Journal of Computers Vol. 34 No. 3, June 2023, pp. 251-271
doi: 10.53106/199115992023063403018

An Efficient and Reliable Blockchain-based Trust Management Model for Electricity Trading Terminal

Zhenzhen Liu[1], Rui Zhou[1], Kangqian Huang[1], Xin Hu[1],
Kaiyang Deng[2], Binsi Cai[2*], Kaiguo Yuan[2]

[1] Information Data Department, Guangdong Power Exchange Center Co. Ltd.,
Guangdong 510080, China

clz0502@163.com, {zhourui, huangkangqian}@gd.csg.cn,278271099@qq.com

[2] A Key Laboratory of Trustworthy Distributed Computing and Service, Beijing University of Posts and Telecommunications,
Beijing 100876, China

{dengkaiyang, caibinsi, flyingdreaming}@bupt.edu.cn

Received 1 March 2023; Revised 1 April 2023; Accepted 17 May 2023

Abstract. A safe and reliable terminal environment is crucial to ensure the security of the electricity trading system. The existing terminal security system based on identity authentication and access control has internal threats that are difficult to solve. For example, multiple internal malicious nodes cause broadcast message tampering attacks and malicious packet loss, resulting in message dissemination failure. Existing blockchain-based trust management systems are good for addressing insider threats, but suffer from low efficiency. In order to solve the internal threat problem of the electric electricity trading system, from the perspective of trust evaluation and trust management of the terminal environment, an efficient and reliable blockchain-based electric trading terminal (ERBTM) trust management model is proposed. First of all, we collect a variety of trust factors to evaluate the credibility of the terminal, which solves the problem of accuracy in the process of trust assessment; secondly, we improve the speed of storing trust values on the chain and ensure the robustness of the system by improving the consensus algorithm; Finally, we designed the structure of the block that stores the trust value to ensure that the trust value is not tampered with. The experimental results show that, compared with similar methods, the ERBTM model can effectively deal with the endogenous security threats of terminals in the electricity trading environment, and has significant advantages in terms of efficiency and reliability.

Keywords: insider threat, trust management, electricity trading terminal, Trusted Practical Byzantine Fault Tolerance

1 Introduction

As one of the key information systems in the energy field, the electricity trading platform plays a key supporting role in the electricity business. Once it is damaged or data is leaked, it may have a major impact on national security and people's livelihood interests. There are a large number of heterogeneous terminal devices in the 5G electricity trading private network. These terminal devices have functions such as data collection, transmission, processing, and storage. However, the environment where the terminal devices are located is open and complex, accompanied by security threats. In terms of the security of the 5G electricity trading private network, internal attacks are more harmful than external attacks. Internal attacks include slander attacks from malicious terminals and counterfeit terminals, black hole attacks [1], flooding attacks [2], sybil attacks [3], etc. If the attack invades the system of the electricity trading center, it will destroy the availability and integrity of the system and bring huge damage to the electricity trading system. Damage may result in information leakage, service unavailability, etc.

Traditional defense measures based on authentication and encryption [4] can only defend against external problems. Since people inside the network often have system keys, this method lacks effective means of defense against attacks inside the network. The network node trust evaluation technology based on sociology is an important way to solve internal threats. Trust evaluation technology guarantees the security of the entire system by limiting or isolating nodes with low trust values. It is widely used in wireless sensor networks [5], cloud services

* Corresponding Author

[6], and mobile ad hoc networks [7]. However, the existing trust management is often implemented by entrusting a trusted third party, which has the problem of a single point of failure. The distributed trust management technology [8] that emerged later solved the single point of failure problem, but malicious nodes in the distributed network can easily tamper with the data stored in the local trust value, so the blockchain-based trust management [9] method came into being.

Blockchain has the characteristics of distributed trust, openness and unforgeability [10], and can provide reliable services for the needs of trust management. However, due to the high communication complexity of the Practical Byzantine Fault Tolerance (PBFT) algorithm and the lack of restrictions on malicious nodes inside the distributed network [11], it is difficult to expand to 100 nodes. [12] Therefore, the blockchain based on the PBFT algorithm is often used in small networks, and it is difficult to apply to large networks such as electricity trading platforms.

In order to solve the above problems, we urgently need an efficient and reliable terminal trust management model to provide trust management services for electricity trading terminals. The main contributions of this paper are as follows:

- We propose an multi-trust factor terminal trust evaluation method to mitigate insider threat attacks launched from terminal devices. We collect a variety of trust factors to participate in trust evaluation and use information entropy to determine the weight of each trust factor, thereby improving the accuracy of trust evaluation.
- We propose an multi-trust factor terminal trust evaluation method to mitigate insider threat attacks launched from terminal devices. We collect a variety of trust factors to participate in trust evaluation and use information entropy to determine the weight of each trust factor, thereby improving the accuracy of trust evaluation.
- The experimental results show that the scheme in this paper has better efficiency and reliability than similar methods, and can effectively identify malicious terminals.

The experimental results show that the scheme in this paper has better efficiency and reliability than similar methods, and can effectively identify malicious terminals.

2 Related Work

2.1 Trust Evaluation

Trust is a recognition and approval of the other party. It is a kind of belief and cognition established in the process of communication, which affects the attitude and method of subsequent communication. In computing, trust is based on technical measures, security, authentication, and authorization. Users trust that computer systems are safe from malware, that sensitive information will not be leaked, and that data is properly processed and protected. At the same time, users also need to trust the security of network communication to ensure that data will not be tampered with or stolen during transmission. Building and maintaining trust is critical in the computing world as it plays an important role in how users use technology, conduct online transactions, and share information. Trust is a spontaneous, subjective and conscious judgment that demonstrates confidence and trust in the other party [13]. The purpose of any evaluation is to determine whether each entity is trustworthy and how to handle the trust relationship between entities. Trust evaluation technology is an important part of security and privacy protection, and has been widely used in many different application scenarios.

An earlier model for studying dynamic trust relationships is the PTM model [14]. Subsequently, Hassan et al. [15] proposed a trust model based on a vector mechanism by introducing a vector operation mechanism. Song [16] proposed a dynamic trust model based on fuzzy logic theory, but it failed to consider the influence of indirect trust and time factors on the model, which limited the dynamic adaptability of the model. Kamvar et al. [17] proposed a global trust model EigenRep, which first obtains the historical evaluation information of the service node, and then obtains the global trustworthiness of the service node based on the global reputation of the interaction partner itself or the local credibility of the requesting node to the interaction partner. reputation. The purpose of trust evaluation is to determine whether each entity is trustworthy and how to handle the trust relationship between entities.

Trust evaluation technology is an important part of security and privacy protection, and has been widely used in many different application scenarios. Kumar et al. [18] applied the trust evaluation technology to the mobile ad hoc network to improve the security of routing, but this method ignored the influence of human factors;

Challagidad et al. [19] proposed a multi-dimensional trust evaluation method in the cloud environment, which accurately evaluates the credibility of the cloud environment from the perspective of various stakeholders, but this method ignores the attack from the data level; Zhang et al. [20] proposed a trust evaluation method based on behavior-level trust and fog computing for internal attacks from the data level, which has certain advantages in detecting hidden data attacks, but this method relies on fog computing technology and is difficult to expand to other scenarios; Cheng et al. [21] proposed a trust evaluation scheme based on a three-valued subjective logic model, which can quickly and accurately evaluate the trust of vehicles, but this model is difficult to play an effective role in the scenario of lack of social information such as the electric power transportation system.

2.2 Blockchain-based Trust Management

The blockchain maintains a shared, reliable, and non-tamperable ledger through collaboration and consensus algorithms among multiple computer nodes. The blockchain consensus algorithm is one of the core technologies to ensure the security and correctness of the blockchain system. Its function is to enable nodes in the network to reach a consensus on the legality of the blockchain data [22]. At present, many consensus algorithms have been proposed, among which Practical Byzantine Fault Tolerance (PBFT) [23] is a widely used consensus algorithm. The PBFT algorithm was originally used to solve the Byzantine general problem [24], and was later used to solve the consensus problem in the distributed system. It can guarantee security and consistency, and has the characteristics of high efficiency and scalability. Since the PBFT algorithm was proposed, it has been widely concerned by academia and industry.

In recent years, with the continuous development and application of blockchain technology, more and more researchers have begun to improve the PBFT algorithm and apply it to blockchain consensus. Inspired by the Proof of Stake (PoS) algorithm, Buchman [25] and others added empty blocks and lock mechanisms to simplify the PBFT consensus process, and assigned different weights to each vote to deal with Sybil attacks [26]. In order to improve the scalability of the PBFT consensus algorithm under the premise of ensuring security, Micali et al. [27] proposed an efficient algorithm, using the encrypted lottery mechanism to elect verifiers and leaders, and then adopting the BA* protocol to reach new blocks the consensus is generated in rounds. In order to reduce the impact of faulty nodes on the blockchain system, Lei et al. [28] proposed a PBFT consensus algorithm based on reputation value. By giving faulty nodes a lower rating in the consensus process, the reputation value of faulty nodes decline, and nodes with low reputation values are less likely to become master nodes, thereby enhancing the security and reliability of the system. Wang et al. [29] applied blockchain technology to the self-organizing network of UAVs, and proposed an improved PBFT consensus algorithm based on reputation values, which selected UAVs with high reputation values as miner nodes and made them Implement consensus, assign voting weights to nodes according to reputation, and demonstrate through experiments that their method can reduce latency. The above research mainly improves the consensus efficiency of the PBFT consensus algorithm by grouping, reducing the number of consensus nodes and optimizing the consensus process, and improves the reliability of the blockchain system by marking malicious nodes through reputation values, etc., which are very innovative , but the existing methods to evaluate the reputation value or feasibility of nodes only refer to the behavior of nodes in the consensus process, but ignore the communication behavior of nodes in the blockchain network, making it difficult to accurately evaluate the credibility of nodes.

The blockchain-based trust management mechanism effectively solves the problem of single point of failure in the centralized trust management model [30], and can solve the problem of inconsistency and incompleteness of stored data in the traditional distributed trust management model. It has been widely studied by domestic and foreign scholars in recent years. Yu et al. [31] proposed a blockchain trust management mechanism based on the POW consensus algorithm. This method can effectively resist Sybil attacks, but the performance of this method is too poor to be applicable to the scenario of electricity trading; Yang et al. [32] proposed a consensus algorithm combining proof-of-work and proof-of-stake, which alleviated the performance bottleneck of the word proof-of-work consensus algorithm, but it still fell short of the need for frequent trust evaluation in electricity trading scenarios; Kouicem et al. [33] proposed a blockchain-based trust management model using the PBFT consensus algorithm. Compared with the model based on the POW consensus algorithm, the performance of this model has been greatly improved, but the communication complexity is still at the polynomial level, and it is difficult to Dealing with Sybil Attacks; Wang et al. [34] proposed a blockchain-based trust management model, and proved the reliability of the scheme through rigorous experiments in cross-platform scenarios. However, the performance of this scheme drops significantly after a large number of consensus nodes increase, and it lacks Efficiency.

3 Model Architecture

In the 5G power trading private network, the trading terminal may be maliciously hijacked by an attacker to become a malicious terminal. Attackers carry out slander attacks, black hole attacks, flood attacks, and sybil attacks through terminals, which seriously damage the security of power trading activities. Trust evaluation technology can effectively resolve these security threats, but due to the limited security protection capabilities of edge computing equipment, it will inevitably reduce the accuracy of trust evaluation and cause great damage to the power trading system. However, many current trust evaluation models do not take effective measures to avoid the negative impact of security risks from edge devices on trust evaluation. In response to the above problems, we propose an ERBTM mechanism to evaluate and manage the trust degree of power trading terminals. The system framework is shown in Fig. 1. The model consists of terminal layer, edge layer and service layer.

The terminal layer refers to a layer system composed of heterogeneous power trading terminals and agent software running on these terminals. These power terminals may be geographically distributed and have different architectures and characteristics. In the terminal layer, the system uses various proxy software to collect various evidences for terminal trust evaluation. These evidences may include the historical data of electricity transactions, the credibility of terminal equipment, the reliability and security of electricity transactions, etc. These evidences can be used to evaluate the credibility and security of the terminal equipment to determine whether the terminal equipment can be used for electricity transactions. In order to realize the functions of the terminal layer, various agent softwares need to be used. The design and implementation of the terminal layer is an important part of the power trading system. By collecting and evaluating various evidences, the credibility and security of terminal equipment can be improved, thereby ensuring the reliability and security of the power trading system.

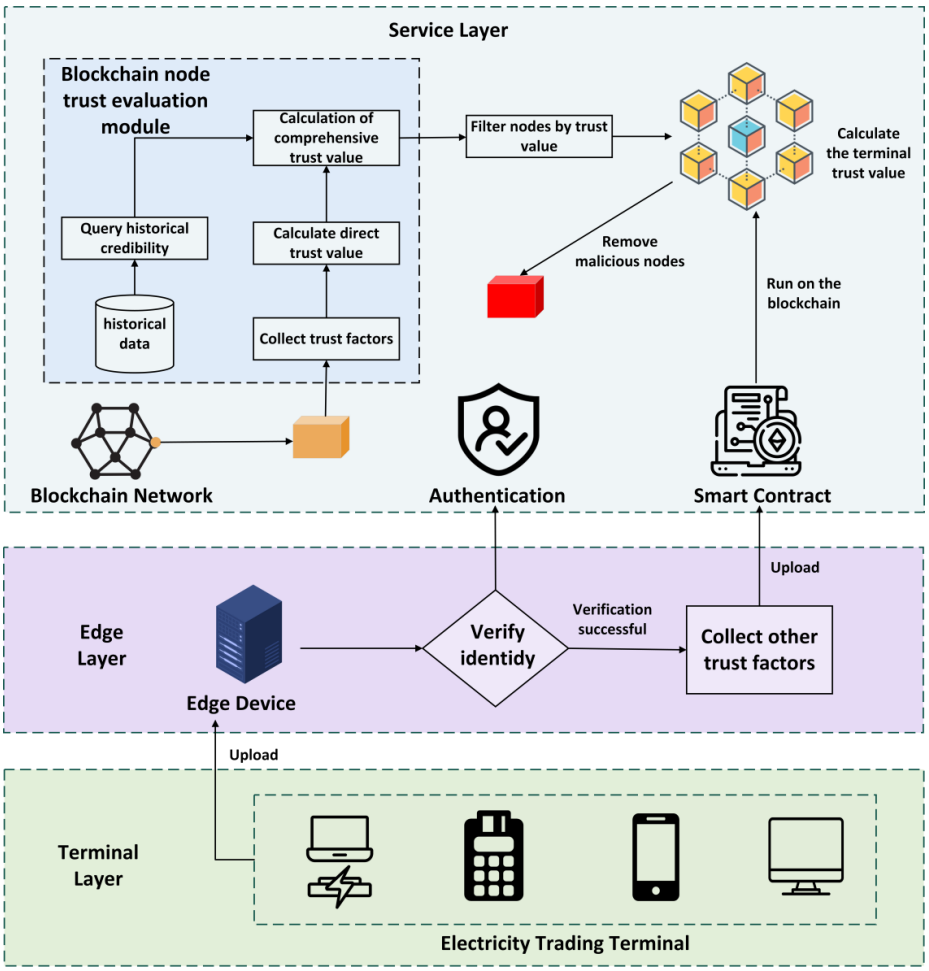

Fig. 1. The framework of our method

The edge layer consists of edge devices and blockchain nodes that support terminal access. These devices are distributed in different geographic regions and are responsible for collecting and processing terminal information in different regions. However, various stakeholders in the edge layer do not trust each other and lack a data sharing platform, resulting in platform data islands. Parties typically do not share data without incentives, making it difficult to establish trust and transfer and share data among multiple parties. Therefore, there is currently a lack of a platform that builds trust for multiple parties in a device-edge-cloud environment, and promotes trusted data transmission and sharing among all parties through an incentive mechanism. To solve this problem, blockchain technology is introduced as a multi-party interactive platform for process supervision and post-audit. Blockchain has the characteristics of traceable historical data, data non-repudiation, non-tampering, security and transparency. This makes blockchain ideal for establishing trust among distrustful parties at the edge layer. On this platform, we leverage smart contracts to dictate the terms of data sharing and access with end-point trust assessments. These smart contracts can automate the process of data sharing and access, and incentivize parties to participate in data sharing through incentive mechanisms. For example, through blockchain-based cryptocurrency incentives, parties can be encouraged to share their data, increasing data reliability and availability. The introduction of blockchain technology in the edge layer provides a multi-party interactive platform for the private electricity trading network, builds trust for multiple parties that do not trust each other, and promotes trusted data transmission and sharing. This will help to improve the security, reliability and transparency of the power trading network, and provide solid technical support for the development of the power trading network.

The service layer is an important part of the power trading system, including blockchain node credibility evaluation services, terminal identity authentication services, and trust value storage services. These services play a vital role in ensuring the security and stable operation of the system. Due to the identity authentication requirements of the terminal and the security issues based on the PBFT consensus algorithm, the service layer realizes the power transaction terminal identification function and the blockchain node trust evaluation function. The terminal identity recognition function is mainly realized by the identity authentication server. This function uses digital certificate-based security authentication technology to authenticate the terminal equipment connected to the network. By verifying the identity of the terminal equipment, it can effectively prevent unauthorized equipment from accessing the power trading network, and improve the security and reliability of the system.

The blockchain node trust evaluation module evaluates its credibility based on the behavior of blockchain nodes. Since there are many blockchain nodes lacking security protection in the edge layer of the power trading system, the credibility of these nodes varies greatly, and attackers can easily launch Sybil attacks on the blockchain system due to these vulnerabilities. Reliability is a necessary means to ensure system security. In this module, the credibility of nodes is evaluated by monitoring the behavior of blockchain nodes, historical transaction records, contribution and other factors. According to the evaluation results, trusted nodes are screened out and malicious nodes are eliminated to ensure the stable operation of the system and data security. In addition to terminal identity authentication and blockchain node trust evaluation, the service layer also provides trust value calculation, storage and query services for the system. Trust value is an important indicator in the private power trading system, which can measure the degree of trust between nodes and is used to ensure the reliability and smooth progress of transactions. The service layer calculates the trust value between the terminal and the blockchain node, stores and manages the trust value information, and provides users with trust value query services, which improves the operability and user experience of the system. The service layer plays a vital role in the private electricity trading network. By providing services such as terminal identity authentication, blockchain node trust evaluation and trust value calculation, storage and query, the service layer guarantees the security and reliability of the private power trading network. The establishment and optimization of the service layer is not only an important part of the construction of a private power trading network, but also an important driver to promote the application of blockchain technology in the power field.

Devices at the terminal layer are different from some devices at the blockchain layer and devices at the edge layer. Terminal devices carry all the operations of users, while other devices mainly provide services to the system. Therefore, in response to this difference, the system should provide different trust evaluation methods to terminal devices, and incorporate user behavior factors into the trust evaluation method, so as to effectively identify malicious nodes.

4 ERBTM System Model

4.1 Terminal Trust Value Calculation

The Efficient and Reliable Power Trading Terminal Trust Management (ERBTM) model based on blockchain can be divided into three parts: terminal trust value calculation, marginal blockchain node credibility evaluation and trust value storage. The calculation of the terminal trust value is mainly for the terminal layer power trading terminal, the credibility evaluation of the edge layer blockchain node is mainly for the edge layer blockchain node, and the trust value storage mainly provides the safe storage and retrieval function of the trust value. This section details how to use these two parts to build a robust and efficient endpoint trust management model.

The electricity trading terminal carries all user behaviors, which makes the trust evaluation of the terminal different from the trust evaluation of the blockchain nodes. More factors related to user behavior need to be considered, and we need to additionally consider the terminal trust evaluation method. Due to relevant regulations, the terminals of this system can only conduct transactions through the internal private network, so the trust evaluation mechanism is mainly used to target insider threats.

I In the face of insider threats, assessing the trust of endpoints is critical. Insider threats usually refer to cyber attacks launched by internal employees of enterprises or governments, third-party service providers, contractors, etc. from the enterprise intranet. When evaluating the trust of the terminal, it is necessary to comprehensively consider the behavior characteristics and communication characteristics of the terminal, and incorporate the user's behavior records and device data transmission characteristics into the evidence system of trust evaluation. In order to achieve this goal, this paper uses the agent software on the terminal to collect user behavior logs. These logs include information such as the number of user authentications per unit time, the similarity of authentication information, whether users log in from different locations, and the security assessment of devices. These behavioral characteristics can provide information about the user's behavior patterns and habits in the system, as well as the security posture of the device. After uploading the collected user behavior logs to the system through a dedicated network, the system uses smart contracts to calculate the trust value of the terminal based on user behavior. The smart contract quantitatively evaluates the credibility of the terminal based on the pre-defined trust evaluation algorithm, combined with user behavior characteristics and device data transmission characteristics.

Usually, users with legal identities can successfully pass identity authentication. The more user authentication attempts, the more failed user authentication attempts, and the lower the credibility of the user. The trust factor related to the user authentication times can be denoted as

$$Tr_{times}^{\Delta t} = \frac{1}{Ver_{times}}, \tag{1}$$

where the Ver_{times} is denoted as the total number of times the user authenticated within Δt.

The pure number of verifications is not enough to accurately represent the trust value of user behavior. Generally speaking, when the user verification fails, the greater the gap between the input verification information i and the correct identity verification j information, the greater the possibility that the user is a malicious user. The similarity is calculated using the cosine similarity, expressed as

$$Tr_{sim}^{t} = \cos(P_i^t, P_i^{t-1}) = \frac{P_i^t \times P_i^{t-1}}{\left\| P_i^t \right\| \cdot \left\| P_i^{t-1} \right\|}, \tag{2}$$

where the P_i^t is denoted as the user's verification information at time t, the P_i^{t-1} is denoted as the user's verification information at time $t-1$.

The user's remote login is also an important factor affecting the user's trustworthiness. Remote login includes changing terminal equipment or abnormal location. Such situations will greatly reduce the user's trust value and require the user to re-authenticate the user more strictly, which is related to remote login. Trust value can be denoted as

$$Tr_{offsite} = \begin{cases} 0, & \text{Remote login} \\ 1, & \text{Normal login} \end{cases}. \tag{3}$$

The device security assessment is mainly performed by the agent software installed on the terminal, mainly based on information such as the software version and operating system version of the system. The trust factor $Tr_{envir} \in [0, 1]$.

The modeling of trust evaluation of terminal communication characteristics adopts the above Bayesian model. Bayesian network is one of the most effective theoretical models in the field of uncertain knowledge representation and reasoning. Using conditional probability to express the correlation between various information elements can complete learning and reasoning under limited, incomplete and uncertain information conditions. The input of Bayesian network is binary, that is, positive or negative. The trust value is updated by updating the probability density distribution.

The trust model based on Bayesian theory uses Bayesian network to describe different trust factors of entities, specify prior probability distribution, deduce and calculate posterior probability according to corresponding Bayesian rules and conditional probability, and evaluate the trust degree of entities through effective multi-source fusion. The calculation method is as follows:

$$DT_{ij}^{t} = E(Beta(\delta, \epsilon)) = \frac{\delta_{ij} + 1}{\delta_{ij} + \epsilon_{ij} + 2}, \tag{4}$$

where the i is denoted as the terminal, and the j is denoted as the edge device interacting with i, the δ_{ij} is denoted as the number of normal interactions between nodes, the ϵ_{ij} is denoted as the number of failed interactions between nodes.

In summary, the user behavior trust value of the terminal at time t is denoted as

$$Tr^{t} = \omega_1 \cdot Tr_{times}^{\Delta t} \cdot Tr_{sim} + \omega_2 \cdot Tr_{offsite}^{t} + \omega_3 \cdot Tr_{envir}^{t} + \omega_4 \cdot DT_{ij}^{t}, \tag{5}$$

$$\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1. \tag{6}$$

Where ω_i represents the weight of the trust factor, we adopt the information entropy theory to dynamically determine the weight of the trust factor, the calculation formula of the information entropy of the random variable is

$$H(x) = -\sum_{i=1}^{n} p(x_i) ln p(x_i), \tag{7}$$

where the $p(x_i)$ represents the probability of random event x_i, and information entropy can represent the value of information. When a kind of information has a higher probability of appearing, it means that it is spread more widely and has higher value. Therefore, the information with higher entropy. The trust factor should be given greater weight. Assuming that there is an evaluator in the system, the evaluator's evaluation of a device according to a certain trust factor can be divided into trust and distrust. F_i represents the trust value of the trust factor, and the probability of evaluation as trust is F_i, then the evaluation The probability of being untrustworthy is $1-F_i$, we can get

$$H(F_i) = -F_i ln(F_i) - (1 - F_i) ln(1 - F_i). \tag{8}$$

According to this theory, dynamic weights can be expressed as

$$\omega_i = \frac{H(F_i)}{\sum_{j=1}^{4} H(F_j)}. \tag{9}$$

In order to calculate the comprehensive trust value of the terminal, it is necessary to consider the influence of historical experience on the current, which can be obtained

$$T^t = Tr^t + \frac{1}{1+\mu\Delta t} \cdot T^{t-1}, \tag{10}$$

where the T^t is denoted as the comprehensive trust value of the terminal at time t, the T^{t-1} is denoted as the comprehensive trust value of the terminal at time $t-1$, the Δt is denoted as the time difference between time t and time $t-1$, the μ is denoted as the attenuation coefficient.

This paper adopts the above formula, takes the public key of the user and the device and the credible evidence collected by the terminal agent software as input, and runs the smart contract to obtain the trust value of the terminal device, the algorithm of the smart contract is shown as algorithm 1.

Algorithm 1. Terminal trust computing algorithm

Input: The public key *PK* of the user and the device, the number of user verifications Ver_{time} from time $t-1$ to time t, the verification information input by the user at time t, the location $site_t$ of user login, the environment evaluation Tr_{envir} of user login, and the system parameter μ.

Output: The trust value Tr_t of the terminal at time t;

1: T_{t-1} Query the time of the results of the previous trust assessment through *PK*;
2: **if** $T_{t-1} != t-1$ **then**
3: Return error time;
4: **end if**
5: $P_{t-1} \leftarrow$ Query the last input information through *PK*;
6: $Tr_{times}^{\Delta t} \leftarrow$ Calculated by Equation 1;
7: $Tr_{sim}^{t} \leftarrow$ Calculated by Equation 2;
8: $Tr_{offsite} \leftarrow$ Calculated by Equation 3;
9: $Tr^t \leftarrow$Calculated by Equation 5;
10: $T^t \leftarrow$ Calculated by Equation 10;
11: Return the trust value *Tr* of the terminal at time t.

4.2 Credibility Evaluation of Edge Blockchain Nodes

The PBFT consensus algorithm is often used to solve the consensus problem of distributed systems in the alliance chain environment, but there are many problems in the PBFT algorithm. First, the communication complexity of the algorithm is at the polynomial level, making it difficult to expand the number of consensus nodes. Secondly, the master node replacement strategy of the algorithm is selected sequentially according to the node number. The lack of consensus node joining and exiting strategies makes the system vulnerable to Sybil attacks and DDos attacks. However, edge layer devices magnify these problems due to weak computing power and limited security protection capabilities. This paper proposes a trust-based practical Byzantine fault-tolerant algorithm (Trusted-PBFT) to solve these problems.

As shown in Fig. 2, the system adopts a trust evaluation method for all nodes to quantify the reliability of each node, and then establishes a trusted priority queue based on the trust value, and puts the node with the highest trust value at the head of the queue as the master node. The system selects n/φ nodes as consensus nodes, and the remaining nodes as candidate nodes.

All nodes need to update blocks synchronously. Among them, the consensus protocol is divided into four stages: pre-preparation, master-standby, confirmation and response. The specific process is similar to that of the PBFT algorithm. When the master node fails, the system takes the first node of the trusted priority queue as the new master node. When the trust value of the consensus node is lower than the system parameter σ, the system will delete the consensus node, and then add it to the recovery pool, and become a candidate node again after initialization.

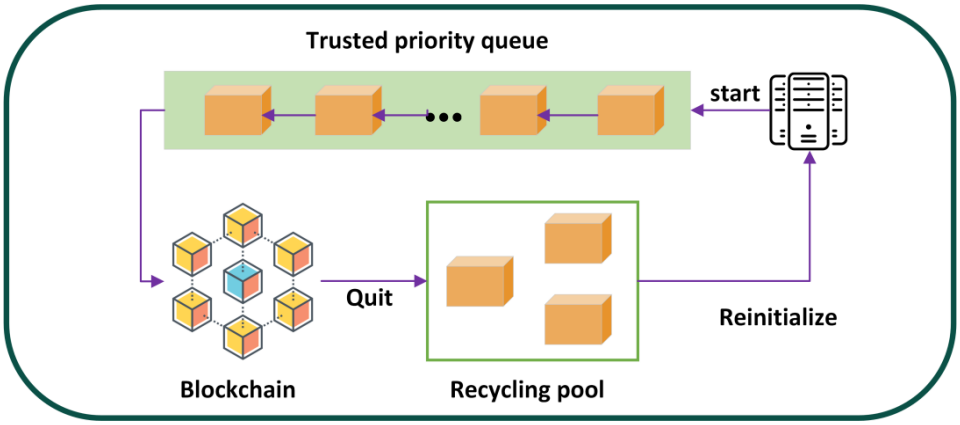

Fig. 2. Trusted-PBFT algorithm framework.

The credible priority queue is the core of this algorithm, which is implemented based on trust evaluation. Trust is an entity's subjective belief in another entity, which is determined by the entity's natural attributes and is dynamic. The dynamism of blockchain distributed nodes is a major challenge for its trust assessment. We calculate the credibility of nodes by comprehensively considering three trust factors: node feedback, node honesty, and node activity.

We use Bayesian theory to calculate the trust factor of node feedback. Ganeriwal et al. use Bayesian theory to evaluate trust. Bayesian formula can be used to fit the reputation distribution and beta distribution, and finally expressed by the statistical expectation of the reputation distribution, that is

$$DT_{ij} = E(Beta(\delta,\epsilon)) = E(f(p|\delta,\epsilon)) = \frac{\delta_{ij}+1}{\delta_{ij}+\epsilon_{ij}+2}, \tag{11}$$

where

$$f(p|\delta,\epsilon) = \frac{p^{\delta-1}(1-p)^{\epsilon-1}}{\int_0^1 u^{\delta-1}(1-u)^{\epsilon-1}du} = \frac{\Gamma(\delta+\epsilon)}{\Gamma(\delta)\Gamma(\epsilon)}p^{\delta-1}(1-p)^{\epsilon-1}, \tag{12}$$

the δ_{ij} is denoted as the number of successful interactions between entities, ϵ_{ij} is denoted as the number of failed interactions between entities. This method mainly reflects the credibility of the node through the communication status of the node, and the feedback trust factor of a certain node is

$$Rep_i = \frac{\sum_{j=1}^{n} DT_{ij}}{n}, \tag{13}$$

where the Rep_i is denoted as the Feedback trust factor of node i . It comprehensively considers the feedback between node i and all nodes.

The honesty of nodes can be calculated by the consensus completion rate. Honest nodes vote normally during each round of consensus, so nodes with a high consensus completion rate are more honest, and this trust factor is represented as

$$H_i = \sqrt{\frac{\vartheta}{n}}, \tag{14}$$

where the n is denoted as the number of times node i participated in consensus, the ϑ is denoted as the number of times the node successfully completed consensus.

The activity of a node refers to the frequency at which it participates in consensus over a period of time, and this trust factor is expressed as

$$p_i = \alpha e^{-\frac{1}{n}} + \beta, \tag{15}$$

where the n is denoted as the number of times node i participated in consensus, the α can adjust the growth rate, the β is responsible for adjusting the threshold of node activity. Therefore, while observing normal and abnormal communication, we calculate the average credibility of the data sent by the node, and combine the data credibility with the communication behavior credibility to obtain the comprehensive credibility of the node. The trust value of the node is represented as

$$T_i = \omega_1 \cdot Rep_i + \omega_2 \cdot H_i + \omega_3 \cdot P_i, \tag{16}$$

where the ω_i is the weight of the trust factor, which is determined using the method of trust entropy in the previous section. After the derivation of the above equations, the algorithm of the trusted priority queue we propose is shown in Algorithm 2. This paper uses this method to screen all nodes, divide the nodes into $n/2$ formula nodes, and $n/2$ candidate nodes, and ensure that $n/2$ is greater than $3f + 1$, where the n is the total amount of all nodes, f is the maximum number of malicious nodes.

Algorithm 2. Trusted-PBFT

Input: Node array A and node trust value array T.
Output: Node A is converted into a trusted priority queue.

```
Function PriorityQueue(A)
    n ← The length of array A
    for i ← 0.5 · n do
        Heapify(A, i, n)
    end for
    for i ← n to 2
        A[1] ← A[i]
        A[i] ← A[1]
        n ← n ←1
        Heapify(A, i, n)// Extract the highest priority element from the heap
    end for
End Function
Function Heapify(A, i, n)// Implement a heap data structure
    l ← 2i
    r ← 2i + 1
    largest ← i
    If l ≤ n and A[T_l] > A[T_i]:/
        largest ← l
    end if
    If r ≤ n and A[T_r] > A[T_i]:
        largest ← r
    end if
```

23: If largest ≠ *i*://
24: *A*[*i*] ← *A*[*largest*]
25: *A*[*largest*] ← *A*[*i*]
26: Heapify(*A*, *largest*, *n*)
27: end if
28: End Function
29: Return A

4.3 Trust Value Store

Trust Value Storage Smart Contract. This section introduces a data storage smart contract, which provides a secure trust value storage service for the system, and the smart contract is deployed on the blockchain at the edge layer. In this system, the raw data collected by the edge devices and the corresponding index data are stored on the data server and the blockchain on the terminal side, respectively. This distributed data storage method can improve data security and reliability. In addition, the system also stores index data information on the blockchain at the edge layer, including the storage address, timestamp and terminal node identification information required to retrieve the original data. By storing these index data, the system can effectively manage and retrieve data stored on different nodes and improve data access efficiency. As a trusted storage medium, the blockchain at the edge layer can ensure the integrity and non-tampering of data. At the same time, since the data is stored at the edge layer, the transmission delay of data in the network can be reduced and the response speed of the system can be improved. By utilizing the edge layer blockchain to store data and index information, the system can achieve secure data storage and efficient data management. Such an architecture can meet the system's demand for high-speed storage and fast retrieval of large amounts of data while ensuring data privacy and security.

The execution of the trust value storage smart contract is divided into three stages. The first stage is the preparation stage of trust value index information. After the edge node collects the data, it calculates the corresponding hash value through the message digest algorithm. Then, the terminal node packs the hash value and other necessary information (the hash value of the original data, the storage address, etc.) into the data index information RawJson, which contains the hash value, the version number of the original data, and the Multiple storage addresses and storage timestamp information. RawJson is serialized and stored in JSON format. Multiple storage addresses of multiple original data sources means that multiple data copies can be generated and stored in multiple trusted storage nodes to prevent loss of original data due to storage node failure. Then, the terminal node signs the RawJson, packs it into a transaction, and uses the transaction as an input parameter to call the data storage smart contract on the terminal side blockchain.

The second stage is the execution stage of the data storage smart contract. At this stage, the smart contract will check the content of the input transaction and confirm that the signing node is registered on the current local network. The smart contract will then return a signed message to the end node containing the transaction address of the executed contract. The current contract execution process will enter the waiting state, waiting for the wake-up message from the data server.

The third stage is the storage of raw data. The data collected by the terminal node will be stored in the data server, which is the data storage source chosen by the terminal node itself, ensuring that the private data is only owned by the terminal node and the trusted data server trusted by the terminal node. Data storage for terminal nodes can be done in batches to optimize storage efficiency.

Trust Value Storage Structure. As shown in Fig. 3, We use a Merkle tree to store the set of trust values under the blockchain. A Merkle tree is a hash tree that aggregates data layer by layer from leaf nodes to form a unique root node. In the blockchain, the Merkle tree is mainly used to organize transaction information, and each transaction corresponds to a leaf node of the Merkle tree, while ensuring that each leaf node is unique. Each leaf node stores the data label information of the corresponding transaction and the hash pointer pointing to the corresponding transaction in the blockchain on the terminal side. It should be noted that this hash pointer is only meaningful in the corresponding local network, because the terminal-side blockchain is a separate instance in each local network, and different local networks are independent of each other.

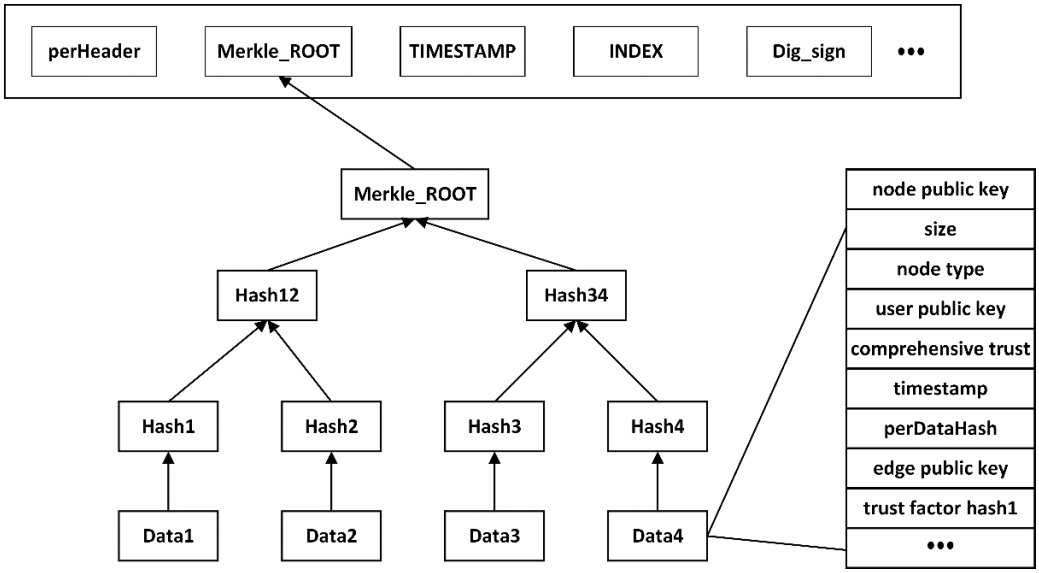

Fig. 3. The block structure

When the number of transactions is relatively large, the Merkle tree divides the transactions into two groups from left to right, and calculates a hash value in each group as the parent node information of the group, and then calculates these parent nodes The same calculation, until the final result is a binary tree with strong hash association, that is, Merkle tree. This organization method can not only effectively reduce the storage space, but also quickly verify the integrity of transaction information, because in the Merkle tree, the hash value of the root node is associated with the hash values of all leaf nodes. If the transaction information changes, the hash value of the root node will also change accordingly. Therefore, it is possible to verify whether the transaction information has been tampered with by comparing whether the hash values of the root node are consistent.

The advantage of the Merkle tree is not only that it can improve the security and efficiency of transaction information, but also that its nodes cannot be tampered with. This is because each node in the Merkle tree is calculated by the hash value of its child nodes, so if you want to modify the value of a node, you need to modify the hash values of all child nodes of the node at the same time. Hash value, this operation is very difficult, because the number of nodes to be modified at the same time is very large. Therefore, each node in the Merkle tree has an immutable characteristic, which is also an important part of blockchain technology.

In the blockchain, each block contains multiple transaction information, which needs to be verified before being added to the blockchain. In the traditional blockchain structure, each block stores all transaction information. Although this method can ensure the integrity of each transaction, it also increases the storage and transmission burden of the blockchain.

The Merkle tree provides a more efficient solution. It can organize all transaction information into a binary tree structure, and compress all transaction information into a root node hash value by calculating the hash value of each node. This method can not only effectively reduce the storage and transmission burden of transaction information, but also ensure the integrity and security of each transaction.

In addition, Merkle trees can quickly verify whether a certain transaction exists in the blockchain. Since each transaction corresponds to a leaf node of the Merkle tree, and the hash value of each node can be quickly calculated according to the structure of the Merkle tree, it is only necessary to compare the hash value of the target transaction with the Merkle The hash value of the root node of the tree can quickly verify whether the transaction exists in the blockchain.

Blockchain is essentially both a chain structure and a decentralized database. It contains several blocks generated by cryptographic methods, and each block contains transaction information of the entire network for a period of time. The transaction information in the block is packaged by the master node and verified by the consensus algorithm, and stored in the block In the distributed ledger of block chain. The block designed in this paper is composed of block header and block body.

Table 1. The block header structure

Element	Describe
perHeader	The hash value of the previous block header
Merkle_ROOT	The hash value of the Merkle tree
TIMESTAMP	The block generation time
INDEX	The block number
Dig_sign	The digital signature of the block generation node
pub_key	The public key of the block generation node

The important elements in the block header are shown in Table 1. The perHeader refers to the hash value of the previous block, which plays the same role as a pointer in the block header. The Merkle_ROOT is the root hash value of the Merkle tree, which can effectively prevent data from being tampered with. The TIMESTAMP refers to the block generation time. The INDEX refers to the block number. The pub_key refers to the public key of the block generation node. The Dig_sign refers to the digital signature of the node that generated the block. The public key is the identity of the node, and the digital signature ensures that the block is generated by the specified node.

Table 2. The block body structure

Element	Describe
node public key	The hash value of the node public key
size	The size of this trust assessment data
node type	The type of node
user public key	The hash value of the user public key
comprehensive trust	The trust value of a node at a certain moment
timestamp	The last trust evaluation time for this node
perDataHash	The last trust evaluation time for this node
edge public Key	The public key of the edge node
trust factor hash	The evidence of trust assessment

The block body mainly contains a Merkle tree. Leaf nodes store the hash value of the trust evaluation data, and the values of other nodes are hash values obtained by computing the values of the child nodes of the node. The trust evaluation data structure is shown in Table 2. We use the public key to represent the unique identity of an entity in the system, and the identities of the evaluated nodes, edge nodes and users are all represented by their public keys. The node type identifies whether the node is an end node or a blockchain node. The comprehensive trust is the result of the node's trust evaluation. The timestamp records the time of the trust evaluation. The perDataHash is a pointer to the data at the last trust evaluation for this node. The trust factor is the input data of the trust evaluation, which is easy to implement due to the exact length of the hash value, and the hash value is stored here.

The algorithm for generating blocks is shown in Algorithm 3, which includes the key steps for generating blocks. The nodes running this algorithm need to be elected through the Trusted-PBFT algorithm proposed by Algorithm 3. After the trust evaluation information is packaged, the system calculates the hash value of the packaged data, and then forms the hash value into a Merkle tree to obtain a Merkle tree The root hash value, and finally get the final block based on information such as the master node public key and digital signature.

Algorithm 3. Block generation algorithm

Input: The set of hash values $list_{factor}$ of trust evaluation factors, the public key $Pubk_{node}$ of the evaluated node, the public key $Pubk_{edge}$of the edge node.
Output: The final block *b*
$Pubk_{master}$ ← Trusted-PBFT algorithm selects the master node;
perHeader ← Get the hash value of the lastest block on the current chain;
Data ← Package trust assessment data according to $list_{factor}$, $Pubk_{node}$ and $Pubk_{edge}$;
perHeader ← Add *hash*(*Data*) to the Merkle tree;
Get the time of the current master node and the signature of the master node
b ← Generate block according to *perHeader*, *perHeader* and other information.
Return the final block *b*;

5 Experiment and Evaluation

The experiment uses docker technology to simulate the nodes and blockchain of the blockchain distributed network. The specific experimental settings and parameters of this experiment are shown in Table 3. Our experimental equipment uses Intel Core i7-9700 cpu, 16GB memory, 1TB hard disk, and the system runs on Ubuntu 20.04 LTS system.

Table 3. The specific experimental settings and parameters of this experiment

Parameter	Value
Cpu	Intel Core i7-9700
Memory	16GB
Hard Disk	1T SSD
Initial trust value	0.5
System	Ubuntu 20.04 LTS
φ	2

5.1 Consensus Algorithm Effectiveness Experiment

Consensus delay refers to the time required from the time the client sends a transaction request to the time the client completes the transaction confirmation. This time includes factors such as the broadcast propagation time of the transaction in the blockchain network, the time the transaction is received and verified by the node, the time the consensus algorithm is executed, and the network delay in the blockchain network. Low latency means that the execution time of consensus algorithms is shorter, and transactions in blockchain networks are confirmed faster, reducing the possibility of blockchain forking. The consensus delay is calculated as

$$Delay_{consensus} = T_C - T_R. \tag{17}$$

where the $Delay_{consensus}$ is denoted as the Consensus delay, the T_c is denoted as the time when the transaction was confirmed, the T_R is denoted as the time of transaction generation. We use the consensus delay as an indicator of performance. In other words, we measure the time from when a transaction is committed to when it is written to the blockchain, which is the consensus latency. In the experiment, the number of blockchain nodes was increased from 40 to 100, aptly describing a large electricity trading network.

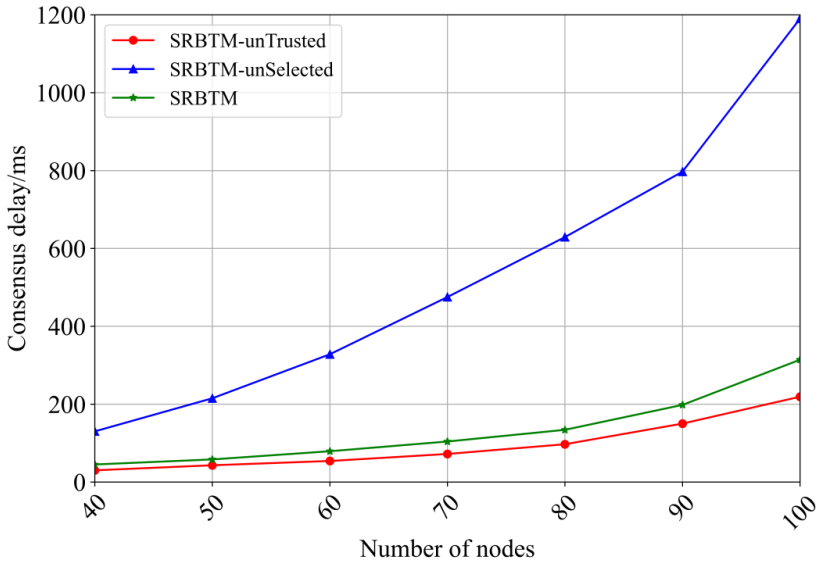

Fig. 4. Consensus latency under different conditions for the same model

As shown in Fig. 4, our method is tested under normal conditions, under the condition of not using a trusted priority queue and randomly selecting a considerable number of nodes, and under the condition of using a trusted priority queue without screening Test and get three different sets of data. We have observed that the trusted priority queue will have an impact on performance, and this performance loss increases with the increase of the number of nodes, but this performance loss is within an acceptable range, which is the same as that obtained after the nodes are not screened. Compared with the group data, it is found that the system delay is greatly reduced through the screening of the trusted priority queue. The experimental results prove the effectiveness of our method in terms of system performance.

5.2 Consensus Latency Comparison Experiment

The experimental environment of this experiment is the same as that of the previous section. As shown in Fig. 5, we compare our model (ERBTM) with the Blockchain-based efficiency Trust management (BC-Trust) [33] model and the Lightweight Blockchain-based Trust Management system (LBTM) [35]. As the number of nodes increases, the advantages of this model become more and more obvious. In the case of 100 nodes participating, the consensus delay of the model in this paper is a quarter of that of the BC-Trust model and LBTM model. Because the BC-Trust model is mainly based on the PBFT algorithm, which has high communication complexity. Although the LBTM model has improved the PBFT model, the trust evaluation mechanism adopted by the model is too complicated, so the additional consumption of this method is higher than that of this method. The experimental data shows that the model in this paper has higher Efficiency than the BC-Trust model and LBTM model, and can better meet the needs of large-scale trading systems.

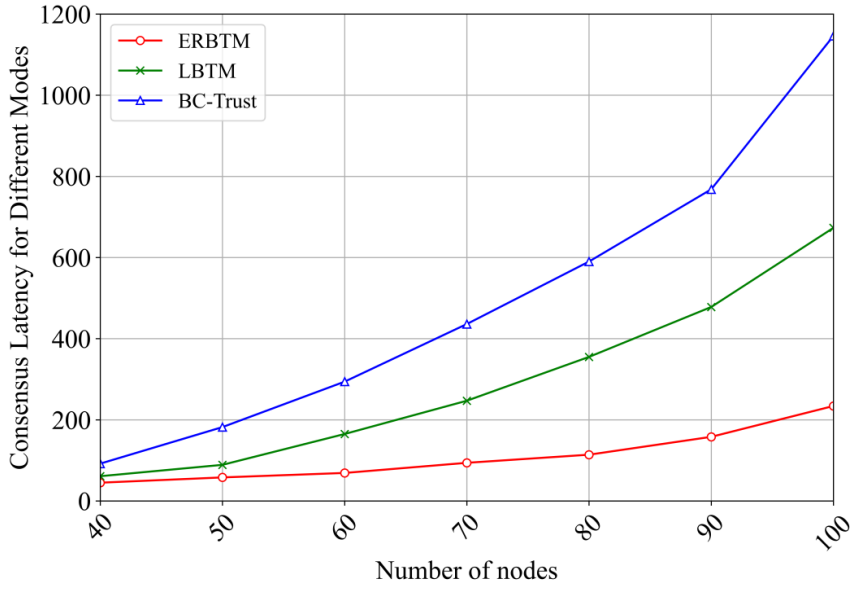

Fig. 5. Consensus latency for different models

5.3 The Impact of Parameters on the Fault Tolerance of Consensus Algorithms

Blockchain is a distributed, decentralized system that always maintains a shared state. The role of consensus algorithms is to enable networks to reach consensus on this state, which may sometimes not be achievable. Therefore, fault tolerance is an important aspect of blockchain technology.

In order to prove that the model in this paper can still maintain high reliability on the basis of reducing the consensus delay, this paper uses the following simulation experiment in Python based on the principle of random sampling. The experiment adopts a blockchain network with a total of 40 nodes, randomly sets some nodes as

malicious nodes. A malicious node has a 70 percent chance of sending an error message. We increase the proportion of malicious nodes from 10% to 50%, repeats the experiment 1000 times, and takes the ratio of the number of successes and the total number of experiments.

We take the transaction success rate as an important criterion for evaluating the system's fault tolerance capability. Transaction success rate refers to the proportion of transactions issued by clients that have been successfully chained. It is defined as

$$Transaction_{sr} = \frac{Transaction_{success}}{Transaction_{success} + Transaction_{failure}}. \tag{18}$$

where the $Transaction_{sr}$ is denoted as the transaction success rate, the $Transaction_{success}$ is denoted as the number of transactions successfully stored in the blockchain, the $Transaction_{failure}$ is denoted as the number of transactions that failed to be stored in the blockchain. In order to measure the influence of φ on system reliability, that is, fault tolerance, $\varphi = \{1, 2, 3, 4\}$ are respectively taken, and the experimental results are shown in Fig. 6.

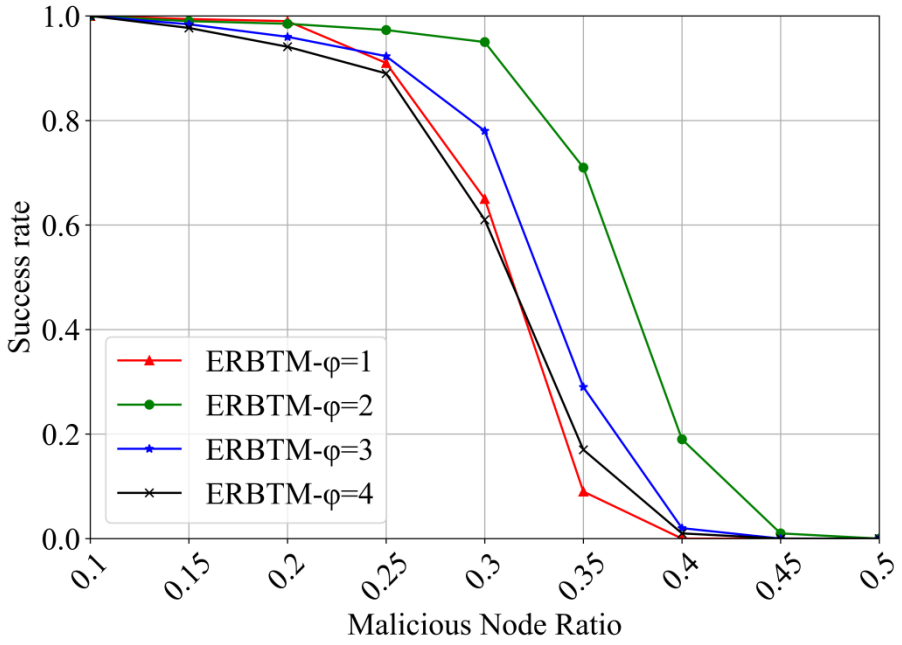

Fig. 6. The relationship between system fault tolerance and φ

It can be found through experiments that the value of φ seriously affects the stability of the system, and the smaller φ is not the better. When φ takes the minimum value of 1, there are a large number of malicious nodes in the system that have not been screened out, so that although the system can maintain a high success rate when there are few malicious nodes, when the proportion of malicious nodes increases, the success rate decreases significantly. When the value of φ gradually increases, the effect of the model is significantly reduced. This is because there are too few nodes participating in the consensus, and there is a certain error in the trust evaluation algorithm, which leads to a decrease in the fault tolerance of the system.

5.4 Trust Management Mechanism Fault Tolerance

The experimental environment of this experiment is the same as that of the previous section. In order to compare the advantages of this model with similar models, this experiment compares the model ERBTM-φ = 2 in this paper with the BC-Trust model. The experimental results are shown in Fig. 7. Because the BC-Trust model adopts

the PBFT consensus, only $\left\lfloor \frac{n}{3} \right\rfloor$ faulty nodes are allowed to exist, and the model in this paper improves the PBFT consensus based on the trusted priority queue, using trust evaluation technology to ensure the fault tolerance of the system.

Firstly, it can be observed from the experimental results that ERBTM-φ = 2 model performs better in fault tolerance compared to the BC Trust model. ERBTM-φ = 2 model can maintain system consistency and stability in the face of node failures and network attacks, and has higher fault tolerance ability. This is due to ERBTM-φ= The 2 model adopts trust evaluation technology, which can detect and identify malicious nodes in a timely manner, and exclude or limit their impact on the system. In contrast, the fault tolerance of the BC Trust model is limited by the constraints of the PBFT algorithm, which can only tolerate faults of a few nodes.

Secondly, ERBTM-φ = 2 model introduces a trust priority queue, which can adjust the participation order of nodes based on their trust evaluation results, improving the system's fault tolerance and robustness. This enables the system to better adapt to complex edge environments and face various attack behaviors, effectively preventing nodes from being breached. In contrast, the BC Trust model does not introduce a similar mechanism, and the participation order of nodes is fixed, making it susceptible to attacks from malicious nodes.

Finally, the results of this experiment are helpful in guiding system design and optimization. Through comparative experiments, we can gain a deeper understanding of the working principle and performance of the model, and discover its advantages in fault tolerance and room for improvement. These findings can provide guidance for further improving and optimizing blockchain systems, thereby enhancing their fault tolerance and robustness, and enhancing their security and reliability.

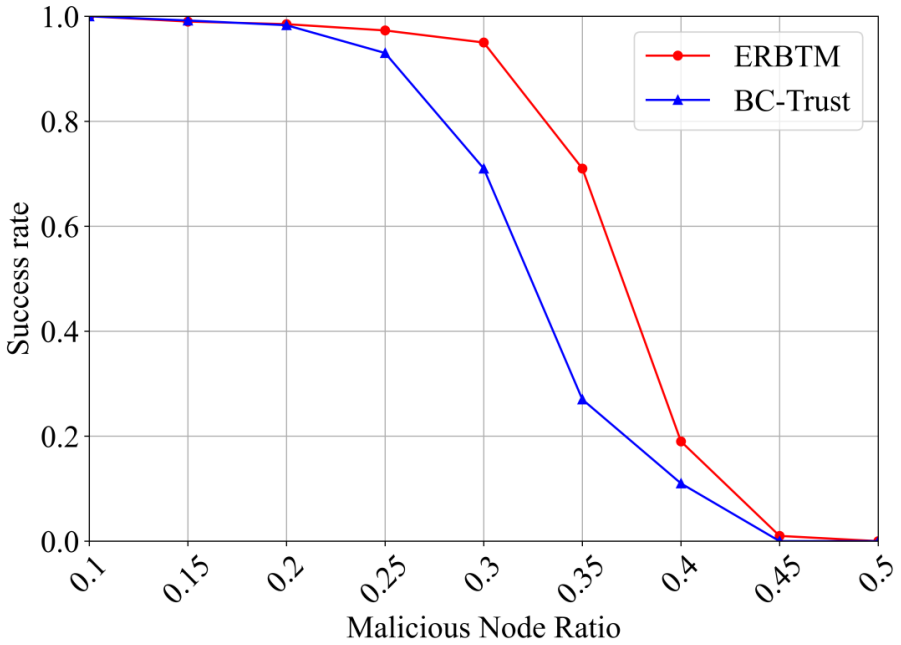

Fig. 7. Comparison experiment of different models of system fault tolerance

5.5 Terminal Trust Evaluate

The dynamic trust evaluation model can evaluate risks in real time, dynamically adjust trust levels, adaptively respond to security events, and quickly identify internal malicious nodes. In the method proposed in this article, the dynamic trust evaluation model is a key technology for screening trusted blockchain nodes and improving system fault tolerance.

In order to verify the ability of the model in this paper to identify malicious terminals, we compared the model in this paper with the INTEM [36] and BLTM [37] models. The INTEM model uses a Bayesian-based direct trust model, and the BLTM model combines direct trust evaluation and recommended trust evaluation, but neither of these two evaluation models combines the multidimensional trust factors of nodes in the electricity trading scenario.

In order to verify that the model in this paper can quickly calculate the trust value of the target node, it can also ensure the stability and accuracy of the trust evaluation process. In this section, we discuss the comprehensive trust value of a normal trusted source node to the same normal trusted target node and the same malicious untrusted target node in one evaluation cycle. For the same normal and trustworthy target node, we expect the comprehensive trust value of the normal and trustworthy source node to remain stable over multiple evaluation cycles. This means that even if there are slight fluctuations or changes in the behavior of the target node during different evaluation cycles, the trust value of the normal and trustworthy source node should remain relatively consistent. This proves the stability of the model in this paper, which can provide reliable trust evaluation results when facing normal nodes. On the other hand, for the same malicious and untrustworthy target node, we expect that within multiple evaluation cycles, the normal and trustworthy source node can accurately identify its untrustworthiness and provide a lower comprehensive trust value. Even if malicious nodes adopt a change strategy and attempt to deceive normal and trustworthy source nodes, this model should be able to continuously detect their malicious behavior and reduce their trust value in a timely manner. This verifies the accuracy and attack resistance of the model in this paper, which can effectively identify malicious nodes and provide reliable trust evaluation.

In this experiment, in the same network environment, that is, the proportion of malicious nodes inside the network is set to 30%, and the comprehensive performance of BLTM, INTEM and the model in this paper in the process of trust calculation is discussed. When a normal and trusted source node interacts with a normal node, the source node's comprehensive trust value for the target node will increase with the increase in the number of interactions. When interacting with a malicious node, the trust value will increase with the increase in the number of interactions decrease.

As shown in Fig. 8, in BLTM, INTEM and ERBTM, as the number of interactions increases, the trust value of the source node to the target node will increase. However, the growth rate of INTEM trust value is the fastest compared with the other two models. The ERBTM model needs to integrate more factors to make judgments, so the growth rate of trust value is slower. The BLTM model needs the recommended trust value of the global node as a reference, so the growth rate is the slowest.

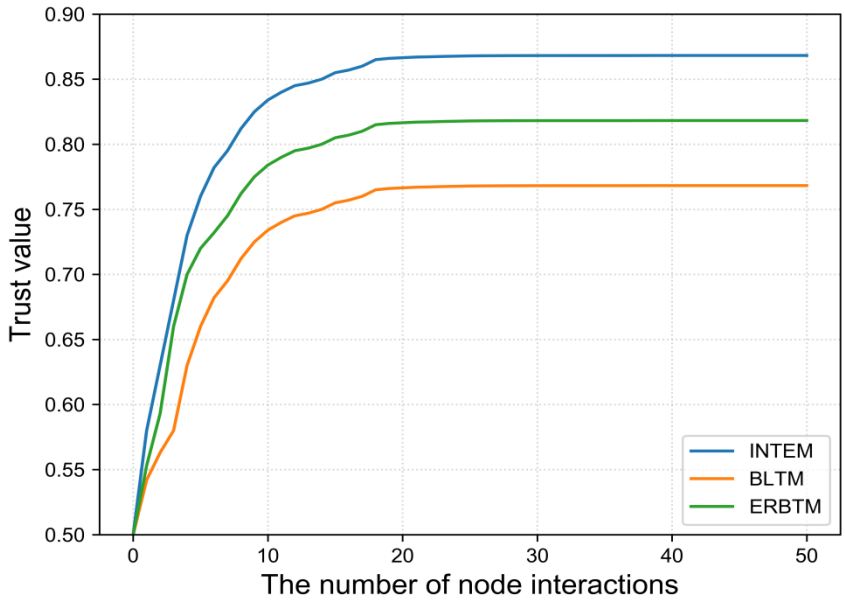

Fig. 8. Normal terminal trust evaluate

As can be seen from Fig. 9, as the experiment runs, the calculated trust values of the three models gradually decrease, but the model in this paper converges faster and converges at a lower result. This indicates that the method proposed in this article is more sensitive to malicious behavior and can identify malicious nodes faster. This is because the method proposed in this article can better describe the uncertainty in application scenarios, and comprehensively consider a variety of trust factors. Comprehensive consideration of multiple trust factors effectively avoids the impact of noise and improves the accuracy of model trust evaluation.

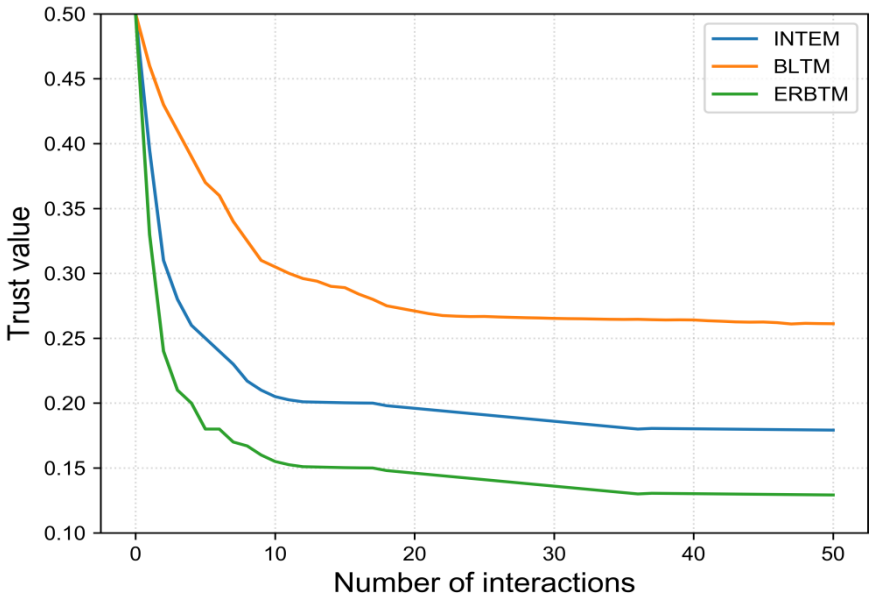

Fig. 9. Malicious terminal trust evaluate

6 Conclusion

This paper proposes a blockchain-based electricity trading terminal trust management model, which can effectively solve the problem of single point of failure in the traditional centralized trust management model. We use blockchain and smart contract technology to conduct trust evaluation and trust management for electricity trading terminals. Aiming at the poor Efficiency of other blockchain-based trust evaluation models, this paper proposes a Trusted-PBFT consensus algorithm based on trusted priority queues. Then we prove the premise of the model in this paper to ensure system reliability through experiments. In this way, the efficiency of the system is improved.

7 Acknowledgement

This work was supported by the China Souther Power Grid Technological Proiect with the Proiect No. GDKIXM20210107.

References

[1] S. Gurung, S. Chauhan, A survey of black-hole attack mitigation techniques in MANET: merits, drawbacks, and suitability, Wireless Networks 26(2020) 1981-2011.
[2] J. David, C. Thomas, Efficient DDoS flood attack detection using dynamic thresholding on flow-based network traffic, Computers & Security 82(2019) 284-295.

[3] D. Maram, H. Malvai, F. Zhang, N. Jean-Louis, A. Frolov, T. Kell, T. Lobban, C. Moy, A. Juels, A. Miller, Candid: Can-do decentralized identity with legacy compatibility, sybil-resistance, and accountability, in: Proc. 2021 IEEE Symposium on Security and Privacy (SP), 2021.
[4] M. Mamdouh, A-I. Awad, A.-A.-M. Khalaf, H.-F. Hamed, Authentication and identity management of IoHT devices: Achievements, challenges, and future directions, Computers & Security 111(2021) 102491.
[5] N. Temene, C. Sergiou, C. Georgiou, V. Vassiliou, A survey on mobility in Wireless Sensor Networks, Ad Hoc Networks 125(2022) 102726.
[6] A.-K. Junejo, I.-A. Jokhio, T. Jan, A Multi-Dimensional and Multi-Factor Trust Computation Framework for Cloud Services, Electronics 11(13)(2022) 1932.
[7] T.-K. Saini, S.-C. Sharma, Prominent unicast routing protocols for Mobile Ad hoc Networks: Criterion, classification, and key attributes, Ad Hoc Networks 89(2019) 58-77.
[8] A. Hbaieb, S. Ayed, L. Chaari, A survey of trust management in the Internet of Vehicles, Computer Networks 203(2022) 108558.
[9] S. Kudva, S. Badsha, S. Sengupta, H. La, I. Khalil, M. Atiquzzaman, A scalable blockchain based trust management in VANET routing protocol, Journal of Parallel and Distributed Computing 152(2021) 144-156.
[10] M. Kassen, Blockchain and e-government innovation: Automation of public information processes, Information Systems 103(2022) 101862.
[11] G. Xu, H. Bai, J. Xing, T. Luo, Y. Gu, S. Liu, A.-V. Vasilakos, SG-PBFT: A secure and highly efficient distributed blockchain PBFT consensus algorithm for intelligent Internet of vehicles, Journal of Parallel and Distributed Computing 164(2022) 1-11.
[12] H. Sukhwani, J.-M. Martínez, X. Chang, K.-S. Trivedi, A. Rindos, Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric), in: Proc. 2017 IEEE 36th symposium on reliable distributed systems (SRDS), 2017.
[13] J.-H. Cho, K. Chan, S. Adali, A survey on trust modeling, ACM Computing Surveys (CSUR) 48(2)(2015) 1-40.
[14] F. Almenarez, A. Marín, D. Díaz, J. Sanchez, Developing a model for trust management in pervasive devices, in: Proc. 2006 Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), 2006.
[15] H. Jameel, L.-X. Hung, U. Kalim, A. Sajjad, Y.-K. Lee, A trust model for ubiquitous systems based on vectors of trust values, in: Proc. 2005 Seventh IEEE International Symposium on Multimedia (ISM'05), 2005.
[16] S. Song, K. Hwang, M. Macwan, Fuzzy trust integration for security enforcement in grid computing, in: Proc. 2004 Network and Parallel Computing, 2004.
[17] S.-D. Kamvar, M.-T. Schlosser, H. Garcia-Molina, The eigentrust algorithm for reputation management in p2p networks, in: Proc. 2003 Proceedings of the 12th international conference on World Wide Web, 2003.
[18] R. Kumar, S. Tripathi, R. Agrawal, Trust-based energy-aware routing using GEOSR protocol for Ad-Hoc sensor networks, Wireless Networks 28(7)(2022) 2913-2936.
[19] P.-S. Challagidad, M.-N. Birje, Multi-dimensional dynamic trust evaluation scheme for cloud environment, Computers&Security 91(2020) 101722.
[20] G. Zhang, T. Wang, G. Wang, Detection of hidden data attacks combined fog computing and trust evaluation method in sensor-cloud system, Concurrency and computation: practice and experience 33(7)(2021) 1-1.
[21] T. Cheng, G. Liu, Q. Yang, A. Liu, W. Jia, Trust assessment in vehicular social network based on three-valued subjective logic, IEEE Transactions on Multimedia, 21(3)(2019) 652-663.
[22] L. Lao, Z. Li, S. Hou, B. Xiao, S. Guo, Y. Yang, A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling, ACM Computing Surveys (CSUR) 53(1)(2020) 1-32.
[23] M. Castro, B. Liskov. Practical Byzantine fault tolerance and proactive recovery, ACM Transactions on Computer Systems (TOCS), 20(4)(2002) 398-461.
[24] A. Sheikh, V. Kamuni, A. Urooj, S. Wagh, N. Singh, D. Patel, Secured energy trading using byzantine-based blockchain consensus, IEEE Access 8(2019) 8554-8571.
[25] H. Zhang, Byzantine Fault Tolerance in the Age of Blockchains and Cloud Computing, in: Proc. Proceedings of the 2022 on Cloud Computing Security Workshop, 2022.
[26] H. Yang, Y. Zhong, B. Yang, Y. Yang, Z, Xu, L. Wang, Y. Zhang, An overview of sybil attack detection mechanisms in vfc, in: Proc. 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), 2022.
[27] J. Chen, S. Micali, Algorand: A secure and efficient distributed ledger, Theoretical Computer Science 777(2019) 155–183.
[28] K. Lei, Q. Zhang, L. Xu, Z. Qi, Reputation-based byzantine fault-tolerance for consortium blockchain, in: Proc. 2018 IEEE 24th international conference on parallel and distributed systems (ICPADS), 2018.
[29] Z. Wang, R. Xiong, J. Jin, C. Liang, AirBC: A Lightweight Reputation-based Blockchain Scheme for Resource-constrained UANET, in: Proc. 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2022.
[30] Z. Cui, F. Xue, S. Zhang, X. Cai, Y. Cao, W. Zhang, J. Chen, A hybrid blockchain-based identity authentication scheme for multi-WSN, IEEE Transactions on Services Computing 13(2)(2020) 241-251.
[31] J. Yu, G. Zhang, D. Lu, H. Liu, Blockchain-based Crowd-sensing Trust Management Mechanism for Crowd Evacuation,

in: Proc. 2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2022.
[32] Z. Yang, K. Yang, L. Lei, K. Zheng, V.-C.-M. Leung, Blockchain-based decentralized trust management in vehicular networks, IEEE internet of things journal 6(2)(2018) 1495-1505.
[33] D.-E. Kouicem, Y. Imine, A. Bouabdallah, H. Lakhkef, Decentralized blockchain-based trust management protocol for the Internet of Things, IEEE Transactions on Dependable and Secure Computing 19(2)(2020) 1292-1306.
[34] C. Wang, S. Chen, S. Chen, X. Xue, H. Wu, Z. Feng, Trust Management for Reliable Cross-Platform Cooperation Based on Blockchain, in: Proc. 2021 IEEE International Conference on Web Services (ICWS), 2021.
[35] M. Amiri-Zarandi, R.-A. Dara, E. Fraser, LBTM: A lightweight blockchain-based trust management system for social internet of things, The Journal of Supercomputing (2022) 1-19.
[36] S. Sathish, A. Ayyasamy, M. Archana, An intelligent beta reputation and dynamic trust model for secure communication in wireless networks, in: Proc. 2018 Industry Interactive Innovations in Science, Engineering and Technology: Proceedings of the International Conference, 2018.
[37] X. Wu, J. Huang, J. Ling, L. Shu, BLTM: Beta and LQI based trust model for wireless sensor networks, IEEE Access, 7(2019) 43679-43690.