

Research on Trusted Security Protection Method for Edge Computing Environment

Jie Wang (✉ wangjie.sgcc@alumni.hust.edu.cn)

State Grid Hubei Electric power Research Institute <https://orcid.org/0000-0003-0396-1627>

Jiangpei Xu

State Grid Hubei Electric Power Research Institute

Jing Li

State Grid Hubei Electric Power Co

Chao Xu

state grid hubei extra high Voltage company

Huihui Xie

State grid hubei extra high voltage company

Research Article

Keywords: Edge computation, Computing environment, Trustworthy security, Protection method

Posted Date: May 25th, 2023

DOI: <https://doi.org/10.21203/rs.3.rs-2931330/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Research on Trusted Security Protection Method for Edge Computing Environment

Jie Wang^{*1}, Jiangpei Xu¹, Jing Li², Chao Xu³, Huihui Xie³

1.State Grid Hubei Electric Power Research Institute, Hubei Wuhan, 430077, China

2.State Grid Hubei Electric Power Co., Ltd, Hubei Wuhan, 430077, China

3.State Grid Hubei Extra High Voltage Company, Hubei Wuhan, 430050, China

*Correspondence should be addressed to Jie Wang: wangjie.sgcc@alumni.hust.edu.cn

Abstract: With the continuous development of mobile communication technology, edge computing technology has been gradually applied to some special business scenarios, such as disaster area rescue and forest fire warning. These application scenarios have certain requirements on the delay of task processing, and need enough network connection and computing resources. Due to the lack of ground network infrastructure, the traditional edge computing scheme can no longer fully meet the needs of users. However, the virtualized environment in cloud computing not only brings convenient services to users, but also faces security threats from different levels. The increasingly rich attack means and the widening of attack surface caused by virtualization have brought new challenges to the security protection of edge computing environment. How to build a secure and credible virtualization environment for cloud computing and alleviate users' concerns about cloud computing security has become an urgent problem to be solved in the further development of cloud computing technology. Based on the analysis of security risks and users' security requirements, this paper puts forward the combination of trusted computing and trust model in cloud computing, to build a safe and credible virtualization environment for users. Experimental results show that the proposed algorithm not only guarantees the validity of the trust model, but also reduces the resource overhead of edge devices in the process of trust evaluation.

Key words: Edge computation; Computing environment; Trustworthy security; Protection method

1 Introduction

Edge computing is a high-tech means developed in recent years, which can provide better technical support in data security and information sharing. The main purpose of edge computing is to effectively improve the availability and scalability of edge servers, so that the integrity of different query results cannot be modified [1]. Cluster servers, especially cloud services, are the technical means to effectively improve resource query and utilization in the future. The iterative algorithm of convex approximation technique is used to solve the local optimization problem [2] for MIMO multivariable integrated system. However, the security strategy of edge computing ensures the security and integrity of information when it is stored on each node of the Internet. The introduction of edge technology in node storage technology can effectively protect the problems caused by cloud interruption [3]. Distributed system security is also a scenario requirement for resource sharing and security, aiming at hosting or broadcasting multiple services on remote or local nodes [4]. Aiming at the problems of resource scheduling, resource management and secure transmission in edge computing, a content and function distribution network is proposed, which is an operating system-level edge computing platform at the bottom, can manage the managed devices in a unified way, and realize the unified scheduling of computing, network, storage and other resources. To solve the security problem of edge

computing, a security mechanism based on public key infrastructure (PKI) is proposed. The content and function distribution network can effectively realize demand distribution, resource scheduling and resource management. Experiments show that the public key infrastructure in the content and function distribution network can effectively resist sniffing attacks, man-in-the-middle attacks and other common attacks in edge computing [5]. In order to improve the security of data processing in the Internet of Things (IOT), a data security processing model of IOT based on edge computing is designed. The application model of the Internet of Things based on edge computing is constructed. In this model, the data redundancy elimination method and the differential privacy protection algorithm of the Internet of Things are successively used to remove the duplicate data and obtain the data with better quality, and then the privacy protection is carried out on the data of the Internet of things, so as to realize the data security processing of the Internet of things based on edge computing [6]. In the mobile Internet, the cellular network has an important research strategy for the resource calculation and application of the base station. Edge computing can effectively delay the user information in the base station [7]. This work studies the fundamental problems of task distribution and active buffering in fog networks under delay and reliability constraints. At the same time, the effectiveness monitoring of existing condition monitoring and protection methods in thermal protection is summarized [8]. Virtualization technology is becoming more and more popular in enhancing the security of operating system. The previous solution is to provide security functions through virtualization technology. This paper presents a security architecture named Cherub, which can be a TCB to achieve various security goals. Cherub is implemented in Linux, and the evaluation shows that Cherub is effective and practical in terms of security and performance [9]. Virtual reality technology, which was born at the end of the 20th century, has brought a new way for heritage protection. Combined with the mature digital processing methods, digital heritage will play an important role in the future. Regarding digitalization projects at home and abroad, it is proved that it is an inevitable trend to use digitalization to protect heritage, and this method has broad prospects [10]. The calculation is carried out from the user identity authentication, and the information of the user is identified at the terminal, thereby realizing an insurance mode for the data security of the user in the browser. When the user enters the system, it adds a layer of protection to the user's information security [11]. Trusted technology in edge computing can effectively enhance information enhancement in information security, and binary system is used to check whether the requestor of the system is in line with the expected security of the platform [12]. Based on the concept, model, industrial development and standardization progress of edge computing, this paper analyzes the main security risks faced by edge computing, and puts forward a reference framework for edge computing security covering seven dimensions, including application security, network security, data security, infrastructure security, physical environment security, security operation and maintenance support, and security management. Suggestions on standardization of edge computing security are put forward from three aspects of standard development, capacity building and application evaluation [13]. Edge computing is applied to power grid technology. The security risks of 5G edge computing are defined from the aspects of network services, hardware environment, and virtualization, edge computing platform, capability opening, application, management and data. Then the corresponding security protection requirements are put forward for the security risks. The security application of 5G edge computing is introduced by taking smart grid as an example [14]. Propose a method to detect anomalies in network traffic based on

techniques proven in intrusion detection research and products, and propose appropriate autonomous responses to anomalies detected by edge computing [15].

2 Trusted Protection of Edge Computing

2.1 Trusted Computing

It is a technology initiated and promoted by Trusted Computing Organization, which aims to protect the credibility of computer system based on hardware trusted platform. From the definition of trusted computing, the credibility of an entity is defined as the anticipation of its behavior, that is, whether an entity is trusted is judged according to whether its behavior conforms to the expected way. In computer system, the connotation of trusted computing includes data trustworthiness, connection trustworthiness, user trustworthiness, process trustworthiness and I/O trustworthiness. Through these factors trustworthiness, the integrity, security, and reliability of the whole process of computer system from power-on startup to task running end can be guaranteed.

The core idea of trusted computing is to construct a trust root with high reliability in computer system at first, and then from this trust root, a bottom-up trust chain is established step by step through the trust measurement technology of iterative computing, to extend the trust boundary to the whole computer system and ensure the credibility and security of the whole system. Figure 1 shows the classic trust chain construction process in computer system: the trust chain starts from BIOS boot block, passes through BIOS and boot sector, reaches the operating system, and further extends to applications and networks.

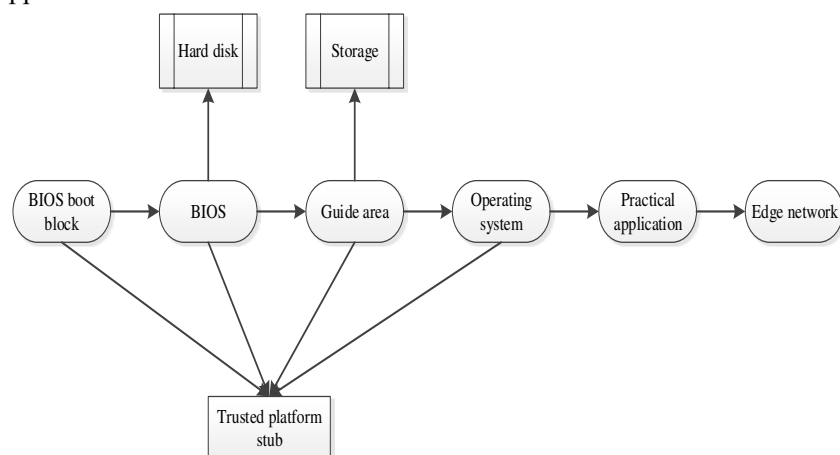


Figure 1 Trust chain of trusted computing platform

2.2 Trusted Protection and Challenges of Virtualized Environment

As the infrastructure in the edge computing environment, the security of virtualized environment is very important for the whole cloud computing security. In the face of more and more security attacks against virtualized environments, enhancing the security of virtual machine systems in computing platforms and building a credible virtualized environment have become a broad consensus and urgent need in the industry. In order to deal with the security threats in edge computing and solve various security attacks faced by users in virtualized environments, researchers have proposed various security protection schemes based on existing security technologies. According to the security requirements of users, the main security protection measures in edge computing environment include confidentiality and privacy protection of main user data, user identity authentication and access control, and deployment of antivirus software, firewall and intrusion detection system in virtual machine system. However, through practice, it is

found that these protection measures only rely on software systems are difficult to solve most security problems in edge computing, and because of the physical isolation between users and their virtual machines, users have concerns about the authenticity of the security status of virtualized systems.

With improvement of edge computing technology and the gradual deepening of the research on trusted protection of virtualized environment, some important progress has been made in solving the security problems of virtualized environment in edge computing based on the idea of trusted computing. However, in the face of updated attack methods and increasing security needs of users, the existing trusted protection of virtualized environment still faces four key challenges (Figure 2).

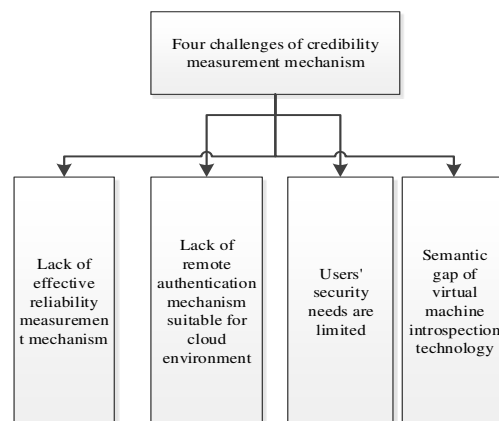


Figure 2 Four challenges of trusted protection

2.3 Problems faced by security protection under edge computing

Edge computing puts local data calculation on the edge of the network, and achieves the purpose of real-time response by processing the spatio-temporal stream data generated by massive terminals nearby. This data processing method reduces communication transmission delay, relieves the shortage of transmission bandwidth resources, and can meet the business requirements of real-time response. However, the introduction of edge computing also brings threats to the traditional security protection architecture, mainly including: (1) the security problem of massive device access. There are many kinds of massive devices, heterogeneous and diverse, and access and exit are dynamic and changeable. If the traditional way of authentication and behavior control of terminal devices will bring great pressure to the system. Therefore, most security protection architectures adopt the principle of zero trust to authenticate and authorize devices dynamically and based on real-time attributes. (2) There are ubiquitous intrusion security challenges in edge computing networks. New types of attacks change endlessly, and the traditional security measures based on feature matching or static intrusion detection model can no longer meet the real-time requirements of existing network security. Therefore, it is urgent to carry out research on real-time attack detection technology for edge computing networks. (3) Edge computing network interconnection service attack conducts security challenges. The existing persistent APT attack has changed from traditional single-point attack to multi-point breakthrough, and the difficulty of detection and defense is increasing. Therefore, fast, and effective collaborative management of group intelligence is a solution to solve the alarm fragmentation of security events in edge computing environment.

2.4 Trust Protection Process

In the edge environment, the frequency of direct communication between Edge Server A and

Terminal B is very high. Therefore, the direct trust of Terminal B can be obtained by evaluating some behavior attributes of Edge Server A and Terminal B in the interaction process. The indirect trust degree is measured by the comprehensive evaluation of the behavior attributes of Terminal B by other terminals that have interactive experience with Terminal B. On this basis, the time decay factor is introduced to measure the change of terminal B's comprehensive trust with the passage of time. Finally, the trust is rated, and the user rights are dynamically adjusted based on the rating results. The trust protection process is shown in Figure 3.

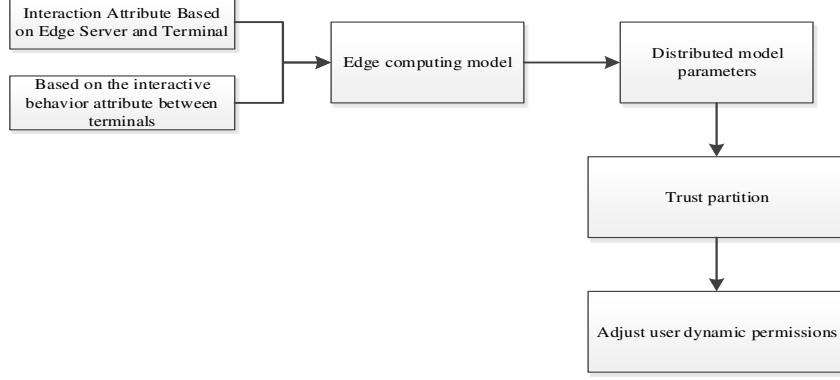


Figure 3 Trust protection process

3 Trusted Security Technology Evaluation

3.1 Reputation Evaluation

The starting point of reputation evaluation is the behavior of terminal equipment, Positive evaluation is given to the terminal devices actively participating in interaction, and the original trust value or negative evaluation is given to the terminal devices that do not participate in interaction for a period. Finally, the local edge server evaluates the trust value of all terminal devices within the management range, and then obtains the overall trust value.

Taking the resource unloading behavior of terminal equipment as an example, the relevant calculation formula is as follows:

$$f_{loc} = \max \prod (f_k^{act} - f_k^{\min})^{\rho_k}, k \in N \quad (1)$$

In the formula, f_k^{act} represents the actual value of resource offloading of the k -th terminal device, $f_k^{\min}$ represents the minimum value of resource offloading of the terminal device, and F represents the total amount of resources in the current environment. ρ_k represents the ratio of the terminal equipment to the total network power when offloading resources. The following formula is one of the calculation methods:

$$\rho_k = \frac{x_{LA,k}^{loc}}{\sum x_{LA,k}^{loc}} \quad (2)$$

Where $x_{LA,k}^{loc}$ represents the communication distance between the k -th terminal device and the local edge server.

$f_k^{\min}$ is usually determined by the performance of the terminal device itself and is usually considered a fixed value. According to the two constraints of formula (3), f_k^{act} can be solved by Lagrange multiplier method, as follows:

$$L(f_k^{act}) = \sum \rho_k \ln(f_k^{act} - f_k^{\min}) + \lambda(\sum f_k^{act} - F) \quad (3)$$

The Lagrangian function L in the above formula belongs to concave function, and the optimal value can be solved by the first-order linear condition, so the subsequent process is not discussed in depth in this paper. Finally, the overall trust value f_{loc} obtained by calculation will be used for resource scheduling and allocation.

3.2 Trust Evaluation

Trust relationship has complex applications in open systems. Edge computing model is based on the effective connection established by different organizations in open systems, which guarantees the basis of trust. Therefore, in the open system, the calculation and prediction of information are dynamic, and there are uncertainties of many factors.

Suppose that there are n edge servers and m terminal devices in the current environment to meet $n \square m$. Edge servers are deployed by cloud service providers at the convergence of terminal nodes, and as an important part of the middle tier, they provide trust management services for the underlying devices. Data sharing and synchronization between edge servers are completed by the cloud, including identity authentication, privacy management and trust evaluation of edge servers. Related research is beyond the scope of this paper. It is assumed that edge servers in the environment are always trusted, and terminal devices are required to have basic computing and interaction capabilities.

3.2.1 Service satisfaction

The trust value of the terminal device to the edge server is expressed as:

$$T_s = \alpha^* sat + \beta^{*rec} \quad (4)$$

Where sat represents the service satisfaction of the terminal device to the edge server, and rec represents the recommended credibility of other terminal devices to the target edge server. α, β represent corresponding coefficients, which can be set according to specific application scenarios. The calculation steps for sat and rec are as follows:

Step 1: When the terminal device and the edge server interact for the first time, the value of rel_s will change according to the following formula:

$$sat(t) = \frac{1}{2} + \left(sat(t_0) - \frac{1}{2} \right) e^{-(t-t_0)\lambda} \quad (5)$$

Where $sat(t_0)$ represents the service satisfaction before interaction, and λ represents the time attenuation factor. t_0 denotes the time when this interaction ends, and $t - t_0$ denotes the

time between the end of this interaction and the start of the next interaction. When the interaction occurs again, the $sat(t_0)$ value of the terminal device to the edge server will be updated.

Step 2: After the sat value is updated, this paper considers the update of the corresponding credibility from the discrete degree of service satisfaction. The following formula can be used as the update reference of the corresponding credibility rel_s :

$$rel_s = \frac{1}{(sat_j^i - sat_j)^2 + 1} \quad (6)$$

Where sat_j represents the service satisfaction of the j-th edge server itself, which is given by the cloud. Finally, after calculating the service satisfaction of the terminal equipment, the same calculation is made for other edge servers.

3.2.2 Recommended trust

Firstly, the terminal device obtains the service satisfaction degree of k terminal devices interacting with the terminal device to the edge server, and if there is no service satisfaction degree in the terminal device, the value is 0; The rec can be updated according to the following formula:

$$rec(t) = \frac{1}{k} \sum_{s=1}^k sat_s^i e^{-(t-t_0)\delta} \quad (7)$$

The recommendation degree obtained by terminal equipment also needs to consider its timeliness. Also considering the dispersion degree of service satisfaction, compare the recommended service satisfaction with the recommended difference. If the difference is small, it means that the service provided by the i-th edge server is relatively stable for terminal equipment, otherwise, it means that the service is unstable; The calculation formula is as follows:

$$rel_r = \frac{1}{(rec(t) - sat_j^i)^2 + 1} \quad (8)$$

Finally, we calculate the recommended satisfaction of the terminal device to the server, and do the same calculation for other edge servers. The trust values of the terminal device to N edge servers are calculated by using equation (5), and the results are arranged in descending order, and the edge server according to $Value_{max}$ is selected as the main service provider.

3.3 Edge Server Protection of Terminal Devices

When the terminal devices get the formal authorization from the edge server, they establish a stable communication relationship with each other. Therefore, the trust and security protection of the terminal devices should not only consider the interaction between the terminal devices, but also consider the evaluation of the terminal devices by the edge server. The trust value of the terminal device is defined as Formula(9):

$$T_e = \theta^* T_e + (1 - \theta)^* T \quad (9)$$

Where T_e denotes the trust evaluation of the terminal device to the terminal device, T denotes the trust evaluation of the edge server to the terminal device, and θ denotes the trust weight. The

calculation method of the above parameters will be discussed below.

If the current terminal device has interacted with other k terminal devices, and the interaction results include effective interaction and invalid interaction. Formula(10) is used to calculate the trust evaluation of the current terminal device.

$$T_e = \frac{1}{k} \sum_{s=1}^k N_s(t) \quad (10)$$

$$N(t) = \frac{\sum_{t_0}^t p_e(t) + 1}{P_t - M_e(t) + 2} \quad (11)$$

The reliability R_e of T_e is defined, and the reliability R_e of this value is considered from the dispersion degree of trust evaluation results and defined as Formula(12).

$$R_e = \frac{1}{\sum_{s=1}^k (T_e - N_s(t))^2 + 1} \quad (12)$$

The trust evaluation of the edge server to the terminal equipment is based on $N_j(t)$, which is the trust evaluation made by the edge server and the interaction history provided by the terminal equipment, which is defined as Formula(13):

$$T_s = \frac{\sum_{s=1}^k N_s(t)^* \omega(s, j)}{\sum_{s=1}^k \omega(s, j)} \quad (13)$$

$$s.t. T_s = 0.5, \sum_{s=1}^k \omega(s, j) = 0 \quad (14)$$

Where $\omega(s, j)$ represents the number of routing hops from the terminal device s to the terminal device j . When the total number of routing hops is 0, it means that the terminal device j has never interacted with the terminal devices, and the trust evaluation of the edge server to the terminal device is 0.5. In this paper, the reliability of this value is considered from the discreteness between the route hop count and the average route hop count. R_s can be calculated by the following formula:

$$R_s = \frac{1}{\sum_{s=1}^k (\omega(s, j)) - \frac{1}{k} \sum_{s=1}^k \omega(s, j)^2 + 1} \quad (15)$$

Trust is a relatively subjective concept, so most of the existing trust evaluation methods use prior knowledge or simulation results as the basis for assigning trust weights, which lacks sufficient

rationality and adaptability when applied in distributed and dynamic networks such as edge computing environments. Therefore, it is necessary to consider a relatively objective and unconstrained trust weight θ allocation method, which is defined as Formula(16):

$$\theta = \frac{R_e}{R_e + R_s} \quad (16)$$

The above formula can be used to adjust the trust weight objectively and dynamically depending on the change of the environment. If the credibility of the trust evaluation between the terminal devices is higher, the trust evaluation has a greater weight in the trust value T_e of the terminal device.

3.4 Trust Model of Edge Server

Let the learning model be $L(x, \varphi)$ means that the solving parameter x predict the trust level of the trust features. In that distribute trust evaluation algorithm based on stochastic gradient descent propose in this paper, the expression of the learning model is as follows:

$$L(x, \varphi) = \xi \left(\sum_{j=1}^m \delta^j(x^j, \varphi^j) \right) \quad (17)$$

In the formula, δ^j represents the specific learning model adopted by the j -th terminal device, and ξ represents the solution function given by the model, which is used to quickly weight and calculate the prediction result of the model. The goal of the learning model is to solve the parameter X satisfying the condition, that is, to solve the following equation:

$$\min \left(\frac{1}{n} \sum_{i=1}^n E(x : \varphi_i, y_i) + \gamma \sum_{j=1}^m z^j(x^j) \right) \quad (18)$$

Where, the function E represents the expected loss function of the model, and the function Z represents the regularization number of the j -th terminal device to solve the parameter. The index value of the trust feature in the model training process is $d(t)$ and t represents the iteration process of the model, and the iteration process $F(x, t)$ is defined as:

$$G(x, t) = E(x : \varphi_{d(t)}, y_{d(t)} + \gamma) \quad (19)$$

If the regularization function Z of the terminal device J is discontinuous, the proximal gradient algorithm can be used to solve the problem, which will not be discussed in depth in this paper. Finally, the parameter X is updated according to the above formula:

$$x_{t+1}^j = x_t^j - \eta_t \nabla^j G(x, t) \quad (20)$$

In the formula, η_t represents the learning speed of the iterative process t . When the parameter x is substituted into the Formula (18) after several iterations to obtain the optimal solution of the value y , the iteration ends.

The security protection of the terminal device can include several related factors such as

accessibility, satisfaction, and trust value. Considering the characteristics of the edge computing environment and the timeliness of the prediction results, this paper synthesizes the trust evaluation methods proposed in this chapter, and only discusses the overall trust value predicted by the algorithm when the trust characteristics of the terminal device are T_s and T_e . The purpose of restricting the value range of T_s and T_e between $[0, 1]$ is to quantify the trust characteristics and facilitate the normalization of the input parameters of the edge computing environment.

4 Experimental Analysis of Trusted Computing Based on Edge Computing

4.1 Environment settings

In order to verify the system performance and resource overhead of the trust model, the paper uses the learning model of the distributed trust evaluation algorithm to verify its efficiency. PSM model and RFSN model have good application effect in network computing. In this paper, the performance of these two models and stochastic model is compared. The experimental area is a square of $200\text{m} \times 200\text{m}$, and the resources of 200 nodes are randomly simulated. The standard parameters are shown in Table 1.

Table 1 Experimental simulation environment

Parameter	Description	Numerical value
M	The proportion of malicious nodes	10%-50%
N	Total number of nodes	200
α	Penalty factor	2
W	Sliding window	20
ES	Number of edge servers	2
L	Edge agent	1
	Develop system	Linux2.6.24
	Development Environment	Ubuntu 20.04
	Operating System Compiler	ARM-Linux-gcc-3.4.

In the experimental application process, the success rate of edge nodes can be set to 80%-90%, and the introduction of non-invasive factors in the simulation environment will cause the edge nodes to work abnormally. The malicious nodes in this paper are randomly generated by the system, serving malicious nodes and honest nodes respectively, and the proportion is 50%. In order to confirm the size of the penalty factor α , this paper examines the impact of the value of α on the direct trust value under different numbers of interaction failures. The experimental results are shown in Figure 4. In order to facilitate the calculation and comparison, α is rounded in the experiment.

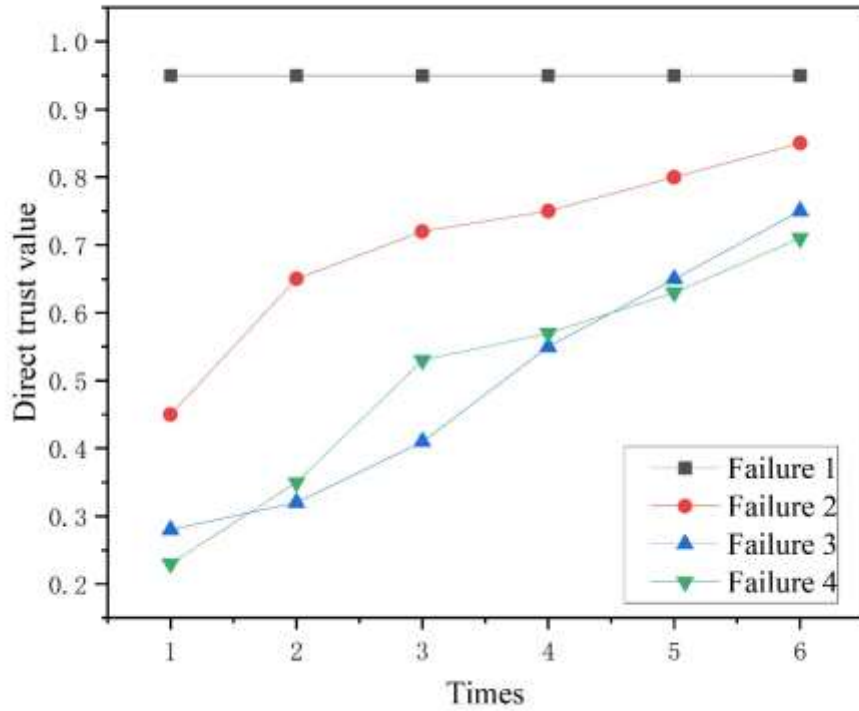


Figure 4 Effect of parameter α on direct trust values

The value of α has no effect on the direct trust value in the case of one failure in Figure 4. In other cases of failure, the direct trust value increases with the increase of α , indicating that the larger the value of the penalty factor, the smaller the penalty for the interaction failure node. When the value of α is greater than 3, after four failures, the trust value is still higher than the median value of 0.5, and the punishment effect is not significant. When $\alpha=1$, two or more interaction failures will cause the trust value of the node to be lower than the intermediate value of 0.5. In general, normal nodes also have a certain error rate, so this paper selects a value with a certain tolerance, and sets the penalty factor to 2. Indeed, the value of α can be adjusted according to the tolerance of nodes in the network to missteps.

4.2 Effectiveness evaluation

4.2.1 Analysis of comprehensive trust evaluation results

The comprehensive information is evaluated correctly, and the interaction ability between malicious nodes and normal nodes is tested experimentally. The initial value of trust for all nodes is set to 0.5. By continuously iterating the corresponding interaction cycles, the camouflage node starts to work from the fifth cycle, and after another 20 interaction cycles, the comprehensive trust evaluation results are shown in Figure 5.

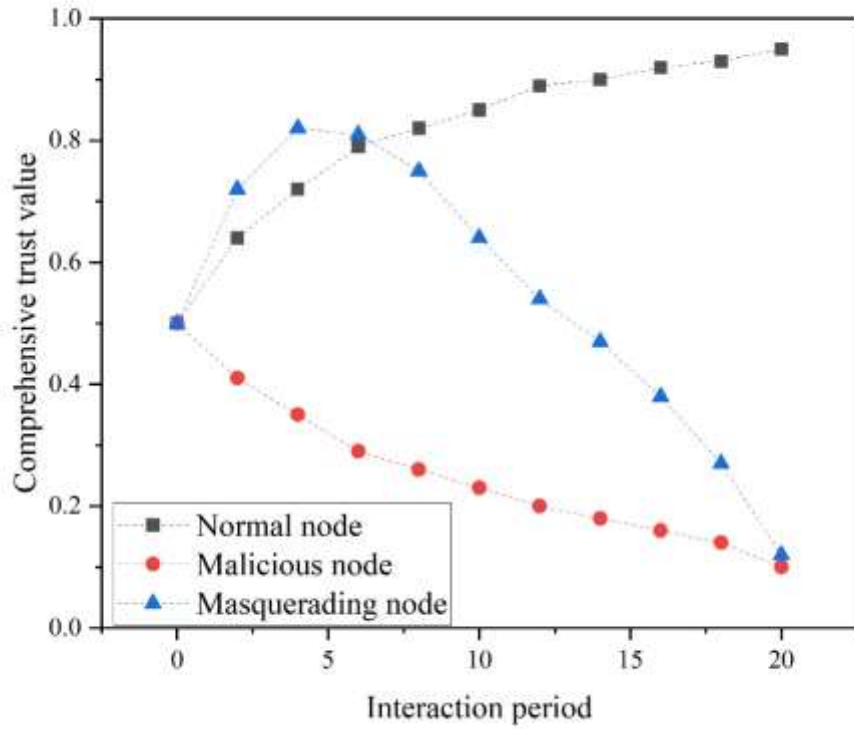


Figure 5 Comparison of trust values between normal nodes and malicious nodes

It can be seen from Figure 5 that with the increase of the number of interactions between nodes, the comprehensive trust values of nodes gradually tend to be balanced within 10 simulation cycles, the trust values of normal nodes gradually increase and approach 1, and the trust values of malicious nodes decrease and approach 0. When the masquerade node reaches a certain trust accumulation, it begins to provide malicious services, and the trust value decreases rapidly, which is lower than the intermediate value of 0.5 in the seventh cycle, so that the obtained trust value has a certain degree of discrimination. Therefore, the model can effectively distinguish between normal nodes and malicious nodes, and the comprehensive trust value obtained can also well reflect the behavior of nodes.

4.2.2 Result analysis of success rate

Successful interaction rate is the ability to successfully resist malicious node attacks from the execution information mechanism of the system. The higher the successful interaction rate is, the better the reliability of the system is. In the experimental part of this paper, 100 interaction cycles are used to analyze the ability of successful interaction rate. In order to verify the feasibility of the scheme, nodes send out corresponding connection requests in each cycle. An edge node may act as a service provider or requestor for other nodes. The proportion of malicious nodes is set to 10%, 20%, 30 and 40% respectively. In the network environment under different proportions, the trusted security mechanism and the node protection ability in the edge computing environment can be correctly evaluated. Figs. 6-8 describe the successful interaction rates of the four models when the proportion of different malicious nodes is 10%-40% respectively.

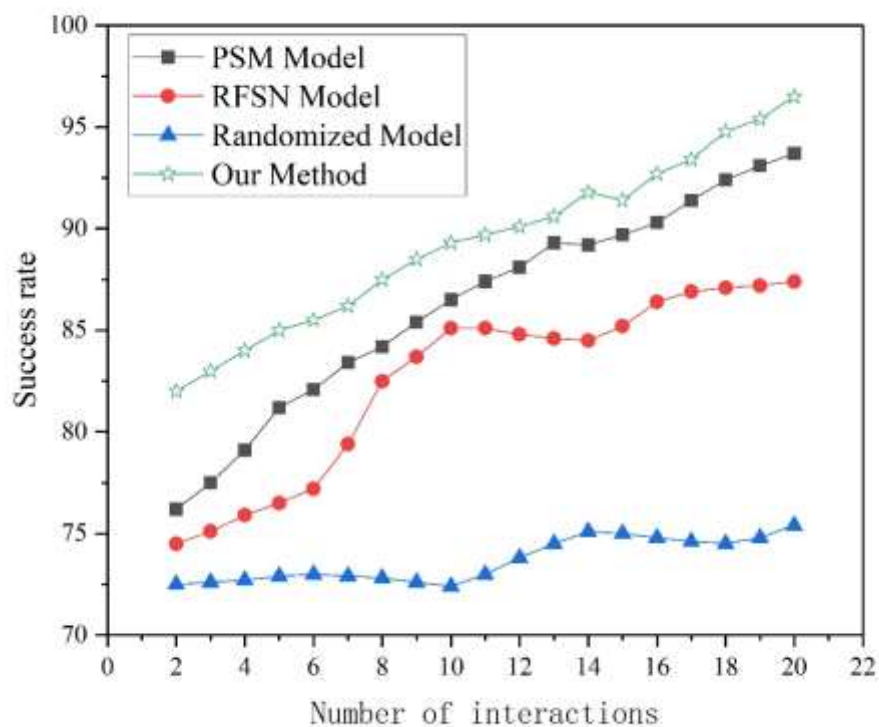


Figure 6 Malicious node ratio 10%

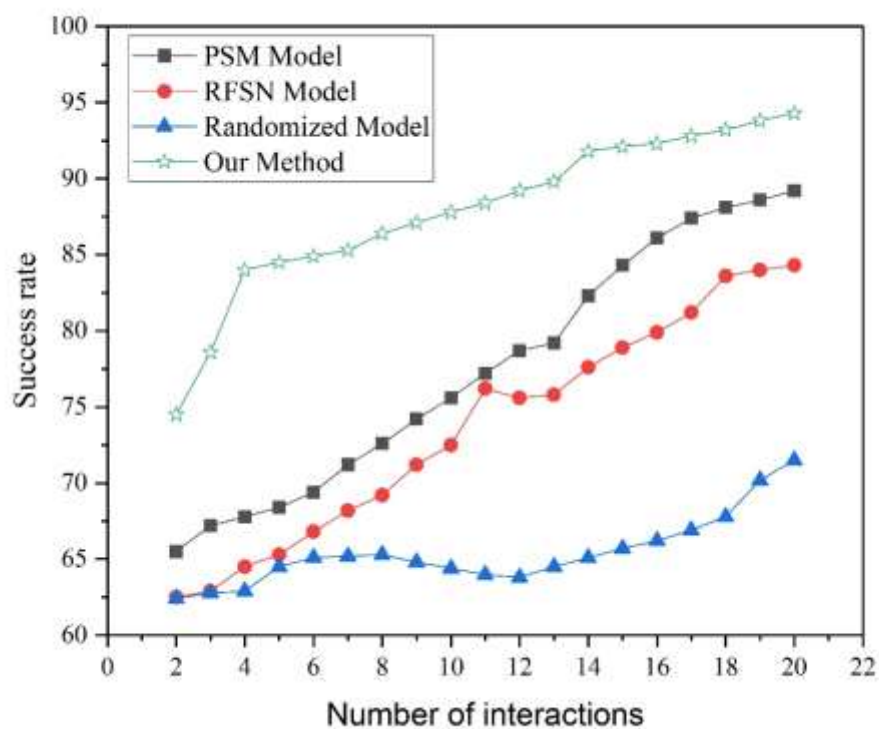


Figure 7 Malicious node ratio 20%

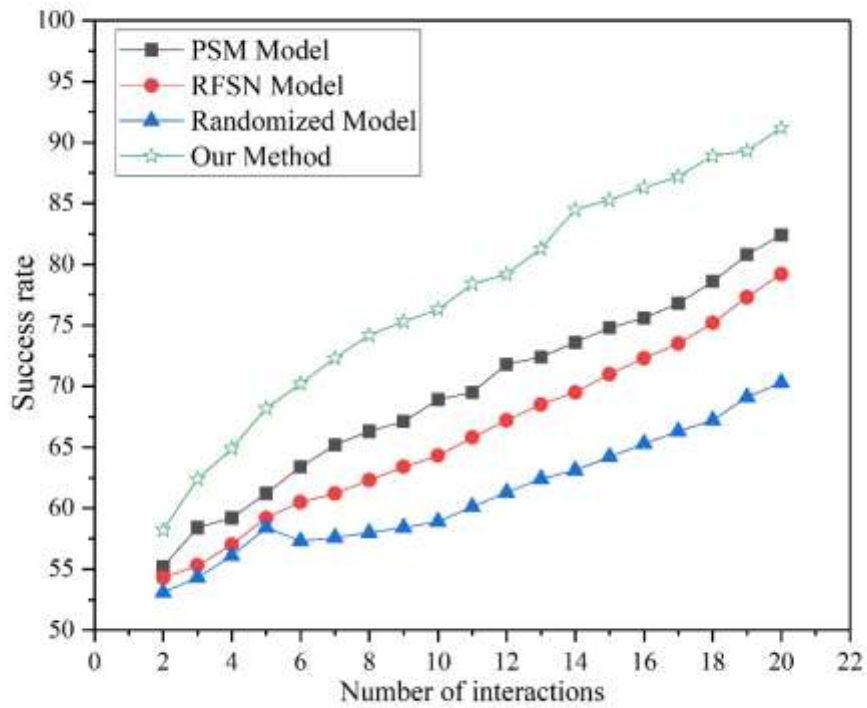


Figure 8 Malicious node ratio 30%

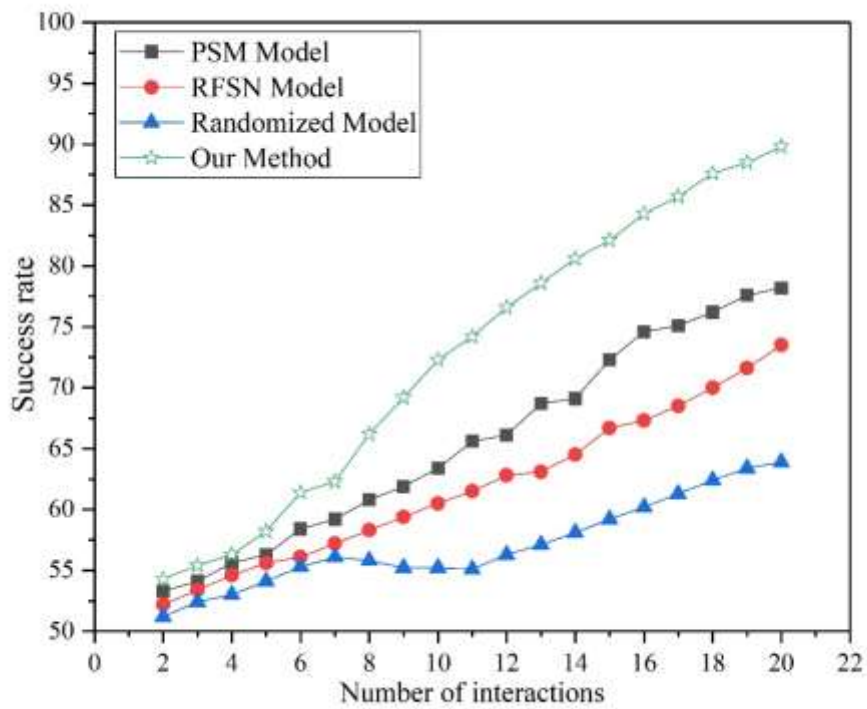


Figure 9 Malicious node ratio 40%

Figures 6-9 show the successful interaction rates under different malicious node ratios. When the malicious node ratio is 10%, with the increase of interaction times, the successful interaction rates of this model and the comparison model will become higher with the increase of interaction times, and the growth rate will gradually slow down, finally reaching a stable state. When the proportion of malicious nodes is 20% and 40%, even if the number of interactions increases, only the successful interaction rate of this model gradually increases, while the successful interaction rate of the other three comparative models does not change significantly and the trend fluctuates. Through the horizontal comparison of the three graphs, with the increase of the proportion of malicious nodes, the adaptation time of the trust model will become longer and the successful interaction rate will also decrease. From the comparison of the models in the figure, the model in this paper can quickly reach a stable state with the increase of interaction times, which shows that the model has strong adaptability and can effectively identify malicious nodes. The change of success rate of each model with the increase of malicious node ratio is shown in Figure 8.

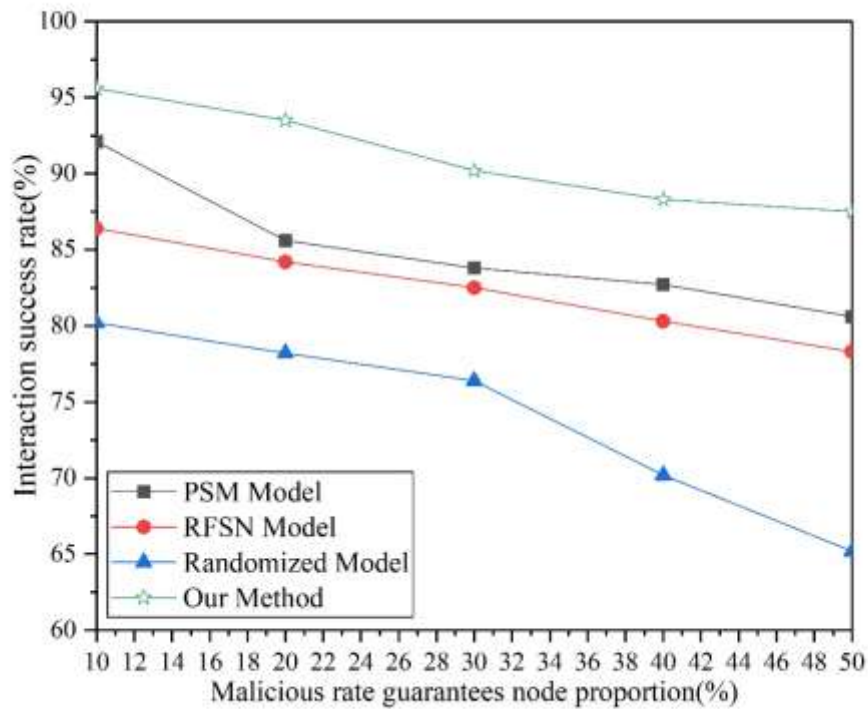


Figure 10 Scale renderings of malicious nodes successfully interacted 100 times

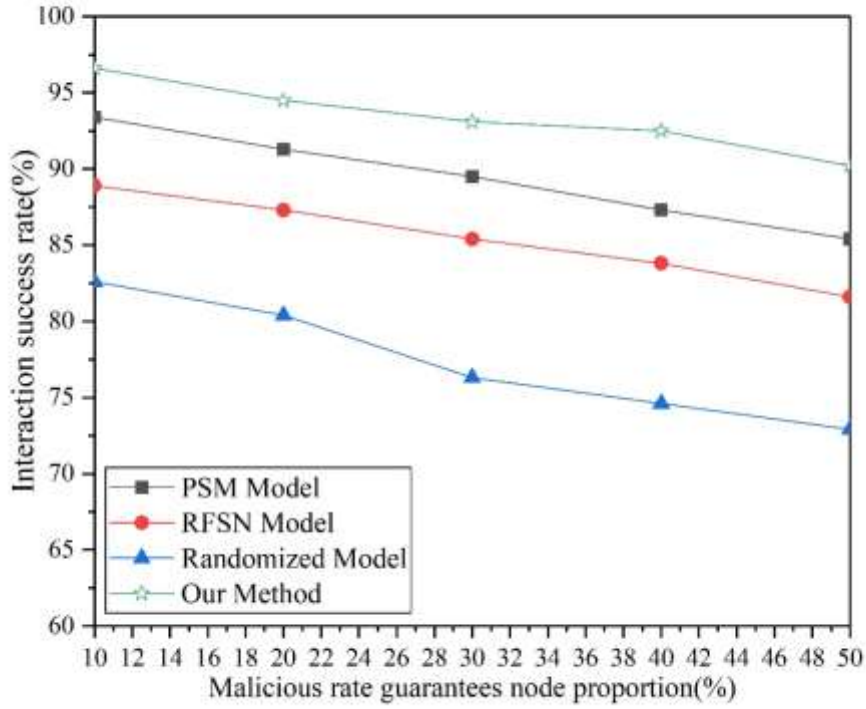


Figure 11 Scale renderings of malicious nodes successfully interacting 200 times

With the increase of malicious nodes in Figures 9-10, the successful interaction rate of each trust model shows a downward trend in different degrees, but the successful interaction rate of this model fluctuates little before and after, and is always higher than the other two models. This is because malicious nodes are filtered through multiple constraints, and the improved DFS algorithm can also eliminate the influence of redundant trust values on feedback trust results. At the same time, in the process of trust calculation, through the punishment of malicious evaluation nodes and the adaptive aggregation method of information entropy, the trust evaluation can be more accurate and better reflect the behavior of nodes.

4.3 Trusted Security Performance Comparison

4.3.1 Time overhead

Time cost is a reflection of the efficiency of the trust model to execute the work. The smaller the time cost is, the shorter the information calculation time is. The network computing performance is calculated according to the comprehensive trust time cost of the system. In the time cost test, 20% and 40% of the malicious nodes are selected for the experiment, and each group of experiments is conducted for 10 times and the average value is taken, as shown in Figure 10-Figure 11. In addition, the time spent by the outsourced user in the algorithm is far less than the time spent by solving the key directly. As a result, the edge computing server has borne a large amount of computational overhead. As the scale of the problem continues to expand, the time cost of problem encryption is longer than that of key generation and result decryption. The problem encryption stage includes two matrix multiplication, matrix and vector multiplication, and one vector addition (see Table 2).

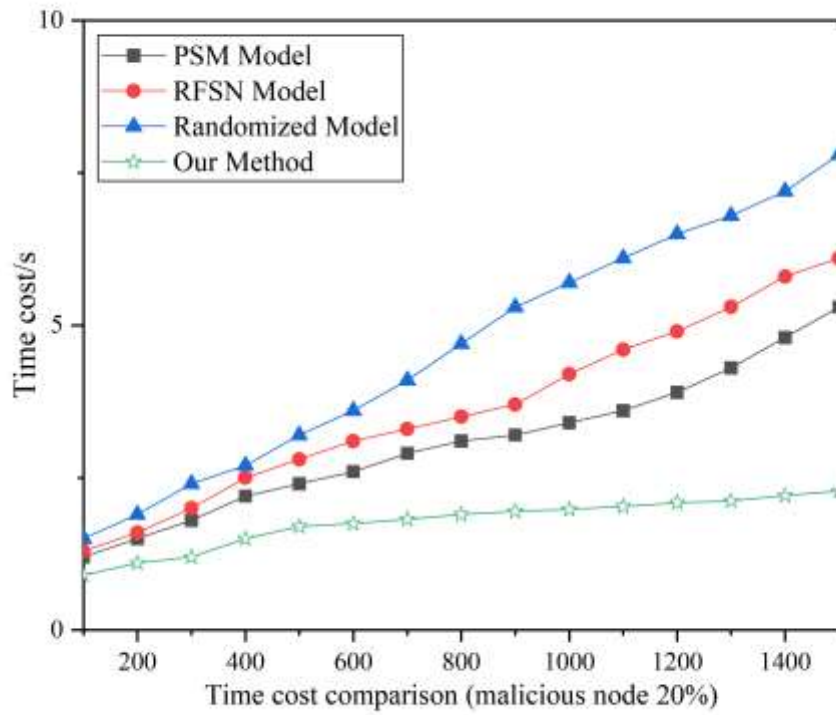


Figure 12 Time overhead comparison (malicious nodes 20%)

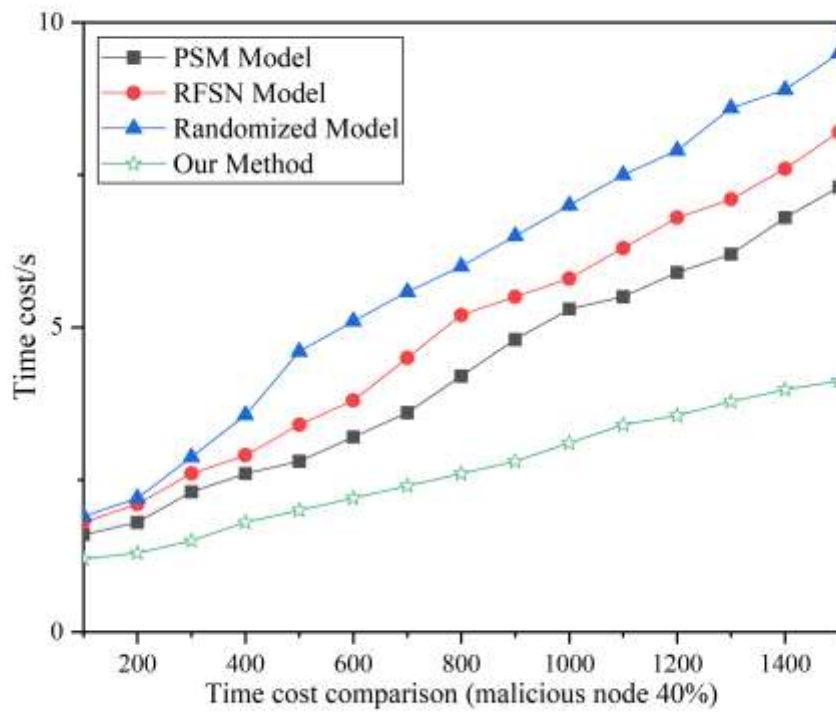


Figure 13 Time overhead comparison (malicious nodes 40%)

Figures 12-13 show the time overhead of the model from 100 nodes to 1500 nodes. When the number of nodes is small, the time cost among the three models has little difference, but with the increase of network scale, it is gradually lower than that of PSM model and RFSN model. Because the above two models use global trust iteration to update the trust information of nodes, the amount of data participating in trust aggregation increases, which leads to longer calculation time. This model optimizes the trust graph, filters nodes with low trust value in advance, deletes some redundant paths, reduces the number of trust searches and the number of trust paths, and reduces the workload of trust calculation.

Table 2 Time cost under different matrix sizes

Dimension	Key generation	Problem decryption	Result decryption	User time spent
600×400	0.03	0.02	0.002	0.05
1200×1000	0.12	0.08	0.008	0.21
1800×1600	0.3	0.25	0.002	0.54
2400×2200	0.54	0.73	0.005	2.36
3000×2800	0.87	1.59	0.007	2.67
3600×4000	1.57	1.43	0.001	3.89

Table 2 shows the time spent in the key generation phase, the problem encryption phase, and the result decryption phase. It can be seen from the table that the time of result decryption is far less than the time of problem encryption and key generation, and the result decryption only includes the product of matrix and vector and vector subtraction.

With the increase of the proportion of malicious nodes, the decryption time of different dimensions is quite different. When the proportion of malicious nodes increases, the cost time of trusted server increases with the proportion of malicious nodes. The time cost of malicious nodes under different matrix sizes is shown in Figure 14.

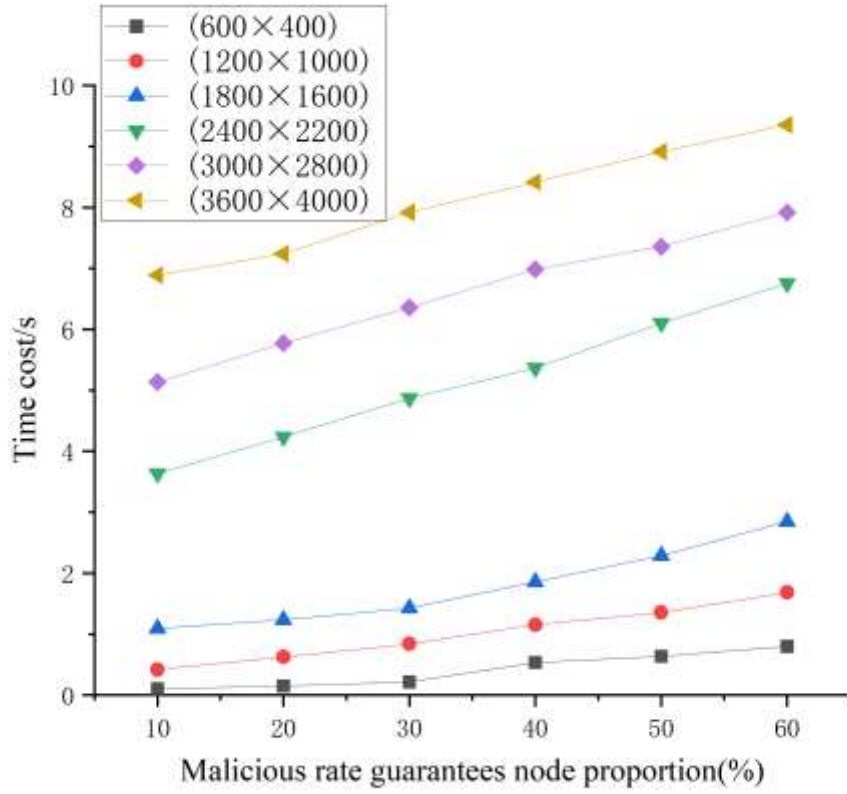


Figure 14 Time cost of malicious nodes under different matrix sizes

4.3.2 Energy overhead

In order to analyze the energy cost of this model, it is assumed that the initial energy of each node is 0.5 J, the energy consumption of each transmission and reception packet is 50nJ/bit, and the packet unit is bit, which is randomly selected within the interval of [3000, 4000]. The packet size of the three models is the same in the same interaction period. The experiment has a total of 100 running cycles, and the change of network energy in the region is shown in Figure 15.

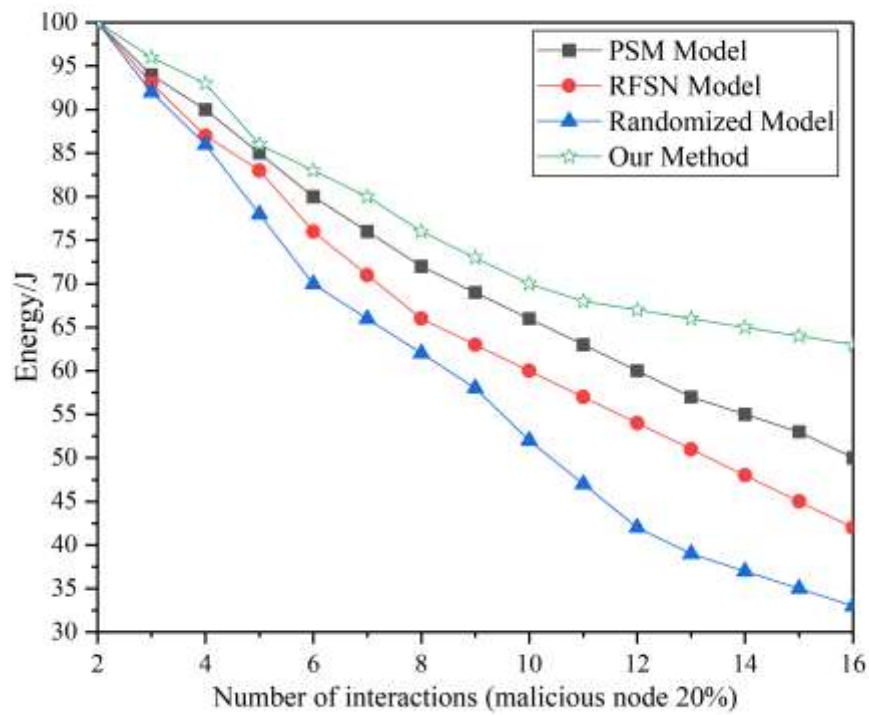


Figure 15 Transmission energy consumption (malicious nodes account for 20%)

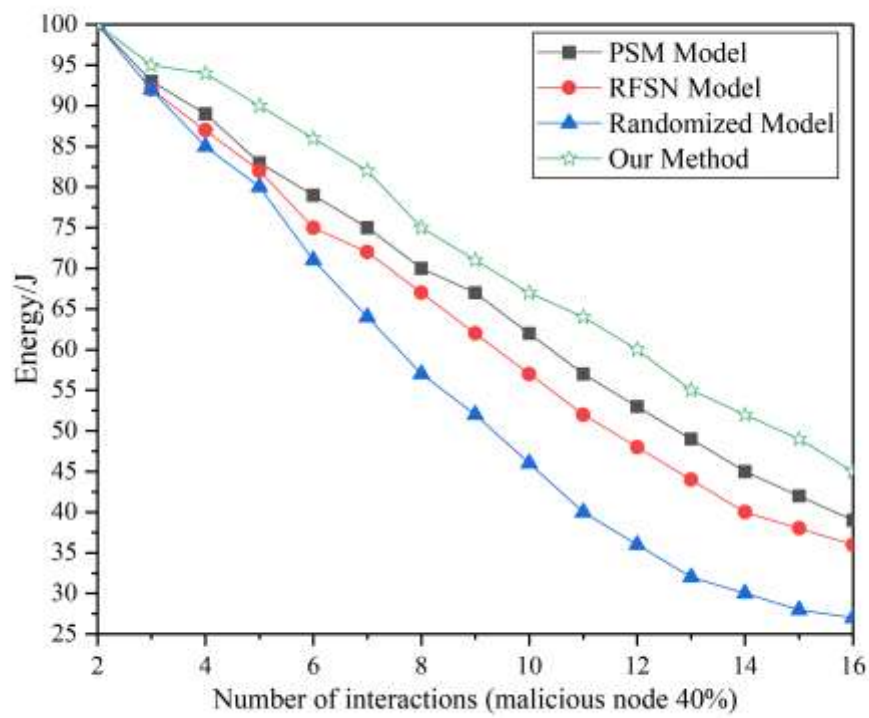


Figure 16 Transmission energy consumption (malicious nodes account for 40%)

Figs. 15-16 show that with the increase of the number of interactions between nodes, the network energy shows a downward trend, but the energy consumption rate of this model is lower than that of the other two models, and finally the remaining energy of the whole network is higher than that of the other two models. This is because the feedback trust is only stored in the edge server, and the edge node only needs to send a query request to the server, while the server only returns the feedback trust value of the target node. In addition, nodes do not need to store too much trust information, and only need to send recommended trust tables to the server regularly. Therefore, this model can effectively reduce the energy consumption in the process of trust transmission and improve the network life cycle. However, with the increase of the proportion of malicious nodes, the downward trend of energy is obvious.

Table 3 compares the time spent on each edge server with the time spent on one server solving the problem. The time spent on each edge server is much less than the time spent on solving the overdetermined equations directly. Moreover, with the continuous expansion of the matrix scale, the acceleration effect on the edge server side is more obvious.

Table 3 Comparison of server-side time spent and edge-server time spent

Dimension	Edge server time spent	Server time spent	Acceleration ratio
600×400	1.24	4	0.42
1200×1000	10.76	16.76	0.67
1800×1600	58.91	43.89	1.67
2400×2200	156.73	92.71	1.73
3000×2800	324.2	201.32	1.54
3600×4000	389.4	345.23	1.72

5 Conclusion

Edge computing migrates tasks to edge devices near the ground, provides users with real-time computing and storage functions, solves the problems of big data processing and computing load in cloud computing centers, can effectively reduce the communication bandwidth and network delay when devices cooperate, and has broad development prospects. At the same time, many users' private data are stored in the edge layer, which makes edge devices vulnerable to malicious attacks such as service fraud and data tampering. However, the edge computing environment lacks stable basic protection measures, which makes it difficult to resist the above attacks, which seriously hinders the development of edge computing. Therefore, finding the security protection method of edge environment has become an important research content in edge computing. This paper analyzes the development history and challenges of edge computing, based on the previous schemes and the limited edge computing environment, the security protection between devices is studied. The specific research work is summarized as follows: A trust security model based on distributed trust evaluation algorithm is proposed, and the trust relationship between nodes is constructed by using directed weighted graph. At the same time, the improved trust evaluation algorithm avoids loop and node detours, reduces the time of trust aggregation, and effectively improves the computational efficiency of the trust evaluation model.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Authors' contributions All the authors have participated in writing the manuscript and have revised the final version. All authors read and approved the final manuscript.

Declarations

Conflict of interest Authors declares that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants and/or animals performed by any of the authors.

Informed consent There is no informed consent for this study.

Funding Statement

This work was supported by 2022 science and technology project of State Grid Hubei Electric Power Co., Ltd, "research and demonstrational application of key technologies of industrial control security proactive defense architecture for novel power systems"(No.52153222001D).

References

- [1] H. H. Pang and K.L. Tan, "Authenticating query results in edge computing," Proceedings. 20th International Conference on Data Engineering, Boston, MA, USA, 2004, pp. 560-571, doi: 10.1109/ICDE.2004.1320027.
- [2] Sardellitti S , Scutari G , Barbarossa S . Joint Optimization of Radio and Computational Resources for Multicell Mobile-Edge Computing[J]. IEEE Transactions on Signal & Information Processing Over Networks, 2015, 1(2):89-103.
- [3] Satyanarayanan M . The Emergence of Edge Computing[J]. Computer, 2017, 50(1):30-39.
- [4] Dheerendra M, Dharminder D, Preeti Y, Y. Sreenivasa Rao, Pandi V, Neeraj K. A provably secure dynamic ID-based authenticated key agreement framework for mobile edge computing without a trusted party - ScienceDirect[J]. Journal of Information Security and Applications, 2020, 55,pp:1-15.
- [5] Wang Yifeng, Chen Xun, Ji Tian. Research on the security mechanism of content and function distribution network [J]. Cyberspace Security, 2022,13 (04): 40-47+54
- [6] [1] Chen Cheng, Xu Jun, Fang Dongkui. Internet of Things Data Security Processing Model Based on edge computing [J]. Information Technology, 2022, No.365 (04): 137-142.
- [7] Beck M T , Werner M , Feld S. Mobile Edge Computing: A Taxonomy[C]// Accepted for the Sixth International Conference on Advances in Future Internet. 2014.
- [8] M. S. Elbamby, M. Bennis and W. Saad, "Proactive edge computing in latency-constrained fog networks," 2017 European Conference on Networks and Communications (EuCNC), Oulu, Finland, 2017, pp. 1-6, doi: 10.1109/EuCNC.2017.7980678.
- [9] Zhang P , Du Y , Habetler T G. A Survey of Condition Monitoring and Protection Methods for Medium-Voltage Induction Motors. IEEE Transactions on Industry Applications, 2018, 47(1):34-46.
- [10] Wei-Xi H U, Pan Z G , Liu X Z. Digitized World Heritage Protection Methods[J]. Acta Simulata Systematica Sinica,2016,91(1-2):73-89.
- [11] Xia S , Tan M . A Trusted Network Framework Based on Integration of Trusted Identity and Access Management Method of Trusted Network Framework. Network Security Technology & Application, 2011,52,pp.56-64.
- [12] Nagarajan A, Varadharajan V. Dynamic trust enhanced security model for trusted platform based services. Future Generation Computer Systems, 2018, 27(5):564-573.
- [13] Liu Xiangang, Qiu Qin, Wang Chenguang, etc. Research on edge computing Security

Technology and Standards. Information Technology and Standardization, 2021, No.436 (04): 25-31.

[14] Z Bin. 5G edge computing Security Research and Application. Telecommunications Engineering Technology and Standardization, 2020, 33 (12): 1-7.

[15] Liu X , Jin C , Chen Z , et al. Research on Information Security Protection of Active Distribution Network Based on Trusted Computing. Journal of Physics: Conference Series, 2021, 2136(1):012013-.