Journal of Science, Computing and Engineering Research (JSCER)
Volume-8, Issue-3, March 2025.
DOI: ***https://doi.org/10.46379/jscer.2025.080303***

Blockchain Based Inter-Organizational Secure File Sharing System

[1]Jayendra Kumar, [2] Devulapally Yathish, [3] Dasari Ragini, [4] Chindam Sreeja

[1]Assistant Professor.Department of CSE,Anurag University,Telangana,India

[2,3,4] Student,Department of CSE,Anurag University,Telangana,India

Article Information

Received	:	10 Feb 2025
Revised	:	15 Feb 2025
Accepted	:	28 Feb 2025
Published	:	04 Mar 2025

Corresponding Author:

Chindam Sreeja

Abstract**—** A consortium of organizations collaborates and exchanges information to create synergies in their operations. Centralized systems of secure transferring of data cannot provide distributed trust and transparency. Blockchain technology can be used to transfer data securely and transparently. This paper proposes a blockchain based secure transferring of data. It can be used by a consortium of organizations to securely exchange files in a distributed fashion. Hyperledger Fabric, an enterprise blockchain framework, is used for blockchain network setup and the development of smart contracts. The Inter Planetary File System (IPFS) is used for storing files in a distributed way. The paper provides the workflow for identity management and file-sharing processes. The proposed system allows a consortium of organizations to share files with confidentiality, integrity, and availability using blockchain.

Keywords: Blockchain, IPFS, file-sharing

Citation: Jayendra Kumar, Devulapally Yathish, Dasari Ragini, Chindam Sreeja. "Blockchain based inert-organizational secure file sharing system", Journal of Science, Computing and Engineering Research, 8(3), March 2025.

I INTRODUCTION

Due to the introduction of the cryptocurrency Bitcoin, blockchain has become one of the most popular technologies. Every transaction that occurs in the network is recorded in the blockchain. The transactions are validated by the members of the blockchain network. The transactions are grouped into blocks, and the blocks are linked to each other using cryptographic mechanisms. This linking of trans actions makes the transactions tamper-proof. Bitcoin provides a peer-to-peer, distributed, anonymous banking platform.

The blockchain participants, i.e., Miners in the case of bitcoin, validate the transactions and create a block for a set of transactions. The miners are rewarded for their validation services in the form of bitcoins. This validation mechanism is known as the consensus protocol. The consensus protocol used by bitcoin is known as proof of-work (PoW), in which all the Miners compete to solve a mathematical problem. The Miner, who is the first one to solve the problem, gets the privilege to add a block to the blockchain and obtain the corresponding reward in the form of bitcoins.

The Bitcoin network is a public, permissionless blockchain network, i.e., anyone can join the bitcoin blockchain network. Blockchain provides data integrity, transparency, and trans action automation through smart contracts. These features of the blockchain can be leveraged in other non-cryptocurrency based enterprise applications. Information sharing/distributed file storage applications could effectively exploit these features. There are solutions and that provide distributed storage space through blockchain in exchange for crypto tokens. All of these solutions operate in an open/public blockchain network environment. For enterprise applications, permissionless blockchain is not suited due to a lack of access control and user accountability. A consortium blockchain is a type of blockchain that allows a consortium of organizations to form a blockchain network in a closed and controlled environment. With pre-identified and designated nodes, the blockchain network validates the transaction blocks and ap pends the corresponding blocks to the blockchain ledger.

The blockchain can be used to store the history of transactions. It cannot be used to store large amounts of data. IPFS (Inter Planetary File System) can be used to overcome this limitation. IPFS is a peer-to-peer distributed database. IPFS is a content-addressable file storage system. i.e., each file stored in the IPFS can be accessed using a content identifier, which is derived from the actual content of the file.

The paper is organized as follows: Section II describes the related work in the area of file-sharing using blockchain technology. Section III describes the proposed system in detail, including the workflow for identity management and file sharing. Section IV describes the experimental setup and test results of the implemented system.

II. RELATEDWORK

Naz [3] et al. have proposed a system of sharing data using Ethereum[10] and IPFS. The data owner shares data with certain anonymous recipients in exchange for Ethers. The document, along with document metadata, is written to IPFS, and the document's content id (CID) is split, encrypted, and written to the blockchain. Some nodes are designated as workers and are responsible for decryption services. Further, there is a review system to determine the quality of the data. Liu[4] et al. has proposed data sharing mechanism using blockchain and cloud storage.

The indexes of the data are stored in the tamper proof blockchain ledger. Access control is provided through smart contracts deployed on the blockchain. Satapathy[5] et al. have proposed a secure authentication architecture based on open-source Hyperledger Fabric [2] and blockchain technology. Each of the users and IoT devices are registered on the blockchain using smart contracts.

All of the entities could be securely authenticated with one another. Sari [6] et al. had proposed a peer-to-peer group-wise file sharing system using Ethereum and IPFS. All the users must be enrolled in order to share the files. Group-level access control is configured to securely share the files.

Table I Comparison Of Existing Approaches With The Proposed System

Approach	Parameter			
	E2E Encryption	Platform	Data Storage	Application Domain
Naz[3]	No	Ethereum	Distributed	File Sharing
Liu[4]	Yes	Ethereum	Centralized	Medical Records
Satapathy[5]	No	Hyperledger Fabric	Not Specified	IoT
Sari[6]	Yes	Ethereum	Distributed	File Sharing
Proposed System	Yes	Hyperledger Fabric	Distributed	File Sharing

Table I compares the proposed system with the existing file sharing approaches using blockchain.

III. THE PROPOSEDSYSTEM

In a consortium of organizations a number of organizations can share data in the form files and synergies their operations. Figure 1 illustrates the high-level view of the proposed system. A blockchain network created among organization1, orga nization2 and organization3. Each of the organizations will host IPFS node, Identity and Interfacing Server (IIS), Smart contract, and blockchain ledger.

The IPFS node is hosted to form the distributed storage network. IIS maintains the identity details in identity database and is also the interfacing point with the smart contract. A smart contract is a program, which contains the business logic of the proposed file-sharing mechanism, is installed on each of the organizations. The blockchain ledger maintains transactions in the form of blocks. To realize the proposed system, the flow of identity manage- ment and file-sharing activities is explained in the following subsections. Table II illustrates the notations used in these transaction flows.

Table II Notations

Notation	Description
pki,ski	Public and private key pair for user i
K	Symmetric Encryption Key
M	File to be shared
M′	Encrypted File
TS	Timestamp
SIG(ski,TS)	Signature on timestamp with user's private key ski
BCID$_1$	Blockchain Identifier of user i
M$_{fid}$	File Identifier
M$_{cid}$	Content ID of the input file
M′$_{metadata}$	File metadata
userdetails	User identity details
R1,R2,...Rn⟩	File receiver list

There will be four major types of transactions in the system:

A. User Identity Registration

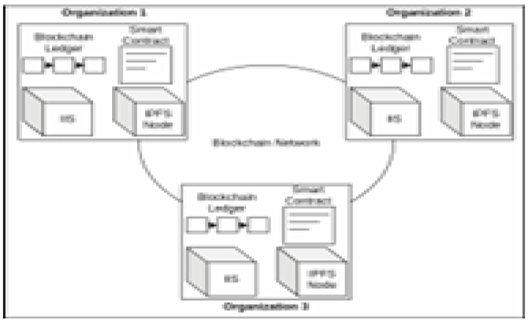


Fig. 1.The proposed system

B. User Identity Authentication

C. File Sharing

D. File Retrieval

A. User Identity Registration

For sharing files among the users of the participant or organizations, the concerned users need to be registered with the blockchain through the smart contract. Figure 2 illus trates the sequence of actions in the user identity registra tion process. A key pair (pki,ski) is generated by the end user application. The private key ski is kept with the user machine and the public key along with user details like name, email, organization, password, etc is sent to the IIS. IIS admin verifies the user identity registration request. Upon successful verification the hash of the password is written to the identity database. IIS sends the user registration request to the smart contract with public key pki and user details. The smart contract generates the blockchain identifier BCIDi and inserts BCIDi, userdetails,pki into blockchain ledger. The blockchain update status(success/failure) and BCIDi is sent to IIS. IIS sends the user identity registration status and BCIDi to the end user application.

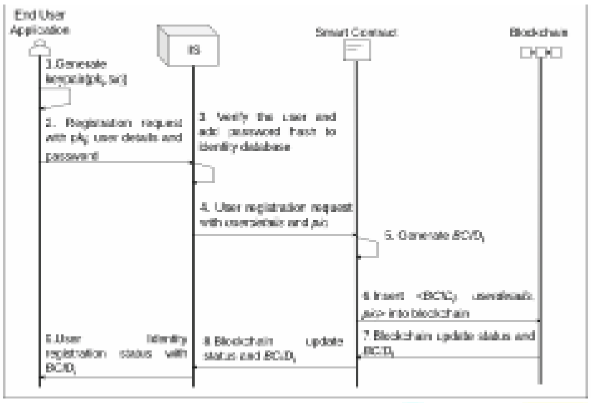

End User Application
IIS
Smart Contract
Blockchain
1.Generate keypair(pk, sk)
2. Registration request with pk, user details and password
3. Verify the user and add password hash to identity database
4. User registration request with userdetails and pk
5. Generate BCID
6 Insert <BCID, userdetails, pk> into blockchain
7 Blockchain update status and BCID
8 Blockchain update status and BCID
9.User identity registration status with BCID

Fig. 2. User Identity Registration

B. User Identity Authentication

The registration process records each user’s identity into the blockchain ledger. Thus blockchain acts as one more factor of authentication along with password. Figure 3 illustrates the user identity authentication process. The end user application sends the login request with BCIDi, password, TS and SIG(ski,TS) i.e., a digital signature on timestamp TS, using user’s private key ski. IIS verifies the user password and upon successful verification sends the blockchain verification request with BCIDi,SIG(ski,TS) , to the smart contract. The smart contract fetches pki i.e., the public key of the user from the blockchain using BCIDi. The smart contract verifies the SIG(ski,TS) using the retrieved public key and TS. The blockchain verification status(success/failure) is sent to IIS. Depending on the blockchain verification status, the user is notified about the status of the user identity authentication request.

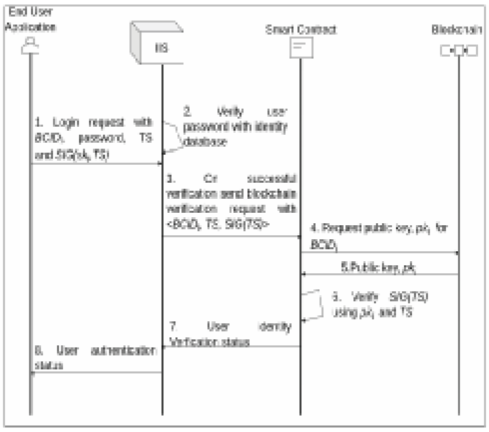

End User Application
IIS
Smart Contract
Blockchain
1. Login request with BCID, password, TS and SIG(sk, TS)
2. Verify user password with identity database
3. On successful verification send blockchain verification request with <BCID, TS, SIG(TS)>
4. Request public key, pk for BCID
5.Public key, pk
6. Verify SIG(TS) using pk and TS
7. User identity Verification status
8. User authentication status

Fig. 3. User Identity Authentication

C. File Sharing

Once the user is registered with blockchain, he can securely share files with any other registered user. Figure 4 illustrates the file-sharing process. The user logs in using the user authentication process. The user selects the file to be shared and specifies the file receivers. The end user application generates the symmetric key, K and encrypts file, M to be shared using K. The encrypted file M′ is uploaded to the IPFS distributed storage. IPFS returns content ID of the uploaded file, M'_{cid} to the end user application. The end user application generates file identifier, M_{fid} for the unique identification of the file. The end user application requests the public keys of R1,R2,...Rn i.e., the receivers with whom the file is to be shared. IIS requests the receiver public keys from the smart contract, which fetches the same from the blockchain and is returned to IIS. IIS shares the receiver public keys with the end user application. The M′ metadata, i.e. the file metadata structure is created as follows:

$$
\begin{aligned}
M'_{metadata} = \{ \\
&[BCIDR1 ,Enc(pkR1 ,M' cid||K)],\\
&[BCIDR2 ,Enc(pkR2 ,M' cid||K)],\\
&.....\\
&[BCIDRn ,Enc(pkRn ,M' cid||K)]\}
\end{aligned}
$$

Where BCIDRi is the blockchain ID of receiver Ri , pkRi is the public key of receiver Ri. Here, Enc(pk_{Ri} ,M′ $_{cid}$||K),is the content ID of the encrypted file M'_{cid} and symmetric key K is encrypted using public key pk_{Ri} . File sharing request is sent with< $M_{fid,}$ $M^{`}_{metadata}$, *fielddetails* > to IIS. Here file details are attributes of the file to be shared i.e., name, size, sender, etc. IIS sends the file metadata upload request to the smart contract. The smart contract saves the file metadata to the blockchain ledger. The file metadata upload status(success/failure) is notified to the IIS. IIS notifies the file-sharing status to the end user application.

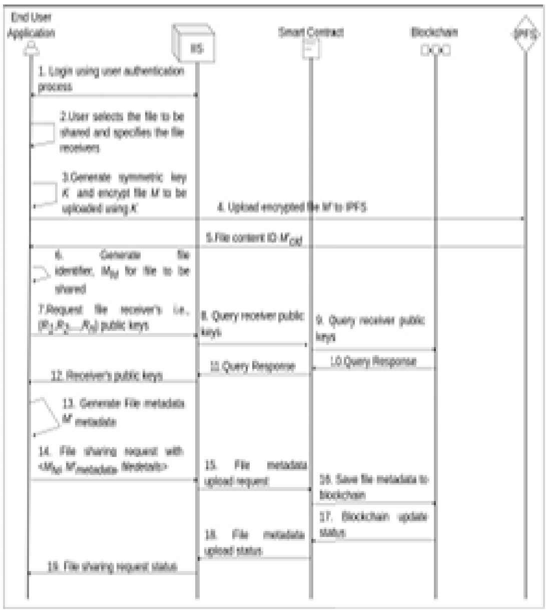

End User Application
IIS
Smart Contract
Blockchain
IPFS
1. Login using user authentication process
2.User selects the file to be shared and specifies the file receivers
3.Generate symmetric key K and encrypt file M to be uploaded using K
4. Upload encrypted file M' to IPFS
5.File content ID M'cid
6. Generate file identifier, Mfid for file to be shared
7.Request file receiver's i.e., (R1,R2,...Rn) public keys
8. Query receiver public keys
9. Query receiver public keys
10.Query Response
11.Query Response
12. Receiver's public keys
13. Generate File metadata M' metadata
14. File sharing request with <Mfid, M'metadata, filedetails>
15. File metadata upload request
16. Save file metadata to blockchain
17. Blockchain update status
18. File metadata upload status
19. File sharing request status

Fig. 4. File Sharing

D. File Retrieval

To securely retrieve the file, the file $_{Metadata}$ has to be fetched from the blockchain and actual file will be retrieved from the IPFS. Figure 5 illustrates the file retrieval process. The user logs in using the user authentication process with his BCID. The user requests the file browser from IIS. IIS requests the list of files the user is allowed to access from the smart contract. The *smart* contract queries the blockchain for the list of files in which the logged in user is a receiver or sender. The smart contract sends the file list to IIS. IIS presents the user with file list with file details like name, size and file identifier i.e., M_{fid} as file browser. The user selects the file to be retrieved from the file browser. IIS obtains the (M_{fid}) of the file from the user selected file and requests the smart contract for the f ile metadata. The smart contract verifies whether the user's BCID is in the sender/receiver list of the requested file's metadata. After verification the smart contract sends the file metadata M′ metadata to IIS. IIS sends the file metadata to the end user application. The end user application decrypts the f ile metadata using user's private key ski to get symmetric encryption key K and content id of the encrypted file M′ cid. The end user application requests the encrypted file M′ from IPFS storage using M′ cid. The encrypted file M′ is decrypted using the symmetric key K and the plain file M is shared with the end user

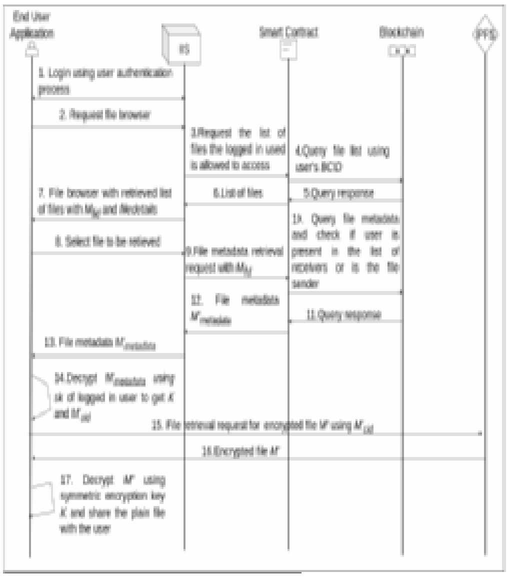
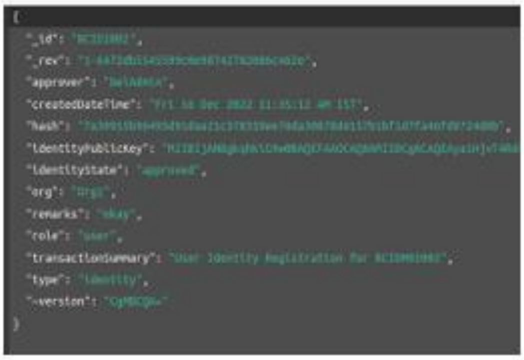

End User Application
IIS
Smart Contract
Blockchain
IPFS
1. Login using user authentication process
2. Request file browser
3.Request the list of files the logged in used is allowed to access
4.Query file list using user's BCID
5.Query response
6.List of files
7. File browser with retrieved list of files with M_{fid} and filedetails
8. Select file to be relieved
9.File metadata retrieval request with M_{fid}
10. Query file metadata and check if user is present in the list of receivers or is the file sender
11.Query response
12. File metadata $M'_{metadata}$
13. File metadata $M'_{metadata}$
14.Decrypt $M'_{metadata}$ using sk of logged in user to get K and M'_{cid}
15. File retrieval request for encrypted file M' using M'_{cid}
16.Encrypted file M'
17. Decrypt M' using symmetric encryption key K and share the plain file with the user

Fig. 5. File Retrieval

IV. EXPERIMENTAL SETUP AND RESULTS

The proposed system is implemented on Ubuntu 20.04 on virtual machine with 16GB RAM, an 8 core processor and *80GB* of Hard disk space. Hyperledger Fabric 2.4.7 has been used for creating the blockchain network. A three organizations blockchain network setup was created using dockers. Hyperledger Caliper [8] benchmarking tool has been used for testing the setup.

The user identity asset and file asset is maintained in couchDB[9]state database of the blockchain. Figure 6 il lustrates the Identity asset representation in the blockchain ledger's state database. Here the figure shows an identity BCID1002 in the blockchain with attributes approver, created Date Time, hash, identity PublicKey, identity State, *organization*, remarks, role, transaction Summary and type. Public key of the identity is kept in the blockchain ledger so that it can be retrieved by the any other users for sharing the files.



```
{
  "_id": "BCID1002",
  "_rev": "1-4472d[illegible]",
  "approver": "[illegible]",
  "createdDateTime": "Fri 16 Dec 2022 11:35:12 AM IST",
  "hash": "[illegible]",
  "identityPublicKey": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA[illegible]
  "identityState": "approved",
  "org": "Org1",
  "remarks": "okay",
  "role": "user",
  "transactionSummary": "User Identity Registration for BCID1002",
  "type": "identity",
  "~version": "[illegible]"
}
```

Fig. 6. Identity asset in blockchain state database

Figure 7 illustrates the file asset in the blockchain's state database. The file metadata will be stored in the encrypted format for each of the receiver of the concerned *file*. Here the figure shows $file_{MetaData}$ for two receivers i.e., BCID1000 and BCID1001.

Fig. 7. File asset in blockchain state database

Table III demonstrates Hyperledger caliper benchmark report. The Caliper tool has been configured to initiate 1,00,000 *transactions*, each for user identity registration, adding $file_{MetaData}$ to blockchain ledger, querying identity and querying file metadata from the blockchain ledger's state database. The report shows the Operations(transactions), Success, Failure, Send rate, Maximum latency, Minimum Latency, Average Latency and TPS(transactions per second)i.e. throughput. For user identity registration the throughput is 98.4 TPS. For adding file metadata to blockchain the throughput is 77.3 TPS. The read/ query transactions had throughput of 100 TPS.

Table III Hyperledger Caliper Test Results

Operation	Success	Failure	Send Rate	Max Lat	Min Lat	Avg Lat	TPS
User Identity	100K	0	100	45.18	0.07	20.15	98.4
Add File Metadata	100k	0	77.3	1.91	0.05	0.10	77.3
Query Identity	100k	0	100	0.02	0.00	0.00	100
Query file metadata	100k	0	100	0.02	0.00	0.00	100

V. CONCLUSION

The proposed system provides secure file-sharing across a consortium of organizations using blockchain. It provides confidentiality, integrity, *and* availability of shared files. It ensures end to end encryption of the files. The content ID of the shared file is stored on the blockchain in a tamper resistant way. The encrypted file and file metadata is stored in a distributed fashion on the distributed IPFS storage and blockchain ledger respectively. The system is realized using open source blockchain framework Hyperledger Fabric and tested using Hyperledger Caliper tool.

VI. FUTURE WORK

The present system has implemented file access control by encrypting and writing file metadata for each receiver in the blockchain ledger. As part future scope the ledger storage space could be optimized by creating user groups and performing encryption at the group level. [17] and [18] could be exploited to optimize the group-level encryption process.

REFERENCES

[1]. S. Nakamoto, “bitcoin: A peer-to-peer electronic cash system,” 2008. [2] ”Hyperledger Fabric Documentation, Release https://readthedocs.org/projects/hlf/downloads/pdf/latest/, on Dec 2022 main”, accessed

[2]. Naz M, Al-zahrani FA, Khalid R, Javaid N, Qamar AM, Afzal MK, Shafiq M. A secure data sharing platform using blockchain and inter planetary file system. Sustainability. 2019 Dec 10;11(24):7054.

[3]. Liu J, Li X, Ye L, Zhang H, Du X, Guizani M. BPDS: A blockchain. [4] Liu J, Li X, Ye L, Zhang H, Du X, Guizani M. BPDS: A blockchain based privacy-preserving data sharing for electronic medical records. In2018 IEEE Global Communications Conference (GLOBECOM) 2018 Dec 9 (pp. 1-6). IEEE.

[4]. Satapathy U, Mohanta BK, Panda SS, Sobhanayak S, Jena D. A secure framework for communication in internet of things application using hyperledger based blockchain. In2019 10th international conference on computing, communication and networking technologies (ICCCNT) 2019 Jul 6 (pp. 1-7). IEEE.

[5]. YSari L, Sipos M. FileTribe: blockchain-based secure file-sharing on IPFS. InEuropean Wireless 2019; 25th European Wireless Conference 2019 May 2 (pp. 1-6). VDE.

[6]. Benet J. Ipfs-content addressed, versioned, p2p file system. arXiv preprint arXiv:1407.3561. 2014 Jul 14.

[7]. ”Hyperledger Caliper Documentation, Getting ”https://hyperledger.github.io/caliper/v0.5.0/getting-started/”, on Dec 2022 Started”, accessed

[8]. ”Apache Couch Database Documentation: Release 3.3.0”, ”https://docs.couchdb.org/ /downloads/en/latest/pdf/”, accessed Dec 2022 on

[9]. Buterin, Vitalik. ”A next-generation smart contract and decentralized application platform.” white paper 3, no. 37 (2014): 2-1.

[10]. ”How IPFS Works”, ”https://datatracker.ietf.org/meeting/interim-2020 dinrg-01/materials/slides-interim-2020-dinrg-01-sessa-an-overview-of the-interplanetary-file-system-ipfs.pdf”, accessed on Dec 2022.

[11]. Nian, Lam Pak, and David Lee Kuo Chuen. ”Introduction to bitcoin.” In Handbook of digital currency, pp. 5-30. Academic Press, 2015.

[12]. Storj Labs, I. ”Storj: A Decentralized Cloud Storage Network Frame work.” (2018).

[13]. Benet, J. ”Filecoin: A Decentralized Storage Network. Protocol Labs, 2017.”

[15] Vorick, David, and Luke Champine. ”Sia: Simple decentralized storage.” Retrieved May 8 (2014): 2018.