IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 October 2022, accepted 5 November 2022, date of publication 9 November 2022, date of current version 18 November 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3221109*

RESEARCH ARTICLE

# Research on Data Security Model of Environmental Monitoring Based on Blockchain

**MANYU ZHAO[1], WEI LIU[1,2], AND KAI HE[3]**

[1]College of Information and Electronic Engineering, Shandong Technology and Business University, Yantai 264005, China
[2]Key Laboratory of Sensing Technology and Control, University of Shandong, Yantai 264005, China
[3]College of Computer Science and Technology, Shandong Technology and Business University, Yantai 264005, China

Corresponding author: Kai He (hekai@sdtbu.edu.cn)

This work was supported in part by the Yantai Science and Technology Innovation Development Plan Project under Grant 2022XDRH015.

**ABSTRACT** With the rapid development of the social economy, the problem of environmental pollution has been widely concerned. The existing environmental monitoring system adopts a hierarchical centralized management structure, which has some problems, such as data silos and the risk of data falsification. Thus, this paper proposes an environmental monitoring data security model based on the blockchain, which uses the blockchain distributed storage mode to realize the secure sharing of monitoring data and curb the behavior of data forgery. A practical Byzantine fault tolerant mechanism based on credit grouping supervision is adopted to reduce the computation and communication overhead caused by data consensus. Through the cloud chain fusion technology, the encrypted monitoring data is stored on the cloud storage server, and the monitoring data credentials are stored on the blockchain to reduce the storage pressure. And AES(Advanced Encryption Standard) combined with the RSA (Rivest-Shamir-Adleman) encryption algorithm to ensure the security of data transmission and storage. Security analysis and experiments demonstrate that our proposed scheme achieves authenticity, integrity, and security for monitored data. In addition, it effectively reduces the computation, communication, and storage overhead of the block nodes.

**INDEX TERMS** Environmental monitoring, blockchain, data security, practical Byzantine fault tolerance mechanism, cloud chain fusion.

## I. INTRODUCTION

With the rapid development of the social economy, the problem of environmental pollution has been widely concerned. Ecological environment departments at all levels have built a large number of environmental monitoring systems to monitor the emission of pollutants such as waste gas [1], waste liquid [2], and solid waste [3]. Monitor pollution data in a timely and accurate way as well as analyze current environmental indicators all-round to provide evidence for environmental law enforcement personnel [4].

The existing environmental monitoring system adopts a hierarchical centralized management architecture, as shown in Fig.1. The field end of the environmental monitoring system collects, computes, analyzes, processes, and stores various pollutant emission data, sewage equipment status and parameters, and other information through data acquisition and transmission equipment, and transmits them to the monitoring center via a transmission network.

There are two main problems with the centralized service architecture.

1) Monitoring data are stored in regulatory departments at all levels, forming data silos [5], which cannot achieve secure data sharing, and emission data cannot be deeply mined and applied. In order to meet the requirements of regulatory departments at all levels, data acquisition equipment often need to transmit data to multiple monitoring platforms at the same time, which increases the operation and maintenance costs.

2) There is a risk of unreliable transmission and tampering with data at all levels of the monitoring system, and the monitoring data cannot be credibly traced. Some companies

The associate editor coordinating the review of this manuscript and approving it for publication was Nitin Gupta.

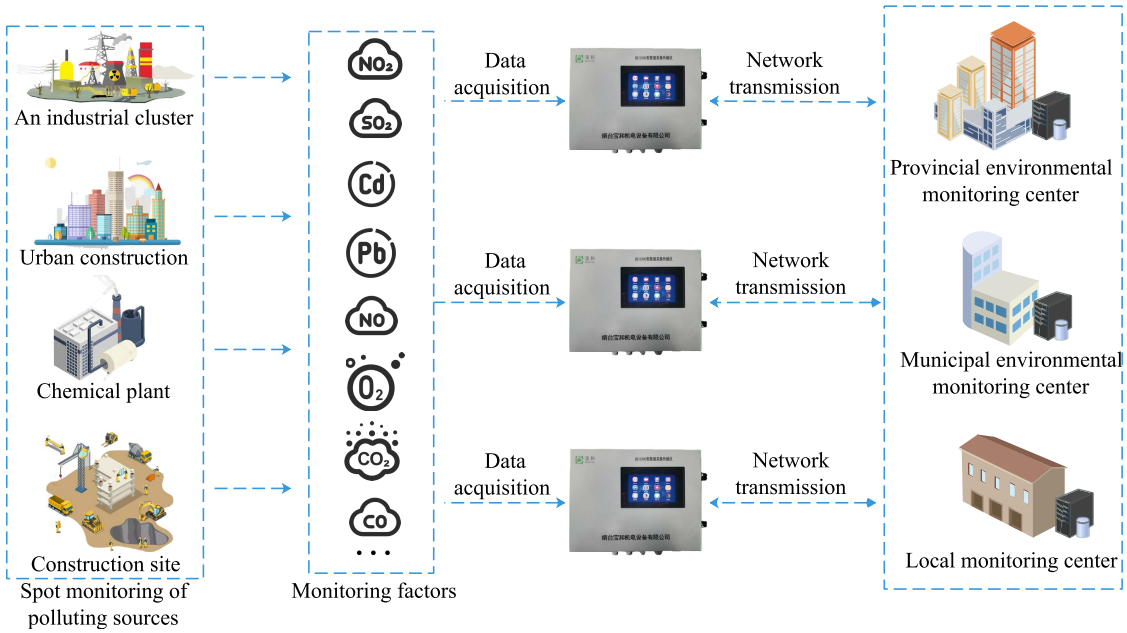


**FIGURE 1.** Existing environmental monitoring model.

questioned the accuracy of the data when faced with penalties from regulators. The regulatory departments often use on-site verification to supervise, fixed evidence, inefficient, and cannot completely solve the problem, increasing the difficulty of environmental governance.

How to achieve efficient and secure transmission, storage, and sharing of data, ensure the authenticity and effectiveness of data, and curb data falsification has become the focus of the current environmental monitoring [6]. Trevathan [7] and Giglione [8] et al. designed a database management system to store monitoring data and realize the collection, management, and sharing of pollutant emission data. Li [9] et al. proposed an environmental monitoring system based on LoRa communication technology. The system is composed of intelligent terminals composed of multiple embedded sensors to collect monitoring data, and the LoRa server stores the monitoring data collected by the gateway in the database. Jiang [10] et al. studied cloud computing platform and marine environment monitoring systems. Parallelizes the processing of corrosive pollution data in the marine environment through virtualization and distributed technology. The above schemes realize the effective collection and analysis of monitoring data from different perspectives, but all the monitoring data is stored in the centralized database, which essentially does not change the centralized management mode of the environmental monitoring system. There are still data silos that the risk of data falsification, and other problems, which leading to a crisis of trust.

The development of blockchain provides new ideas to solve the above problems. Zhong [11] et al proposed an On-site Construction Environment Monitoring (OCEM) framework supporting blockchain. Collect pollutant data through sensors and upload them to the blockchain to prevent data tampering and ensure data transparency among OCEM participants. Song [12] et al. built a framework for a blockchain-based Hazardous Waste Transfer (HWT) management system, which accomplish the sharing of different type of information among participants and ensured the real-time supervision and management of the whole process. Kassou [13] et al. proposed a monitoring and management system based on blockchain and the Internet of Things (IoT). Combined with the smart contract and the mechanism of Delegated Proof of Stake (DPoS) [14], to ensure the effective management, coordination, and monitoring of wastewater and waste discharge.

The above schemes make use of the characteristics of blockchain decentralization to effectively solve the problem of data silos in the environmental monitoring system. But there are still great challenges in practical application. With the increase in the number of nodes, the data acquisition equipment, as a distributed node of the blockchain, it needs to consume a lot of computation, communication, and storage overhead in the packaging, consensus, encryption, and storage of block data. The statistics of the actual environmental monitoring show that the data acquisition equipment needs to save the minute historical data once a minute, and one-minute historical data is about 120 bytes. Assuming that the data acquisition equipment monitors 15 emissions at the same time, the storage capacity is about 2.4M a day. If there are 1000 nodes in the blockchain, each piece of data acquisition equipment needs to consume about 2.4G of storage space per day to store the block data. In the environmental monitoring system, data acquisition equipment plays the role of a data gateway. On the one hand, they collect, analyze and process

real-time monitoring data [15], to calculate and store historical data such as minutes, hours, and days. On the other hand, the monitoring data is transmitted to the supervision platform through the transmission network. But most of the existing data acquisition equipment uses embedded system [16], and their computing and storage capacity is limited. Therefore, the cost of computation, communication, and storage should be minimized to not affect the normal operation of the data acquisition equipment. In addition, because the data acquisition equipment is often installed in remote power plants, sewage treatment plants, and other places, the wireless network is generally used. The transmission delay or node disconnection caused by unstable network signals should be considered to ensure that the system can agree on the data normally.

To address these challenges, we propose a security model of environmental monitoring data based on an blockchain and use blockchain technology to achieve credible traceability and secure sharing of monitoring data. The model integrates Peer-to-Peer (P2P) [17], consensus mechanism, cryptography, and other technologies to effectively reduce the computing, communication, and storage overhead of block nodes.

The main contributions of this paper are summarized in the following four aspects.

1) To solve the problems of data silos and the risk of data falsification in the existing environmental monitoring system, we propose an environmental monitoring model based on alliance chain to realize the secure sharing of monitoring data and curb data forgery. The model only allows users who have passed identity authentication to join the chain, realizes the sharing of monitoring data in the chain and ensures the authenticity and security of environmental monitoring data.

2) To reduce the computing and communication overhead caused by data consensus, a Group Supervised Byzantine Fault Tolerance (GSBFT) mechanism is designed. The consensus node is divided into node types according to the credit value, and the supervisor node is introduced to monitor the behavior of leaders. By increasing the node dynamically to optimize the consistency protocol, the number of interactions needed to reach consensus is reduced and the efficiency of data consensus between blocks is improved.

3) For the problem of monitoring data storage capacity. Through the cloud-chain fusion technology, i.e. the storage mode of ''index on the chain, storage under the chain'', only the monitoring data credentials composed of hash value and index value are stored in the chain, and the monitoring data is stored in the cloud system under the chain, which reduces the storage pressure of the block node and improves the storage capacity of the system.

4) Aiming at the security problems of data transmission and storage that may be brought about by cloud chain fusion. In this paper, we design an AES symmetric encryption algorithm combined with an RSA asymmetric encryption algorithm to reduce the computational overhead of data encryption on the premise of ensuring the security of data transmission and storage.

The rest of this article is organized as follows. Section II introduces the related work. Section III proposes an alliance chain-based environmental monitoring model. Section IV proposes a practical Byzantine fault tolerance mechanism based on group supervision. Section V introduces a secure storage scheme for environmental monitoring data. Section VI analyzes the security advantages of the model, and analyzes the performance of the proposed scheme through experiments. Finally, the summary and prospect of this paper.

## II. RELATED WORK

The introduction of blockchain technology has brought a new decentralized solution for environmental monitoring. However, in the process of practical application, we should note that the computational, communication, and storage overhead is caused by data consistency and block data encrypted transmission and storage [18].

The consensus mechanism is the key to blockchain technology, which affects the processing capacity and security of blockchain. PBFT can reach a consensus in the scenario of a few node failures or forged messages and has a reliable fault tolerance rate. Many studies [19], [20] use the PBFT mechanism to achieve consensus among data sets. Due to the high communication complexity and low consensus efficiency of PBFT, the system has a lot of communication overhead in the process of data consensus. Kotla [21] et al. proposed the Speculative Byzantine Fault Tolerance (SBFT) consensus mechanism to reduce the communication overhead of the PBFT mechanism. The algorithm requires each consensus node to directly process the request sent by the client, and send the processing result to the client to enter the confirmation stage. On the premise that the client does not make mistakes, the communication overhead will be greatly reduced. However, if the client is abnormal, the whole system will be damaged. Liu [22] et al. proposed the FastBFT algorithm, which uses information aggregation technology to combine the hardware-based Trusted Execution Environment (TEE) with lightweight secret sharing primitives. Ideally, the communication complexity of the algorithm is reduced from $O(n^2)$ to $O(n)$, but in the case of failure, the complexity is equivalent to that of PBFT. Therefore, in the environment monitoring model based on the blockchain, choosing the appropriate consensus mechanism can ensure the security of the system and reduce unnecessary costs.

In the environmental monitoring blockchain system, if all the monitoring data were packaged and stored in the blockchain, it would cause great storage pressure on the nodes. The ''cloud chain fusion'' technology chooses to store the index information of the data to the blockchain, while the complete data content is uploaded to the cloud platform to complete the data storage [23], [24], [25]. Zhu [26] et al. proposed a Controllable Blockchain Data Management (CBDM) model. The system uses a cloud system to improve the storage efficiency of the model, and metadata only stores in each block, which reduces the waste of distributed storage. Cha [27] et al. designed a distributed

system to ensure the integrity and security of data. Through cloud storage technology, the data storage problem of smart city environment management based on blockchain is solved. Compared with the existing centralized system, it has higher security, faster transaction speed, and better data storage efficiency. The combination of blockchain and cloud storage can improve the scalability of the platform and provide users with flexible storage space.

The environmental monitoring system should ensure the authenticity and validity of the monitoring data. However, in the ''cloud chain fusion'' system, data may face the risk of being attacked and tampered with in the process of transmission and storage [28]. Data encryption is an effective means to achieve authenticity and security of monitoring data [29]. Hur [30] proposed a smart grid data security scheme based on an attribute encryption algorithm. The private key distributed by the authority is associated with the attribute set, and the ciphertext is associated with the formula on the attribute. Because the pairing calculation will increase linearly with the size of the attribute, if the length of the ciphertext is long, it will consume a lot of computation and communication overhead for encryption and decryption. Ullah [31] et al. adopt an Attribute-Based Access Control (A-BAC) policy, combined with AES for encryption, and use an elliptic curve Diffie-Hellman key exchange protocol for key sharing to ensure the security of data sharing and storage. Peng [32] et al. proposed a blockchain data transmission security scheme based on fully homomorphic encryption, which uses IoT devices to collect asymmetric encrypted public key data to avoid data tampering in the process of data transmission. However, the attribute-based encryption scheme and the fully homomorphic encryption scheme will bring a lot of computation and take a long time in the process of key distribution and data management. For this model, it is necessary to design a scheme that can not only ensure the security of monitoring data transmission and storage but also make the overhead as small as possible.

## III. A DATA SECURITY MODEL OF ENVIRONMENTAL MONITORING DATA BASED ON BLOCKCHAIN

This section introduces the security model of environmental monitoring data based on blockchain, and focuses on the system architecture and theoretical basis of the model.

### A. SYSTEM MODEL

The environmental monitoring model based on the alliance chain is shown in Fig.2. The model is built on the alliance chain, and only certified environmental regulatory departments, sewage factories, social institutions, and environmental protection enterprises can participate in it, realize the sharing of monitoring data in the chain, and ensure the privacy and security of environmental monitoring data.

The model mainly consists of four components: pollutant data analysis, environmental monitoring blockchain, cloud storage, and scenario application.

1) Pollutant data analysis: According to different pollutant emission sources, different types of instruments are used to monitor the pollutant emission status and process parameters. such as $NO_2$ and $SO_2$ emissions in air quality monitoring, and Pb and COD emissions status in water quality monitoring.

2) Environmental monitoring blockchain: While performing the original data gateway function, the data acquisition equipment acting as a distributed node in the blockchain, and it is responsible for sending the collected surveillance data to the blockchain and encrypting it through an encryption algorithm. After passing by consensus, the monitoring data credentials are added to the blockchain, and the complete monitoring data is transmitted to the cloud platform.

3) Cloud storage: Responsible for providing enormous data storage services for storing complete environmental monitoring data, equipment information, etc.

4) Scenario application: The environmental monitoring blockchain system provides secure and credible monitoring data for regulatory authorities at all levels, monitoring point enterprises, and individuals. The regulatory departments at all levels can evaluate the environmental quality more objectively and accurately, and implement environmental management and decision-making. Monitoring enterprises can use credible monitoring data to achieve data transactions. Such as carbon emissions trading, emissions trading, pollution permit trading, and so on.

### B. DESIGN OBJECTIVE

The model aims to achieve the following design goals.

1) Data integrity storage: Data integrity is an important basis to reflect the authenticity and reliability of data, including the integrity of data storage and use [33]. This paper proposes a hybrid storage mode of cloud chain fusion, which can reduce the pressure of blockchain storage and ensure the integrity of monitoring data.

2) Data security and access control: The current cloud service cannot provide strong security defense measures, this paper uses an encryption algorithm to encrypt the monitoring data to ensure the authenticity of the data and achieve data security. The selected alliance chain is deployed in the network, and only users who pass the identity authentication can access the data and participate in consensus to achieve access control.

3) System availability: The introduction of blockchain changes the hierarchical centralized management mode of the existing environmental monitoring model, so that the monitoring data can be traced. In this paper, both in the data consensus and the choice of encryption algorithm, the carrying capacity of data acquisition equipment is fully considered. To ensure data security and low system overhead.

## IV. GROUP SUPERVISION OF BYZANTINE FAULT TOLERANCE MECHANISM

### A. OVERVIEW OF CONSENSUS ALGORITHMS

As the core technology of blockchain, the consensus mechanism is an essential guarantee for selecting accounting

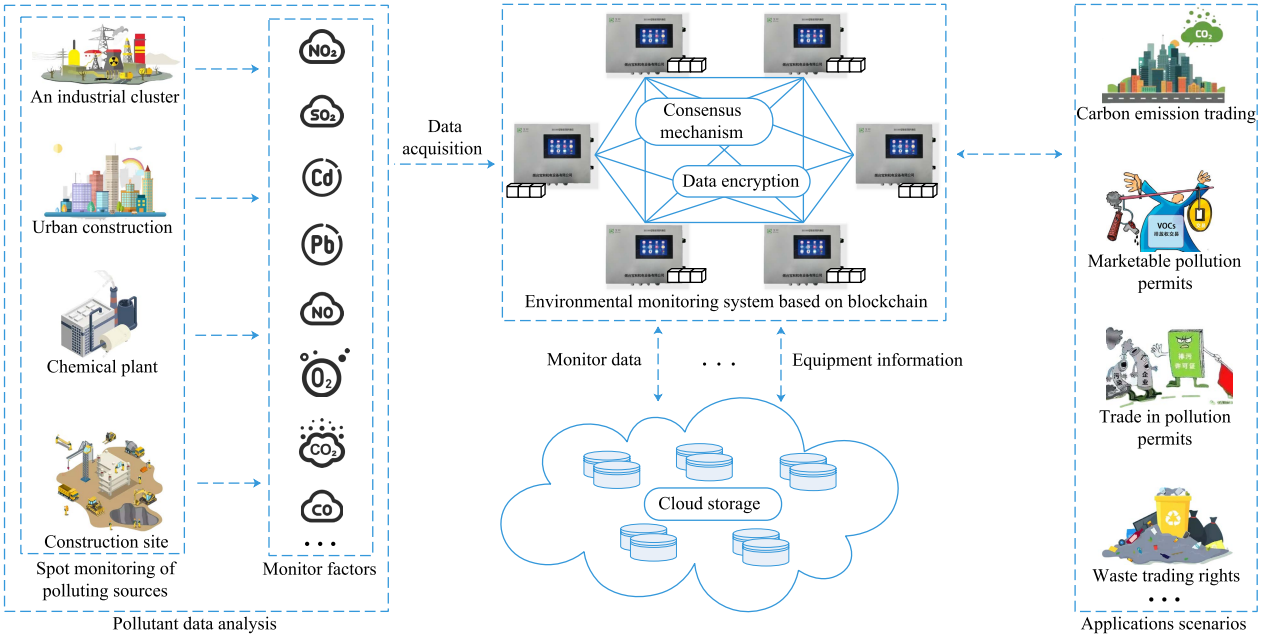


**FIGURE 2.** Data security model of environmental monitoring based on alliance chain.

nodes and achieving data consistency of participating nodes. The common consensus mechanisms are Proof of work (PoW) [34], Proof of Stake (PoS) [35], DPoS and Practical Byzantine Fault Tolerance (PBFT) [36]. PoW was first used in Bitcoin. The algorithm is simple and easy to implement, and the security is relatively high, but reaching a consensus requires a lot of computation, extreme energy consumption, and low speed, resulting in a lot of waste of resources [37]. PoS reduces the waste of PoW resources, but accounting rights are mostly controlled in the nodes with the highest rights and interests, weakening decentralization. DPoS has a short time and low energy consumption, which reduces the waste of computing power to a certain extent, but the enthusiasm of ticket holders to participate in voting is not high, and the rights and interests are centralized. [38].

By analyzing these consensus algorithms, and considering the characteristics of environmental monitoring. PoW, PoS, and DPoS consensus algorithms are not suitable for environmental monitoring. In contrast, PBFT enables distributed systems to reach consensus in the scenario of a small number of nodes making errors or forging messages, and is usually used in alliance chain, with the characteristic of reliable fault tolerance.

### B. CREDIT GROUPING POLICY AND CONSENSUS NODE SELECTION

With the increase of consensus nodes in the alliance chain, the application of PBFT in the environmental monitoring blockchain system can lead to an increase in system overhead and other problems. In addition, since the system cannot eliminate the error nodes, it will lead to repeated errors in the process of operation, which affect the stability of the system. To remedy the above shortcomings, this paper designs the GSBFT mechanism for the environmental monitoring blockchain system based on the original advantages of the PBFT consensus algorithm and combined with the Raft [39] consensus idea. The improved PBFT algorithm can reduce communication overhead and maintain high performance of the system. The GSBFT consensus algorithm has made the following improvements.

1) Credit mechanism: Measure the trustworthiness of the monitoring node by the credit value, analyze the authenticity and integrity of the data transmitted by the monitoring node, and get the credit value of each node. According to the credit value, all monitoring nodes are classified into different roles, namely, leader, supervisory, and ordinary node. If nodes are unable to complete the task within the specified time due to network delays, downtime, or malicious behavior, then the system will reduce the credit value. The system updates the credit values of nodes and reorders them each cycle $T$. The node transition relation is shown in Fig. 3.

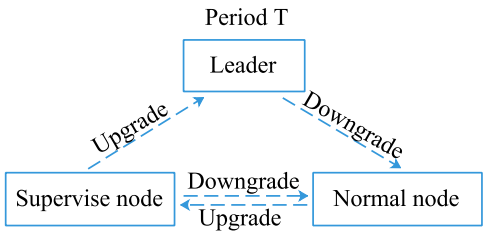


**FIGURE 3.** Node transition relation.

2) Grouping strategy: Give priority to the leaders with high credit value as the primary node, and participate in the

consensus through the PBFT mechanism. In addition to leaders, the same number of supervisory nodes are selected in the same way, and the remaining nodes are randomly grouped. The Raft mechanism is used within each group to participate in the consensus.

3) Supervision mechanism: The supervision node is introduced to supervise the behavior of the leader to ensure that Raft can tolerate Byzantine nodes during the consensus process. A group may have one or more supervisor nodes, and a supervisor node can supervise the leaders of multiple groups at the same time. Once a leader is found to have a different message than the other group leaders, the node is judged to be malicious. The group supervision strategy is shown in Fig.4.

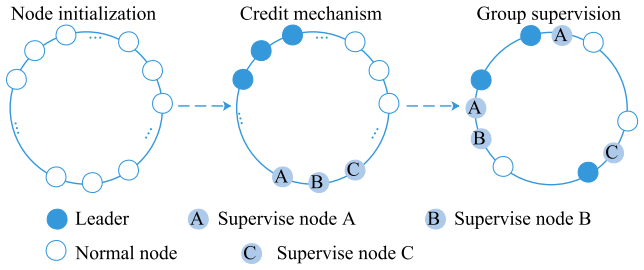


**FIGURE 4.** Group supervision Strategy.

### C. GSBFT CONSISTENCY PROTOCOL

This section details the consistency protocol of the GSBFT algorithm. The GSBFT consistency protocol consensus is shown in Fig.5.

1) GSBFT-PRE-PREPARE phase: The primary node packages and processes the monitoring data into a data block $m$, assigns the sequence number $N$ and appends the view number $v$ and other information. According to the content of the data block $m$, generates the PRE-PREPARE message $\langle\langle GSBFT-PRE-PREPARE, v, N, d, t\rangle \delta_p, m\rangle$, where $d$ is the message digest of $m$, $t$ is the timestamp to ensure the consensus order, and $\delta_p$ is the signature of the primary node to the message $m$. Then the primary node broadcasts the message to other nodes, and the other nodes check $v$, $d$, and $\delta_p$ after receiving the message. If the check passes, they enter the GSBFT-PREPARE phase. Otherwise, the primary node is questioned and broadcast to replace the primary node.

2) GSBFT-PREPARE phase: After passing the message verification, the node broadcasts the message $\langle GSBFT-PREPARE, v, N, d, i\rangle \delta_i$ to other nodes except its own node, where $i$ is the current node number, $i \in \{1, 2, \cdots, |R|\}$, $|R|$ is the total number of nodes, $\delta_i$ is the signature of the current node $i$ to $m$. The remaining nodes examine the received messages, and when any node enters the next stage after receiving $2f+1$ PREPARE message, $f$ is the number of tolerable Byzantine nodes. Otherwise, the consensus process is suspended, the consensus failure is determined, and the consensus process is and re-initiated by the primary node.

3) GSBFT-COMMIT phase: Each node enters the GSBFT-COMMIT phase after passing the message verification in the GSBFT-PREPARE phase and ensuring the receipt of $2f+1$ messages that the node has passed the verification. The GSBFT-COMMIT phase is mainly used to verify the correctness of the monitoring data saved by each node. Send the message $\langle GSBFT-COMMIT, v, N, d, i\rangle \delta_i$ to other nodes, including the primary node, the node receives the COMMIT message and verifies the message sequence $N$, message digest $d$, and view number $v$. If the verification is successful, the COMMIT message is written to the log. When any node receives the commit votes of $2f+1$ different consensus nodes, the GSBFT-COMMIT phase is completed and enters the Raft-PRE-PREPARE consensus phase. If not enough messages are received within a certain period, the consensus fails.

4) Raft-PRE-PREPARE phase: The leaders of each group package the messages and generate a log $\langle Raft-PRE-PREPARE, N, d, i\rangle \delta_i$, and broadcasts to the follower nodes.

5) Raft-PREPARE phase: The follower receives the log and provides feedback to the leader and generates the corresponding log.

6) Raft-COMMIT phase: If the leader receives more than $n_r/2$ messages ($n_r$ represents the number of member nodes in the group) according to the feedback of the followers, the leader thinks that the message is valid and has reached a consensus, i.e., the message is written to the blockchain.

## V. SECURE STORAGE SOLUTION FOR ENVIRONMENTAL MONITORING DATA

### A. DATA PUBLISHING AND STORAGE

To solve the problem of monitoring data storage capacity. In this paper, the hybrid storage mode of cloud chain fusion is realized by combining blockchain with cloud servers. Fig.6 shows the storage structure of cloud chain fusion.

Fig. 6 (a) shows the monitoring data credentials for block $i$. The sender packages the uploaded monitoring data into blocks and reaches a consensus through the GSBFT consensus mechanism. The monitoring data credentials composed of the hash value and index value of the monitoring data are stored in the blockchain to prevent the monitoring data from being maliciously tampered with. The monitoring data credentials are shown in Table 1.

**TABLE 1.** Monitoring data credential content.

| Symbolic representation | Meaning |
|---|---|
| $Hash(Block_i)$ | The Hash value of the current block. |
| Timestamp | The time when data is written to the block. |
| $Sig_i$ | The node signature that processes the current monitoring data. |
| $Index_i$ | Data index, pointing to monitoring data stored in the cloud. |
| Equipment number | Unique identification number of monitoring equipment. |

Fig. 6 (b) shows the monitoring data structure of block $i$. The AES algorithm is used to encrypt the monitoring data,

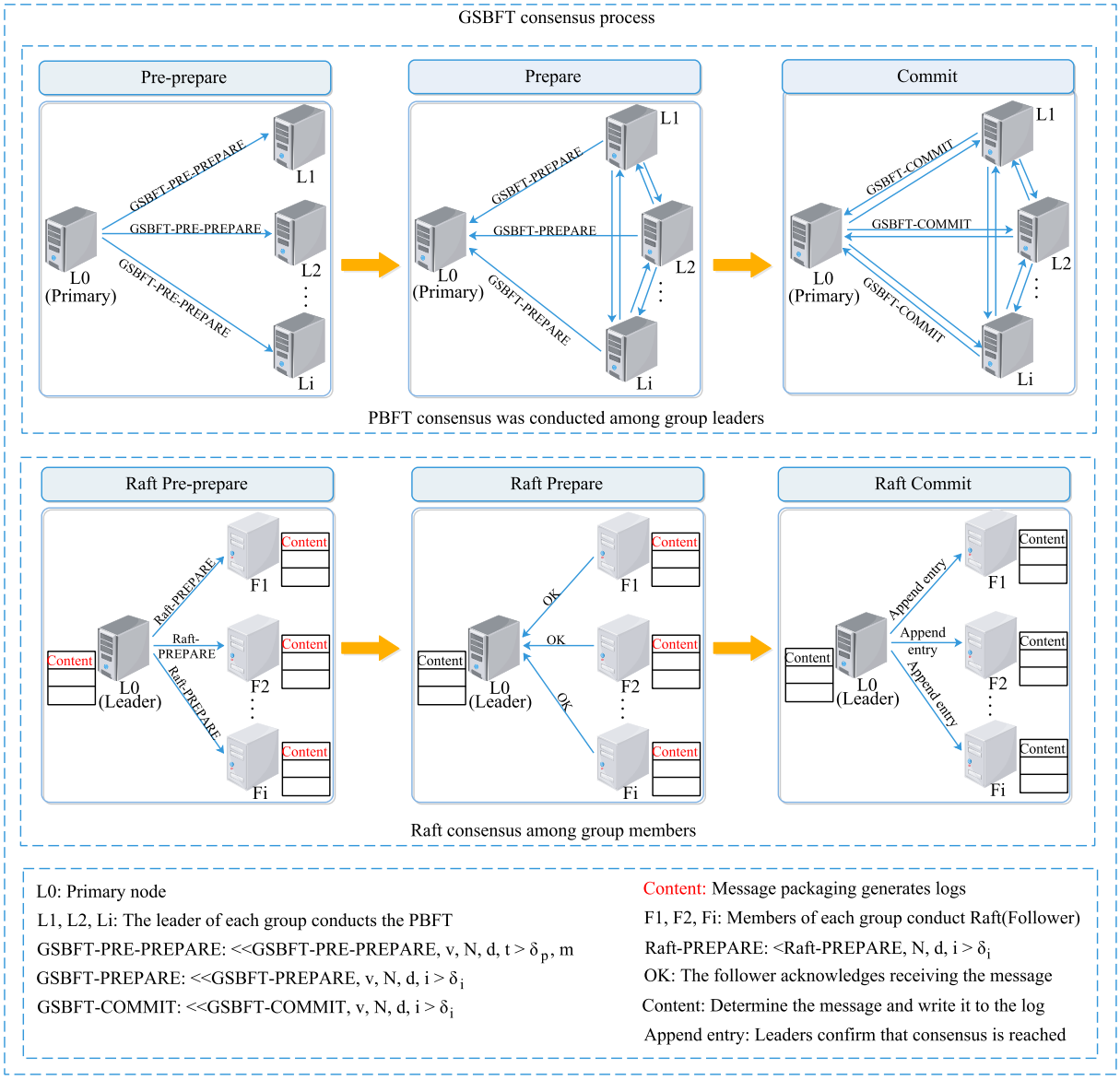


**FIGURE 5.** GSBFT Consistency protocol execution flowchart.

and the public key $Pk_i$ of the RSA algorithm competes to encrypt the key $k$ of the AES algorithm. Upload the encrypted monitoring data to the cloud database under the chain to realize the effective storage of all monitoring data. Through the Hash chain to achieve efficient and rapid traceability analysis of monitoring data. The monitoring data are shown in Table 2.

### B. DATA ENCRYPTION MECHANISM

A more desirable approach based on blockchain-based data storage architecture is to take data separation, with meta-data or hash values retained on the chain, and complete data stored under the chain. This can give better play to its scalability, but also ensure the traceability and integrity of the data. In terms of data storage, cloud storage has the advantages of massive storage capacity, flexible expansion, high data availability, and so on. However, the current cloud service cannot provide strong security measures, its security

**TABLE 2.** Monitoring data structure content.

| Symbolic representation | Meaning |
|---|---|
| Monitor Data | Details of monitoring data. |
| Timestamp | The time when data is written to the block. |
| $Sig_i$ | The node signature that processes the current monitoring data. |
| Data Type | Monitoring data category. |
| Equipment number | Unique identification number of monitoring equipment. |

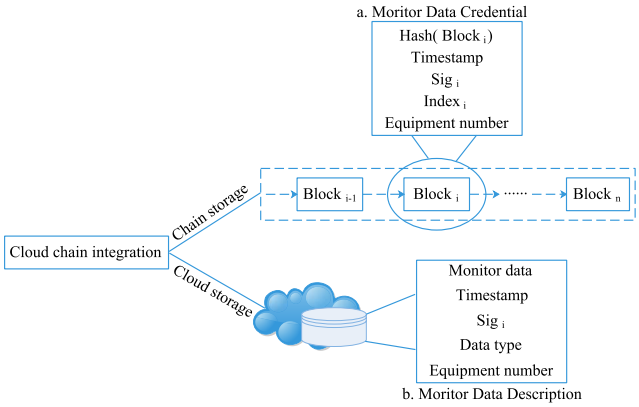


**FIGURE 6.** Cloud Chain Fusion Storage Architecture.

**TABLE 3.** Main symbols and their meanings.

| Symbolic representation | Meaning |
|---|---|
| md | Monitoring data. |
| $C_{AES}$ | Monitoring data ciphertext encrypted by AES algorithm. |
| $k$ | Key of AES algorithm. |
| $C_{RSA}$ | Ciphertext obtained by encrypting the key of AES algorithm by RSA algorithm. |
| $Pk_i$ | Public key of RSA algorithm. |
| $Sk_i$ | Private key of RSA algorithm. |
| $\alpha$ | Common modulus. |
| $Hash(C_{AES})$ | Hash value of $C_{AES}$. |
| $Sig(C_{AES})_{Sk_i}$ | Digital signature of $C_{AES}$. |
| $index$ | Content indexing of monitoring data. |
| $C_{index}$ | Content index ciphertext encrypted by AES algorithm. |
| $C_{Block}$ | Monitoring data voucher. |
| $Hash_{Block}(C_{AES})$ | Hash value of $C_{Block}$. |
| $Sig_{Block}(C_{AES})$ | Digital signature of $C_{Block}$. |

problems are still questioned. Such as illegal users accessing the cloud server, resulting in illegal data acquisition or malicious tampering of data. Therefore, this model adopts AES symmetric encryption combined with RSA asymmetric encryption to ensure the security of monitoring data in the process of transmission and storage. The specific encryption process is as follows, and the main symbols used are shown in Table 3.

1) The data acquisition equipment collects the uploaded monitoring data *md*, and packages the data for consensus verification.

2) The data acquisition equipment performs AES symmetric encryption on *md* to obtain $C_{AES}$, and the key $k$ is encrypted by the $Pk_i$ to obtain $C_{RSA}$.

AES symmetric encryption algorithm is detailed in Algorithm 1.

**Algorithm 1** AESencrypt(byte[] Md, byte[] k)

**Input:** *md* is the content to be encrypted, $k$ is the symmetric key ( $k$ is a randomly generated 16-bit string)

**Output:** Ciphertext $C_{AES}$

```
if k.length ≠ 16 then
    Throws a runtime exception with the message "AES symmetric key k must be 16 bytes"
else
    Create a cryptograph cipher
    if Map md to 4 × 4 byte matrix in 16 byte groups then
        return C_AES
    else
        Throws a runtime exception with the message "Data encryption failed"
    end if
end if
```

RSA asymmetric encryption key generation is detailed in Algorithm 2.

**Algorithm 2** Map<String, String> generateKeyPair()

**Input:** None

**Output:** Public-Private Key Pair: map.put(key:"publicKey", $Pk_i$), map.put(key:"privateKey", $Sk_i$)

```
SecureRandom(): generating random primes p and q to satisfy p ≠ q
Calculate the α = p × q and euler number Φ(α) = (p − 1)(q − 1)
Randomly generated e ∈ Z*, which satisfies 1 < e < Φ(α) and is mutually prime with Φ(α)
Find the modulo inverse element d of e with respect to Φ(α) such that d × e mod Φ(α) = 1
Get map.put("publicKey", Pk_i)
Get map.put("privateKey", Sk_i)
return map
```

RSA asymmetric encryption algorithm is detailed in Algorithm 3.

**Algorithm 3** RSAencrypt(byte[] Source, byte[] publicKey)

**Input:** *source* is encrypted data, i.e., the symmetric key k; *publicKey* is the RSA public key $PK_i$

**Output:** Ciphertext $C_{RSA}$

```
Encryption of the source using the publicKey of RSA
return C_RSA
```

3) The $C_{AES}$ calculates the Hash value by the hash function SHA-256, $Hash(C_{AES}) = SHA(C_{AES})$, and digitally sign through Formula (1).

$$Sig(C_{AES})_{Sk_i} = (Hash(C_{AES}))^d \bmod \alpha \tag{1}$$

Verify $Hash(C_{AES})$ and $Sig(C_{AES})_{Sk_i}$. After verification, the $C_{AES}$, $Sig(C_{AES})_{Sk_i}$ and other information will be uploaded to the cloud database.

4) The *index* is encrypted by $k$ to get $C_{index}$. The $Hash(C_{AES})$, $Sig(C_{AES})_{Sk_i}$, Timestrap, $C_{index}$, and device number are packaged into the $C_{Block}$, and digitally sign it through Formula (2).

$$Sig(C_{Block})_{Sk_i} = (Hash(C_{Block}))^d \bmod \alpha \tag{2}$$

Verifies whether $Hash(C_{Block})$ and $(Sig(C_{Block})_{Sk_i})^e \bmod \alpha$ are the same. If the same, it means that the monitoring data has not been tampered with during the upload process, and the $C_{Block}$ is uploaded to the blockchain, otherwise the monitoring data will be invalidated and will not be stored in the blockchain.

5) When decrypting the monitoring data. Firstly, the symmetric key $k$ is obtained by using the $Sk_i$ decryption in Formula (3).

$$k = {C_{RSA}}^{Sk_i} \bmod \alpha \tag{3}$$

AES symmetric algorithm uses $k$ to decrypt the $C_{index}$ to get $index = D(k, C_{AES})$. Get $C_{AES}$ through the *index*. Calculate whether the Hash value of the $C_{AES}$ is consistent with the Hash value stored in the blockchain, ensure that the data has not been tampered with, and verify the Timestamp avoids replay attacks. After the verification is passed, the monitoring data is finally obtained by decrypting it with $k$.

The RSA decryption algorithm is detailed in Algorithm 4.

**Algorithm 4** RSAdecrypt(byte[] Cryptograph, byte[] privateKey)

**Input:** *cryptograph* is the data to be decrypted, i.e., the symmetric key k, *privateKey* is the RSA private key $Sk_i$

**Output:** Symmetric key $k$

1: Decryption of the *cryptograph* using the *privateKey* of RSA
2: **return** $k$

AES decryption algorithm is detailed in Algorithm 5.

**Algorithm 5** AESdecrypt(byte[] Data, byte[] k)

**Input:** *data* is the content to be decrypted, which is the ciphertext $C_{AES}$, and $k$ is the symmetric key

**Output:** Monitoring data *md*

1: **if** k.length $\neq$ 16 **then**
2: Throws a runtime exception with the message ''AES symmetric key $k$ must be 16 bytes''
3: **else**
4: Create a cryptograph *cipher*
5: **if** Decrypt using symmetric key $k$ to get *md* **then**
6: **return** *md*
7: **else**
8: Throws a runtime exception with the message ''Data decryption failed''
9: **end if**
10: **end if**

## VI. EXPERIMENTAL RESULTS AND ANALYSIS

### A. SECURITY ANALYSIS

#### 1) DATA INTEGRITY AGAINST TAMPERING

Each block in the blockchain records the Hash value of the previous block. if you want to change the block data, the attacker needs to tamper with the Hash value of the previous block. The unidirectionality of the Hash function determines that the modification of any node will involve other nodes, which leads to the high cost of data tampering. Meanwhile, when generating data blocks, the environmental monitoring blockchain adopts the GSBFT mechanism. Assuming that the total number of nodes in the current system is $n = 3f + 1$, and if some nodes want to tamper with the data, at least $f + 1$ nodes vote for the transaction. Assuming that the total number of nodes in the current system is 100, and each node has a 50% chance of becoming a malicious node, the fault tolerance rate of the PBFT consensus mechanism is about 33%. For some nodes, the probability of tampering with data is only $1.16 \times 10^{-10}$. Therefore, when the monitoring data is verified by consensus, the monitoring data credential is written into the blockchain, and the complete monitoring data is uploaded to the cloud database, thus ensuring the integrity of the monitoring data.

#### 2) DATA AUTHENTICITY AND TRACEABILITY

During data transmission, each block will attach information such as digital signature $Sig(C)_{Sk_i}$ and message digest to ensure the legitimacy and authenticity of the current data. Only the transactions verified by the consensus algorithm are allowed to be written into the blockchain, and will be Timestamp is stamped when writing, so that the blockchain has the characteristic of time order. The chain storage structure is formed by connecting the Hash value to ensure that the blockchain data can be traced back to the source. Regulators can trace back to suspicious data through the blockchain, avoiding the negative impact of malicious nodes.

#### 3) ANTI-NODE ATTACKS

In the case of no third-party organization, the blockchain achieves the consensus mechanism to generate data blocks among nodes, and each node maintains a complete data chain. it avoids the defect of the whole network interruption caused by the failure of the central node or malicious attack. The decentralized storage mode is adopted to realize the common maintenance of the data of each node, and the data damage or loss of any node will not affect the overall operation, thus solving the fault problem of a single node.

### B. PERFORMANCE ANALYSIS

In this experiment, the experimental data are all derived from the minute data collected by the data acquisition equipment, which is framed depending on the ''Data Transmission Standard for Pollutant Online Monitoring System HJ/T212-2017'' (HJT212-2017). The experimental process includes the GSBFT consensus mechanism, and the process of

encryption and decryption of monitoring data by AES combined with the RSA algorithm. The PBFT algorithm and the GSBFT algorithm are implemented in the go language, and the experiment is compared intuitively. The experimental environment configuration is shown in Table 4.

**TABLE 4. Experimental environment.**

| Configuration | Detailed Information |
|---|---|
| CPU | Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz |
| Operating system | Windows 10 |
| RAM | 8GB DDR4 |
| Hard Disk | 1.82TB HDD |
| Golang version | go1.17 windows/amd64 |

### 1) CONSENSUS DELAY

The consensus delay is an important indicator to reflect the performance of the consensus algorithm, which refers to the time required from transaction initiation to transaction completion. Reducing the consensus delay can improve the operation efficiency of the system. The consensus delay for GSBFT is calculated by Formula (4), as follows.

$$T_{DelayTime} = T_{Accomplish} - T_{Receive} \tag{4}$$

$T_{DelayTime}$ indicates the consensus delay, $T_{Accomplish}$ indicates the time for the block to complete the consensus confirmation, and $T_{Receive}$ indicates the time when the transaction is written to the block.

This experiment simulates the GSBFT consensus process by configuring several virtual nodes. Under the premise that there are no wrong nodes, the number of nodes is taken as the experimental variable. The ordinate is the delay consumption in the consensus process, and the abscissa is the number of nodes participating in the consensus. 200, 400, 600, 800, and 1000 nodes are selected to participate in the node consensus in the system, each group of nodes performs 10 experiments to take the average as the delay of this transaction. The performance of the algorithm under these nodes is analyzed and studied through the experimental results. In the same experimental environment, the experimental results of PBFT and GSBFT with different number of groups and members in different groups are shown in Fig. 7.

As can be seen from Fig. 7, the delay growth of PBFT algorithm is faster than that of the GSBFT algorithm. For the GSBFT algorithm with the determined number of groups, the performance of consensus delay is basically stable with the increase of nodes. In the case of different groups, the consensus delay tends to increase slowly. With the increase in the number of nodes, the gap between the delay of the GSBFT algorithm and that of the PBFT algorithm will become larger and larger. By comparison, it can be seen that the GSBFT algorithm shows a lower delay in network communication under the same experimental environment.

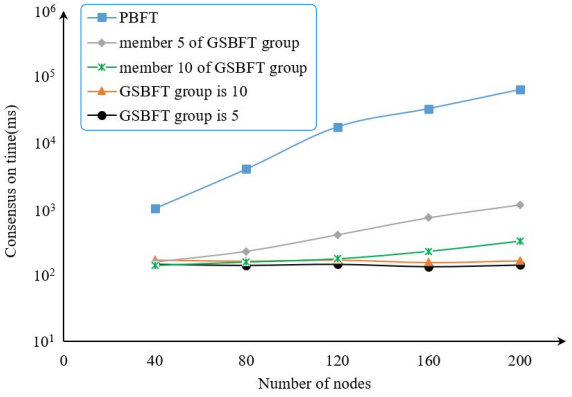


**FIGURE 7. Comparison of PBFT and GSBFT consensus delay.**

### 2) COMMUNICATION OVERHEADS

With the increase of nodes, the PBFT algorithm needs a lot of data communication when it comes to the consensus between nodes. Assuming that the current number of consensus nodes is $n$, the number of communications needed in the Pre-prepare phase of the PBFT algorithm is $n-1$, the number of communications needed in the Prepare phase is $(n-1)^2$, and the number of communications needed in the Commit phase is $n(n-1)$. Therefore, the total number of communications in PBFT is $2n(n-1)$, and the communication complexity is $O(n^2)$.

In the GSBFT algorithm, the number of communications needed by the Raft algorithm in the logging phase is $n-1$, the number of communications needed in the Commit phase is also $n-1$, the total number of communications is $2(n-1)$, and the communication complexity is $O(n)$. The GSBFT algorithm reduces the communication overhead caused by data consensus in the system by grouping the nodes joining the consensus. Assuming that the current number of groups is $g$, the Raft algorithm consensus is used in the group, the communication complexity is $O(\frac{n}{g})$, the PBFT algorithm consensus is used outside the group, and the communication complexity is $O(g^2)$. Therefore, compared with the PBFT algorithm, the communication complexity of the GSBFT algorithm is reduced from $O(n^2)$ to $O(\frac{n}{g}) + O(g^2)$.

As shown in Fig. 8. The ordinate is the number of communications consensus by the node, and the abscissa is the number of nodes in the system that participate in the consensus. 20, 40, 60, 80, and 100 nodes are taken to participate in node consensus in the system, and each group of nodes performs 10 experiments, and takes the average value as the number of communications for this transaction. Under the different number of nodes, the communication times and their changing trends of PBFT, GSBFT with 5 members within a group, and GSBFT with 5 groups are compared.

As can be seen from the picture, the communication overhead increases as the number of nodes in the system increases. The communication cost of GSBFT algorithm is much less than that of PBFT algorithm in either a fixed number of packets or a fixed number of members in a group, and the speed of

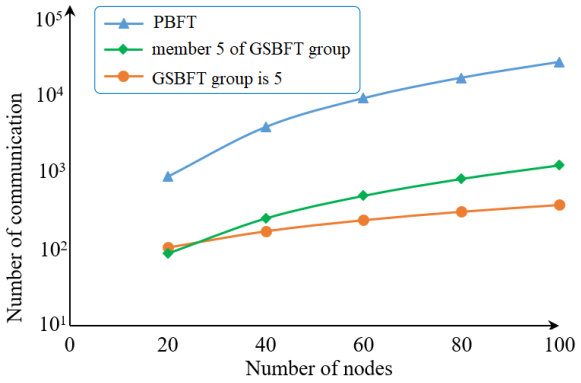


**FIGURE 8.** Comparison of PBFT and GSBFT communication overhead.

communication times increasing with nodes is relatively slow compared with PBFT algorithm. The experimental results show that GBFT greatly reduces the communication overhead in the system.

### 3) STORAGE STABILITY

In the practical application scenario of the data acquisition equipment, with the increase of the amount of monitoring data stored in the block, the data storage content will greatly affect the response time of the environmental monitoring blockchain system. In this experiment, 8 groups of flue gas emission data transactions are randomly selected to compare the storage stability of the monitoring data under the existing environmental monitoring model with that under the environmental monitoring blockchain model. As shown in Fig. 9, the abscissa is the number of transactions, and the ordinate is the storage occupancy.

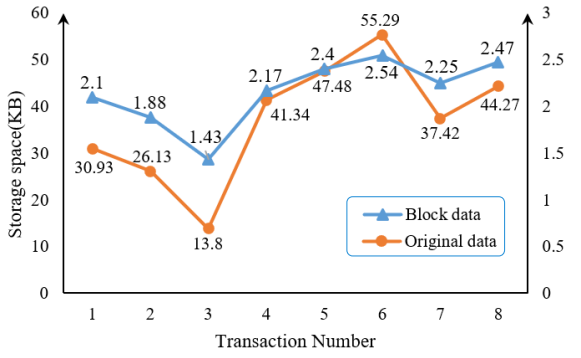


**FIGURE 9.** Comparison of data stability under different storage environments.

In the environmental monitoring blockchain model, only monitoring data credentials composed of Hash values and index values of monitoring data are stored in the chain, and the length of the data is fixed. For the existing environmental monitoring model, due to the different monitoring factors and monitoring objects, the data length is also different. The comparison of the two sets of data transactions shows that the increase or decrease of the length of monitoring data has little impact on the storage of the environmental monitoring blockchain and shows better stability. Therefore, the storage of the environmental monitoring blockchain system can provide a faster response and maintain lower storage consumption.

### 4) DATA ENCRYPTION AND DECRYPTION

In practical application, the efficiency of data encryption and decryption proposed in this paper is closely related to the overall time delay of the system. To better analyze the feasibility of encryption and decryption of monitoring data, 30 minute data are taken as a group, and 5, 10, 15, 20, 25, and 30 groups of monitoring data are set to test the encryption algorithm proposed in this scheme. Each group of data is tested 10 times, and the average value is taken as the time spent on encryption and decryption. Fig. 10 shows that data encryption takes time, Fig. 11 shows that data decryption takes time, the abscissa is the number of monitoring data groups, and the ordinate is time consumption.

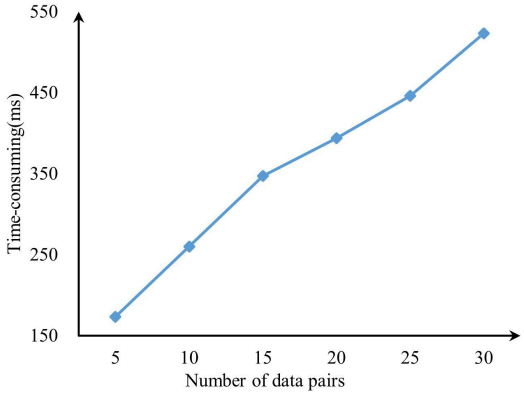


**FIGURE 10.** Data encryption efficiency.

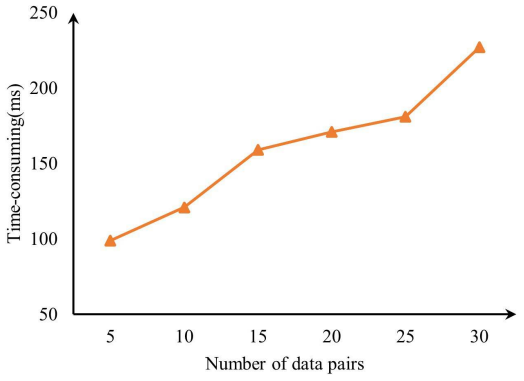


**FIGURE 11.** Data decryption efficiency.

As can be seen from Fig. 10 and Fig. 11, with the increase of the amount of monitoring data, the time consumed in the process of encryption and decryption of data gradually increases, but all of them are kept in a low time range, so that the scheme can maintain a low delay while ensuring

**TABLE 5. Comparison of consensus algorithms.**

| | Rely on tokens | Node management | Consistency | Throughput/TPS | Scalability | Communication complexity | Consensus delay |
|---|---|---|---|---|---|---|---|
| PoW [34] | Yes | No license | Will bifurcate | $< 10$ tx/s | Strong | $O(n^2)$ | High |
| PoS [35] | Yes | No license | Will bifurcate | $< 300$ tx/s | Strong | $O(n^2)$ | High |
| DPoS [14] | Yes | No license | No bifurcation | $< 10^3$ tx/s | Weak | $O(n^2)$ | Higher |
| Raft [39] | No | Permission required | No bifurcation | $< 10^4$ tx/s | Strong | $O(n)$ | Low |
| PBFT [36] | No | Permission required | No bifurcation | $< 10^3$ tx/s | Weak | $O(n^2)$ | Low |
| SBFT [21] | No | Permission required | No bifurcation | $10^3 - 10^4$ tx/s | Strong | $O(n)$ | Low |
| FastBFT [22] | No | Permission required | No bifurcation | $10^3 - 10^4$ tx/s | Strong | $O(n \log_2 n)$ | Low |
| GSBFT | No | Permission required | No bifurcation | $10^3 - 10^4$ tx/s | Strong | $O(\frac{n}{g}) + O(g^2)$ | Low |

the security of monitoring data during transmission and storage. At the same time, the feasibility of the scheme is proved.

#### 5) COMPARISON OF CONSENSUS SCHEMES

This section compares the proposed consensus mechanism with the eight existing consensus options, as shown in Table 5. The throughput (TPS) of PoW, PoS, and DPoS is low and depends on tokens, and nodes can join without permission, which is not suitable for current application scenarios. Compared with PBFT, Raft can only tolerate failure nodes. Therefore, this paper chooses the GSBFT consensus algorithm to achieve data consistency in the environmental monitoring blockchain. Compared with other mainstream improved PBFT algorithms, although the throughput of the GSBFT algorithm is not optimal, in the application scenario of this paper, this scheme can take into account a higher fault tolerance rate and lower communication overhead, ensuring the credibility of the participating consensus nodes, so that each node can participate in the consensus more democratically. In application scenarios with high requirements for data security and system overhead, this scheme has obvious advantages.

## VII. CONCLUSION

To solve the problems of data silos and data falsification in the existing environmental monitoring system, this paper proposes a data security scheme of environmental monitoring based on blockchain. It breaks the centralized management mode of the existing environmental monitoring system, and it is a subversive change to the current environmental supervision mode. We design a practical Byzantine fault tolerant mechanism based on group supervision (GSBFT), which reduces the computing and communication overhead of blocks. The data storage structure of cloud chain fusion significantly reduces the data storage pressure of the monitoring node. The monitoring data is encrypted by the AES symmetric algorithm combined with the RSA asymmetric algorithm, which ensures the security of monitoring data in transmission and storage. The experimental results show that the proposed scheme can effectively improve the security and reliability of the environmental monitoring process, taking into account the communication overhead and storage pressure of the system. The model proposed in this paper mainly solves the problems existing in environmental monitoring, and how to use credible monitoring data for environmental transaction needs to be considered. Therefore, in the future, our work will further consider the combination of smart contracts to optimize the environmental monitoring blockchain model. To realize the automatic trading of emission rights, the emission data of enterprises will become real assets. Promote the application of environmental monitoring blockchain to a wider range of scenarios.

## REFERENCES


[1] X. Pang, L. Chen, K. Shi, F. Wu, J. Chen, S. Fang, J. Wang, and M. Xu, "A lightweight low-cost and multipollutant sensor package for aerial observations of air pollutants in atmospheric boundary layer," *Sci. Total Environ.*, vol. 764, Apr. 2021, Art. no. 142828, doi: 10.1016/j.scitotenv.2020.142828.

[2] J. Saez, R. Catalan-Carrio, R. M. Owens, L. Basabe-Desmonts, and F. Benito-Lopez, "Microfluidics and materials for smart water monitoring: A review," *Analytica Chim. Acta*, vol. 1186, Nov. 2021, Art. no. 338392, doi: 10.1016/j.aca.2021.338392.

[3] S. Das, S.-H. Lee, P. Kumar, K.-H. Kim, S. S. Lee, and S. S. Bhattacharya, "Solid waste management: Scope and the challenge of sustainability," *J. Cleaner Prod.*, vol. 228, pp. 658–678, Aug. 2019, doi: 10.1016/j.jclepro.2019.04.323.

[4] E. Biber, "The challenge of collecting and using environmental monitoring data," *Ecol. Soc.*, vol. 18, no. 4, pp. 1–14, 2013, doi: 10.5751/ES-06117-180468.

[5] C. Yin, J. Xi, R. Sun, and J. Wang, "Location privacy protection based on differential privacy strategy for big data in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3628–3636, Aug. 2018, doi: 10.1109/TII.2017.2773646.

[6] B. Zhou and X. Li, "The monitoring of chemical pesticides pollution on ecological environment by GIS," *Environ. Technol. Innov.*, vol. 23, Aug. 2021, Art. no. 101506, doi: 10.1016/j.eti.2021.101506.

[7] J. Trevathan and R. Johnstone, "Smart environmental monitoring and assessment technologies (SEMAT)—A new paradigm for low-cost, remote aquatic environmental monitoring," *Sensors*, vol. 18, no. 7, p. 2248, Jul. 2018, doi: 10.3390/s18072248.

[8] G. Giglione, A. Annibaldi, A. Iaccarino, R. Capancioni, G. Borghini, F. Ciabattoni, S. Illuminati, G. Pace, F. Memmola, and G. Giantomassi, "An integrated web-based GIS platform for the environmental monitoring of industrial emissions: Preliminary results of the project," *Appl. Sci.*, vol. 12, no. 7, p. 3369, Mar. 2022, doi: 10.3390/app12073369.

[9] W. Li, G. Liu, and J. Choi, "Environmental monitoring system for intelligent stations," *Concurrency Comput., Pract. Exper.*, vol. 33, no. 2, p. e5131, Jan. 2021, doi: 10.1002/cpe.5131.

[10] Y. Jiang, J. Dong, X. Qi, and F. Wang, "Improvement of monitoring technology for corrosive pollution of marine environment under cloud computing platform," *Coatings*, vol. 12, no. 7, p. 938, Jul. 2022, doi: 10.3390/coatings12070938.
[11] B. Zhong, J. Guo, L. Zhang, H. Wu, H. Li, and Y. Wang, "A blockchain-based framework for on-site construction environmental monitoring: Proof of concept," *Building Environ.*, vol. 217, Jun. 2022, Art. no. 109064, doi: 10.1016/j.buildenv.2022.109064.
[12] G. Song, Y. Lu, H. Feng, H. Lin, and Y. Zheng, "An implementation framework of blockchain-based hazardous waste transfer management system," *Environ. Sci. Pollut. Res.*, vol. 29, no. 24, pp. 36147–36160, May 2022, doi: 10.1007/s11356-021-17489-0.
[13] M. Kassou, S. Bourekkadi, S. Khoulji, K. Slimani, H. Chikri, and M. L. Kerkeb, "Blockchain-based medical and water waste management conception," in *Proc. E3S Web Conf.*, vol. 234, Feb. 2021, p. 70, doi: 10.1051/e3sconf/202123400070.
[14] Bitshares. *Delegated Proof of Stake*. Accessed: 2019. [Online]. Available: https://docs.bitshares.org/en/master/technology/dpos.html
[15] J. Shah and B. Mishra, "IoT-enabled low power environment monitoring system for prediction of PM2.5," *Pervas. Mobile Comput.*, vol. 67, Sep. 2020, Art. no. 101175, doi: 10.1016/j.pmcj.2020.101175.
[16] C. B. D. Kuncoro, "Automatic wireless ambient air and weather condition monitoring system for outdoor environment monitoring applications," *Sensors Mater.*, vol. 32, no. 1, pp. 337–356, 2020.
[17] J. A. D. Donet, C. Pérez-Solà, and J. Herrera-Joancomartí, "The Bitcoin P2P network," *Financial Cryptogr. Data Secur.*, vol. 8438, pp. 87–102, Oct. 2014, doi: 10.1007/978-3-662-44774-1_7.
[18] J. Wang, J. Chen, Y. Ren, P. K. Sharma, O. Alfarraj, and A. Tolba, "Data security storage mechanism based on blockchain industrial Internet of Things," *Comput. Ind. Eng.*, vol. 164, Feb. 2022, Art. no. 107903, doi: 10.1016/j.cie.2021.107903.
[19] Z. Pang, Y. Yao, Q. Li, X. Zhang, and J. Zhang, "Electronic health records sharing model based on blockchain with checkable state PBFT consensus algorithm," *IEEE Access*, vol. 10, pp. 87803–87815, 2022, doi: 10.1109/ACCESS.2022.3186682.
[20] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, "A scalable multi-layer PBFT consensus for blockchain," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 5, pp. 1146–1160, May 2021, doi: 10.1109/TPDS.2020.3042392.
[21] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, and E. Wong, "Zyzzyva: Speculative Byzantine fault tolerance," in *Proc. 21st ACM SIGOPS Symp. Operating Syst. Princ.*, 2007, pp. 45–58, doi: 10.1145/1294261.1294267.
[22] J. Liu, W. Li, G. O. Karame, and N. Asokan, "Scalable Byzantine consensus via hardware-assisted secret sharing," *IEEE Trans. Comput.*, vol. 68, no. 1, pp. 139–151, Jan. 2019, doi: 10.1109/TC.2018.2860009.
[23] X. Yang, T. Li, X. Pei, L. Wen, and C. Wang, "Medical data sharing scheme based on attribute cryptosystem and blockchain technology," *IEEE Access*, vol. 8, pp. 45468–45476, 2020, doi: 10.1109/ACCESS.2020.2976894.
[24] Y. Ren, Y. Leng, J. Qi, P. K. Sharma, J. Wang, Z. Almakhadmeh, and A. Tolba, "Multiple cloud storage mechanism based on blockchain in smart homes," *Future Gener. Comput. Syst.*, vol. 115, pp. 304–313, Feb. 2021, doi: 10.1016/j.future.2020.09.019.
[25] R. Awadallah, A. Samsudin, J. S. Teh, and M. Almazrooie, "An integrated architecture for maintaining security in cloud computing based on blockchain," *IEEE Access*, vol. 9, pp. 69513–69526, 2021, doi: 10.1109/ACCESS.2021.3077123.
[26] L. Zhu, Y. Wu, K. Gai, and K.-K. R. Choo, "Controllable and trustworthy blockchain-based cloud data management," *Future Gener. Comput. Syst.*, vol. 91, pp. 527–535, Feb. 2019, doi: 10.1016/j.future.2018.09.019.
[27] J. Cha, S. K. Singh, T. W. Kim, and J. H. Park, "Blockchain-empowered cloud architecture based on secret sharing for smart city," *J. Inf. Secur. Appl.*, vol. 57, Mar. 2021, Art. no. 102686, doi: 10.1016/j.jisa.2020.102686.
[28] S. Fu, C. Zhang, and W. Ao, "Searchable encryption scheme for multiple cloud storage using double-layer blockchain," *Concurrency Comput., Pract. Exper.*, vol. 34, no. 16, p. e5860, Jun. 2020, doi: 10.1002/cpe.5860.
[29] B. Sowmiya, E. Poovammal, K. Ramana, S. Singh, and B. Yoon, "Linear elliptical curve digital signature (LECDS) with blockchain approach for enhanced security on cloud server," *IEEE Access*, vol. 9, pp. 138245–138253, 2021, doi: 10.1109/ACCESS.2021.3115238.
[30] J. Hur, "Attribute-based secure data sharing with hidden policies in smart grid," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 11, pp. 2171–2180, Nov. 2013, doi: 10.1109/TPDS.2012.61.
[31] Z. Ullah, B. Raza, H. Shah, S. Khan, and A. Waheed, "Towards blockchain-based secure storage and trusted data sharing scheme for IoT environment," *IEEE Access*, vol. 10, pp. 36978–36994, 2022, doi: 10.1109/ACCESS.2022.3164081.
[32] S. Peng, Z. Cai, W. Liu, W. Wang, G. Li, Y. Sun, and L. Zhu, "Blockchain data secure transmission method based on homomorphic encryption," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–9, Apr. 2022, doi: 10.1155/2022/3406228.
[33] H. Wang and J. Zhang, "Blockchain based data integrity verification for large-scale IoT data," *IEEE Access*, vol. 7, pp. 164996–165006, 2019, doi: 10.1109/ACCESS.2019.2952635.
[34] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed: 2008. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf
[35] S. King and S. Nadal, "PPCoin: Peer-to-peer crypto-currency with proof-of-stake," *Self-Published Paper*, vol. 19, no. 1, pp. 1–6, Aug. 2012.
[36] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," *ACM Trans. Comput. Syst.*, vol. 99, no. 1999, pp. 173–186, 2002, doi: 10.1145/571637.571640.
[37] M. Conti, S. Kumar, C. Lal, and S. Ruj, "A survey on security and privacy issues of Bitcoin," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3416–3452, Apr. 2018, doi: 10.1109/COMST.2018.2842460.
[38] G. Xu, Y. Liu, and P. W. Khan, "Improvement of the DPoS consensus mechanism in blockchain based on vague sets," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4252–4259, Jun. 2020, doi: 10.1109/TII.2019.2955719.
[39] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Conf. USENIX Annu. Tech. Conf.*, May 2014, pp. 305–319.

**MANYU ZHAO** received the B.E. degree from Shandong Technology and Business University, in 2016, where she is currently pursuing the master's degree. Her research interests include environmental monitoring and blockchain application.

**WEI LIU** received the Ph.D. degree in control theory and control engineering from the China University of Mining and Technology, Beijing, in 2008. He is currently an Associate Professor with Shandong Technology and Business University. His research interests include signal processing, computer monitoring, and sensor technology.

**KAI HE** received the Ph.D. degree in information technology from Nankai University, Tianjin, in 2006. He is currently an Associate Professor with Shandong Technology and Business University. His research interests include environmental monitoring, distributed computing, blockchain application, and network storage.