Hindawi
Shock and Vibration
Volume 2020, Article ID 8846429, 9 pages
https://doi.org/10.1155/2020/8846429

Research Article

Consensus Mechanism of IoT Based on Blockchain Technology

Yue Wu,[1,2] **Liangtu Song**,[1,2] **Lei Liu**,[1,2] **Jincheng Li**,[1,2] **Xuefei Li**,[1,2] **and Linli Zhou**[1,2]

[1]*Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, Anhui, China*
[2]*University of Science and Technology of China, Hefei 230026, Anhui, China*

Correspondence should be addressed to Yue Wu; wyw533k@mail.ustc.edu.cn

Received 10 August 2020; Revised 27 August 2020; Accepted 21 September 2020; Published 7 October 2020

Academic Editor: Chaoqun Duan

Applying blockchain technology to the Internet of Things (IoT) remains a huge challenge. To meet the actual needs of IoT, a lightweight and high-throughput consensus mechanism, combined with blockchain technology, is proposed in this study. Blockchain nodes use the Diffie–Hellman algorithm for key negotiation. Sensors and blockchain nodes can use the shared key to generate HMAC (Hash-based Message Authentication Code) signatures for sensor-aware transactions and use the Verifiable Random Function to implement block nodes. Offline fast election, which is the node that wins the election, becomes the block node. Machine learning methods are also introduced to identify or remove outliers in the sensor data before such data are uploaded to the chain. Experimental results show that the system throughput synchronously increases as the test load increases. Moreover, when the test load is 800 tps, the system throughput reaches the maximum, close to 600 tps. When the test load exceeds 800 tps, the actual system throughput starts to drop, and approximately 90% of transactions have a delay time within 5000 ms. This method can be used in a lightweight IoT system.

1. Introduction

With the continuous development and progress of technologies such as sensor technology, computer control technology, embedded technology, and wireless network data communication, the Internet of Things (IoT) has shown great development worldwide [1]. IoT is considered the third world information industry wave after the computer and the Internet and is developing rapidly. By 2020, more than 50 billion smart devices will be connected to the network [2]. Distributed processors and communication hardware send/receive data from the environment, thus generating huge data. However, at present, most IoT architectures require a central hub or server, which allows data storage and transmission between several devices in the network space. These smart devices use sensors, embedded processors, and communication hardware to send/receive data from the environment, thereby generating huge amounts of data. Therefore, security and privacy have become the greatest challenges of IoT [3].

According to the characteristics of the IoT platform, blockchain technology can solve the security and management problems that a large amount of smart device data can appear in a centralized system framework [4]. Blockchain is a new application model similar to an integrated distributed database, consensus mechanism, peer-to-peer (P2P) transmission, and asymmetric encryption algorithm [5]. Blockchain has the characteristics of decentralization, openness, autonomy, anonymity, and information tampering. The use of blockchain technology for IoT has attracted increasing attention in this direction. Literature [6] constructed a distributed data management system on the basis of blockchain and trusted execution environment. The system stores the encrypted hash value in the blockchain and stores the original encrypted data in a trusted execution environment. The system also provides corresponding data access control strategies through smart contracts to ensure the security, privacy, and integrity of the IoT dataset. Literature [7, 8] proposed a lightweight architecture on the basis of blockchain for IoT. Such an architecture maintains security and privacy while reducing blockchain overhead. IoT devices combine private ledger with centralized management similar to blockchain to optimize energy consumption. High-resource devices create an overlay network to implement a publicly accessible distributed blockchain,

ensure end-to-end privacy and security, and use distributed authentication to reduce block verification processing time, which is ultimately implemented in smart home applications. Literature [9] proposed a new role and authority arbitration structure in IoT that is a fully distributed access control system for IoT based on blockchain technology and is evaluated in a real IoT scenario. The results provided in this article indicate that, in specific scalable IoT scenarios, blockchain technology can be used as an access control technology to enhance security. Literature [10, 11] evaluated the role of blockchain in strengthening network security and protecting privacy. Using practical applications and practical examples, the distributed sensitivity of the blockchain likely causes attack sensitivity problems. Ways to solve problems related to blockchain-based identity and access control system are also discussed. In addition, certain key challenges related to IoT security are put forward. However, the current mainstream blockchain implementation is mainly aimed at digital currency applications, such as Bitcoin and Ethereum [12]. These blockchains are directly used in IoT and have the following problems:

(i) The computing power and storage resource of IoT devices are generally low, making the performance of complex cryptographic calculations difficult [13]. Mainstream blockchain platforms usually use computationally intensive asymmetric key technology as user identification and transaction verification mechanism; one example is the secp256k1 elliptic curve asymmetric key pair used in Bitcoin and Ethereum [14], which requires more computing power than what most IoT devices can provide. Taking the common IoT hardware platform Arduino Uno (CPU frequency: 16 MHZ) as an example, performing an elliptic curve private key signature and an elliptic curve public key verification bell takes more than five and eight seconds, respectively [15]. Therefore, an appropriate cryptographic security mechanism must be selected to allow IoT devices to access the blockchain and ensure a certain degree of security [16].

(ii) IoT requires high data throughput. At present, mainstream blockchain technology still cannot easily support mainstream public chains that use digital currency applications as the main target scenarios, such as Bitcoin or Ethereum. The primary purpose of consensus is to avoid double-spending [17]; that is, a coin is consumed multiple times so that the blockchain can meet the needs of certain users at low throughput. For example, the throughput of Bitcoin is 7 tps [18], whereas that of Ethereum is 15 tps. Considering that the sampling period of sensors in IoT is usually in the order of seconds or even milliseconds and a large number [19], which means that the data on IoT devices have high-throughput requirements, a consensus mechanism suitable for IoT with high throughput should be designed [20].

(iii) IoT can improve the credibility of data on the chain, but it lacks effective safeguards. Although the blockchain can guarantee that such data cannot be tampered with, tampering with the data source is powerless. It affects the trustworthiness of data on the blockchain. IoT can be directly uploaded to the chain through sensor data, thereby maximizing the credibility of the data on the chain, which puts high requirements on the confirmation of transactions in the blockchain consensus algorithm to identify sensor data.

The research on this topic is applicable to the blockchain consensus mechanism of IoT. The main contents include the research on cryptographic algorithms and mechanisms suitable for lightweight IoT devices to access the blockchain. We attempt to avoid excessive communication overhead, and we introduce machine learning methods to identify or remove outliers in sensor data before data are uploaded to the chain.

2. System Model

As shown in Figure 1, the system model is divided into four layers, from bottom to top.

(i) Perception layer is mainly composed of lightweight IoT devices, such as sensors or drivers, and adjacent IoT devices are connected to IoT gateways.

(ii) Gateway layer mainly comprises IoT gateways, and each gateway is responsible for the access of the part of sensor/driver devices. Gateways are responsible for identifying the abnormal sensor data and submitting the processed transactions to the blockchain nodes in the network layer.

(iii) Network layer is mainly composed of blockchain nodes which form P2P networks with each other. Blockchain nodes are responsible for caching transactions (from the gateway or upper application), selecting master nodes, identifying blocks, and resolving the chain fork conflict.

(iv) Application layer mainly comprises various decentralized applications (DApps) based on blockchain. DApp accesses the data on the chain through blockchain nodes or submits transactions to the chain.

Compared with the traditional Internet of Things system model, the blockchain technology is not introduced in the network layer, but the centralized server is adopted as the core of the system. The upward application layer and the downward gateway layer directly interact with the centralized server for data. When data is collected continuously by multiple different devices on the network, the amount of data is very large. The traditional Internet of Things system is managed by a central service manager, so the system will be responsible for thousands of devices. In this way, for such a large amount of data, the system must have a storage device

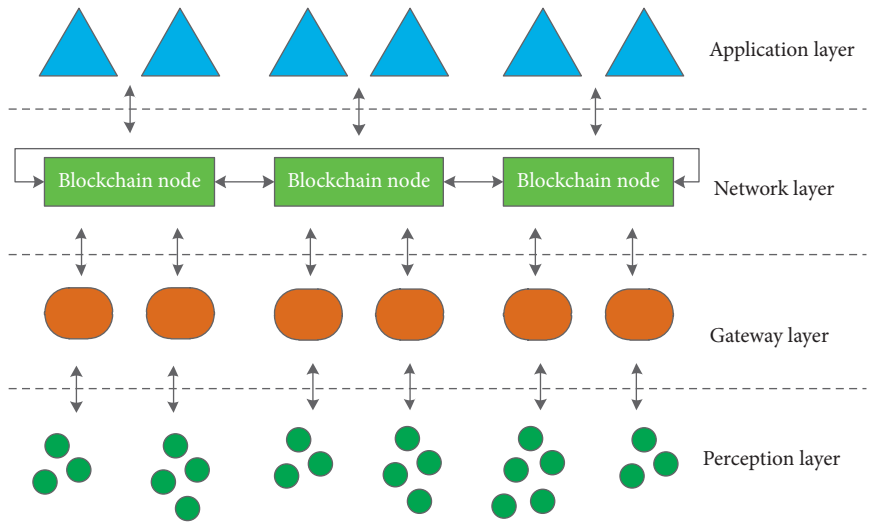

Figure 1: System model of IoT based on blockchain.

that can handle all the data, a powerful data processing system, and an efficient mobile phone and transmission system. After the introduction of the blockchain in the network layer, there is no central server and each node has its own copy of data, which can be processed independently. After the system obtains the transmission parameters, it compares the other nodes in the network with the transmission parameters. If they are in agreement, then, they will reach a consensus and can deal with trading blockchain technology to solve this problem to some extent. In addition, once the central server is hacked, all the data stored in the central server will be destroyed. Blockchain networks maximize their strengths and avoid their weaknesses. All data is encrypted and stored in the network, and only participating nodes can access it. And the nodes of network verification must be consistent to ensure that the data will not be maliciously modified.

The overall flow of the algorithm is shown in Figure 2.

After the system starts, the sensor first requests access to the IoT gateway, and the sensor periodically uploads data to the gateway. At the same time, after the sensor is started, it uses the Diffie–Hellman algorithm for key negotiation with the blockchain node and uses the shared key to sense transactions for the sensor to generate HMAC signature. After the HMAC signature verification of the sensor data and the abnormal data is removed, the IoT gateway starts to submit ordinary transactions. To avoid the computational overhead in PoW consensus and the communication overhead in BFT consensus, this study adopts Verifiable Random Function (VRF) to realize the offline fast election of block nodes. The node that wins the election becomes the block producer. Then, the dominant node packs the transaction and produces the block. When there are multiple legal block-producing nodes, multiple legal blocks will appear at the same height; that is, bifurcation occurs. The consensus algorithm in this article uses the longest chain rule to resolve fork conflicts; that is, the chain with the largest length is regarded as the optimal chain.

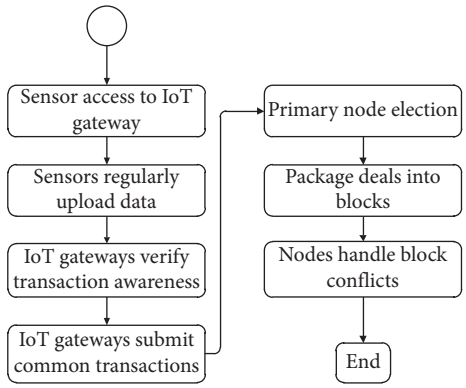

Figure 2: Algorithm process.

As shown in Figure 2, the system has two types of transactions: sensor-aware transactions that are submitted to gateways, and normal transactions that occur on blockchain nodes or upper-level applications. Considering the performance difference between IoT terminal devices and node or application hosts, we design different verification mechanisms for both transactions.

(i) Transaction aware: from sensors to gateways, the symmetric encryption algorithm is used for transaction verification.

(ii) Ordinary transactions: the asymmetric encryption algorithm is used for transaction verification.

Gateways use machine learning models to detect the abnormality in the sensor perception transaction, attach the detection result and reencapsulate it, and broadcast it to the blockchain network. After receiving and verifying the broadcast transaction, blockchain nodes temporarily store it in the pending transaction pool and wait for block confirmation. We divide the continuous time into consensus rounds at equal intervals and blockchain nodes package pending transactions, produce blocks, and broadcast to the blockchain network in rounds. To improve throughput, in a

round, certain nodes are randomly selected as dominant nodes, and only the selected dominant nodes can produce blocks. Given that multiple leading nodes may be selected in a round, nodes may receive blocks of the same height broadcast from multiple leading nodes. We use the longest chain rule to solve the blockchain fork conflict.

3. Consensus Mechanism

3.1. Key Agreement Protocol. As shown in Figure 3, after sensors are started, the Diffie–Hellman algorithm is used for key negotiation with the blockchain nodes.

(1) Sensors and blockchain nodes share prime number p and base g.

(2) Sensors first select private key a and, then, send public key A's a to blockchain nodes:

$$A = g^a \bmod p. \tag{1}$$

(3) Nodes first select a private key b and, then, send b's public key B to sensor nodes:

$$B = g^b \bmod p. \tag{2}$$

(4) Sensors calculate the shared key:

$$K_{\text{sensor}} = B^a \bmod p. \tag{3}$$

(5) Nodes calculate the shared key:

$$K_{bc-\text{node}} = A^b \bmod p. \tag{4}$$

(6) Considering that $K_{\text{sensor}} = K_{bc-\text{node}}$, the shared key can be used by sensors and blockchain nodes to generate HMAC signatures for sensor-aware transactions.

3.2. Sensor Transaction Verification. After nodes receive sensor transactions, the verification process is divided into two steps:

(i) Verify the HMAC signatures of sensor data

(ii) Identify and label abnormal sensor data

3.2.1. Sensor Data Signature Verification. As shown in Table 1, the purpose of MAC, the message verification code, is to verify the data source and its integrity. In this article, we use an MAC based on a cryptographic hash function, or HMAC. HMAC requires a shared key between a sender and a receiver. For the specified data and key K, the HMAC calculation formula is as follows:

$$\text{HMAC}(K, \text{data}) = H\Big(\big(K_0 \oplus \text{opad}\big) \| H\Big(\big(K_0 \oplus \text{ipad}\big) \| \text{data}\Big)\Big). \tag{5}$$

(i) H: selected hash function,

(ii) K_0: key K preprocessed result,

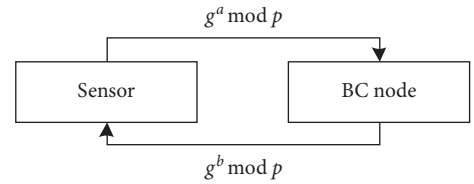

Figure 3: Sensor and node key agreement protocol.

Table 1: Sensor transaction structure.

#	Field	Type	Explanation
1	Sender	String	Sensor ID
2	Time	Int	Send time, Linux timestamp
3	Value	Float	Sensor sampling data
4	MAC	String	MAC code

(iii) opad: complete data outside, bytes of hash block length $0 \times 5c$,

(iv) ipad: fill in the data, the length of the hash block byte 0×36,

(v) : splicing operation symbol,

(vi) $\oplus$: XOR operation symbol.

Sensor perception transaction adopts the following structural definitions:

Before sensors calculate the HMAC codes of transaction structures, other parts, except the HMAC field, must first be serialized into data, and, then, the HMAC calculation can be performed. The algorithm for the serialized data of sensor transaction tx is as follows:

$$\text{data} = \text{bytes}(tx.\text{sender}) \| \text{bytes}(tx.\text{time}) \| \text{bytes}(tx.\text{value}). \tag{6}$$

(i) bytes: convert parameters to bytecode stream,

(ii) : splicing two streams.

3.2.2. Sensor Abnormal Data Detection

(1) Sensor Data Fusion. First is the sensor data fusion. We assume that the data from sensor i in time slot t are $d_{ti} \in R$, and the number of sensors connected to the same blockchain node is n. The sensor data $D_t = [d_1, \ldots d_n]$ accessed by this blockchain node in time slot t thus constitute the state of the external environment at time t, taking the data of k continuous time slots to form $D = [D_1, D_2, \ldots D_k]^T$, which is a $k \times n$ in the matrix. Each row represents the state of the external environment at a time, and each column represents the time series of a sensor. For example, the following figure shows matrix D which is formed by the data of six sensors in three time slots, which is shown in Figure 4.

According to the different characteristics of sensor data, we use two unsupervised machine learning algorithms to detect abnormal data in the sensor fusion data represented by the $k \times n$ matrix.

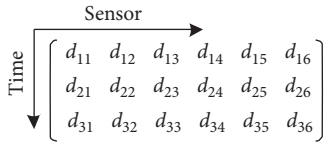

Figure 4: Time slot matrix.

(2) HBOS Abnormal Data Detection Algorithm. If the data of n sensors connected to a gateway is irrelevant, then, we can use the HBOS algorithm to calculate the anomaly score of each time slot sample using the statistical characteristics of the sensor data. The HBOS algorithm process is as follows:

(i) Use the time series data of a single sensor to calculate its histogram, count the number of occurrences of each value (domain), and calculate its relative frequency.

(ii) Normalize the histogram of each sensor, so that the maximum frequency is 1, to ensure that different sensors have the same weight on the final abnormality score.

(iii) Use the following formula to calculate the anomaly score of the sensor fusion data for each time slot:

$$\mathrm{HBOS}_k = \sum_{i=1}^{n} \log\left(\frac{1}{\mathrm{hist}_i(k)}\right), \tag{7}$$

where $\mathrm{hist}_i(k)$ represents the density estimation of the k-th time slot data of the i-th sensor in the statistical histogram of the sensor and HBOS_k represents the anomaly score of the sensor fusion data of the k-th time slot. The HBOS algorithm assumes that the sensor data are uncorrelated. Therefore, the above HBOS score calculation formula is actually the application of the Naive Bayes probability model (Naive Bayes Model) in the log domain.

(3) Autoencoder Abnormal Data Detection Algorithm. If the sensor data are related, then the unsupervised deep learning model Autoencoder can be used to detect abnormal sensor fusion data.

As shown in Figure 5, autoencoder is a specially constructed neural network structure. By setting the neural network output as input, training data without annotation are obtained. The input data can be simplified/lessened after training Rank representation (Code) through certain regularization conditions. In the sensor data anomaly detection algorithm based on autoencoder, we mainly use the output and input residuals to determine the anomaly score of each sample. The algorithm is briefly described as follows:

(i) Construct the autoencoder model, the numbers of output layers, and output layer neurons to be n.

(ii) Train with sensor data D.

(iii) Calculate the anomaly score of each sample, where d represents the reconstructed data of the i-th sensor:

$$S_k = \sum_{i=1}^{n} \left(d_i - \widehat{d}_l\right)^2. \tag{8}$$

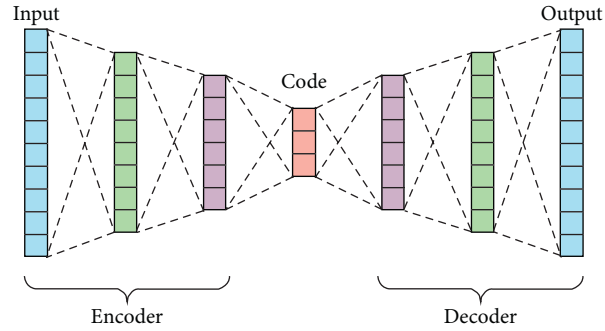

Figure 5: Autoencoder neural network.

3.3. Block Node Selection. To avoid the computational overhead in PoW consensus and the communication overhead in BFT consensus, this study adopts Verifiable Random Function (VRF) to realize the offline fast election of block nodes. VRF is a cryptographic hash of the public key version. Only the private key holder can calculate the hash, but any participant who knows the public key can verify the correctness of the hash.

The algorithm flow is summarized as follows:

(1) The private key holder uses the private key SK and the public input data alpha to calculate the hash beta and evidence pi:

$$\mathrm{beta} = \mathrm{vrf\_proof2hash}(\mathrm{pi}). \tag{9}$$

(2) The verifier uses the hash provider's public key PK, evidence pi, and input data alpha to recalculate hash beta_2. If it matches the provider, the hash is correct:

$$\mathrm{beta}_2 = \mathrm{vrf\_verify}(\mathrm{PK}, \mathrm{alpha}, \mathrm{pi}). \tag{10}$$

In this study, the consensus algorithm uses a round consensus mechanism, which divides the time into rounds of fixed length. In each round, VRF is used to determine whether the current node of the current round is selected as the block-generation node, and if so, the block is broadcasted. VRF requires all parties involved to hold a key pair, and the block selection algorithm flow of each round is as follows:

(1) Calculate the shared information alpha of the current round, where t represents the current time and T represents the duration of the round:

$$\mathrm{alpha} = \frac{t}{T}. \tag{11}$$

(2) The node uses its private key and shared information alpha to calculate the hash beta and evidence pi:

$$\begin{aligned} \mathrm{pi} &= \mathrm{vrf\_prove}(\mathrm{SK}, \mathrm{alpha}), \\ \mathrm{beta} &= \mathrm{vrf\_proof2hash}(\mathrm{pi}). \end{aligned} \tag{12}$$

Assuming that the probability of winning a node in an election is p, and the selection experiment for each node n times is repeated, the probability X_k of finally

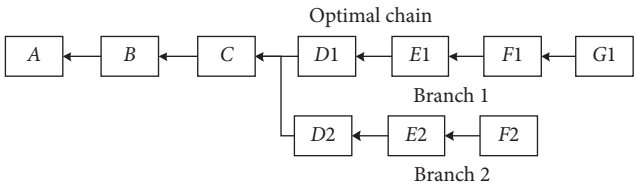

Figure 6: Longest chain rule.

selecting k nodes conforms to the binomial distribution, namely,

$$X_k \sim b(n, p). \tag{13}$$

Calculate the corresponding k when X_k is the following value, where hashlen represents the bit length of hash beta:

$$p_{\text{node}} = \frac{\text{beta}}{2^{\text{hashlen}}}. \tag{14}$$

If $k > 0$, then the node is selected.

(3) The selected block node broadcasts the block, with its public key and evidence.

(4) After receiving the block, the other nodes calculate the alpha value of the current round, use the public key PK of the block-producing node in the block and the evidence pi to calculate the hash, and perform Step 3. After verifying that the node is indeed a block-producing node, change the area. The block is also added to the local chain.

3.4. Chain Conflict Resolution. The aforementioned algorithm for selecting a block-generation node cannot guarantee that only one block-generation node is selected in the same round. When multiple legal block-producing nodes exist, multiple legal blocks can appear at the same height, that is, a fork. The consensus algorithm in this study uses the longest chain rule to solve the bifurcation conflict; that is, the chain with the maximum length is regarded as the optimal chain.

As shown in Figure 6, the development at height D has a fork, but the length of Branch 1 exceeds that of Branch 2. Thus, Branch 1 is regarded as the optimal chain.

4. Results

4.1. A Transaction Delay Time. Transaction delay time represents the time it takes for transactions to be submitted to the block confirmation. From the data sent by sensors to the final transactions entering blocks, the total delay time can be divided into two sections.

$$\text{Latency} = L_1 + L_2, \tag{15}$$

where L_1 represents the time from sensor submission to gateway detection completion and L_2 represents the time from gateway submission transaction to block packaging. The following figure shows the schematic structure of the

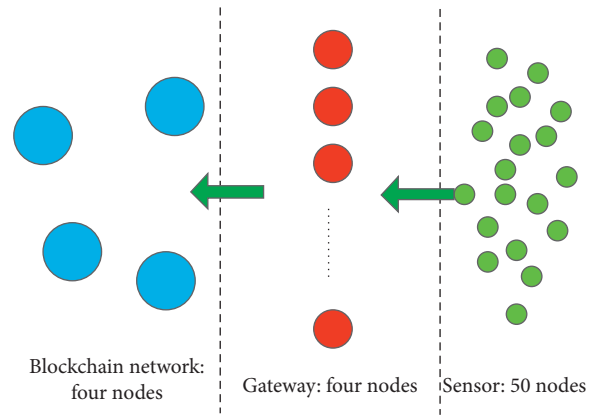

Figure 7: Transaction delay structure diagram.

transaction delay time simulation experiment, which is shown in Figure 7.

The histogram of transaction delay statistics is shown in Figure 8.

Approximately 90% of transactions have a delay time within 5000 ms.

Figures 9 and 10 show the proportion of the total delay in the L_1 and L_2 stages (from sensors to gateways).

The average proportion of the overall delay from the sensor to the gateway confirmation stage is approximately 10%. Figure 11 displays a comparison chart of the two-stage delay.

The transaction delay in the first phase remains basically unchanged, whereas that in the second phase exhibits an obvious periodic characteristic. The reason is that the data submission of sensors is periodic; thus, the transaction submission of gateways also remains basically periodic. When the moment in which gateways submit transactions is close to the next block-generation round, only a short transaction delay can be confirmed; otherwise, a large transaction delay time is required.

4.2. Throughput. Throughput examines the total amount of blockchain transaction processing per unit time. The simulation experiment parameters are selected as follows:

(i) Blockchain nodes: four.

(ii) Workload node: 10 tps/node, 10–100, corresponding to 100–1000 input tps.

(iii) Leader node election: Vrf-PoW.

The schematic of the simulation experiment is shown in Figure 12.

The experimental results are shown in Figure 13.

Within a certain range (test load <800 tps), the system throughput synchronously increases as the test load increases, and when the test load is 800 tps, the system throughput reaches the maximum which is close to 600 tps. When the test load exceeds 800 tps, the actual system throughput begins to decrease, indicating that the simulation system has been overloaded.

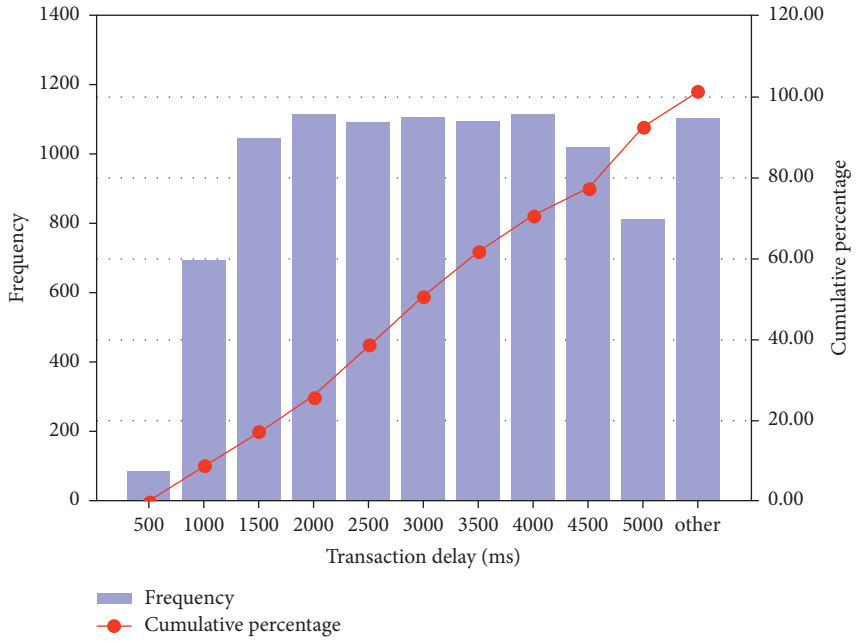

Figure 8: Transaction delay analysis.

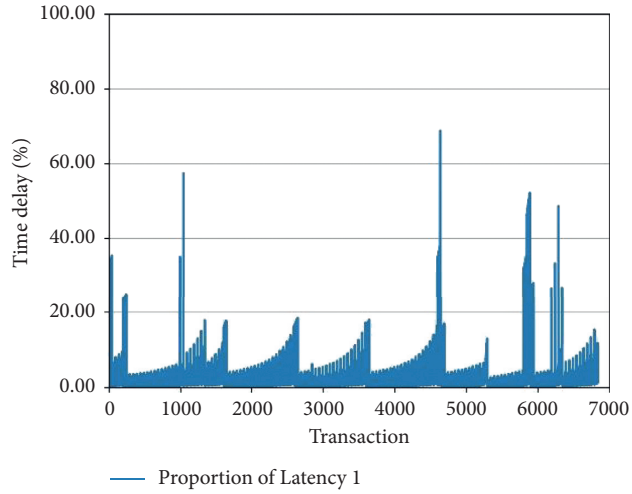

Figure 9: Proportion of Latency 1.

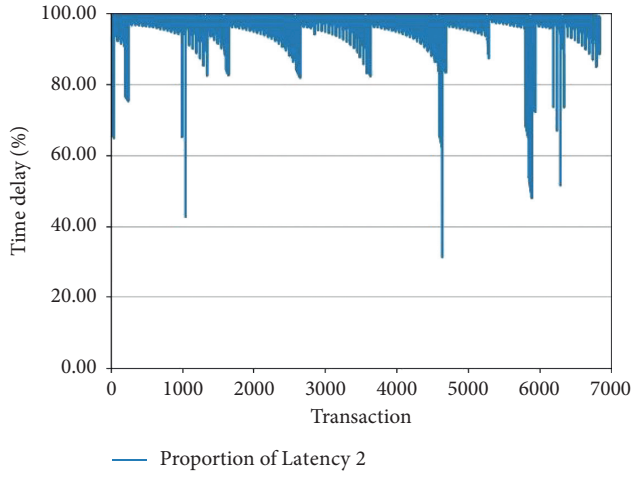

Figure 10: Proportion of Latency 2.

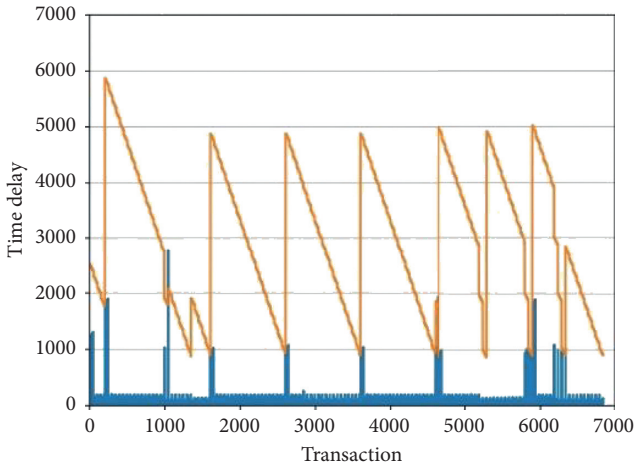

Figure 11: Comparison of transaction delay between two stages.

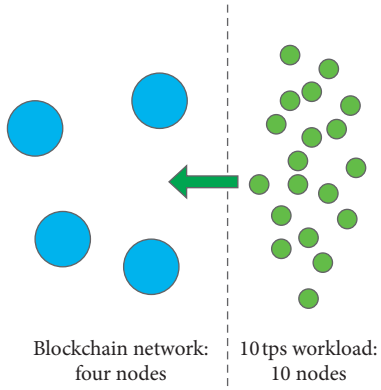

Figure 12: Throughput structure diagram.

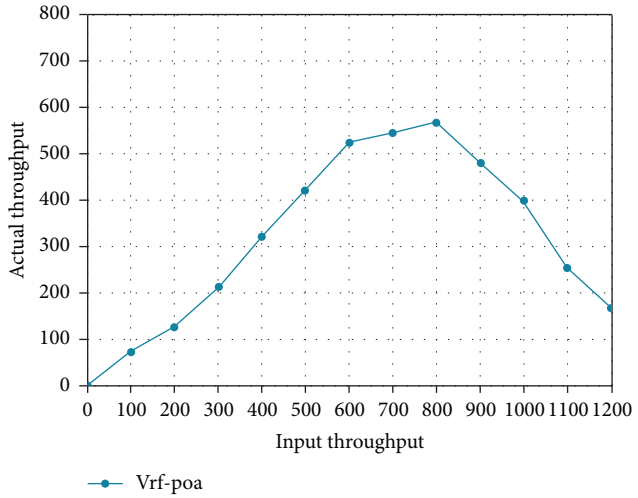

Figure 13: Throughput analysis.

5. Conclusion

Based on the blockchain theory and IoT system architecture, this study proposes a blockchain consensus mechanism suitable for lightweight IoT devices. The Diffie–Hellman algorithm is used for key negotiation with blockchain nodes, sensors, and zones. Blockchain nodes can use this shared key to generate HMAC signatures for sensor-aware transactions, including VRF, to achieve the offline fast election of block nodes. The winning node becomes the block node. Machine learning means are also introduced to identify or eliminate outliers in sensor data before the data are uploaded.

We believe that continuous research on blockchain and IoT can bring about major changes in the industry and promote continuous social development.

Data Availability

The data that support the findings of the research are available from the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This study was supported by Key Research and Development Plan of Anhui Province (201904a06020056).

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (IoT): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] M. A. Khan and K. Salah, "IoT security: review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[3] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[4] S. K. T. Mehedi, A. A. M. Shamim, and M. B. A. Miah, "Blockchain-based security management of IoT infrastructure with Ethereum transactions," *Iran Journal of Computer Science*, vol. 2, no. 3, pp. 189–195, 2019.

[5] J. Wu and N. Tran, "Application of blockchain technology in sustainable energy systems: an overview," *Sustainability*, vol. 10, no. 9, p. 3067, 2018.

[6] G. Ayoade, V. Karande, L. Khan, and K. Hamlen, "Decentralized IoT data management using blockchain and trusted execution environment," in *Proceedings of IEEE International Conference on Information Reuse and Integration (IRI)*, pp. 15–22, IEEE, Salt Lake City, UT, USA, July 2018.

[7] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an optimized blockchain for IoT," in *Proceedings of 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, IEEE, Pittsburgh, PA, USA, pp. 173–178, April 2017.

[8] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Lsb: a lightweight scalable blockchain for iot security and privacy," 2017, https://arxiv.org/abs/1712.02969.

[9] O. Novo, "Blockchain meets IoT: an architecture for scalable access management in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.

[10] N. Kshetri, "Blockchain's roles in strengthening cybersecurity and protecting privacy," *Telecommunications Policy*, vol. 41, no. 10, pp. 1027–1038, 2017.

[11] A. Panarello, N. Tapas, G. Merlino, F. Longo, and A. Puliafito, "Blockchain and iot integration: a systematic survey," *Sensors*, vol. 18, no. 8, p. 2575, 2018.

[12] D. Vujičić, D. Jagodić, and S. Ranđić, "Blockchain technology, bitcoin, and ethereum: a brief overview," in *Proceedings of 17th International Symposium Infoteh-Jahorina (Infoteh)*, IEEE, East Sarajevo, Bosnia and Herzegovina, pp. 1–6, March 2018.

[13] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, and D. Qiu, "Security of the internet of things: perspectives and challenges," *Wireless Networks*, vol. 20, no. 8, pp. 2481–2501, 2014.

[14] H. Mayer, "ECDSA security in bitcoin and ethereum: a research survey," *CoinFaabrik*, vol. 28, p. 126, 2016.

[15] D. Mahto, D. A. Khan, and D. K. Yadav, "Security analysis of elliptic curve cryptography and RSA," in *Proceedings of the world congress on engineering*, vol. 1, pp. 419–422, London, UK, July 2016.

[16] L. Zhou, L. Wang, Y. Sun, and P. Lv, "Beekeeper: a blockchain-based iot system with secure storage and homomorphic computation," *IEEE Access*, vol. 6, pp. 43472–43488, 2018.

[17] J. L. Zhao, S. Fan, and J. Yan, *Overview of Business Innovations and Research Opportunities in Blockchain and Introduction to the Special Issue*, Springer Open, New York, NY, USA, 2016.

[18] A. Kiayias and G. Panagiotakos, "Speed-security tradeoffs in blockchain protocols," *IACR Cryptology ePrint Archive*, vol. 1019, 2015.

[19] D. Guinard, V. Trifa, and E. Wilde, "A resource oriented architecture for the web of things," in *Proceedings of 2010 Internet of Things (IOT)*, pp. 1–8, IEEE, Tokyo, Japan, December 2010.

[20] Q. He, N. Guan, M. Lv, and W. Yi, "On the consensus mechanisms of blockchain/dlt for internet of things," in *Proceedings of 2018 IEEE 13th International Symposium on Industrial Embedded Systems (SIES)*, IEEE, Graz, Austria, pp. 1–10, June 2018.