecture is cumbersome. The author proposes Meta-key for secure data sharing in a decentralized storage system based on blockchain also focuses on the collusion-free property of the proposed cryptographic protocol and proved it strictly. Wohrer et al. [11], explains the solidity used for creating smart contracts in blockchain and its difficulty. All the security issues which have been resolved are 1.Checks-Effects Interaction, 2.Emergency stop, 3.Speed bump, 4. Rate limit, 5. Mutex and 6. Balance limit. The knowledge related to these issues can be found in grey literature and many blog articles. In [12], blockchain provides scalability, security, and sustainability, it is also helpful to transform the way of doing business. In this paper, the author is trying to conduct a comprehensive survey on the technical and application of blockchain technology by discussing its structure to different consensus algorithms. The author has also explained, the structure of blockchain consists of data, timestamp, and address of the previous block in hash form. The timestamp is recording the time when the block was created. A hash function is the one that takes an input of any length and generates the output with a unique fixed length. Each block contains a hash value of the previous block. therefore, security is increased in Blockchain. It uses proof of stack (POS), proof of work(POW) consensus algorithm as a measure to discourage the attacks of Denial of Service and miner can validate transactions in a block depending on the amount the user holds respectively. Therefore, blockchain technology is exceeding recognized and appraised due to its decentralized infrastructure and peer-to-peer nature. In paper [13] D. Sivaganesan has suggested the use of blockchain to improve security and provide transparency in IoT applications. The author has proposed a smart logistic system for the pharmaceutical sector to keep track of the shipment for medicines. This system combines blockchain and IOT, it also includes a smart contract to bond manufactures, distributes, and the dispenser legally. The use of smart contract avoids third person intervention as well as provides improved security and transparency in the transaction.

In the proposed system, smart contracts are used to store file details in the blockchain and also transfer the cryptocurrency from the user's wallet to the peer's wallet. AES encryption algorithm for enhancing the security of user's data stored in cloud storage. The proposed system maps the user's wallet address with the user's file so that only the legitimate owner can access the file's data. Users' information is stored in Ethereum blockchain. The Ethereum blockchain network allows the use of smart contracts through which information of file uploaded by the user is stored in the blockchain. The proposed system encrypts and decrypts data every single time for upload and download operation. The system uses the IPFS Protocol to distribute files efficiently across several peers in the network.

IV. METHODOLOGY

The proposed system works in four modules shown in fig 1. The user first creates an account on the metamask. The user's account address and wallet balance are fetched in the app through web3.js from the metamask. Users select the file to upload through file picker. System checks for the number of available peers. Further, the AES algorithm uses the user's wallet address as a key and encrypts the uploaded file. A payment dialogue seeks for the user's confirmation. On confirming the payment, the user's file is stored across available peers using IPFS protocol. IPFS then returns a hash value consisting of the path of the file. This path is then mapped with the user's address using a smart contract and gets stored securely in the blockchain. To achieve high availability and reliability of data, the uploaded data is replicated on three peers. For better performance system blacklists peers every time they are unavailable for data retrieval.

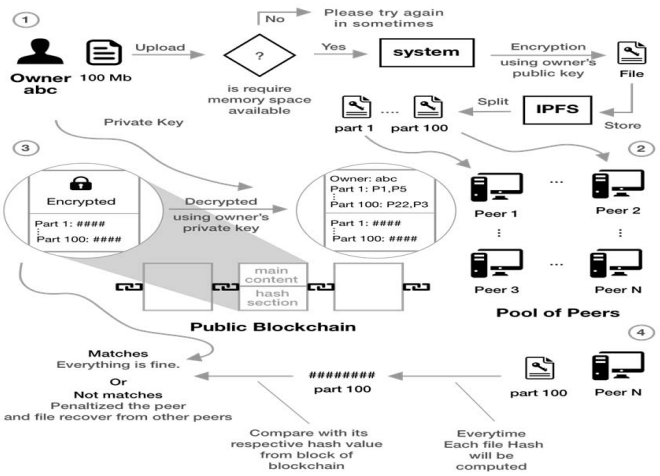

Fig. 1. System Design for Decentralized Cloud Storage

The terminology is briefly discussed below-

Metamask: Browser extension which acts as a bridge to connect with the ethereum network.

Ethereum network: It is an open-source, public blockchain-based distributed computing platform. Ethereum uses smart contracts where one can add business logic to make decentralized applications as per the business requirements.

Peers: These are the users of the system who have pledged to rent their free storage for another user's to store files.

AES: Advance Encryption Standard (AES) is a symmetric-key algorithm that supports block length of 128 bit and can have a key size of 128, 192, and 256 bits.

IPFS protocol: IPFS is an open-source peer to peer file transfer protocol.

A. Uploading of file

User uploads file using the file picker. The system checks the file size and ensures storage availability in the network. The file is uploaded when enough storage is available. Then system performs step B. Users are notified to try again when enough storage is unavailable.

B. Encryption of file

The uploaded file is encrypted using AES 256 bit algorithm. The encryption key is generated using the user's wallet address and randomly generated salt value. This encryption key along with an IV is used to encrypt user's data. This maintains the confidentiality of the user's data.

C. Storing of file across multiple peers

The encrypted file is then divided into blocks of 64KB and sends to different peers across the network with the help of the IPFS protocol. The proposed system uses a private IPFS network to allow registered peers to store the file in the network. The file block is replicated on multiple peer's storages for high availability using the IPFS cluster.

D. Storing of file across multiple peers

IPFS returns a hash value which indicates the path of the file. The hash value along with metadata is mapped with the user's wallet address and is stored in the blockchain using a smart contract. Smart contracts are like agreement and are used to eradicate the need for a third party. They control the transaction between nodes or assets between parties under certain conditions. This is lines of code stored on a blockchain network and are automatically executed when predetermined terms and conditions are met.

In our proposed system preconditions for the smart contract to execute are:

1) Enough Space is available in the network to store files.
2) The user has sufficient wallet balance to pay the peers.

```
pragma solidity ^0.5.0;

contract DecentralizedDrive {
    struct FileDetails {
        string hashValueOfFile;
        string nameOfFile;
        uint dateofTransaction;
        string typeOfFile;
        }

  //Mapping file details with users account
    address

    mapping(address => FileDetails[]) files;

     function addFile (string memory filehash,
    string memory fileName,  string memory fileType,
     uint date) public {
        files[msg.sender].push(FileDetails({
    hashValueofFile:filehash
        , nameOfFile:fileName, typeOfFile:fileType,
    dateOfTransaction: date}));
        }

    function getFile(uint position) public view
    returns(string memory, string memory, string
    memory, uint) {
        FileDetails memory file = files[msg.sender][
    poisiton];
        return (file.hashValueofFile, file.
    nameofFile, file.typeOfFile, file.
    dateOfTransaction);
    }

  }
```

Fig. 2. Smart Contract to store file details

The above smart contract stores all the files details in the structure named FileDetails and maps this structure with the user's address. It consists of two functions, one to add a new file and another to get the details of the uploaded file.

E. Paying the peers for file storage

Once the file is stored across peers, total cryptocurrency is calculated and is deducted from the user's wallet. This cryptocurrency is first transmitted to the smart contract from the user's wallet. With the smart contract, this amount is distributed to the peers who have stored the user's file.

```
pragma solidity ^0.5.0;

contract DecentralizedDrive {
    mapping(address => uint256) public deposits;

    function depositsFund(address[] memory payee)
    public payable {
        uint256 amount = msg.value;

        for(uint i=0;i<payee.length;i++)
        {
        deposits[payee[i]] = deposits[payee[i]] + (
    amount)/payee.length;
        }
    }

    function withdraw(address payable payee) public
    {
        uint256 payment = deposits[payee];
        deposits[payee] = 0;
        payee.transfer(payment);
    }

    function checkBalance() public view returns(uint
    )
    {
        return address(this).balance;
    }
}
```

Fig. 3. Smart Contract to transfer payment to peers

The above smart contract consists of two methods depositFund() and withdraw(). Users can transfer the payment to peers via a smart contract. The user has to first deposit funds into the smart contract using depositFund() function. This payment is then transferred to peers via smart contract by withdraw() function.

V. RESULTS

A. Designed System

To access the system, users first sign up on metamask and login with the registered credentials. Successful login takes users to the home screen for selecting the file to upload.

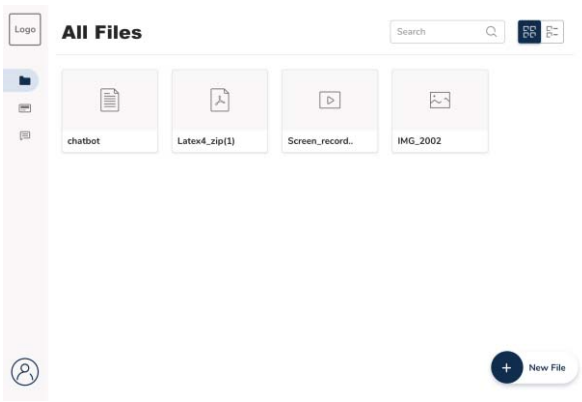

Fig. 4. GUI to upload the files

System checks for storage availability based on selected file size. The selected file is encrypted using AES 256 bit algorithm when sufficient storage is available. The system will compute the total cost of storing the file. Once the cost is the calculated system will check if the user's wallet balance is more than the calculated cost. If the user has sufficient balance then he/she is prompt to pay the cryptocurrency to store the file.

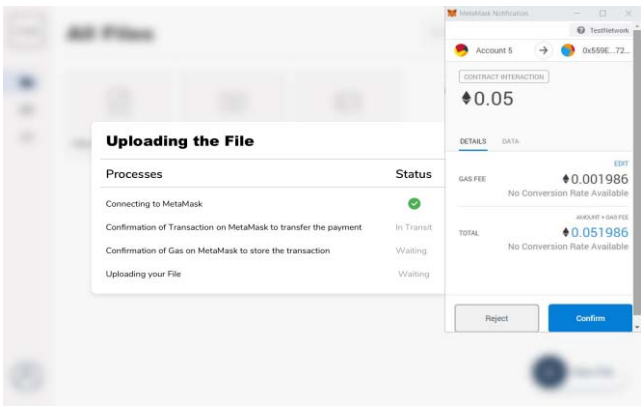

Fig. 5. Payment Confirmation

After a successful payment file is split into blocks and store across peers using IPFS protocol and the corresponding hash value is stored in the blockchain.

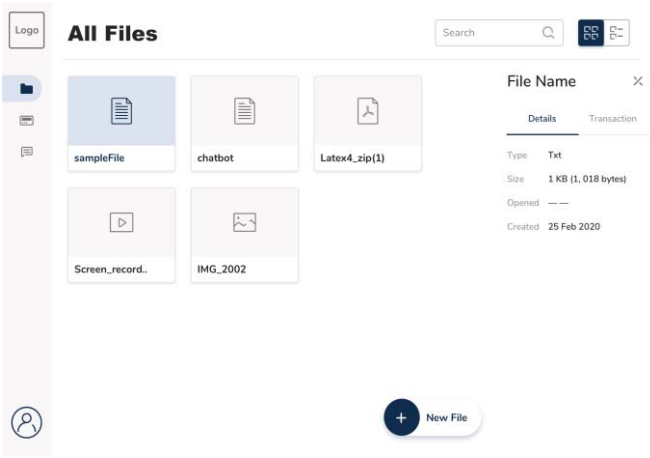

Fig. 6. File uploaded successfully

Fig. 6 represents successfully uploaded file details. Once the file is successfully uploaded ipfs returns a hash value indicating the path of the file. This will be mapped with the user's wallet address and will be stored in the blockchain with the help of a smart contract.

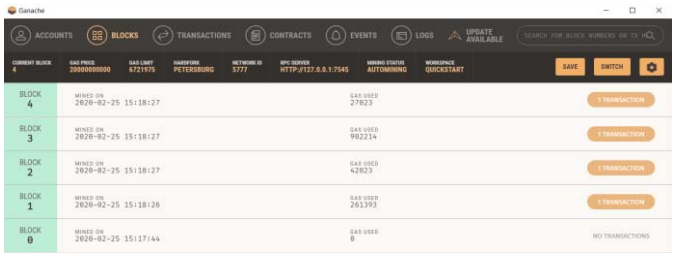

Fig. 7. Creation of blockchain

VI. ANALYSIS

With the help of the AES 256bit algorithm user's files are encrypted which increases the confidentiality of the user's data as shown below.

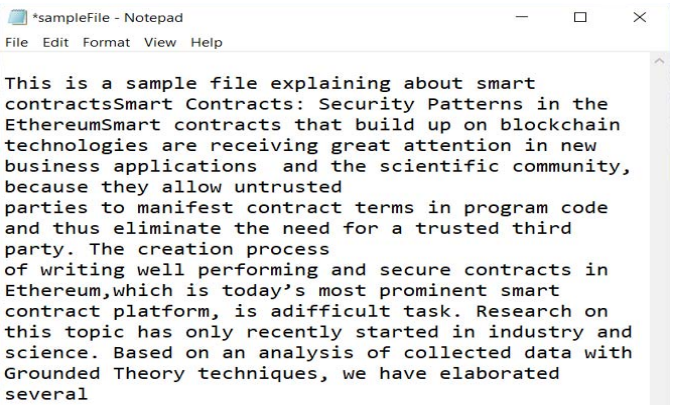

Fig. 8. Sample File before Encryption

Fig. 9. Sample File after encrypting using AES 256bit algorithm

The time required to upload files depends on the file size and availability of peers. As the file size increases the time required to upload the file increases. As shown in the figure, below as the file size increase from 32 KB to 36 MB the time required to upload the file increase from 12 sec to 80 sec.

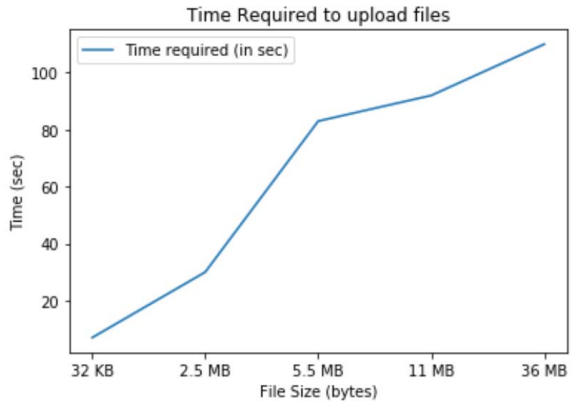

Fig. 10. Analysis of time required to upload file

The total time required to upload the file decreases with increases in the number of peers. More the number of peers available on the network less time is required to upload the file. Fig. 11 represents the total time required by 2.5mb file gradually decreases as the no of peers increases.

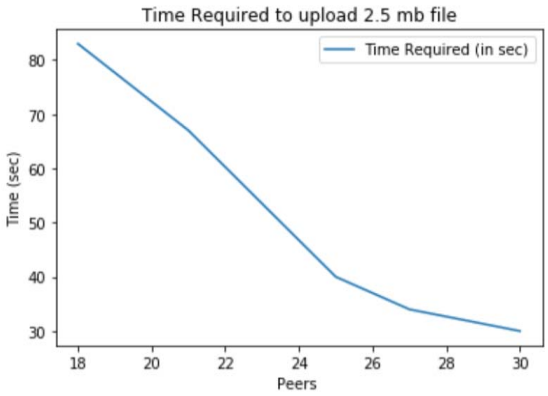

Fig. 11. The time required to upload 2.5 MB concerning the number of peers.

Fig. 12 and 13 show the replication of the file on multiple peers. Replication of the file on multiple peers helps the proposed system to achieve reliability. Users' files can be retrieved even if a peer is offline.

Fig. 12. The file system of peer-1 where blocks of encrypteduser's file are stored.

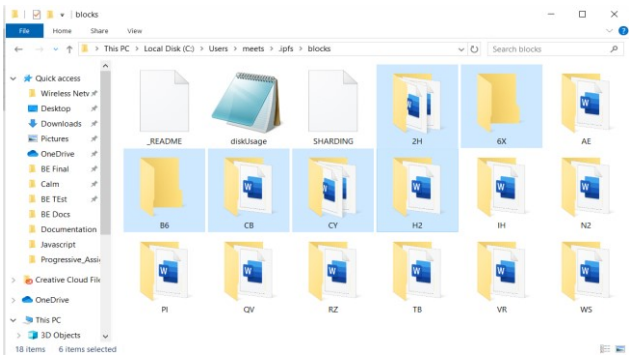

Fig. 13. The file system of peer-2 where certain blocks of encrypted user's file are replicated.

VII. CONCLUSION

The proposed system enhances the security of data by encrypting and distributing the data across multiple peers in the system. Implemented system uses the AES 256bit encryption algorithm to encrypt the data ensuring the confidentiality of the user's data. Encrypted data is then distributed and stored across peers in the network using the IPFS protocol. Our system not only solves the privacy and security concerns of centralized cloud storage but also provides a medium for the peer to rent their underutilized storage and earn cryptocurrency in return thereby, maximizing the storage resource utilization.

VIII. FUTURE SCOPE

In the future, an adaptive scheduling algorithm can be incorporated with which files can be accessed multiple times by the user as compared to the one which is accessed rarely. This will help to ensure that frequently accessed files are available easily to the user whenever required. Also, a credit system can be added with which each peer will be assigned a default 100 credit, based on their system uptime, and several successfully served file access that requests their credits will be either deducted or added. Peer's with more credits will be given higher priority for data storage.

REFERENCES

[1] Bernard Marr, "How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read." Forbes, 2018.

[2] Zhe, Diao, "Study on Data Security Policy Based On Cloud Storage" 2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids) IEEE, 2017.

[3] Lee, Bih-Hwang, Ervin Kusuma Dewi, and Muhammad Farid Wajdi. "Data security in cloud computing using AES under HEROKU cloud." 2018 27th Wireless and Optical Communication Conference (WOCC). IEEE, 2018.

[4] Nakamoto, Satoshi,"Bitcoin: A peer-to-peer electronic cash system", (2008).

[5] Zyskind, Guy, and Oz Nathan," privacy: Using blockchain to protect personal data",IEEE Security and Privacy Workshops. IEEE, 2015.

[6] Cachin, Christian," Architecture of the hyperledger blockchain fabric", Workshop on distributed cryptocurrencies and consensus ledgers. Vol. 310. 2016.

[7] Buterin, Vitalik, "A next-generation smart contract and decentralized application platform", white paper (2014).

[8] Ruj, Sushmita, et al, "BlockStore: A Secure Decentralized Storage Framework on Blockchain" 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA). IEEE, 2018.

[9] Benet, Juan, "IPFS - Content Addressed, Versioned, P2P File System." 2014.

[10] Li, Dagang, et al. Meta-Key: "A Secure Data-Sharing Protocol Under Blockchain-Based Decentralized Storage Architecture", IEEE Networking Letters 1.1 (2019): 30-33.

[11] Wohrer, Maximilian, and Uwe Zdun,"a Smart contracts: security patterns in the ethereum ecosystem and solidity", International Workshop on Blockchain Oriented Software Engineering (IWBOSE). IEEE, 2018.

[12] Sum, V. "SECURITY AND PRIVACY MECHANISM USING BLOCKCHAIN." Journal of Ubiquitous Computing and Communication Technologies (UCCT) 1.01 (2019): 45-54

[13] Sivaganesan, D. "BLOCK CHAIN ENABLED INTERNET OF THINGS." Journal of Information Technology 1.01 (2019): 1-8.