DPW-RRM: Random Routing Mutation Defense Method Based on Dynamic Path Weight

Hui Jin[1], Zhaoyang Li[2], Ruiqin Hu[1], Jinglei Tan[1*], and Hongqi Zhang[1]
[1] State Key Laboratory of Mathematical Engineering and Advanced Computing
Zhengzhou, Henan 450001 CN
[e-mail: itplayer123@126.com; zero_hrq@163.com; zhq37922@126.com]
[2] School of Cyberspace Security, Zhengzhou University
Zhengzhou, Henan 450001 CN
[e-mail: hpu_lzy@163.com]
*Corresponding author: Jinglei Tan

Received June 15, 2022; revised June 14, 2023; accepted October 30, 2023;
published November 30, 2023

Abstract

Eavesdropping attacks have seriously threatened network security. Attackers could eavesdrop on target nodes and link to steal confidential data. In the traditional network architecture, the static routing path and the important nodes determined by the nature of network topology provide a great convenience for eavesdropping attacks. To resist monitoring attacks, this paper proposes a random routing mutation defense method based on dynamic path weight (DPW-RRM). It utilizes network centrality indicators to determine important nodes in the network topology and reduces the probability of important nodes in path selection, thereby distributing traffic to multiple communication paths, achieving the purpose of increasing the difficulty and cost of eavesdropping attacks. In addition, it dynamically adjusts the weight of the routing path through network state constraints to avoid link congestion and improve the availability of routing mutation. Experimental data shows that DPW-RRM could not only guarantee the normal algorithmic overhead, communication delay, and CPU load of the network, but also effectively resist eavesdropping attacks.

Keywords: Moving target defense, software defined network, cyber security, network eavesdropping, routing mutation.

A preliminary version of this paper appeared in International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM 2022). This version has extended our study by adding the theoretical analysis of the link congestion detection based on network state constraints, the mutation path calculation and selection based on weighted random routing, and the methods of the mutation strategy delivery.

http://doi.org/10.3837/tiis.2023.11.014 **ISSN : 1976-7277**

1. Introduction

Monitoring attack [1] is a very harmful network attack. The attacker lurks in the network to collect communication data in nodes or links by means of port traffic mirroring and data replication [2], and then uses analysis tools to analyze the communication data. The data is classified and analyzed to obtain important confidential in-formation such as target passwords and communication sessions, which provides intelligent support for the next network attack, and seriously threatens network security. Communication data encryption is one of the technologies to defend against network snooping, but there are some limitations in practical applications, and some encryption protocols have certain flaws, and attackers can crack communication data through vulnerabilities.

At the beginning of the traditional network architecture design, more consideration is given to availability and stability, and the main purpose is to provide stable network services. Although its static network configuration ensures the stability of the system and reduces the complexity of maintenance, it also provides convenience for the implementation of network attacks. The routing paths in the network architecture are relatively static and deterministic, and the node connections in the network topology are always non-uniformly distributed. Some important nodes tend to appear in the network topology structure, and these important nodes participate in most of the connections in the network. From the attacker's point of view, its fixed communication path and presence of important nodes provide significant advantages for the implementation of monitoring attacks. In addition, the monitoring attack is a passive attack with the characteristic of concealment, which is difficult to be detected and prevented by traditional network security devices.

The deterministic and static characteristics of the network architecture [3] make attackers have the advantage in network attack and defense. In order to reverse the imbalance between attack and defense, moving target defense [4-7] is proposed as a dynamic defense concept of "changes the rules of the game". It transfers the system attack surface [8] by dynamically changing the network configuration to improve the attack difficulty and cost of implementation. Routing mutation [9], as one of moving target defense techniques, combines routing paths with the idea of moving target defense to circumvent malicious eavesdropping attacks by dynamically changing the routing paths of the two communicating parties in the network, increasing the difficulty and cost of attack implementation, and improving the active defense capability of the network.

In this paper, using the flexible, programmable, centralized and controllable features of Software Defined Network (SDN) [10-12], a Random Routing Mutation defense method based on Dynamic Path Weights (DPW-RRM) is implemented in SDN. DPW-RRM completes the path calculation and weight initialization according to the depth search algorithm and the centrality index of the network, and constructs a path weight database; By monitoring the network link congestion status in real time, and referring to the congestion control scheme of the TCP protocol [13], the path weight is dynamically adjusted to prevent link congestion caused by random mutation and im-prove the availability of routing mutation; The weighted random algorithm is used to select paths to prevent important nodes from excessively forwarding data and improve the effectiveness of routing mutation; The flow table update strategy of "reverse order addition, priority coverage" is adopted to reduce the impact of routing mutation on network communication.

The rest of the paper is organized as follows. Section 2 reviews related work of routing mutation. In section 3, we propose the system model of MPW-RRM. In section 4, we verify the effectiveness and applicability of the proposed model through simulation experiments and numerical analysis. Section 5 summarizes our conclusions.

2. Related Work

Duan et al. [14] first proposed an active random routing mutation technique (RRM), which randomly changes the routing of communication flows in the network to defend against reconnaissance, snooping, and DoS attacks. RRM uses Satisfaction Modular Theory (SMT) to impose node load, link load and node repeatability constraints on communication paths to solve the communication paths that meet the conditions. On the basis of RRM, LEI et al. [15] further proposed a more complete routing solution constraint scheme based on STM, which adds the switch flow table capacity constraint and forwarding delay constraint, and updates the flow table strategy by adopting the scheme of "reverse order addition and order deletion", finally the monitoring success rate under different network parameters is analyzed theoretically. Jafarian et al. [16] modeled the interactive behavior between routing selection defense and DOS attack, using game theory and STM to constrain network security, performance and Qos to select the best routing, and discussed the defense effect under different network and adversarial parameters.

Zhang et al. [17] proposed a routing mutation mechanism based on reinforcement learning, which solves the routing mutation space through STM, introduces the Q-Learning algorithm to iteratively select the routing path from the routing mutation space, and adaptively adjusts the learning rate through security awareness. Zhao et al. [18] proposed an SDN-based double-hop communication (DHC) method for network monitoring attacks, which realized time-based end information and routing mutation, and proposed a weight-based routing path filtering method to avoid the excessive use of some intermediate routing nodes, so that the attacker can monitor a large number of data packets at a specific node. This method aims to distribute the communication traffic evenly among multiple paths in the network. Reference [19] proposed an SDN-based Hybrid Routing Randomization (HRR) scheme, which implements routing mutation by adding a mapping layer between the physical interface of the routing and the corresponding logical address. Chen et al. [20] proposed an SDN-based intranet dynamic defense system (SIDD) for intranet security issues. The system constrains link capacity and QoS, uses the K shortest path algorithm to find the path, and achieves the effect of resisting reconnaissance scanning attacks combined with IP mutation.

At present, a lot of related research works have been done on routing mutation, but there are still some problems in the existing methods.

(1) Solving routing paths based on satisfiability modular theory (SMT). The SMT solves the path through the conditional constraints and its solution time increases exponentially with the increase of the network size, and each mutation needs to enter the constraints to solve the problem again. In addition, too strict constraints will reduce the space for routing selection and the randomness of routing.

(2) Generating routing paths based on graph theory algorithms. The routing path generated based on the graph theory algorithm can be used continuously without changing the network topology, but there is a lack of relatively complete constraints for routing path solution in related research work.

In view of the above problems, this paper generates routing paths based on graph theory algorithm, and divides path calculation and selection into two steps of path calculation and path selection. In the path calculation step, the calculation of the routing path and the centrality of the network nodes is completed, and the initial weight database of the path is constructed. In the path selection step, network state constraints and non-repeatability constraints are introduced, and path weighted random selection is performed. The solution complexity is reduced by step-by-step calculation, and more complete constraints are introduced.

3. System Model of DPW-RRM

The DPW-RRM system model is shown in **Fig. 1**, where the blue and green lines indicate different communication paths, and the red line indicates that the attacker is conducting a listening attack. The model mainly includes link congestion detection based on network state constraints and mutation path calculation and selection based on weighted random routing. The link congestion detection based on network state constraints mainly includes the collection of link delay information and port flow in-formation, and the detection of congested links is completed by collecting the link de-lay information and port flow information of the network in real time. Hop path calculation and selection based on weighted random routing includes topology discovery, path calculation, path initial weight calculation and path weighted random selection. The topology discovery completes the construction of the topology database in the network initialization phase, and updates the topology database through event information when the network topology changes. When a host in the network initiates communication, the path calculation calculates a mutation path for both parties of the communication according to the topology database information. The initial path weight calculation completes the initialization of the path weight according to the network topology information and the network centrality theory. The path weighted random selection dynamically controls the path weight according to the link congestion information, and selects the path through the weighted random algorithm. Finally, the deployment of the path policy is completed through the delivery of the flow table.

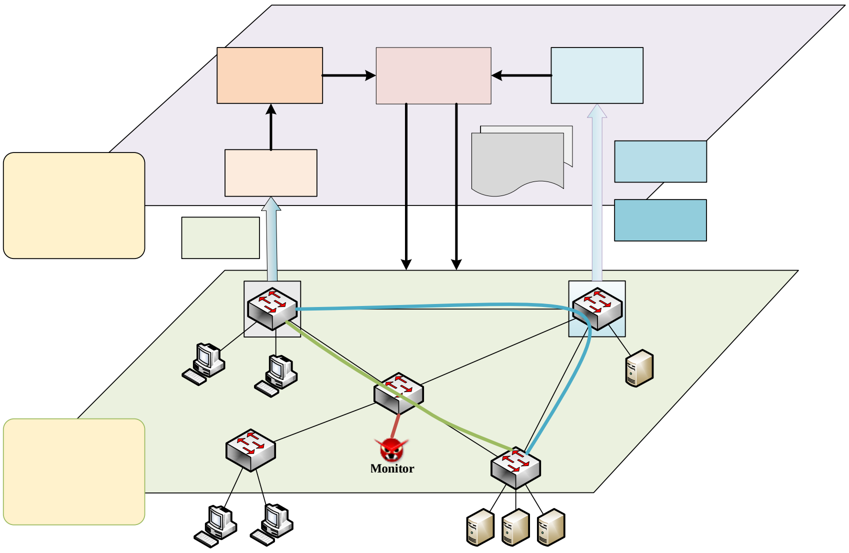


Fig. 1. System Model of DPW-RRM.

3.1 Link Congestion Detection Based on Network State Constraints

The processing capacity of network equipment nodes is limited. When the net-work link is overloaded [21], the network transmission performance will be degraded, resulting in increased link transmission delay and packet loss rate, and even network paralysis, which seriously affects the quality of communication services. Therefore, when the bandwidth of a certain link in the network is heavily occupied, the mutation path that includes this link should be avoided as far as possible to prevent network failures. In order to detect network congestion, this paper uses SDN controller to continuously collect link delay and switch port traffic transmission, and characterizes network congestion through link delay and port used bandwidth. The link delay col-lection and port used bandwidth detection are alternatively implemented by the delay detection module and the port bandwidth collection module, and the network con-gestion detection is completed by the congestion detection module.

3.1.1 Delay Detection Module

According to the different detection methods, there are two main ways of the current network delay detection in SDN: active monitoring and passive monitoring. Active monitoring sends data packets with special marks to the network, and uses the controller to track and analyze the data packets to obtain the link delay, which will introduce additional overhead and burden to the network. However, passive detection uses some existing protocols in the SDN architecture to collect the link delay without affecting the network.

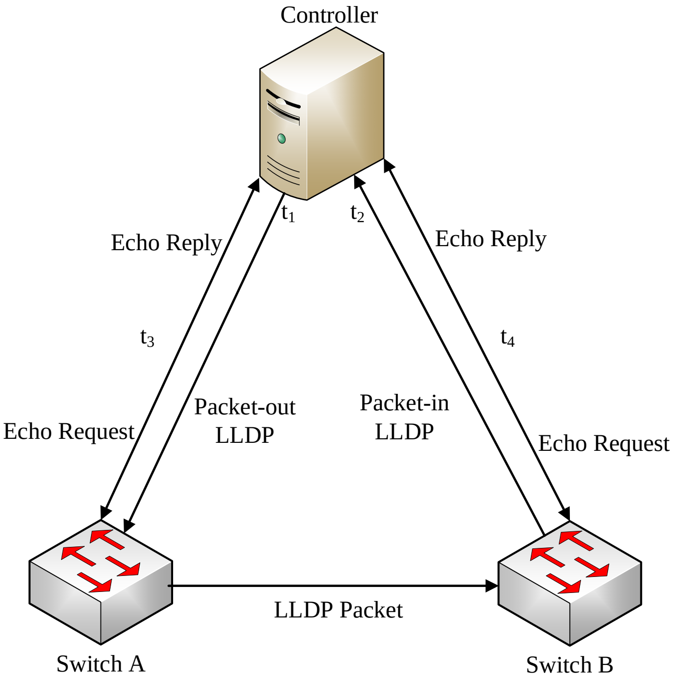


Fig. 2. Working principle of LLDP protocol.

In this paper, the passive detection method is adopted, and the LLDP (Link Layer Discovery Protocol) protocol [21] data packet in SDN is used as the carrier to collect the link delay. The working principle and delay calculation of the LLDP protocol are shown in Fig. 2. The controller encapsulates the LLDP data packets and sends them to switch A in the form of Packet-Out. Switch A receives the LLDP data packets and sends them to all ports in a flooded way. Switch B receives the LLDP data packets and sends them to the controller through the matching flow table. During this process, the time and when the controller sends and receives LLDP data packets are recorded respectively, and the round-trip delays and between the controller and switch A and switch B are obtained by using the echo message. Finally, the link delay between switch A and switch B is obtained by formula (1).

$$T_{AB} = t_2 - t_1 - \frac{(t_3 + t_4)}{2} \tag{1}$$

$$P_d = \frac{T_{AB} \times \left(Count\left(path^i\right) + 1\right) \times 2}{D} \tag{2}$$

According to the network communication delay recommended reference value D and network scale given by ATM, Diffserv, ITU-T and other system standards, the ratio P_d between the current link delay T_{AB} and the reference delay D is obtained, as shown in formula (2) , where $Count\left(path^i\right)$ is the total number of nodes in path $path^i$.

3.1.2 Bandwidth Acquisition Module

The port counter of physical ports is defined in the OpenFlow protocol. It includes statistical information such as packets and bytes received and sent by the port. At the same time, the *PortStatsRequest* event message is provided to obtain the counter statistics of each port of the switch. The controller sends the *PortStatsRequest* message at period *T* to obtain the port counter byte statistics $N(t)$ at time *t*, and calculates the ratio of the byte change at two times to the cycle *T* and bandwidth capacity *V* to obtain the bandwidth occupancy percentage P_t in the *T* period, as shown in the formula (3).

$$P_t = \frac{N(t) - N(t-T)}{T \times V} \tag{3}$$

3.1.3 Congestion Detection Module

The congestion detection module uses the data obtained by the delay detection module and the bandwidth acquisition module to detect the congestion status of the link, which can ensure that the controller can grasp the link load status of the entire network in real time, so as to avoid the chain congestion caused by random routing selection in routing mutation. During the network operation, the delay and bandwidth information of each path is obtained through the delay detection module and the bandwidth acquisition module, and the congestion detection module evaluates the information to judge the congestion status of the link. According to the ratio P_d of the link delay to the reference delay and the port bandwidth occupancy ratio P_t, the link congestion degree P_c is obtained, and the link congestion degree is classified as lower than 50%, between 50% and 80%, and higher than 80% of three cases, which is used to dynamically adjust the path weight when the path is selected. The degree of link congestion is calculated by formula (4).

$$P_c = max(P_t, P_d),\ \forall l_i \tag{4}$$

3.2 Hop Path Calculation and Selection Based on Weighted Random Routing

The research has shown that no matter the size of the network topology, the node connection always presents the characteristics of non-uniform distribution. Some important nodes tend to appear in the network topology, and these nodes participate in most of the connections in the network. Therefore, when these important nodes are attacked, they will have a significant impact on the entire network. Compared with fixed routing paths in static networks, random routing mutation can alleviate the im-portance of these nodes to some extent through path transformation, but the effect is general (shown in Section 4-1). In this paper, we introduce network centrality defined from different angles to determine the important nodes and reduce the frequency of important nodes appearing in path selection, so as to reduce the importance of important nodes in the network and prevent attackers from monitoring important nodes to obtain a large amount of communication data. In addition, this section will intro-duce the link congestion information proposed in Section 3.1 into the path selection, dynamically adjust the weight of the path selection to avoid link congestion that may be caused by path mutation. This section will complete the calculation and selection of hop paths based on weighted random routing through three parts: path calculation, path initial weight calculation and path weighted random selection. The calculation steps are shown in **Fig. 3**.

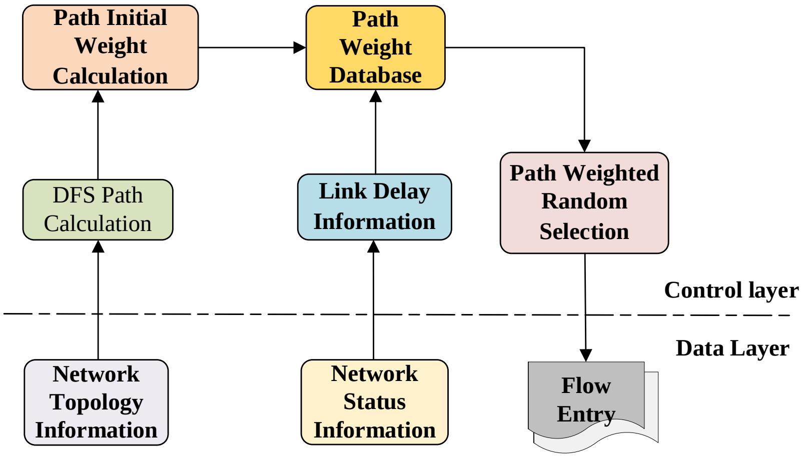

Path Initial Weight Calculation
Path Weight Database
DFS Path Calculation
Link Delay Information
Path Weighted Random Selection
Control layer
Data Layer
Network Topology Information
Network Status Information
Flow Entry

Fig. 3. Mutation path calculation and selection.

3.2.1 Path Calculation

In SDN, the controller obtains the network topology of the entire network through the LLDP protocol. The topology discovery principle of the LLDP protocol is shown in **Fig. 2**. The controller sends LLDP packets with flooding instructions to all SDN switches in the network through Packet-Out. Once the SDN switch receives the LLDP packets from the controller, it will flood the LLDP packets to all ports according to the instructions. After receiving them, the neighboring switch will transmit them through Packet-In to the controller by matching the corresponding flow entry. The controller analyzes and processes the LLDP data packets, and establishes link information between the two SDN switches. When all SDN switches in the network complete the process in the same way, the controller is able to establish network-wide topology information. According to the network topology, while the host in the network initiates communication, the controller traverses the mutation routing nodes through the depth-first search algorithm (DFS), finds all paths between the two communicating parties, and removes the excessively long paths to get the path set $Paths_{s\to t}$.

3.2.2 Path Initial Weight Calculation

The structural nature of the network topology itself leads to the appearance of important nodes, which will appear repeatedly in the path set $Paths_{s\to t}$ of two hosts, making the important nodes forward a large amount of communication data in the process of random path selection.

In order to reduce the frequency of the occurrence of important nodes in the network topology and prevent the attacker from obtaining excessive profits from the important nodes. In this section, we introduce three centrality indicators [22] to characterize the importance of nodes in the network, namely degree centrality, betweenness centrality and proximity centrality. According to the importance of the nodes, the initial weight of the path including the important nodes are reduced, thereby reducing the frequency of the important nodes appearing in the path selection is declined. The three centrality indicators are defined as follows.

(1) Degree centrality

The number of nodes connected to a node is called the degree of the node. The more nodes are directly connected, the higher the degree of the node. Therefore, the degree of a node often indicates the importance of the node in the network. The degree of a node is also the most direct indicator to judge the importance of a node in network analysis. In this paper, the degree centrality of nodes is measured by formula (5), where $deg\left(u\right)$ represents the degree of node *u*, and *V* represents the set of nodes in the network.

$$D_u = \frac{\deg(u)}{V-1} \tag{5}$$

(2) Intermediate centrality

Intermediate centrality indicates the number of times a node acts as the shortest path between any two nodes in the network. The higher the intermediate centrality of a node, the higher the frequency of the node in the shortest path, so the more important it is in the network. The intermediate centrality of a node is measured by formula (6), where $sPaths_{s\to t}(u)$ represents the number of shortest paths from *s* to *t* passing through node *u*, and $sPaths_{s\to t}$ represents the number of shortest paths from *s* to *t*.

$$B_u = \sum_{s\neq t\neq u\in V} \frac{sPaths_{s\to t}(u)}{sPaths_{s\to t}} \tag{6}$$

(3) Proximity centrality

Proximity centrality is used to measure the average distance from one node to other nodes in the network. As shown in formula (7), where d_{uv} represents the geodesic distance between nodes u and v, as the higher the degree of network center of the node, the lower the average geodesic distance. Therefore, the reciprocal processing is performed, and *V* is the node set in the network.

$$C_u = \frac{V-1}{\sum_{v(v\neq u)} d_{uv}} \tag{7}$$

After obtaining the three indicators of degree centrality, intermediate centrality and proximity centrality of the node, we weight them to obtain the centrality weight C_u of the node, as shown in formula (8), where the weights of the three indicators are determined by the network administrator according to the network topology. The weight of the path set *Paths* is initialized using the centrality weight C_u of the node. The weight initialization calculation of the path is shown in formula (9), where $avg\left(Path_{s\to t}^{j}\right)$ represents the mean value of the node centrality weight of the path $Path_{s\to t}^{j}$, which is calculated by dividing the total weight of the node in the path by the path length. The higher the centrality weight C_u of the nodes in the path, the higher the average weight of the node, so the reciprocal processing is performed on it, and finally the initial weight $weight\left(Path_{s\to t}^{j}\right)$ of the path is obtained.

$$\begin{gathered} C_u = \omega_d * D_u + \omega_b * B_u + \omega_c * C_u, \\ \omega_d + \omega_b + \omega_c = 1 \end{gathered} \tag{8}$$

$$weight\left(Path_{s\to t}^{j}\right) = \frac{\sum_{Path_{s\to t}^{i} \in Paths_{s\to t}} avg\left(Path_{s\to t}^{i}\right)}{avg\left(Path_{s\to t}^{j}\right)} \tag{9}$$

3.2.3 Path Weighted Random Selection

After the initial weight of the path is obtained according to the static properties of the network (namely, network topology) in Section 3.2.2, the weighted random path selection algorithm is used to select the path, which can effectively reduce the frequency of important nodes appearing in the path. Therefore, the importance of important nodes is reduced in the actual routing mutation operation. However, random selection of mutation paths based on path weights without constraints may cause link congestion and decrease network performance. Therefore, the algorithm refers to the congestion control of the TCP protocol, and dynamically controls the path weight ac-cording to the link load level returned by the link congestion detection module. The path weight control process is as follows:

(1) In the congestion avoidance stage, the congestion detection module detects that the link load reaches 80%. The weight of the path where the link is located is decreased, thus entering the "slow start" stage, where the path weight is set to 1.

(2) In the "slow start" stage, the congestion detection module detects that the link load is lower than 50%. When the weight of the path where the link is located does not exceed the initial weight obtained by formula (9), the path weight of each hop cycle will increase exponentially, that is, $weight(path_{s\to t}^{j}) \times 2$, otherwise the path weight will be assigned the maximum weight obtained by formula (9).

(3) In the "additive increase" phase of congestion avoidance, the congestion detection module detects that the link load is between 50% and 80%. When the weight of the path where the link is located does not exceed the maximum weight obtained by formula (9), the weight of each hop cycle is linearly increased, that is, $weight(path_{s\to t}^{j}) + 1$, otherwise the path weight is assigned the maximum weight obtained by formula (9).

Algorithm 1: Path selection algorithm

Input:

($Paths_{s\to t}$) : path set
($weights_{s\to t}$) : weight set
($repeat_{s\to t}$) : non-repeatability constraint
($congestionLinks$) : congested link
$RandNum \in [0, \text{sum}(weights_{s\to t})]$: random number

Output:

$Path_{s\to t}$: Mutation path

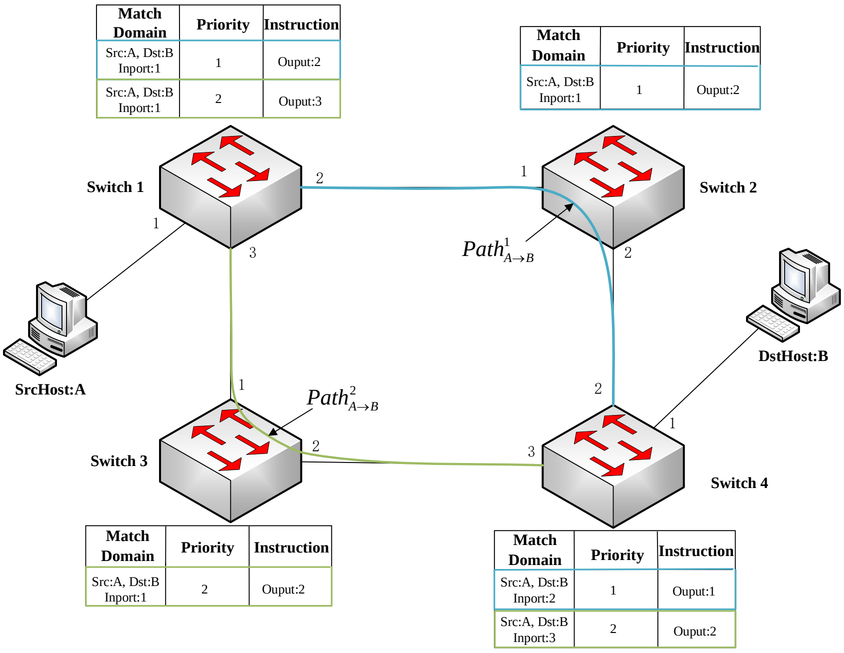
```
pathSelect($availablePaths_{s\to t}$, $weights_{s\to t}$, $RandNum$)
  max = 0
  for $path$ in $availablePaths_{s\to t}$ do
    $max$ += $weights_{s\to t}[path]$
```

```
5.          if $max$>= $RandNum$ do
6.            return $path$
7.    for $path$ in $Paths_{s\to t}$ do
8.      for $clink$ in $congestionLinks$ do
9.        if $clink$ in $path$
10.         update $weights_{s\to t}[path]$ according $status(clink)$
11.     update $weights_{s\to t}[path]$
12.   if $Len(usedPaths_{s\to t})$< $repeat_{s\to t}$ do
13.     $usedPath_{s\to t}$.append($Path_{s\to t}$)
14.   else
15.     $usedPath_{s\to t}$.update($Path_{s\to t}$)
16.   $availablePaths_{s\to t}$ = $Paths_{s\to t}$ - $usedPath_{s\to t}$
17.   $Path_{s\to t}$ = pathSelect($availablePaths_{s\to t}$, $weights_{s\to t}$, $RandNum$)
```

The path selection algorithm is shown in Algorithm 1. Lines 1 to 6 of the algorithm represent the function ***pathSelect***, which selects the mutation path from the set of available paths according to the path weight and random number. Lines 7 to 11 of the algorithm dynamically adjust the path weights according to the network status, where *congestionLinks* is the set of links with a link congestion level higher than 50%, and line 10 processes paths with different weights according to the link congestion level, line 11 Weight processing is performed on paths with a link congestion level lower than 50. Lines 12 to 15 of the algorithm update the list of used paths according to the non-repeatability constraint. Lines 16 to 17 of the algorithm obtain the set of available paths and call the function ***pathSelect*** to obtain the mutation path $Path_{s\to t}$.

3.3 Mutation strategy delivery

To implement the routing mutation strategy, the controller needs to update the flow table entry on the SDN switch at the data layer and execute the routing path mutation. Due to the distributed nature of the data layer switches, the flow table update consistency problem is inevitable when the flow table entries are updated. In order to ensure the consistency of the flow table update and avoid packets error processing, this paper adopts the strategy of "reverse order addition, priority override" to update the flow table. When the flow table entries are updated, the flow table is delivered to the switches in the path in reverse order, that is, the source switch is sent the flow table last. In addition, the new mutation flow table entry will have a higher matching priority. Assuming that during the communication between the source host A and the destination host B, the communication path is changed from $Path_{A\to B}^{1}$ hop to $Path_{A\to B}^{2}$, then the communication data flow is transmitted in the following manner.



Match Domain	Priority	Instruction
Src:A, Dst:B Inport:1	1	Ouput:2
Src:A, Dst:B Inport:1	2	Ouput:3

Match Domain	Priority	Instruction
Src:A, Dst:B Inport:1	1	Ouput:2

Match Domain	Priority	Instruction
Src:A, Dst:B Inport:1	2	Ouput:2

Match Domain	Priority	Instruction
Src:A, Dst:B Inport:2	1	Ouput:1
Src:A, Dst:B Inport:3	2	Ouput:2

Fig. 4. Mutation strategy update.

(1) When the communication path $Path^1_{A\to B}$ is used, the communication data between the host A and the host B is transmitted on the path $Path^1_{A\to B}$, as shown in the blue line in **Fig. 4**.

(2) When installing the flow table for the communication path $Path^2_{A\to B}$, the controller reverses the lower flow table of the switch in the mutation path. When the flow table is not installed to the same common prefix node as $Path^1_{A\to B}$ (switch 1 in **Fig. 4**), The communication data between host A and host B is still transmitted on the path $Path^1_{A\to B}$.

(3) When the flow table of the communication path $Path^2_{A\to B}$ is installed to the same common prefix node as $Path^1_{A\to B}$, the communication data between host A and host B are transmitted on $Path^2_{A\to B}$ by matching the flow table entry of $Path^2_{A\to B}$ because the flow table entry of $Path^2_{A\to B}$ has a higher priority.

(4) The path $Path^1_{A\to B}$ related flow table entry are deleted when they reach the ***hard_timeout*** setting time, relieving the pressure on the flow table capacity of the switch.

When host A stops communicating with host B, the flow entry related to the communication path will be deleted due to the expiration of the ***idle_timeout*** time, and then the FlowRemoved event will be triggered to make the controller detect the interruption of communication between host A and host B, and stop the selection and update of the path.

4. Experimental verification and analysis

Usually, due to limited capabilities and resources, it is difficult for an attacker to launch a monitoring attack on all nodes and links of the target network. In the experiment, it is assumed that the attacker randomly selects some nodes in the network and obtains communication data through port traffic mirroring to resume the communication session. Therefore, this section measures the effectiveness of the mutation algorithm by using the ratio of the data monitored by the attacker in the cracked node to the total transmitted data. In addition, the host communication delay and CPU load of the virtual machine after the introduction of routing mutation are collected to measure the performance and overhead of the proposed routing mutation algorithm.

An SDN switch network is created through Mininet [23] with the OpenFlow1.3 protocol standard as the southbound interface. Ryu is used as the SDN controller to control the traffic transmission of the network. The resources owned by the virtual machines running Mininet and Ryu are: Intel i7-9750h 4-core 2.6GHz, 5G memory. The experimental topology is shown in **Fig. 5**.

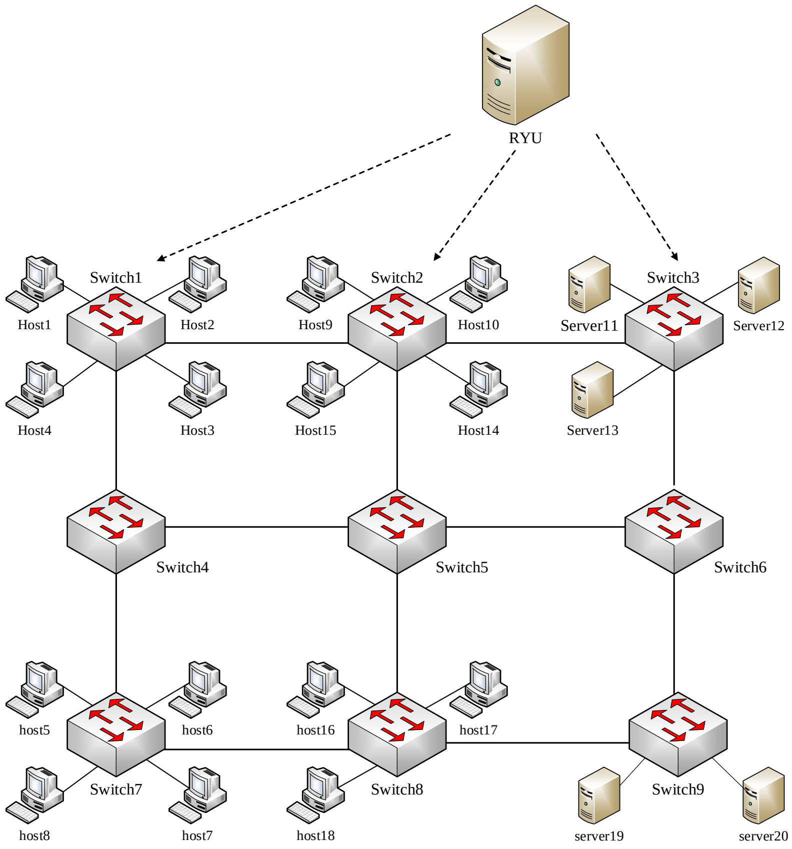


Fig. 5. Experimental topology.

4.1 Validity Verification

The experiment sets up 8 pairs of hosts in the network (which includes host1 and server20) to communicate with each other at the speed of 100Mb/s for 10 minutes, with a routing mutation period of 10s, and the maximum link load capacity set to 600Mb/s. In this communication state, a partial link congestion degree $P_c \geq 80\%$ occurs in some cycles, which triggers the path weighted random selection for the congestion management (described in Section 3.2.3). Monitor the communication data between host1 and server20 on all switch nodes (except source and destination nodes). The experiment is divided into four communication schemes:

static network, random routing, DPW-RRM without link congestion, and DPW-RRM under link congestion. The experimental results are shown in **Fig. 6**, where the horizontal coordinate is the switch node number, and the vertical coordinate represents the percentage of the number of packets monitored by the switch nodes to the total.

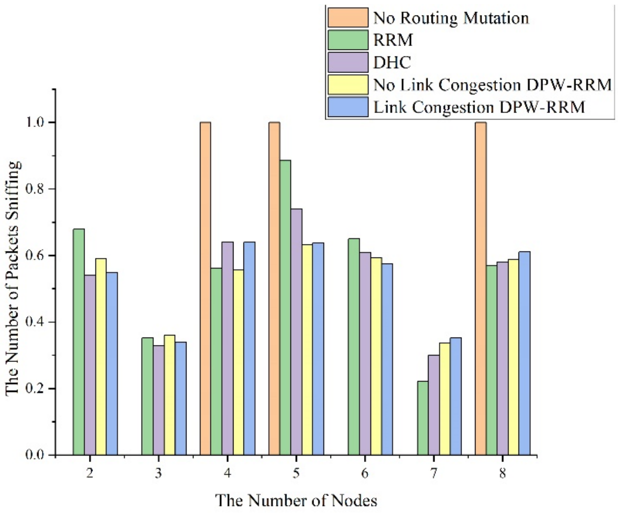


Fig. 6. The number of packets the attacker listens to on the node.

As can be seen from the figure, since the routing path in the static network will not change once it is generated, the shortest path between host1 and server20 is obtained according to the shortest path algorithm as (1-4-5-8-9). Therefore, the attacker can monitor all communication data between host1 and server20 on any of the switches (4,5,8). In contrast, the random routing network will distribute the communication data on all switch nodes, however, the randomly generated forwarding path will cause some important nodes (such as node 5) to appear repeatedly in multiple paths, so that the attacker in important nodes can obtain most of the communication data and may cause link congestion of important nodes. Several experimental results show that the random routing strategy will make the important node 5 forward more than 80% of the communication data. The DPW-RRM mutation scheme proposed in this paper can effectively solve this problem. As shown in the **Fig. 6**, after introducing the path weighted random selection algorithm, the forwarding communication data of important node 5 is reduced to 63%. Moreover, the DPW-RRM can dynamically adjust the path weight before network link congestion to avoid the degradation of the network transmission performance caused by the overload of the network link.

4.2 Performance Testing

Compared with the traditional network, the routing mutation network improves the active defense capability of the network, but it will bring more performance consumption. This section will discuss the performance influence brought by the introduction of DPW-RRM from the aspects of algorithmic overhead, transmission delay, and CPU load.

4.2.1 Algorithmic Overhead

This section compares the efficiency of the STM algorithm and the DPW-RRM algorithm based on the graph theory algorithm in solving the path. To ensure the reliability of the comparison, a variety of complex topologies of Topology zoo [24] are selected for experimental comparison, and the relevant information of the topology is shown in **Table 1**. The data layer network is generated by the topology simulation through Mininet, and comparative experiments are carried out on different topologies to obtain the solution time of the STM algorithm and the DPW-RRM algorithm.

Table 1. Different experimental topologies

Topology name	Abilene	Ans	Agis	BICS	SANET	DFN	Ans
Node	11	18	25	33	43	58	18
Link	14	25	30	48	45	87	25

The experimental results are shown in **Fig. 7**, where the abscissa represents different topology names, and the ordinate represents the time it takes to solve the path of the topology. It can be seen that as the number of different topology nodes increases, the solution time of the STM algorithm increases exponentially, while the solution time of the DPW-RRM algorithm shows an overall increasing trend, but the growth trend is significantly smaller than that of the STM algorithm. The reason for the increase in the number of topology BICS to SANET nodes but the decrease in the solution time is that DPW-RRM relies on the deep search algorithm to calculate the path, while topological BICS, although the number of nodes is less than that of SANET, has a higher complexity of the topology, and therefore the solution time is longer. In addition, the method of solving path using the STM constraint has to input all the constraints to solve the paths again for each mutation. However, DPW-RRM splits the path calculation and selection into two steps to execute path calculation and path selection. The communicating parties only need to calculate the paths of both parties at the first communication, and only the path selection algorithm is executed during the mutation process thereafter, which compresses the time complexity to a constant level and is only related to the number of paths between the communicating parties.

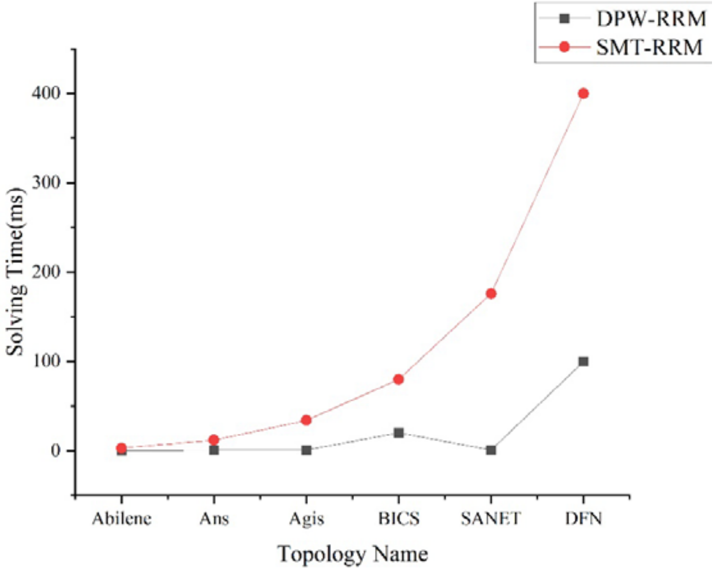


Fig. 7. Solution time of mutation space in different topologies.

4.2.2 Transmission Delay

The routing and forwarding paths in traditional networks are usually the shortest paths, and some longer routing paths are selected for mutation in DPW-RRM networks to ensure the randomness of the routings, thus leading to the increase of network delay. In this section, the network delay is collected and analyzed under different routing mutation cycles, and the experimental results are shown in **Fig. 8**. The horizontal coordinate is the transmission rate of data packets sent from host1 to server20, and the vertical coordinate is the delay.

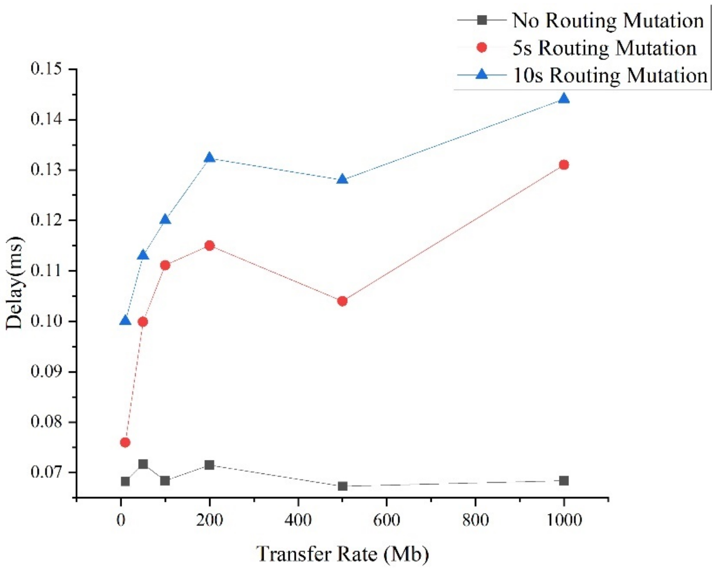


Fig. 8. Comparison of transmission delays with different mutation periods.

As can be seen from the figure, he DPW-RRM network has a higher communication delay compared to the traditional network, and the communication delay is further improved as the reduction of the routing mutation period. In the traditional network, the transmission delay is between 0.67ms and 0.71ms, and with the increase of transmission rate, the delay of DPW-RRM network with 10s mutation period is increased between 0.08ms and 0.63ms, and the delay of DPW-RRM network with 5s mutation period is increased between 0.32ms and 0.76ms.

The delay improvement of the DPW-RRM network is mainly in two aspects: first, one is the improvement of routing path length, which will cause the communication data packets to pass through more links and switch nodes in the network, thus bringing about a higher time delay. Second, the delay caused by the installation of the flow table, according to the flow table update strategy of "reverse order addition", when the flow table is updated to the same prefix routing node, such as the routing path changes from (1-4-5-8- 9) to (1-2-3-6-9), when node 1 updates the flow table, it will have a certain impact on the data flow in transmission, thus resulting in an increase in delay. Therefore, the transmission delay of the DPW-RRM network is higher than that of the traditional network, and the delay will increase with the increase of the routing mutation frequency.

4.2.3 CPU Load

The routing and forwarding paths in traditional networks are basically unchanged after they are generated. In contrast, the DPW-RRM network needs to dynamically change the routing path for both communication parties according to the routing mutation cycle. Therefore, more routing selection, flow table generation and flow table installation operations are required, which results in an increase in the CPU load of the controller. The main influencing factor is the routing mutation period. The experimental results are shown in **Fig. 9**, where 10 pairs of hosts in the network communicate with each other to observe the CPU load under different mutation cycles. The horizontal coordinate is the running time, and the vertical coordinate is the CPU load. As can be seen from the figure, the CPU load of the traditional network is between 11.3% and 13.5%, the CPU load of routing mutation with a mutation cycle of 10s is between 13.5% and 16.3%, and the CPU load of routing mutation with a mutation cycle of 5 s is between 17.4% and 19.1%, and the CPU load increases as the mutation period decreases. The reason is that as the mutation period decreases, the controller performs more frequent routing selection, flow table generation, and flow table installation operations, resulting in the increase of CPU load.

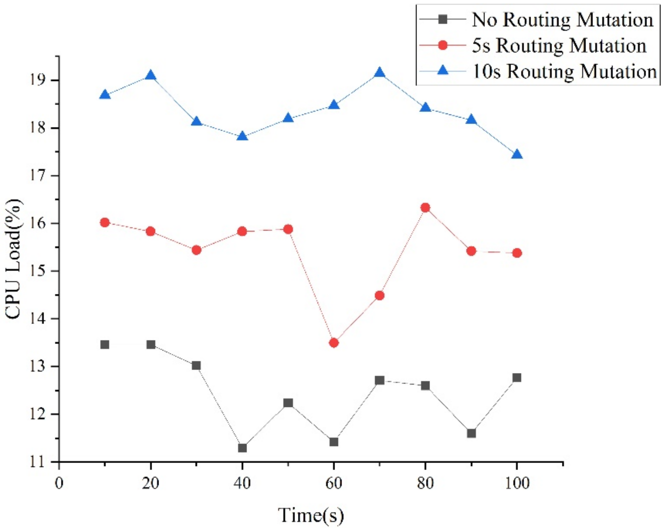


Fig. 9. Comparison of CPU load with different jump cycles.

In addition, in order to ensure the communication stability of the routing mutation network during the mutation period, when the mutation flow table is updated, the new and old flow entries will reside in the switch for a period of time at the same time, resulting in increased consumption of flow table space.

5. Conclusion

Aiming at monitoring attacks, this paper proposes a random routing mutation defense method based on dynamic path weights—DPW-RRM, which balances the path weights by introducing the network centrality metric to portray the importance of nodes; dynamically adjusts the weighted weights of routing paths through network state constraints to avoid network link congestion caused by routing mutation; dynamically changes routing paths through weighted random routing mutation, increasing the difficulty and cost of attackers to perform monitoring attacks. The experimental results show that DPW-RRM can ensure the normal overhead and basic performance of the network system, and effectively disperse the communication traffic among multiple paths to resist the monitoring attack.

In the current work, there are still some problems in the calculation and selection of paths in terms of flexibility. Future work will introduce a reinforcement learning framework to select mutation paths at node granularity and optimize path mutation strategies to further improve the active defense capability of the network.

References

[1] Jan M A, Nanda P, He X, et al., "A Sybil attack detection scheme for a forest wildfire monitoring application," *Future Generation Computer Systems*, vol. 80, no. 3, pp. 613–626, Mar. 2018. Article (CrossRef Link)

[2] Duohe MA, Qiong LI, Dongdai LIN, "Moving target defense against network eavesdropping attack using POF," *Journal on Communications*, vol. 39, pp. 73–87, Feb. 2018. Article (CrossRef Link)

[3] Guilin C, Baosheng W, Tianzuo W, et al., "Research and Development of Moving Target Defense Technology," *Journal of Computer Research & Development*, vol. 53, no. 5, pp. 968–987, May. 2016. Article (CrossRef Link)

[4] Dilli P. Sharma, Simon Yusuf Enoch, Jin-Hee Cho, Terrence J. Moore, Frederica F. Nelson, Hyuk Lim, Dong Seong Kim, "Dynamic Security Metrics for Software-Defined Network-based Moving Target Defense," *Journal of Network and Computer Applications*, vol. 170, Nov. 2020. Article (CrossRef Link)

[5] Jing-lei T, Heng-wei Z, Hong-qi Z, et al., "Optimal temporospatial strategy selection approach to moving target defense: A FlipIt differential game model," *Computers & Security*, vol. 108, pp. 102342, Sep. 2021. Article (CrossRef Link)

[6] Cho J H, Sharma D P, Alavizadeh H, et al., "Toward proactive, adaptive defense: A survey on moving target defense," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 709–745, Jan. 2020. Article (CrossRef Link)

[7] Jing-lei T, Hui J, Heng-wei Z, et al., "A survey: When moving target defense meets game theory," *Computer Science Review*, vol. 48, 100544, 2023. Article (CrossRef Link)

[8] Jing-lei T, Hui J, Rui-qin H, et al., "WF-MTD: Evolutionary Decision Method for Moving Target Defense Based on Wright-Fisher Process," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 6, pp. 4719-4732, 2023. Article (CrossRef Link)

[9] Z. Zhou, C. Xu, X. Kuang, T. Zhang and L. Sun, "An efficient and agile spatio-temporal route mutation moving target defense mechanism," in *Proc. of 2019 IEEE International Conference on Communications* (*ICC*), Shanghai, China, pp. 1–6, May. 2019. Article (CrossRef Link)

[10] ZHANG L, LIN H, HUAN W, et al., "Software defined network flow rule conflict detection system based on OpenFlow," *Journal of Computer Applications*, vol. 42, no. 2, pp. 528–533, Feb. 2022. Article (CrossRef Link)

[11] Yang X, Xu H, Liu J, et al., "Achieving high reliability and throughput in software defined networks," *Computer Networks*, vol. 197, pp. 108271, Oct. 2021. Article (CrossRef Link)

[12] Ramprasath J, Seethalakshmi V, "Mitigation of malicious flooding in software defined networks using dynamic access control list," *Wireless Personal Communications*, vol. 121, no. 1, pp. 107–125, Jun. 2021. Article (CrossRef Link)

[13] Shu-hong Z, "Research and Implementation of TCP Congestion Control Mechanism Based on SDN in Data Center Net-work," *Chinese Journal of Computers*, vol. 40, no. 9, pp. 2167–2180, Sep. 2017.

[14] D Duan Q, Al-Shaer E, Jafarian H, "Efficient random route mutation considering flow and network constraints," in *Proc. of 2013 IEEE Conference on Communications and Network Security* (*CNS*), National Harbor, MD, pp. 260–268, Oct. 2013. Article (CrossRef Link)

[15] Cheng L E I, "Network moving target defense technique based on optimal forwarding path migration," *Journal on Communications*, vol. 38, no. 3, pp. 133–143, Mar. 2017. Article (CrossRef Link)

[16] Jafarian J H, Al-Shaer E, Duan Q, “Formal approach for route agility against persistent attackers,” in *Proc. of European Symposium on Research in Computer Security*, pp. 237–254, 2013. Article (CrossRef Link)

[17] Zhang T, Kuang X, Zhou Z, et al., “An intelligent route mutation mechanism against mixed attack based on security awareness,” in *Proc. of 2019 IEEE Global Communications Conference* (*GLOBECOM*), Waikoloa, HI, USA, pp. 1-6, Dec. 2019. Article (CrossRef Link)

[18] ZHAO Z, GONG D, LU B, et al., “SDN-based double hopping communication against sniffer attack,” *Mathematical Problems in Engineering*, vol. 2016, Jan. 2016. Article (CrossRef Link)

[19] S. Wang, Y. Zhou, R. Guo, J. Du and J. Du, “A Novel Routing Randomization Approach for Moving Target Defense,” in *Proc. of 2018 IEEE 18th International Conference on Communication Technology* (*ICCT*), Chongqing, China, pp. 11-15, Oct. 2018. Article (CrossRef Link)

[20] CHEN Y, HU H, CHENG G, “The Design and Implementation of a Software-Defined Intranet Dynamic Defense System,” *Acta Electronica Sinica*, vol. 46, no. 11, pp. 2604–2611, Nov. 2018. Article (CrossRef Link)

[21] Yang Li, Zhi-Ping Cai, Hong Xu, “LLMP: Exploiting LLDP for Latency Measurement in Software-Defined Data Center Net-works,” *Journal of Computer Science and Technology*, vol. 33, pp. 277–285, Mar. 2018. Article (CrossRef Link)

[22] Zhixiong W S Z, “Review on researches of network centrality algorithm,” *Library and Information Service*, vol. 54, no. 18, pp. 107-110+148, Sep. 2010.

[23] Haeeder Munther Noman, Noman Haeeder Munther, Jasim Mahdi Nsaif, “POX Controller and Open Flow Performance Evaluation in Software Defined Networks (SDN) Using Mininet Emulator,” in *Proc. of IOP Conference Series: Materials Science and Engineering*, vol. 881, Apr. 2020. Article (CrossRef Link)

[24] Knight S, Nguyen H X, Falkner N, et al., “The internet topology zoo,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765-1775, Sep. 2011. Article (CrossRef Link)

Hui Jin was born in Ningxia, P.R. China, in 1988. He received his M.S. degree from the Zhengzhou Information Science and Technology in 2020. His research interests include security game theory and reinforcement learning.

Zhaoyang Li was born in Henan, P.R. China, in 1995. He received the MA.Eng. degree from Zhengzhou University in 2022. His research interests include attack-defense modeling and proactive defense.

Ruiqin Hu was born in Hubei, P.R. China, in 1995. He received the MA.Eng. degree from the Zhengzhou Information Science and Technology Institute in 2022. His main research interests include network security and moving target defense.

Jinglei Tan was born in Shandong, P.R. China, in 1994. He received his Ph.D. degree in the Zhengzhou Information Science and Technology in 2022. He has been a lecturer with State Key Laboratory of Mathematical Engineering and Advanced Computing since 2022. His research interests include moving target defense, deception defense and intelligent game theory.

Hongqi Zhang born in Hebei, P.R. China, in 1962. He received the Ph.D. degree in Zhengzhou Information Science and Technology in 1998. His main research interests include network security and classication protection, etc. Since 2002, he has been a Professor and a Ph.D. Supervisor with the China National Digital Switching System Engineering & Technological Research Center. He is a member of the advisory committee of cyber security teching in higher education, ministry of eduction. P.R. China. He is also the Editor of the *Chinese Journal of Network and Information Security*. Mr. Zhang received prizes, such as the second prize of the National teaching prize in 2009, the National Network Security Outstanding Teacher Award in 2017 and the first prize of the National Science and Technology Progress Award in 2018. He is a senior member of China Computer Federation.