2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)

Research and application of data sharing platform integrating Ethereum and IPFs Technology

Sun Jianjun

Admissions and Employment Office
Shandong Vocational College of Science and Technology
Weifang, China
e-mail:631901036@163.com

Li Ming

College of Intelligence and Information Engineering
Shandong University of Traditional Chinese Medicine
Jinan, China
e-mail:dianmail@sina.com

Ma Jingang*

College of Intelligence and Information Engineering
Shandong University of Traditional Chinese Medicine
Jinan, China
corresponding author's
e-mail:ma_jingang@126.com

***Abstract*—The development of the Internet has increased the amount of data exponentially, and the reliable transmission, sharing, and storage of data have become the primary problems of data sharing. This article combines the decentralized and irreversible characteristics of the Ethereum blockchain, integrates IPFS distributed storage technology, and proposes a data sharing platform under a new technology environment to ensure data security, user rights protection, and high-speed data processing. Provide a new technology application environment for current data sharing.**

Keywords-ethereum; IPFS; data sharing; blockchain

I. INTRODUCTION

With the rapid development of Internet technology, the amount of digital information data is also increasing at a huge (very fast) rate. The Internet platform has formed a huge user circle, and the amount of information it generates is naturally inestimable. Research on data sharing technology has become one of the key topics in the new era. Blockchain technology is the mainstream of today's emerging technologies, with features such as tamper resistance, transaction transparency, and trustless. The essence is a decentralized distributed ledger, which is jointly maintained by the nodes in the network and follows the same consensus mechanism, and then updates the contents of the ledger, and finally realizes trusted transactions between untrusted nodes in the distributed network. Blockchain technology is the general term for data encryption, consensus mechanism and smart contract technologies used in this process. IPFS (InterPlanetary File System) is a distributed file system with the advantages of permanent, decentralized storage and point-to-point file transfer[1]. The combination of IPFS and blockchain technology can ensure the security and stability of data, amplify the performance of platform data management and sharing, and provide high technical support and secure platform environment for data resource management, information security protection, user rights protection, and information exchange interaction[2].

II. BRIEF INTRODUCTION OF RELATED TECHNOLOGIES

A. Ethereum and smart contracts

Ethereum is a new technology at the bottom of the blockchain. The blockchain mechanism based on cryptography and P2P networks is an open source programmable blockchain platform with smart contract functions. It has also become the world's largest smart contract technology Blockchain. Ethereum can be viewed as a series of protocols. Any code that complies with the protocol can be executed on the Ethereum virtual machine EVM. These protocols have the characteristics of decentralization and provide solutions for the rapid development of decentralized applications. Ethereum has shared distributed processing features, ensuring the consistency of data in the network environment. The process and results of each data executed by Ethereum will be stored in the blockchain. The process of the transaction is open and transparent. The data storage is safe and real-time, and the information will not be tampered with, reducing the number of invalid data and wrong information. Its smart contract function also provides innovative space for software development, becoming the primary choice for current smart development[3].

B. IPFS interstellar file system

IPFS is also known as the interstellar file system and is one of the representatives of distributed file storage systems. Any node in the IPFS network is independent and does not depend on other nodes. The data access will select the nearest node, which speeds up the data transmission and reduces the storage space occupancy rate. When the information is stored in the IPFS node, the IPFS node will Identify the unique hash value of the file according to the incoming file information, the data will be permanently stored in the network environment. When extracting the file information, you can find the hash value according to user needs and obtain the file from the network node[4].

C. Introduction to RSA and AES encryption algorithms

RSA is an asymmetric encryption algorithm and is a public key cryptosystem that uses different encryption and decryption keys. The RSA algorithm is the first algorithm that can be used for both encryption and digital signatures. The operation is simple and easy to understand, but its

2473-3636/20/$31.00 ©2020 IEEE
DOI 10.1109/DCABES50732.2020.00079

operation speed is relatively slow. AES is a symmetric encryption algorithm that supports the encryption of subkeys, and AES has a parallel processing mode, and the operation speed is relatively fast. Both RSA and AES are currently commonly used encryption algorithms.

III. Ethereum + IPFS data sharing platform design

This article fully considers the unique advantages of the blockchain. Aiming at the problems and limitations of the current technology, taking the data sharing platform as an example, with the help of IPFS technology, we propose a data sharing platform of Ethereum + IPFS. The system innovates data sharing technology from the aspects of data encryption security, uniqueness of smart contracts and decentralization, and builds a system platform with high reliability, high trust, easy supervision and low cost[5].

A. Overview of model architecture and functions

The data sharing platform based on Ethereum and blockchain technology combines data encryption algorithms, smart contracts, blockchain structure and IPFS to achieve data upload, encryption, storage, sharing and other functions. Figure 1 shows the model architecture of the system. After the user uploads the data information, the system will encrypt the original data using the RSA algorithm, and the encrypted key will be saved to the data security management layer and also stored on the IPFS. The original ciphertext is double-encrypted by AES data encryption algorithm to generate ciphertext and transmitted to IPFS for distributed storage. Due to the decentralized feature of IPFS, the data on storage and IPFS will be easily found. The double encrypted ciphertext enters the smart contract set module from IPFS and passes the arbitration contract , Escrow contracts, users and data rights protection related contracts, smart contract modules can be written according to needs. When other users or third-party applications need to share and call existing data in the system, they must first send a request to the smart contract set module. The smart contract set module analyzes the sender and the request event and sends a transaction request confirmation to Ethereum. The Ethereum miner makes the request judgment and stores the request. And if the contract requirements are met, the sender is granted the corresponding authority. The user or third-party application with authority will submit an application to IPFS, call the data in IPFS to achieve data sharing.

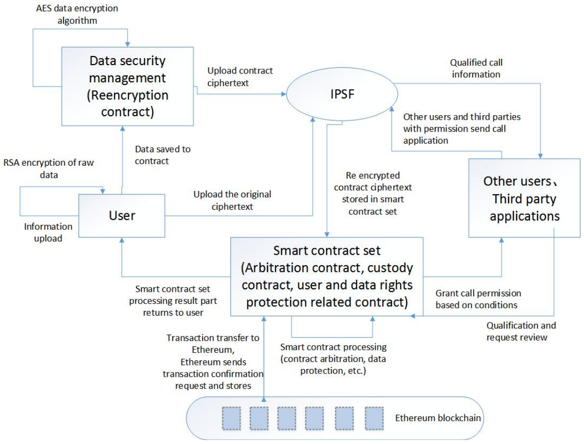

AES data encryption algorithm
Data security management (Reencryption contract)
Upload contract ciphertext
IPSF
Qualified call information
Other users and third parties with permission send call application
Data saved to contract
RSA encryption of raw data
User
Information upload
Upload the original ciphertext
Re encrypted contract ciphertext stored in smart contract set
Other users、Third party applications
Smart contract set (Arbitration contract, custody contract, user and data rights protection related contract)
Smart contract set processing result part returns to user
Grant call permission based on conditions
Qualification and request review
Transaction transfer to Ethereum, Ethereum sends transaction confirmation request and stores
Smart contract processing (contract arbitration, data protection, etc.)
Ethereum blockchain

Figure 1. Figure 1 Ethereum + IPFS data sharing platform architecture diagram

The Ethereum + IPFS data sharing platform has the functions of intelligently identifying the user's identity and screening transaction information. Combining the information identification system provided by IPFS and Ethereum can realize the independent verification of user information and the identification and authorization of data information without the third-party certification body. The smart contract set module provides contract agreements, verifies user information in the form of logical operations and smart protocols, and combines with the actual access control model and digital signature technology to form a user identity authentication and data information verification service system. The specific data flow is:

- The user issues a transaction application (file access request, information call request, etc.);
- The transaction request is sent to the smart contract set. First, request to match the corresponding smart contract module;
- The smart contract module verifies the user information of the sender, verifies the identity, and determines its legality;
- The information verified by the smart contract identity will enter the Ethereum blockchain module, and the smart contract will retrieve the transaction-related requirements from the Ethereum blockchain and match them;
- After screening and checking the conditions of the Ethereum blockchain, the transaction confirmation information will be returned to the corresponding smart contract set, and then the file information will be returned to the file access node to realize data invocation and sharing.

The core of the entire data sharing platform is the information verification, registration, and identification functions formed by Ethereum and IPFS. The confirmed transaction will be returned to the storage and IPFS interstellar file storage system to ensure the storage and backup of information.

B. *Data re-encryption*

System data encryption adopts RSA + AES re-encryption to ensure data security. The theoretical execution process of re-encryption is: the data receiver first creates an RSA key pair, and sends the RSA public key to the sender while saving the RSA private key; The sender creates an AES key and encrypts the plaintext of the transmitted data. The sender encrypts the AES key with the existing RSA public key to form an AES ciphertext and sends it to the receiver. After receiving the encrypted AES key and ciphertext, the receiver decrypts the data with the help of the RSA private key, and decrypts the ciphertext with the decrypted AES key to generate plain text. The process of data re-encryption in this system platform is:

- When the user applies for data upload and storage: the user uploads the data through the platform webpage and sends a request. The data generates an AES key A1, and IPFS creates an RSA key pair RO. The uploaded plaintext is encrypted by A1 to generate ciphertext AS1, and the public key R1 created by RSA is sent to the client program. The client encrypts AS1 with the received R1 and sends it to IPFS. IPFS uses the RSA private key to decrypt the encrypted AS1. Finally, IPFS decrypts the ciphertext with the decrypted A1 to obtain the final plaintext and stores it.
- When a third-party user invokes a data request: sending data calls, sharing requests. The request is sent to IPFS after passing the verification of the smart contract set and the Ethereum blockchain. IPFS randomly generates the AES key A1, and the requester creates an RSA key pair RO.IPFS encrypts the plaintext with A1 to generate ciphertext AS1, and sends the public key R1 created by RSA to IPFS. RFS received by IPFS encrypts AS1 and sends it to the requester. The requester uses the RSA private key to decrypt the encrypted AS1. Finally, the requester decrypts the ciphertext with the decrypted A1 to obtain the final plaintext. IPFS record keeping throughout the process.

The entire data sharing process is re-encrypted to ensure the security of the data, making full use of the distributed and intelligent features of smart contracts and IPFS to ensure strict control and security of the process.

C. *Smart Contract Set*

Smart contract set module includes data management contract DS (management data master contract), arbitration contract ZC (contract arbitration function), escrow contract TC (data custody to IPFS related agreement execution), user and data rights protection BH (may include credit points Management, user black and white list execution, etc.) and custom smart contract ZD, can add smart contracts according to the actual environment to ensure the sound function. The transaction information will be reviewed by the Turing computing language and related certification system of the Ethereum blockchain, and a distributed processing environment will be provided to ensure the fairness of the transaction and the efficient processing speed.

The entire smart contract set uses the data management contract DS as the main contract, which is responsible for integrating other sub-contracts in the contract set and creating quick call interface methods for each contract. The following sub-contract functions are implemented under the main contract:

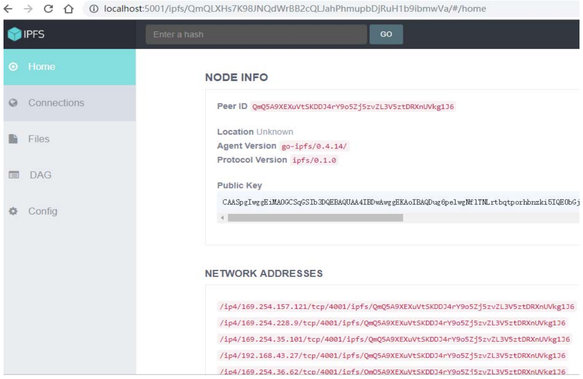
- Arbitration contract (ZC) implements contract arbitration according to the access control mechanism, including the rationality and maturity of contract construction;
- The escrow contract (TC) stores and manages the data information according to the data type, source and corresponding copyright protection regulations, etc., and provides users with agency-based data escrow to ensure the real-time legal provisions and agreements;
- User and data rights protection (BH) compares and analyzes based on the existing data information of the platform, and maintains file information on the premise of guaranteeing user copyright;
- Search contract (CD) record data types and corresponding field keywords, use hash values to save and query data, and improve search efficiency;
- User Credit Registration (UR) is responsible for registering and remarking the user information registered on the platform, screening and verifying user behavior, showing the user's honesty in the form of credit points, and setting up a punishment system for low trustworthy users to ensure The platform environment is pure.

The smart contract set module relies on the protocol information stored in the Ethereum blockchain, and is called after comparison and verification to ensure the accuracy and comprehensiveness of the data information and realize the intelligence of the platform.

IV. IMPLEMENTATION BASED ON ETHEREUM AND IPFS DATA SHARING PLATFORM

A. *System environment construction and code implementation*

The operating environment of this platform is Apple MacBook 15 (2019), 16G memory, 6 core Intel core i5 processor, and the operating system is macOS High Sierra. This platform selects Solidity programming language to develop smart contracts and Visual Studio Code software as the development environment, selects Mongodb database to store data, and uses HTML, JS and other auxiliary development front ends. After starting the database and IPFS, the basic information data of the system is uploaded. In this paper, 500,000 data information including scientific research data, experimental data, website data, etc. are extracted from the major data platforms and stored in the platform background. As basic data, it provides basic platform data information for data matching and verification. Figure 2 shows the interface after IPFS starts.


localhost:5001/ipfs/QmQLXHs7K98JNQdWrBB2cQLJahPhmupbDjRuH1b9ibmwVa/#/home
IPFS
Enter a hash
GO
Home
Connections
Files
DAG
Config

NODE INFO

Peer ID QmQ5A9XEXuVtSKDDJ4rY9o5Zj5zvZL3V5ztDRXnUVkg1J6

Location Unknown
Agent Version go-ipfs/0.4.14/
Protocol Version ipfs/0.1.0

Public Key

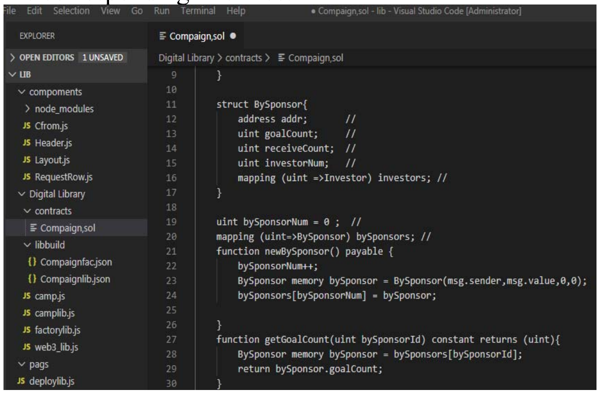
CAASpgIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQDug6pelwgNf1TNLrtbqtporhbnzki5IQE0bGj

NETWORK ADDRESSES

```
/ip4/169.254.157.121/tcp/4001/ipfs/QmQ5A9XEXuVtSKDDJ4rY9o5Zj5zvZL3V5ztDRXnUVkg1J6
/ip4/169.254.228.9/tcp/4001/ipfs/QmQ5A9XEXuVtSKDDJ4rY9o5Zj5zvZL3V5ztDRXnUVkg1J6
/ip4/169.254.35.101/tcp/4001/ipfs/QmQ5A9XEXuVtSKDDJ4rY9o5Zj5zvZL3V5ztDRXnUVkg1J6
/ip4/192.168.43.27/tcp/4001/ipfs/QmQ5A9XEXuVtSKDDJ4rY9o5Zj5zvZL3V5ztDRXnUVkg1J6
/ip4/169.254.36.62/tcp/4001/ipfs/QmQ5A9XEXuVtSKDDJ4rY9o5Zj5zvZL3V5ztDRXnUVkg1J6
```

Figure 2. Figure 2. IPFS startup interface screenshot

Figure 3 shows part of the code implemented by the smart contract of the data sharing platform. The Lib module constructs the smart contract of the platform, and Lib-Digital Library.sol is the main module code of the smart contract. Smart contract protocol can be customized. The compoments and pags folders contain the interface architecture of the platform. Camp.js And factorylib.js can read the Application binary interface generated by the Digital Library.sol smart contract. The implementation of the entire smart contract includes multiple sub-contracts, and when it is called, data sharing and intercommunication are realized through the corresponding interface.

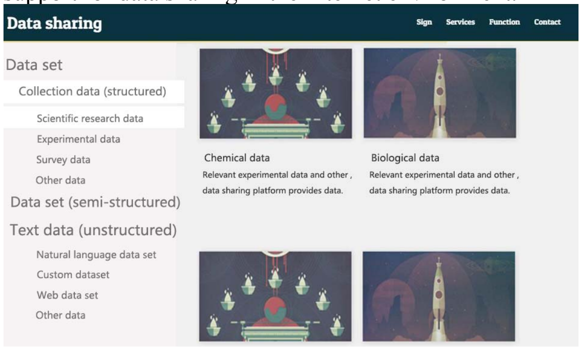


```
    }

    struct BySponsor{
        address addr;       //
        uint goalCount;     //
        uint receiveCount;  //
        uint investorNum;   //
        mapping (uint =>Investor) investors; //
    }

    uint bySponsorNum = 0 ;  //
    mapping (uint=>BySponsor) bySponsors; //
    function newBySponsor() payable {
        bySponsorNum++;
        BySponsor memory bySponsor = BySponsor(msg.sender,msg.value,0,0);
        bySponsors[bySponsorNum] = bySponsor;

    }
    function getGoalCount(uint bySponsorId) constant returns (uint){
        BySponsor memory bySponsor = bySponsors[bySponsorId];
        return bySponsor.goalCount;
    }
```

Figure 3. Part of the screenshot of the data sharing platform smart contract implementation

B. *System operation and demonstration*

Figure 4 shows a partial screenshot of the operation of the data sharing platform. The platform can upload and process structured, semi-structured, and unstructured data. The B / S architecture is used to implement the data sharing function. Users can login to upload and download data. The entire system structure is clear, and data sharing is realized on the basis of the web page frame. After the user logs in, the data can be checked, downloaded, uploaded, copyright information protection and other functions.. The platform independently implements data storage and maintenance, is convenient to use and comprehensive in information, and provides technical support for data sharing in the Internet environment.



Relevant experimental data and other , data sharing platform provides data.

Relevant experimental data and other , data sharing platform provides data.

Figure 4. screenshot of data sharing platform operation

V. SUMMARY

This article combines the Ethereum blockchain and IPFS technology to implement a data sharing platform. On the basis of distributed and decentralized data processing, the functions of smart contracts and data re-encryption are integrated to provide a safe, smart and convenient shared space for data, improve the reliability of data sharing, and ensure the safety of users and data information. Provide strong technical support for data processing and sharing. In view of the problem of limited data processing speed of this system, it is necessary to optimize for intelligent algorithms to improve data processing capabilities. The next step will be to improve the platform's operating efficiency under stable data sharing.

REFERENCES

[1] Ma Zongbao. Blockchain solution for big data applications [J]. Computer Products and Circulation, 2020 (04): 133.

[2] Chen Jie, Zhang Zaiyue, Zhang Xiaoru. Research on Reptile Smart Contract Integrating IPFS and Ethereum [J / OL]. Software Guide: 1-4 [2020-05-03]. Http://kns.cnki.net /kcms/detail/42.1671.TP.20191122.1626.082.html.

[3] Zhao Li. Design and Implementation of Distributed Collaborative Text Editing Website Based on IPFS and Ethereum [D]. Beijing University of Posts and Telecommunications, 2019.

[4] Su Xiongye. Research and Implementation of Big Data Sharing Model and Key Mechanism Based on Blockchain [D]. Beijing University of Technology, 2018.

[5] Yin Long, Wang Hongwei. Research on IPFS-based distributed data sharing system [J]. Internet of Things Technology, 2016, 6 (06): 60-62.