Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 14, no. 4, April 2018, pp. 765-784
DOI: 10.23940/ijpe.18.04.p19.765784

A Multi-Transaction Mode Consortium Blockchain

Jiarui Zhang*

Department of Computer Science and Technology, Hefei University, Hefei, 230601, China

Abstract

It is difficult to apply the bitcoin transaction system's blockchain technology outside the electronic currency transaction. This paper proposes a Multi-Transaction Mode Consortium Blockchain (MTMCB), which generalizes the traditional buying and selling transaction into event processing of the business system. It prevents all kinds of high value event processing results from repudiation and tampering. MTMCB is low-cost and obtains credible results without accumulating technology. It extends the user type (not only the PC user), the storage type (not only local storage) and the storage content (not only the block data). MTMCB optimizes transaction verification, consensus decision, block generation, block storage, comparative verification and block linking mechanism. It provides visual audit services, including event proof, abnormal transaction early warning, tampering discovery and reconstruction, etc. Meanwhile, example simulation, security analysis and performance analysis are processed. Its results show that MTMCB has the same security level as the bitcoin core transaction system, and it has good adaptability in data resource protection of high value event processing.

Keywords: event processing; transaction validation; consensus decision; block generation; block storage; comparison validation; block linking

(Submitted on January 13, 2018; Revised on February 12, 2018; Accepted on March 23, 2018)

1. Introduction

Blockchain technology was invented by Satoshi Nakamoto in 2008, which was recorded in "Bitcoin: A Peer-to-Peer Electronic Cash System". Its principal application is: supporting the "decentralization" and non-server architecture of electronic currency transactions. Blockchain has a clever structure, security of the algorithm, small occupancy of bandwidth, anti-repudiation and tampering prevention, etc. One of the most important features is getting reliable results in a non-trust environment with low cost and no accumulation. Consortium Blockchain is a type of blockchain that needs to be licensed. It is also known as the Permissioned Blockchain. The Consortium Blockchain network is composed of several ordinary transaction nodes and only one regulatory node. At present, the representative of the Consortium Blockchain is R3 (Corda distributed book) and Hyperledger. The core technology of the existing blockchain has the following technical features:

- Single transaction type. It can only adapt to "buy" and "sell", and all of its transactions are converted to the "transfer" mode of the process from sellers to the buyers.
- The transaction validation and the block storage cannot have appendixes. Which means picture, image, audio, and video data of the transaction process or result cannot be attached.
- The block data is stored locally. The block data is only saved in a local block node. Mobile devices and ATMs cannot be used as transaction nodes because they cannot be stored locally.
- The block generation efficiency is controlled by the value of "Bits". It generates a block about every 10 minutes.

The above four technical characteristics allows the existing blockchain technology to be only applied to the PC based encryption currency buying and selling transactions.

2. Related Work

Blockchain technology has been available for ten years with the birth of the Bitcoin transaction. During the period, research

* Corresponding author.
E-mail address: zhangjiarui099@163.com

on it was given considerable attention by the public. According to the technical level, currently, the research on blockchain technology can be divided into the following types:

- Introduce, publicize and prospect blockchain technology and describe its application prospect.
- An in-depth analysis of the internal structure and safety features of the existing blockchain technology.
- Based on blockchain technology, it proposes an idea, concept, solution, and architecture to solve some of the difficult problems
- Learn or upgrade some of the features of blockchain and develop a specific application to produce special application effects in some areas

Abbasi et al. [1] proposed a novel technique, named VeidBlock, to generate verifiable identities by following a reliable authentication process. These entities are managed by using the concepts of blockchain ledger and are distributed through an advance mechanism to protect them against tampering. All identities created using VeidBlock approach are verifiable and anonymous. Therefore, it preserves user privacy in the verification and authentication phases. Low-power IoT end-devices do not possess enough horsepower to run a software client for intensive blockchain calculations.

Chakravorty et al. [2] discussed the impact of blockchain technology on social media networks. Hence, they proposed a new social media network technology based on blockchain with the construction centered around the user. It allows users to track and share each block of content ownership control and formed a truly decentralized, secure, and anonymous distribution network. Hardjono et al. [5] described a privacy-preserving method for commissioning an IoT device into a cloud ecosystem. They introduce the ChainAnchor architecture that provides device commissioning in a privacy-preserving way.

Leiding et al. [6] combined Vehicle Ad-hoc Networks (VANETs) and Ethereum's blockchain-based application concepts to enable a transparent, self-managed and decentralized system, which is self-regulating and in no need of a central managing authority. Li et al. [7] proposed a new blockchain architecture designed to meet industrial standards. They take advantage of the concept of satellite chains that can run in parallel without negotiating a consensus protocol, thus greatly enhancing the scalability of the architecture.

Nijeholt et al. [9] stated a DecReg framework based on blockchain technology to solve the "double-financing" problem in Factoring. Ricardo Neisse and Steri et al. [10] proposed the use of a blockchain-based approach to support data accountability and provenance tracking. Their approach relies on the use of publicly auditable contracts deployed in a blockchain, which increase the transparency with respect to the access and usage of data.

Özyılmaz et al. [11] created a proof of concept to enable low-power, resource-constrained IoT end-devices access to a blockchain-based infrastructure. To achieve this aim, an IoT gateway is configured as a blockchain node and an event-based messaging mechanism for low-power IoT end-devices is proposed. Raju et al. [12] mentioned a data bank idea, which is used for data decentralization storage in medical care and the education field.

Svetinovic [13] proposed the integration of blockchain technology and Internet of things. Meanwhile, he discussed the decentralized smart grid energy trading security and privacy issues based on blockchain technology. Vo et al. [14] demonstrated a blockchain-based solution for transparently managing and analyzing data in a pay-as-you-go car insurance application. This application allows drivers who rarely use cars to only pay insurance premiums for particular trips they would like to travel. The solution ensures that all the data pertaining to the actual trip and premium payments made by the users are transparently recorded so that every party in the insurance contract including the driver, the insurance company, and the financial institution is confident that the data are tamper-proof and traceable.

Chanson et al. [3] showed how blockchain technology can enable privacy by presenting an odometer fraud prevention system. It records mileage and GPS data of cars and secures that on the blockchain, which strongly hinders odometer fraud. Users own and control their data, and, at the same time, data integrity is ensured. This facilitates the certification of that data. Decker et al. [4] developed a high security PeerCensus system. PeerCensus acts as a certification authority, which manages peer identities in a peer-to-peer network and ultimately enhances Bitcoin and similar systems with strong consistency.

Liang et al. [8] propose a decentralized and trusted cloud data provenance architecture using blockchain technology. Blockchain-based data provenance can provide tamper-proof records, enable the transparency of data accountability in the cloud, and help enhance the privacy and availability of the provenance data. They make use of the cloud storage scenario and choose the cloud file as a data unit to detect user operations for collecting provenance data. They design and implement

ProvChain, an architecture to collect and verify cloud data provenance, by embedding the provenance data into blockchain transactions.

Xu et al. [15] proposed EPBC (Efficient Public Blockchain Client), a novel and efficient transaction verification scheme for public ledgers, which only requires lightweight users to store a small amount of data that is independent of the size of the blockchain. Xing et al. [16] proposed BGPCoin, which is a trustworthy blockchain-based Internet resource management solution. It provides compliant resource allocations and revocations, as well as a reliable origin advertisement source. By means of a smart contract to perform and supervise resource assignments on the tamper-resistant Ethereum blockchain, BGPCoin yields significant benefits in the secure origin advertisement and dependable infrastructure for object repository compared with RPKI.

Zupan et al. [17] developed a HyperPubSub system. It provides middleware for automation of intelligent contracts. It is mainly used in the business chain technology application in the vertical market. The system builds a decentration of consortium blockchain, providing publish / subscribe, communication environment, message security and privacy protection.

To summarize, most existing research focuses on the decentralization and distributed storage characteristics of blockchain technology and solves the problems of end-devices lightening, security, privacy protection and data tampering in some specific fields. While the comprehensive innovation of the existing blockchain technology makes it more widely applicability, the research of important scientific impact is very rare.

In this paper, a Multi-Transaction Mode Consortium Blockchain (MTMCB) is designed. The existing blockchain technology has been completely innovating:

- The types of transactions are diversified. The traditional buying and selling transaction is generalized to the event processing of the business system. Therefore, a database operation, an approval, a certification, a visit, a responsibility identification, a property right judgment or other high value event processing can be used as a transaction.
- The storage content is diversified. It can not only store block data corresponding to events, but also store events related to multi-format appendix data, such as pictures, images, audio, video files and other evidence or results.
- The type of storage is diversified. The block data and appendix data can be stored locally and can be stored on the cloud end corresponding to the user's address according to the user type.
- The type of user is diversified. They can be PC users, mobile devices users, and ATM users.
- The speed of block generation is determined by the processing speed of the MTMCB system, which is not limited by human beings.
- The transaction validation, consensus determination, block generation, block storage, comparison verification and block linking mechanism are optimized.
- It provides a visual audit service, including proof of service, abnormal transaction early warning, tampering discovery and reconstruction.

Compared with the existing blockchain technology, MTMCB has better and wider applicability. All kinds of high value event processing results can achieve anti disavow, anti-tampering, low cost and credible results through MTMCB. It will greatly expand the application realms of blockchain technology.

3. The Design of MTMCB

3.1. Summary

MTMCB includes the Regulatory Node System (RNS) and the Transaction Node System (TNS). RNS is deployed on a PC or a server in the consortium blockchain network. It includes three parts: initialization, transaction process and audit service. Among them, the initialization part only needs to be executed once, the transaction process is processed for daily operation, and the audit service can be executed randomly according to the needs. TNS is deployed under the specified directory of each user node or the cloud specified directory of the user address. It includes: end-transaction version management program, transaction verification program, block generation program and block storage program. The transaction nodes can be PC users, mobile devices users and automated teller machines (ATM) users, etc. The data interaction between RNS and TNS is realized through the JSON RPC peer to peer communication mechanism. The node address generation, transaction process, data transmission, block data storage and so on are processed by security encryption mechanisms. RNS is associated with the business system in real time through "bridging". (Figure1. and Figure2.)

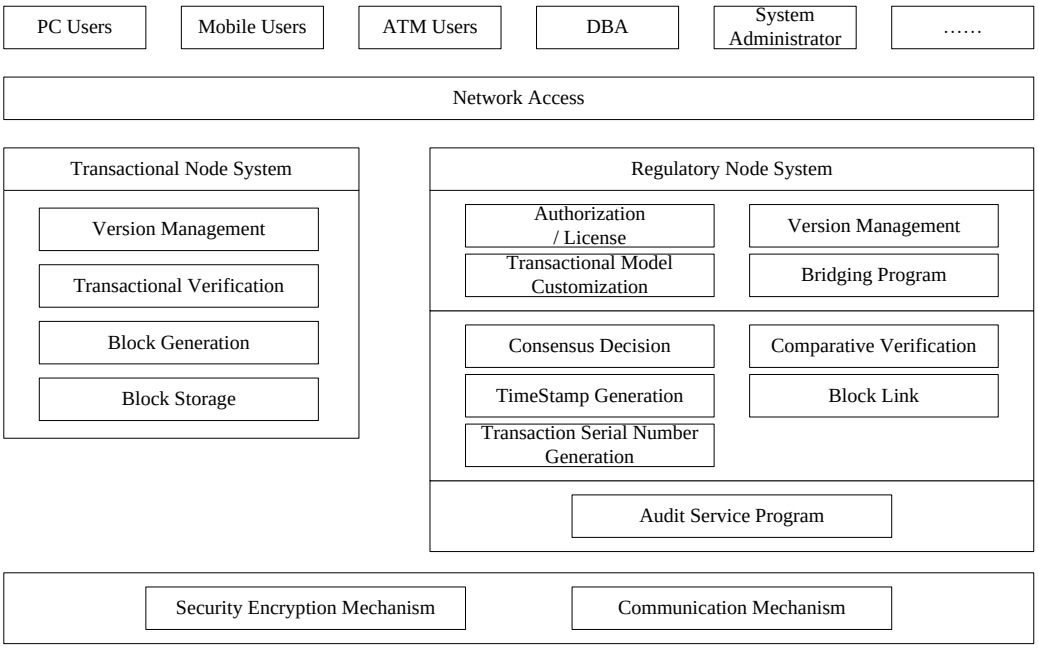

Figure 1. The overall architecture diagram of MTMCB

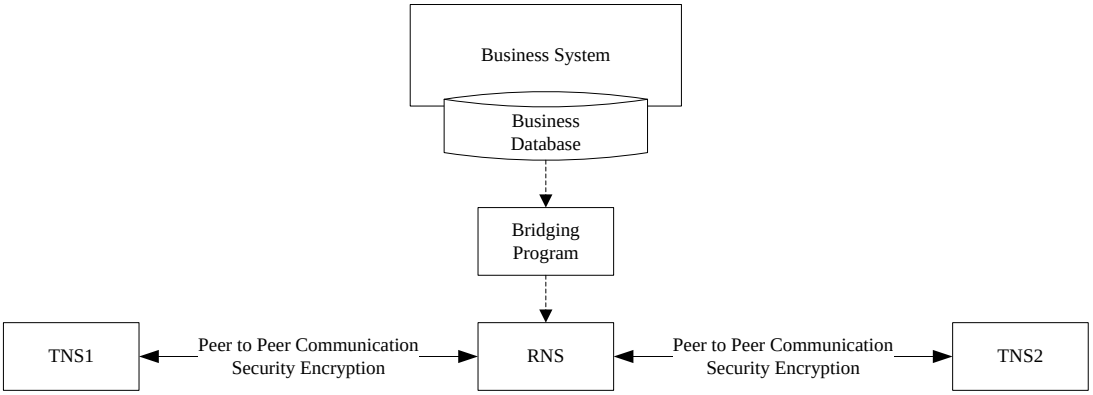

Figure 2. The relational schematic diagram of consortium blockchain and business system

3.2. The Initialization Section

The initialization section includes the version management program at the regulatory node end, the authorization / license program, the transaction model customization program and the bridging program.

- The Version Management Program at the Regulatory Node End

It responds to the upgrade RNS itself. It is responsible for packaging the upgraded TNS, data files, and setting up the upgrade sign.

- The Authorization / Licensing Program

It authorizes / licenses the transaction nodes or users who intend to join the consortium blockchain, returns the address, marks the storage characteristics of the node or user, applies for space, and automatically installs the version management program. The results of the operation are stored in the user file. See Figure 3, Figure 4, Figure 5 and Figure 6.

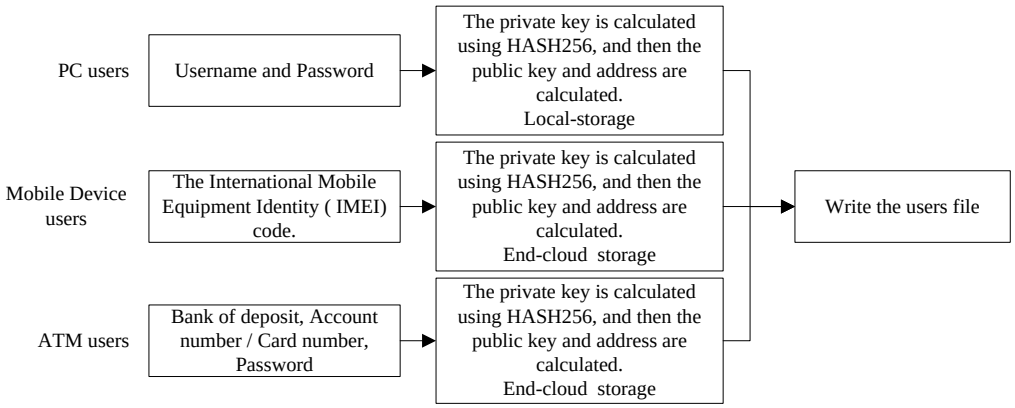

Figure 3. The authorization / licensing flow diagram

User Type	;	Username	;	Address	;	Public Key	;	Local Access Entrance	;	HOME Directory	;	The Date of Joining	;	The SHA256 Value of this Record	#

Figure 4. The storage structure schematic diagram of the User File (PC users)

In the figure above:

"User Type": 1 digits, the range from 0 to 2. 0 represents the regulatory node, 1 represents the DBA user, 2 represents ordinary PC user.

";": separators, the same as below.

"Username":20 characters. the user's name.

"Address": 10 characters. the user's transaction address.

"Public Key": 32 characters. the user's public key.

"Local Access Entrance":30 characters. including the IP address and port number in the local.

"HOME Directory": 50 characters.

"The Date of Joining": the date that the node or the user joins the consortium blockchain.

"The SHA256 Value of this Record": 32 characters. checkout item. the SHA256 value associated with all the previous data items.

"#": line-end symbol. the same as below.

User Type	;	Bank of Deposit	;	Account NUmber / Card Number	;	Address	;	Public Key	;	End Could Access Entrance	;	HOME Directory	;	The Date of Joining	;	The SHA256 Value of this Record	#

Figure 5. The storage structure schematic diagram of the User File (ATM users)

In the figure above:

"User Type": 1 digits, the value must be 3. to represent the ATM user.

"Bank of Deposit":60 characters. the bank of deposit of account number or card number.

"Account Number / Card Number": 20 characters. the account number or card number of the user transaction.

"End Could Access Entrance":30 characters. including the IP address and port number in the end-could.

User Type	;	The IMEI Code	;	Address	;	Public Key	;	End Could Access Entrance	;	HOME Directory	;	The Date of Joining	;	The SHA256 Value of this Record	#

Figure 6. The storage structure schematic diagram of the User File (mobile device users)

In the figure above:

"User Type": 1 digits, the value must be 4. to represent the mobile device user.

"The IMEI Code":20 characters. the International Mobile Equipment Identity code.

- The Customizing Program for Transaction Model

In the specified business system, it abstracts its characteristics and configures the transaction coding, transaction name, transaction structure collection, appendix mark, transaction description, and so on. The result of configuration is written into the transaction model file. See Figure7. and Figure8.

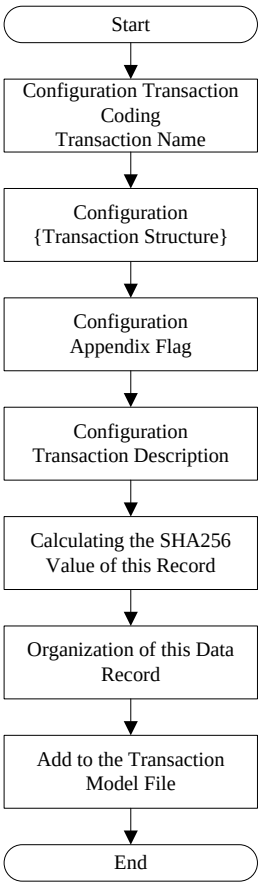

Figure 7. The transaction model customization flow diagram

Transaction Coding	;	Transaction Name	;	{Transaction Structure}	;	Appendix Flag	;	Transaction Description	;	The SHA256 Value of this Record	#

Figure 8. The storage structure schematic diagram of the Transaction Model File

In the figure above:
"Transaction Coding": 2 digits, 00 to 99.
"Transaction Name":20 characters.
"Transaction Structure": a set, it's the abstraction of the event.
"Appendix Flag": 1 digit, 0 or 1. 0 is no appendix, 1 is appendix.
"Transaction Description":40 characters.

- The Bridging Program

It is installed in a business system background. In normal operation, business management, and system maintenance processes, it catches the current business data and attachments by monitoring real-time capture. Extracting data and generating transaction data depends on the agreement of the business transaction structure in the model file. After that, the data is submitted to RNS.

3.3. The Transaction Processing Section

- The Transaction Processing Model

In the figure below:

Only an ordinary transaction node is drawn. If there are more than one, the processing process is the same as that of the above.

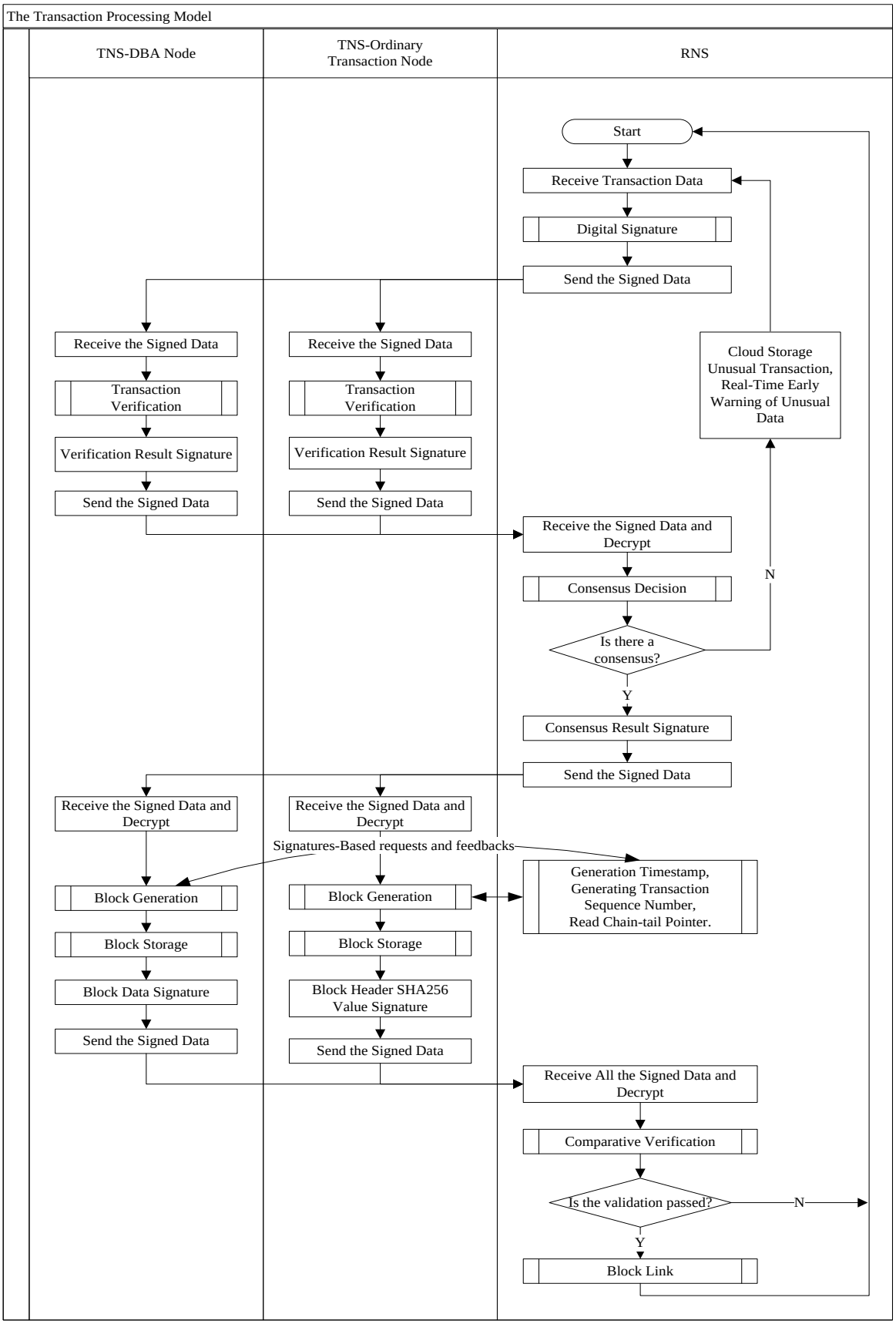

Figure 9. The business model diagram of transaction process

- The Digital Signature Flow

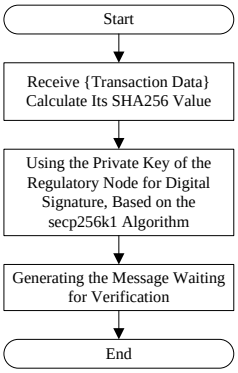

Figure 10. The digital signature flow diagram

- The Transaction Verification Flow

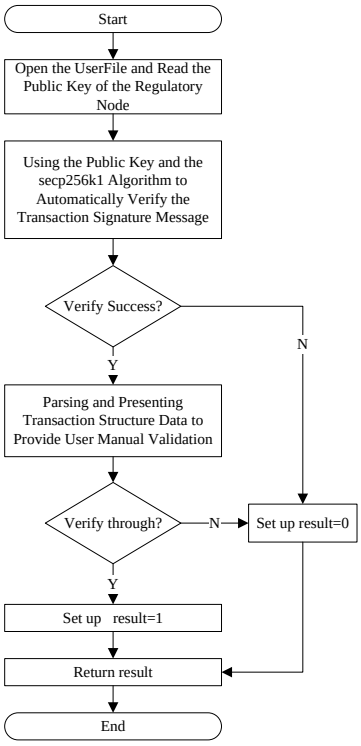

Figure 11. The transaction verification flow diagram

- The Consensus Decision Flow

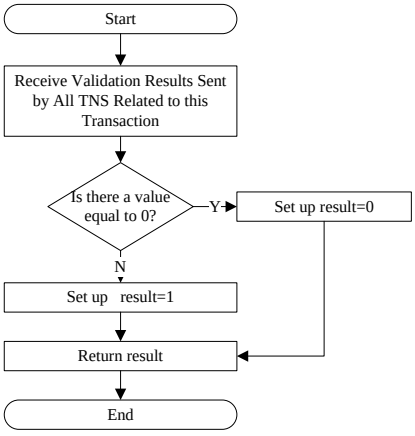

Figure 12. The consensus decision flow diagram

- The Block Generation Flow

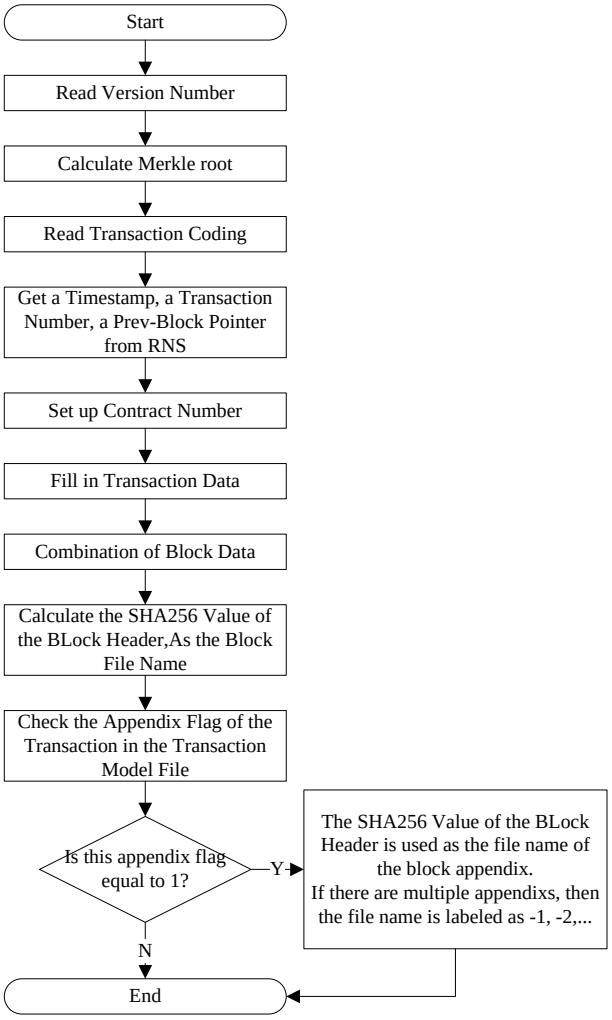

Figure 13. The block generation flow diagram

In the figure above, the block structure and the block header structure are shown in Table 1 and Table 2.

Table 1. The block structure

Size (Bytes)	Item
90	Block Header
Variable Byte	Transaction Data

Table 2. The block header structure

Size (Bytes)	Item
4	Version Number
32	Prev-Block Pointer
32	Merkle root
4	Timestamp
2	Transaction Coding
8	Transaction Number
8	Contract Number

For the storage structure of the transaction sequence number file, see Figure14.:

Transaction Coding	;	Current Sequence Number	;	The SHA256 Value of this Record	#

Figure 14. The storage structure of the Transaction Sequence Number File

In the figure above:
“Transaction Coding”: 2 digits, 00 to 99.

"Current Sequence Number": 8 Bytes, the current sequence number corresponding to the transaction coding.

- The Block Storage Flow

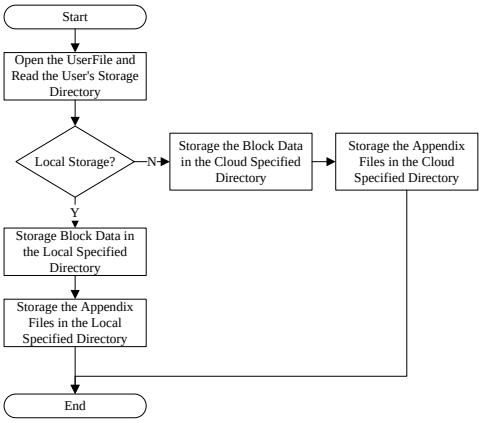

Figure 15. The block storage flow diagram

- The Comparative Verification Flow

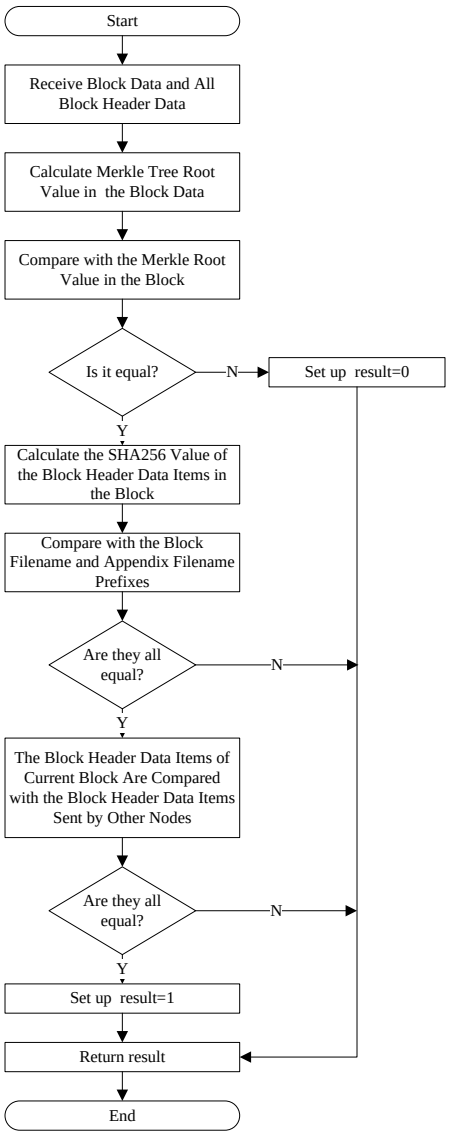

Figure 16. The comparative verification flow diagram

- The Block Linking Flow

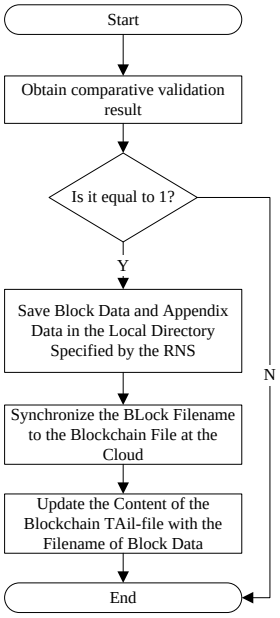

Figure 17. The block linking flow diagram

The Schematic Diagram of the Blockchain, see Figure18:

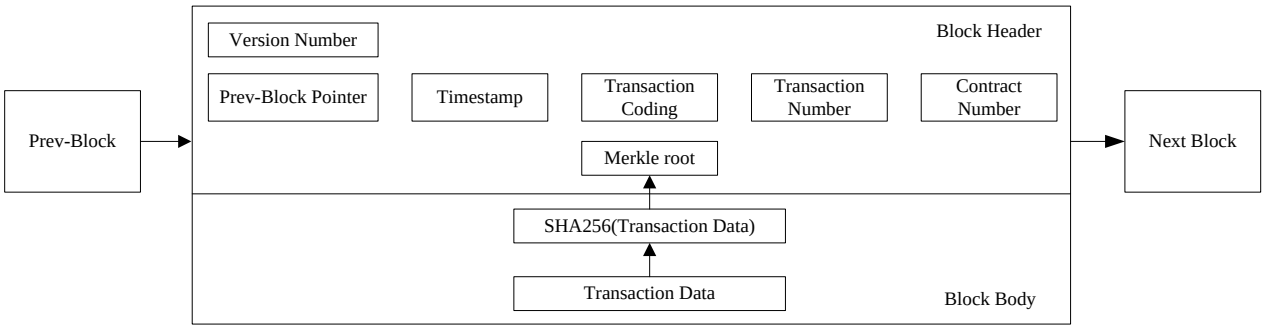

Figure 18. The schematic diagram of the blockchain

The storage structure of the end-cloud blockchain file, see Figure19.:

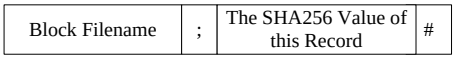

Figure 19. The storage structure of the End-Cloud Blockchain File

In the figure above:
"Block Filename":32 Bytes, the name of block data file.

The storage structure of the Chain-Tail file, see Figure20.:

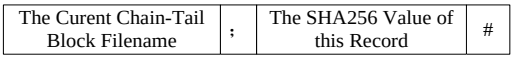

Figure 20. The storage structure of the Chain-Tail File

In the figure above:
"The Current Chain-Tail Block Filename":32 Bytes.

3.4. Audit Services Section

- The Overall Flow of Audit Services

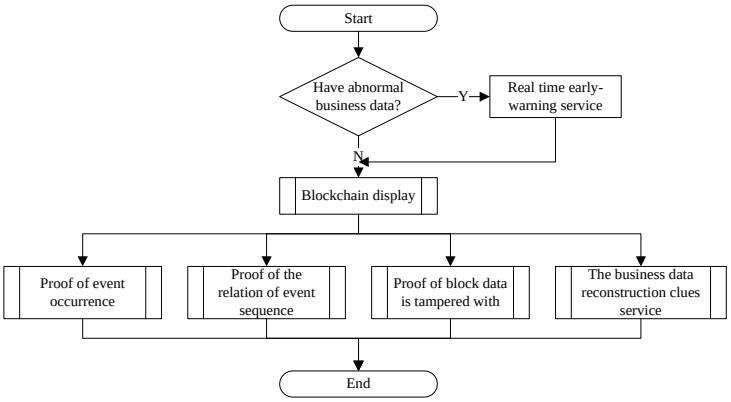

Figure 21. The overall flow diagram of audit services

The storage structure of the unusual transaction data file, see Figure22.:

Time	;	{Transaction Data}	;	Transaction Node Number	;	{User Address}	;	Operation Command	;	The SHA256 Value of this Record	#

Figure 22. The storage structure of the Unusual Transaction Data File

In the figure above:
"Time": The format is YYYY:MM:DD:hh:mm:ss.
"{Transaction Data}": a set, the elements of the collection are not authenticated transaction data.
"Transaction Node Number": 1 digit, the number of users of the transaction.
"{User Address}": a set, the user's address, the number of elements of the set equals the "Transaction Node Number".
"Operation Command":200 Bytes, the operation command corresponding to the transaction.

- The Process Flow of Blockchain Display

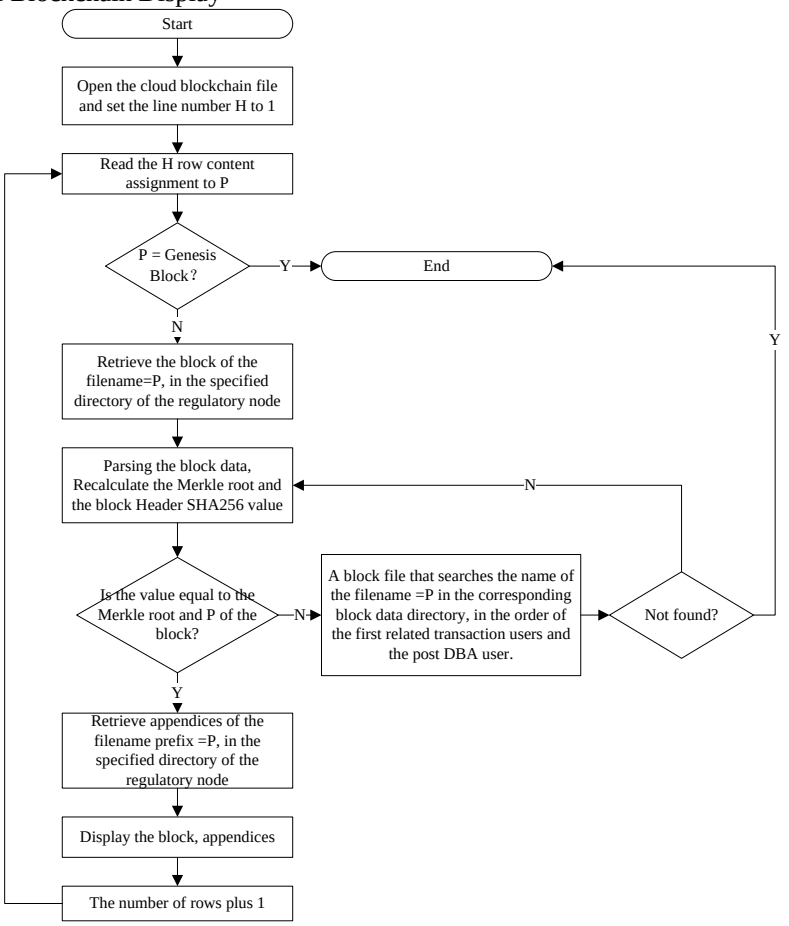

Figure 23. The process flow diagram of blockchain display

- The Process Flow of Event Happening Proof

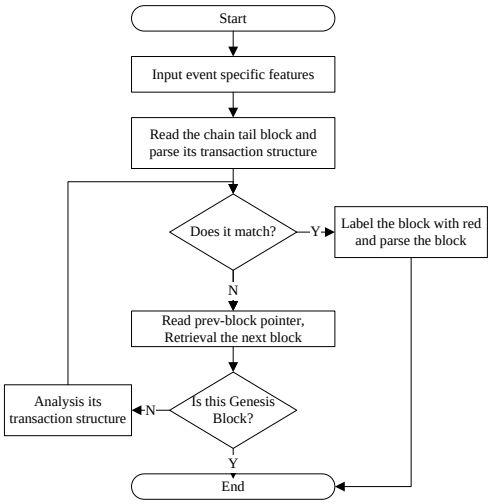

Figure 24. The process flow diagram of event happening proof

- The Process Flow of Event Sequence Relation Proof

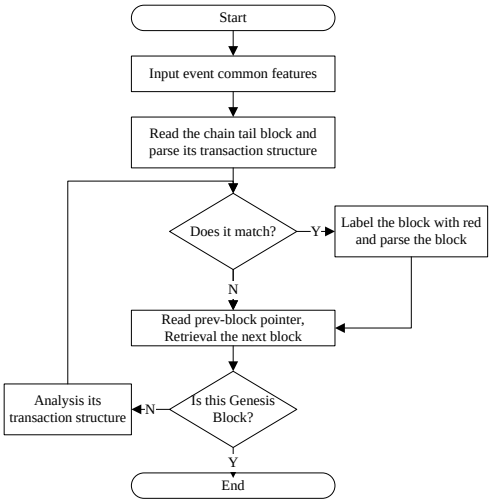

Figure 25. The process flow diagram of event sequence relation proof

- The Process Flow of Finding Block Data Was Tampered With

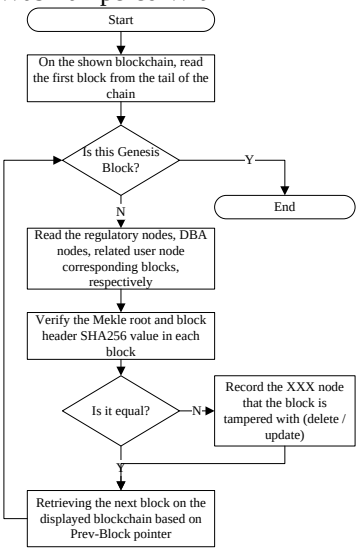

Figure 26. The process flow diagram of finding block data was tampered with

- The Process Flow for Providing Data Reconstruction Cue Services

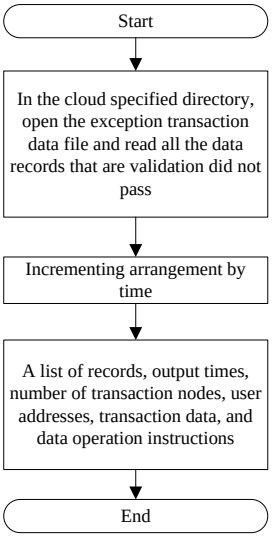

Figure 27. The process flow diagram for providing data reconstruction cue services

4. Example simulation

A system of registration of (knowledge / matter) property rights operated by the greffier in accordance with the prescribed business flow is conducted. The results of the operation are stated below: the applicants who are in conformity with the registration conditions are registered with the property rights, and the registration certificates of property rights are issued. The system's management business database is operated by DBA.

It is assumed that the use unit of the system is the "A City Property right Management Center", the user name of the greffier are "Zhangwei" and "Wangfang", and the username of DBA is "admin". The structure of the data table for the registration of property rights is shown in Table 3.

Table 3. The structure of information table of property right registration

Name of Item	Label of Item	Data Type	Primary Key	Empty
ID	ID Number	Integer	Yes	No
RequireName	Applicant's Name	String	No	No
RequireThing	Name of Matter	String	No	No
ResultName	Name of Property Right	String	No	No
ResultNum	Registration Number	String	No	No
OperName	Greffier	String	No	No
OperDate	Date of Registration	Date	No	No
ResultFileName	Certificate Filename	String	No	No

We do the following abstract to the business system:

The Transaction: Property Registration Event.

The Transaction Structure: {Property Registration Department, The Greffier, The DBA User, Applicant's Name, The Name of Property Right, The Registration Number, The Certificate Filename }.

Appendix: Certificate of Registration of Property Rights.

Transaction Users: Greffier , DBA.

Storage Properties: Local Storage, There is appendix (JPG file).

In the initialization phase:

Generating the user files through the authorization / license program, as follows:

0; regulator;……#

1;admin;…….#

2;zhangwei;……#

2;wangfang;……#

Generating the transaction model files through the transaction model custom program is as follows:

00; Registration of Property Rights; {! A City Property Right Management Center, OperName, !admin, RequireName, ResultName, RusultNum, ResultFileName}; 1; The Record of the Result of a Registration of Property Rights; SHA256 Value#

Among them, the "!" following is a constant.

Obtaining transaction structure data and corresponding operation command through bridging programs is as follows:

The Transaction Data:{ A City Property right Management Center, zhangwei, admin, liudawei, Housing Property Registration, A1800678, E:\FileData\A1800678.jpg}

The Transaction Command: INSERT INTO Table1 (ID, RequireName, RequireThing, ResultName, ResultNum, OperName, OperDate, ResultFileName) VALUES (35, "liudawei", "registration of property rights", "housing property registration certificate", "A1800678", "zhangwei", 20180101, "E:\FileData\A1800678.jpg")

The transaction process:

- Step 01. RNS digital signature received transaction data, as follows:
  { A City Property right Management Center, zhangwei, admin, liudawei, Housing Property Registration, A1800678, E:\FileData\A1800678.jpg}, r, s
  Among them, the r, s is the signature parameter calculated according to the private key and random number of the regulatory node.
- Step 02. RNS sends the signed data to TNS in which this transaction related users (zhangwei and admin) are located.
- Step 03. The zhangwei's TNS and the admin's TNS receives the data and performs the transaction validation.
- Step 03-01. Calculating the v according to the public key of the regulatory node. If v=r, then automatically verify through, enter the Step03-02. Otherwise, the verification result is "false" and enter the Step04.
- Step 03-02. Manual verification. Take out the transaction data and analyze it to:
  "zhangwei" has applied the "house property right registration" to liudawei's application. The certificate number is "A1800678", which corresponds to the certificate file (click to show the certificate pictures).
  Please confirm the authenticity of the event: [ ] true [ ] false
  Receiving the selection result, if "true" is selected, then the verification result is true. If "false" is selected, then the verification result is "false".
- Step 04. The related TNS signatures its own transaction validation results and sends it to RNS.
- Step 05. RNS receives the signature message of the related TNS, and declassified verification.
- Step 05-01. If the validation is passed, enter the Step06.
- Step 05-02. If the validation is not passed, enter the Step01.
- Step 06. RNS consensus decision:
- Step 06-01. If all the received is true, the consensus is judged to be "true" and enter the Step07.
- Step 06-02. If the incomplete is "true", the consensus is judged to be "false" and enter the Step01.
- Step 07. RNS signatures the result of the consensus decision and sends it to the related TNS.
- Step 08. The related TNS receives the signature message, decryption verification:
- Step 08-01. If the validation is passed, enter the Step09.
- Step 08-02. If the validation is not passed, enter the Step01.
- Step 09. The Related TNS block generation:
- Step 09-01. The related TNS requests the timestamp, the transaction sequence number, the chain-tail pointer to RNS. They sign the request information and send them to RNS.
- Step 09-02. RNS decrypts the request signature. If the verification fails, the return result is empty. Otherwise, calculate and retrieve the corresponding value and place it on the return result.
- Step 09-03. The return result value is signed and sent to the corresponding TNS.
- Step 10. The related TNS receives the signature message, decryption verification:
- Step 10-01. If the validation is passed, enter the Step11.
- Step 10-02. If the validation is not passed, enter the Step01.
- Step 11. The items are assigned according to the block structure. The HASH256 value of the block header is calculated.
- Step 12. The related TNS performs block storage:
- Step 12-01. Open the user file to get the storage location of the user's. Open the transaction model file and read the attribute value of the appendix in this transaction.

- Step 12-02. Under the specified subdirectory of the storage location, the HASH256 value of the block header is used as the block name, and the block data is stored. The appendix data is stored in the same name of the block in the other specified subdirectory of the storage location.
- Step 12-03. The “admin” user signs the block data and appendix file and send them to RNS. The “zhangwei” user signs the block header data and send it to RNS.
- Step 13. RNS receives the signature message, decryption verification:
- Step 13-01. If the validation is passed, enter the Step14.
- Step 13-02. If the validation is not passed, enter the Step01.
- Step 14. RNS performs comparison verification:
- Step 14-01. Regard to block data, the Merkle root is calculated and compared with the Merkle root in the block:
- Step 14-01-01. If equal, enter Step14-02.
- Step 14-01-02. If unequal, enter Step01.
- Step 14-02. For block data, the HASH256 value of the calculated block header is compared with the block filename and the appendix filename:
- Step 14-02-01. If equal, enter Step14-03.
- Step 14-02-02. If unequal, enter Step01.
- Step 14-03. The header data items in the block data are compared with the block header data items sent by other users:
- Step 14-03-01. If equal, enter Step15.
- Step 14-03-02. If unequal, enter Step01.
- Step 15. The result of returning the comparison validation is "true".
- Step 16. RNS performs block linking:
- Step 16-01. Open the user file to get the storage location. Under the specified subdirectory of the storage location, the HASH256 value of the block header is used as the block name, and the block data is stored. The appendix data is stored in the same name of the block in the other specified subdirectory of the storage location.
- Step 16-02. Open the end-cloud blockchain file and append the block filename to the file.
- Step 16-03. Open the chain-tail file and update the only record of the file with the block filename.
- Step 17. Enter Step01.

5. Comparison and analysis of the simulation results

5.1. Security Contrast

We have a security comparison between MTMCB and Bitcoin Transaction System (BTS):

- The private key, public key, and address generation: In BTS, the private key is a random number that satisfies certain conditions, and the public key and address are calculated separately for the private key. In MTMCB, these are the same as BTS.

- The data transmission between nodes: The data transmission between nodes involves data security in the transmission process. In BTS, the private key and the secp256k1 algorithm are used for digital signatures, and the public key and the secp256k1 algorithm are used to verify. The purpose of this is to check whether the data is tampered during the transmission process. In MTMCB, these are the same as BTS.

- The transaction verification:

  The verification method: In BTS, we use the public key and secp256k1 algorithm to verify. In MTMCB, in addition to the above verification, artificial verification is added so that all forged and tampered business data can be found in time. In addition, in the manual verification process, the display of the related appendix files makes the manual verification more accurate.

  The scope of validation: In BTS, all users participate in the verification. In MTMCB, only the users of relevant transactions (the all direct participant in the transaction, a transaction data vindicator) verify.

- The consensus decision: In BTS, the consensus mechanism such as PoW/PoS is adopted. In MTMCB, the adoption of "only all through, is the consensus" is the mechanism. It can be better adapted to the consensus among the limited

nodes.

• The block generation: In BTS, the "mining" mechanism is used to generate the block. In MTMCB, all verifiers of every transaction generate the block.

• The block storage:
The storage location: In BTS, the block data is stored locally; in MTMCB, the block data can be stored locally, and can also be stored in the cloud.
The stored content: In BTS, only block data is stored; in MTMCB, not only is block data stored, but also appendix file data is stored.

• Comparative validation: In BTS, there is no function. In MTMCB, this function is added. It aims to ensure that the blocks generated by the direct participants and the data maintainer of each transaction are completely consistent.

• The block linking: In BTS, there is no function. In MTMCB, there is a complete block data in the regulatory node. According to the forward pointer (Prev-block), it can form logical blockchain and save the filename of each block in the end-cloud.

• Prevent the data files from being tampered with: In BTS, there is no detailed record. In MTMCB, every record of every data file sets up the check field (the HASH256 value of the record). Once the data file is tampered with, it can be found when reading the file.

Table 4. Results of comparison analysis of safety

Comparison Contents	In BTS	In MTMCB
The private key, the public key, and the address generation	Calculation based on random number, Hash256, RIPEMD160 and so on	Same as BTS
The data transmission between nodes	Based on secp256k1 signature / verification	Same as BTS
The transaction verification	Automatic verification based on secp256k1	Same as BTS + Manual verification
Consensus decision	PoW/PoS	The principle of unanimous adoption
The block generation	All users	The stakeholder
The block storage	Local, only block data	Local + end-cloud, the block data + the appendix data
Comparative validation	No	Yes
The block linking	No	Yes
Prevent the data files from being tampered with	No	Yes

5.2. Comparison of the Audit Services

- The proof of event occurrence and the proof of the sequence of events: This service is provided in the both BTS and MTMCB.
- The discovery and reconstruction of the block being tampered with: In BTS, this service is not provided. In MTMCB, this service is provided.
- Real time early-warning of abnormal transactions: In BTS, this service is not provided. In MTMCB, this service is provided.
- Clues to data reconstruction of abnormal transactions: In BTS, this service is not provided. In MTMCB, this service is provided.

Table 5. Results of comparison analysis of the audit services

Comparison Contents	In BTS	In MTMCB
The proof of event occurrence and the proof of the sequence of events	Implementation based on blockchain retrieval	Same as BTS
The discovery and reconstruction of the block being tampered with	No	Yes
Real time early-warning of the abnormal transactions	No	Yes
Clues to the data reconstruction of the abnormal transaction	No	Yes

5.3. Comparison of Blockchain Managements

- The Block generation efficiency: In BTS, one block is produced in about 10 minutes. In MTMCB, the efficiency of

block generation depends on the efficiency of business system event processing and the processing efficiency of RNS and TNS.

- The blockchain retrieval efficiency: In BTS, we can achieve the retrieval from the chain-tail to the Genesis Block through the interface. However, at present, the number of accumulated blocks is about 525600, and the time spent retrieving is large. In MTMCB, the overhead time of retrieving blockchain is very small.
- The blockchain form: In BTS, the blockchain bifurcations may be caused by the block generation mechanisms. In MTMCB, the blockchain must be "linear", but it may appear as the "broken chain" phenomenon (because tampering blockchain is not successful). The reconstruction method of "broken chain" is introduced in the next article.
- Prevent block data from being tampered with: In BTS, the blocks with chain-ages of more than 6 cannot be tampered with by using power consumption and consensus mechanism to prevent tampering. In MTMCB, the probability of a blockchain being successfully destroyed is P_s (see formula (1)).

Table 6. Results of comparison analysis of the blockchain managements

Comparison Contents	In BTS	In MTMCB
The Block generation efficiency	A block is produced in about 10 minutes	Depends on the efficiency of business system event processing
The blockchain retrieval efficiency	A lot of time spending	The spending of time is very small
The blockchain form	Maybe forking	Linear
Prevent block data from being tampered with	By using power consumption and consensus mechanism	The game of the business busy degree and the tamper time

6. The probability of the blockchain being destroyed

In MTMCB, tampering the blocks of ordinary transaction user nodes and DBA user nodes does not have any impact on the blockchain. Only tampering of the blocks of the regulatory node can destroy the blockchain. Therefore, we describe the "the blockchain being destroyed successfully" as a way to tamper with one or some block data of the regulatory node, and successfully integrate it into the blockchain to achieve a spurious truth. "The probability of the blockchain being destroyed successfully" is marked as P_s.

Assume that the regulatory node is tampered with B_i, and its "chain-ages" is m: In MTMCB, the average time of producing a block is T_a. The process flow chart is displayed in Figure 28.

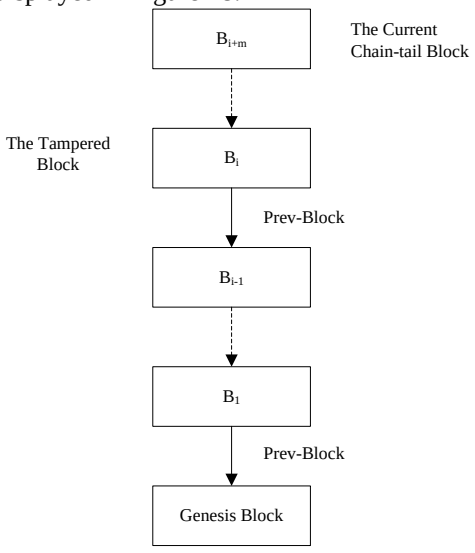

Figure 28. The schematic diagram of the “Chain-Ages”

The process and time to tamper with the B_i block:

- Step 1. Read the Perv-Block pointer of the B_i block, and the time spent as T_1.
- Step 2. Tamper with the B_i block data, calculate the Merkle Root, calculate the filename of the tampered block, and the time spent as T_2.
- Step 3. Tamper with all appendage filename prefixes associated with B_i, and the time spent as T_3.
- Step 4. Tamper with all the appendix files content related to B_i to match the tampered data, and the time spent as T_4.

- Step 5. Tamper with the corresponding record in the end-cloud blockchain files, and the time spent as T_5.

According to the P_s definition, if you want to tamper with success, you have to tamper B_i to B_{i+m} before a newly generated block is linked. From B_{i+1} to B_{i+m} blocks, except Step4, each of the other steps must be executed. Therefore,

$$P_s = P((\sum_i^{i+m}(T_1 + T_2 + T_3 + T_5) + T_4) < T_a) \quad (1)$$

In formula (1), Step4 is executed only once because the appendix file types are diverse (such as PDF, JPG, WMV, mpg, etc.). So, you have to manually modify them one by one, and therefore, T4 may be much larger than others. The biggest factor that affects T_a is the busy degree of business. Regarded as the "sparse" business, the time of illegal persons for forgery may be enough, and the probability of tampering success will increase. During the automatic generation of "meta-block" by RNS (it can be removed when the automatic retrieval blockchain), from time to time the block generation is busy, and so the T_a will be very small, which makes P_s very close to 0. This method will soon be in the next article in detail.

7. Conclusions

Based on the core of the existing blockchain technology, this paper constructed a Multi-Transaction Mode Consortium Blockchain (MTMCB) for the data resource protection of the business system. It has realized the diversification of transaction types, the diversification of user types, the diversification of storage types, the diversification of storage contents, and the optimization of transaction process. The example proves that MTMCB has the same security, as well as better and wider applicability compared with the Bitcoin transaction system. Furthermore, it can adapt to the application of data resource protection in business systems, greatly expanding the application field of blockchain technology. In the future, we will continue to study the "meta-block" methods of reducing P_s.

Acknowledgements

This work was supported in part by Talent Fund Project of Hefei University (No. 14RC08) as well as the Ministry of Science and Technology of P.R China Innovation Fund for Technology based Firms Projects (No. 11C26213401181).

References

1. A. G. Abbasi, Z. Khan, "VeidBlock: Verifiable Identity using Blockchain and Ledger in a Software Defined Network" in *Proceedings of the 10th International Conference on Utility and Cloud Computing* (UCC '17 Companion), pp.173-179, Austin, Texas, USA, December 2017.
2. A. Chakravorty, C. Rong, "Ushare: user controlled social media based on blockchain" in *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication* (IMCOM '17), pp.64-67, Beppu, Japan, January 2017.
3. M. Chanson, A. Bogner, F. Wortmann, and E. Fleisch, "Blockchain as a privacy enabler: an odometer fraud prevention system" in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers* (UbiComp '17), pp. 13-16, Maui, Hawaii, September 2017.
4. C. Decker, J. Seidel, and R. Wattenhofer, "Bitcoin meets strong consistency" in *Proceedings of the 17th International Conference on Distributed Computing and Networking* (ICDCN '16), pp.76-80, Singapore, Singapore, January 2016.
5. T. Hardjono, N. Smith, "Cloud-Based Commissioning of Constrained Devices using Permissioned Blockchains" in *Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security* (IoTPTS '16), pp.29-36, Xi'an, China, May 2016.
6. B. Leiding, P. Memarmoshrefi, and D. Hogrefe, "Self-managed and blockchain-based vehicular ad-hoc networks" in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct* (UbiComp '16), pp. 137-140, Heidelberg, Germany, September 2016.
7. W. Li, A. Sforzin, S. Fedorov, and G. O. Karame, "Towards Scalable and Private Industrial Blockchains" in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts* (BCC '17), pp.9-14, Abu Dhabi, United Arab Emirates, April 2017.
8. X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, "ProvChain: A Blockchain-based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability" in *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing* (CCGrid '17), pp.468-477, Madrid, Spain, May 2017.
9. H. L. Nijeholt, J. Oudejans, and Z. Erkin, "DecReg: A Framework for Preventing Double-Financing using Blockchain Technology" in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts* (BCC '17), pp.29-34, Abu Dhabi, United Arab Emirates, April 2017.
10. R. Neisse, G. Steri, and I. Nai-Fovino, "A Blockchain-based Approach for Data Accountability and Provenance Tracking" in *Proceedings of the 12th International Conference on Availability, Reliability and Security* (ARES '17), pp.362-365, Reggio Calabria, Italy, August 2017.
11. K. R. Özyılmaz, A. Yurdakul, "Integrating low-power IoT devices to a blockchain-based infrastructure: work-in-progress" in *Proceedings of the 13th ACM International Conference on Embedded Software 2017 Companion* (EMSOFT '17), pp.244-248, Seoul, Republic of Korea, October 2017.

12. S. Raju, V. Rajesh, and J. S. Deogun, “The Case for a Data Bank: an Institution to Govern Healthcare and Education” in *Proceedings of the 10th International Conference on Theory and Practice of Electronic Governance* (ICEGOV '17), pp.538-539, New Delhi AA, India, March 2017.
13. D. Svetinovic, “Blockchain Engineering for the Internet of Things: Systems Security Perspective” in *Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security* (IoTPTS '17), pp.1-1, Abu Dhabi, United Arab Emirates, April 2017.
14. H. T. Vo, L. Mehedy, M. Mohania, and E. Abebe, “Blockchain-based Data Management and Analytics for Micro-insurance Applications” in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (CIKM '17), pp.2539-2542, Singapore, Singapore, November 2017.
15. L. Xu, L. Chen, Z. Gao, S. Xu, and W. Shi, “EPBC: Efficient Public Blockchain Client for lightweight users” in *Proceedings of the 1st Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers* (SERIAL '17), pp.28-31, Las Vegas, Nevada, December 2017.
16. Q. Xing, B. Wang, and X. Wang, “POSTER: BGPCoin: A Trustworthy Blockchain-based Resource Management Solution for BGP Security” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (CCS '17), pp.2591-2593, Dallas, Texas, USA, October 2017.
17. N. Zupan, K. Zhang, and H. A. Jacobsen, “Hyperpubsub: a decentralized, permissioned, publish/subscribe service using blockchains: demo” in *Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference: Posters and Demos* (Middleware '17), pp. 15-16, Las Vegas, Nevada, December 2017.

Jiarui Zhang was born on Aug. 24, 1964**.** He is Master of computer technology. Currently, he is a professor level senior engineer at Hefei University, China. His major research interests include blockchain technology, information resource protection, and data sharing. He has published many papers in related journals. He has a number of patents of invention in recent years.