Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.048922

ARTICLE

Securing Forwarding Layers from Eavesdropping Attacks Using Proactive Approaches

Jiajun Yan, Ying Zhou*, Anchen Dai and Tao Wang

School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen, 518107, China

*Corresponding Author: Ying Zhou. Email: zhouying5@mail.sysu.edu.cn

Received: 21 December 2023 Accepted: 14 February 2024 Published: 25 April 2024

ABSTRACT

As an emerging network paradigm, the software-defined network (SDN) finds extensive application in areas such as smart grids, the Internet of Things (IoT), and edge computing. The forwarding layer in software-defined networks is susceptible to eavesdropping attacks. Route hopping is a moving target defense (MTD) technology that is frequently employed to resist eavesdropping attacks. In the traditional route hopping technology, both request and reply packets use the same hopping path. If an eavesdropping attacker monitors the nodes along this path, the risk of 100% data leakage becomes substantial. In this paper, we present an effective route hopping approach, called two-day different path (TDP), that turns communication paths into untraceable moving targets. This technology minimizes the probability of data leakage by transmitting request data and reply data through different paths. Firstly, a brief introduction to the network model and attack model involved in this paper is given. Secondly, the algorithm and processing method of the TDP are proposed. Thirdly, the paper proposes three different metrics to measure the effectiveness of the proposed approach. Finally, theoretical analysis and simulation results show that the TDP can effectively reduce the percentage of data exposure, decrease eavesdropping attack success probability, and improve the unpredictability of the path.

KEYWORDS

Route hopping; moving target defense; software-defined network; two-way different path; metrics

1 Introduction

The software-defined network is a new networking paradigm, aimed at augmenting flexibility and manageability by segregating the control plane from the data plane in conventional switches and routers. SDN offers several advantages, including plane separation, centralized control, and network programmability. Currently, SDN shows promising development prospects in areas such as network function virtualization, network security defense, and 5G network technologies. Integrating network function virtualization with SDN significantly improves network management and fosters the efficient use of resources. Furthermore, SDN's centralized control framework provides a distinct benefit in terms of enhancing network transparency and facilitating the enforcement of security policies, thereby supporting the formulation of more sophisticated security strategies. During the deployment of 5G, SDN provides the capability for flexible control and efficient management of network resources,

essential for satisfying the elevated data rates and reduced latency demands of 5G technology. SDN has found extensive applications across a variety of areas, including smart grids, campus networks, large data center networks, cloud computing, IoT, and edge computing. However, software-defined networks have become susceptible to security threats in recent years, including scanning, Denial of Service, and eavesdropping attacks.

Moving target defense embraces a proactive approach to network security, shifting away from the pursuit of an impenetrable, flawless network. Instead, it focuses on creating a dynamic, heterogeneous, and unpredictable network environment. This approach aims to enhance system randomness or diminish system predictability, thereby complicating potential attacks. Currently, many researchers have implemented the concept of moving target defense in the security defense of software-defined networks [1]. Steinberger et al. [2] introduced MTD solutions to defend against threats faced by high-speed software-defined networks. Luo et al. [3] proposed a hybrid strategy combining MTD and honeypots to address threats in software-defined networks. Narantuya et al. [4] utilized multiple controllers to enhance the multiplexing capability of the MTD strategy. Various defense strategies are devised for different attack methods in SDN. Jafarian et al. [5] proposed an address mutation approach called OpenFlow Random Host Mutation (OF-RHM) to transparently mutate IP addresses with high unpredictability and high rate. The authors report that OF-RHM can invalidate 99% of the collected information. Furthermore, Jafarian et al. [6] proposed a multipath routing approach, called random route mutation which considers flow, network, security constraints, attacker's capabilities, and attacker's strategies. They state that this approach can decrease the percentage of disrupted packets to below 10%, as compared with single-path routing schemes. Additionally, Ma et al. [7] introduced a moving target defense strategy to thwart eavesdropping attacks, the full protocol stack randomization and message packaging randomization are realized by protocol-oblivious forwarding. The authors claim that this innovative approach significantly lowers the likelihood of message interception by attackers and complicates the process of message reassembly.

Route hopping is an important technology within the domain of moving target defense, enhancing the unpredictability of communication paths by dynamically altering the communication paths and routing rules among network nodes. Eavesdropping attacks, characterized by their covert nature, stand as a primary concern within security defenses. Route hopping plays an essential role in countering such attacks, making it more difficult for adversaries to intercept complete communication packets. In this paper, route hopping is used to protect the forwarding layer of SDN from eavesdropping attacks. Firstly, a two-day different path approach based on the software-defined network is proposed. In the TDP approach, request packets and reply packets use different paths for transmission, thereby enhancing the unpredictability of the communication path. Secondly, the algorithm and processing procedure of the TDP are proposed. Thirdly, to measure the effectiveness of TDP, we introduce metrics including the percentage of data exposure, eavesdropping attack success probability, and route hopping entropy. Finally, the effectiveness of theoretical analysis and the TDP in defending against eavesdropping attacks is verified through experiments.

The main contributions of our paper are summarized as follows:

1) The two-way different path approach is proposed to increase the difficulty for attackers to intercept the complete request and reply packets.

2) The route-hopping process is optimized to ensure packet integrity during transmission.

3) Three metrics are proposed to measure the advantages of the TDP in terms of data leakage, resistance to eavesdropping attacks, and path unpredictability.

4) The effectiveness of the proposed approach and the correctness of the theoretical analysis were verified in the experimental networks.

The rest of this paper is organized as follows. Related works are discussed in Section 2. Section 3 briefly introduces the network model of SDN and the attack model. Section 4 describes the algorithm and processing procedure of the TDP. Section 5 presents network topology and evaluation metrics. The experimental results are discussed in Section 6. Finally, Section 7 concludes the paper.

2 Related Work

The combination of route hopping and software-defined networks has attracted the interest of many researchers. Currently, many researchers are studying the hopping mechanism of route hopping. Authors in [8] first proposed the technology of adding a dynamic mapping layer between logical routing and physical routing to achieve a larger space range of route randomization, higher route randomization frequency, and smaller route randomization costs. In [9], a weighted random routing hopping scheme based on network state constraints was proposed. This scheme adjusts weighted values based on the network state to randomly select routing paths, enhancing the unpredictability of path selection. In [10], a routing mutation trigger mechanism based on network traffic anomaly detection was proposed. Furthermore, to enhance the unpredictability of the path during the randomization process, an optimal routing path selection algorithm based on the improved ant colony algorithm was proposed. In [11], an SDN-based multipath routing application is designed to increase the difficulty for an eavesdropper attempting to intercept the communication data flow between Supervisory Control and Data Acquisition devices. Authors in [12] identified a vulnerability in existing multipath methods that could result in 100% data leakage and propose a two-way multipath approach to mitigate the issue of complete data leakage in current multipath methods. Besides, most recent studies model route hopping as a constraint satisfaction problem. They satisfy route hopping requirements by considering performance constraints, time constraints, and space constraints. Duan et al. in [13] modeled capacity, overlap, and quality of service as constraint satisfaction problems. They proposed a random route mutation technology capable of simultaneously altering multiple flows, applicable in both traditional networks and SDN networks. The authors in [14] proposed corresponding constraints from the three dimensions of forwarding path capacity, delay, and reachability, and proposed an optimal routing path generation method based on the security capacity matrix. Zhang et al. [15] utilized the Jaccard distance matrix and temporal constraints to enlarge the mutation space. This mutation space can be dynamically changed to enhance unpredictability. Concurrently, they proposed strategies involving route weights and pre-distribution of flow entries, aimed at balancing network traffic and reducing time overhead respectively. The authors in [16] proposed an MTD technology based on adaptive forwarding path migration. Traditional path mutation approaches often overlook the problems of performance constraints of the forwarding path and inappropriate combination of mutation paths and mutation period. This paper addresses these issues by applying satisfiability modulo theory and a mutation path generation algorithm, which is grounded in the network security capacity matrix.

Additionally, integrating route hopping with other moving target defense technologies is currently an important research area. In [17], a double-hopping communication method was proposed, in which both the routing path of the communication and the end information of the packet are changed dynamically. In [18], a path-hopping communication method based on SDN was proposed. By assigning data to different paths for transmission and dynamically changing end information, the overhead of eavesdropping attacks and the difficulty for attackers to recover data is increased.

In summary, route hopping is an effective approach to defend against eavesdropping attacks. However, there are still two problems with the above method. One is that after a path hopping, the request and reply packets are transmitted through the same path, which is easy to intercept and analyze by the attacker [9–13]. If an attacker successfully monitors the transmission path of the current time interval, it could result in complete data leakage; The second is the lack of a suitable approach for evaluating the effectiveness and unpredictability [14–18]. To solve these problems, we propose an approach of two-way different paths, aiming at increasing the difficulty for attackers to intercept the complete request and reply packets. Meanwhile, we propose three evaluation metrics to measure the effectiveness and unpredictability of route hopping. These metrics include the exposure rate, the eavesdropping attack success rate, and the route hopping entropy.

3 System Model

3.1 Network Model of SDN

As shown in Fig. 1, the architecture of Software-Defined Network comprises six components: Application layer, northbound interface, control layer, southbound interface, forwarding layer, and data layer.

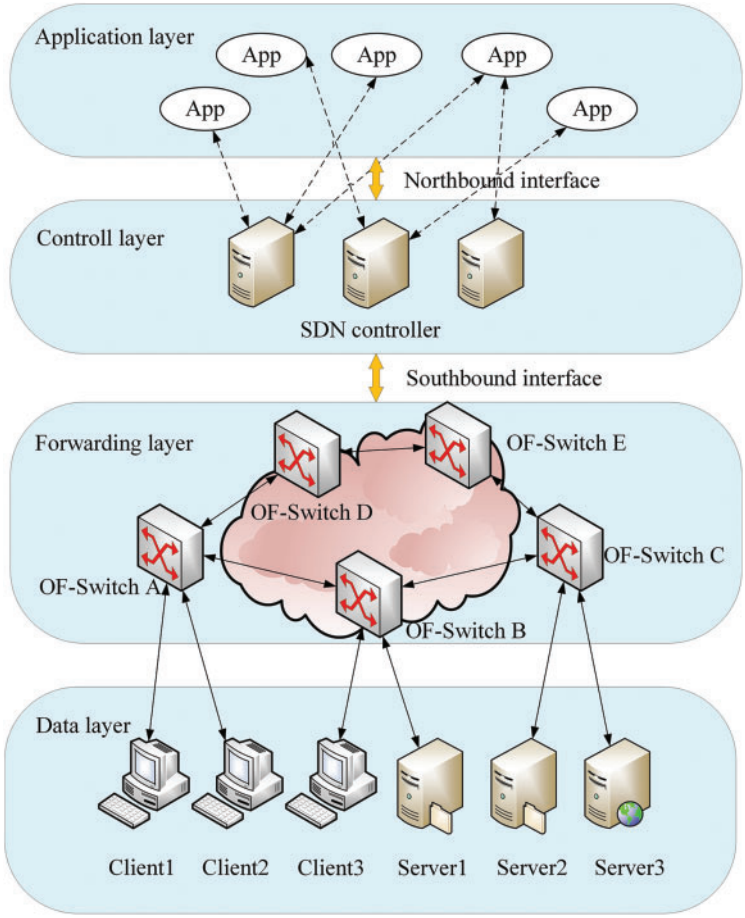

Figure 1: The architecture of SDN

Application layer: This layer can realize unified management of network resources and enables the development of applications based on application programming interface (API). It controls the low-level devices through the API provided by the control layer and develops various personalized applications based on the northbound interface.

Northbound interface: This interface realizes the interactive communication between the application plane and the control plane through the protocol. Northbound interface protocols are primarily responsible for offering an abstract network view, enabling applications to directly control the behavior of the network and conveniently access underlying network resources.

Control layer: This layer centrally manages all devices in the network. The control layer is the core of SDN, composed of various controllers. It communicates with the forwarding layer via southbound interfaces and with the application layer through northbound interfaces.

Southbound interface: This interface realizes the information transmission between the forwarding layer and the controller through the protocol. The southbound interface protocol primarily implements two functions. Firstly, it provides the collected switch information to the control layer, delivers control strategies to the forwarding layer, and guides the forwarding actions of the forwarding layer. Secondly, it plays a key role in network configuration and management.

Forwarding layer: This layer is responsible for flow table processing, data forwarding, and status collection. The forwarding layer makes forwarding decisions based on flow table entries provided by the control layer. It focuses on data processing based on flow entries. Unlike traditional switches, SDN switches do not handle control logic tasks like link discovery, address learning, and route calculation.

Data layer: This layer is composed of various terminal devices, including various clients, servers, etc.

3.2 Attack Model

The smart grid based on Software-Defined Networking is a power system that integrates SDN technology to achieve a more flexible, intelligent, and manageable electrical network. In the pursuit of flexibility and efficiency, smart grids face a series of security threats, including control plane attacks, spoofing and deception, and eavesdropping attacks. Within eavesdropping attacks, traffic monitoring and control plane monitoring are crucial methods of attack. We consider the following attack behaviors in this work.

Eavesdropping attack: Eavesdropping attacks refer to a type of attack in which an attacker attempts to obtain sensitive information without authorization. Such attacks typically involve monitoring, interception, and interpretation of communication channels. Attackers may eavesdrop through physical methods, network interception, or malware.

As shown in Fig. 2, within the smart grid framework, an attacker can identify all possible request and reply paths for source and destination addresses and eavesdrop on some communication nodes. In addition, attackers can analyze and reorganize data through intercepted data.

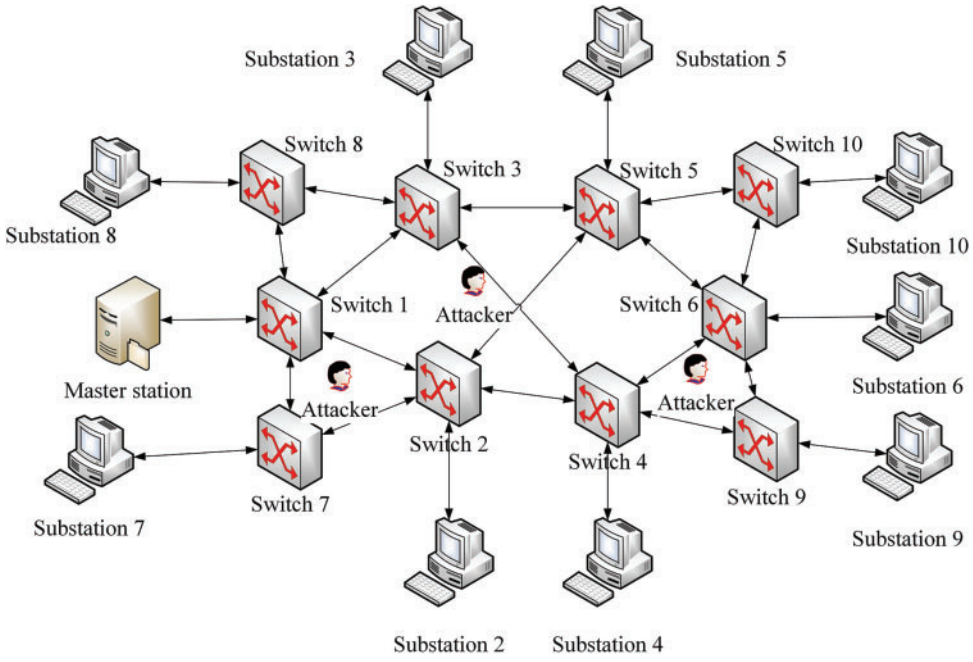

Figure 2: Devices in forwarding layer and data layer of SDN-based smart grid

4 Two-Way Different Path Approach

In smart grids that deploy traditional route hopping technology, whether it is communication between substations or communication between users and the substation, both the request packet and the reply packet transmit in the same hopping path, potentially resulting in complete data leakage. To mitigate the risk of data leakage, this paper proposes a two-way different path approach based on software-defined networks. During the hopping period, request data and reply data use different transmission paths, reducing the relevance of information intercepted by an attacker at a specific node or link.

4.1 Two-Way Different Path Algorithm in SDN Controller

The proposed two-way different path algorithm is presented in Algorithm 1, which is designed to be deployed in the POX controller.

As shown in Algorithm 1, the controller processes all legitimate data packets from the switch by extracting the source (src) and destination (dst) IP addresses, and then checks whether the transmission path of the source and destination IP addresses already exists. If it already exists, the original path is used to transmit the packet. Otherwise, the controller calculates all communication paths from the source to the destination IP address (*Multipaths(src,dst)*). In the routing calculation module, the Floyd-Warshall-based algorithm is used to calculate 'n' shortest paths. A subset of *Multipaths(src,dst)* is selected as a pool of the request (*Request Multipaths(src,dst)*) and reply paths (*Reply Multipaths(src,dst)*) based on specific constraints. After completing the routing path calculation, a path is randomly selected as the request path (*Requestpath(src,dst)*) and the corresponding flow table is delivered to the switch according to the selected request path. Then, a path is randomly selected as the reply path (*Replypath(dst,src)*) and the flow table is delivered to the switch based on the selected reply

path. Finally, the switch completes the transmission process of the data packet according to the flow table.

When the hopping interval T is reached, the request path, the reply path and the flow table are updated. The controller sends the flow table to the corresponding OpenFlow Switch (OF-Switch) according to the new request and reply paths. This OF-Switch then executes the data packet transmission process according to the new flow table.

Algorithm 1: POX controller algorithm

```
for all legitimate packets p from OF-Switches do
    src=packet_in.src
    dst=packet_in.dst
    if Multipaths(src,dst) exist then
        Requestpath(src,dst)=Random(Request Multipaths(src,dst))
        Replypath(dst,src)=Random(Reply Multipaths(dst,src))
    end if
    if Multipaths(src,dst) not exist then
        Multipaths(src,dst)=calculateallpaths(src,dst)
        Request Multipaths(src,dst)=subset(Multipaths(src,dst))
        Reply Multipaths(dst,src)=subset(Multipaths(src,dst))
        Requestpath(src,dst)=Random(Request Multipaths(src,dst))
        Flowmod(Requestpath)
        Replypath(dst,src)=Random(Reply Multipaths(dst,src))
        Flowmod(Replypath)
    end if
    if reach the hopping intervals T then
        update Requestpath(src,dst)
        update Flowmod(Requestpath)
        update Replypath(dst,src)
        update Flowmod(Replypath)
    end if
end for
```

4.2 The Process of the Two-Way Different Path

As shown in Fig. 3, consider a scenario where a message is transmitted between Substation 7 and Substation 10. In the traditional route hopping process, during a hopping interval, the controller randomly selects < Switch 7, Switch 2, Switch 5, Switch 10 > as the request routing path. Subsequently, < Switch 10, Switch 5, Switch 2, Switch 7 > is selected as the reply routing path.

In the approach of this paper, after completing the path calculation of the source and destination IP addresses, the controller randomly selects < Switch 7, Switch 2, Switch 5, Switch 10 > as the request routing path in the path set and then delivers the flow table to the switches in the selected path. Different from the traditional route hopping approach, the controller may not use < Switch 10, Switch 5, Switch 2, Switch 7 > as its reply routing path, but will randomly select < Switch 10, Switch 5, Switch 3, Switch 1, Switch 7 > as its reply routing path, then deliver the flow table to the corresponding switch. This approach further increases the dispersion of messages, thereby increasing the complexity of eavesdropping attacks.

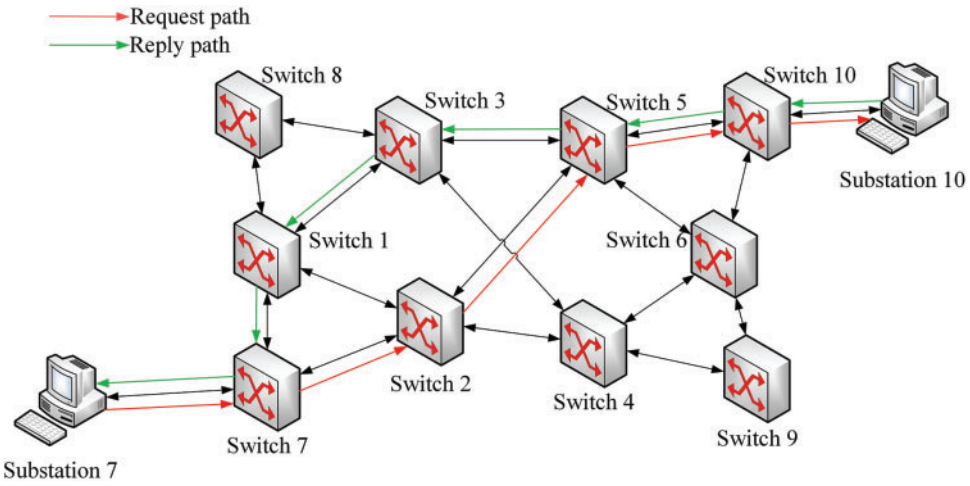

Figure 3: The process of the two-way different path

During the process of route hopping, the traditional mechanism for updating flow tables may result in the loss of data packets. To this end, we have designed a flow table update mechanism characterized by "sequential addition and delayed deletion". "Sequential addition" means that the hopping controller installs flow table information on the nodes along the hopping path in the sequential direction from the source node to the destination node. "Delayed deletion" means that the hopping controller will wait for one complete communication cycle before deleting the old flow table rules. Take the request process of Fig. 3 as an example, when substation 7 initiates communication with substation 10. Assuming that the communication path of the current hopping period is < Switch 7, Switch 2, Switch 5, Switch 10 >, and the routing path of the next hopping period is < Switch 7, Switch 2, Switch 4, Switch 6, Switch 10 >, the specific steps for the establishment and update of the flow table are as follows:

(1) Initially, the controller calculates all communication paths between substation 7 and substation 10. Subsequently, < Switch 7, Switch 2, Switch 5, Switch 10 > is randomly selected as the communication path and the flow table is delivered sequentially.

(2) Assume that the route hopping period is T seconds. After T seconds, the controller randomly selects < Switch 7, Switch 2, Switch 4, Switch 6, Switch 10 > as the request path from the calculated routing path, and then updates the old flow table in Switch 7 through the modification command. At last, the controller sends a new flow table to Switch 2, Switch 4, Switch 6, and Switch 10.

(3) After waiting for the maximum delay in communication between Substation 7 and Substation 10, the controller will deliver a flow entries deletion command to delete the old flow entries in Switch 2, Switch 5, and Switch 10.

Under the flow table update mechanism of "sequential addition, delayed deletion", assuming that the old routing path is denoted as RP_{old}, the new routing path is denoted as RP_{new}, and a hopping route is denoted as RP_i, the following situations exist:

(1) $RP_i \notin RP_{new} \cup RP_i \notin RP_{old}$: It indicates that RP_i does not belong to the old routing path, nor does it belong to the new routing path. Consequently, since data does not enter RP_i, it will not be transmitted via RP_i.

(2) $RP_i \notin RP_{new} \cup RP_i \in RP_{old}$: It indicates that RP_i belongs to the old routing path and does not belong to the new routing path. As a result, communication data will continue to be transmitted along the old routing path.

(3) $RP_i \in RP_{new} \cup RP_i \in RP_{old}$: It indicates that RP_i belongs to both the old routing path and the new routing path. Consequently, it implies that the routing path of the next hopping period is the same as the routing path of the current period. Thus, communication data will be forwarded according to the corresponding flow table entry.

(4) $RP_i \in RP_{new} \cup RP_i \notin RP_{old}$: It indicates that RP_i belongs to the new routing path and does not belong to the old routing path. Consequently, New communication data will be transmitted on the new routing path. In addition, due to the flow table update mechanism of "sequential addition, delayed deletion", existing communication data continues to be transmitted in the old routing path until the transmission completion.

According to the above discussion, the flow table update process of "sequential addition, delayed deletion" not only guarantees the continuity of data transmission but also ensures the integrity of data packets during the transmission process. This establishes it as a dependable mechanism for flow table updates.

5 Deployment and Simulation Experiment

5.1 Simulation Experiment

To validate the feasibility and effectiveness of the TDP, we utilized mininet and POX controller [19] to construct an experimental network topology (N_1), as shown in Fig. 4.

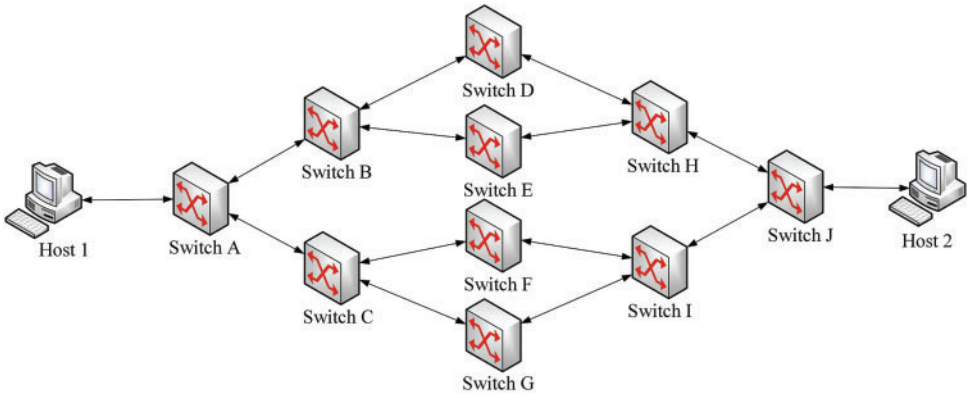

Figure 4: The network topology used in the simulation

5.2 Metrics

5.2.1 The Percentage of Data Exposure

Eavesdropping attackers will eavesdrop on the transmission link. The percentage of data exposure refers to the probability that the link eavesdropper intercepts a complete request and reply data. Assuming that m represent the total number of data packets transmitted from a sender S to a receiver R in the data plane. Assume that there are n selected paths for data transmission from S to R, with S_1 and S_2 being adjacent switches in the forwarding plane. This analysis is concerned with packet exposure occurring when both request and reply packets are intercepted on the same link. The estimated percentage of data exposure E of a link $< S_1, S_2 >$ is defined as:

$$E\left(< S_1, S_2 >\right) = \frac{\frac{2m}{b} \times \varepsilon}{2m} \tag{1}$$

where b denotes the number of combinations of request and reply paths. When the request path and the reply path are the same hopping path, $b = n$; when the request path and the reply path use different hopping path, $b = n^* n$; ε represents the total number of paths where both the request and reply path use this link. The percentage of data exposure is a metric for assessing the risk of data being compromised on this link. The larger $E\left(< S_1, S_2 >\right)$ is, the easier it is for the data to be exposed. Moreover, this formula also helps us better understand where the TDP algorithm is optimized.

5.2.2 Eavesdropping Attack Success Probability

Eavesdropping attack success probability measures the probability that the attacker intercepts certain data from the specified source host to the destination host and then successfully reorganizes it to reconstruct the original data. In graph theory, a 'cut' is an operation that divides the nodes of a graph into two disjoint parts. The 'minimum cut' refers to finding a way to divide the graph so that the cost of cutting is minimized. In a software-defined network, we can abstract the devices in the forwarding layer into a graph. Consequently, the eavesdropping attack success probability can be analyzed using a theory similar to the minimum cut.

Assume that the network topology $G = < V, E >$ is an undirected connection graph, where V is a set of forwarding nodes and E is a set of links. V contains a forwarding nodes, where the attacker can randomly eavesdrop on e nodes ($e \leq a$), and the set of eavesdropping is represented as V_e. The eavesdropping set consists of the eavesdropping node set V_e^n, $V_e = \left\{V_e^1, \ldots, V_e^n\right\}$, where $V_e^n = \left\{h_1^n, \ldots, h_e^n\right\}$. Assume that the cut set of the network topology is U_i, and the cut set consists of a series of cut node sets $u_k, U_i = \{u_1, \ldots, u_k\}$, where $u_k = \{node_1, \ldots, node_k\}$, V_e^b is the effective eavesdropping set, $V_e^b = \left\{v_e^1, \ldots, v_e^b\right\}$, where $v_e^b = \left\{h_e^1, \ldots, h_e^b\right\}$, $v_e^b \supseteq u_k$, $V_e^b \subseteq V_e$.

In traditional route hopping, the eavesdropping attack success probability can be calculated by

$$P_t = \frac{\sum_{b=1}^{e} \left|V_e^b\right|}{C_a^e} \tag{2}$$

In the approach of this paper, we assume that the set of request node is C_i, and the set of reply node is S_i, when $C_i \cap V_e^b \neq \varnothing$, $S_i \cap V_e^b \neq \varnothing$, the minimum cut set of eavesdropping nodes is satisfied, a network attacker is capable of intercepting the entirety of the transmitted data. Since $0 \leq P_t \leq 1$, the range of eavesdropping attack success probability P_d is

$$(P_t)^2 \leq P_d \leq P_t \tag{3}$$

5.2.3 Entropy

1) Information entropy

Information entropy is a fundamental concept in information theory that serves as a measure of information quantity, describing the uncertainty of a random variable. The greater the information entropy, the higher the uncertainty of the information. Conversely, lower information entropy indicates reduced uncertainty.

In probability theory, for a discrete random variable X, its probability distribution is denoted as $P(X)$. The information entropy $H(X)$ can be calculated by the following formula:

$$H(X) = -\sum_{x_i \in X} p(x_i) \log(p(x_i)) \tag{4}$$

where x_i denotes the possible values of X, and $p(x_i)$ denotes the probability of the corresponding value of x_i.

2) Route hopping entropy

Referring to the concept of information entropy in information theory, route hopping entropy is proposed to measure the uncertainty of the hopping path. During route hopping, assuming that H is the entropy of route hopping, the entropy of route hopping can be defined as follows:

$$H(R) = -\sum_{R_i \in R} p(R_i) \log(p(R_i)) \tag{5}$$

where R represents the set of paths that can be selected, R_i denotes the *ith* selected path, and $p(R_i)$ is the probability of the corresponding value of R_i.

Route hopping entropy serves as a metric to evaluate the effectiveness of route-hopping strategies. A higher route hopping entropy signifies increased path uncertainty, thereby rendering the prediction of the next hop path more challenging for an attacker. Consequently, enhancing route-hopping entropy can help improve attack costs and defense effects.

6 Experimental Results and Analysis

6.1 Effectiveness and Scalability

6.1.1 Effectiveness

In our experimental scenario, host 1 runs the ping command to communicate with host 2. This procedure continues for a total duration of 10 min, with the route hopping interval configured at 30 s and the number of selectable paths denoted as n, fixed at 4. To compare the effectiveness of different approaches, both the approach in literature [13] and the TDP are implemented for the transmission of data packets. Considering that the majority of route hopping adopts the approach in literature [13], the approach RRM in literature [13] can be regarded as a traditional route hopping approach.

As illustrated in Table 1, we conducted a comparative analysis using data from four hopping periods for different hopping approaches. In the approach of literature [13], both request and reply data are transmitted through the same routing path during each hopping interval. In contrast, our route hopping approach generally uses different routing paths to transmit request and reply data within the same hopping period.

Table 1: Comparison of routing paths of different approaches

Route hopping period	Message	Host1 ping Host2	
		Route path	
		The approach in [13]	Our approach
T_1	Request message	Host1→SwitchA→SwitchB→SwitchD→SwitchH→SwitchJ→Host2	Host1→SwitchA→SwitchB→SwitchD→SwitchH→SwitchJ→Host2
	Reply message	Host2→SwitchJ→SwitchH→SwitchD→SwitchB→SwitchA→Host1	Host2→SwitchJ→SwitchH→SwitchE→SwitchB→SwitchA→Host1
T_2	Request message	Host1→SwitchA→SwitchC→SwitchF→SwitchI→SwitchJ→Host2	Host1→SwitchA→SwitchC→SwitchF→SwitchI→SwitchJ→Host2
	Reply message	Host2→SwitchJ→SwitchI→SwitchF→SwitchC→SwitchA→Host1	Host2→SwitchJ→SwitchI→SwitchG→SwitchC→SwitchA→Host1
T_3	Request message	Host1→SwitchA→SwitchB→SwitchE→SwitchH→SwitchJ→Host2	Host1→SwitchA→SwitchB→SwitchE→SwitchH→SwitchJ→Host2
	Reply message	Host2→SwitchJ→SwitchH→SwitchE→SwitchB→SwitchA→Host1	Host2→SwitchJ→SwitchI→SwitchF→SwitchC→SwitchA→Host1
T_4	Request message	Host1→SwitchA→SwitchC→SwitchG→SwitchI→SwitchJ→Host2	Host1→SwitchA→SwitchC→SwitchG→SwitchI→SwitchJ→Host2
	Reply message	Host2→SwitchJ→SwitchI→SwitchG→SwitchC→SwitchA→Host1	Host2→SwitchJ→SwitchI→SwitchG→SwitchC→SwitchA→Host1

6.1.2 Scalability

In this section, an experimental network (N_2) was established, comprising one hundred switches labeled S1 through S100. Each switch is connected to two hosts, denoted as h1s1, h2s1,..., h1s100, h2s100. Then, we deploy TDP to N_2. In our experimental scenario, h1s1 runs the ping command to communicate with h1s9, repeating the experimental steps in 6.1.1. The experimental results are shown in Table 2.

It can be concluded from the experimental results that in large networks, the request path and reply path of data are also generally different within the same hopping period. This shows that the TDP is suitable for large networks and the TDP is scalable.

Table 2: Comparison of routing paths of different network sizes

Route hopping period	Message	Our approach (TDP)	
		Route path	
		N_1 (Host1 ping Host2)	N_2 (h1s1 ping h1s9)
T_1	Request message	Host1→SwitchA→SwitchB→SwitchD→SwitchH→SwitchJ→Host2	h1s1→S1→S3→S4→S9→h1s9
	Reply message	Host2→SwitchJ→SwitchH→SwitchE→SwitchB→SwitchA→Host1	h1s9→S9→S4→S45→S1→h1s1
T_2	Request message	Host1→SwitchA→SwitchC→SwitchF→SwitchI→SwitchJ→Host2	h1s1→S1→S45→S4→S9→h1s9
	Reply message	Host2→SwitchJ→SwitchI→SwitchG→SwitchC→SwitchA→Host1	h1s9→S9→S2→S3→S1→h1s1
T_3	Request message	Host1→SwitchA→SwitchB→SwitchE→SwitchH→SwitchJ→Host2	h1s1→S1→S3→S2→S9→h1s9
	Reply message	Host2→SwitchJ→SwitchI→SwitchF→SwitchC→SwitchA→Host1	h1s9→S9→S2→S13→S1→h1s1
T_4	Request message	Host1→SwitchA→SwitchC→SwitchG→SwitchI→SwitchJ→Host2	h1s1→S1→S3→S4→S9→h1s9
	Reply message	Host2→SwitchJ→SwitchI→SwitchG→SwitchC→SwitchA→Host1	h1s9→S9→S4→S3→S1→h1s1

6.2 Result and Analysis for the Percentage of Data Exposure

In our experimental scenario, host1 runs the ping command to communicate with host2. This procedure continues for a total duration of ten minutes, with the route hopping interval configured at five seconds. The number of selectable paths denoted as n, is configured to be either 3 or 4. To evaluate and compare the percentage of data exposure of different approaches, both the RRM and the TDP are employed for data packet transmission. The total number of data packets passing through each transmission link is counted and the calculation results are shown in Fig. 5.

As shown in Fig. 5, for both $n = 3$ and $n = 4$, the percentage of data exposure caused by using the TDP is significantly lower. This is because the request data packet and the reply data packet are transmitted through different paths most of the time in the TDP. Under TDP, the number of request and reply path combinations are n^*n, compared to n in the traditional route hopping approach. According to formula (1), the percentage of data exposure of the TDP approach is lower than the traditional route hopping approach. In addition, the more paths n can be selected, the lower the percentage of data exposure is. What's more, when $n = 3$, the percentage of data exposure for links AB and HJ are higher than for other links. Similarly, when $n = 4$, the percentage of data exposure for links AB, HJ, AC, and IJ is higher than for other links, owing to their more frequent reuse in the hopping routes.

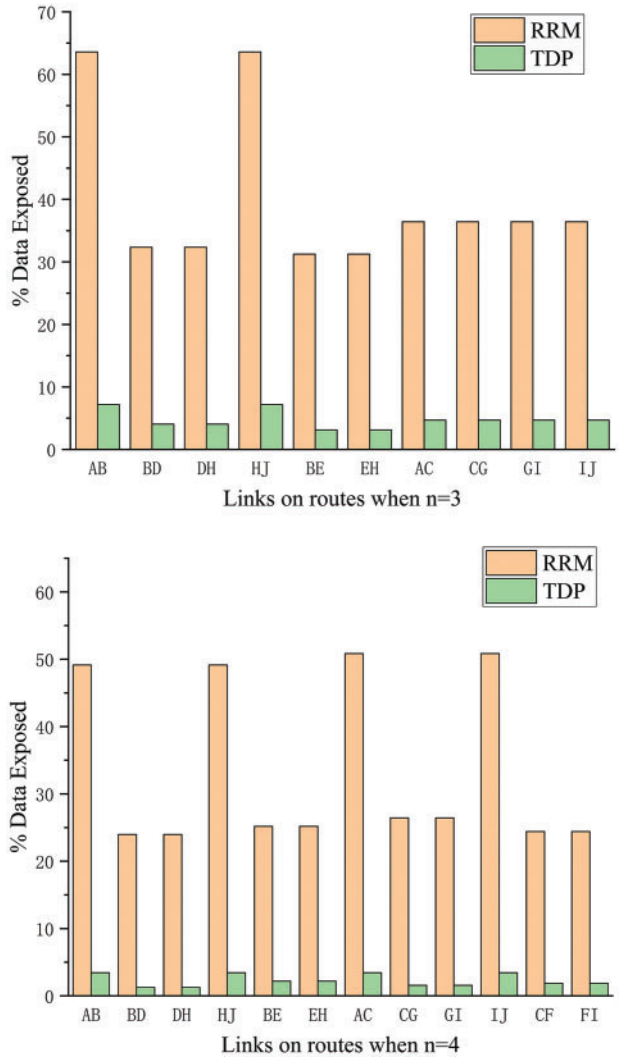

Figure 5: Comparing data exposure using different approaches

6.3 Result and Analysis for Eavesdropping Attack Success Probability

In our experimental scenario, refer to the experimental steps in Section 6.2, the main difference lies in the configuration of the number of selectable paths, which are set to 2, 3, and 4. For comparative experiments, we assume that the attacker can eavesdrop from 1 to 10 nodes, respectively. In our approach, we try to avoid duplication of request paths and reply paths. According to the discussion in Section 5.2.2, we counted the probabilities of obtaining complete communication data with varying numbers of eavesdropping nodes for both approaches under different scenarios. The results of these calculations are illustrated in Fig. 6.

As shown in Fig. 6, it is observed that for both the approach proposed in this paper and the traditional route hopping approach, the eavesdropping attack success probability gradually increases to 100% with an increasing number of eavesdropping nodes. When the quantity of eavesdropping nodes remains constant, the eavesdropping attack success probability in the TDP is found to be lower

than that in the traditional routing hopping approach. This remains the case until the point where an attacker is required to eavesdrop on the complete network topology. At this point, the eavesdropping attack success probability for both approaches becomes equivalent, as observed in scenarios like when $n = 4$. Theoretically, in the TDP, an increase in the number of available paths should lead to a reduction in the eavesdropping attack success probability. However, during this experiment, an increase in the number of selectable paths did not result in a significant change in this probability. This issue can be attributed to the constraints of the experimental network topology, which offers only four selectable paths. Consequently, attackers are required to eavesdrop on the set of nodes that satisfy the minimum cut.

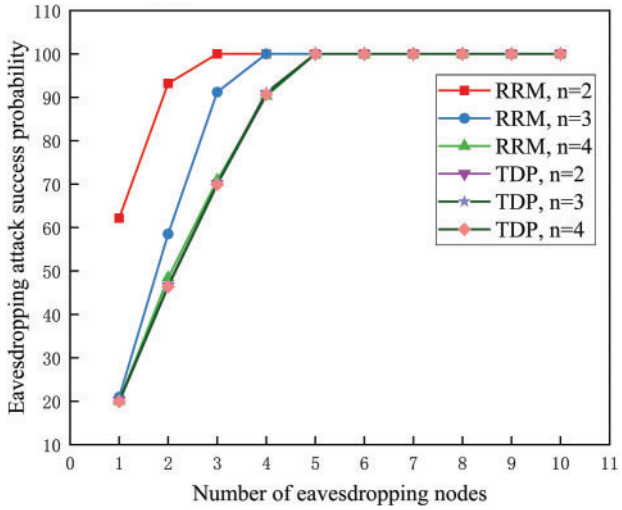

Figure 6: Eavesdropping attack success probability with different numbers of eavesdropping nodes

6.4 Result and Analysis for Unpredictability

In our experimental scenario, host 1 functions as the client, while host 2 operates as the server. The client executes the wget command every 10 s to obtain web page information from the server. The duration of the experiment was established to be 10 min. For comparative analysis, the TDP and the approach in [12] were employed. We quantified the probability associated with each combination of request and reply paths, and subsequently calculated the route hopping entropy utilizing formula (5).

As shown in Fig. 7, for n values of 2, 3, and 4, the route hopping entropy of the two approaches is basically consistent with the theoretical value calculated by formula (5). The route hopping entropy of the TDP is higher than that using the traditional route hopping approach, which indicates that TDP has greater unpredictability and enhanced defensive capabilities. This advantage is attributed to TDP's capability to provide a greater number of request and reply path combinations. Moreover, as n increases, the entropy values for both approaches increase, indicating that a larger pool of optional paths helps to increase the unpredictability of the hopping paths.

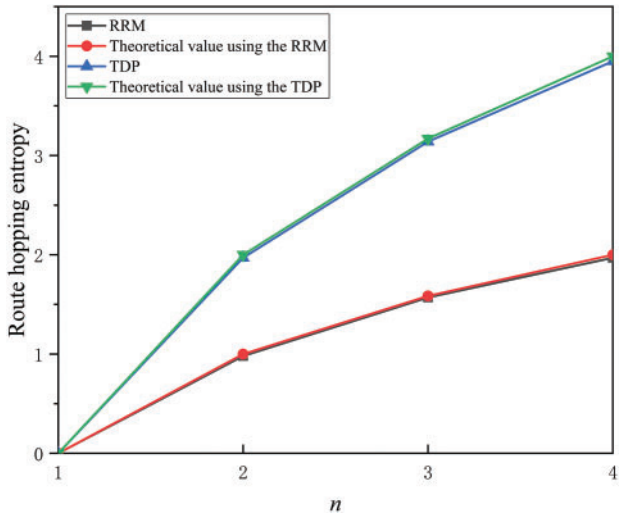

Figure 7: Route hopping entropy for different number of path selections

7 Conclusion

In this paper, a two-way different path approach is proposed. The TDP improves path unpredictability by selecting different paths to transmit request packets and reply packets. To evaluate the TDP's effectiveness in reducing data leakage, resisting eavesdropping attacks, and increasing path unpredictability, three metrics are proposed, namely the percentage of data exposure, eavesdropping attack success probability, and route-hopping entropy. The final theoretical analysis and experimental results show that the TDP is better than the traditional route-hopping approach in reducing the percentage of data exposure, decreasing the eavesdropping attack success probability, and improving the route hopping entropy. Although the TDP approach adopts a random hopping mechanism to better increase unpredictability, it will cause some additional overhead. In the future, we will study adaptive route-hopping methods.

Acknowledgement: We would like to thank the anonymous reviewers for their helpful and constructive comments.

Funding Statement: This research was partially funded by the Natural Science Foundation of Guangdong Province under Grant Number 2021A1515011910, and by the Shenzhen Science and Technology Program under Grant No. KQTD20190929172704911.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization: J.Y. and Y.Z.; methodology: J.Y. and Y.Z.; software: J.Y.; validation: J.Y., Y.Z., A.D. and T.W.; writing—original draft preparation: J.Y.; writing—review and editing: J.Y., Y.Z., A.D. and T.W.; supervision: Y.Z. and T.W.; project administration: J.Y., Y.Z., A.D. and T.W.; funding acquisition: Y.Z. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, "A survey of moving target defenses for network security," *IEEE Commun. Surv. Tutor.*, vol. 22, no. 3, pp. 1909–1941, 2020. doi: 10.1109/COMST.2020.2982955.

[2] J. Steinberger *et al.*, "DDoS defense using MTD and SDN," in *NOMS 2018–2018 IEEE/IFIP Netw. Oper. Manag. Symp.*, Taipei, Taiwan, IEEE, 2018, pp. 1–9.

[3] X. Luo, Q. Yan, M. Wang, and W. Huang, "Using MTD and SDN-based honeypots to defend DDOS attacks in IOT," in *2019 Comput. Commun. IoT Appl. (ComComAp)*, Shenzhen, China, IEEE, 2019, pp. 392–395.

[4] J. Narantuya *et al.*, "SDN-based IP shuffling moving target defense with multiple SDN controllers," in *2019 49th Annu. IEEE/IFIP Int. Conf. Depend. Sys. Netw.-Supplement. Vol. (DSN-S)*, Portland, OR, USA, IEEE, 2019, pp. 15–16.

[5] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Openflow random host mutation: Transparent moving target defense using software defined networking," in *Proc. 1st Workshop Hot Top. Softw. Def. Netw.*, Helsinki, Finland, 2012, pp. 127–132.

[6] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Formal approach for route agility against persistent attackers," in *Comput. Secur.–ESORICS 2013:18th Eur. Symp. Res. Comput. Secur.*, Egham, UK, Sept. 9–13, 2013, Springer, pp. 237–254.

[7] D. Ma, L. Wang, C. Lei, Z. Xu, H. Zhang and M. Li, "Thwart eavesdropping attacks on network communication based on moving target defense," in *2016 IEEE 35th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Las Vegas, NV, USA, IEEE, 2016, pp. 1–2.

[8] S. Wang, Y. Zhou, R. Guo, J. Du, and J. Du, "A novel route randomization approach for moving target defense," in *2018 IEEE 18th Int. Conf. Commun. Technol. (ICCT)*, Chongqing, China, IEEE, 2018, pp. 11–15.

[9] Z. Li, J. Tan, R. Hu, H. Zhang, F. Chen and X. Xu, "A weighted random routing hopping scheme based on network state constraints," in *2022 Int. Conf. Mach. Learn. Cloud Comput. Intell. Min. (MLCCIM)*, Xiamen, China, IEEE, 2022, pp. 171–174.

[10] J. Liu, H. Zhang, and Z. Guo, "A defense mechanism of random routing mutation in SDN," *IEICE Trans. Inf. Syst.*, vol. 100, no. 5, pp. 1046–1054, 2017. doi: 10.1587/transinf.2016EDP7377.

[11] E. G. da Silva, L. A. D. Knob, J. A.Wickboldt, L. P. Gaspary, L. Z. Granville and A. Schaeffer-Filho, "Capitalizing on SDN-based SCADA systems: An anti-eavesdropping case-study," in *2015 IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, Ottawa, ON, Canada, IEEE, 2015, pp. 165–173.

[12] A. Aseeri, N. Netjinda, and R. Hewett, "Alleviating eavesdropping attacks in software-defined networking data plane," in *Proc. 12th Annu. Conf. Cyber Inf. Secur. Res.*, Tennessee, Oak Ridge, USA, 2017, pp. 1–8.

[13] Q. Duan, E. Al-Shaer, and H. Jafarian, "Efficient random route mutation considering flow and network constraints," in *2013 IEEE Conf. Commun. Netw. Secur. (CNS)*, National Harbor, MD, USA, IEEE, 2013, pp. 260–268.

[14] C. Lei, D. Ma, H. Zhang, Q. Han, and Y. Yang, "Network moving target defense technique based on optimal forwarding path migration," *J. Commun.*, vol. 38, no. 3, pp. 133–143, 2017. doi: 10.11959/j.issn.1000-436x.2017056.

[15] B. Zhang, L. Han, and S. Sun, "Dynamic random route mutation mechanism for moving target defense in SDN," in *2021 6th Int. Symp. Comput.. Inform. Process. Technol. (ISCIPT)*, Changsha, China, IEEE, 2021, pp. 536–541.

[16] Y. Hu *et al.*, "Moving target defense based on adaptive forwarding path migration for securing the SCADA network," *Secur. Commun. Netw.*, vol. 2021, pp. 1–15, 2021. doi: 10.1155/2021/1704125.

[17] Z. Zhao, D. Gong, B. Lu, F. Liu, and C. Zhang, "SDN-based double hopping communication against sniffer attack," *Math. Probl. Eng.*, vol. 2016, no. 2, pp. 1–13, 2016. doi: 10.1155/2016/8927169.

[18] C. Zhang, Y. Bu, and Z. Zhao, "SDN-based path hopping communication against eavesdropping attack," in *Opt. Commun. Opt. Fiber Sens. Opt. Mem. Big Data Storage*, Beijing, China, SPIE, vol. 10158, 2016, pp. 135–142.

[19] R. Patel, P. Patel, P. Shah, B. Patel, and D. Garg, "Software defined network (SDN) implementation with POX controller," in *2022 3rd Int. Conf. Smart Electron. Commun. (ICOSEC)*, Trichy, India, IEEE, 2022, pp. 65–70.