Enhanced DDoS Defense in SDN: Double-Layered Strategy with Blockchain Integration

Jialin Tian
The Computer and Information College
Fujian Agriculture And Forestry University
Fuzhou, China
tianjialin@fafu.edu.cn

Zhaogang Shu*
The Computer and Information College
Fujian Agriculture And Forestry University
Fuzhou, China
zgshu@fafu.edu.cn
*Corresponding author

Shuwu Chen*
The Computer and Information College
Fujian Agriculture And Forestry University
Fuzhou, China
chenshuwu@fafu.edu.cn
*Corresponding author

Haihui Xie
The Computer and Information College
Fujian Agriculture And Forestry University
Fuzhou, China
xiehh@fafu.edu.cn

Xiaolong Liu
The Computer and Information College
Fujian Agriculture And Forestry University
Fuzhou, China
xlliu@fafu.edu.cn

Caiyu Qiu
The Computer and Information College
Fujian Agriculture And Forestry University
Fuzhou, China
qiucaiyu@fafu.edu.cn

2024 13th International Conference on Communications, Circuits and Systems (ICCCAS) | 979-8-3503-8627-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCCAS62034.2024.10652857

***Abstract*—With the development of technologies such as cloud computing, big data, and the Internet of Things (IoT), Software-Defined Networking (SDN) has emerged as a novel network architecture in today's Internet era. It can separate the control plane from the data plane, allowing rapid packet forwarding in the Internet through a centralized controller. However, SDN environments are vulnerable to traditional Distributed Denial of Service (DDoS) attacks. This paper proposes a new dual layer strategy to try to mitigate the question. First, by using blockchain technology and smart contract in the northbound interface to store the flow tables required for SDN networks, security is increased. Then, we use the Token Bucket algorithm and Time Window algorithm to build the first-tier strategy to defend against obvious DDoS attacks. To detect unobvious DDoS attacks, we design the second-tier strategy that uses a composite data feature correlation coefficient calculation method and the Isolation Forest algorithm to perform binary classification on data, thereby identifying abnormal traffic. We use the currently publicly available DDoS dataset CIC-DDoS2019 for experimental verification. The results show that using this strategy in SDN networks results in an average deviation of data Round-Rip Time (RTT) approximately 38.86% lower than in the original SDN networks without this strategy. Additionally, the accuracy of DDoS attack identification reaches 91.29%. This means that with the implementation of this strategy, DDoS attacks can be effectively identified without compromising the stability of data transmission in SDN network environments.**

Keywords—Software Defined Networks, Distributed Denial of Service, Blockchain, Machine learning.

I. INTRODUCTION

In today's rapidly evolving landscape of internet technology, the demand for flexibility and efficiency in the network architecture is becoming increasingly prominent. As a significant transformation in the field of networking, Software-Defined Networking (SDN) technology come into being. This network architecture is designed based on the separation of the data forwarding plane and the control plane. Its structure expands the possibilities for network design and management, consolidating the myriad operations required for network control into a software component known as the SDN controller. By employing centralized control, it greatly simplifies the execution of new services, selection of management policies, and the reconfiguration of networks from a software perspective. The northbound interface facilitates communication between the control plane and the SDN controller, enabling upper-layer applications within the plane to send commands and requests to the controller. Between the SDN controller and the data plane, the southbound interface allows the controller to send control commands to network devices within the plane and receive status information from them. The network architecture built by SDN is a programmable network [1]. At present, there are many mature network protocols developed based on SDN. The OpenFlow protocol is one of the common protocols [2].

Distributed Denial of Service (DDoS) attacks disrupt legitimate user access by flooding the target with a large volume of malicious data packets, depleting available resources distributed across compromised nodes [3]. Research indicates that DDoS poses the greatest threat to SDN. Such attacks exploit SDN networks by exhausting switch flow table caches, controller resources, or blocking link bandwidth. The escalating threat of DDoS to network operators and internet service providers underscores its status as one of the most formidable challenges in cybersecurity, with no perfect solution currently available. Particularly with the advent of the 5G era, the exponential growth in the number of network devices presents a significant threat to DDoS defense.

Consequently, exploring methods to defend against DDoS in SDN is of paramount importance.

Although SDN enables easy implementation of inter-domain data exchange within constructed network topologies, the presence of DDoS attacks, often involving overwhelming volumes of attack data, may lead to significant wastage of network resources for inter-domain data exchange in SDN. However, with the emergence of new technologies, the utilization of blockchain paves the way for low-cost, highly flexible, and efficient solutions for cross-domain collaboration. Blockchain not only leverages smart contracts and its inherent consensus mechanisms to address challenges in information exchange across multiple network domains but also prove effective in providing decentralized collaboration in trustless network environments. Blockchain technologies such as Bitcoin, Ethereum, and distributed ledgers demonstrate high levels of security and transparency in various domains. Their application in SDN networks for DDoS defense holds significant promise for development.

Amezcua Valdovinos et al. [4] summarize the research work on SDN and outlines the issues that need to be addressed in combating DDoS attacks. Lian et al. [5] propose FRChain, a blockchain-based SDN data forwarding security solution, ensuring the security of SDN data forwarding and detecting suspicious nodes in the network by storing flow rules in the blockchain. Zeng et al. [6] utilize blockchain to establish global trust relationships and propose a reputation evaluation mechanism to guarantee security and routability. Li et al. [7] introduce the blockCSDN framework for information management and intrusion detection in SDN networks. Ma et al. [8] propose a DDoS attack detection algorithm using heterogeneous integrated feature selection and random forest algorithm. Jian et al. [9] present an information-theoretic DDoS detection method and utilize smart contracts in blockchain. Marvi et al. [10] propose a hybrid approach combining feature selection and extraction using unsupervised machine learning methods to detect the DDoS attacks.

Through understanding of relevant research, we find that the existing DDoS attack defense strategies combine with blockchain essentially rely on subjective judgment through blacklist and whitelist settings. Additionally, the use of blockchain may also affect the stability of data transmission in SDN networks. Therefore, this work proposes double-layered strategy with blockchain Integration. The contributions of this work are:

- We regard DDoS attack detection as a binary classification problem, dividing data transmission in SDN networks into benign network flows and malicious network flows. To simulate real-world scenarios as closely as possible, we utilize the CIC-DDoS2019 [11], a public DDoS dataset containing 77 statistical features based on network flows for 11 different types of DDoS attacks.
- For combating DDoS attacks, we employ the FISCO BCOS consortium blockchain as the container for storing flow tables in SDN at the northbound interface of POX-based SDN switches. Additionally, we devise a cross-domain data flow transmission scheme based on smart contracts, ensuring the security of stored flow tables while meeting the high-frequency access requirements of SDN controllers.
- In the DDoS detection algorithm section, we adopt a hybrid approach combining unsupervised machine learning algorithms, namely the Isolation Forest algorithm, with time series and token bucket algorithms, to design a two-tier detection strategy to address the complexity of real-world DDoS attack scenarios. Furthermore, we discuss the calculation method for feature value coefficients that would better facilitate network flow feature selection when utilizing the CIC-DDoS2019 dataset as the basis for the analysis.

II. METHODOLOGY

This section will provide a detailed overview of the proposed strategy for mitigating DDoS attacks, which is mainly divided into two parts: the handling of flow table data by blockchain and DDoS detection algorithms. At the northbound interface of SDN, we deploy the FISCO BCOS consortium blockchain. Inter-domain, smart contracts are utilized to collect relevant information about data within periodic intervals. Intra-domain, the first layer of mitigation strategy, detects obvious flood-type DDoS attacks. The second layer of mitigation strategy, incorporating the Isolation Forest algorithm, is used to detect less perceptible DDoS attacks.

TABLE I. VARIABLES IN THE SMART CONTRACT

Variables	*Explanation*
user	***Types of accounts for external participants***
collaboratorsAddr	***Addresses of collaborative strategy participants***
collaborators	***The set of mappings from the addresses of collaborating participants to the corresponding structures***
data	***A network data message***
dataset	***Refers to all network stream data currently stored in the blockchain***
index	***Refers to the serial number of the target object***

A. Flow Table Data Processing

Similar to the traditional routing and forwarding process, in SDN network topology, when the data plane sends forwarding data requests to the SDN controller through the southbound interface, the SDN controller also needs to access the flow table to accurately control the data plane to complete a forwarding action. In this work, we deploy the FISCO BCOS consortium blockchain and deploy contracts to it. Then, specific SDN controllers are authorized and authenticated within the blockchain. After successful authentication, the authorized SDN controllers interact with the FISCO BCOS client through blockchain APIs at the northbound interface to store flow table information. The utilization of blockchain in the collaboration process enables transparency and openness. In summary, smart contracts should fulfill the following functions:

- The owner of the smart contract needs to possess the capability to determine whether a participant is authorized to access the blockchain.

- Authorized participants of the contract can update local flow table information.

This work proposes a specific smart contract design. Smart contracts are implemented using the Solidity programming language. Table 1 lists variables that can only be contracted and their corresponding explanations.

IsCollaborator(user): The main role of this function takes the account of an external participant (i.e., user) as input to determine whether the external account is authorized and verified. If the participant's account exists in the smart collaboration contract, it returns true; otherwise, it returns false. This function's algorithm can be found in Algorithm 1.

Algorithm 1: IsCollaborator

```
Input: user
Output: bool
if collaboratorsAddr.length == 0 then
    return False;
else
    if collaboratorsAddr[collaborators[user].index]
    == user then
        return True;
    else
        return False;
    end
end
```

GetIndexByData(data): The central idea of this function is to compare the data information that authorized and verified users want to update with the data already existing in the blockchain. If the data exists in the dataset, it returns its corresponding index; otherwise, it returns null. During the implementation, the timestamp when the data is stored in the blockchain serves as the index, and characteristics such as the source MAC address serve as the data content. When searching in the dataset, it compares the source MAC address in the data content and returns the corresponding index. The algorithm of this function can be found in Algorithm 2.

Algorithm 2: GetIndexByData

```
Input: data
Output: index
for index ← 0 to dataset.length do
    if dataset[index] == data then
        return index;
    end
end
return null;
```

B. Detection Strategy

This strategy consists of two layers. The first layer strategy targets ordinary DDoS attacks by continuously reading network flow data stored in the blockchain over a period of time and analyzing the frequency of certain features in this data to determine whether it should be classified as a DDoS attack source. In this work, the source MAC address and the timestamp stored in the blockchain are used as the analysis objects. Using a time-series algorithm, it checks whether the timestamp of each data falls within a set time interval, reads the source MAC address of the data within the interval, and if the number of occurrences of the same source MAC address within the interval exceeds the threshold set according to the network topology, it is considered a suspected DDoS attack source. The token bucket algorithm is then applied to limit the data generated by this attack source by setting the total number of tokens in the token bucket and the token issuance rate (one token allows the attack source to perform one data forwarding), ensuring that it does not affect the forwarding of normal data flows. The principle of updating the number of tokens in the token bucket is described by Formula (1). $B(t)$ represents the number of tokens in the token bucket at t (it is the current time in seconds), $B\ (t\ -\ 1)$ represents the number of tokens in the token bucket at the previous second, R represents the token generation rate of the token bucket, i.e., the number of tokens generated per second, and Int represents the time interval between the current time and the last update of the token bucket quantity.

$$B(t) = B\ (t\ -\ 1) + R\ \times Int \tag{1}$$

The second layer of the strategy targets DDoS attacks that are not easily detected. We used the CIC-DDoS2019 dataset as data support. This dataset contains 77 statistical features based on network flows for 11 different types of DDoS attacks. Due to limited computing resources and the fact that the Isolation Forest algorithm used in this strategy is an unsupervised learning algorithm, the training set does not require label annotation, so only data under the SYN type is used to create the training set (which only contains normal traffic) and the test set (which contains both normal traffic and SYN abnormal traffic). According to the logic of the second layer DDoS detection strategy, this strategy analyzes the features of the data flows stored in the blockchain over a period of time, and uses the Isolation Forest algorithm to classify the data flows into two types: normal traffic and abnormal traffic. Then, the token bucket algorithm is invoked to throttle the data transmitted by the source MAC addresses of the abnormal traffic. To determine the sequential order of analyzing data features and the quantity of analyzed data features, we utilize the variable X to record all feature labels in the dataset. By employing a composite feature selection strategy, we computed the feature correlation coefficient sets, X-scores. Then, we use the recursive feature elimination method to sequentially eliminate features with low coefficient values. Finally, by comparing the accuracy of the analysis results of the Isolation Forest algorithm, the optimal feature combination and analysis order are determined as the basis for selecting feature values for subsequent test sets (i.e., network flow data stored in the blockchain for each time period). We also compare the final accuracy with the accuracy of using a single mainstream feature selection algorithm.

$$X_{\text{norm}} = \frac{X_{\text{score}} - X_{\min}}{X_{\max} - X_{\min}} \tag{2}$$

In the comparison process, to present the feature value coefficients obtained by various feature selection algorithms in a comparable manner, we employ min-max normalization. We

first extract the maximum and minimum values, $X_{\max}$ and $X_{\min}$ from the feature value coefficient set X_{score} and then normalized X_{score} according to Formula (2), converting it into a set of scores, X_{norm}, within the range [0, 1] for this feature selection algorithm. Due to the adoption of a composite feature selection method in this work for the calculation and selection of data feature coefficients, the screening method is represented in Table 2.

TABLE II. COMPOSITE FEATURE SELECTION STRATEGY

Algorithm	*Definition*
Mutual Information	***Reciprocal information measures how much information the presence/absence of a feature has to make a correct prediction for Y.***
Random Forest	***The generalization of the model is improved by constructing multiple decision trees to reduce the risk of overfitting a single decision tree.***
Recursive Feature Elimination	***By recursively training the model and removing unimportant features, the optimal subset of features is selected.***

III. RESULTS AND DISCUSSION

In this chapter, We'll set up an SDN topology and observe ping connectivity between hosts to assess the impact of our defense strategy on SDN switches. For DDoS attack detection, the first layer of defense is 100% effective against flooding attacks surpassing its threshold. Less detectable attacks will be simulated using the CIC-DDoS2019 dataset, stored in the blockchain, and processed by the second-layer strategy to evaluate DDoS defense accuracy.

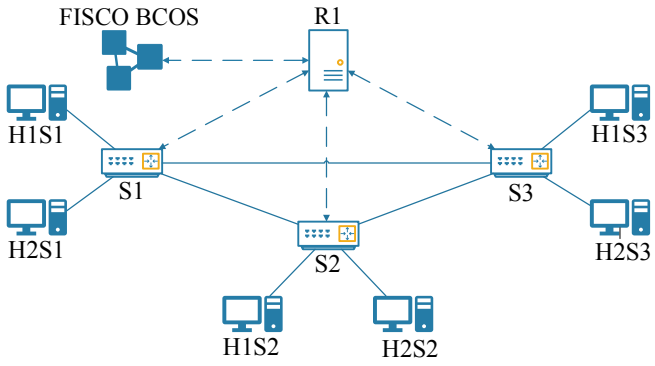

Fig. 1. Software defined network topology.

A. Simulation Environment and Network Topology

This study is conducted on the Ubuntu 20.04 system, using Mininet [12] as the environment simulation tool for building network topologies. We chose the Pox [13] as the controller for SDN and use Scapy as the traffic generation tool to simulate hosts in the SDN network. In Fig. 1, our topology consists of 1 FISCO BCOS blockchain, one Pox controller, 3 OpenvSwitch switches, and 6 hosts. Each switch and 2 hosts form an SDN subnet. The SDN controller serves as a node of the FISCO BCOS [14] blockchain, which is built using the Python SDK, deploys smart contracts through the WeBASE [15] component, and compiles them using the Solidity languages to form corresponding application ports. When the SDN topology runs for the first time, switches perform broadcast addressing, and the SDN controller records the flow data sent by switches through their ports (including their original MAC addresses, etc.). This information is stored in the blockchain as flow table information. Subsequent data forwarding prioritizes sending requests to the SDN controller to obtain flow table information in order to complete forwarding actions.

B. Comparison of Flow Table Data Forwarding with the Original Controller

In this work, we propose a detection and mitigation strategy for DDoS attacks, which mainly affects the normal forwarding process of the controller in the network flow table storage. To assess the extent of this impact, we compare the stability of SDN controllers using our proposed strategy (i.e., the Pox controller integrated with FISCO BCOS blockchain-based flow table storage structure at the northbound interface of the SDN controller) with the original Pox controller that does not use our strategy.

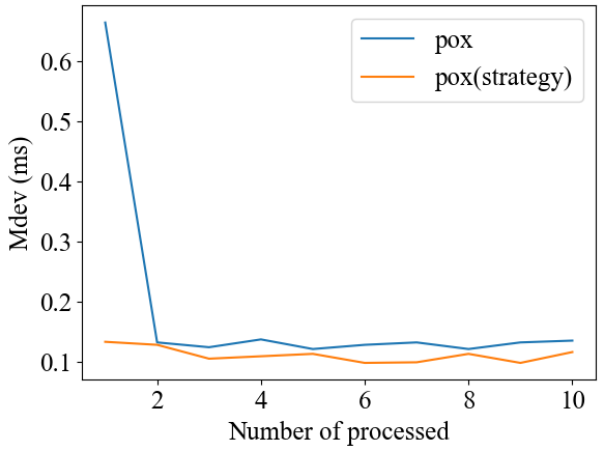

Fig. 2. Line graph of network stream transmission stability.

We use the mdev value, representing the mean deviation of round-trip time (RTT) when transmitting data between hosts, as a stability indicator. In the experiments, we conduct 10 sets of comparisons by pinging between two hosts, H1S1 (the host H1 under switch S1) and H2S3 (the host H2 under switch S3), in the same network topology environment. We collect the mdev values from 10 consecutive ping connections between the hosts in each set, resulting in a total of 10 sets of comparisons, and generate a comparison line chart as shown in Fig. 2. It is evident that the mdev value of the Pox controller, using blockchain s the flow table storage container, for controlling data transmission in the network topology is generally lower than that of the original Pox controller. The mean mdev value is 0.1122ms, which is lower than the original Pox's mean mdev value of 0.1835ms. The former's numerical decrease is approximately 38.86% compared to the latter. This is because the strategy employs a financial blockchain, which is inherently suitable for handling massive data and emphasizes transaction stability. Additionally, our design smart contracts efficiently handle data processing. Therefore, when the SDN controller accesses the blockchain via northbound interfaces, stable access is consistently maintained. As a result, implementing this strategy does not affect normal data

forwarding within the topology and may even enhance the stability of network flow transmission to some extent.

C. DDoS attack Detection Accuracy

In pattern recognition and information retrieval, binary classification metrics based on true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) help measure system performance. In our study on DDoS attack detection, we utilized the SYN subset of the CIC-DDoS2019 public dataset. We trained our model on 70336 instances of normal traffic and tested it on 907 instances of mixed normal and abnormal traffic. These metrics are crucial for evaluating our model's performance in correctly identifying DDoS attacks. TP represents correctly identified normal traffic, while TN represents correctly identified anomalous traffic, FP represents normal traffic wrongly identified as anomalous, while FN represents anomalous traffic wrongly identified as normal. These parameters are used to calculate accuracy (ACC), a key metric for performance evaluation. The accuracy of correctly detecting attacks is defined as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

Based on Formula (3), we separately calculate and compare the accuracy of the final DDoS detection strategy after employing Pearson correlation coefficient, kernel methods, distance correlation, Spearman correlation coefficient, lasso technique, and our composite feature selection algorithm as data preprocessing steps, as shown in Table 3. From the table, it is evident that the composite feature selection algorithm used in this strategy achieves a final accuracy of 91.29%, significantly ensuring the accuracy of DDoS detection in the final strategy. This is attributed to the strategies proposed in this work, especially the effective feature selection of composite feature values utilized in detecting network flow data in the second-tier strategy, as well as the Isolation Forest algorithm's effective binary classification of data.

TABLE III. COMPARISON OF PREDICTION ACCURACY UNDER DIFFERENT FEATURE SELECTION ALGORITHMS

Feature Filtering Algorithm	*Prediction Accuracy (%)*
Proposed method	***91.29***
Lasso	***84.46***
Pearson	***83.47***
Spearman	***82.55***
Kernel	***81.12***
Distance_corr	***80.13***

IV. CONCLUSIONS

In this work, we propose a defense strategy based on blockchain and machine learning algorithms to address the network security challenges faced by SDN environments. Our approach integrates multiple techniques, including smart contracts, token bucket algorithm, Isolation Forest algorithm, and a customized composite feature selection algorithm, to enhance the detection and mitigation capabilities against DDoS attacks. We utilize smart contracts to collect relevant data information at intervals across domains, and employ a two-layer detection method for DDoS attack detection and defense within domains. The results demonstrate that our approach can ensure stable transmission of network flows to the greatest extent possible, effectively identify DDoS attacks, and these findings are validated through experimentation. Of course, our strategy also faces limitations such as a small experimental validation dataset and insufficient validation. In the future, we plan to replace the existing test dataset with a larger DDoS attack dataset for more accurate testing of DDoS detection accuracy. Additionally, we aim to continue improving our customized composite feature selection algorithm to further enhance the accuracy of our methods.

ACKNOWLEDGMENT

This work is supported by Natural Science Foundation by Technology Department of Fujian Province, China (No. 2020J01574), Industry-University-Research Innovation Fund for Future Network Technology by Education Department of China (No.2021FNA05003), Industry-Research Project from Network Communication Company (No.KH230139A).

REFERENCES

[1] Luo, S., Wu, J., Li, J., et al.: A defense mechanism for distributed denial of service attack in software-defined networks. In: Ninth International Conference on Frontier of Computer Science and Technology (FCST2015), 2015, pp 325–329. IEEE (2015)

[2] McKeown, N., Anderson, T., Balakrishnan, H., et al.: OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Comput. Commun. Rev. 38(2), 69–74 (2008)

[3] Kaur P, Kumar M, Bhandari A. A review of detection approaches for distributed denial of service attacks. Syst Sci Control Eng. 2017;5(1): 301-320.

[4] Valdovinos I A, Pérez-Díaz J A, Choo K K R, et al. Emerging DDoS attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges and future directions[J]. Journal of Network and Computer Applications, 2021, 187: 103093.

[5] Lian W, Li Z, Guo C, et al. FRChain: A Blockchain-based Flow-Rules-oriented Data Forwarding Security Scheme in SDN[J]. KSII Transactions on Internet & Information Systems, 2021, 15(1).

[6] Li W, Wang Y, Meng W, et al. BlockCSDN: towards blockchain-based collaborative intrusion detection in software defined networking[J]. IEICE TRANSACTIONS on Information and Systems, 2022, 105(2): 272-279.

[7] Zeng Z, Zhang X, Xia Z. Intelligent blockchain-based secure routing for multidomain SDN-enabled IoT networks[J]. Wireless Communications and Mobile Computing, 2022, 2022: 1-10.

[8] Ma R, Wang Q, Bu X, et al. Real-Time Detection of DDoS Attacks Based on Random Forest in SDN[J]. Applied Sciences, 2023, 13(13): 7872.

[9] Su J, Jiang M. A Hybrid Entropy and Blockchain Approach for Network Security Defense in SDN-Based IIoT[J]. Chinese Journal of Electronics, 2023, 32(3): 1-11.

[10] Marvi M, Arfeen A, Uddin R. An augmented K‐means clustering approach for the detection of distributed denial‐of‐service attacks[J]. International Journal of Network Management, 2021, 31(6): e2160.

[11] Sharafaldin I, Lashkari A H, Hakak S, et al. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy[C] 2019 International Carnahan Conference on Security Technology (ICCST). IEEE, 2019: 1-8.

[12] Mininet, Available at: http://mininet.org./, 2024.

[13] Pox, Available at: https://github.com/noxrepo/pox/, 2024.

[14] FISCO BCOS, Available at: https://github.com/FISCO-BCOS/,2024

[15] Webase, Available at: https://gitee.com/WeBank/WeBASE/,2024