Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 19(12): 11935–11956.
DOI: 10.3934/mbe.2022556
Received: 24 June 2022
Revised: 01 August 2022
Accepted: 04 August 2022
Published: 17 August 2022

Research article

Lightweight blockchain fuzzy decision scheme through MQTT and Fibonacci for sustainable transport

Zhongxue Yang[1], **Yiqin Bao**[1,*], **Yuan Liu**[2], **Qiang Zhao**[3], **Hao Zheng**[1] **and Wenbin Xu**[4]

[1] School of information engineering, Nanjing XiaoZhuang University, Nanjing 211171, China
[2] Business School, Jinling University of Science and Technology, Nanjing 211199, China
[3] Department of Information Systems Schulich School of Business, Toronto 416647, Canada
[4] Nanjing Huazhu Industrial Intelligent Equipment Co., Ltd., Nanjing 211175, China

* **Correspondence:** Email: baoyiqin@njxzc.edu.cn; Tel: +8613851549080.

Abstract: The unprecedented progress in field of IoT enabled rapid developments in the vehicle intelligent transportation systems and most of these provide services in a centralized way. However, the centralized system architecture is vulnerable to the external attacks as a result both information and equipment are prone to eavesdropping and destruction. Therefore, there is a trend to apply blockchain technology to the vehicle intelligent transportation systems in order to achieve sustainable transportation. Nevertheless, the system is so great and very sophisticated and the ultimate task will be harder to implement. In view of this, an attempt is made in this paper to propose a lightweight fuzzy decision blockchain scheme through MQTT and Fibonacci, and through this scheme, the extent of blockchain server can be scaled and easy to deploy. Also through MQTT, reliable communication and transmission of blockchain can be realized. LF-BC is formed by using DH and Fibonacci transformation to enhance security, and F-PBFT consensus algorithm can reduce the communication overhead and improve the fault tolerance tremendously. Using LF-BC scheme, the experimental results show that the fault tolerance rate is significantly improved by 22.3%, and the sustainable safety and reliability of the vehicle intelligent transportation system is increased consumedly. At the same time, the feasibility of the scheme is also verified by taking specific cases.

Keywords: blockchain; MQTT; PBFT; LF-BC; DH; Fibonacci; sensor networks

1. Introduction

As rapid development of internet of things (IoTs), a large number of devices in the IoTs and the volume of data generated make it further messy and huge. Further, there are high security risks while establishing an ecosystem containing all devices of the IoTs [1]. The blockchain is a decentralized ledger based on the consensus mechanism, cryptography and distributed storage in the intelligent transportation system. A perfect protocol and architecture design are essential to build a blockchain secure and reliable storage and trading network. In the field of IoT, blockchain can be used to provide security and privacy for user data in the network [2]. In the field of big data, blockchain can be used to improve data security and data quality [3]. In the field of healthcare domain Agbo et al. [4] made blockchain technology in healthcare: A systematic review. In the field of agri-food domain, Mirabelli [5] proposed a blockchain-based solution for agri-food supply chains. In the field of energy domain, Andoni et al. [6] proposed Blockchain technology in the energy sector: A systematic review of challenges and opportunities. In the field of Construction industry domain, Plevris et al. [7] proposed. Blockchain in civil engineering, architecture and construction industry: State of the art, evolution, challenges and opportunities. Bakogiannis et al. [8] proposed leveraging blockchain technology to break the cloud computing market monopoly in the field of cloud computing. In order to obtain data security, blockchain is also used in the field of fog computing and edge computing [9]. Khan et al. [10] pointed out that the scalability problem in the public blockchain hinders the provision of the best solution for enterprises and industries. Key management, encryption algorithm and smart contract are the core features of blockchain security [11]. Blockchain applications pay more attention to the core blockchain technologies and jointly provide a solid security cornerstone for blockchain applications from multiple dimensions such as security, data security, management security and application security [12], Bhat et al. [13] designed the agricultural food supply chain management system based on blockchain and IoT. Huh et al. [14] studied the blockchain consensus algorithm for feasible management of new and renewable energy. Astarita et al. [15] reviewed the application of blockchain in transportation system.

Message queue telemetry transmission (MQTT) is one of the common communication protocols for any platform in the IoTs. Whether from the perspective of overall security, equipment compatibility and resource consumption, MQTT protocol has certain advantages and is the most competitive communication protocol in the current IoT. However, the user password login form adopted by MQTT protocol has some security problems [16]. Diffie-Hellman (DH) key exchange algorithm is used to share public key and private key pairing for encryption and decryption in a secured manner. This algorithm is named after its inventor Diffie and Hellman [17], DH is not an encryption algorithm, but a secure key exchange algorithm, which is commonly used in cloud computing and blockchain systems [18]. Fibonacci sequence, also known as golden section sequence, is mathematically defined by recursive method: $F(0) = 0, f(1) = 1, f(n) = f(n-1) + F(n-2)$ $(n \geq 2, n \in N)$ is used in many fields of physics and chemistry [19] and the third-order Fibonacci sequence is also used in the encryption and decryption applications [20].

Aiming at the problems of complex system and difficulty in implementation of the application of blockchain in intelligent transportation system, this paper proposes a lightweight fuzzy decision blockchain scheme through MQTT and DH Fibonacci technologies. Through this, the blockchain service scheme is scalable and easy to implement. DH and Fibonacci transformation are adopted to form lightweight fuzzy decision blockchain (LF-BC), which enhances the security and ensures the sustainable safety of the intelligent transportation system. The main contributions of this paper include three aspects:

- DH and Fibonacci transform are combined to form the fuzzy practical Byzantine fault tolerance (DH-FIB) algorithm which is used to generate LF-BC and to enhance the security.
- Realize scalable blockchain service scale and realize reliable communication of blockchain through MQTT protocol.
- In LF-BC blockchain system, using F-PBFT consensus algorithm, both fault tolerance rate and the transmission efficiency are significantly improved.

The paper is organized as follows. An extensive literature review on related technologies is undertaken in the second part. The third part proposes a lightweight fuzzy decision blockchain scheme. The fourth part compares and analyzes the experimental results. The fifth part summarizes the conclusion.

2. Related research

In this paper, we implement a trusted lightweight blockchain scheme based on MQTT and Fibonacci, discuss the recent existing research on MQTT protocol, Fibonacci matrix, DH algorithm, blockchain technology and propose to apply DH-FIB algorithm to blockchain scheme.

2.1. MQTT protocol

MQTT was originally developed by IBM as a communication protocol for telemedicine services. It is widely used in the IoTs because of its simplicity, specification, low overhead and easy implementation [21]. MQTT-publish based communication protocol, which creates and controls the hierarchy of topics, or subscribes to multiple topics. In March 2013, Oasis announced that MQTT was the preferred standard for the emerging IoTs messaging protocol. The model of MQTT protocol applied to the IoTs [22] is shown in Figure 1.

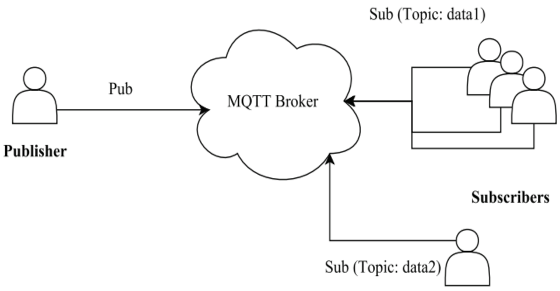

Figure 1. The data communication over MQTT protocol.

MQTT protocol frame format is shown in Table 1:

Table 1. MQTT frame format.

Control header	Packet length	Variable length header	Payload
1 Byte	1 to 4 Bytes	0–Y Bytes	0–X Bytes

The type of MQTT protocol package [23], is shown in Table 2.

MQTT protocol only specifies the message format and does not limit the load content and format of users. Therefore, we can customize the load format. There are three options for load format: 1) Hexadecimal/Binary: In this, there is no readability, but the flow can be controlled to relatively small extent; 2) String: This is easy to read, but is still not the best choice; 3) JSON-key value pair: The hierarchy is concise and clear, which is easy to read and write. While implementing the blockchain scheme, considering the security of confidentiality, hexadecimal format is adopted along with Modbus to implement the monitoring and control of the terminal.

Table 2. MQTT packets type.

Packets type	Type value	Details
CONNECT	0x01	Client request to connect to server
CONNACK	0x02	Connect acknowledgment
PUBLISH	0x03	Publish message
PUBACK	0x04	Publish acknowledgment
PUBREC	0x05	Publish received (assured delivery part 1)
PUBREL	0x06	Publish received (assured delivery part 2)
PUBCOMP	0x07	Publish received (assured delivery part 3)
SUBSCRIBE	0x08	Client subscribe request
UNSUBSCRIBE	0x0a	Unsubscribe request
PINGREQ	0x0c	PING request
DISCONNECT	0x0e	Client is disconnecting

MQTT protocol is simple, standardized and has certain advantages. It is the most competitive communication protocol in the current IoTs [24].

2.2. Fibonacci matrix

Fibonacci, an Italian mathematician in the 13th century, introduced an interesting sequence (called as Fibonacci sequence) from the problem of rabbit reproduction, record as *{Fn}*. It is a second-order recursive sequence, $F_0 = 0, F_1 = 1, F_{n+1} = F_n + F_{n-1}, n = 1,2,3\dots$, and we call each term of this sequence as Fibonacci number. Researchers are interested in Fibonacci sequence over years because of its unique properties and are widely using in computational mathematics, optimization theory, operations research and other fields [25,26].

Reference [27] defines Fibonacci matrix T_2, which corresponds to the second-order Fibonacci sequence *{Fn}* and is expressed as:

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = T_2^n \quad , \quad n = 1, 2, 3\dots \tag{1}$$

Reference [28] defines the third-order Fibonacci matrix T_3, which corresponds to the third-order Fibonacci sequence *{Fn}−a* third-order recursive sequence $F_0 = 1, F_1 = 1, F_2 = 2, F_{n+1} = F_n + F_{n-1} + F_{n-2}, n = 2, 3, 4\dots$, and is expressed as:

$$\begin{pmatrix} F_n & F_{n+1}-F_n & F_{n-1} \\ F_{n-1} & F_n-F_{n-1} & F_{n-2} \\ F_{n-2} & F_{n-1}-F_{n-2} & F_{n-3} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^n = T_3^n \quad n = 3, 4, 5\ldots \tag{2}$$

,

Since the determinant $|T_3| = 1$ corresponding to the third-order Fibonacci matrix, T_3 is reversible.

$$T_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \qquad T_3^{-1} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & -1 \end{pmatrix} \tag{3}$$

Similarly, $|T_3^n| = 1$, so T_3^n is also reversible. The formulae for calculating T_3^n and $(T_3^n)^{-1}$ are as follows:

Since:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^{k1+k2} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^{k1} * \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^{k2} \tag{4}$$

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & -1 \end{pmatrix}^{k1+k2} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & -1 \end{pmatrix}^{k1} * \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & -1 \end{pmatrix}^{k2} \tag{5}$$

So:

$$T_3^n = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}^n \qquad (T_3^n)^{-1} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & -1 \end{pmatrix}^n \tag{6}$$

For example, $n = 5$, T_3^5 and $(T_3^5)^{-1}$ are calculated by formula (6):

$$T_3^5 = \begin{pmatrix} 13 & 11 & 7 \\ 7 & 6 & 4 \\ 4 & 3 & 2 \end{pmatrix} \qquad (T_3^5)^{-1} = \begin{pmatrix} 0 & -1 & 2 \\ 2 & -2 & -3 \\ -3 & 5 & 1 \end{pmatrix} \tag{7}$$

We make $F_n = T_3^n$, $F_n^{-1} = (T_3^n)^{-1}$, use the characteristics of the third-order Fibonacci matrix T_3^n reversible sum formula (6), take n as a parameter, and apply F_n and F_n^{-1} to the encryption process of LF-BC blockchain in this paper.

2.3. DH Encryption algorithm

Cryptography is the most important part of the blockchain system and the basic technology to realize the blockchain [29]. DH is an asymmetric encryption algorithm, which is commonly used in blockchain encryption schemes. It is also a public key algorithm [30]. Each communication party generates a private key and a public key, where the private key is not public. After the public key is publicly exchanged, the communication parties can calculate the shared key, and the shared key calculated by both parties is the same. Therefore, both parties can encrypt and decrypt the message [31,32] through the shared key, so as to realize the encrypted transmission of the message. DH negotiates a shared key through unreliable channels. In many industrial security protocols, such as SSL/TLS, SSH, Kerberos, ISO-9798-3, just fast key and PKI key distribution are widely used.

Alice sends the encrypted transmission process of plaintext *M* to Bob, as shown in Figure 2.

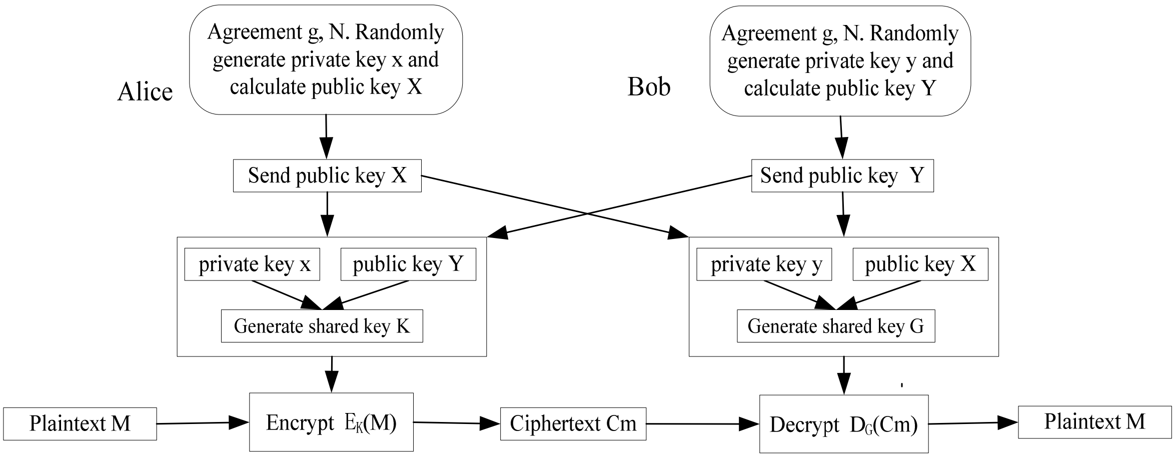

Figure 2. DH encryption process flow diagram.

Encryption process (Alice)

Step 1: First agree two parameters g, N with Bob;

Step2: Randomly generates own private key x, and calculate $X = g^x \, mod \, N$, X as the public key;

Step3: Select the public key $Y = g^y \, mod \, N$, is sent by Bob and calculate the key: $K = Y^x \, mod \, N = g^{xy} \, mod \, N$, K as the shared key with Bob;

Step 4: Calculate the encrypted data $Cm = E_{k1}(M)$;

Step5: Discloses sending Cm to Bob.

Decryption process (Bob)

Step1: First agree two parameters g, n with Alice;

Step2: Randomly generates own private key y, and calculates $Y = g^y \, mod \, N$, Y as the public key;

Step3: Select the public key $X = g^x \, mod \, N$ sent by the sender and calculate the key, $G = X^y \, mod \, N = g^{x\,y} mod \, N$, G as the shared key with Alice;

Step4: Publicly receives cm and decrypts data, $D_G(Cm) = D_G(E_K(M)) = M$;

Step5: Finally obtains m from the slave.

DH is the basis of authentication key exchange (AKE) scheme in the post quantum computing era [33]. It combines DH and Fibonacci transformation to form LF-BC blockchain, which is a specific application of DH in vehicle intelligent transportation system.

2.4. Blockchain technology

Blockchain technology was originally applied in the field of Internet finance to solve the trust problem of the third-party central institutions and provide a trust mechanism for decentralized transactions. The central idea of blockchain technology comes from the Bitcoin project released and implemented in 2009. The core technologies involved in the project mainly include cryptography, incentive mechanism, workload proof mechanism, P2P network, distributed database [34] and other technologies. It has four key characteristics.

- Decentralized network

The essence of blockchain is a decentralized distributed database, which is composed of a series of blocks generated by cryptography. The blocks contain data information that cannot be tampered within a certain period of time. Block contains data record, current block root hash, previous block root hash, time stamp and other information. Each user node in the network is both a client and a server. Each node has a complete backup of the whole database, and all nodes insert data through a specific consensus algorithm [35,36].

- Trusted blockchain structure

A blockchain consists of a block header and a block body as shown in Figure 3.

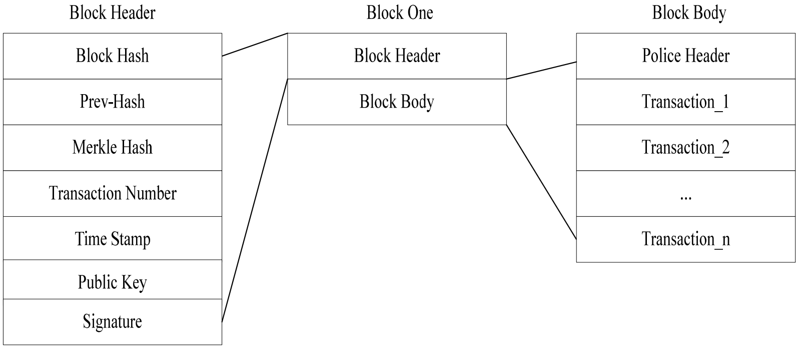

Figure 3. One block structure.

In Figure 3, block hash represents the hash value of the block header, pre-hash represents the hash value of the previous block, Merkel hash represents the Merkel hash value generated by the transaction number *Tx_id*, transaction number represents the number of transaction orders, time stamp represents the time stamp of the generated block, public key represents the public key of the generated block, and signature represents that the private key encrypts and signs of the user information. Policy header represents the permission control header of the block body [37], *Transaction_i (i = 1...n)* represents several transaction orders of the block body.

- Smart contract

Smart contracts are programmable protocols that allow the execution of the contract terms and agreements. The concept was first proposed by Szabo, who defined a smart contract as "a computerized transaction protocol that executes the terms of a contract". Thus, smart contracts can be deployed to a blockchain database and users can develop computer codes based on contractual clauses [38].

- Consensus algorithm

Practical byzantine fault tolerance (PBFT) algorithm is a consensus algorithm proposed by Miguel Castro and Arbara Liskov in 1999. PBFT is an algorithm to solve the consistency of state machine replicas in distributed scenarios [39]. The implementation of PBFT is divided into five steps: (1) Request, (2) Pre-prepare, (3) Prepare, (4) Commit, (5) Reply [40].

The process of PBFT algorithm is shown in Figure 4.

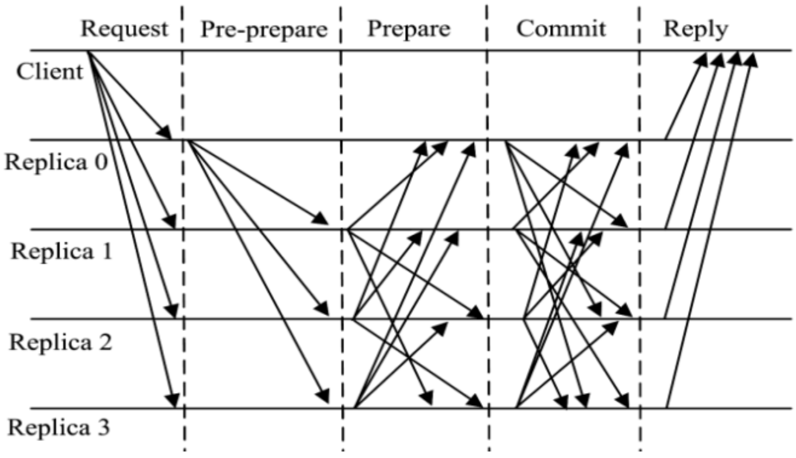

Figure 4. PBFT algorithm process.

PBFT is a state machine replication algorithm. State machines replicate in different nodes of the distributed system. The copy of each state machine not only saves the state of the service, but also realizes the operation of the service. For convenience of description, assuming that n is the total number of cluster nodes, the security and consistency of the consensus cluster can be guaranteed when the failed nodes do not exceed *(n−1)/3*. Therefore, the fault-tolerance rate of PBFT is:

$$F_1 = (n-1)/3 \tag{8}$$

In other words, as long as the nodes exceeding $(n-1)-F_1 = 2*(n-1)/3$ are honest, the validity of blockchain data can be guaranteed.

3. Lightweight fuzzy decision blockchain scheme

3.1. Lightweight fuzzy decision blockchain scheme architecture

The lightweight fuzzy decision blockchain scheme mainly includes four parts: client, terminal, blockchain system and cloud plan work. 1) The client is the user's operating software, which can be a web client or APP; 2) The terminal mainly realizes GPS positioning, signal processing and control, and then realizes communication with the cloud platform; 3) Blockchain system is a service system responsible for data security, which refers to MQTT broker; 4) Cloud platform is a network platform responsible for remote communication and data management. The system architecture is shown in Figure 5, the communication between client, terminal, blockchain system and cloud platform is carried out through the publish and subscription (Pub/Sub) topic of MQTT. there is implementation process in Section 3.5.

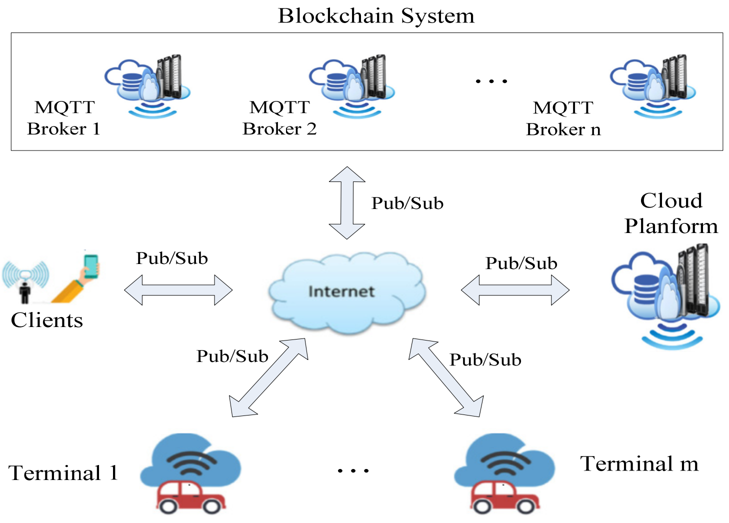

Figure 5. Lightweight fuzzy decision blockchain system architecture.

Based on MQTT and Fibonacci fuzzy decision blockchain scheme, due to the use of LF-BC blockchain in the architecture, the security problem of the traditional centralized architecture has been effectively solved. When the system is implemented, it mainly includes two transactions: 1) User registration and 2) Data transmission, as shown in Figure 6.

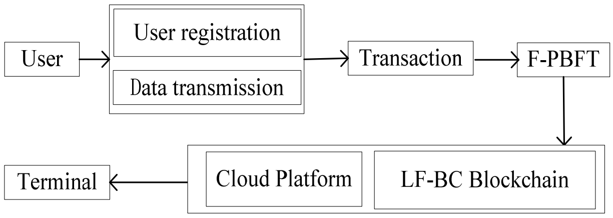

Figure 6. Data flow of LF-BC blockchain transactions.

In the user registration stage, the user inputs the user information to form a transaction order, and the registration information is encrypted and stored on the nodes of the blockchain through cryptographic technology and consensus algorithm F-PBFT; while in the data transmissionstage, the data message forms a transaction order. Through cryptography and consensus algorithm F-PBFT, the data message is stored in the blockchain and the information is transmitted to the terminal.

3.2. DH-FIB fuzzy decision encryption algorithm

The basic principle of generating DH-FIB algorithm from Fibonacci matrix to DH encryption algorithm is as follows: 1) The sender and the receiver generate their own private key vectors, such as $x = (x1,x2,x3)$, and generate their own public key vectors X and Y. 2) After X and y are exchanged, generate shared key vectors K and G, and calculate that the values of K and G are equal. 3) The sender's original m is encrypted into ciphertext Cm through K, and C is multiplied by Fibonacci matrix Fn to generate ciphertext vector $Em = C*Fn$, $Em = \{E1,E2,E3\}$, Em and sends it to the receiver. 4) The receiver multiplies the received em by the Fibonacci inverse matrix $Fn-1$, *Em*Fn−1* = *Cm*Fn*Fn−1* = *Cm*, finally decrypts *Cm* through the key vector *G* to obtain the original vector *Em= (M1,M2,M3*), and finally compares *M1*, *M2*, *M3* through the voting algorithm to obtain the original *M*.

DH-FIB is called fuzzy decision encryption algorithm. Since the encryption key is random and Fibonacci transformed, the ciphertext generated each time is different. For example, the ciphertext obtained by the middleman is ambiguous. Secondly, due to the use of key vector, it is necessary to make decision on the result to achieve the purpose of fault tolerance, reliability and sustainability.

The encryption process of user sending plaintext m to Terminal is shown in Figure 7.

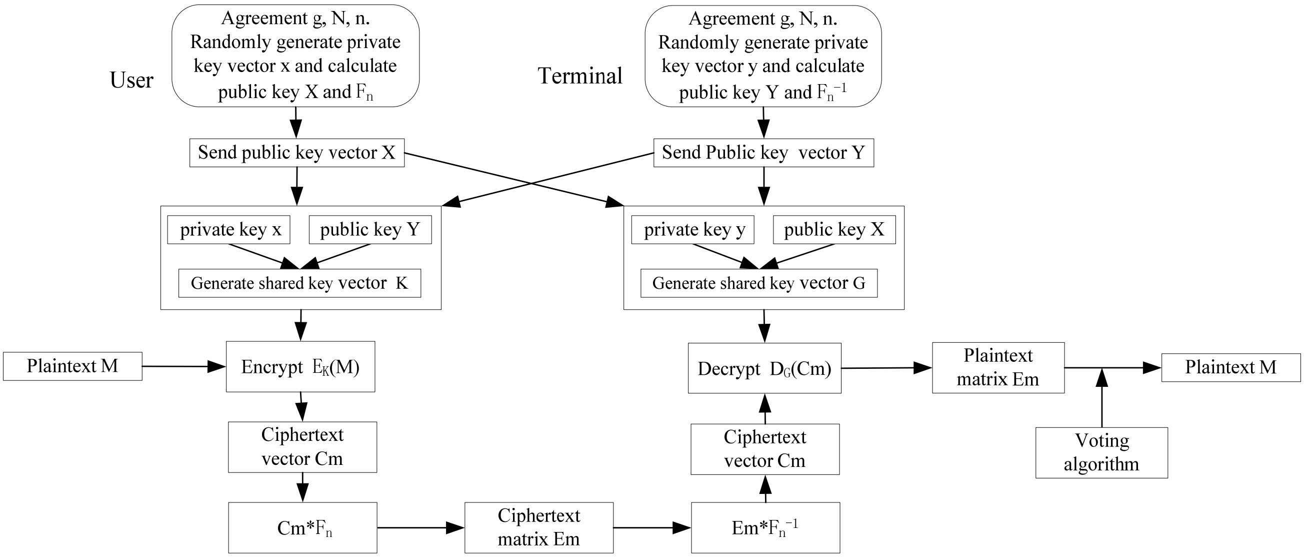

Figure 7. DH-FIB encryption process flow diagram.

- Encryption process (user)

Step 1: First agree three parameters *g, N, n of* F_n with terminal;

Step 2: Randomly generate private key vector *x = (x1,x2,x3)* and calculate public key vector: $X = (g^{x1} \bmod N, g^{x2} \bmod N, g^{x3} \bmod N)$;

Step3: Select terminal's public key vector $Y = (g^{y1} \bmod N, g^{y2} \bmod N, g^{y3} \bmod N)$ and calculate: $K = Y^x = (g^{x1*y1} \bmod N, g^{x2*y2} \bmod N, g^{x3*y3} \bmod N)$; *K* is defined as *(K1, K2, K3)*, and is used as the shared key with terminal;

Step 4: Calculate the encrypted data *C*, for example, the original file m has 5 bytes;

M = (M1,M2,M3,M4,M5) (even for *n* bytes, the principle remains the same), calculate encryption matrix: $Cm = E_k(M)$.

$$Cm = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \\ c_{41} & c_{42} & c_{43} \\ c_{51} & c_{52} & c_{53} \end{pmatrix} \tag{9}$$

Step 5: Select the third-order Fibonacci matrix F_n, see formula (6);

Step 6: Calculate the encryption matrix *Em* by formulas (9) and (6);

$$Em \;=\; Cm{*}F \;=\; \begin{pmatrix} E_{11} & E_{12} & E_{13} \\ E_{21} & E_{22} & E_{23} \\ E_{31} & E_{32} & E_{33} \\ E_{41} & E_{42} & E_{43} \\ E_{51} & E_{52} & E_{53} \end{pmatrix} \tag{10}$$

Step 7: Convert *Em* to encrypted vector;

$$Em \;=\; \{E1,E2,E3\} \tag{11}$$

Step 8: Publicly send *Em* to terminal.

Decryption process (terminal)

Step 1: First agree three parameters g, N, n of F_n with user;

Step 2: Randomly generate their own private key vector $y = (y1,y2,y3)$ and calculate public key vector: $Y = (g^{y1}\ mod\ N,\ g^{y2}\ mod\ N,\ g^{y3}\ mod\ N)$;

Step 3: Select user's public key vector $X = (g^{x1}\ mod\ N,\ g^{x2} mod\ N,\ g^{x3} mod\ N)$, and calculate: $G = X^y = -(g^{x1*y1} mod\ N,\ g^{x2*y2} mod\ N,\ g^{x3*y3} mod\ N)$, G is defined as $(G1,\ G2,\ G3)$, and is used as the shared key with user;

Step 4: Publicly accept T, $T= \{E1,E2,E3\}$;

Step 5: Select the third-order Fibonacci inverse matrixF_n^{-1}, see formula (6);

Step6*:* The decryption transformation matrix *Cm* is calculated by formulae (10) and (6). $Cm= E{*}F_n^{-1}\ mod\ N$;

Since:

$$F_n{*}F_n^{-1} \;=\; \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{12}$$

So:

$$Cm \;=\; \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \\ C_{41} & C_{42} & C_{43} \\ C_{51} & C_{52} & C_{53} \end{pmatrix} \tag{13}$$

Step 7: The decryption matrix is calculated by *G* and formula (13).

$$Mn = D_G(Cm) = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \\ M_{41} & M_{42} & M_{43} \\ M_{51} & M_{52} & M_{53} \end{pmatrix} \tag{14}$$

Set:

$$Mn = \{M1,\ M2,\ M3\} \tag{15}$$

From formula (14), *M1*, *M2*, *M3* should be equal;

Step 8: The original text is obtained from the decryption Matrix *Mn* through the voting algorithm.

3.3. LF-BC fuzzy decision blockchain and MQTT frame

The DH-FIB algorithm is applied to the blockchain system, and the generated blockchain is called lightweight fuzzy decision blockchain (LF-BC). Corresponding to a specific application, such as the remote monitoring system, the client in the system monitors the terminal through MQTT. We define the remote terminal as RTU, the control message adopts MQTT frame format, and the domain in MQTT frame format adopts DH-FIB blockchain structure, as shown in Figure 8. A blockchain includes a block header and a block body. The block body includes a policy header and several transaction orders. The transaction order is defined as a Modbus message.

The block header is used to ensure the security and integrity of blockchain. Police header is used for permission control and is the basis of blockchain data structure. The authorizer is a single entity user or multiple user groups [41].

LF-BC blockchain has three main functions:

- Authority management

Authority management completes permission control through police control header, which includes record number (No), authorizer, *RTU_id*, active and state. RTU authority management function is a very important part of LF-BC blockchain system, which is directly related to the security of application system.

- Storage management

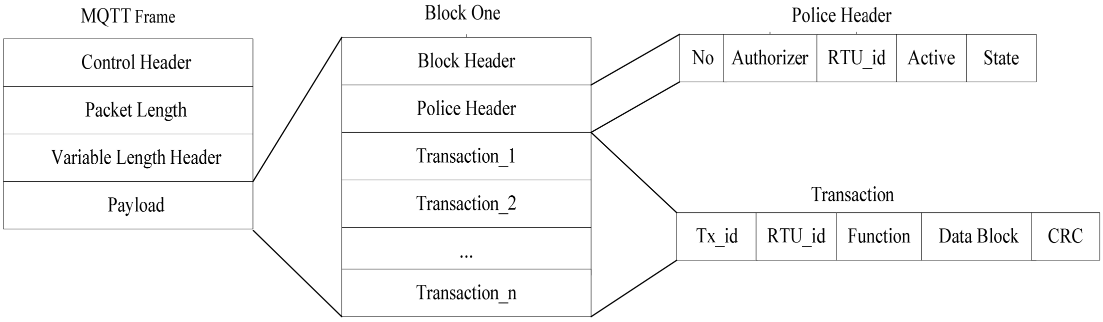

Figure 8. LF-BC blockchain and MQTT frame format.

Storage management is mainly used to manage the transactions. The transaction includes: transaction number (*Tx_id*) and Modbus message. Modbus protocol is the basic protocol commonly used in industrial automation system [42], which is also widely used in intelligent transportation system. The Modbus format is shown in Table 3.

Table 3. Modbus protocol format.

RTU_id	Function	Data_Adress	DATA	CRC

- DH-FIB encryption and fault tolerance management

LF-BC blockchain adopts encryption fault-tolerant management for transactions. An example used in this paper is the remote monitoring system. Communication messages need to be encrypted to prevent man in the middle attack and eavesdropping.

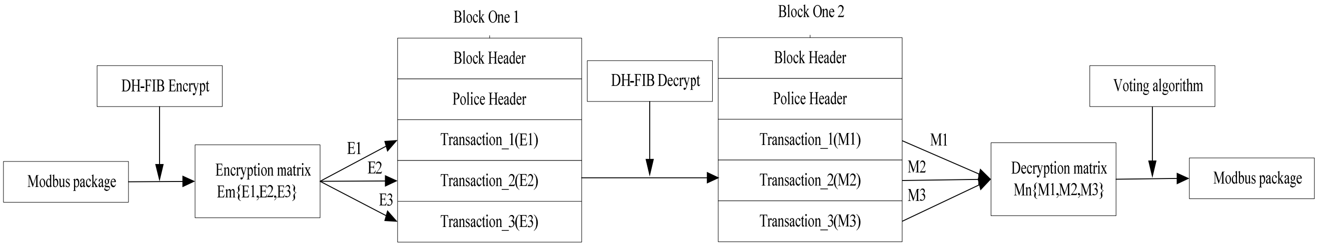

Figure 9. Encryption and decryption process of Modbus message transaction.

As shown in Figure 9, Modbus messages generate encryption vectors *Em = {E1,E2,E3}* through DH-FIB encryption algorithm. See formula (11), *E1*, *E2* and *E3* is used as three transactions of the blockchain respectively. After receiving the blockchain, the receiver decrypts it through DH-FIB algorithm to form three new transactions *Mn = {M1,M2,M3}*, as shown in formula (14), and finally restores it to Modbus message through fault-tolerant voting algorithm.

3.4. *F-PBFT consensus algorithm*

Ordinary blockchain schemes use PBFT consensus algorithm. A complete PBFT consensus needs to be completed twice communication, the complexity is *O(n2)*, and its fault tolerance rate is *F1 = (n−1)/3*, see formula (9). This paper defines the consensus algorithm of the LF-BC blockchain scheme as F-PBFT . Through DH-FIB encryption algorithm, a complete message forms a consensus node. The consensus node contains three transaction orders *Mn = {M1,M2,m3}*. See formula (15). Take *{M1,M2,m3}* as a group with a fault tolerance of *F2 = 1/3*. Vote in the group to reach consensus. Therefore, A complete F-PBFT consensus needs to be completed twicecommunication, the complexity is *O(n2/3)*, which greatly reduces the complexity of the communication.

Since one message generates three transaction orders and PBFT voting algorithm is adopted at the same time, the fault tolerance rate of F-PBFT is:

$$F3 = (n-1)-(1-F1)*(1-F2) = (n-1) - (1-1/3)*(1-(n-1)/3) = 5*(n-1)/9 \tag{16}$$

3.5. Implementation of MQTT communication

The composition of the system is shown in Figure 5, and the subscription topic definition of its components is shown in Table 4:

Table 4. Definition of subscrip topic.

Component	Subscrip topic
MQTT-1, MQTT-2, ..., MQTT-n	MQTT_topic
Cloud Platform	Cloud_topic
Client	Client_topic
Terminal	Terminal_topic

MQTT communication process:

The process that the client sends data to the terminal through MQTT is as follows:

- The client sends information to the cloud platform by publish topic (Cloud_topic);
- The cloud platform receives the information and sends information to MQTT-1, MQTT-2, ..., MQTT-n by publish topic (MQTT_topic);
- After receiving the information, MQTT-1, MQTT-2, ..., MQTT-n performs the consensus algorithm and sends information to the cloud platform by publish topic (Cloud_topic);
- The cloud platform receives three groups of information. The cloud platform sends a group of information to the terminal through the voting algorithm by publish topic (Terminal_topic).

3.6. Implementation Process of LF-BC blockchain scheme

The implementation process of LF-BC blockchain security scheme realizes the security of communication messages through the system without leakage. It mainly involves: encryption method, MQTT transmission, consensus algorithm, blockchain authentication, etc., mainly including: 1) User registration process and 2) data transmission process.

- User registration process

The flow chart of user registration (registered user name and password) in the system is shown in Figure 10.

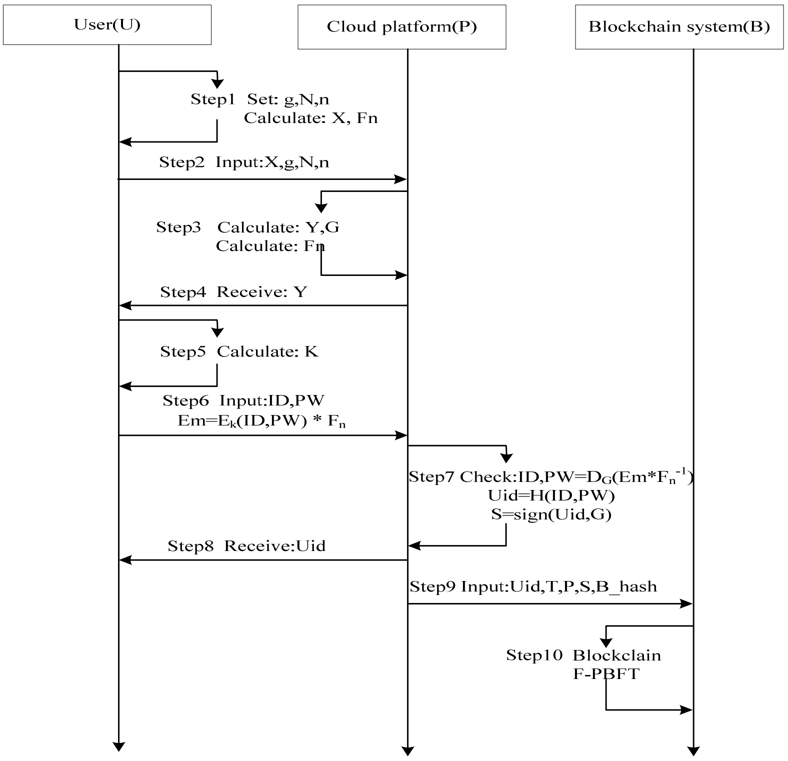

Figure 10. User registration flow chart of user.

Step 1: User sets three parameters *g, N, n of* F_n, randomly generates $x = (x1,x2,x3)$, and calculates X and F_n. $X = (g^{x1} \mod N, g^{x2} \mod N, g^{x3} \mod N)$;

Step 2: User sends *g, N, n, X* to cloud platform; *U=>P:{g, N, n, X}*;

Step 3: Cloud platform generates $y = (y1, y2, y3)$ and calculates *Y and G*; $Y = (g^{y1} mod N, g^{y2} mod N, g^{y3} mod N)$; $G = X^y=(g^{x1*y1} mod N, g^{x2*y2} mod N, g^{x3*y3} mod N)$;

Step 4: Cloud platform sends *Y* to user; *P=>U:{Y}*;

Step 5: User calculates K; $K= Y^x=(g^{x1*y1} mod N, g^{x2*y2} mod N , g^{x3*y3} mod N)$;

Step 6: User inputs *ID* and *PW*, calculates: $Em= E_k (ID, PW)* F_n$; *U=> P:{ID, Em}*;

Step 7: Cloud platform decrypts *Em*, calculates:

a) $ID, PW=D_G(Em*\underline{F_n}^{-1})$;

b) Judge whether the ID and PW are equal to those in the database;

c) Calculate user id: *Uid=H(ID, PW)*;

d) Calculate digital signature: *S=sign(id, G).*

Step 8: Cloud platform sends *Uid* to user; *P=>U:{Uid}*;

Step 9: Cloud platform sends *Uid*, timestamp *T*, public key; *P*, *S* and block*B_hash*to blockchain system; *P=>B:{Uid, T, Y, P, B_hash}*;

Step 10: Blockchain system forms a blockchain and vote the transaction through the F-PBFT consensus algorithm and saves it to each blockchain node.

• Data transmission process

The security of data transmission is the most critical part of LF-BC blockchain scheme, which realizes the encryption and integrity of data. The data transmission flow chart is shown in Figure 11.

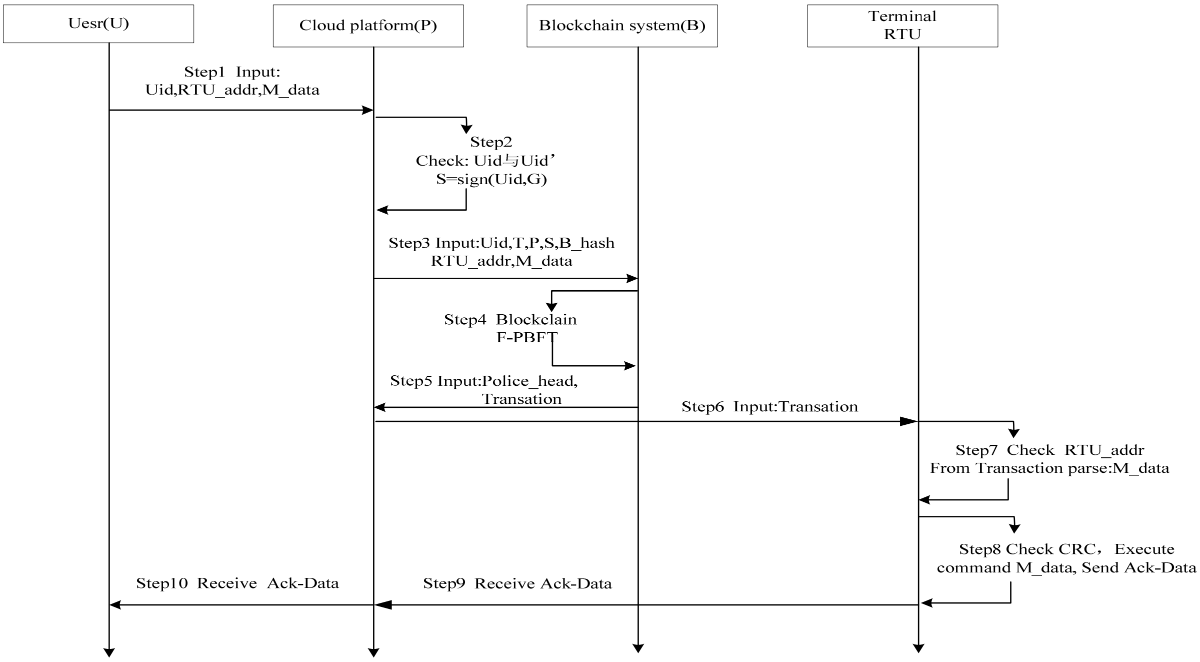
Uesr(U)
Cloud platform(P)
Blockchain system(B)
Terminal RTU
Step1 Input: Uid,RTU_addr,M_data
Step2 Check: Uid与Uid’ S=sign(Uid,G)
Step3 Input:Uid,T,P,S,B_hash RTU_addr,M_data
Step4 Blockclain F-PBFT
Step5 Input:Police_head, Transation
Step6 Input:Transation
Step7 Check RTU_addr From Transaction parse:M_data
Step8 Check CRC，Execute command M_data, Send Ack-Data
Step10 Receive Ack-Data
Step9 Receive Ack-Data

Figure 11. Blockchain data transmission flow chart of user.

Step 1: User input *Uid*, *RTU_addr*, Modbus packet M_data; U=>P:{Uid, RTU_addr, M_data};

Step 2: Cloud platform checks whether the *Uid* is equal to the *Uid* of the database, and makes digital signature: *S=sign(Uaddr, G)*;

Step 3: Cloud platform sends the address *Uid*, *T*, *P*, *S* and block*B_hash* to blockchain system; *P=>B:{Uid, T, P, S, B_hash}*;

Step 4: Blockchain system forms ablockclaim transaction and vote the transaction through the F-PBFT consensus algorithm, saves it to each block node;

Step 5: Blockchain system sends Police_head and transaction to cloud platform; B=>P:{Police_head, Transaction}

Step 6: Cloud platform sends Transaction to RTU; *P=>RTU:{Transaction}*;

Step 7: RTU checks whether the *RTU_addr* is consistent with its own, parse packet *M_data* from the transaction;

Step 8: RTU checks whether the CRC is correct and executes Modbus command *P_data*;

Step 9: RTU sends Ack-data to Cloud platform; *RTU=>P:{Ack-data}*;

Step 10: Cloud platform sends Ack-data to user. *P=>U:{Ack-data}*.

4. Testing and comparison

4.1. Case realization

This section takes the vehicle intelligent transportation system as an example to realize the specific application. First, establish a LF-BC blockchain through MQTT broker. Only authorized nodes can read blocks, execute smart contracts and verify new blocks in this blockchain. Secondly, the pre approved nodes agree and verify the blockchain through F-PBFT. The authorized nodes include data load terminal, cloud platform, MQTT broker and client.

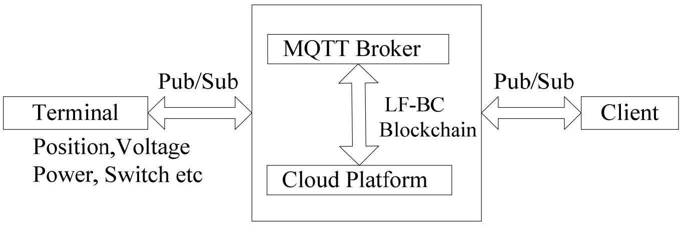

Figure 12. Application block diagram of intelligent transportation system.

The application block diagram of vehicle intelligent transportation system is shown in Figure 12. After the load terminal completes the monitoring, it encapsulates the monitoring data to be uploaded (such as equipment type, position, voltage, power, switch, monitoring time, etc.), then digitally signs these data in blockchain, and sends the signed data to cloud platform and blockchain system through MQTT. Finally, The client remotely monitors and controls the device through MQTT.

The following two sections emphasize the performance index analysis to illustrate the implementation effect of the system and verify the feasibility of this scheme.

4.2. Safety performance analysis

The lightweight blockchain scheme suggested in this paper is significantly different from the existing centralized monitoring system and analyzes the security of the blockchain scheme from the following three aspects.

- Decentralization: The decentralized network architecture is adopted in which the transaction information is verified by all nodes, and finally a trusted blockchain is formed. Even if some nodes fail or are attacked by hackers, the failed nodes can still return to normal. The centralized system architecture however, is vulnerable to attacks (such as DDoS attacks, SQL injection attacks, etc.).
- Non tamperability: Digital signature can resist forgery attack under random key, so it can not be easily tampered. The fault tolerance performance is improved and the success rate of tampering and modification will be very low in the adoption of DH-FIB encryption algorithm and F-PBFT consensu.
- Traceability: The consensus node will verify the authenticity of the transaction. Each F-PBFT block has the hash value of the previous block, which can be traced back to the source Modbus message. When a malicious node deletes or modifies the content of a block, the system can easily restore and trace all transaction data.

4.3. Fault-tolerance performance analysis

The collection was verified using three methods: the first was manual checking (MAC); the second one was common checking system (CMC)-referred as reading data through a single meter reading software and then checking data through a single image recognition software (such as Baidu cloud recognition) when the data could not be read; and the third one was the Petri parallel system, which was a parallel system of data acquisition and image recognition based on PN.

The higher the fault tolerance rate of the system, the higher is the transmission efficiency and the lower the fault tolerance rate, the lower is the transmission efficiency. The fault tolerance rates of the three schemes are compared below:

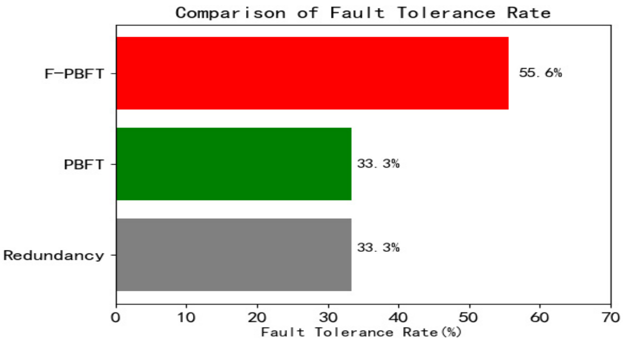

Figure 13. Comparison of fault tolerance of three schemes.

- Fault tolerance rate of the data redundancy: Centralized architecture often collects data. For example, a message is transmitted redundantly for three times, and the minority obeys the majority voting is adopted, so the fault tolerance rate is *1/3*.
- PBFT fault tolerance rate: The common blockchain architecture adopts that if malicious nodes are evenly distributed on each blockchain, when the number of nodes does not exceed *1/3* of the total number of nodes, the consensus result is correct and has no impact on the system output result. The fault tolerance ability of the system is $(n-1)/3$, so the fault tolerance rate is *1/3*.

- F-PBFT fault tolerance rate: The lightweight blockchain architecture in this paper adopts the F-PBFT consensus algorithm. One message generates three transaction orders, as shown in formula 16. The total fault tolerance is *5*(n−1)/9*. Therefore, the fault tolerance is 5/9.

On comparing the fault tolerance rates of the three schemes in the histogram in Figure 13, it is observed that the fault tolerance rate of F-PBFT has increased by 22.3%.

4.4. Data transmission time overhead comparison

Three architectures are compared by testing the data transmission time overhead on the load equipment monitoring system with wireless interference namely, 1) Centralized Architecture (CT-A), 2) Ordinary blockchain Architecture (BC-A); 3) Lightweight fuzzy decision blockchain Architecture (LF-BC-A). In the experiment, each Modbus message data contains 20 bytes, and the error code of one byte is randomly set in every three messages. The transmission time is calculated for the number of messages. B10 represents 10 messages, B20 represents 20 messages, and so on. B100 represents 100 messages. Communication parameters: 9600: 8: n: 1 (rate 9600 bps, 8-bit data bits, no parity check, 1-bit stop bit). Test under the same environment. The results are shown in Table 5.

Table 5. Time cost of transmitting encrypted data messages (s).

Architecture	B10	B20	B30	B40	B50	B60	B70	B80	B90	B100
CT-A	1.2	2.2	3.5	4.5	5.6	7.2	9.0	10.0	12.1	14.2
BC-A	1.1	2.5	3.8	4.8	5.9	8.0	9.5	11.2	13.4	15.0
LF-BC-A	1.1	2.0	3.1	4.2	5.1	6.3	7.2	8.3	9.1	10.5

The comparison curve based on Table 5 is shown in Figure 14, which intuitively reflects the time overhead of encrypted data transmission. X-axis represents the number of messages to be transmitted, and y-axis represents the time of message transmission.

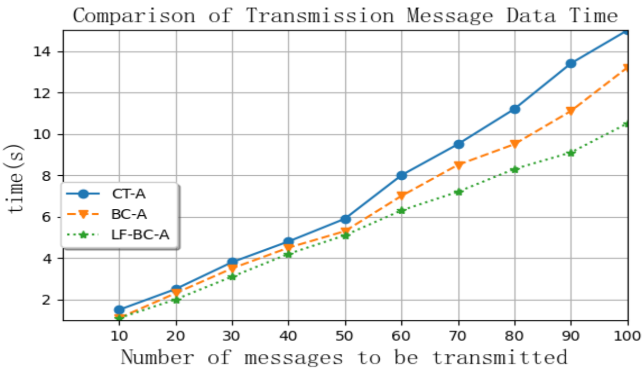

Figure 14. Time cost curve of transmitting encrypted messages.

Following observations are made from Figure 14. 1) In the centralized architecture, when the message is found to have error code (the error code can be judged by the check code in the message), the message needs to be retransmitted, so the transmission time of encrypted data is lower than that of the ordinary blockchain and lightweight blockchain in this paper; 2) On the other hand, in case of the PBFT fault tolerance algorithm, the ordinary blockchain architecture is more efficient than the centralized architecture; 3) whereas, in the F-PBFT fault tolerance algorithm, the LF-BC blockchain architecture has lower data transmission time overhead and the highest efficiency. Both theory and experiment prove that the LF-BC blockchain scheme in this paper has high transmission efficiency in the case of wireless interference.

4.5. Performance index comparison

This sub-section compares the performance of the three architectures by testing on the vehicle intelligent transportation system, combined with six performance indexes: 1) Data flow, 2) Complexity, 3) Fault tolerance, 4) Security, 5) Reliability, 6) Maintainability. The analysis results are shown in Table 6. The inferences are as follows:

Table 6. Comparison of three architecture performance indexes.

Performance index	Centralized architecture (CT-A)	Blockchain architecture (BC-A)	Lightweight blockchain architecture (LF-BC-A)
Data flow	Small	Big	Medium
Complexity	Small	Big	Medium
Fault tolerance	Bad	Good	Best
Security	Bad	Good	Good
Reliability	Bad	Good	Good
Maintainability	Easy	Not easy	Easy

- Compared with the ordinary blockchain architecture, the LF-BC blockchain architecture in this paper has smaller scale, easier implementation and less complexity, but the data flow is obviously larger than the centralized architecture.
- Further due to the addition of Fibonacci transform on the basis of HD, its security and encryption performance are better.
- The above analysis infers that the fault tolerance and reliability of this scheme have significantly improved.
- The scheme of this paper is lightweight blockchain, which has small scale and is easier to maintain.

Based on the above characteristics, the LF-BC blockchain architecture has certain scalability and can be applied to similar industries, such as agricultural food supply chain system, smart grid system, etc.

5. Conclusions and future work

This paper proposes a lightweight fuzzy decision blockchain LF-BC scheme based on MQTT and Fibonacci. Through this scheme, the major problem of complex and difficult deployment of blockchain is solved. The reliable communication of the blockchain is realized through MQTT, the LF-BC blockchain is formed by HD algorithm and Fibonacci matrix, which enhances both the security and the fault-tolerance rate by F-PBFT consensus algorithm. Moreover, from the realization of the case under consideration and its comparative analysis of the performance demonstrate the feasibility of this scheme and ensures the sustainable safety of the intelligent transportation system.

This blockchain LF-BC scheme is a lightweight scheme, which has not been verified in a large-scale system. It has only been applied in an urban on-board intelligent transportation system, and has certain limitations. It is envisaged in future research to expand the application of blockchain based on the combination of HD algorithm and Fibonacci matrix, which not only can be applied to the blockchain scheme of intelligent transportation system, but also useful in other aspects.

Acknowledgments

This study was supported by the Natural Science Foundation Project of China (61976118), the Natural Science Foundation Project of Jiangsu Province (BK20180142), and the key topics of the "13th five-year plan" for Education Science in Jiangsu Province (B-b /2020/01/18).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. Z. Wang, Y. Zhou, C. D. Huang, Q. Q. Miao, Survey on blockchain solution for big data, *Comput. Sci.*, **46** (2019), 6–10.
2. K. Salah, M. A. Khan, IoT security: Review, blockchain solutions, and open challenges, *Future Gener. Comput. Syst.*, **82** (2017), 395–411. https://doi.org/10.1016/j.future.2017.11.022
3. A. Azaria, A. Ekblaw, T. Vieira, A. Lippman, MedRec: Using blockchain for medical data access and permission management, in *2016 2nd international conference on open and big data (OBD)*, **8** (2016), 25–30. https://doi.org/10.1109/OBD.2016.11
4. C. C. Agbo, Q. H. Mahmoud, J. M. Eklund, Blockchain technology in healthcare: A systematic review, *Healthcare*, **7** (2019), 56–65. https://doi.org/10.3390/healthcare7020056
5. C. Mirabelli, Blockchain-based solutions for agri-food supply chains: A survey, *Int. J. Simul. Process Model.*, **17** (2021), 1–15. https://doi.org/10.1504/IJSPM.2021.120838
6. M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, et al., Blockchain technology in the energy sector: A systematic review of challenges and opportunities, *Renewable Sustainable Energy Rev.*, **100** (2019), 143–174. https://doi.org/10.1016/j.rser.2018.10.014
7. V. Plevris, N. D. Lagaros, A. Zeytinci, Blockchain in civil engineering, architecture and construction industry: State of the art, evolution, challenges and opportunities, *Front. Built Environ*, **8** (2022). https://doi.org/10.3389/fbuil.2022.840303
8. T. Bakogiannis, I. Mytilinis, K. Doka, G. Goumas, Leveraging blockchain technology to break the cloud computing market monopoly, *Computers*, **9** (2020), 9–15. https://doi.org/10.3390/computers9010009
9. S. Tuli, R. Mahmud, S. Tuli, R. Buyya, FogBus: A blockchain-based lightweight framework for edge and fog computing, *J. Syst. Software*, **154** (2019), 22–36. https://doi.org/10.1016/j.jss.2019.04.050
10. D. Khan, L. T. Jung, M. A. Hashmani, Systematic literature review of challenges in blockchain scalability, *Appl. Sci.*, **11** (2021), 9372–9378. https://doi.org/10.3390/app11209372

11. Y. Zhu, G. Gan, D. Deng, F. Ji, A. Chen, Security research in key technologies of blockchain, *Inf. Secur. Res.*, **2** (2016), 1090–1097.
12. J. Wang, L. Li, Y. Yan, W. Zhao, Y. Xu, Security incidents and solutions of blockchain technology application, *Comput. Sci.*, **45** (2018), 352–355.
13. S. A. Bhat, N. F. Huang, I. B. Sofi, M. Sultan, Agriculture-food supply chain management based on blockchain and IoT: A narrative on enterprise blockchain interoperability, *Agriculture*, **12** (2022), 40–47. https://doi.org/10.3390/agriculture12010040
14. J. H. Huh, S. K. Kim, The blockchain consensus algorithm for viable management of new and renewable energies, *Sustainability*, **11** (2019), 3184–3193. https://doi.org/10.3390/su11113184
15. V. Astarita, V. P. Giofrè, G. Mirabelli, V. Solina, A review of blockchain-based systems in transportation, *Information*, **11** (2020), 21–28. https://doi.org/10.3390/info11010021
16. F. Chen, Y. Huo, K. Liu, W. Tang, J. Zhu, Z. Sui, A study on MQTT node selection, in *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*, **16** (2020), 622–623. https://doi.org/10.1109/MSN50589.2020.00101
17. A. Chaturvedi, N. Srivastava, V. Shukla, A Secure wireless communication protocol using Diffie-Hellman key exchange, *Int. J. Comput. Appl.*, **126** (2015), 126–132. https://doi.org/10.5120/ijca2015906060
18. J. Athena, V. Sumathy, Survey on public key cryptography scheme for securing data in cloud computing, *Circuits Syst.*, **8** (2017), 77–92. https://doi.org/10.4236/cs.2017.83005
19. J. L. Herrera, J. J. Bravo, C. A. Gómez, Curious generalized Fibonacci numbers, *Mathematics*, **9** (2021), 2588–2598. https://doi.org/10.3390/math9202588
20. M. Akbiyik, J. Alo, On third-order bronze Fibonacci numbers, *Mathematics*, **9** (2021), 2606–2612. https://doi.org/10.3390/math9202606
21. D. Eridani, E. D. Widianto, Performance of sensors monitoring system using raspberry Pi through MQTT protocol, in *2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, **2018** (2018), 587–590. https://doi.org/10.1109/ISRITI.2018.8864473
22. N. Tantitharanukul, K. Osathanunkul, K. Hantrakul, P. Pramokchon, P. Khoenkaw, MQTT-Topics management system for sharing of open data, in 2*017 International Conference on Digital Arts, Media and Technology (ICDAMT)*, **2017** (2017), 62–65. https://doi.org/10.1109/ICDAMT.2017.7904935
23. R. Bryce, T. Shaw, G. Srivastava, MQTT-G: A publish/subscribe protocol with geolocation, in *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, **41** (2018), 1–4. https://doi.org/10.1109/TSP.2018.8441479
24. F. Chen, Y. Huo, K. Liu, W. Tang, J. Zhu, Z. Sui, A study on MQTT node selection, in *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*, **16** (2020), 622–623. https://doi.org/10.1109/MSN50589.2020.00101
25. G. Y. Lee, S. H. Cho, The generalized pascal matrix via the generalized Fibonacci matrix and generalized pell matrix, *Korean Math. Soc.*, **45** (2018), 479–491. https://doi.org/10.4134/JKMS.2008.45.2.479
26. S. Chen, W. Zhu, The general term and property of the five order Fibonacci series, *Hainan Daxue Xuebao*, **2014** (2014), 241–244.
27. X. Xie, Discussion and application of Fibonacci matrix, *Sci. Technol. Inf.*, **24** (2008), 2–7.

28. L. Peng, Properties and applications of third-order Fibonacci sequence, *J. Putian Univ.*, **5** (2006), 5–8.
29. Z. Chen, Q. Li, Improved PBFT consensus mechanism based on K-medoids, *Comput. Sci.*, **46** (2019), 101–107.
30. F. Chao, Z. Quan, J. T. Chao, A reliable Diffie-Hellman key exchange protocol automatic proof, *J. Commun.*, **2011** (2011), 119–123.
31. S. Pohlig, M. Hellman, An improved algorithm for computing logarithm over GF(p) and its Cryptographic significance, *IEEE Trans. Inf. Theory*, **1998** (1998), 458–471. https://doi.org/10.1109/TIT.1978.1055817
32. P. Kocher, Timing attacks on implementations of Diffie-Hellman, RSA DSS and other systems, *Adv. Cryptology*, **1996** (1996), 104–113. https://doi.org/10.1007/3-540-68697-5_9
33. M. Chen, A composable authentication key exchange scheme with post-quantum forward secrecy, *J. Comput. Res. Dev.*, **57** (2020), 2158–2176.
34. S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, 2008. Available from: https://bitcoin.org/bitcoin.pdf.
35. Y. Yuan, F. Wang, Development status and prospect of blockchain technology, *J. Autom.*, **4** (2016), 481–494.
36. A. E. Aadroul, Y. Manevich, Hyperledger fabric: a distributed operating system for permissioned blockchains, in *Proceedings of the thirteenth EuroSys conference*, (2018), 1–15. https://doi.org/10.1145/3190508.3190538
37. A. Dorri, S. Kanhere, R. Jurdak, Blockchain for IoT security and privacy: The case study of smart home, in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops*, (2017), 5–17. https://doi.org/10.1109/PERCOMW.2017.7917634
38. D. Magazzeni, P. McBurney, W. Nash, Validation and verification of smart contracts: Are search agenda, *Computer*, **9** (2017), 50–57. https://doi.org/10.1109/MC.2017.3571045
39. M. Castro, B. Liskov, Practical Byzantine fault tolerance and proactive recovery, *ACM Trans. Comput. Sys.*, **20** (2002), 398–161. https://doi.org/10.1145/571637.571640
40. W. Zhou, M. Long, Secure transmission scheme of environmental monitoring data based on blockchain, *Comput. Sci.*, **47** (2020), 315–320.
41. J. Fan, Y. Li, W. Wu, Y. Feng, Base station dynamic ring information monitoring system based on dual blockchain, *Comput. Sci.*, **46** (2019), 155–164.
42. S. Figueroa-Lorenzo, J. Añorga Benito, S. Arrizabalaga, Modbus access control system based on SSI over hyperledger Fabric blockchain, *Sensors*, **21** (2021), 5438. https://doi.org/10.3390/s21165438