A Survey on File Storage & Retrieval using Blockchain Technology

Yash Ranka[1], Jainam Bagrecha[2], Kavish Gandhi[3], Bhargav Sarvaria[4], Prof. P. M. Chawan[5]

[1,2,3,4]U. G. Student, Department of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India
[5]Associate Professor, Department of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India

---

Abstract - *Data drives business around the globe and is a crucial aspect of the industry, which makes misuse of data equally dangerous. Data needs to be stored and transferred securely in order to maintain its confidentiality and integrity. Ethereum platform and technologies like swarm and whisper can make it possible to make a secured file storage and retrieval Decentralized Application (DApp). With the help of blockchain we propose an architecture which is much safer and easier for human use and it also solves the existing data storage and retrieval difficulties.*

***Key Words*: Blockchain, Swarm, Whisper, Ethereum, File Storage & Retrieval, P2P, IPFS, ENS**

1. INTRODUCTION

A blockchain is a chain of blocks, where each block contains a set of records (transactions) and the blocks are linked using cryptography[1]. Blocks hold batches of valid transactions. The transactions are hashed and encoded together into a Merkle tree. Every block includes the cryptographic hash of the previous block in the blockchain which links and creates a chain of blocks. The linked blocks form a chain. This iterative process confirms the integrity of the previous block, all the way back to the original genesis block.

There are many applications of blockchain viz. Cryptocurrency, to secure Internet of Things, smart

Contracts based transactions. One promising application of blockchain is File Transfer (Storage and Retrieval), which can be implemented using Peer-to-Peer network, Directed Acyclic Graph, Distributed Hash Table.

2. EXISTING TECHNOLOGIES

2.1 Distributed File System

Traditional file storage on a single computer can be made more efficient using Distributed File System

(DFS) [2]. Data which is stored on a single computer can be split and stored on multiple independent computer (nodes). DFS makes file storage reliable by allowing an user to retrieve data even when one or more nodes of the network goes down, the data is replicated on several nodes which increases availability of data, only authenticated users can store and retrieve data from DFS which makes it more secure and if the system runs out of space we can add more nodes to the system which makes DFS scalable.

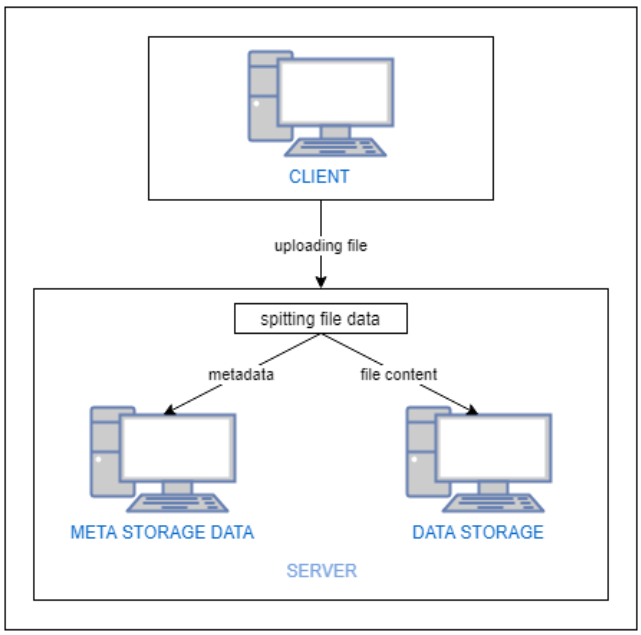

Fig -1: DFS Architecture

A file is split into metadata and its content and stored on the system while uploading. When a user wishes to download the same file, he provides the Meta information and the file is retrieved from multiple nodes and provided to the user, this makes it appear as a single coherent system for the user. Thus for a user it appears to be stored as it were being stored on his own computer. Thus DFS is reliable, scalable, secure and more available but reduces the time of access for a file. DFS requires algorithms for synchronization, locking and maintaining data consistency.

2.2 InterPlanetary File System

The InterPlanetary File System (IPFS) [3] is a peer-to-peer distributed file system that seeks to connect all computing devices with the same system of files. The file is divided into small blocks, the block is then hashed and distributed across the IPFS network. The redundant information is removed because the hashes are same for similar files. IPFS provides a high throughput content-addressed block storage model, with content-addressed hyperlinks. A generalized Merkle DAG (Directed Acyclic Graph), which can be used to build versioned file systems, blockchain, etc. A distributed hash table, an incentivized block exchange, and a self-certifying namespace is used by IPFS. There is no point of failure in IPFS and it is not necessary for the nodes to trust each other.

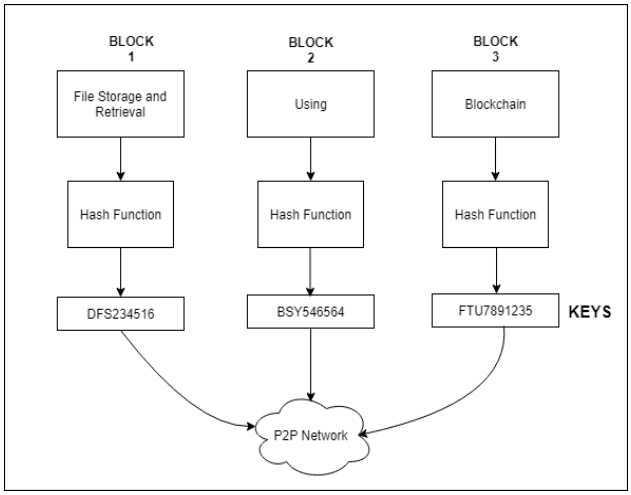

Fig -2: Overview of IPFS

2.3 Secured File Storage and Retrieval using Blockchain

One of the simplest way to store/retrieve files is to upload the file directly in the blocks of a blockchain. The complexity to verify a block is directly proportional to the size of the block and hence, larger file size will lead to bigger blocks which eventually makes the network inefficient with high latency. Though the file remains secured because of decentralized nature of blockchain, we cannot use it practically.

2.4 Storing the hashes of files on Blockchain

One way to increase the efficiency of the file storage/retrieval is to use the hash of the file on the blockchain instead of using the blocks of file. It further provides auditable accountability of exactly what content has been authorized by whom for sharing and transfer through the file transfer guard. Continuous updating the ledger for every file transfer gives an immutable record of the entire life cycle of file. [4]

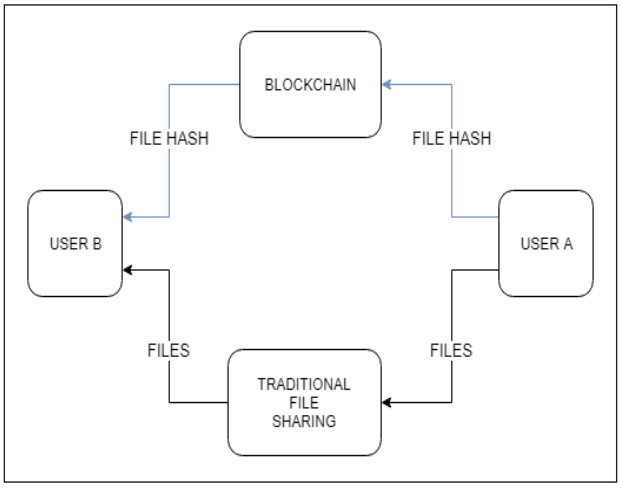

Fig -3: Storing hashes on Blockchain

2.5 Filecoin

Filecoin [5] is a kind of cryptocurrency where individual data providers are provided with incentives for their storage resources for storage of data pieces in the filecoin network. The incentives are given based on the amount of resources shared by the individual. Large number of such individuals form a global distributed network that runs filecoin protocol for file storage. The contributing users may opt out if they wish and yet the availability of the data is maintained by the distributed network.

Filecoin makes use of two kinds of nodes: Storage nodes & Retrieval nodes. The file is first encrypted and then segments of the file are distributed to storage nodes. The retrieval nodes need to be near the storage nodes so that the transfer of data is fast. Only the authorized person can retrieve it using her private key.

2.6 Storj

A peer-to-peer (P2P) cloud storage network which uses client-side encryption allows users to transfer and share data without depending on a third party storage provider. Data availability is a function of popularity, rather than utility which makes the P2P networks unfeasible for production storage systems. A solution in the form of a challenge-response verification system coupled with direct payments is used by Storj [6].

The file is encrypted and sharded. The shards are then distributed over a decentralized network of storage nodes which are known as “farmers”. The data owners are responsible for everything from pre-processing shards to collecting them, managing file encryption keys, etc.

2.7 Decentralized file transfer using Multichain framework [7]

This is a file transfer system where the receiver first sends a request message to the sender with body as burn address and its public key via email or facebook. The RSA 2048 algorithm is used to generate public-key private- key pair. The file is encrypted using Advanced Encryption Standard (AES) and the public key. Then the encrypted file is sharded into fixed size blocks and then encoded using Hex encoding algorithm. The encoded parts and the RSA public key is then stored on a private blockchain. At the receiver’s end, the different blocks of file are decoded using hex decoding and merged together. The merged file is decrypted using the private key (of the respective public key) and AES to produce the desired file. [8]

3. PROPOSED SYSTEM

Our primary aim lies in solving the problem of decentralized data storage, restricted data access and preventing data redundancy using peer-to-peer network storage clubbed with blockchain technology. The proposed

tech stack for the same includes Ethereum[9] swarm[10] for data storage, distribution and retrieval over the network, Ethereum whisper[11] protocol for sharing access rights by the file owner and various Asymmetric cryptographic algorithms along with data compression algorithms for maintaining data confidentiality and efficient data storage. The process is as follows:

We take the file which user needs to upload and encrypt it using a new public-key private-key pair which will be held by the file owner. The Asymmetric encryption technique can help ensure the read-write access privileges. The user with just public key can have read access and the user with private key can have write access. Symmetric encryption algorithms can also be used as per the requirements. Once the file is encrypted it is further processed for data compression using a predefined data compression algorithm for efficient file storage. The compressed and encrypted file is then distributed over the swarm network and a unique hash for the file is calculated by the swarm protocol through which the file can later be accessed.

Once the file is distributed over the swarm network, the generated hash for the file is then added to the blockchain as a transaction to preserve the file integrity. Thus we record the hash of the file by uploading it on the blockchain.

Once the hash of the file is recorded on the blockchain, the file owner can then decide to grant access privilege to various other users using a messaging protocol- Ethereum Whisper. The owner needs to send the file hash along with the public key of the encrypted file by which the other user will be able to decrypt the encrypted file and hence access it and the entire message is encrypted using the receiver's public key, making the message unintelligible for others.

The receiver can then decrypt it using her private key. Thus even if someone tries to access the encrypted file by fetching the file hash from the public blockchain she won't be able to decrypt the file without the key. Thus, preserving the confidentiality.

The problem of data redundancy has been automatically solved as we calculate a unique hash for each file. Thus, if we have two exactly same files both files will effectively produce the same hash thus solving the redundancy conflicts.

To make the file system more human-readable we make use of the Ethereum Name Server (ENS). ENS works the same way as the Domain Name Server, it effectively stores the mapping of file hashes with the human readable filenames using distributed hash tables. Thus the hashes can be uniquely named as per the user convenience. Along with improving readability, ENS serves an important purpose. Generally the files stored this way are immutable and static in nature as a slight change in file content would effectively produce a new hash which will again increase the redundancy of the network. ENS solves this by efficient mapping from old files to new files, thus adding up more to the user convenience. ENS can also be used to maintain version control (like git).

The above process considers file uploaded to be the one with limited access where the file owner shares the access permissions with the limited set of users. However, some files could be made available to general public where we do not need to specify the access permissions. Such files can be directly compressed and stored over the network. The users who wants the access of such public files can directly access it on the blockchain via some blockchain explorer that searches for file hashes on the blockchain. To make it more convenient for users such files can be explicitly marked as public with the help of ENS.

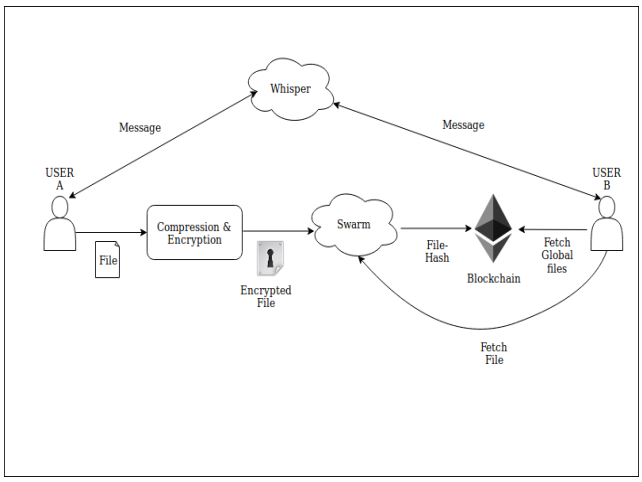

Fig -4: Proposed System Architecture

4. CONCLUSION

In this paper we proposed an enhanced mechanism for data storage and transfer. The whole mechanism uses cryptography, blockchain and ethereum technology to carry out the task as opposed to older mechanisms. The given mechanism will make the task more secure through cryptography, maintain data integrity through hash codes and data readily available with decentralized data storage on multiple hosts.

REFERENCES

[1] Blockchain-Wikipedia "https://en.wikipedia.org/wiki/Blockchain"

[2] Pavel Bžoch - "Distributed File Systems", "https://www.kiv.zcu.cz/site/documents/verejne/vyzkum/publikace/technicke-zpravy/2012/tr-2012-02.pdf"

[3] J. Benet, "IPFS - content addressed, versioned, P2P file system, (2014)", "https://github.com/ipfs/ipfs/blob/master/papers/ipfs-cap2pfs/ipfs-p2p-file-system.pdf"

[4] Securing your cross-domain file transfers with blockchain, "https://www.ibm.com/blogs/blockchain/2018/05/securing-your-cross-domain-file-transfers-with-blockchain/"

[5] Filecoin: "A Cryptocurrency Operated File Storage Network" , "https://filecoin.io/filecoin-jul-2014.pdf"

[6] Storj - A Peer-to-Peer Cloud Storage Network, "https://storj.io/storj.pdf"

[7] Dr Gideon Greenspan, "MultiChain Private Blockchain", "https://www.multichain.com/download/MultiChain-White-Paper.pdf"

[8] SriBalaji, Vignesh Mohan, Soundarya, "Secure and Decentralized File Transfer Application using Blockchain" , "http://troindia.in/journal/ijcesr/vol4iss4/169-175.pdf"

[9] Vitalik Buterin, "Ethereum" , "https://ethereum.org/"

[10] Ethereum Swarm, "https://swarm-guide.readthedocs.io/en/latest/introduction.html"

[11] Ethereum whisper, "https://github.com/ethereum/wiki/wiki/Whisper-PoC-2-Protocol-Spec"