Multimedia Tools and Applications (2021) 80:31887–31906
https://doi.org/10.1007/s11042-021-11183-6

Network data security sharing system based on blockchain

Xinghua Lu[1] · Peihao Liu[1] · Yiran Ke[1] · Hao Zhang[1]

Received: 17 September 2020 / Revised: 11 January 2021 / Accepted: 24 June 2021 /
Published online: 21 July 2021

Abstract
Traditional network data security sharing system ignores that many people share data simultaneously, which leads to poor real-time performance. Therefore, the authors designed a network data security sharing system based on blockchain technology. In the hardware design, the PCI encryption card is used to encode the data. The microprocessor is used to access the system's external equipment, and DDR-SDRAM dynamic storage area and NAND flash static memory are used as network data. In the software section of the system, a secure transmission mechanism is established. The cp-abe method is used to encrypt the network shared data, and the multi-person digital envelope technology is used to share the data. These two methods contribute to the design of the network data security sharing system. In the experiment, eight users share the data. The system login response time, key distribution time, data encryption time, and key update time are taken as the experimental objects. Experimental results show that the system response time, key distribution time, data encryption time, and key update time of the system are shorter than those of the comparison method.

Keywords Blockchain · Network data · Secure sharing · Smart contract · Storage · Transmission management thread

1 Introduction

At present, many data resources are accumulated in various types of information systems, and these data are generally challenging to merge and link, leading the industry into a "data island" dilemma [16]. Data sharing effectively promotes the cross-reference of various data, enhances the potential value of the data, and brings enormous social and economic benefits. However, more and more data relates to the personal privacy or related privacy of the users, such as

✉ Xinghua Lu
xhlu@gdtu.edu.cn

[1] Huali College Guangdong University of Technology, Guangzhou 511325, China

clinical diagnosis results, date of birth, and medical card number in medical files. If this data is improperly shared, privacy leakage will inevitably occur. For this part of the data, a network data security sharing system based on cloud storage is generally adopted, open to specific people or organizations within a limited range and period [8]. In this system, fine-grained data sharing carried out in the manner of 'specific users-specific scenarios-specific resources', and the data sharing process is also tracked. However, this method has limited storage capacity, and it is challenging to deploy a large number of access strategies in real time through transactions. When multiple users share data at the same time, the real-time performance of the system is disturbed.

Shen et al. [15] proposed a data-sharing system based on cloud computing, and this system relies on the transplanted group signature algorithm to perform identity authorization and authentication. The multi-owner searchable encryption mechanism is used to enable group members to share data effectively to achieve group data sharing. Dileep Kumar Pattipati et al. [14] proposed data encryption and sharing scheme based on a level-ordered access structure. The existing access structure cannot achieve data encryption and sharing because they do not perform the required ordering. In this access system, there is a built-in order between each layer. The application that passes the system test and final acceptance test is released into the production environment. The actual use shows that the above systems can resist many possible attacks, but their real-time performance is inadequate. Based on this, a network data security sharing system based on blockchain is designed.

Blockchain is one of the most revolutionary emerging technologies in recent years. From the perspective of text and data structure, blockchain includes two types of data structures: block and chain. A block is a digital block used to store all transactions within a period of time. Chain refers to the interconnection or connection of stamping blocks to form an uninterrupted and unique chain structure. Blockchain is composed of several key elements, including five elements: modern encryption technology, distributed peer-to-peer network communication technology, decentralized consensus algorithm, incentive mechanism, and programmable script code. These elements combine with each other to form a technical combination of data processing, data verification, data exchange, and data storage. As a new type of distributed system, blockchain realizes a decentralized data mutual trust mechanism, using the combination of blockchain access control protocol and encryption technology to ensure that user data is not lost or tampered, and privacy is not leaked. Therefore, it is of great significance to apply blockchain to network data security sharing system design.

To this end, a network data security sharing system based on blockchain technology is designed. First, the overall framework of the network data security sharing system based on blockchain technology is given. The system hardware is designed and the PCI encryption card is used as the security. As for security equipment, S3C6410 microprocessor and DDR SDRAM memory are used to build hardware memory space. A database based on blockchain technology can decentralize and perform the trustless processing of network data. Besides, using CP-ABE method to gather network data attributes and access resources correlation, it can access the ciphertext information according to the authorized attributes of the receiver, and then realize network data encryption. Moreover, when the node requests shared perception data from the node, the identity of the network node is checked through the key. The data is decrypted according to the private key provided by the node. The constraint condition outputs the corresponding result. Before outputting the data to the node, the public key is used to encrypt the data to realize the network data security sharing system's design based on blockchain technology.

2 Design of network data security sharing system framework based on blockchain

Figure 1 indicates the framework of network data security sharing system based on blockchain:

The unified identity authentication module includes government personnel, approving personnel, and system administrators who have undergone strict identity authentication and connected to the system to complete a series of operations, use the services provided by the system, and maintain the system. Data sharing, data visualization and data sharing business processing help todesign system software. Data encryption and decryption can be complete by data collection, task scheduling, data sharing and data transmission. Then the decrypted data are transmitted to the system hardware to realize information interaction. Other public service systems are physically separated from this system. Data transmission is carried out through a special data exchange interface if information exchange is to be carried out.

3 Hardware design of network data security sharing system based on blockchain

The hardware structure diagram of the network data security sharing system is shown in Fig. 2.

As shown in Fig. 2, the network data is transmitted to the PCI encryption card through the microprocessor. The encryption algorithm is integrated into the hardware to improve the security of the system. Then the encrypted data is transmitted to the memory: DDR SDRAM dynamic memory and NAND FLASH. Model static memory for data storage.

3.1 Encryption card design

A PCI encryption card is chosen. It is a security device based on encryption and decryption chip and PCI bus card technology. It mainly provides security functions such as data encryption, data integrity, digital signature, and access control for the system. It can be used

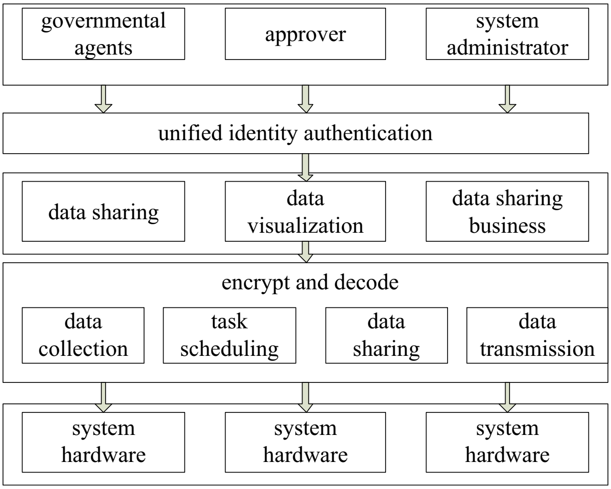

Fig. 1 Network data security sharing system framework based on blockchain

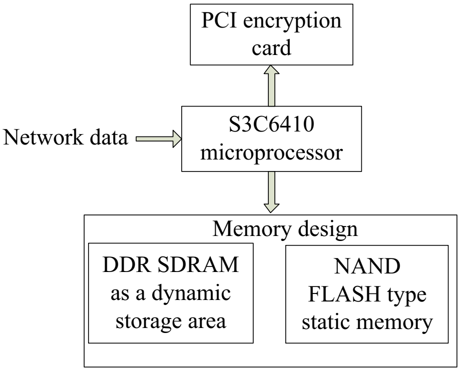

Fig. 2 Hardware structure of network data security sharing system based on blockchain technology

for computer file protection, E-mail system security, office automation security, database protection, and network encryption. The data encryption of the PCI encryption card includes two ways: software implementation and hardware implementation. Software implementation refers to performing encryption algorithms in the PCI configuration embedded microprocessor or DSP chip. Integrating encryption algorithms into hardware can improve the security of the system. The design details of the encryption algorithm in the software are shown below.

The entire PCI encryption card hardware is centered on an FPGA, and is equipped with FPGA configuration chips, clocks, and other modules. PCI interface, 3DES, and MD5 algorithm are all implemented inside FPGA. The encryption scheme of the encryption card designed this time is implemented with a PCI IP core in a dedicated chip for cryptography. This encryption card uses the second scheme: the soft core described by VHDL is downloaded to the FPGA. The PCI IP core conforms to the standard PCI2.2 protocol and supports I / O operations, configuration reads, write operations, bus BUS_MASTER reads, BUS_MASTER writes, DMA interrupt mode, and DMA data transfer mode. The IP core design is mainly implemented with finite state machines, including 7 state machines consisting of interrupt response, burst read operation, burst write operation, configuration read operation, configuration write operation, I / O read-write transmission, and memory read-write transmission. The workflow is shown in Fig. 2. After receiving the frame start signal (#FRAM signal is low), according to the command on the PCI bus C/BE [3..0], it enters one of the seven state machines. After the operation is completed, the host sends an end-of-frame signal (#FRAM is low), entering the waiting state, and enters the next operation.

The PCI interface IP core is used to complete the encryption card design, and the implementation of data transmission must be determined first. The PCI interface IP core supports two 256 K bytes of I/O space and a 4 M byte PCI memory space. In PCI data transmission, data transmission mainly depends on operations such as I/O read, I/O write, memory read, and memory write. I/O reading and writing commands are used to read and write data from a device mapped to the I/O address space. Memory reading and writing commands are used to read and write data from a device mapped to the memory address space. In the encryption card's initial design, the data is mapped to the I/O address space for operation. The basic working process of the system is as follows. The data enters the PCI interface module from the PC and is first stored in the I/O storage unit specified in the PCI interface module. Under the control of the control module, it enters the cryptographic algorithm module, and the data is processed and output into the output buffer, and the driver will transfer the calculation results to the application.

3.2 Microprocessor design

S3C6410 microprocessor with ARM1176JZF-S core is adopted. The microprocessor adopts advanced high-speed bus structure, and has 8-level pipeline. When the peripheral voltage of the core board reaches 1.2 V, the system clock frequency can reach 667 MHz. The size of the chip is 13 mm * 13 mm. The chip uses FBGA packaging technology, and its pins number is 424.

The physical address space supported by the microprocessor is 32 bits. The purpose of the physical space is roughly divided into two parts. Part of the space is used as a storage space, and the remaining space is used to connect external devices, which creates convenience for the processor to use addresses to access external devices. The address space of the main memory accessed is $0 \times 00000000 \sim 0 \times 6FFFFFFF$. Then the storage space of this size is divided into four operation spaces: static storage space, dynamic storage space, internal storage space, and image boot space.

The S3C6410 microprocessor [6, 19, 20] has the above three clocks and three types of phase-locked loops needed by special external devices. The three types of phase-locked loops are: one is to generate the ARNPLL, referred to as APLL, and the second is to generate HCLK and the main phase-locked loop required, short for MPLL, and the third is the external phase-locked loop required for the clock provided to the special external device, referred to as EPLL. The core board design selects four different types of crystal oscillators. The first is a 12 MHz crystal oscillator that generates a system clock for the processor. The second is a crystal oscillator that provides a clock signal of 32.768 kHz for the real-time clock. The third is a passive crystal oscillator that provides the display module with a required clock of 27 MHz. The fourth is a 48 MHz passive crystal oscillator required for USB operation.

The main function of the power control module of the processor is to provide the best power management solution and reduce the power consumption of the entire system. The power management module has four working modes: general clock control gate mode, IDLE mode, termination mode, and sleep mode. Table 1 indicates the description of each power supply designed by the core board.

3.3 Memory design

The DDR SDRAM with a capacity of is adopted as the dynamic storage area, and the NAND FLASH with a capacity of 128 M is adopted as the static storage area. The dynamic memory is convenient for the operation of the system software program. The static memory is used to store the BootLoader boot program, the roo$2 \times 64M$t file system image file and other data.

Table 1 Circuit designed by core board

Power supply voltage name	Voltage magnitude	Degree of control
VDD-INT	1.3 V	Controllable
VDD-ARM	1.2 V	Controllable
VDD-ADC	3.3 V	Out of control
VDD-OTGI	1.2 V	Controllable
VDD-RTC	3.3 V	Controllable
VDD-MDDR	2.5 V	Out of control
VDD-IO	3.3 V	Out of control
VDD-ALIVE	1.2 V	Out of control
VDD-OTG	3.3 V	Controllable

The data transmission of DDR SDRAM occurs at the rising and falling edges of the system clock. Its data transmission speed is twice that of the system clock. It has the characteristics of low heat dissipation, low power consumption, and high frequency. The K4H511638D chip from Samsung is used. The capacity of the chip and the normal operating voltage of the chip are $32M \times 16$ and 2.5 V, respectively. The chip is composed of 4 banks and uses a 66-pin TSOP package type.

NAND FLASH type static memory is adopted, and its data is read in a block as a unit. The block is easy to read and write data operations, and its size is usually 512 bytes. The core board uses K9F1G08 chip of Samsung, and the chip has a large capacity and reliable performance [5, 7, 18]. The normal operating voltage of the chip is 2.7 V ~ 3.6 V. The chip uses 8-bit width input and output pins, which can be used to enter commands, data and addresses, and output data during the reading operation.

4 Software design of network data security sharing system based on blockchain

4.1 Establishment of a secure transmission mechanism

To ensure the security of data transmission, a secure transmission mechanism is established, which is implemented through the transmission management thread. The transmission management thread is a thread synchronized with the sampling thread. This thread mainly transmits or stores the latest sampling data according to data sharing requirements. The sampling period may be short (the shortest period is 3 s). Therefore, the transmission management thread synchronized with the sampling thread must complete the latest sampling data processing within 3 s. Considering the quality of network transmission and the load of communication server, the transmission management thread may take more than 3 s to transfer data. As a result, the sampling thread cannot strictly follow the sampling cycle.

Based on the above considerations, in this system, the online transmission thread is responsible for the network data transmission that may cause a long-term response, and the transmission management thread only transfers the latest sampling data to the online transmission thread. Suppose the transmission management thread finds that the latest sampling data cannot be transmitted to the network in time. In that case, the transmission management thread stores the latest sampling data locally to ensure that the sampling data is not lost. The main implementation process of the transmission management thread is as follows:

First, the mutex of the sampling buffer is acquired. The sampling thread and the transmission management thread mutually access the shared resource 'sampling buffer'. The transmission management thread must acquire an exclusive lock when reading data from the sampling buffer to prevent the transmission management thread from reading inconsistent data.

Second, the transmission management thread is blocked and need to wait to acquire the mutex. When the sampling thread is writing data to the sampling buffer, because the sampling thread has acquired the mutex for accessing the sampling buffer, the transmission management thread fails to acquire the mutex. The transmission management thread is blocked until the sampling thread releases the mutex.

Third, the system status needs to be determined. The system status is a global variable used to reflect whether the sampling workstation can normally communicate with the communication server. When the system state is normal, the transmission management thread can wake up the online transmission thread to transmit data. When the system state is abnormal, the transmission management thread stores the data locally. The network state variables are jointly maintained by the online transmission thread and the network diagnostic thread.

Fourth, the online transmission status of the system data need to be determined. When the online transmission thread is idle, it can immediately transmit the latest sampling data. When it is busy, indicating that it is transmitting the last sampling data, and the current sampling data cannot be updated to the communication server, and the transmission management thread should store the current sampling data locally.

The rapid development of digital communication technology and the tremendous improvement of computer capabilities have made the use of the Internet in various commercial, government and social interactions involving the transmission of various complex data and multimedia objects to increase exponentially. While ensuring the privacy of information, it has become vital to ensure sensitive transactions and personal transaction content on open networks, but the challenges are also growing. Therefore, the research field of information and multimedia security has aroused more and more interest, and its application range has been greatly expanded. The two most obvious solutions to protect information privacy are provided through encryption and steganography. Encrypting secret messages converts them into observable but meaningless noise-like data, while concealment conceals the existence of secret information by hiding it in ordinary communications, which will not cause unwelcome snooping. Digital steganography involves using images, video and audio signals as masking objects to hide secret bitstreams. The suitability of media files for such purposes is due to the high degree of redundancy and the most widely exchanged digital data. The field of steganography has become the focus of information security. Because every Web site relies on multimedia, such as audio, video and images. Steganography is a technology that can embed secret information into digital media without damaging the quality of its carrier. The third party is neither aware of the existence of secret information, nor aware of the existence of secret information. Therefore, keys, digital signatures, and private information can all be safely transmitted in an open environment [1].

To ensure the safe storage of sampling data [11, 12, 17], the blockchain is used to construct the database. Blockchain system can significantly reduce costs, reduce risks and management costs, improve liquidity, and increase the opportunities for innovative products and services. Each data block contains the information of the entire network transaction for a period, which is used to verify the validity of the information (anti-counterfeiting) and generate the next block. Therefore, the blockchain is a technical solution that can collectively maintain a reliable database in a decentralized and trustless manner. Figure 3 indicates the structure of each block in the blockchain.

The blockchain system is composed of a top-down data layer, a network layer, a consensus layer, an incentive layer, a contract layer, and an application layer. The data layer encapsulates the chain structure of the underlying data block and the related asymmetric public and private key data encryption technology and time stamping technology. It is the lowest data mechanism in the whole blockchain.

Table 2 indicates the data output format of the database built by blockchain:

According to Table 2, the file resource security operation is mainly composed of three parts: the main body of access, the file object accessed, and the type of access. During the network data sharing process, a large amount of data is shared. To not affect the next cycle of sampling, the transmission management thread must release the sampling buffer area as soon as possible, so the data in the ‘sampling buffer’ is copied to the database. Table 3 indicates the data sampling simulation and switch description:

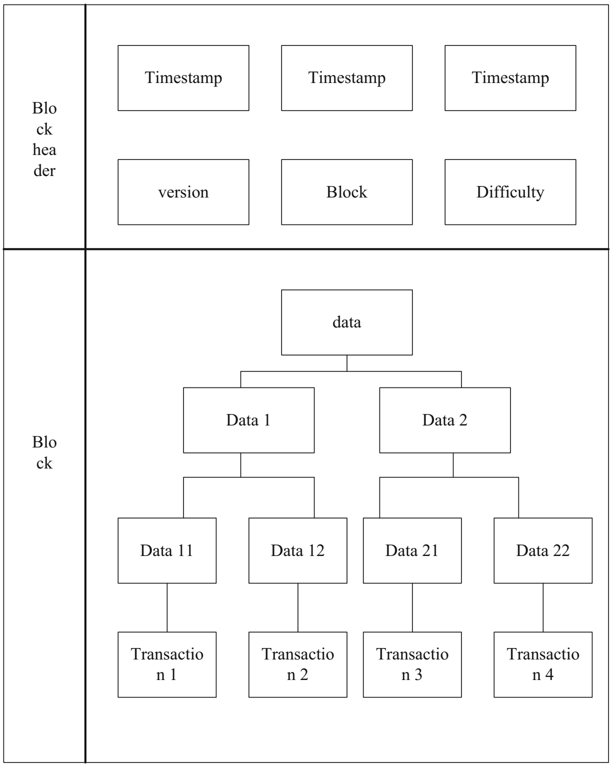

Fig. 3 Blockchain structure

Business logics such as access control, transaction verification, data registration, and key distribution are implemented on the blockchain. Operations such as data storage, key generation, and data encryption are placed under the blockchain. Data exchange between on-chain and off-chain is completed through transactions and smart contracts.

After the data arrives at the data interface module, the data is transferred to a secure computing container on a secure one-way transmission channel. All operations are recorded on the blockchain to ensure traceability, tampering, and forgery of the records. And the data is transmitted through the transmission mechanism to ensure the safety of data. After each data collaborator transfers the data to the secure computing container, the received data will be operated. At the same time, the operation process and results are also recorded on the blockchain. Data is transferred to public services. The data results of the cooperative operation

Table 2 Database content storage format based on blockchain

Serial number	Field name	Field description
1	UserID	username
2	Harddisk	Network hard drive name
3	FileID	Controlled file name
4	Key	File encryption and decryption
5	Permission	File operation authority

Table 3 Data sampling analog and switch description table

Serial number	Data sampling analog	Data sampling switch
1	Primary key	Switch description primary key
2	Analog description primary key	sample value
3	sample value	Sampling status
4	Sampling status	Sampling time of sampling station
5	Sampling time of sampling station	Real-time sampling data arrival time
6	Real-time sampling data arrival time	Source site
7	Attributes	overtime time
8	status	status
9	Trend time	
01	System primary key	

are transmitted to the relevant public service area through the physically isolated data exchange interface, and the process and results of this operation are also recorded on the blockchain.

Table 4 shows the difference between blockchain technology and traditional network data security sharing.

As shown in Table 4, compared with traditional network data security sharing technology, the system management mechanism of blockchain technology is decentralized management. It does not require a central administrator, and can perform read and write control simultaneously and also track previous transaction information. Therefore, the confidentiality performance is better.

4.2 Network data encryption

Data is transmitted based on the above transmission mechanism. Because the network data may be tampered when transmitted through the transmission mechanism, in order to improve the security of network data sharing, the CP-ABE method is used to encrypt the network data. This method is an encryption method based on data attributes. The sender specifies a policy for accessing the cipher, and associates the network data attribute set with accessing the resource. The receiver can access the cipher information according to its authorization attribute. The preferred initialization algorithm is executed on a trusted key distribution center, and its calculation formula is:

$$S = A(m, n)^{*}p + k \tag{1}$$

In formula (1), S is the cipher information of the network data, m represents the security factor, n represents the attribute space, p represents the system public key, and k is the system master key.

Table 4 The difference between blockchain technology and traditional network data security sharing

Blockchain technology	Traditional network data security sharing technology
Distributed control does not require a central administrator	Need for central management
Not only can save relevant information in real time, but also track previous transaction information	Only record current information
Confidentiality is good, can read and write control at the same time	It can also perform read-write control at the same time, with relatively poor confidentiality

Based on the initialization of the above algorithm, a key is generated in a trusted key distribution center. A user key associated with the attribute set is generated based on the system public key, the system master key, and the attribute set submitted by the data requester:

$$USK = (p+k)^{*}R \tag{2}$$

In formula (2), *USK*represents the user key and *R* represents the attribute set submitted by the data requester.

According to the key generated by the above process [2, 9, 13], the Linux buffer mechanism is used to encrypt the data. The mechanism mainly includes two aspects. The first is the position of the encryption and decryption operations in the file system, and the second is the segment encryption. When encrypting data, it must be performed at a suitable location in the system, which is not only easy to implement but also has a small impact on the performance of the system. For writing, encryption is only performed when it is needed to write to disk, and for reading, decryption is only performed when it is close to the buffer sent to user space. In this way, data in the buffer exist in the form of plain text, and the Linux buffering mechanism is fully utilized to improve the performance of the file system. The efficiency of the file system is improved by the buffer. If the encryption operation is located in the buffer, the buffer will be cipher, and the core needs to decrypt each time the data in the buffer needs to be read. Therefore, a reasonable way is to store plain text in the Linux buffering mechanism. If the data is in the buffer, the actual performance will not be lost too much. In the system, the file data is segmented according to the page size, and each segment is called an extended segment. For the Linux buffering mechanism, the size of the data block is related to the specific file system and is a multiple of 512. The maximum data block size is 4096 bytes, which is the size of a page. The operation of encryption and decryption is performed on data blocks. The size of the data that can be encrypted by an encryption algorithm at a time is called a data packet. Most symmetric encryption algorithms use 64-bit data packets. In this time, encryption uses 128-bit data packets, so a data block contains multiple data packets. The entire data block is encrypted using a 128-bit key symmetric encryption algorithm. In this method, the initial vector of the subsequent data packet is mainly provided by the encryption result of the previous data packet. For the first data packet, its initial vector is randomly generated and does not require encryption, and is stored in the extended section of the encrypted file in plain text. Then the encryption algorithm is:

$$W = P/Y + F \tag{3}$$

In formula (3), *W*represents the cipher encrypted based on attributes, *P* is the message to be encrypted, *Y*is the access control parameter associated with the access policy, and *F* represents the requester who satisfies the access policy.

After the data encryption is completed, the public key is input to generate a cipher encrypted based on the data attributes. Only requesters with access policies can decrypt the cipher. During the decryption process, the attribute set is associated with the access resource, and the receiver can access the cipher information according to his or her authorized attribute. This decryption process is suitable for access control applications such as private data sharing.

4.3 Network data security sharing

Completing the encryption of the network data through the above encryption algorithm. On the basis of the establishment of the above-mentioned secure transmission mechanism and the encryption of the network data, the network data is shared. When a node requests a node to share the sensing data, the node first checks the node identity with the key. After reaching a consensus, the node formulates access constraints (such as data sharing range, time limit and times), then decrypts the data according to the private key, which is provided by the node, and outputs the corresponding results according to the constraints. Before outputting the data to the node, using the above public key to encrypt the data, and decrypt it with its own private key. The details as follows:

In a shared access request, a node sends a sensory data sharing request to the node. The request contains information such as the purpose, time, and number of data accesses. After verifying the identity of the access node, the node formulates access constraints for the access, such as the data sharing range, timeliness and times, and then authorizes access, and sends these conditions and the pseudonym private key which is corresponded to the accessed data block to the neighboring data aggregator.

After the data aggregator verifies the information, the result is output, and it starts to execute the smart contract, locks the script. According to the access constraints set by the node, it decrypts the shared data according to the provided symmetric key. It uses the public key of the access node to perform asymmetric encryption on the shared data and output the result.

Sharing data transmission, if the data access node and the accessed node are within the coverage of the same data aggregator, the data aggregator directly sends the data to the data access node, otherwise, the current node executes the smart contract and sends the encrypted result to the adjacent data aggregator of the access node.

Accessing the specified data: after receiving the data, the data access node decrypts the data with its own private key and performs data read access.

In the process of network sharing, there exist multiple users sharing data. Thus, the multi person sharing technology of encrypted files must be realized through multi person digital envelope [3, 4, 10]. That is to add multiple digital envelopes to the encrypted file. When using it, users should first unpack the corresponding digital envelope with their private key, obtain the symmetric key, and then use the symmetric key to open the file. The digital envelope uses a symmetric cryptosystem and an asymmetric public-private key cryptosystem. The sender of the information firstly encrypts the information to be sent using a randomly generated symmetric password to obtain the encrypted result A, and then uses the public key of the receiver to encrypt the random symmetric password to obtain the encrypted result B. The encrypted result B is called a number the envelope. Digital envelope and encryption result A are combined to form the final encryption result (A + B). When transferring information, the information receiver must decrypt the digital envelope B with his private key to obtain the symmetric password, and then use the symmetric password to decrypt A to obtain the original text information, which ensures the authenticity and integrity of data transmission. Figure 4 indicates how a digital envelope works:

The application method of this technology in the secure storage system is as follows:

First, the basic conditions, each user has a personal digital certificate and corresponds private key (and stored in a hardware storage device such as an IC card, and KEY), and the digital certificates of all electronic document security storage system of system users are stored in the certificate issuing server of the PKI/CA basic platform. It is convenient for each user to obtain another digital certificate (public key) by email and certificate common name.

Second, all users account registration information in the electronic document secure storage system must contain an email address corresponding to their personal digital certificates.

Third, if user A adds or creates a file in the electronic document secure storage system, user A uses his own digital certificate to perform the encryption process of Figure C.

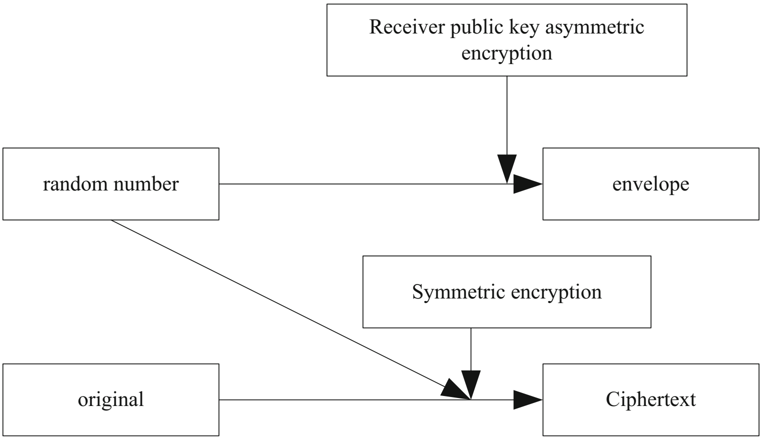

Fig. 4 Information encryption process

Fourth, if user A authorizes files which created by individuals can be read by user B and user C in the electronic document secure storage system, user A uses the digital certificates of user A and user B to encrypt the image C. Note: The digital certificates of user B and user C can be automatically obtained from the certificate issuing server through the email address in the user account and downloaded to the terminal of user A.

Fifth, if user A opens a file created or stored by an individual, or opens a file authorized by another person, user A uses the private key in the personal certificate to decrypt the picture D. Because these files are created or saved certificate of User A (the public key) is encrypted with a digital envelope, according to the technique described above, user A can naturally use their private key to unlock these files.

Sixth, if user A saves the file currently opened in the electronic document secure storage system, firstly confirms the allowed openers of the current file (if there are user D and user E in addition to user A), then the method of save is the same as that of authorizes files: using the digital certificates of user A, user D, and user E for the encryption process in Figure C.

The above six points can enable multiple people to share data. This process is actually to use the encrypted file multi-person sharing technology to solve the problem of permission control for files that are stored centrally. Besides, the addition, authorization, and saving processes of the above files all include the file encryption process. In actual applications, encryption is firstly performed on the local user and then uploaded to the server of the shared system. At the same time, the opening process of the above files includes the file decryption process. In the actual application, the files are downloaded from the server of the electronic document secure storage system, and then they are decrypted. So, the problem of network security transmission of centralized storage files can be solved. Table 5 indicates the key application response command formats and application command formats involved in this process:

When the network data is shared, the user may lose his password, and then the private key of the user is also lost. If there is no data recovery mechanism, user files will be lost forever. Therefore, by adding a data recovery function to the design of the network data security sharing system software, the administrator can recover the files of the user. Figure 5 indicates the data recovery process:

First, each file header has a file encryption key set. Every user who has permission to access the file has a key in the key set. This key is encrypted by the public key of the user. There is an

Table 5 Key application response command format and application command format

Key application response command format		Application order format	
Description of content	Length (bytes)	Description of content	Length (bytes)
Command length	4	Description of content	4
Command number	2	Command number	2
error code	2	Key type	2
Key number	8	Key usage	2
Key version number	4		
Key type	2		
Key usage	2		
Key digest algorithm	2		
Key digest value length	2		
Key digest value	L1		
Key content length	4		
Key content	L2		

administrator encrypted file key in the head of each file, so the administrator can use his private key to recover all files to complete the secure sharing of network data.

In summary, the software flowchart is shown in Fig. 6.

5 Experiment

To verify the actual application performance of the network data security sharing system based on the blockchain designed this time, experiments were conducted. The experiment took the system login response time, key distribution time, data encryption time, and key update time as the experimental objects. To ensure the rigor of the experiment, the literature [15] and literature [14] systems were compared with this design system, and the four experimental items were used as the objects to compare the real-time performance of the three systems.

5.1 Experimental environment

The implementation of the entire platform is implemented in the VS2010 development environment, by using C ++ language, configuring the test environment, and initializing the

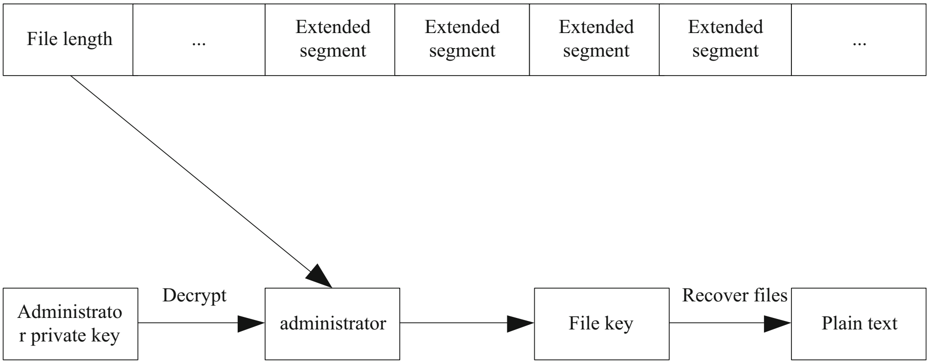

Fig. 5 Network security data sharing process

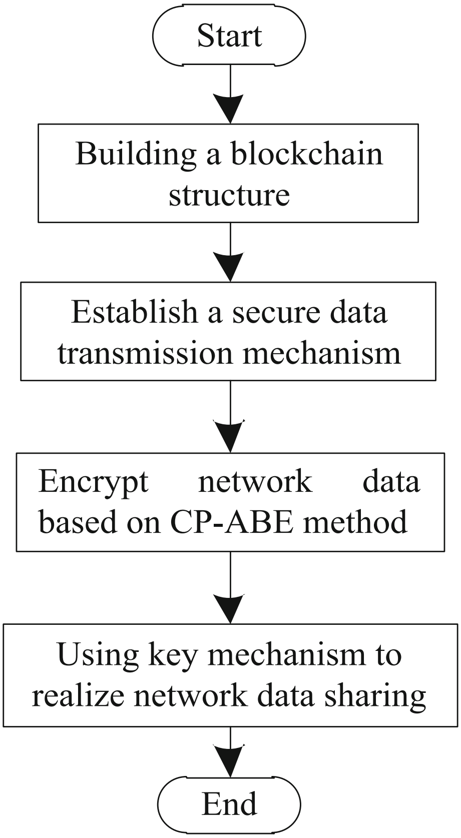
Start
Building a blockchain structure
Establish a secure data transmission mechanism
Encrypt network data based on CP-ABE method
Using key mechanism to realize network data sharing
End

Fig. 6 Software flow chart of network data security sharing system based on blockchain technology

experimental data, using the file data of a website as the experimental object. Table 6 indicates the main file operation permissions of the three systems under test:

After the experiment file is prepared, the daemon was started to connect the node server to the network. The file upload operation is performed to obtain the hash value of the file, and then turn to query or download files from multiple nodes based on the hash value.

To realize the communication between the user side and the server side, the hardware environment and the software environment need to be advanced to ensure normal communication. To build a blockchain network environment, Rinkeby is used, and it is an Ethereum testnet. The consensus mechanism used is PoA. Compared with PoW, the consensus algorithm has a controllable block generation time and the advantage of power consumption.

Table 7 indicates the hardware environment and its configuration:

Table 8 indicates the software environment installed on each computer:

In the experiment, the data sharing was monitored by file synchronization. The monitoring of file synchronization is built on the basis of multi-threaded transmission of safety and reliable transmission module. Through the application of Java Swing technology, the current synchronized user and the file information synchronized by the user are displayed. The implementation class of the main interface is the DownloadManager class in the cloud download. Manager package, which continuously extracts the intermediate data transmitted by other file transmission classes for real-time display of user synchronized file information.

Table 6 Experiment Folder Authorization Form

Serial number	content	specific description
1	Folder authorization	Right-click after selecting a folder in a shareable private file Right-click after selecting a file in a shareable private file After selecting the folder, click Authorize in the right-click menu
2	Verify authorization	Read-only authorize a folder to other users Download and authorize a folder to other users Can authorize a folder to other users
3	Verification permission canceled	Authorize multiple folders to the same person first, and then cancel the authorization of one of the folders
4	Subdirectory authorization	Authorize the parent and child directories to the same person Authorize the parent directory and the subdirectory to the same person at the same time, the subdirectory permissions are less than the parent directory
5	Duplicate folder node authorized by others	The parent directory and the sub-directory are authorized at the same time, and the directory node of the authorized person will be repeated. At this time, the path information of all nodes is unified to be shorter
6	Number of unread files	The authorized person checks whether the number of unread files authorized by others is correct Open, download, copy unread files Reauthorize and modify the read files

Table 9 indicates eight different size files shared by the experiment:

During the experiment, the default transmission thread of each file is set to four, and the above 8 files of different sizes are selected for synchronization. The size of the files that each user wants to share is different, showing a gradually increasing trend. Set 1000 users to the system login response time, key distribution time, data encryption time and key update time respectively. The top 500 used this design system, and the last 500 used traditional systems for data sharing. The systems being compared were implemented in the same environment and optimized on average. The user network topology is a ring topology, and the structure is that the nodes form a closed ring. Each node in the ring network is connected in a closed ring communication line connected end to end through a ring interface, and any node on the ring can request to send information. The transmission medium is from one end user to another end user until all end users are connected into a ring. Data is transmitted between nodes in one direction in the loop, and information is transmitted from one node to another.

Table 7 Hardware environment table

Equipment use	basic configuration	IP address	other instructions
Network safe server	P4 3.0, 512 M memory, DOM disk, 4X100M network card	192.168.200.250	
File server 1	INTEL Xeon 3.0, 1GM memory, 120G, 100 M network card	192.168.101.220	Linux samba
File server 2	P4 3.0, 1G memory, 120G, 100 M network card	192.168.200.11	Winxp pro+SP2
PC 1	P4 1.8, 512 M RAM, 80 hard disk	192.168.200.12	Winxp pro+SP2
PC 2	P4 3.0, 512 M RAM, 120 hard disk	192.168.200.13	Winxp pro+SP2

Table 8 Software environment table

Serial number	Equipment category	Software name and version
1	Network safe server	Network safe 5.5.0
2	PC 1	File certificate, network safe CSP client
3	PC 2	Minghua key driver, network safe SDK Client

5.2 Comparison of system login time

System login time test refers to the test of system response performance. Only when the index requirements are met, can the realized system have the ability to be put into practical application. Under the normal network connection, the normal response time of system login is 5 s, for 1000 users A total of 8 tests were performed, and the experimental comparison results are shown in Fig. 7.

According to Fig. 7, under 8 tests, the login time of the user using this design system is less than 5 s, generally about 3 s. The system login response is faster, while the document [15] and document [14] system login The time is generally longer, generally above 6 s, up to 8 s, and the system login time is longer.

5.3 Comparison of key distribution time

Key distribution time mainly includes user registration, user symmetric parameter negotiation, user attribute private key generation, etc. The experimental comparison results are shown in Fig. 8.

Analyzing Fig. 8 we can see that the amount of data contained in each test is different. When the file size increases, there is a large gap in the key distribution time for users. Figure 8 indicates that the users who use this design system are less affected by the file size, and the key distribution time is shorter, less than 100 s. However, users who apply the document system

Table 9 Experimental file parameters

Serial number	Login user name	Size of test files
1	openlab00	45 M
2	openlab01	68 M
3	openlab02	125 M
4	openlab03	132 M
5	openlab04	206 M
6	openlab05	448 M
7	openlab06	680 M
8	openlab07	1128 M
9	openlab08	45 M
10	openlab09	68 M
11	openlab010	125 M
12	openlab011	132 M
13	openlab012	206 M
14	openlab013	448 M
15	openlab014	680 M
16	openlab015	1128 M

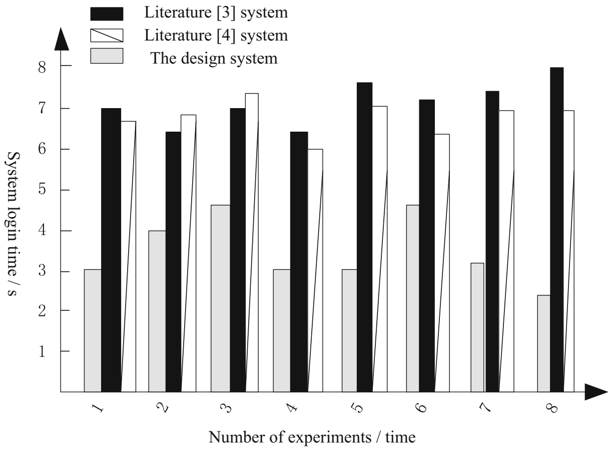

Fig. 7 System login time comparison

are greatly affected by the file size, and the key distribution time gradually increases, and the distribution time is longer, up to 380 s. This is because the larger the user's file, the longer it takes to encrypt the data, which increases the data transmission time.

5.4 Comparison of data encryption time

After the users generating the symmetric parameter and attributes private key, the system is used to encrypt the data. The number of attributes in the user private key and access control structure is set to 0 in the experiment. Figure 9 indicates the time results of two systems encryption.

According to the above experimental results, it can be seen that users who applied this design system have much shorter data encryption time, around 50 s, and much faster response time. The literature [15], literature [14] system are affected by the increase of the shared data by user. The gradually increase of data encryption time, and the longer encryption time.

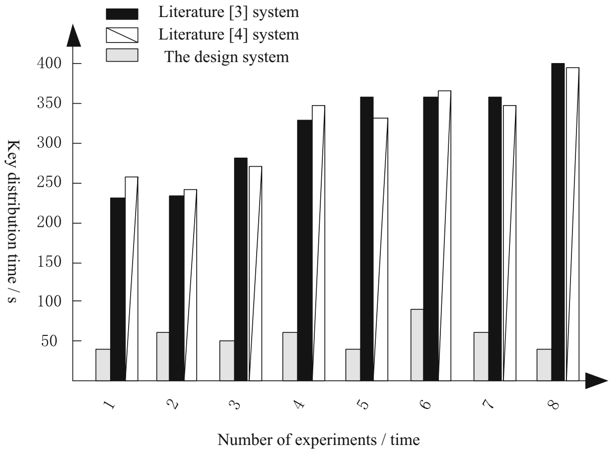

Fig. 8 Comparison of key distribution time

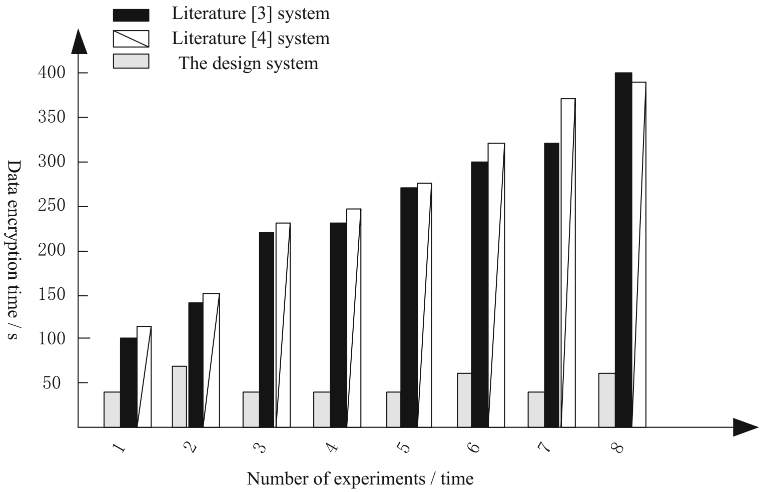

Fig. 9 Comparison of data encryption time

5.5 Comparison of key update time

Figure 10 indicates the result of key update time comparison:

The comparison results indicate that the key update time of the design system is very small. Generally, the key can be updated in time after the key is issued, which can reduce user resource consumption and improve the real-time performance of the network data security sharing system. The literature [15] and literature [14] systems consume a lot of time in the key update, which seriously increases the response time of the data sharing system.

The comparison results indicate that the network data security sharing system based on blockchain in this paper takes much less time than the traditional system in system login response time, key distribution time, data encryption time, and key update time. These results show that the real-time performance of the designed system is better than that of the traditional system.

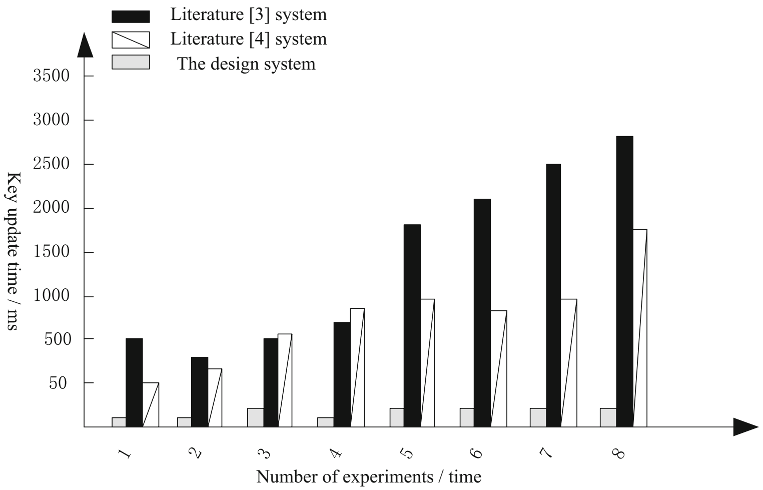

Fig. 10 Key update time

Geng [4] et al. proposed a heterogeneous data integration sharing system for coastal international trade. To solve the problem of insufficient performance of traditional heterogeneous data integration and sharing systems, a coastal international trade heterogeneous data integration and sharing system based on middleware and XML was designed. The system introduces the key technologies of system design, namely middleware and XML. Then designed a four-tier architecture based on middleware and XML, and then analyzed heterogeneous data integration modules, including intermediary programs, packaging programs and Web service management. Finally, the system was implemented and tested. The results show that the system has excellent query speed, accuracy, integration and whether it supports data structure. However, the performance of the system in terms of data encryption and system operation stability is not satisfactory.

The system login time determines the stability and efficiency of the system. Experiments show that the network data security sharing system based on blockchain greatly shortens the login time required by the system and provides a stable platform for data encryption.

Key distribution time mainly includes user registration, user symmetric parameter negotiation, user attribute private key generation. The network data security sharing system based on the blockchain can shorten the key distribution time and quickly realize the generation of user attribute private key, which plays a vital role in data encryption security.

Data encryption time is the most crucial parameter of network data security sharing system. The shorter the data encryption time is, the faster the data response time will be. The data encryption time of the designed system is the shortest, showing that the system can quickly realize data encryption.

6 Conclusion

This paper designs a network data security sharing system design based on blockchain technology. Experimental analysis shows that the system's login response is faster, users are less affected by file size, key distribution time is shorter, and data encryption time is shorter. Short, can quickly realize the key update, has better encryption, and the system runs efficiently.

There are still many security issues in the data sharing system that need to be resolved. In the actual application environment, the threat of conspiracy attacks by the two parties still exists. How to study a more reasonable solution for multi-key management to resist collusion attacks from the key management party is a key to solving ky security.

Declarations

Conflict of interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Abdulla AA (2015) Exploiting similarities between secret and cover images for improved embedding efficiency and security in digital steganography. https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.694896
2. Al-Saleem SM, Ali A, Khan N (2018) Energy efficient key agreement scheme for ubiquitous and continuous remote healthcare systems using data mining technique [J]. Clust Comput 21(5):1–12

3. Feilong T, Yang LT, Tang C et al (2018) A dynamical and load-balanced flow scheduling approach for big data centers in clouds [J]. IEEE Trans Cloud Comput 6(4):915–928
4. Geng D (2020) Design of heterogeneous data integration and sharing system for coastal international trade [J]. J Coast Res 103(sp1):718
5. Hassija V, Chamola V, Garg S, Krishna DNG, Kaddoum G, Jayakody DNK (2020) A Blockchain-based framework for lightweight data sharing and energy trading in V2G network [J]. IEEE Trans Veh Technol 69(6):5799–5812
6. Herrera-Quintero LF, Samper-Zapater JJ, Svitek M, David W (2018) Special section on ITS services to Smart City context [guest editorial][J]. IEEE Intell Transp Syst Mag 10(2):4–5
7. Leeuwen GV, Alskaif T, Gibescu M et al (2020) An integrated blockchain-based energy management platform with bilateral trading for microgrid communities [J]. Appl Energy 263(C):114–123
8. Lei K, Zhang Q, Lou JJ et al (2019) Securing ICN-based UAV Ad Hoc networks with blockchain[J]. IEEE Commun Mag 57(6):26–32
9. Li CT, Lee CC, Weng CY (2018) A secure three party node authentication and key establishment scheme for the internet of things environment [J]. J Internet Technol 19(1):147–155
10. Li XY, Ma HD, Yao WB et al (2018) Data-driven and feedback-enhanced trust computing pattern for large-scale multi-cloud collaborative services [J]. IEEE Trans Serv Comput 11(4):671–684
11. Liao X, Li K, Yin J (2017) Separable data hiding in encrypted image based on compressive sensing and discrete fourier transform [J]. Multimed Tools Appl
12. Liao X, Guo S, Yin J et al (2017) New cubic reference table based image steganography [J]. Multimed Tools Appl 77(4):1–18
13. Nowlin BT, Wang J, Schafer JL et al (2018) Monocyte subsets exhibit transcriptional plasticity and a shared response to interferon in SIV-infected rhesus macaques [J]. J Leukoc Biol 103(1):141–155
14. Pattipati DK, Tentu AN, Venkaiah VC et al (2016) Sequential secret sharing scheme based on level ordered access structure [J]. Int J Netw Secur 18(5):874–881
15. Shen J, Liu D, Lai C-F et al (2017) A secure identity-based dynamic group data sharing scheme for cloud computing [J]. J Internet Technol 18(4):833–842
16. Singh M, Aujla GS, Bali RS (2020) A deep learning-based blockchain mechanism for secure internet of drones environment [J]. IEEE Trans Intell Transp Syst PP(99):1–10. https://doi.org/10.1109/TITS.2020.2997469
17. Wang J, Gao J, Yang X et al (2019) Online data reveal key factors on salary expectation [J]. Dianzi Keji Daxue Xuebao/J Univ Electron Sci Technol China 48(2):307–314
18. Wen F, Min F, Zhang YJ, Yang C (2019) Crude oil price shocks, monetary policy, and China's economy [J]. Int J Financ Econ 24(2):812–827
19. Yang J, Wen J, Jiang B, Wang H (2020) Blockchain-based sharing and tamper-proof framework of big data networking [J]. IEEE Netw 34(4):62–67
20. Yin H, Guo DC, Wang K et al (2018) Hyperconnected network: a decentralized trusted computing and networking paradigm [J]. IEEE Netw 32(1):112–117

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.