Hindawi
Security and Communication Networks
Volume 2021, Article ID 9495093, 9 pages
https://doi.org/10.1155/2021/9495093

*Research Article*

# A Distributed Security SDN Cluster Architecture for Smart Grid Based on Blockchain Technology

**Ao Xiong,[1] Hongkang Tian,[1] Wenchen He,[1] Jie Zhang,[2] Huiping Meng,[3] Shaoyong Guo,[1] Xinyan Wang,[3] Xinyi Wu,[4] and Michel Kadoch[5]**

[1]*State Key Laboratory of Networking & Switching Technology, Beijing University of Posts & Telecommunications, Beijing 100876, China*
[2]*Beijing Fibrlink Communications Co., Ltd., Beijing 100070, China*
[3]*State Grid Henan Electric Power Company Information and Communication Company, Zhengzhou 450018, China*
[4]*Christian-Albrechts-Universität zu Kiel, Kiel, Schleswig-Holstein 24118, Germany*
[5]*Department of Electrical Engineering, École de Technologie Supérieure ÉTS, Université du Québec, Montreal 8871, Canada*

Correspondence should be addressed to Hongkang Tian; tianhk@bupt.edu.cn

Received 7 September 2021; Accepted 22 October 2021; Published 8 November 2021

Academic Editor: Zhili Zhou



This paper proposes a smart grid distributed security architecture based on blockchain technology and SDN cluster structure, referred to as ClusterBlock model, which combines the advantages of two emerging technologies, blockchain and SDN. The blockchain technology allows for distributed peer-to-peer networks, where the network can ensure the trusted interaction of untrusted nodes in the network. At the same time, this article adopts the design of an SDN controller distributed cluster to avoid single point of failure and balance the load between equipment and the controller. A cluster head was selected in each SDN cluster, and it was used as a blockchain node to construct an SDN cluster head blockchain. By combining blockchain technology, the security and privacy of the SDN communication network can be enhanced. At the same time, this paper designs a distributed control strategy and network attack detection algorithm based on blockchain consensus and introduces the Jaccard similarity coefficient to detect the network attacks. Finally, this paper evaluates the ClusterBlock model and the existing model based on the OpenFlow protocol through simulation experiments and compares the security performance. The evaluation results show that the ClusterBlock model has more stable bandwidth and stronger security performance in the face of DDoS attacks of the same scale.

## 1. Introduction

The smart grid manages power assets by combining the communication technology and smart devices. In recent years, the SDN (software defined network)-based smart grid has been extensively studied, and it is composed of the network controller, smart grid equipment, and a communication network [1–3]. SDN provides a centralized management control structure and a programmable data communication interface, making the smart grid easier to expand at the information level [4]. At the same time, at the network level, it is also easy to be maliciously modified by some malicious attackers, which will make the network more vulnerable to attacks [5]. The attacker will first capture the data of the SDN network to obtain the communication properties of the network, such as the upper limit of the bandwidth that the network can bear. Based on these data, a flooding attack is launched to the network, which will eventually cause the network to collapse. This is an attack method that SDN networks will easily encounter—DDoS attacks [6]. In addition to the DDoS attacks, which attack network devices, SDN networks may also encounter data packet hijacking. This situation is also difficult to be prevented and has a major impact on network security [7].

The SDN network is a network that operates based on flow rules. The information flow rules are issued to the switch through the control layer, and the switch forwards and processes data based on the flow table according to the

flow rules. The correctness and consistency of the flow rules are the key factors to ensure the network security. If the switch implements a forged flow rule, it will forward the packet incorrectly. The SDN network usually forwards data based on the OpenFlow protocol. In the forwarding process, the SDN controller first formulates a data forwarding strategy and sends it to the flow table of the switch on the data layer. The following types of attacks may be encountered during this data forwarding process: the attacker controls the SDN controller to issue the wrong flow rules to the flow table. The switch at the data layer does not have the ability to judge the correctness of the issued flow rules, and only forward data according to the flow rules. Therefore, tampering with the flow rules issued by the controller is an attack method faced by the SDN network. The attacker controls the switch to forward data packets maliciously. Therefore, one way to ensure the security of the SDN network is to strengthen the security of the SDN controller, and the other is to strengthen the security of the flow 2 to ensure consistency of flow rules [8].

In recent years, many scholars have studied how to strengthen the security of the SDN networks. Part of the research considers the hardware direction, improve the hardware security performance of the equipment to ensure the security of the network, or ensure the security of the network by increasing the complexity of the encryption algorithm. Obviously, these two methods are both costly and poorly scalable. If the network scale is large, it will consume a lot of physical resources and network computing resources. Therefore, the current popular studies are all based on network function virtualization. Through network function virtualization, some network security functions of the smart grid are realized in virtual machines, which greatly reduce costs and have higher scalability. The application scenarios are instantiated, so this article is also based on network function virtualization to simulate network devices in the form of nodes, and security functions are implemented through applications.

On the other hand, the SDN network is a centralized control network. One of the problems that the centralized control network is prone to is the single point of failure. The SDN controller in the network is hijacked by an attacker, which will directly cause the network to collapse. At the same time, in the centralized control network scenario, since the control information of the network is concentrated on the only SDN controller, the controller will bear too much burden. Then, the communication performance of the network will be limited by the performance of the controller, which greatly limits the communication performance of the network. Therefore, distributed control networks have gradually emerged. The emergence of distributed SDN controllers has solved the above problems. One is to avoid single points of failure. If one of the SDN controllers is attacked, the scope of the impact will be doubled. Another advantage is that the pressure on the control layer is reduced, and the upper limit of the performance of the entire network is increased, so that the network performance will not be limited by the performance of the control layer. This article realizes the design of the distributed SDN control network through the emerging technology of blockchain. Blockchain is a new type of distributed database in which the information of the database is stored in the form of blocks, and each block has a unique hash value identification. In addition, the blockchain has the characteristics of decentralization and does not rely on central nodes. The distributed SDN control network based on blockchain has stronger security.

This article is researched through the classic smart grid scenario of substation automation, using network topology and communication services that comply with the IEC 61850 standard. The IEC 61850 standard was originally a designated standard for substation automation and was later extended to most aspects of smart grid communication.

## 2. Related Works

In recent years, there has been much research on SDN and blockchain, especially the application of blockchain to the network architecture of SDN to improve the direction of security. Tselios and Kotsopoulos introduced blockchain into the solution of SDN and Internet of Things problems and proposed a distributed cloud architecture based on blockchain to improve security [9]. Xiao et al. conducted an overall analysis of the blockchain consensus protocol, determined the core components of the blockchain consensus protocol, and compared the performance of the consensus protocol through different performance indicators [10]. Chakrabarty and Engels proposed a smart city security system architecture that includes four basic IoT architecture modules. The architecture mainly uses a key management system to mitigate network attacks and enhance the security of the architecture [11]. Flauzac et al. proposed the concept of the SDN domain and defined the way in which multiple domains are connected to each other and the method of enhancing domain security. This is a new type of SDN architecture. By dividing the SDN control scope by domains, the pressure on the SDN controller can be reduced, and the management of nodes in the network is clearer and more direct [12]. Dorri et al. introduced blockchain to the research of smart homes to ensure the communication security of smart home networks [13].

At the same time, another research direction to enhance network security is to propose new algorithms for detecting and mitigating attacks. Based on the concept of SDx, Wang et al. proposed an IOT framework that includes SDIOT controllers, gateways, and switches and proposed algorithms to detect DDoS attacks and mitigate DDoS attacks. By calculating the cosine similarity of the transmission rate data vectors of the SDIOT switch port, it is determined whether a DDoS attack has occurred in the network [14]. Dharma et al. considered the duration of DDoS attack detection and the duration of the attack and proposed a time-based method for DDoS detection and mitigation of attacks. The detection was performed by counting the number of invalid data packets within a defined time window when the network receives DDoS attacks [15]. Establishing statistical models and applying machine learning models are also a popular research direction for detecting DDoS attacks. Kousar et al.

introduced a new framework, Apache Spark, to monitor DDoS attacks, using DSL-KDD CuP as a benchmark dataset. Experimental simulations prove that the framework has higher performance than decision trees and optimizes the process time and training time [16].

The above detection schemes are all based on a centralized SDN network. Jia and Liang proposed a distributed DDoS chain monitoring framework based on the blockchain, using AdaBoost and Random Forest, as integrated learning strategies and designed multiple indicators to monitor the framework. The experimental results show that the framework has excellent performance in detecting DDoS attacks [17]. Hussain et al. considered converting network traffic data into image data, and based on the CNN model, using ResNet for the converted image data, which greatly improved the detection accuracy [18]. Sun et al. used the BiLSTM RNN neural network to train the dataset and classify real-time traffic data and verified the accuracy of the detection algorithm through experiments [19]. Su et al. proposed a DDoS attack detection algorithm based on a mixed traffic prediction model. The algorithm uses RBF neural network technology to train and predict network traffic and sets thresholds to reduce environmental noise. Experimental simulation proves that the algorithm has a high flow prediction accuracy [20].

Bordel et al. defined a theoretical framework with high trust in IoT scenarios based on the blockchain and conducted relevant experimental verifications to prove that the framework has higher security [21]. Li and others introduced quantum technology into blockchain research, introduced the structural framework of quantum blockchain, and summarized the advantages and development prospects of this direction [22]. Fu et al. proposed an antinoise location method based on a multinorm regularization matrix using the Euclidean distance matrix to express the reconstruction problem of EDM as a multinorm regularization matrix model. In addition, it can be observed through experiments that the model has a high accuracy [23]. Li and others applied blockchain to energy transactions and proposed a high-security energy transaction system called the energy blockchain. At the same time, an optimal pricing strategy based on the Stackelberg game is proposed, which reduces the limitation of the high latency of the blockchain through a credit-based payment method [24]. Liu and others applied blockchain to the food supply chain and proposed a blockchain-based food traceability framework. The framework uses the PBFT consensus algorithm to improve the processing performance of the system [25]. This article is an attempt to use blockchain to provide security functions in a smart grid that supports SDN and to use blockchain to ensure data flow security.

## 3. ClusterBlock Architecture Design

This chapter introduces the specific details of the ClusterBlock model proposed in this article, which mainly includes four parts: secure communication architecture, distributed control strategy, network monitoring attack process, and detection algorithm, which are introduced in detail below.

*3.1. ClusterBlock Design Overview.* This section mainly introduces the secure communication architecture under the background of smart grid based on blockchain and SDN. The core idea is to use blockchain technology to improve the overall security of the network and reduce the loss caused by attacks. The blockchain ensures the consistency of the control layer strategy and the flow rules in the data layer flow table to prevent data from being tampered due to network attacks. The system architecture mainly includes three layers as shown in Figure 1.

*3.1.1. Data Layer.* One of the underlying smart grid communication equipment is a substation aggregation unit, which is used to collect data; the other is a gateway, which is used to connect to the wide area network and interact with external networks. The main equipment of the data layer is the switch, and the switch mainly processes and forwards data according to the flow table. The flow rules in the flow table are mainly issued according to the SDN controller; therefore, the security of the control layer strategy is very important.

*3.1.2. Control Layer.* In past research, SDN networks generally used a centralized SDN controller to manage the entire network, but the latest research shows that the design of distributed SDN controllers can maximize the network performance. The use of multiple controllers can not only balance the load between the device and the controller and minimize data packet loss but also enhance the safety performance of the SDN controller to avoid a single point of failure. Therefore, the SDN controller in the control layer adopts a cluster structure, and each cluster becomes an SDN domain. At the same time, in order to reduce the network delay in each SDN domain, an SDN controller is selected as the cluster head in each SDN domain and is responsible for coordinating and controlling the transmission of control commands within the network in the SDN domain. In the proposed architecture, all SDN controllers are connected to each other in a distributed blockchain manner, so that each smart grid device in the network can communicate easily and efficiently.

*3.1.3. Blockchain Layer.* The application of blockchain technology can protect the security and integrity of data. One of the disadvantages of blockchain is the large amount of computing power, which is necessary to maintain large-scale distributed ledgers. This problem can be alleviated through the design of the SDN domain and the cluster head SDN controller mentioned above. Today's networks are becoming more and more complex, and the number of nodes is increasing. The load of a conventional centralized SDN controller is too large, and it is prone to single point of failure. Dividing a huge network into several SDN domains for management can greatly reduce the complexity of the network. It can also greatly reduce the computational complexity caused by the introduction of the blockchain. On the other hand, each SDN domain adopts a cluster head

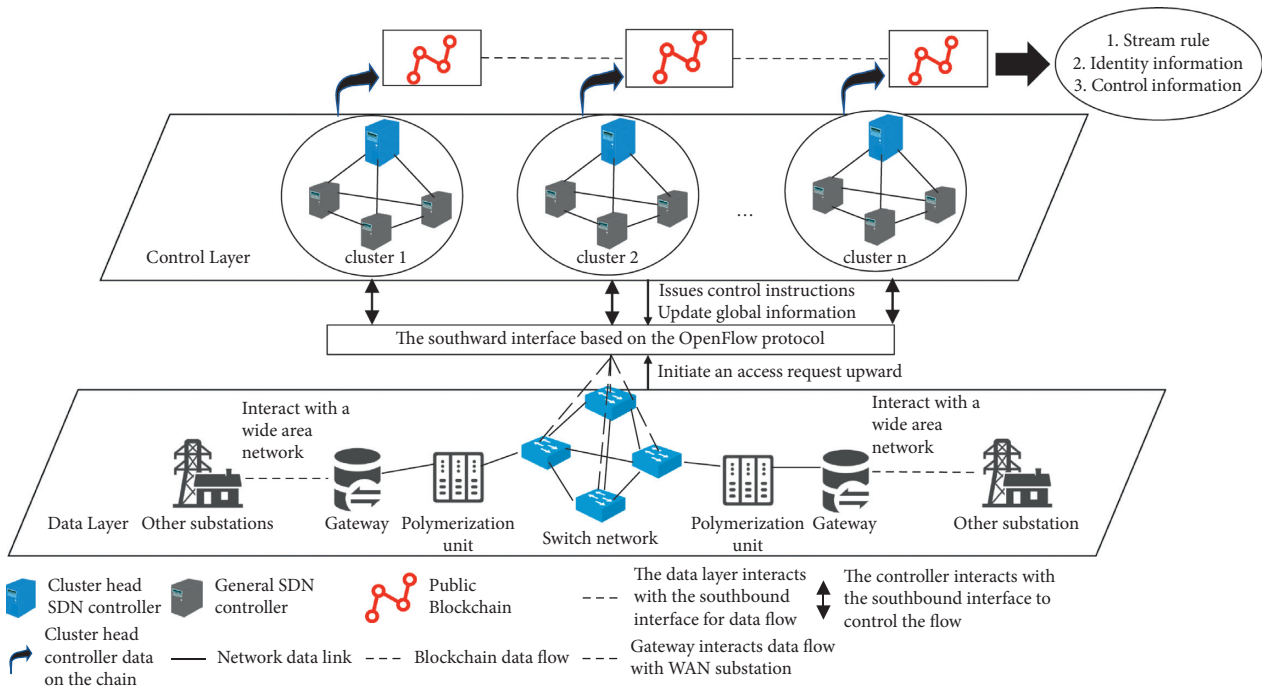


Figure 1: ClusterBlock design overview.

controller as the main controller and other controllers as a supplementary design, which is also a design that takes network security into consideration. When the cluster head controller is attacked and cannot work normally, other SDN controllers in the network can be switched to the cluster head controller to ensure the network security.

The cluster head controller manages the data on-chain problem of the blockchain. When a device in the SDN domain needs to be on-chain, it first needs to initiate an application to the control layer, and the cluster head controller verifies whether the device has been on-chain. If it has not been chained, the device is assigned an exclusive identifier of the SDN domain to identify the SDN domain to which the device belongs. In the future, when the SDN controller issues a control policy, it will only issue a control policy to the device with the SDN domain ID. When the number of devices in an SDN domain is higher than a certain threshold, the devices will also be migrated to other SDN domains. At this time, the cluster head controller of this SDN domain will detect the SDN domain with the smallest number of devices recorded on the blockchain. The identification of the SDN domain is allocated to the device, thereby completing the migration of the device in the SDN domain.

*3.2. Distributed Control Strategy Based on Blockchain Consensus.* First, the switching node of the data layer sends a request to the control layer. The request type includes a routing flow rule request and an authentication request. After receiving the request, the cluster head controller first verifies whether the SDN domain identifier of the request source node is the SDN domain to which the cluster head controller belongs and then verifies whether the node identifier of the node is recorded in the database. If the verification fails, the request is directly discarded. After the verification is passed, the cluster head controller broadcasts in the controller cluster, and the controller group conducts a consensus according to the PBFT consensus algorithm. After the consensus is completed, it is fed back to the cluster head controller. The controller stores the information that needs to be saved for this decision, such as identity information, flow rule information and controller strategy information, routing strategy, or load balancing strategy information, and stores it on the chain, and, at the same time, returns information to the data layer to update the global strategy of the switch network information, node's identity information, and flow table information.

(1) Flow table forwarding rule data uploading: when the cluster head controller sends forwarding policies to the data layer, it will issue the flow rules required by the data nodes, and the data nodes will record them in their own flow table according to the issued forwarding policies. Then, the latest flow table on the chain is stored.

(2) Network node data on-chain: after the network node of the data layer accesses the SDN domain, the cluster head controller binds the corresponding SDN domain ID and the node's ID on the chain. The content of blocks of the blockchain are as shown in Table 1.

(3) Global control strategy data uploading: the global control strategy is saved by the cluster head controller itself, such as load balancing strategy, routing strategy, etc. The policy information is also stored on

Table 1: Blocks of the blockchain.

| Variable name | Details |
|---|---|
| Publishers | The identifier of the SDN controller |
| Keys | Release identifier |
| Data | Flow rules |
| BlockTime | Block creation time |
| TxID | Block ID |

the chain through the cluster head controller to enhance the security of the network as shown in Figure 2.

The storage content of the controller's data flow rules on the blockchain is shown in Table 2:

At the same time, the switching node will also periodically send specific information to the controller, such as periodic network traffic information, data link change information that needs to be recorded, device online and offline, etc. The controller updates the control information according to the information and stores the information on the blockchain.

The innovative point of the distributed control strategy based on the blockchain consensus is that it first changes the conventional centralized SDN controller mode, turning a single controller into a controller cluster that performs distributed control through a consensus algorithm, including a cluster and several common controllers of the head controller. The controller cluster uses the PBFT consensus algorithm for consensus. The advantage of this is that if a controller is attacked and becomes a malicious node, as long as the number of malicious nodes in the controller cluster does not exceed 1/3 of the total, the result is still credible, which greatly increases the robustness of the SDN control layer.

*3.3. SDN Network Monitoring Attack Method Based on Blockchain.* This section proposes a specific monitoring attack method based on a blockchain-based SDN network. The main purpose of this method is to detect and report network threats. It is mainly divided into three stages. One stage is to build a complete network view; the second stage builds a vector network containing traffic information; the third stage detects network attacks and makes corresponding treatments based on the detected network attacks.

In the first stage, in order to conduct a comprehensive analysis of the network, the smart contract module parses all communication data packets in the network.

In the second stage, the smart contract module will analyse all the OpenFlow data packets to obtain topology data and transmission status data, extracts metadata feature sets from the topology data, obtains the network topology status and network information from the header of the OpenFlow packet, obtains flow information, as well as the flow rule information of the network, and finally construct a vector network containing the communication data flow.

In the third stage, the smart contract module monitors the data interaction, flow rules and global strategy information of the data layer, and monitors whether there are malicious nodes or whether the network is under attack. This module recognizes whether the network is under attack through the strategy specified by the application layer. The specific monitoring method is roughly given below.

When the switch receives a new data packet, the switch first checks whether there is a matching flow rule in its flow table, and then the switch sends a request to assign a flow rule for the data packet to the control layer through the southbound interface API. The control layer receives the issuing rules of the application layer, sends the rules to the data layer switch, and, at the same time, stores the flow rule information on the chain, and then the switch forwards the given service according to the issued new flow rule. At the same time, the data layer switch node accesses the blockchain and compares the flow rules therein with the flow rules issued by the controller. If the rules obtained by the two methods are different, it means that the issued rule is not correct, it is a malicious flow rule, the controller has been attacked, and the attack has been successfully detected at this time.

When the monitoring module detects a new data flow, no alarm signal will be issued. Only when the monitoring module detects that the current data flow rule is issued by a malicious control node, the flow rule is inconsistent with the chain rule, and the current flow rule cannot be specified by the application layer. When the rules are modified, the monitoring module will send out an alarm signal at this time, and other conditions will not cause the monitoring module to send out an alarm signal. At the same time, the rerouting of data flow rules will not cause the monitoring module to issue an alarm signal because the rerouting rules are generated by a trusted controller, which reduces unnecessary alarm signals in the incremental graph network. At the same time, through a custom algorithm, the reply message data of each switch on the data flow path are collected to monitor the data flow statistics, and it was compared with the blockchain data to determine whether the flow rule of the switch has deviated.

*3.4. Update Method of Flow Rules in the Blockchain Network.* Figure 3 shows that the network contains several controllers as core nodes and many switches as basic data forwarding nodes. The distributed SDN control network based on blockchain mainly includes SDN controller nodes and data nodes. The controller node is also the validation node. The controller node maintains updated data flow rule information, data node identity information, and data layer network control policy information in its own database. The data node initiates requests to other data nodes and controller nodes and responds to requests. It is mainly composed of data layer forwarding devices or switches of the smart grid. A data node of this class is defined as a request node of the blockchain network if the node's operation is a request to probe the flow rules in the flow table of another node. All other nodes that respond to the request of this node are responding nodes. Response nodes may be common data forwarding nodes or core control nodes.

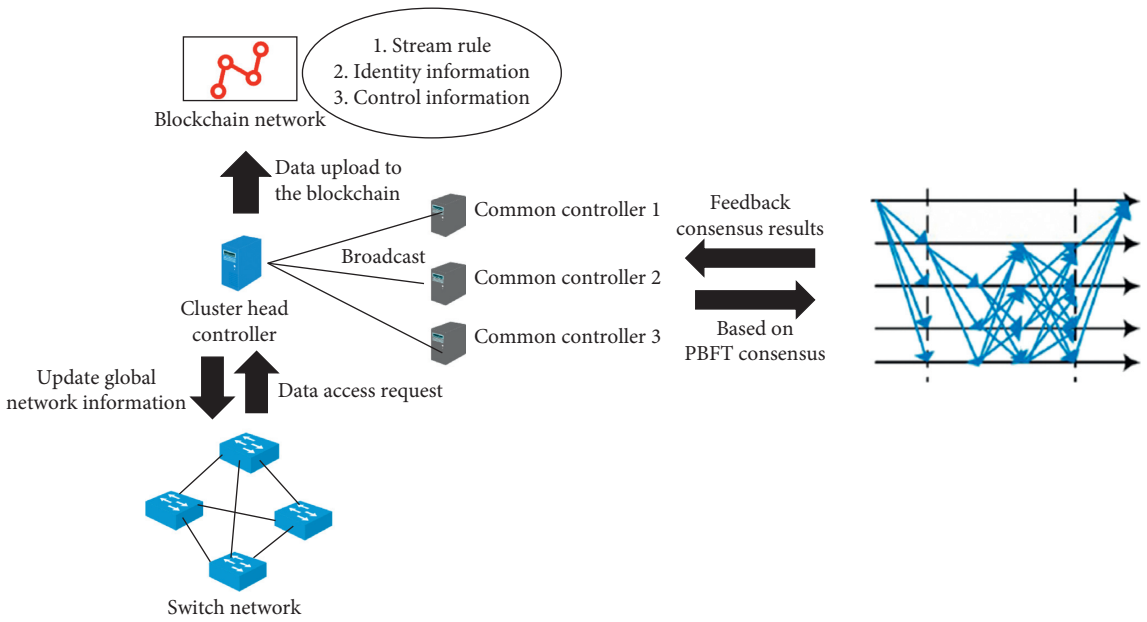


Figure 2: Distributed control strategy based on blockchain consensus.

Table 2: Storage content of the controller's data flow rules.

| Variable name | Details |
|---|---|
| ID | Rule identifier |
| TableID | Identifier of the flow table |
| DeviceID | The identifier of the device that executes the flow table rule |
| Type | Flow rule type (input/output) |
| InputPort | Input port number |
| OutputPort | Output port number |
| Priority | Flow rule priority |
| SourceMacAddress | Source MAC address of the flow rule |
| DestinationMacAddress | Destination MAC address of the flow rule |

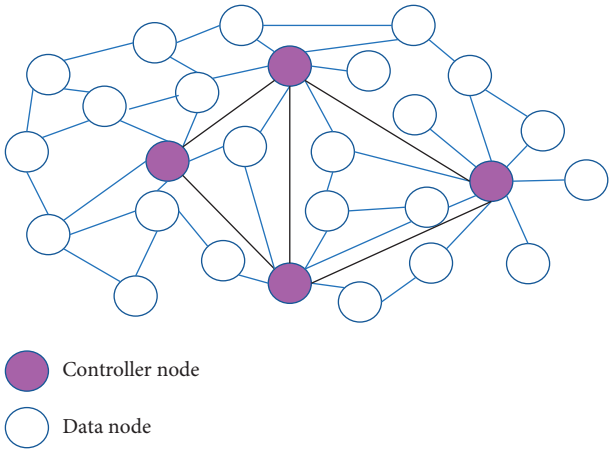


Figure 3: The overall overview of the blockchain network.

The attack detection algorithm designed in this paper mainly introduces the Jaccard similarity coefficient to compare the similarity of the data packet transmission rate of the switch port.

First, the port transmission rate is processed as a vector. Then, the data packet transmission rate of the switch port is obtained, and the port data packet transmission rate of the first $i$ switch to $r_i$ ($i = 1, 2, \ldots$) is set. The transmission rate is

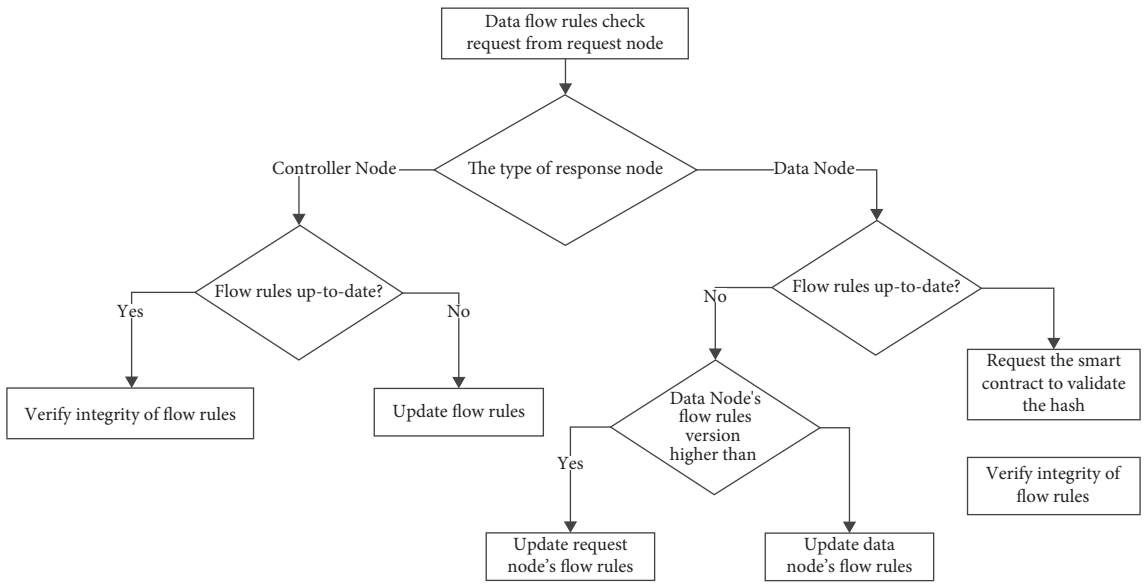


Figure 4: Update process of data flow rules in distributed SDN control network based on blockchain.

set to two sets according to the parity group in turn, and the odd group is $O = \{O_1, O_2, \ldots, O_n\} = \{r_1, r_3, \ldots, r_{2j-1}\}$. The even number is $E = \{E_1, E_2, \ldots, E_n\} = \{r_2, r_4, \ldots, r_{2j}\}$, where $n$ is the number of vectors in the set. Then, the Jaccard similarity coefficient is calculated. The Jaccard similarity coefficient is an index used to measure the similarity of two sets. The larger the Jaccard value, the higher the similarity between the two sets. The Jaccard similarity coefficient of odd and even arrays is calculated as follows:

$$J(O,E) = \frac{O \cdot E}{O + E - O \cdot E},$$

$$J(O_i,E_i) = \frac{\sum_{i=1}^{n}(O_i \times E_i)}{\sum_{i=1}^{n}O_i + \sum_{i=1}^{n}E_i - \sum_{i=1}^{n}(O_i \times E_i)},$$

$$J(r_i,r_j) = \frac{\sum_{i=2k,j=2k-1}^{2m}(r_i \times r_j)}{\sum_{i=2k}^{2m}r_{2k} + \sum_{j=2k-1}^{2m}r_{2k-1} - \sum_{i=2k,j=2k-1}^{2m}(r_i \times r_j)}, \quad (1)$$

where $m = 1, 2, \ldots, k$. By comparing Jaccard, we can get the degree of change of the port rate. When the change exceeds a certain threshold, it is judged that the network is under attack.

Figure 4 describes the update process of the data flow rules in the distributed SDN control network based on the blockchain. When the smart grid data forwarding device requests the update of the flow rules, the device acts as the requesting node. When a data packet requesting a flow rule update is circulating on the network, other nodes in the network, including all controller nodes and response nodes, will respond to the request data packet.

If the destination node is a controller node, the controller will first detect the version number of the flow table of the requesting node. If the version number of the flow table is the latest version, the blockchain database is then requested to match and compare the integrity of the flow table of the requesting node. If there is a mismatch during the detection process, the controller updates the flow table information of the node.

When the destination node is a normal node, the node will first compare whether its own flow table version number

Table 3: Simulation parameters.

| Simulation parameters | Values |
|---|---|
| Simulator | Mininet/PyEthereum |
| Type of SDN controllers | ROX |
| Number of SDN controller clusters | 3 |
| Number of SDN controllers in each cluster | 4 |
| Number of smart grid equipment node | 400 |
| Packet size | 512 byte |
| Software environment | Ubuntu 18.04 LTS |
| Routing protocol | ClusterBlock/ OpenFlow |

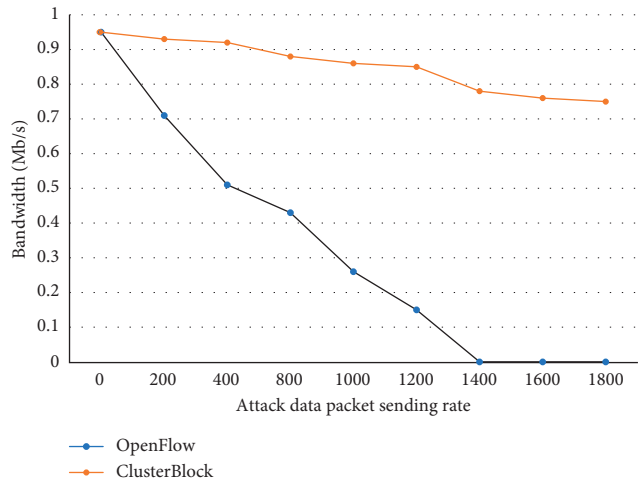


Figure 5: Comparison of safety performance under 2M bandwidth.

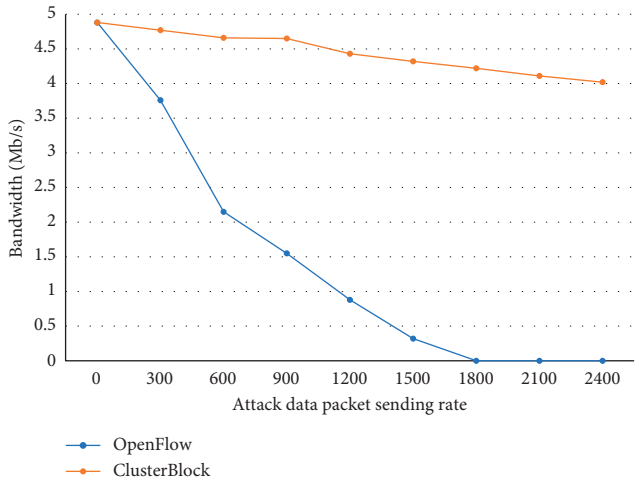


Figure 6: Comparison of safety performance under 5M bandwidth.

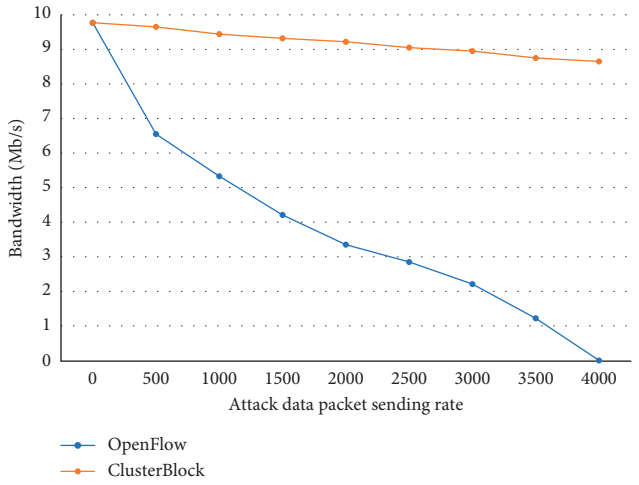


Figure 7: Comparison of safety performance under 10M bandwidth.

is the same as that of the source node. If the version number is the same, the destination node requests the smart contract module in the blockchain network to verify the source node flow table and the hash value. If the verification is successful, it is proved that the flow rule table of the source node is correct and is the latest version, and the destination node returns a response data packet to the source and the node. If the content of the flow table of the destination node and the source node are different, the destination node requests the controller to update the flow tables by itself and the source node.

## 4. Results and Discussion

In order to evaluate the safety capability of the model in this paper, this paper sets up a software and hardware test environment to evaluate and compare with the conventional SDN network based on the OpenFlow protocol. Simulation parameters are shown in Table 3.

This article uses the MININET SDN network simulator and ROX SDN controller to implement the SDN cluster based on the Ethereum platform in the Ubuntu 18.04 LTS platform and uses the PyEthereum test tool to test the functions of the blockchain part. The simulation environment is composed of 3 clusters, namely, 3 SDN domains, each SDN domain is equipped with 4 SDN controllers, among which the smart grid equipment node of the data layer uses MININET to simulate 400 data nodes to simulate the data interaction at the bottom of the smart grid.

This paper measures the bandwidth of clients launching DDoS attacks on the network, which are initiated by clients at different speeds to the switch, and evaluates the bandwidth impact of using and not using the ClusterBlock model. As shown in Figure 5, the bandwidth tested in both models is 1.9 M/S without attack. After a DDoS attack is launched, the bandwidth decreases rapidly as the attack rate increases. When the DDoS attack rate reaches 400 packets/s, the bandwidth drops to almost half. When the DDoS attack rate reaches 1400 packets per second, the network is down and data cannot be transmitted. On the other hand, using the ClusterBlock model, the bandwidth performance remains almost constant, with only a slight decrease in the whole process.

As shown in Figures 6 and 7, this paper also changes the bandwidth upper limits for testing. At the upper limits of 5M and 10M bandwidth, the ClusterBlock model has a significantly more stable performance in the face of DDoS attacks.

## 5. Conclusions

The evaluation results show that under the same scale of DDoS attack security performance, the ClusterBlock model has a more stable bandwidth and a stronger performance.

## Data Availability

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

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This work was supported by the (State Grid Corporation of China) Science and Technology Project "Key Technology and Application of New Multi-Mode Intelligent Network for State Grid" (5700-202024176A-0-0-00).

## References

[1] E. Molina, E. Jacob, J. Matias, N. Moreira, and A. Astarloa, "Using software defined networking to manage and control iec 61850-based systems," *Computers & Electrical Engineering*, vol. 43, no. 1, pp. 142–154, 2015.

[2] X. Dong, H. Lin, R. Tan, Z. Kalbarczyk, and R. Iyer, "Software-defined networking for smart grid resilience: opportunities and challenges," in *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security Workshop*, pp. 61–68, New York, NY, USA, October 2015.

[3] T. Mahmoodi, V. Kulkarni, W. Kellerer, and P. Mangan, "VirtuWind: virtual and programmable industrial network prototype deployed in operational wind park," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 9, pp. 1281–1288, 2016.

[4] J. Wu, M. Dong, and K. Ota, "Big data analysis based security cluster management for optimized control plane in software-defined networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 27–38, 2018.

[5] E. U. Haq, H. Xu, L. Pan, and M. Irfan Khattak, "Smart grid security: threats and solutions," in *Proceedings of the 13th International Conference on Semantics, Knowledge and Grids*, pp. 188–193, Beijing, China, August 2017.

[6] V. Dehalwar, A. Kalam, M. L. Kolhe, and A. Zayegh, "Review of detection, assessment and mitigation of security risk in smart grid," in *Proceedings of the 2nd International Conference on Power and Renewable Energy*, pp. 1077–1081, Chengdu, China, September 2017.

[7] E. Y. Dari and M. Essaaidi, "An overview of smart grid cyber-security state of the art study," in *Proceedings of the 3rd International Renewable and Sustainable Energy Conference*, pp. 1–7, Marrakech, Morocco, December 2015.

[8] M. Ojo, D. Adami, and S. Giordano, "A sdn-IOT architecture with nfv implementation," in *Proceedings of the Globecom Workshops (GC Wkshps), 2016 IEEE*, pp. 1–6, IEEE, Washington, DC, USA, December 2016.

[9] I. P. C. Tselios and S. Kotsopoulos, "Enhancing sdn security for IOTrelated deployments through blockchain," in *Proceedings of the Third International Workshop on Security in NFV-SDN*, Berlin, Germany, December 2017.

[10] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[11] S. Chakrabarty and D. W. Engels, "A secure IOT architecture for smart cities," in *Proceedings of the Consumer Communications & Networking Conference (CCNC), 2016 13th IEEE Annual*, pp. 812-813, IEEE, Las Vegas, NV, USA, January 2016.

[12] O. Flauzac, C. Gonzalez, A. Hachani, and F. Nolot, "Sdn based architecture for IOT and improvement of the security," in *Proceedings of the 2015 IEEE 29th International Conference on*, pp. 688–693, IEEE, NW Washington, DC, USA, March 2015.

[13] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IOT security and privacy: the case study of a smart home," in *Proceedings of the 2ND IEEE PERCOM Workshop On Security Privacy And Trust In The Internet of Things*, Hawaii, HI, USA, March 2017.

[14] J. Wang, Y. Liu, W. Su, and H. Feng, "A DDoS attack detection based on deep learning in software-defined Internet of things," in *Proceedings of the 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pp. 1–5, Victoria, BC, Canada, October 2020.

[15] N. I. G. Dharma, M. F. Muthohar, J. D. A. Prayuda, K. Priagung, and D. Choi, "Time-based DDoS detection and mitigation for SDN controller," in *Proceedings of the 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 550–553, Busan, Korea, August 2015.

[16] H. Kousar, M. M. Mulla, P. Shettar, and D. G. Narayan, "DDoS attack detection system using Apache spark," in *Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–5, Coimbatore, India, January 2021.

[17] B. Jia and Y. Liang, "Anti-D chain: a lightweight DDoS attack detection scheme based on heterogeneous ensemble learning in blockchain," *China Communications*, vol. 17, no. 9, pp. 11–24, 2020.

[18] F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad, and G. A. Shah, "IoT DoS and DDoS attack detection using ResNet," in *Proceedings of the 2020 IEEE 23rd International Multitopic Conference (INMIC)*, pp. 1–6, Bahawalpur, Pakistan, November 2020.

[19] W. Sun, Y. Li, and S. Guan, "An improved method of DDoS attack detection for controller of SDN," in *Proceedings of the 2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET)*, pp. 249–253, Beijing, China, August 2019.

[20] Y. Su, X. Meng, Q. Meng, and X. Han, "DDoS attack detection algorithm based on hybrid traffic prediction model," in *Proceedings of the 2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, pp. 1–5, Shandong, China, September 2018.

[21] B. Bordel, R. Alcarria, D. Martín, and Á. Sánchez-Picot, "Trust provision in the internet of things using transversal blockchain networks," *Intelligent Automation & Soft Computing*, vol. 25, no. 1, pp. 155–170, 2019.

[22] C. T. Li, Y. S. Xu, J. H. Tang, and W. J. Liu, "Quantum blockchain: a decentralized, encrypted and distributed database based on quantum mechanics," *Journal of Quantum Computing*, vol. 1, no. 2, pp. 49–63, 2019.

[23] X. Fu, W. Liu, Z. Li, L. Chen, and R. Wang, "Noise-tolerant wireless sensor networks localization via multi-norms regularized matrix completion," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2409–2419, 2018.

[24] Z. Li, J. Kang, Y. Rong, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 14, pp. 3690–3700, 2018.

[25] J. Liu, X. Sun, and K. Song, "A food traceability framework based on permissioned blockchain," *Journal of Cyber Security*, vol. 2, no. 2, pp. 107–113, 2020.