IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 18, 2019, accepted October 30, 2019, date of publication November 29, 2019, date of current version December 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2956805

Learning Based Adaptive Network Immune Mechanism to Defense Eavesdropping Attacks

MINGYUAN LIU[1], DEYUN GAO[1], GANG LIU[1], (Student Member, IEEE), JINGCHAO HE[2], LU JIN[1], CHUNLIANG ZHOU[1], AND FUCONG YANG[1]

[1]School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
[2]School of Telecommunications Engineering, Xidian University, Xian 710126, China

Corresponding author: Deyun Gao (gaody@bjtu.edu.cn)

This work was supported by the Fundamental Research Funds for the Central Universities of China under Grant 2019JBM011.

ABSTRACT Encryption mechanisms improve the security of transmitting data. Nevertheless, attackers might silently eavesdrop packets to crack sensitive information or launch cyberattacks to damage the network performance of victims. In this paper, we propose a learning based adaptive network immune mechanism (LANIM) to prevent the eavesdropping attacks. Specifically, LANIM is equipped with three *defense lines* and one *constraint*. The first defense line focuses on making decisions about abnormal network conditions by the minimum risk machine learning algorithms. The second defense line is the encryption strategy which focuses on the intent and application. The programmable devices implement novel policies such as multipath transmission and packet encapsulation. LANIM inherits the existing countermeasures based on computational complexity, which is the third defense line. Besides, the policy dynamically updates with random seeds is the constraint of the cyberattack. The attackers have to conquer these three immune lines before the new policy update otherwise the offensive is shattered. We implement LANIM with the P4 language and the Smart Identifier Network (SINET) framework, evaluate the ability to resist the hazards of eavesdropping attacks and explore the trade-offs between security and performance.

INDEX TERMS Machine learning, security and performance trade-offs, adaptive network immune mechanism, eavesdropping attacks, SINET.

I. INTRODUCTION

Nowadays, the Internet has become an essential element in human life. A large amount of information is stored or transmitted on the network, and some are not even encrypted, which has become the main target of attackers. For example, the hypertext transfer protocol (HTTP) is widely used by online services without encryption mechanisms. There is a significant risk of eavesdropping when the user sends sensitive information through HTTP.

There are two general approaches to solve existing problems: improving the existing network and constructing a new network architecture. Security is an indispensable issue for designers in future network architecture designs. Named Data Networking (NDN) is a data-centric network architecture [1], [2]. Its purpose is to clarify the intent of data traffic and prevent attacks. The network system can monitor the required data and strengthen the control of network traffic security. Locator Identifier Separation Protocol (LISP) and MobilityFirst network architecture decouple user information from an access network by separating routing locators and user identifiers, which can protect the security and integrity of user information [3]–[5]. These network architectures give us much inspiration. However, man-in-the-middle (MITM) attacks and cyber-eavesdropping are not entirely solved and may even get worse because the network design concept and architecture are different [6]. Software defined networking (SDN) is an emerging networking paradigm, which enables the rapid network innovation [7]. SDN is featured with flexible programmability by decoupling the data plane and the control plane [8], which brings agile and flexible network management and promotes network innovation. Recently, the research community has recognized the limitation of the OpenFlow data plane. Some recent efforts have been devoted to enhancing the programmability of the SDN data plane [9], [10]. Nevertheless, this new architecture

The associate editor coordinating the review of this manuscript and approving it for publication was Chi-Yuan Chen.

still has security problems in conventional networks such as probe attack, man-in-the-middle attack.

The identity mapping system of the Smart Identifier Network (SINET) dramatically reduces the illegal network traffic [3]. Programming Protocol-Independent Packet Processors (P4) can maintain information in the network components during runtime based on its register data structure, aiming to enhance the network components programmability [10]. P4 is a typical language that allows the flexible definition of protocol header fields, parsers, and tables [11]. Therefore, it proposes an abstraction for general network component behaviors that brings the possibility of a custom forwarding process [9]. Programmable network components can flexibly provide multiple security services to the top layer in the SINET architecture.

In 2006, Bellare, Kohno, and Shoup introduced the concept of ''stateful PKE'' (StPKE), which requires the senders to maintain some state information [12]. Alternatively, the classical PKE schemes are called stateless PKE [13]. State encryption mechanisms generally increase the overhead of communication, this problem has been noted in some studies [14], [15]. Other studies have implemented real-time changes in encryption policies based on the Network Time Protocol (NTP) and Precision Time Protocol (PTP) [16], [17]. The precise synchronization of these protocols incurs a lot of overhead. Our method relies on the timestamp update, which does not need synchronization. At the same time, LANIM has the flexibility to customize the scalability header which enables packets to be forwarded in program design network components.

Currently, the reliability of most security algorithms depends on the complexity of large-scale factorization. Therefore, the more substantial threats to encryption methods such as Rivest-Shamir-Adleman (RSA) algorithm comes from the continuous improvement of computing power and the factorization algorithms. Hackers still have the possibility to crack shorter keys. Meanwhile, the existing obfuscated encryption methods of the Internet are mostly static and cannot be dynamically updated.

The factors of network system security are complex and unpredictable. We mainly focus on security in the temporal and spatial. From the perspective of time, it will be affected by economic and political factors. For example, network administrators might face an unpredictable risk of a large number of attacks on a particular date. From the perspective of space, the location of the network system deployment and the attributes of the access users will affect the quality of traffic on the network. Different network scenarios put forward different security level requirements. Due to both time and space factors, the configuration of network security is very complicated for network administrators. It is vital to solve this problem with an intelligent solution. Therefore a learning-based adaptive network immune mechanism (LANIM) is proposed to achieve this goal.

The structure of this paper is as follows. We present the problem and related work in Section I. Section II presents the background of network architecture and eavesdropping attack. We describe the model of the learning-based adaptive network immune mechanism in Section III. We describe encryption policy defenses and the process of dynamic updates. Section IV introduces the machine learning method of network perception minimum risk which is the intelligent defense line of the immune system. Section V describes a method of measuring network performance based on network entropy. Section VI explores the trade-offs between security and efficiency of LANIM.

II. BACKGROUND

Fig. 1 shows an intelligent security transmission architecture in SINET. Different from the current Internet, SINET featured with three vertical layers: Smart Pervasive Service Layer (L-SPS), Dynamic Resource Adaptation Layer (L-DRA), and Collaborative Network Component Layer (L-CNC). L-SPS is used for the registration and management of services. L-DRA organizes network functional groups to resource allocation and makes optimal decisions. A group of network components or nodes with similar functionality form a functional group. These functional groups can implement in-path caching, mobility support, security enhancements, etc. L-CNC consigns network components to perform specific tasks such as routing, data transfer and caching. [18].

Fig. 1 illustrates an example application involving security services. In the top layer (L-SPS), several service resolution units (SRUs) are used to register the provider's services, such as security SRUs and video SRUs. For security services, each security policy first registers at the security SRU with appropriate service identifiers (SIDs) and service behavior descriptions (SBDs). If the user or smart controller requests a security service, the security SRU processes the request message to determine the candidate SD and SBD. Then, the requirements of the required services are propagated to the L-DRA, and several network

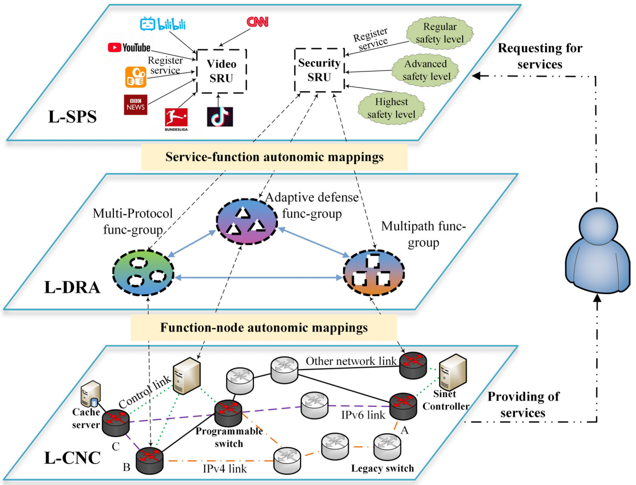


FIGURE 1. An application illustration in SINET: intelligent security transmission architecture.

functional groups are logically constructed in the L-DRA through the service function mapping process. There are three functional groups in this example. These functional groups manage network nodes to provide the desired functions through the function-component mapping process [3]. As Fig. 1 shows, the L-CNC contains distributed intelligent SINET controllers, programmable switches and legacy switches. There are control links between the intelligent controller and the programmable switches. Intelligent SINET controllers can sense network attacks and deliver response strategies from a global perspective. L-CNC supports multiple protocols such as IPv4, IPv6. The programmable switches can deploy P4 programs to change forwarding policies and coexist with other legacy forwarding devices. For example, network tunnel technology can establish communication between switch A and switch B. At the same time, switch A to switch B constitute a multi-path, cross-protocol transmission scenario. Eventually, user or smart controller will acquire this security service through the cooperation of corresponding network nodes. The mechanism we proposed is to coordinate func-groups based on the perceived network environment to provide different levels of security services.

The purpose of LANIM is to avoid the hazards of eavesdropping, rather than accurately predicting the occurrence of eavesdropping attacks. Eavesdropping attacks are classified into global eavesdropping and packets eavesdropping.

Global attacks are also known as session message attacks. An attacker eavesdrops on all packets of a session. Then, these packets are parsed and decrypted according to the network protocol to yield the complete session information [19]. The premise of the global attack is the attacker can eavesdrop on all the transmission paths and reversely analyze the encryption method of each packet.

Packets eavesdropping attacks are also known as partial eavesdropping attacks. The attacker does not steal all session data packets, which is different from session message attacks. The ultimate goal of the attack is to initiate other network attacks against the target system based on the small amount of eavesdropped information. For example, an eavesdropper statistically analyzes part of the data stream to obtain source-destination address. From the results of this analysis, DDoS attacks, replay attacks, and MITM attacks are further performed [20].

There are three levels in the network eavesdropping attack chains.

(1) The attacker silently invades the network path by traffic mirroring and copies the network packet.

(2) The attacker parses the network protocol of the acquired data packet, extracts and decrypts the packet header or payload.

(3) The attacker recomposes the content of the session message with all the stolen information or use partial information to launch other attacks. The attacks degrade network performance and paralyze the network.

The main goal of a cyber-tapping attack defense is to break these three kill chains as much as possible. Although intrusion detection is also a means of defense, LANIM defends the system by destroying the second and third stages of the chain.

LANIM utilizes a stateful policy transformation mechanism to immune global eavesdropping attacks. This mechanism makes it extremely difficult to crack all packets of a session in the valid time. LANIM uses machine learning to prevent the network attacks. These network attacks are the ultimate goal of packets eavesdropping behavior. By blocking the packets eavesdropping attack chain, LANIM smashes the opponent's conspiracy and protects the system's performance. In the next section, we will elaborate the learning-based adaptive network immune mechanism.

III. VIEW ON LANIM

Learning-based adaptive network immune mechanism takes the unpredictable offset time in the network as seeds to dynamically update the encryption policy. The LANIM contains timestamps to prevent replay attacks and constantly changing encryption policies enhances the security of system. The seed of the selection algorithm is the offset of the system time. We define a new package format with 4 bytes for state and decision domains. The LANIM becomes more intelligent and expandable as a consequence of the framework of SINET and P4.

A. FORMAT OF PACKET

We define a flexible package structure to implement stateful encrypted communication and adopt programmable switches to process and forward customized packets. The P4 language defines the processing and forwarding logic of programmable switches.

TABLE 1. The format of the datagram for stateful encrypted communication.

Decision Domain	UF(1bit)
	CF(1bit)
	PID(14bit)
		Update the encryption policy
		Confirm the new policy
		Indicate the encryption policy

State Domain	TS(32bit)	The seed of policy updates

Table 1 defines the datagram format for stateful encrypted communication. We consider the options domain for the IPv4 packet to reduce the overhead. However, due to the lack of options domains in the IPv6 protocol, custom fields occupy the extended header of the IPv6 packets. The interaction message between hosts contains two domains: the state domain and the decision domain. We classify the update flag (UF), confirm flag (CF) and policy identification (PID) as the decision domain, and the time stamp (TP) as the state domain. Each stateful encrypted packet contains four fields: UF, CF, PID, and TS. A brief description of each field is given in Table 1.

UF: The length of the update flag is 1 bit. The value of UF is 1 during the encryption policy update phase and the updates are not allowed if UF $= 0$. The value of the update flag is set by thresholds, users and controllers.

CF: The length of the confirm flag is 1 bit. When the receiver receives a packet that cf is 1, the receiver confirms that the sender's encryption policy has been updated in accordance with the negotiation and sents the packets according to the new encryption policy. At the period of the initial communication, the value of CF is 1 in the policy Acknowledgement (PACK) packet.

PID: The length of policy identification is 14-bit. PID indicates which encryption policy is adopted in the next phase during the policy update. PID is a unique identifier for policy and maps to the policy database. PID and encryption policy is a set of key-value pairs (KVP).

TS: The length of the timestamp is 32-bit. The timestamp is the state domain of packets. The seed of the policy selection action is the time parameter for the packet. The length of system timestamp is 64-bit, the first 32 bits are the integer parts and the last 32 bits are fractional parts. In order to reduce the overhead, the middle 32 bits are available. TS is consists of the last 16 bits of the integer part and the first 16 bits of the fractional part.

A maximum transmission unit (MTU) is the largest size packet or frame, specified in octets, that 1500 bytes are Ethernet standard MTU. The stateful encryption method we proposed occupies 4 bytes in the packet, which is about 0.4% of the MTU. In the next section, we present how the hosts negotiate encryption policy during the first communication.

B. POLICY INITIALIZATION

In this section, we describe the initialization process of phase negotiation, which is a specific example of LANIM. When the first communication starts, the communication hosts exchange the summary of the policy database and synchronize the policy information of the database. Policies shared by the two hosts are selected out as the policy set for encrypted communication between hosts. The receiver and sender confirm the adopted policies during initialization. The smart controller implements the maintenance of encrypted database through secure network links. The schematic diagram of the process is shown in Fig. 2. There are four steps in the initialization phase.

Stage 1: Host A is the initiator of the communication. Host A sends the policy database description (PDD) packet whose payload contains indexes of the confusion encryption strategy database.

Stage 2: Host B begins communication after receiving a packet from Host A. Host B proofreads the local encryption policy database based on the database index extracted from the packet. Thus host B get the B obtained encryption strategies maintained by both hosts. Host B encapsulates the intersection of the two policy database indexes in the policy database response (PDR) packet and sends PDR to host A.

Stage 3: After receives the PDR packet, host A selects the encryption policy according to the process in Fig. 3, and stores this policy in the register. Then host A loads the index value of encryption policy in the policy request(PREQ) packet and sends this packet to host B.

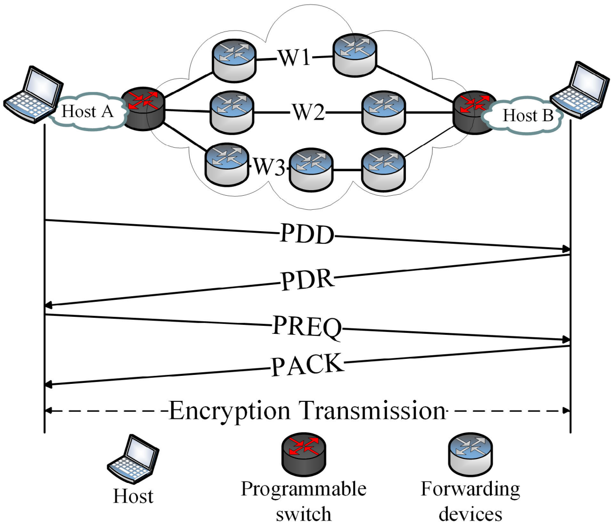


FIGURE 2. The hosts negotiate the encryption policy through a series of packets. This process is for the first encrypts communication or controllers issue encryption policy database updates.

Stage 4: Host B parses the PREQ packet to obtain the encryption policy selected by host A. Then, host B stores this policy in the register and sends the policy Acknowledgement (PACK) packet to host A.

Stage 5: Host A receives the policy acknowledgment packet. Then host A encapsulates packets with the encryption method confirmed by each other and the encrypted communication is established.

We transmit the index of encryption policies instead of the plain text of the encryption policies to determine the encryption policy. The communication hosts select the corresponding encryption policy according to the index in the encryption policy database. A smart controller issues the maintenance instruction of the encrypted database through a secure path.

C. ENCRYPTION MECHANISM DESIGN

This section describes how the stateful encryption system works in LANIM. Our approach is not to abandon existing security methods. We propose a novel security mechanism based on the programmable network component in the SINET architecture. This solution enhances safety and flexibility compared to the original security scheme, which can be used in specific security equipment to accommodate different security scenarios, such as security agencies, railway networks, etc. We apply this security mechanism to the granularity of the stream and packet.

We propose a novel encryption transmission scheme, in which the encryption and decryption of each data packet do not require synchronization time. After initialization, a stateful encrypted transmission is performed between the communicating hosts. Each packet contains the sender's system timestamp. Network jitter and the difference in system time are utilized to record the network offsets. Different offsets are obtained as the seeds of the policy selection. Although the timestamps are contiguous, the encryption strategy is not strictly continuous. Consequently, the scheme is difficult to predict and crack. The decision domain of the packet determines the policy changes, whose settings are discrete and encrypted.

The value of the update flag in the packet decision domain is to enable update encryption. It changes in three ways: 1) Updated regularly by the threshold. 2) The controller settings. 3) User Settings.

Updating the policy through time threshold is the most common method, but there are some limitations. This method cannot control the unexpected malicious behaviors in the network system. For example, paths in a multipath system are affected by a malicious attack, which can reduce the throughput and response rate of the network system and affect network performance. The usual way to deal with this situation is to increase the frequency of policy updates, but changing encryption policies frequently can increase the impact of the network. In the multi-path communication system, the host perceives the capability of each path through the routing strategy then continuously optimizes the network performance according to the delay, bandwidth, packet loss rate. The system will gradually converge to a stable and effective transmission state. The increasing oscillation of the routing algorithm stems from frequently changes routing strategy, which can disrupt transmission efficiency and even make the network unavailable.

The cache server caches the link information of the forwarding links. The control link between the cache server and the SINET controller reduces the overhead of each switch in the forwarding links. The cache server preprocesses the data to obtain the characteristic value of the data set. The controller determines whether the encryption policy needs to be updated based on the link performance and the minimum risk algorithm. If an exception occurs in the network, the controller sends an update command to the client via the switch, and the client sets the update flag to 1. This mechanism enables users to manually update encryption policies in case of emergency. The user can initiate a policy update request with control commands.

The policy is selected according to the timestamp when the host enters the policy change phase. T_{n-1} is the timestamp from the received packet. T_n is the timestamp of the host. We take T_{n-1} and T_n as the seeds of the algorithm, which determines the strategy selected in the next phase. The process of selecting an encryption policy by timestamp is shown in Fig. 3. In section VI, we evaluate the computation complexity of this process in Fig. 3.

Fig. 4 brief reports the process of encryption policy update based on UF. After getting the packet, the host parses the packet with the encryption method stored in the register. Secondly, the host checks the decision domain and state domain of the packet. If the UF = 1, the host starts the update step, otherwise waits for the next packet.

In particular, the advantage of this approach is that it reduces overhead and improves efficiency because clients do not have to synchronize. On the other hand, LANIM prevents man-in-the-middle attacks because of checking timestamps. While various methods can improve security, too much computation challenges forwarding efficiency, and we will discuss trade-offs in overhead and benefits in later sections.

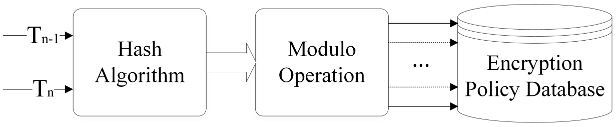


FIGURE 3. The hosts hash the timestamps. The hash value is converted to the index value based on the modulo operation. The hosts save the encryption policy index in the register and use the index value to select the encryption method.

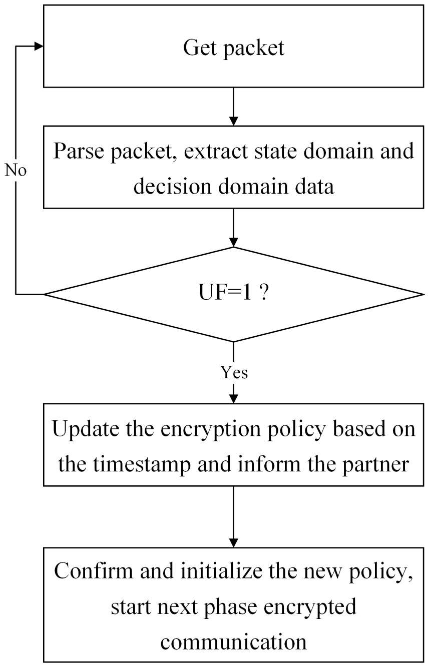

Get packet
Parse packet, extract state domain and decision domain data
No
UF=1 ?
Yes
Update the encryption policy based on the timestamp and inform the partner
Confirm and initialize the new policy, start next phase encrypted communication

FIGURE 4. Policy update flowchart.

D. AN USE CASE

In this section, we present an example of a controller update. Fig. 5 shows a use case. After initialization, the two users establish a communication link. There are control links between the controller and the users. The controller collects network state information and determines the security level of the network environment through intelligent algorithms.

The controller issues the command when the network exception is detected. The controller sends update instructions (packet ①) to user A, which sets the update flag to 1. User A performs policy update operation based on $S_n = (T_n - T_{n-1})_{Algorithm}$ and then encapsulates the new policy S_3 into the packet ②.

After receiving packet ②, user B writes the new encryption mode into the register, sets CF to 1, and sends an update confirmation packet to A with the original encryption policy.

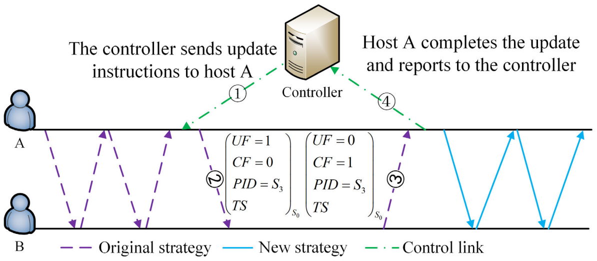


FIGURE 5. The controller updates the encryption policy.

Packet ③ is a update confirmation package with effective confirm flag. After receiving the update confirmation package, user A reports the controller and begins to send packets with the new encryption strategy. The controller clears the cache when it receives the update report from the user.

IV. MACHINE LEARNING ALGORITHM

Learning-based adaptive network immune mechanism has intelligent controllers. The SINET controller senses the state of the network and makes decisions according to the machine learning model to improve network performance. The cache server caches packets in the network and extracts features. The intelligent divides network traffic into regular traffic and abnormal traffic controller with machine learning algorithms. This section will introduce the motivation of anomaly detection, the selection of data sets, and the machine learning algorithm based on minimum risk.

A. OUTLIER DETECTION

Outliers are the data which does not match with the rest of the data in the dataset [21]. Outlier detection distinguishes anomalous data hidden in a normal data set. In a sense, anomaly detection is also a classification problem. The flow in the network has two categories: normal and abnormal. Our goal is to determine which type is more appropriate for the observed data stream. It is then concluded that anomaly detection is more suitable for discovering variations of known attacks than discovering unknown malicious activity. In this case, the system can be trained with known attack samples and regular background traffic to enable a more reliable decision-making process.

A basic rule of supervised machine learning is to train a system with samples of all classes. The training set for each class should include abundant representatives. However, the occurrence of abnormal network traffic is accidental, and abnormal data and regular data are often unbalanced. The basic idea of the oversampled SMOTE algorithm is to analyze a small number of samples and generate new samples based on a small number of samples, and add new samples to the data set [22]. SMOTE provides a method of data balancing, which indicates that balanced data sets play an essential role in machine learning. Reinforcement learning keeps learning knowledge in the interaction with the environment according to the rewards or punishments obtained [23], but it requires a large amount of computation and a long response time.

In this paper, we regard exception detection as a binary problem, and the controller determines whether a behavior is normal or abnormal. Data sets based on binary classification and machine learning algorithms are provided in the next section.

B. DATASET DESCRIPTION

A public IP address will naturally receive a large amount of traffic flows when it is exposed to the network. Some of these traffic flows are necessary services, and some are not expected. A set of characteristics can describe each network access. Packets in the data stream can be captured and recorded for research and outlier detection. The machine learning model classifies packets received in the future.

The KDD99 data set was compiled by Stolfo *et al.* and was widely used in the evaluation of anomaly detection methods [24]. The establishment of KDD99 is based on the data collected by the US Department of Defense's Advanced Research Projects Agency in 1998. These data have been criticized by McHugh [25]. The imperfection in the KDD99 data set is that a large number of redundant records can lead to skewed statistics results which have worse detection rates on the low-frequency records. NSL-KDD99 improves the aforementioned datasets. There are no duplicate records in the NSL-KDD99 datasets. Redundant records are removed to enable the classifiers to produce an unbiased result [26]. There are four types of attack records available in the data set: DOS, U2R, probe, and R2L. A brief description of the aforementioned attacks is as follows.

Denial of Service Attack (DOS) [27], [28]: A DOS attack prevents the victim host from receiving and processing external requests or unable to respond to external requests promptly. The attacker creates a large amount of useless data to congest the network that leads to the attacked host so that the victim host may communicate with the outside world regularly. Such as SYN flood attacks [26].

User to Root Attack (U2R): The attacker logs in to the victim system using a seemingly innocuous account, circumventing some authentication or exploiting the vulnerability of the system or website. The attacker gets root (highest authority) permission and then logs in for some illegal operations.

Remote to Local Attack (R2L): An attacker gains local access to the machine remotely using unauthorized access from a remote computer, filtering out data from the machine, modifying data. The attacker conceals his true identity and pretends to be an innocuous user.

Probing Attack: The act of trying to gather information about a computer network. Defining a probe refers to an attack on a computer network or DNS server to obtain a valid IP address, active port number, host operating system type, and security vulnerability.

The distribution of these four attacks across NSL-KDD_20 percent data set is shown in Table 2. Based on the conclusions of the previous section, we divide all the elements in this dataset into two categories, normal and abnormal. We take five continuous features that are selected

TABLE 2. NSL-KDD_20 percent data set.

Type	Support
Dos Class	9234
Probe Class	2289
U2R Class	11
R2Lclass	209
Danger Class	11743
Normal Class	13449

from the 41 features of the data set as the data set of machine learning [29]. The continuous features are as follows. Num_failed_logins: the number of failed user login attempts. Num_root: the number of root user accesses. Serror_rate: the percentage of connections that have "SYN" errors in the last two seconds of connections with the same target host as the current connection. Srv_serror_rate: the percentage of connections that have "SYN" errors in the last two seconds of the connection with the current connection. Dst_host_srv_count: the number of connections in the first 100 connections that have the same service as the current target host. We randomly divide the data set into test sets and training sets.

According to the characteristics of packets eavesdropping attacks in section II, we find that eavesdropping attacks are positively correlated with attacks in NSL-KDD99. The application of NSL-KDD99 data set can break the attack chain of packet eavesdropping attacks and prevent possible harm. In the next section, we describe the application of the minimum risk algorithm in Bayesian classifier.

C. MINIMUM RISK BAYESIAN CLASSIFICATION

The risk function states the cost of each action. The minimum risk method converts probabilities into a decision. The goal of adopting the machine learning model is to minimize classification errors and pursue the minimum error rate.

In Bayesian classification, we analyze the training set to obtain the probability distribution function and the prior probability of each parameter. We have a hypothesis that the given instance appertains to a particular class then calculates a posterior value to allocate a class label for the test instance in the minimize risk. The approach requires only one scan of the whole data [30]. It is a practical and fast method for some classification problems.

The probability density function of continuous variables conforms to the Gaussian distribution. In our method, we use a normal distribution for likelihood estimation. Equation (1) presents the probability density function of the continuous variable.

$$P\left(x_i|c\right) = \frac{1}{\sqrt{2\pi}\sigma_{c,i}} \exp\left(-\frac{\left(x_i - u_{c,i}\right)^2}{2\sigma_{c,i}^2}\right) \quad (1)$$

Equation (2) is the posterior probability of sample x.

$$P\left(\omega_j|x\right) = \frac{p\left(x|\omega_j\right)P\left(\omega_j\right)}{\sum_{i=1}^{c} p\left(x|\omega_i\right)P\left(\omega_i\right)}, \quad j = 1, \ldots, c \quad (2)$$

The loss function describes the loss of the decision. Where α_i is the action and ω_j denote the natural state.

$$\lambda\left(\alpha_i, \omega_j\right), \quad i = 1, \ldots, k, \ j = 1, \ldots, c \quad (3)$$

We set up a risk matrix to improve the high false positive rate and high false positive rate of the detection system. Network administrators flexibly adjust risk matrices to address different security scenarios. The risk matrix R_{ij} is

$$R_{ij} = \begin{pmatrix} \lambda_{11} & \cdots & \lambda_{1j} \\ \vdots & \ddots & \vdots \\ \lambda_{i1} & \cdots & \lambda_{ij} \end{pmatrix} \quad (4)$$

If we make a decision α_i for sample x, the conditional risk is

$$R\left(\alpha_i|x\right) = \sum_{j=1}^{c} \lambda\left(\alpha_i|\omega_j\right) P\left(\omega_j|x\right) \quad (5)$$

Equation (5) infers the risk of various actions, and then equation (6) selects the action of the minimum risk. Bayesian risk is the minimum overall risk, and the risk of the minimum risk decision is Bayesian risk.

$$\alpha = \underset{i=1,\ldots,k}{\mathrm{argmin}} R\left(\alpha_i|x\right) \quad (6)$$

Anomaly detection in the system is defined as a binary problem. In order to break the kill chains of eavesdropping attacks, we do not classify the attack accurately, and the system only judges whether the traffic is dangerous or normal. Therefore, the controller applies the above method to make the minimum risk decision for the network flow.

D. RESULT METRIC

In the field of machine learning, the matching matrix visualizes algorithm performance. T (True) and F (False) evaluate whether the judgment result of the model is correct. P (Positive) denotes the number of real negative cases in the data. N (Negative) denotes the number of real negative cases in the data. Here are four values:

(1) True positives, TP. The model correctly classified positive samples

(2) True negatives, TN. The model correctly classified negative samples.

(3) False positives, FP. The negative sample is predicted to be positive by the model and there a false alarm.

(4) False negatives, FN. The positive sample is predicted to be negative and there is a false negative.

In order to measure the performance of the classifier, equation (7) shows the concept of accuracy.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

Precision refers to the proportion of real cases (TP) in all positive cases (TP+FP) judged by the model. If the target limits the number of false positives, then the predicted rate can be used as a performance indicator. When the model has a high prediction rate, the prediction rate is called a positive predictive value (PPV).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

The recall rate refers to the ratio of the positive case (TP) correctly judged in the model to all positive cases (TP+FN). As a performance indicator, recall rates are more concerned with classifying positive samples as negative samples (FNs).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

In general, when the precision is high, the recall is low. A method to balance both the predictive rate and the recall rate is the fractional F-score. F-score is the harmonic mean of precision and sensitivity.

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{10}$$

The data set is divided into 75% training set and 25% test set. The training set is an input to the machine learning model to train the model. The test set verifies the performance of the model. Table 3 shows the measurements of predicted results.

TABLE 3. Evaluation of results.

	precision	recall	F-score	support
Normal	0.88	0.9	0.89	3695
Danger	0.88	0.86	0.87	3197
avg/total	0.88	0.88	0.88	6892
The test set accuracy: 0.88				

In the 3,695 positive samples, the prediction rate is 0.88, the recall rate is 0.9, and the F-score is 0.89. In the 3,197 negative samples, the prediction rate is 0.88, the recall rate is 0.86, the F-score is 0.87, and the test set accuracy is 0.88. The result shows that our model is capable of distinguishing the most harmful network behaviors

The accuracy-recall curve shows the trade-offs between recall and accuracy for different thresholds. Each point on the curve corresponds to a possible threshold for the decision-function. For example, in Fig. 6 we note that the recall rate is 0.8 at a location with an accuracy of about 0.95.

The points in the top right corner indicate the high accuracy and the high recall rate for the same threshold. The curve starts from the top left corner, which corresponds to a low threshold, and all samples are classified as positive. The performance of the model gets better when the curve approaches the upper right corner. Increasing the threshold allows the curve to move in a more accurate direction, which decreases the recall rate.

The receiver operating characteristics curve is referred to as the ROC curve. Fig. 7 shows the ROC curve of the learning model. Similar to the precision-recall curve, the ROC curve takes into account all possible thresholds for a given classifier, but it shows the false positive rate (FPR) and the true positive rate (TPR). A perfect classifier does not have any prediction errors and the area under curve (AUC) is 1.

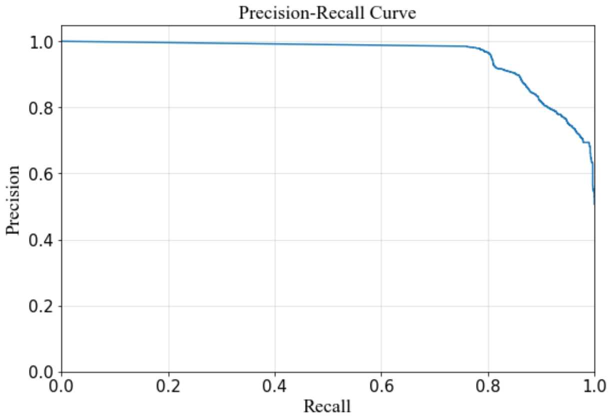


FIGURE 6. Precision-recall curve.

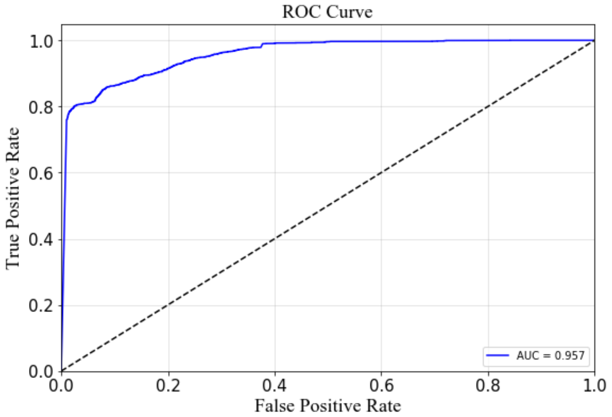


FIGURE 7. ROC curve.

An ideal ROC curve is close to the upper left corner. We hope the classifier will have a high recall rate while keeping the lower false positive rate.

V. NETWORK ENTROPY: A METHOD OF SYSTEMATIC EVALUATION

There are many factors that influence network performance, so it is difficult to measure the network performance with a single physical parameter. A network entropy weight-based approach is adopted to capture the complexities of the phenomenon. We propose a novel method to characterize the capabilities of the system. We provide a novel method based on network entropy to characterize the state change of the network and propose a network quality parameter (Q) that characterizes the capabilities of the system.

A. NETWORK ENTROPY

Entropy is a measure of system confusion, and information entropy is a concept from information theory. According to this theory, if an event contains more information, then its uncertainty and randomness are stronger. In general terms, this means information increases uncertainty or entropy [26]. We choose the evaluation indicators and obtain the security measures of the network system by quantifying the performance evaluation indicators. Network entropy reflects different levels of uncertainty of network systems. The formal description of network entropy is as follows.

In an observation, supposing the set of parameters to describe network performance is X, which has $\{x_1, \ldots, x_n\}$. We define m observations to form a performance matrix $\mathbf{P}'$, the rows of the matrix are network parameters, and the columns are the number of observations. The performance matrix is defined in equation (11).

$$\mathrm{P}' = \left(p'_{ij}\right)_{\mathrm{n*m}} \tag{11}$$

We divide multiple network quality parameters into two categories: cost parameters and benefit parameters. The minimization of cost parameters is what we expect. The elements in the matrix $\mathbf{P}'$ are dimensionless. The cost parameters are normalized by equation (12), such as delay, packet loss rate, etc.

$$p_{ij} = \frac{\max_i \left\{p'_{ij}\right\} - p'_{ij}}{\max_i \left\{p'_{ij}\right\} - \min_i \left\{p'_{ij}\right\}} \tag{12}$$

We hope that the value of benefit parameters is close to the maximum, such as bandwidth, throughput. Equation (13) normalizes the benefit parameters.

$$p_{ij} = \frac{p'_{ij} - \min_i \left\{p'_{ij}\right\}}{\max_i \left\{p'_{ij}\right\} - \min_i \left\{p'_{ij}\right\}} \tag{13}$$

The new matrix is formed in equation (14). P_{ij} represents the normalized value of the parameter i of the observation j.

$$\mathrm{P} = \left(p_{ij}\right)_{\mathrm{n*m}} = \begin{pmatrix} \mathrm{p}_{11} & \cdots & \mathrm{p}_{1\mathrm{m}} \\ \vdots & \ddots & \vdots \\ \mathrm{p}_{\mathrm{n}1} & \cdots & \mathrm{p}_{\mathrm{nm}} \end{pmatrix} \tag{14}$$

The entropy of each element in the matrix is defined as follows, where h_{ij} is entropy.

$$\mathrm{h}_{ij} = -p_{ij} \log_2 p_{ij} \tag{15}$$

Each column of the matrix is a survey of network performance. Network entropy is the expectation of each attribute entropy. Equation (16) shows the network entropy H, where $i \in \{1, 2, \cdots, n\}, j \in \{1, 2, \cdots, m\}$, w_i denotes the weight of each attribute.

$$H_j = \sum w_i \times h_{ij} \tag{16}$$

Furthermore, let ΔH denotes the network entropy change. ΔH can be interpreted as an impact on the system after adopting a new strategy. $\Delta H > 0$ means the system performance degradation and more delivery overhead. In a network attack environment, ΔH denotes the effect of an attack on the network system. Cyber-attacks are destructive when ΔH is large.

$$Q = 1 - H \tag{17}$$

Equation (17) defines the system quality parameter Q. According to equation (16), we can confirm that Q is between 0 and 1. The quality of the system is better when the value of the parameter Q is closer to 1.

B. THE WEIGHTED METHOD

In the previous section, we introduced the concept of network entropy. In this section, we will discuss how to determine the weight of entropy. We believe that for the fairness of the multi-parameter system, the weight is a means to balance the parameters.

The Analytic Hierarchy Process is used to construct a pairwise comparison judgment matrix, the leading eigenvalues and eigenvectors of the matrix are obtained. Then we conduct a test of the judgment matrix. We eliminate internal conflicts of data and verify the validity of the data through consistency testing. Therefore, we derive the weight of each attribute in the network entropy such as throughput, RTT, delay and delay jitter. Setting the weight of each indicator to reflect its contribution to the overall network entropy is a crucial and difficult point. The value of w is determined according to the purpose of the attack and the type of network service. In order to improve security and computational complexity, we mainly consider the loss of processing delay. In order to ensure information security, users accept the appropriate amount of bandwidth reduction. For instant messaging network systems consisting of satellite networks or heterogeneous networks, the impact of packet loss rate and delay jitter is significant.

In practical applications, the weights of the attributes are obtained by establishing a judgment matrix for each indicator. The process is as follows.

Firstly, we establish a pairwise comparison judgment matrix. The judgment matrix is a comparison of the relative importance of a set of attributes $\{A_1, \ldots, A_n\}$ under the constraint condition. In different application scenarios, decision-makers believe that the importance of each attribute is different, and their weights are not the same. According to the Delphi method, the numbers 1-9 and their reciprocals are used to describe the network state description parameters. Equation (18) defines the judgment matrix A.

$$\mathrm{A} = \left(a_{ij}\right)_{n*n} \tag{18}$$

$\lambda_{\max}$ is the unique maximum eigenvalue of the judgment matrix, and the components of W are positive. Finally, the weight vector obtained is normalized in equation (19).

$$Aw = \lambda_{\max} w \tag{19}$$

$$CI = \frac{\lambda_{\max} - n}{n - 1} \tag{20}$$

$$CR = \frac{CI}{RI} \tag{21}$$

Equation (20) calculates the consistency index (CI), n is the number of attributes. Further, Table 4 obtains the value of the random consistency index (RI). Consistency ratio (CR) as defined in equation (21). When CR < 0.10, it is deemed that the consistency of the judgment matrix is acceptable [31].

TABLE 4. Average consistency index.

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14
RI	0	0	0.52	0.89	1.12	1.24	1.36	1.41	1.46	1.49	1.52	1.54	1.56	1.58

Otherwise, the judgment matrix should be modified appropriately, and the values of elements of the judgment matrix should be adjusted.

Thus far, the section introduces how to determine the weight of each network entropy. According to the network entropy weight method, we can synthesize various parameters and get a fair parameter to measure network performance. In the following experiments, we will take the network entropy change ΔH and network parameter Q as indicators.

VI. SIMULATION RESULTS

In this section, we will discuss the ability of LANIM to resist eavesdropping attacks. We use dynamic update strategies to encrypt packets. A set of session messages are encrypted by different policies, which makes global eavesdropping attacks extremely difficult. For packets eavesdropping attacks, the learning mechanism breaks the kill chains and smashes the opponent's conspiracy which protects the system's performance. The deployment of P4 programs and the choice of strategies increase the computational complexity. Meanwhile, we discuss security and performance trade-offs in this section.

A. SAFETY ASSESSMENT

To obtain the complete session information, hackers have to take all packets of all the network links and successfully decrypts and reorganizes them. LANIM is dynamically updated based on timestamps. The encryption policy for each packet is difficult to predict because it is related to random network offsets before the policy update. Using brute force to crack this mechanism will take an unacceptable time.

1) DYNAMIC UPDATE

The constraint of defense are dynamic updates. If a hacker wants to crack a session message successfully, he must get all the packets and crack them before the new update. It is difficult for hackers to get all the packets from the multi-path background traffic. The dynamic encryption shortens the available time for cracking.

2) INTELLISENSE

The first line of defense is intellisense. The smart controller applies the minimum risk Bayesian algorithm to detect threats and avoids them in time. LANIM adopts multiple risk functions to adapt to different spatial and temporal dimensions. Hackers have to beat machine learning algorithms to break into the network system. The machine learning mechanism breaks the kill chain of packets eavesdropping attacks and prevents the harmful phase.

3) ENCRYPTION STRATEGY

The encryption policy for the second line is stored in the encryption policy database. This plan is based on intent and application which is different from traditional encryption mechanisms based on computational complexity. The transceiver system time difference and network latency offset are the seeds of the policy selection operation. The seed is transformed into the policy index through the hash algorithm and the modulo operations. We transmit the index in the network instead of the plaintext of the encryption policy. This strategy allows a set of packets to be encrypted in multiple methods, and the attacker's decryption time will increase exponentially. The multi-path policy can change the transmission links and construct multi-path transmission schemes. The middlemen are unable to get all the packets in multiple transmission links to prevent the eavesdropping attacks.

4) EXISTING MECHANISMS

The third defense line is the existing encryption method for packets and networks. Take the typical encryption algorithm AES-256 adopted by TLS/SSL as an example. An average of 2^255 $\approx$ 5.8*10^76 random numbers are used as the key for encryption and decryption. Suppose the hackers use brute force to crack the method to find the right key, which will consume a lot of computing resources. Using the computing power provided by the bitcoin network, the time required for 2^255 AES operations is 2^255/2^64.4753 $\approx$ 2.3*10^57 seconds $\approx 6.3 * 10$^53 hours $\approx 2.6 * 10$^52 days $\approx 7.2 * 10$^49 years. Brute force cracking is almost impossible to achieve, but mathematical algorithms may greatly reduce computational time in the future. Therefore, adaptive encryption algorithm based on intent and application has great potential. For example, LANIM has the flexibility to change the transport path based on the user's intent in the SINET architecture which ensures data security.

The programmable switch with P4 in L-CNC increases processing latency and controller communication time which limits the processing performance of the programmable switch BMV2. These will degrade system performance. Policy updates increase computational overhead. In the network security scenario, performance and security are contradictory. The intelligent controller makes the system's strategy more intelligent and LANIM balances performance and security in different scenarios. In the next section, we focus on the equilibrium problem in terms of performance.

B. SAFETY AND EFFICIENCY

The LANIM implements a dynamically changing encryption strategy that utilizes timestamps and original packet encapsulation formats. There are three main ways to update: manual update, periodic updates and controller updates. The update primarily affects the processing delay and transmission delay. Processing delays include timestamp-based operations, computation of machine learning models, and compilation of new strategies. The programmable switch BMV2 uses the

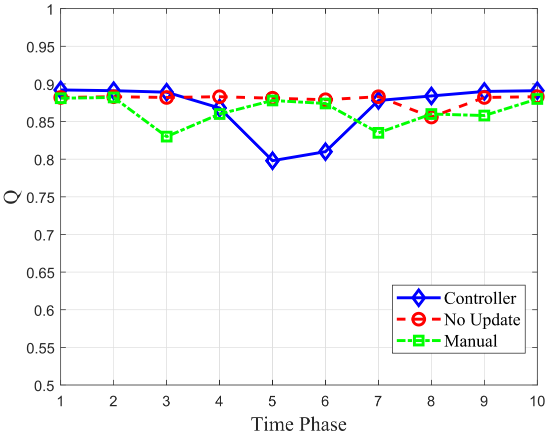


FIGURE 8. The impact of different update categories on the network (*Q*). Manually trigger updates in phases 2 and 6. The controller triggers the update in phase 3.

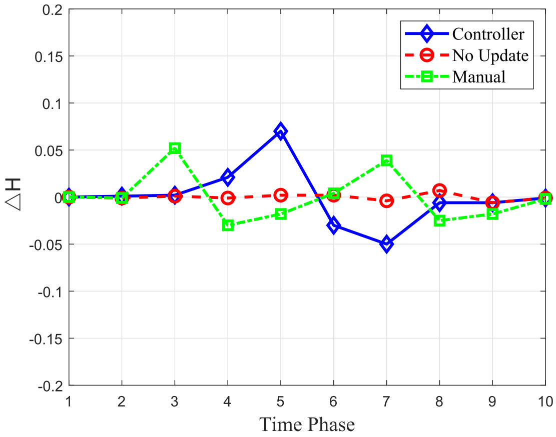


FIGURE 9. The impact of different update categories on the network (Δ*H*). Manually trigger updates in phases 2 and 6. The controller triggers the update in phase 3.

P4 language to extract the code of the new encryption policy from the register and compiles to overwrite the original policy [31]. The transmission delay includes the delay of the packets in and packets out when the smart controller initiates an update. In actual experiments, we found the communication between the switch and the controller is the primary factor affect performance.

We simulate the impact of system performance with the three update methods. Fig. 8 and Fig. 9 show the average of ten simulation results. Q is defined in equation (17) as a comprehensive description of network performance.

We perform manual updates in the second and six phases. Manual update is to initiate a policy update including packet parsing, timestamp operations, and encryption policy selection. Fig. 8 shows the update reduces the Q value. In the third and seventh phases, the Q value reached a minimum and the performance decreased by 9.3%.

The controller initiates the update in the third phase. The Q value decreases, and the Q value returns to a normal level after about 4 phases. The Q value is reduced by approximately 15%, and network performance returns to the normal level after four phases. Controller updates have a longer impact on the system than manual updates. As seen in fig. 8, our mechanism affects performance slightly, but destroys the kill chain of eavesdropping attacks and immunizes network hazards. According to the derivation in the previous section, it is almost impossible to brute force the entire data of a conversation between updates. The controller updates have the most significant impact and the most prolonged duration on the system. The packet in and packet out between switch and controller increases the transmission delay. Typically, It takes a long round trip time when the first packet requests the controller.

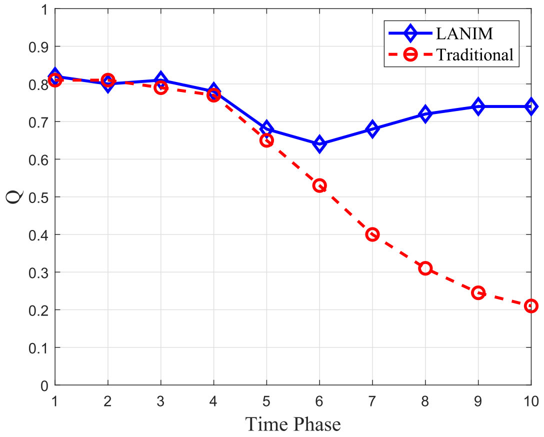


FIGURE 10. The performance (*Q*) of LANIM under network attack.

Fig. 9 shows the relationship between ΔH and policy transformation. According to equation (16), ΔH describes the extent to which the system is affected. The deviation of the curve from the X-axis is related to the performance of the system. When $\Delta H > 0$, the performance of the system decreases, and when $\Delta H < 0$, the system performance gradually improves. Fig. 9 illustrates that controller updates have a greater impact on the system than the switch initiates updates. However, a controller deploying LANIM can intelligently initiate a small number of updates and reduce the impact on the network when the controller perceives network reliability. Threats to network performance often occur suddenly. It brings much overhead that the system always deploys the highest-level defense mechanism against an accidental security incident. Frequent manual updates are more expensive than controller updates. Therefore, it seems unwise to increase the frequency of updates in exchange for security. It is difficult to adapt to multiple environments. This contradiction is resolved by dynamically sensing the network environment using a Bayesian model that minimizes risk.

We analyze the performance under cyber-attacks and the results are shown in Fig. 10 and Fig. 11. In the third phase, the DDoS attack tool (LOIC) sends packets to attack the switch continuously [32]. The controller adopts a response strategy for IP addresses. Fig. 10 shows the performance of both strategies when attacked. The degradation of traditional networks is significantly higher than that of LANIM systems. In the sixth to tenth stages, traditional network performance

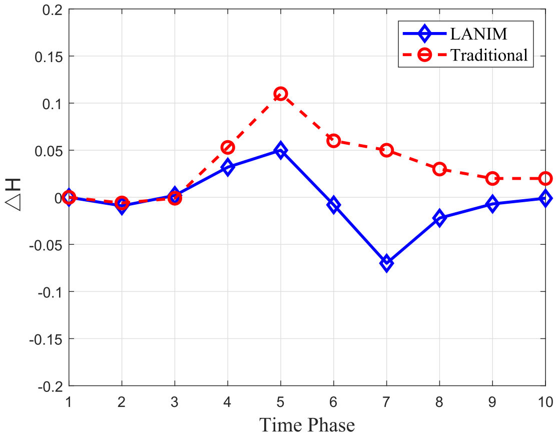


FIGURE 11. The performance (ΔH) of LANIM under network attack.

continues to decline. After the sixth phase, LANIM is less affected by the attack. It can be seen that the resistance of the intelligent control system to the network attack and the performance of the network is improved.

Fig. 11 shows the ΔH curve. After the third phase, the ΔH starts to increase. LANIM's curve starts to be less than zero in phase 6, indicating that network volatility decreases. In the eighth to tenth stages, the ΔH of both networks approaches zero. However, it is worth noting that the red line is close to the downtime in the eighth to tenth stages, while the blue line still maintains a high Q value. LANIM has more advantages in an insecure network environment. The SINET architecture makes the data forwarding more efficient due to the collaboration in functional groups, and the intelligent controller can quickly cope with different network scenarios. The stateful encryption mechanism effectively blocks the killing chain of the eavesdropping attack.

VII. CONCLUSION

This paper proposes the learning based adaptive network immune mechanism which can be widely applied to various networks, such as IP network and SINET. LANIM can flexibly adapt to various network scenarios by changing the risk function. The intelligent controller senses the network state and uses the real-time stateful encryption strategy to ensure the security of the information. We provide a systematic evaluation method based on network entropy. Finally, we verify the security and effectiveness of LANIM from an experimental and theoretical perspective. The results show that LANIM can enhance transport security and adapt to various security scenarios in SINET architecture. LANIM is valuable for operators to protect network systems from further damage. In future work, we will focus on the advanced machine learning algorithm and enhance the adaptability of LANIM.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their invaluable comments on this article. The authors also thank Y. Qin and J. Shi for their contributions to the experimental platform.

REFERENCES

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," *Commun. ACM*, vol. 55, no. 1, pp. 117–124, Jan. 2012.

[2] G. Liu, W. Quan, N. Cheng, K. Wang, and H. Zhang, "Accuracy or delay? A game in detecting interest flooding attacks," *Internet Technol. Lett.*, vol. 1, no. 2, p. e31, Mar./Apr. 2018.

[3] H. Zhang, W. Quan, H.-C. Chao, and C. Qiao, "Smart identifier network: A collaborative architecture for the future Internet," *IEEE Netw.*, vol. 30, no. 3, pp. 46–51, May/Jun. 2016.

[4] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, *The Locator/Id Separation Protocol (lisp)*, document RFC 6830, Cisco Systerm, San Jose, CA, USA, 2013.

[5] I. Seskar, K. Nagaraja, S. Nelson, and D. Raychaudhuri, "Mobilityfirst future Internet architecture project," in *Proc. 7th Asian Internet Eng. Conf.*, Nov. 2011, pp. 1–3.

[6] J. Liang, J. Jiang, H. Duan, K. Li, T. Wan, and J. Wu, "When HTTPS meets CDN: A case of authentication in delegated service," in *Proc. IEEE Symp. Secur. Privacy*, May 2014, pp. 67–82.

[7] W. Quan, N. Cheng, M. Qin, H. Zhang, H. A. Chan, and X. Shen, "Adaptive transmission control for software defined vehicular networks," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 653–656, Jun. 2019.

[8] W. Quan, Y. Liu, H. Zhang, and S. Yu, "Enhancing crowd collaborations for software defined vehicular networks," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 80–86, Aug. 2017.

[9] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, and G. Varghese, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 87–95, Jul. 2014.

[10] H. Song, "Protocol-oblivious forwarding: Unleash the power of sdn through a future-proof forwarding plane," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw.*, Aug. 2013, pp. 127–132.

[11] C. Sun, J. Bi, H. Chen, H. Hu, Z. Zheng, S. Zhu, and C. Wu, "SDPA: Toward a stateful data plane in software-defined networking," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3294–3308, Dec. 2017.

[12] M. Bellare, T. Kohno, and V. Shoup, "Stateful public-key cryptosystems: How to encrypt with one 160-bit exponentiation," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, Nov. 2006, pp. 380–389.

[13] P. Yang, R. Zhang, and K. Matsuura, "Stateful public key encryption: How to remove gap assumptions and maintaining tight reductions," in *Proc. Int. Symp. Inf. Theory Appl.*, Dec. 2009, pp. 1–6.

[14] T. T. Mapoka, S. J. Shepherd, and R. A. Abd-Alhameed, "A New Multiple Service Key Management Scheme for Secure Wireless Mobile Multicast," *IEEE Trans. Mobile Comput.*, vol. 14, no. 8, pp. 1545–1559, Aug. 2015.

[15] Z. Zhou and D. Huang, "An optimal key distribution scheme for secure multicast group communication," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–5.

[16] T.-K. Wang and F.-R. Chang, "Network time protocol based time-varying encryption system for smart grid meter," in *Proc. 9th IEEE Int. Symp. Parallel Distrib. Process. Appl. Workshops*, May 2011, pp. 99–104.

[17] F. Rodríguez-López, F. Girela-López, and J. Díaz, "A hardware assisted implementation of time varying encryption system," in *Proc. IEEE Int. Symp. Precis. Clock Synchronization Meas., Control, Commun. (ISPCS)*, Sep. 2018, pp. 1–6.

[18] H. Li and H. Zhang, "A survey on smart collaborative identifier networks," *China Commun.*, vol. 15, no. 3, pp. 168–185, Mar. 2018.

[19] E. Al-Shaer, "Toward network configuration randomization for moving target defense," in *Moving Target Defense*. New York, NY, USA: Springer, 2011, pp. 153–159.

[20] N. Asokan, V. Niemi, and K. Nyberg, "Man-in-the-middle in tunnelled authentication protocols," in *International Workshop on Security Protocols*. Berlin, Germany: Springer, 2003, pp. 28–41.

[21] H. C. Mandhare and S. R. Idate, "A comparative study of cluster based outlier detection, distance based outlier detection and density based outlier detection techniques," in *Proc. Int. Conf. Intell. Comput. Control Syst. (ICICCS)*, Jun. 2017, pp. 931–935.

[22] N. Soonthornphisaj, T. Sira-Aksorn, and P. Suksankawanich, "Social media comment management using smote and random forest algorithms," in *Proc. 19th IEEE/ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput. (SNPD)*, Jun. 2018, pp. 129–134.

[23] M. Min, X. Wan, L. Xiao, Y. Chen, M. Xia, D. Wu, and H. Dai, "Learning-based privacy-aware offloading for healthcare IoT with energy harvesting," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4307–4316, Jun. 2019.

[24] S. Otoum, B. Kantarci, and H. T. Mouftah, "On the feasibility of deep learning in sensor network intrusion detection," *IEEE Netw. Lett.*, vol. 1, no. 2, pp. 68–71, Jun. 2019.

[25] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.

[26] L. Liu, J. Zhou, X. Guo, and R. Qi, "A method for calculating link weight dynamically by entropy of information in SDN," in *Proc. IEEE 22nd Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2018, pp. 535–540.

[27] G. Liu, W. Quan, N. Cheng, H. Zhang, and S. Yu, "Efficient DDoS attacks mitigation for Stateful forwarding in Internet of Things," *J. Netw. Comput. Appl.*, vol. 130, pp. 1–13, Mar. 2019.

[28] G. Liu, W. Quan, N. Cheng, B. Feng, H. Zhang, and X. S. Shen, "BLAM: Lightweight Bloom-filter based DDoS mitigation for information-centric IoT," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[29] M. S. Pervez and D. M. Farid, "Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs," in *Proc. IEEE 8th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA)*, Dec. 2014, pp. 1–6.

[30] F. Gumus, C. O. Sakar, Z. Erdem, and O. Kursun, "Online naive Bayes classification for network intrusion detection," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2014, pp. 670–674.

[31] C. Zhang, J. Bi, Y. Zhou, and J. Wu, "HyperVDP: High-performance virtualization of the programmable data plane," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 556–569, Mar. 2019.

[32] A. Zimba, Z. Wang, and H. Chen, "Bayesian-Poisson based modeling of cyber attacks in cloud computing networks," in *Proc. IEEE 2nd Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, Dec. 2017, pp. 316–320.

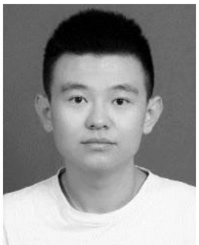
MINGYUAN LIU received the bachelor's degree in communication engineering, in 2018. He is currently pursuing the Ph.D. degree with the School of Electronic and Information Engineering, Beijing Jiaotong University (BJTU), Beijing, China. He has participated in national research programs of China. His current research interests include the future networks, software defined networking (SDN), and cybersecurity.

DEYUN GAO received the B.Eng. and M.Eng. degrees in electrical engineering and the Ph.D. degree in computer science from Tianjin University, China, in 1994, 1999, and 2002, respectively. He spent one year as a Research Associate with the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology. He then spent three years as a Research Fellow with the School of Computer Engineering, Nanyang Technological University, Singapore. In 2007, he joined the faculty of Beijing Jiaotong University as an Associate Professor with the School of Electronics and Information Engineering and was promoted to a Full Professor, in 2012. In 2014, he was a Visiting Scholar with the University of California at Berkeley, USA. His research interests include the Internet of Things, vehicular networks, and the next-generation Internet.

GANG LIU was born in Wuhu, Anhui, China, in March 1993. He is currently pursuing the Ph.D. degree with the National Engineering Laboratory for Next Generation Internet Technologies (NGIT), Beijing Jiaotong University (BJTU), Beijing, China. His current research interests include information centric networking (ICN), software defined networking (SDN), and network function virtualisation (NFV). He is a Student Member of ACM.

JINGCHAO HE is currently pursuing the bachelor's degree with the School of Telecommunications Engineering, Xidian University. His primary research interests include wireless networks and machine learning, with a focus on satellite IP networks and space-air-ground integrated networks.

LU JIN received the B.E. degree in communication and information systems from Beijing Jiaotong University, Beijing, China, in 2018, where she is currently pursuing the M.D. degree with the School of Electronic and Information Engineering. She has participated in several national research programs of China, such as Intelligent Vehicle Networking Program and 973 Program. Her research interests include the future Internet, multipath TCP, LTE-V2X, and machine learning.



CHUNLIANG ZHOU received the B.E. degree in communication and information systems from Beijing Jiaotong University, Beijing, China, in 2018, where he is currently pursuing the M.D. degree with the School of Electronic and Information Engineering. He has participated in several national research programs of China. His research interests include machine learning and cloud computing.

FUCONG YANG received the B.E. degree in communication and information systems from Beijing Jiaotong University, Beijing, China, in 2017, where she is currently pursuing the M.D. degree with the School of Electronic and Information Engineering. She has participated in several national research programs of China, such as 973 Program. Her research interests include the future Internet, In-band network telemetry, and machine learning.