IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 15, 2020, accepted September 25, 2020, date of publication October 5, 2020, date of current version October 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3028603

IPsec Cryptographic Algorithm Invocation Considering Performance and Security for SDN Southbound Interface Communication

XIMIN YANG[1], **DEQIANG WANG**[2], **WAN TANG**[1], **(Member, IEEE),**
WEI FENG[1], **AND CUITAO ZHU**[3]

[1]College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China
[2]Sangfor Technologies Inc., Changsha 410205, China
[3]College of Electronics and Information Engineering, South-Central University for Nationalities, Wuhan 430074, China

Corresponding author: Wan Tang (tangwan@scuec.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61671483 and Grant 61772562, and in part by the China Education and Research Network (CERNT) Innovation Project under Grant NGII20150106.

ABSTRACT The introduction of IPsec into software-defined networking (SDN) can secure communication in an SDN southbound interface, i.e., communication between the controllers and the switches. However, due to the static configuration of IPsec cryptographic algorithms, the invocation of these algorithms cannot dynamically self-adapt to traffic fluctuations in SDN southbound communication. To address the contradiction between link security and communication performance incurred by IPsec encryption, an evaluation model to find a trade-off between communication performance and link security is presented in this paper. An invocation mechanism based on the Free-to-Add (FTA) method is also proposed to optimize the invocation mode of cryptographic algorithms in traditional IPsec. Based on the real-time network status and the impact of the IPsec encryption process on the network latency and throughput, a feedback-based scheduling scheme is designed to enable the IPsec algorithms in use to be flexibly replaced and synchronously switched, and two policies are applied to determinate the appropriate encryption algorithm(s). The validity and effectiveness of the FTA-based mechanism are verified and evaluated on an SDN/OpenFlow platform in which IPsec security gateways are deployed. The feedback-based scheduling scheme is evaluated in terms of packet processing latency, distribution of optional encryption intensity, and the hit rate of encryption intensity.

INDEX TERMS Algorithm invocation, communication security, IPsec, software-defined networking (SDN), southbound interface (SBI).

I. INTRODUCTION

The software-defined networking (SDN) paradigm decouples the control plane from the underlying data plane [1]. It introduces network programmability and other features to promote network flexibility by adapting to constantly changing network conditions and facilitating the verification and deployment of the network. Although some new features have been introduced into the SDN to provide more convenience for network management [2], [3], several new types of security threats have emerged due to a lack of consideration of security issues in the original design of the SDN architecture.

The OpenFlow protocol is a widely adopted communication standard for the southbound interface (SBI) in SDN networks, and it is critical to establish a isecure SBI communication. As the control plane communicates with the data plane using OpenFlow-supported instructions, the feature of separation between these two planes means that the control flows are insecure when passing through exterior network links. OpenFlow therefore cooperates with the transport layer security (TLS) protocol to secure the communication between the SDN controller(s) and the switches (i.e., SBI communication) [5]. Nevertheless, the TLS protocol is too complicated to verify and is insufficiently robust against man-in-the-middle (MITM) attacks to guarantee security, and thus has become an optional support that is unnecessary for OpenFlow [6]. Without the security protection of TLS, TCP-based SBI communication is vulnerable to tapping and forgery of control information, which makes the network insecure and unreliable.

The associate editor coordinating the review of this manuscript and approving it for publication was Shaohua Wan.

Therefore, Internet Protocol security (IPsec) is introduced in this paper to guarantee security in the SBI of the controller and maintain secure communication between the controller and the switches in the SDN network. IPsec has been widely adopted to protect communications between entities at IP layer, such as the host-to-host, gateway-to-gateway, and host-to-gateway communications [7]. That is, IPsec, which was originally developed for IPv6, can also ensure communication security in the IP layer and does not require extra support from the controller.

Much research has focused on reducing the complexity and improving the efficiency of IPsec cryptographic algorithms [8]. However, the rigid invocation of cryptographic algorithms in IPsec makes it inflexible and less able to meet the security demands of current networks, which are becoming more complicated than before. The system should be able to guarantee highly secure and reliable communications [9]. Moreover, the IPsec cryptographic algorithms have ever-increasing effects on communication performance, and the security demands are highly diversified due to the varied nature of the characteristics of the network traffic. It is imperative to achieve the best trade-off between the performance and security of software-defined networks [10]. All these issues require further exploration.

A flexible invocation mechanism for the IPsec encryption algorithms will be studied in this paper. Compared with the native IPsec algorithm invocation in SBI communication in SDN networks, we provide the following contributions:

- We present an evaluation model to find a trade-off between the communication performance and the link security.
- We design an FTA-based invocation mechanism with feedback-based scheduling, based on both the real-time network status and the impact of the IPsec cryptographic process on network performance.
- We propose two policies for feedback-based scheduling for the selection of the appropriate encryption algorithm(s).

The rest of this paper is organized as follows. Section II discusses the issues to be addressed in this paper. Section III presents an evaluation model for analyzing the balance between network performance and link security. Section IV describes the FTA-based invocation mechanism of IPsec cryptographic algorithms and the feedback-based scheduling. Section V demonstrates the proposed scheme via experiments and compares it with the original IPsec scheme. Two proposed determination policies are also evaluated in three modes. Finally, the conclusions of this paper are drawn in Section VI.

II. BACKGROUND AND MOTIVATION

A. COMMUNICATION SECURITY IN THE SOUTHBOUND INTERFACE

Prior work related to SDN is mostly focused on deployment scenarios, and little research has addressed the security issues of this architecture [2], [11]. SDN controllers now encounter many security threats, and the main countermeasures are to design and implement new secure SDN controllers or to develop combinable security module libraries for controllers [12]. The first SDN controllers to be developed were mainly responsible for controlling and scheduling network resources, and insufficient attention was paid to security. Even worse, the transmission of SDN control messages through the SBI was implemented using the network configuration protocol (NETCONF), which is protected only by TLS and lacks practical encryption and integrity checks [10]. With more and more varied potential security threats surfacing, the further development of SDN will be severely impeded.

The security of SDN controllers has become one of the core tasks for current research in this area, and several schemes have been designed to enhance SBI security [4], [13], [14]. These new controllers achieve better and more comprehensive security than general SDN controllers and reduce the risk from the SBI [15]. However, some risks still cannot be eliminated by secure controllers while exchanging control messages with switches, for instance MITM attacks, which exploit the flaw in the TLS protocol, and the risks of tapping and forging control messages when using TCP connections [16], [17].

In typical deployment scenarios involving SDN, these security problems tend to be handed over to links that are external to the controller, meaning that these security issues are addressed by the network itself [18]. The most common attacks taking place in the SBI include distributed denial of service (DDoS) and source address spoofing. Many studies have been undertaken relating to defending against DDoS attacks [19], [20], and these typically protect the network using connection migration. Based on the counter information in the controller, AVANT-GUARD [20] removes all unmatched packets except SYN/ACK packets to eliminate threats to SBI communication by adding connection migration and an incentive trigger module. Defending against source address spoofing attacks is based on source address authentication by using security protocols, e.g., IPsec and TLS.

When it comes to network security protocols, data transmission is more deployable and efficient in IPsec than in TLS. In addition, IPsec encrypts the data at the network layer, which is a more comprehensive and secure method of communication between controllers and switches without involving other application traffic [21], [22]. A security controller based on SDN can provide IPsec-based flow protection in two main scenarios: gateway-to-gateway and host-to-host. The research presented in [23] focuses on the lack of a mechanism for dynamic key distribution to network security functions (NSFs) whose job is to protect data traffic between network resources by implementing IPsec.

In summary, as the standard security protocol suite of IPv6, IPsec can guarantee security of the SBI communication for the SDN controller and meet the network requirements for a network evolving towards IPv6.

B. TRADE-OFF BETWEEN IPSEC ENCRYPTION AND COMMUNICATION PERFORMANCE

It is also imperative to evaluate the impact of the link security provided by IPsec on the performance required of SBI communication in SDN networks. An increase of encryption strength will result in higher costs and lower performance [24]. So we should try to decrease the cost to maintain the performance, with the promise that system security is guaranteed.

When deploying IPsec for secure communication, the packing/unpacking of security headers and encryption/decryption payload are included in the packet processing; these require more time and more system resources, which ultimately reduces the overall communication performance. Network delay and throughput are two key network performance metrics. They will be affected and change due to the fluctuation in the traffic and shifts in the distribution of the packet sizes. Although the impact is small and the network performs well when the CPU load of the network devices is low, the situation will change if IPsec is introduced. IPsec encryption processing will aggravate the network congestion under high traffic conditions, producing lower throughput and higher delay and traffic fluctuation. In addition to the overhead from the IPsec security headers, packet processing becomes more complex, which results in longer processing times and more energy consumption. As the overload of the security headers increases, so does the payload.

A further aim is to achieve the best trade-off between IPsec encryption and communication performance [10]. On a link secured with IPsec, variations in the traffic and the distribution of the packet sizes may have a noticeable impact on throughput and network delay. In addition, this impact varies with the processing performance of the encryption algorithms.

C. ISSUES RELATED TO IPSEC ALGORITHM INVOCATION

A flexible mechanism for the selection and invocation of an IPsec cryptographic algorithm that considers the real-time network status is urgently needed in order to achieve a good trade-off between IPsec encryption and communication performance [23]. Some issues related to the invocation of the IPsec algorithm will be discussed below.

The IPsec protocol is mature in terms of its architecture but rigid with regard to the invocation of its cryptographic algorithms. Taking the case of StrongSwan, an open source IPsec-based virtual private network (VPN) solution for Linux, as an example, an algorithm for encrypting data is specified in the configuration file, and the user must modify the configuration file in order to use another algorithm. The demand for customized algorithms and for more algorithms to be supported by IPsec is also urgent in various application scenarios, alongside the mounting importance of network security. As far as our knowledge extends, little research has focused on flexibility with regard to the subsequent invocation of IPsec algorithms.

When IPsec is adopted to provide security for SDN controller–switch communication, it is inconvenient for the user to add a customized algorithm to the switches, because the vendors limit the modifications of the switches, and the addition or upgrading of a device therefore results in high costs. Furthermore, IPsec encryption/decryption will decrease the performance, even though there are certain performance requirements for the SBI communication. When the traffic fluctuates, consumption forms a bottleneck problem of communication and amplifies the variation in traffic and communication performance. In view of this, it is necessary to either upgrade the network devices to reduce the impact, or try to strike a balance between the link security offered by IPsec and the communication performance of the SBI in SDN networks through an effective IPsec algorithm invocation. The determination of the strength of the algorithm needs to consider not only whether the algorithm can withstand current known attacks but also an invocation of the algorithm based on its encryption strength [25].

III. EVALUATION MODEL OF THE BALANCE BETWEEN IPSEC ENCRYPTION AND COMMUNICATION PERFORMANCE

In this section, we will analyze the impact of IPsec encryption on the communication performance and present an evaluation model for the balance between them.

A. CONTROLLER–SWITCH COMMUNICATION, ASSUMPTIONS AND NOTATION

The framework for communication between the controller and the switch, or SBI communication in the IPsec-deployed SDN scenario, is illustrated in Figure 1. The metrics used to evaluate the communication performance, namely, delay and throughput, only relate to the SBI communication, rather than that of the whole network. The network devices (also called nodes) include SDN controllers and OpenFlow switches.

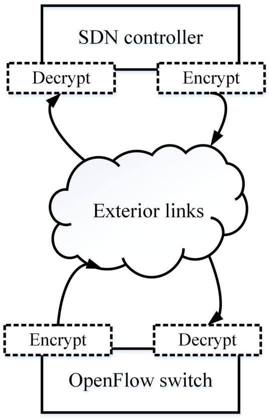


FIGURE 1. IPsec-deployed link communication between SDN controller and OpenFlow switch.

To simplify the analysis of the effect of the algorithms, we make three assumptions:

- *Assumption 1:* There are multiple algorithms with a similar security intensity and different system resource requirements, and the performance of these algorithms has been optimized.
- *Assumption 2:* The link bandwidth is not considered to be a system resource that is required for the IPsec encryption/decryption or communication.
- *Assumption 3:* The variation in network delay is only associated with the queuing latency and the processing latency of encryption/decryption, and other latencies remain unchanged.

Our study does not include concrete encryption algorithms, so it is assumed that the encryption strength of each algorithm is already known as given in Assumption 1.

The methods of defending against known attacks and the time of these attacks should be taken into account [25], [26] when considering the strength of the algorithm, and the selection of the algorithm is the job of the network administrator. Therefore, in this paper we just assume that the encryption strengths of the IPsec algorithms have been classified into different levels without considering how to select the algorithms.

For clarity, the notation used to describe the evaluation model, policies, and algorithms in the following sections are listed in Table 1.

TABLE 1. Notation used in modeling and analyzing.

Symbol	Description
E	level of link security
M	evaluation value of balance model
G	distribution of package size
Q	amount of transferred data
F	flow status
S	network status
T	throughput
D	network latency
D_t	propagation latency
D_p	processing latency of node
D_q	queue latency
D_e	encryption/decryption latency
D_f	forwarding latency
D_n	latency of the nth sampling
D_{max}	upper limit of normal latency
W	amount of transferred payload
W_i	amount of queued payload in the ith node
w_i	amount of payload of the ith package
w_s	weight of communication performance
w_e	weight of link security
P_i	processing performance of the ith node
p_j	processing performance of the jth encryption algorithm
k	number of packets
h_i	length of the header of the ith IPsec ESP
c_0	constant

B. IMPACT OF IPSEC ON NETWORK PERFORMANCE

Data encryption and communication involve significant network resource overheads, and the performance bottleneck in some network nodes can create network congestion and performance fluctuation.

1) LATENCY OF CONTROLLER–SWITCH COMMUNICATION

In the scenario of the communication between the controller and the switch shown in Figure 1, the latency D consists of three parts: the propagation latency D_t, the processing latency D_p, and the queue latency D_q, and is calculated as follows:

$$D = D_t + D_p + D_q, \tag{1}$$

where the propagation latency D_t is mainly related to the hardware environment of the network rather than the deployment of IPsec. We therefore ignore it when computing the network delay.

The processing latency D_p includes the forwarding latency D_f and the encryption/decryption latency D_e:

$$D_p = D_f + D_e. \tag{2}$$

The packets forwarded between the controller and the switches must get through the encryptiondecryption units, and the time spent in these units is the major contribution to the forwarding latency of these packets. Since it is generally stable, the forwarding latency D_f is regarded as a constant c_0. Unlike D_f, the values of the encryption/decryption latency D_e will have major variations when different cryptographic algorithms are applied. The encryption/decryption time for the IPsec encapsulating security payload (IPsec ESP) is relatively fixed, and we therefore neglect changes in this quantity and merge it into the constant c_0.

The queue latency D_q is determined by the forwarding performance P_i and the amount of queued data W_i at each node i, and is calculated by (3):

$$D_p = \sum_i \frac{W_i}{P_i}. \tag{3}$$

At a forwarding node, the processing performance has a certain value, meaning that the value of the queue latency D_q is positively correlated with the size of the queued payload. On the other hand, for a node executing encryption/decryption, the performance of the cryptographic algorithms may increase the processing time and thus affect the queue latency.

Given the above, we know that for the IPsec communication shown in Figure 1, the queue latency D_q and encryption/decryption latency D_e are the main factors influencing the latency D, and other delays with relatively fixed values can be set as constants. The delay D can therefore also be formulated as follows:

$$D = D_q + D_e + c_0. \tag{4}$$

The value of D will increase linearly with the queue latency D_q as the queue becomes congested due to the increasing traffic. However, D is also affected by the encryption/decryption process of IPsec. The queue latency D_q and encryption/decryption latency D_e vary with the cryptographic algorithm(s) used, the processing speed, the cryptographic capacity of the nodes, etc.

2) THROUGHPUT OF CONTROLLER–SWITCH COMMUNICATION

The deployment of IPsec also influences the network throughput. The packing/unpacking of IPsec ESPs and encryption/decryption result in greater resource consumption. Moreover, the link throughput will degrade, since the queue latency of the nodes implementing IPsec cryptographic processes is higher than that of the normal nodes due to more time to encrypt the packets. These nodes may become a performance bottleneck and cause network fluctuation.

The throughput T is given by

$$T = \frac{W}{D}, \tag{5}$$

where W and D denote the amount of payload transferred and network latency, respectively. When W is given and unchangeable, a higher latency D goes along with the lower throughput T. However, a more serious effect is that the throughput will dramatically reduce if the traffic starts to fluctuate significantly and the queue latency increases sharply.

The types and sizes of the control messages through the SDN/OpenFlow SBI vary, and W is therefore calculated as the sum of the package payload of each control message w_i transferred via SBI:

$$W = \sum_{i=1,k} w_i. \tag{6}$$

After encapsulating the IPsec ESP header h_i into the payload, the amount of data transferred between the controller and switches Q can be expressed as

$$Q = \sum_{i=1,k} (h_i + w_i). \tag{7}$$

According to (6) and (7), we have

$$Q = W + \sum_{i=1,k} h_i. \tag{8}$$

The throughput T is then computed by

$$T = \frac{Q}{D} = \frac{W + \sum_{i=1}^{k} h_i}{D_q + D_e + c_0}. \tag{9}$$

From (9), we can see that the throughput will change according to the distribution of the package sizes when W and the bandwidth are invariant. A greater number of small packages means a higher number of packets k, and the consequent header overhead may lead to lower throughput.

3) RELATION BETWEEN IPSEC SECURITY AND NETWORK PERFORMANCE

Based on the previous analysis, we know that if the IPsec encryption/decryption processes is executed quickly with few resources, the processing latency of encryption/decryption and the queuing latency of data forwarding in the link will decrease. Thus, we can obtain the following information about the relation between IPsec security and network performance:

- IPsec encryption/decryption may degrade communication performance. The stronger the encryption, the greater the resource consumption and the more processing time required, leading to a greater impact on the communication performance.
- When the data traffic fluctuates, the offset of the packet size distribution will result in a reduction in throughput and an increase in latency. The impact of the IPsec encryption/decryption on the communication performance can be reduced to below a guaranteed security level.

C. BALANCE EVALUATION MODEL

An evaluation model is presented in (10) to determine the exact feedback based on the degree of balance for scheduling algorithms and used in our IPsec invocation mechanism:

$$M = \frac{w_e}{w_s}, \tag{10}$$

where M, w_e and w_s denote the degree of balance, the weight of link security, and the weight of communication performance, respectively. For instance, if the network performance is the highest priority factor in a certain scenario, a higher-level encryption algorithm is needed in IPsec when the value of M varies over the effective range.

The weight of link security w_e is adjusted according to the network security status, and its value will increase when the risk of network security increases. With increasing security level of the IPsec encryption algorithm, the security risk keeps declining, and the security weight w_e is also reduced to the initial value of zero. The weight of communication performance w_s is related to the throughput, latency and the possibility of security risks. When an attack or security risk is detected, then security becomes the primary goal, and w_s remains the same.

Assumption 3 indicates that the network delay depends only on the queuing latency and encryption/decryption latency. If the latency exceeds the maximum latency of the normal link, this means that the queue congestion is very heavy. In this case, it is necessary to increase the value of w_s and to switch to an encryption algorithm that can provide better performance in order to guarantee communication performance. The weight of the communication performance is calculated using (11):

$$w_s = \begin{cases} max(1, \frac{D_n}{D_{max}}), & if \quad w_e = 1 \\ 1, & if \quad w_e \neq 1 \end{cases}, \tag{11}$$

where D_n denotes the latency obtained from the nth sampling and D_{max} is the maximum latency of the link status in normal operation. Since changing from one encryption algorithm to another does not vary continually the performance of the encryption, yet the latency of the sampled network may be continuously changing, we set the algorithm switch criterion according to (12), where p_j is the processing performance of the jth encryption algorithm in the algorithm set A. Note that the switch criterion is a range of values rather than an exact value, and the security level of the candidate algorithms for scheduling should meet the security demand.

$$\begin{cases} if\ M \geq \frac{p_j}{p_{j+1}}, & \text{switch to an algorithm with a higher security level, until reaching the inital algorithm.} \\ if\ M \leq \frac{p_j}{p_{j-1}}, & \text{switch to an algorithm providing higher process performance, until reaching the algorithm with the lowest security level.} \end{cases} \tag{12}$$

IV. INVOCATION MECHANISM FOR IPSEC CRYPTOGRAPHIC ALGORITHMS

A. IPSEC IN SDN ARCHITECTURE

Due to the separation of the control and data planes, SDN controllers and switches are in different network locations. Controllers are usually high-performance hosts or servers, meaning that the deployment of IPsec is straightforward and convenient. In most OpenFlow switches (e.g., Juniper EX4550), vendors tend to limit modifications. Thus, local implementation of certain customized demands of users becomes difficult, such as that of specific security demands.

In this case, a computer card or development board (e.g., the Raspberry Pi) can be added to the OpenFlow switches to build an IPsec secure gateway, as shown in Figure 2. The open architecture of IPsec facilitates the addition of new or customized cryptographic algorithms, and it is helpful in constructing a communication system with stronger closure and higher-level security. Moreover, IPsec can guarantee secure communication between controllers and OpenFlow switches via the IPsec security gateways. The added computer card or development board will enable optional functions and easily managed operation without exerting a negative impact on the configurations of the OpenFlow switches, system operation, data forwarding, and so on.

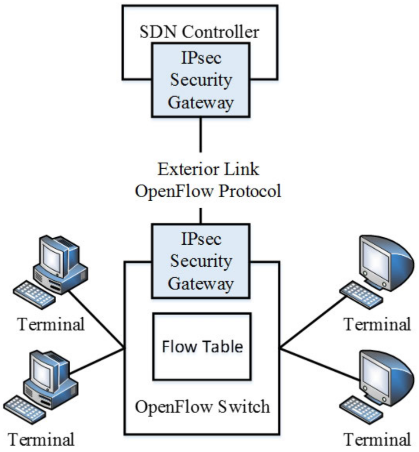


FIGURE 2. IPsec deployment scheme in SDN.

B. FREE-TO-ADD INVOCATION OF CRYPTOGRAPHIC ALGORITHMS

According to the workflow of StrongSwan 5.4.0, we know that the process of IPsec encryption communication can be divided into two parts: Internet key exchange (IKE), and encryption in session (ES), which is subsequently carried out in the kernel. Retaining the basic IPsec workflow, an invocation mechanism (shown in Figure 3) called FTA was proposed in our previous paper [27] to provide flexible addition and invocation of a cryptographic algorithm for IPsec in SDN networks. Two software-defined interfaces, an algorithm-control interface, and a subalgorithm interface were designed for use in FTA.

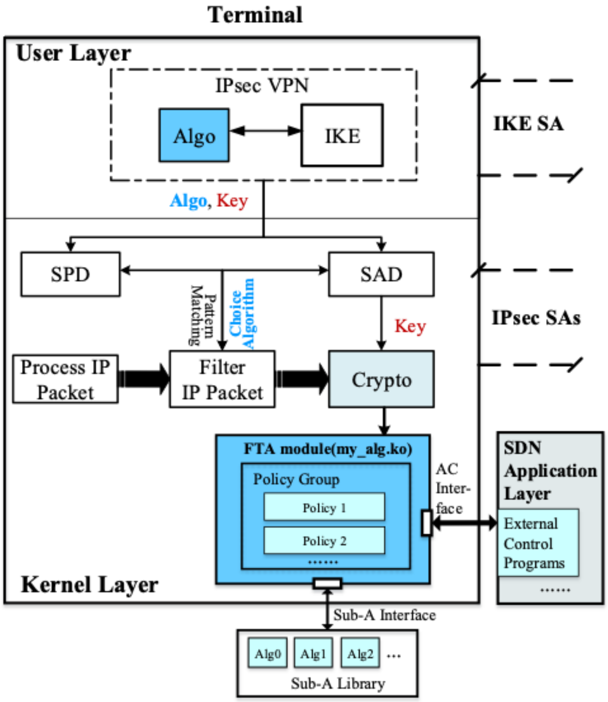


FIGURE 3. Invocation process of encryption algorithms in FTA-based IPsec (SA: security association; SPD: security policy database; SAD: SA database; AC: algorithm control; Sub-A: subalgorithm).

The FTA-based invocation is to some extent based on the concept of software-defined everything, and the usage of IPsec is made scalable and flexible with the two opening interfaces. The FTA mechanism does not directly invoke the specific encryption/decryption module *Algo.ko*, but invokes an FTA implementation module, called *my_alg.ko*.

In an IPsec-deployed link communication, the terminals are divided into two categories: the controlling terminal and the receiving terminal (i.e., server and client). An IPsec-encrypted tunnel based on StrongSwan 5.4.0 is established between the server and client terminals, to which new algorithms have been successfully added. The security of the TCP sessions between the server and client is ensured due to the existing encrypted tunnel, and these terminals communicate with the FTA module via *netlink* to ensure synchronous control of algorithms [27].

The implementation of an FTA-based invocation mechanism consists of an external control unit, and controlling-receiving, netlink and algorithm-selection units.

- External control unit
  The main task of the external control unit is to invoke and combine the specific cryptographic algorithms through the algorithm-control interface. Flexible combinations of algorithms (i.e., combined policies) and multiple encryptions are supported via the subalgorithm and algorithm–controller interfaces.
- Controlling-receiving unit
  This unit is responsible for synchronizing the policies in both terminals of communication. The controlling terminal, known as the server, first specifies a policy and initiates policy synchronization. The other end (namely, the client) then receives the policy issued by the server and synchronically maintains consistency with the server.
- Netlink unit
  The task of the netlink unit is to deliver the received IPsec policy to the specified location in the kernel. After confirming that the received policy is synchronized, both parties in communication (i.e., server and client) will distribute the policy to the corresponding kernel layers. Having received the message from the controlling-receiving unit and sent it to the algorithm module of the kernel for verification, the netlink unit will wait for the next message if the policy is validated; otherwise, it will repeatedly send the policy to the kernel until verification is obtained.
- Algorithm-selection unit
  In the algorithm-selection unit, a specific cryptographic subalgorithm is selected and executed according to the policy delivered by the netlink unit. Here, we refer to the algorithm management module as *my_alg.ko* (as shown in Figure 3); this handles the functions of initialization, unloading, login, logout, etc. The selected cryptographic algorithms can be implemented in various kernel modules or by using a unified subalgorithm library. The algorithms are selected by following the steps in Algorithm 1, while the IPsec stack of the kernel encrypts or decrypts the ESP payload.

The function of the cryptographic algorithm is to serve as the subalgorithm interface. Based on the policy forwarded by the netlink unit, the algorithm-selection unit finds the appropriate subalgorithms and performs cryptographic operations.

C. FEEDBACK-BASED ALGORITHM SCHEDULING

In this section, we present a feedback-based adjustment model that takes into consideration the trade-off between communication performance and link security, and we examine how to schedule, select, and adjust suitable IPsec cryptographic algorithms based on the model.

Algorithm 1 Algorithm Selection

```
1  Load my_alg.ko and allow algorithms to login to the kernel;
2  Wait for IPsec invocation;
3  Input the data to be encrypted or decrypted;
4  Read the policy forwarded by the netlink unit, and select the specific algorithm(s) according to the policy;
5  Output the encrypted or decrypted data;
6  if my_alg.ko is not unloaded then
7      goto Step 2;
8  end
9  Allow algorithm(s) to logout, and unload my_alg.ko;
10 return;
```

1) FEEDBACK-BASED ADJUSTMENT MODEL

This feedback model is implemented by the control program supported in the SDN application layer. The status S of the IPsec-secured links is defined by a triplet (T, D, E), where T, D and E denote throughput, latency, and security level of the link, respectively. Each cryptographic algorithm or policy corresponds to a certain level of security, and the level of link security E limits the other two factors T and D when the traffic load and the distribution of the packet size do not vary. Hence, the security level of link E should be adjusted according to the current values of T and D and the security demand.

We have designed a feedback-based adjustment model for the IPsec security algorithms, as shown in Figure 4. Having set the level of security, all the algorithms form an algorithm set A $(a_1, a_2, \ldots, a_n)$, $n \geqslant 2$, in which a higher index of an algorithm means that it can provide greater security strength. In the encryption process, a cryptographic algorithm can be replaced by another algorithm with a higher security level when the demand for security increases.

A median algorithm $a_j (j = [n/2])$ is selected as the initial algorithm for SBI communication. The algorithm to be used will be switched according to the feedback from the later communication performance evaluation. The simple adjustment model maintains a balance between the encryption security and communication performance, in which the key is to select a suitable encryption algorithm to be used based on the demands for both security and performance.

The invocation of cryptographic algorithms can be performed using a multiplexer. In this approach, the data packets are processed using different ciphers according to their encryption intensities, and these ciphers work in parallel. The multiplexer can be implemented either in hardware or software.

2) SCHEDULING IPSEC CRYPTOGRAPHIC ALGORITHMS

In the adjustment model presented in Figure 4, the feedback information indicates which algorithm should be chosen for the cryptographic process. An appropriate encryption

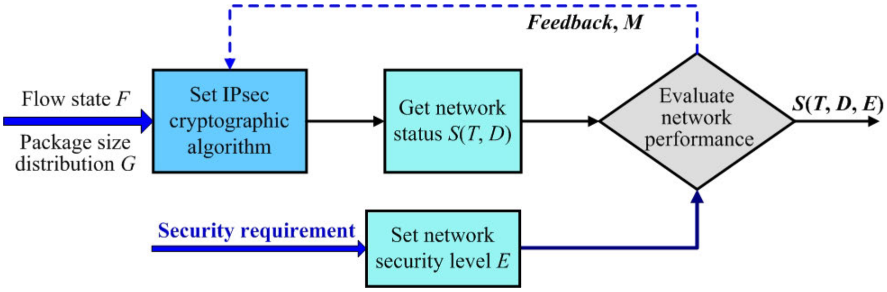

Feedback, M
Flow state F
Package size distribution G
Set IPsec cryptographic algorithm
Get network status S(T, D)
Evaluate network performance
S(T, D, E)
Security requirement
Set network security level E

FIGURE 4. Adjustment model based on feedback information for scheduling IPsec cryptographic algorithms.

algorithm is one that can meet the demand for communication performance and link security in a balanced way.

In the invocation mechanism, we schedule the IPsec encryption algorithms based on the encryption feedback generated after using the evaluation model, and switch the algorithms cooperating with the dynamic weight transformation achieved by using Algorithm 2. In this way, while the link security and communication performance change, the IPsec encryption algorithm being used for SBI communication can be replaced by another more appropriate algorithm to maintain the balance between the two factors. The description of the scheduling process is given in Algorithm 3 below.

```
Algorithm 2 Dynamic Weight Calculation
    Input: Performance status of IPsec-secured links
           $S(T, D, E)$
    Output: Weight of communication performance $w_e$,
            weight of link security $w_s$
    begin
        while get link status S do
            if some attack is detected then
                determine current security level $E$ and
                update $S$;
                set $w_e$;
                continue;
            else
                collect current link latency according to a
                given frequency;
                $w_s = D_n/D_{max}$;
            end
        end
    end
```

3) CRYPTOGRAPHIC ALGORITHM DETERMINATION AND ADJUSTMENT

In this section, we focus on how to determine the encryption intensity based on the requirements of network security and performance. Here, encryption intensity, or level of encryption, is an alternate name for the security strength discussed in Section IV-C1; this is the primary indicator used to select an appropriate encryption algorithm(s) and determine the mode of adjustment of the encryption algorithms in the feedback system.

```
Algorithm 3 IPsec Encryption Algorithm Scheduling
    Input: Flow status $F$, distribution of package size $G$,
           level of network security $E$
    Output: Appropriate algorithm $a_j$
    begin
        Get the performance of algorithm set of Level $E$:
        $P(p_1, p_2, \ldots, p_n)$;
        $j = n/2$;
        Get current communication performance
        $S(T, D, E)$ and network security status;
        Get dynamic weights $w_e$ and $w_s$;
        Calculate $M = (w_e/w_s)$;
        if $M \leq (p_j/p_{j-1})$ then
            Choose a high performance processing
            algorithm: $a_j = a_j - 1$;
        else
            if $M \geq (p_j/p_{j+1})$ then
                Choose $a_j = a_j + 1$;
            end
        end
    end
```

Before describing the policies for determining the encryption intensity, we give the definition of encryption intensity:

Definition 1: Encryption intensity is a metric used to measure the security capability required by the network. It can be classified into several integer grades according to the encryptgraphic strengths of the algorithms, where a value of one represents the weakest encryption intensity, and the strongest encryption intensity E_{max} is preset according to the encryptgraphic requirement.

The system must at least meet the minimum security requirement when selecting the encryption algorithm [26], and the encryption strength of an algorithm is mainly decided by the key length, namely, the complexity of the algorithm [28], [29]. The highest security level of algorithms (i.e., E_{max}) is decided by following these two principles.

a: POLICIES FOR DETERMINING THE ENCRYPTION INTENSITY

We propose two policies to determine the encryption intensities in the FTA mechanism: an encryption precedence (EP) policy and a gradual adjustment (GA) policy.

EP policy: This gives precedence to encryption algorithms that provide more efficient encryption while guaranteeing the network performance requirement. Based on the packet processing latency, it selects an appropriate encryption intensity e_n between the minimum value of one and a pre-defined maximum threshold E_{max}; e_n is equal to the strongest of the encryption intensity values that allows the network to achieve the performance requirements.

Equation (13) shows how e_n is selected, where d is an integer variable used to measure the status of the packet processing latency and is determined according to (14). In the EP policy, the value of the previously used encryption intensity has no impact on the subsequent determination and serves only as a variable parameter for adaptation to different application scenarios.

$$e_n = max(E_{max} - d, 1). \tag{13}$$

In (14), d is the dependent variable, and r is the independent variable ($0 < r$) which is related to the packet processing latency D_p and the max network latency D_{max}. It is apparent that the more closely D_p approximates D_{max}, the greater the value of d. In this paper, we pre-set E_{max} to a value of 5. The security levels of the encryption algorithms given in Table 2 are classified into five categories (ranging from 1 to 4, and $E_{max} = 5$) with legacy being the lowest strength, followed by baseline, standard, high, and ultra. The value of d is determined by four thresholds that describe the range of r: r_1, r_2, r_3 and r_4.

$$d = f(r = \frac{D_p}{D_{max}}) = \begin{cases} 4, & if \quad 0 < r \leq r_4 \\ 3, & if \quad r_4 < r \leq r_3 \\ 2, & if \quad r_3 < r \leq r_2 \\ 1, & if \quad r_2 < r \leq r_1 \\ 0, & if \quad r_1 < r < 1 \end{cases}. \tag{14}$$

TABLE 2. Security level of encryption algorithm.

Level	Security strength	Algorithm
1	legacy	DES, MDS, RC4, SHA-1
2	baseline	3DES
3	standard	AES-128, SHA-256
4	high	AES-192, SHA-384
5	ultra	AES-256, SHA-512

GA Policy: The GA policy takes the encryption intensity of the previously used algorithm into consideration, and determines whether a stronger or weaker encryption intensity will be selected, according to the packet processing latency. The value of the selected encryption intensity must be no larger than the preset value of E_{max}.

Equations (15) and (16) show how to obtain the value of e_n. In (15), e_n and e_{n-1} denote the encryption intensity of the current algorithm to be selected and that of the previous algorithm, respectively.

$$e_n = max(min(e_{n-1} + \Delta d, E_{max}), 1). \tag{15}$$

$$\Delta d = f(r = \frac{D_p}{D_{max}}) = \begin{cases} -1, & if \quad 0 < r < r_2 \\ 0, & if \quad r_2 \leq r \leq r_1 \\ 1, & if \quad r_1 < r < 1 \end{cases}. \tag{16}$$

b: FEEDBACK MODE

In this paper, three feedback modes are defined, based on the processing latency, in order to achieve the real-time latency of packet processing which is required and critical for the two policies presented above.

***Mode** 1:* Determine the encryption intensity based on the processing latency of each arriving packet:

$$e_{n+1} = delay(packet_n). \tag{17}$$

***Mode** 2:* Adjust the encryption intensity based on the average processing latency of the k packets arrived before for each arriving packet.

$$e_{n+1} = \frac{1}{k} \sum_{i=n-k}^{k} delay(packet_i). \tag{18}$$

***Mode** 3:* Provide an encryption intensity based on the average processing latency of per k packets for each sequence of k packets. These packets can be regarded as a packet group, and the same encryption intensity can be used as that determined for the previous group.

$$e_{n+1} = \frac{1}{k} \sum_{i=n-n\%k-k}^{n-n\%k} delay(packet_i). \tag{19}$$

Unlike Mode 1, Mode 3 updates the encryption intensity once k packets have been processed using the previous encryption intensity (i.e., the last used encryption intensity), rather than constantly adjusting the encryption intensity as each packet arrives.

In addition, each cryptographic algorithm with the selected encryption intensity is only applicable to those packets that enter the cryptographic algorithm scheduler after it has been selected for encryption.

V. EXPERIMENT AND VERIFICATION

In this section, we will describe the experimental environment and parameter settings, and evaluate whether the proposed invocation mechanism can guarantee the security of SBI communication in SDN networks.

A. VALIDITY OF FTA-BASED INVOCATION MECHANISM

A small-scale testbed was built to verify the validity of the FTA-based mechanism proposed in this paper. The topology is shown in Figure 5, and the configuration information is given in Table 3. The communication between the SDN controller and the OpenFlow switches in the testbed was SBI communication; two Raspberry Pis acted as IPsec gateways, using IPsec to secure the OpenFlow-based SBI communication. The FTA module was added to IPsec as a new algorithm, using the traditional method in IPsec [27]. New cryptographic

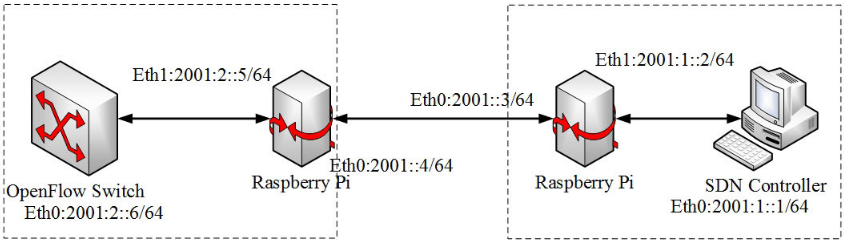
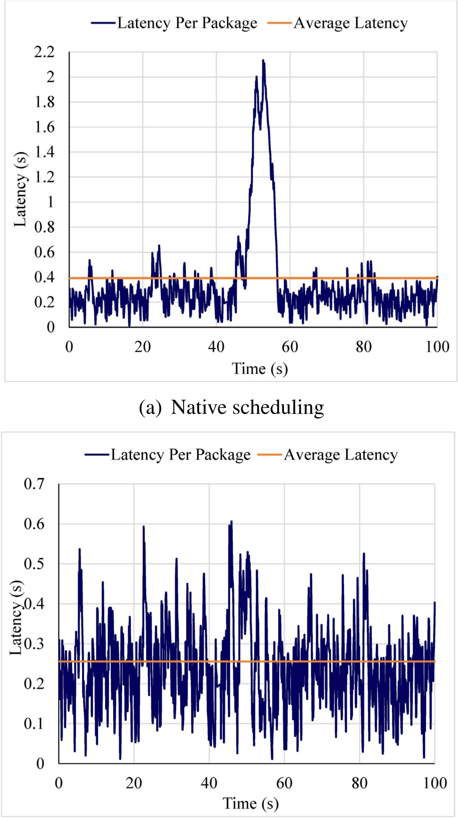


FIGURE 5. Topology of the SDN testbed with IPsec gateway.

TABLE 3. Simulation software and hardware.

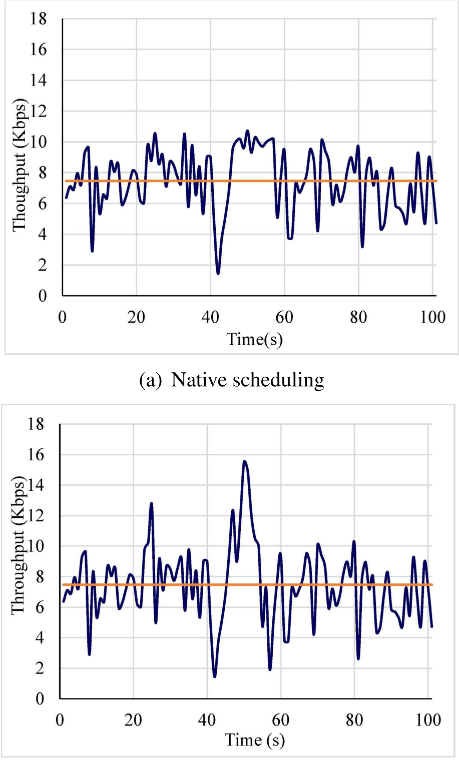
SDN device	OpenFlow switch	IPsec security gateway	SDN Controller
Equipment	Raspberry Pi 3B	Raspberry Pi 3B	DELL Inspiron 14 7000
Operating system	Raspbian Stretch Lite	Raspbian Stretch Lite	Ubuntu 16.04.3
Software	OpenvSwitch 2.3.0	StrongSwan 5.4.0	OpenDaylight Beryllium SR4
Hardware	ARM Cortex-A53 1.2GHz, Quad Core, 1G RAM, USB 2.0		i5-4200 2.8GHz, 8G RAM, USB 2.0



(a) Native scheduling

(b) Feedback-based scheduling

FIGURE 6. Variations in latency using different scheduling schemes for IPsec cryptographic algorithms.



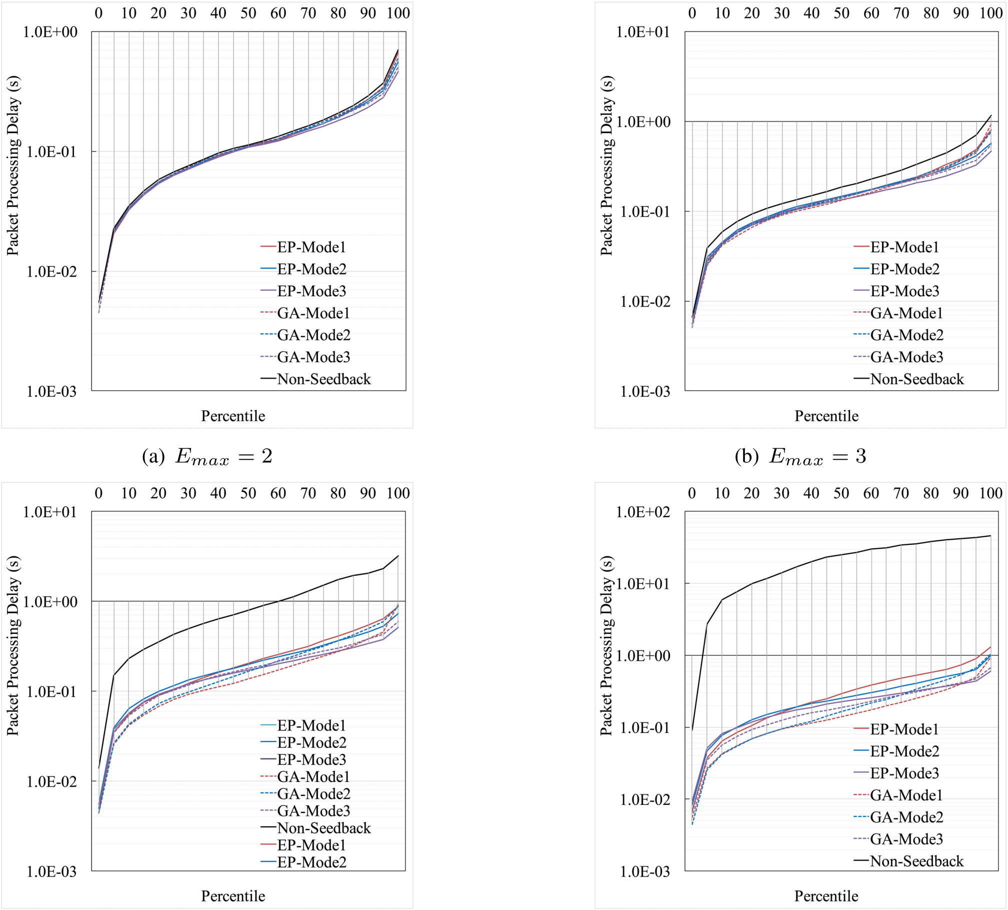
(a) Native scheduling

(b) Feedback-based scheduling

FIGURE 7. Variations in link throughput using different scheduling schemes for IPsec cryptographic algorithms.

algorithms and the control module were installed on terminals in advance.

Using the testbed (shown in Figure 5), the performances of the FTA-based and original IPsec approaches were compared using three cryptographic algorithms: *AES128*, *DES* and *3DES*. The relevant implementation modules of these algorithms were also inserted into the FTA module. In this experiment, *netperf* and *Iperf3* were used as the test tools. From the results presented in Table 4, it can be seen that, in these two cases, the network latencies and link throughputs are approximately the same under variations within the normal range. In addition, the actual bandwidths tested by Iperf3 in both mechanisms are almost the same. Hence, the use of the FTA-based mechanism has no obvious impact on the network performance.



(a) $E_{max} = 2$

(b) $E_{max} = 3$

(c) $E_{max} = 4$

(d) $E_{max} = 5$

FIGURE 8. Percentiles of packet processing latencies in statistics, applying the determination policies of encryption intensity (EP: encryption precedence policy; GA: gradual adjustment policy).

TABLE 4. Performance comparisons between native IPsec and FTA-based IPsec.

Item	Method	Cryptographic algorithm		
		AES128	DES	3DES
Latency (ms)	Native IPsec	2.63	2.62	2.62
	FTA-based IPsec	2.75	2.53	2.62
Requests/response (#/s)	Native IPsec	378	382	381
	FTA-based IPsec	363	394	381
Bandwidth (Mbps)	Native IPsec	64.1	55.7	34.8
	FTA-based IPsec	63.6	53	34.7

TABLE 5. Parameter setting.

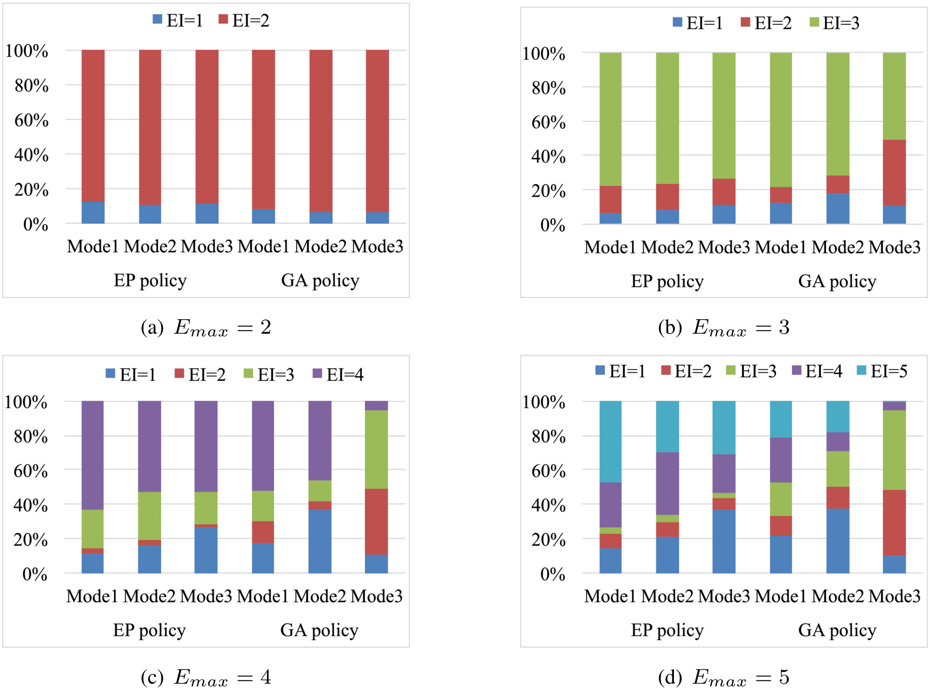
Parameter	Value
Packet generation exception	10
Packet size p_i	64..1 518 Byte
Basic latency	0.02 s
Upper limit of latency D_{max}	0.4 s
Algorithm processing performance	15 Kbps, 12 Kbps, 10 Kbps, 8 Kbps, 6 Kbps

B. VERIFICATION OF FEEDBACK-BASED SCHEDULING

To evaluate the suitability of the feedback-based scheduling scheme, the latency and throughput are used as performance metrics.

The settings of the simulation parameters are given in Table 5, in which the packets arrival following a Poisson distribution, while the sizes of the packets follow a uniform distribution. The probed packets belong to the Packet-in or Packet-out messages.

The results shown in Figure 6 indicate that the use of the feedback-based IPsec can provide a more stable latency with a concentrated distribution and fluctuation within a narrow range. From Figure 6(a), it can be seen that the link with the native scheduling scheme achieves an average latency of 0.39 s and the distribution of the latency is scattered with large variation. The results for the case where the feedback-based scheduling scheme is employed are presented in Figure 6(b). The average latency of the feedback-based case is 0.26 s. When detecting a latency exceeding the normal range of values, the algorithm currently in use will be replaced by another to keep the peak value of single-packet latency below 0.61 s.



(a) $E_{max} = 2$

(b) $E_{max} = 3$

(c) $E_{max} = 4$

(d) $E_{max} = 5$

FIGURE 9. Hit rates of encryption intensities applying EP and GA policies in three feedback modes for a total of 2 023 adjustments (EI: encryption intensity; EP: encryption precedence; GA:gradual adjustment).

The results presented in Figure 7 show that when the fixed-strength algorithm is adopted (the strength level is 3 in here), the throughput will decrease as the packet-arrival rate increases, such as the results at the fourth, 11th, 12th, 20th, and 30th seconds. However, after introducing the feedback-based mechanism for scheduling IPsec encryption algorithms, the system will actively reduce the requirement for security strength and encryption latency to promote the throughput due to the network performance degradation. It is obvious that the reduction of the encryption strength gives rise to the increase of throughput at the time 1.2 s. Furthermore, the system will continue to reduce the encryption strength to ensure the throughput rate, while the network load keeps increasing.

From Figure 7, it is clear that our proposed mechanism will increase the encryption strength to guarantee the system security if the throughput rate continues to increase, e.g., the cases from 9.66 s to 10.56 s, and from 71.27 s to 72.54 s. While the throughput is guaranteed, the encryption strength tends to remain at a level close to that of a fixed-strength encryption algorithm. Nevertheless, as the encryption intensity of the system fluctuates, the network throughput may be higher or lower compared to the fixed-strength encryption. Note that the overall throughput rate remains at a medium level due to the fixed distributions of the packet arrival rate and packet sizes in our simulation experiments.

In summary, the feedback-based scheduling can effectively switch between IPsec cryptographic algorithms and provide the SBI communication with a good balance between link security and communication performance.

C. EVALUATION OF ALGORITHM DETERMINATION POLICIES

This section evaluates the impact of the two proposed determination policies for encryption intensity on network performance.

1) EXPERIMENTAL SCHEME

The experimental platform of this section is based on Matlab, in which the emulator is used to simulate the algorithm scheduler of the invocation mechanism proposed in Section IV. The platform is in software encryption mode and provides five encryption schedulers with different encryption intensities. The scheduling time of each encryption algorithm (i.e., the processing latency D_p) is fixed at 0.1 ms. The four thresholds r_1, r_2, r_3 and r_4 for the EP policy describing the range of the independent variables are 0.9, 0.8, 0.75 and 0.5, respectively, and the two thresholds for the GA policy are 0.8 and 0.2. It should be noted that the greater the value of E_{max}, the more encryption intensities there are for the system to choose from when scheduling IPsec algorithms.

The packet interval follows an exponential distribution with a mean of 0.1 s, and the packet size p_{size} follows a uniform distribution within the range $[a, b]$, where $a = 64$ and $b = 1\,518$. The encryption latency D_e of a packet is equal to the product of the packet size and the encryption intensity. All experiments were implemented in the scheduler of the data sender, and the forwarding latency D_f is therefore neglected. The default value of the maximum encryption intensity threshold E_{max} is 5, and the number of interval packets k is 10.

2) PACKET PROCESSING LATENCY

The processing latencies of the packets through the SBI are illustrated in Figure 8, where different determination policies for the encryption intensity are applied. We can see that both of the proposed policies can reduce the processing latency of packets in three feedback modes.

The non-feedback mode in Figure 8 means that once an algorithm is selected for encryption according to the requirements of security and network performance, it is not replaced until a modification instruction is received from the system. Compared with the non-feedback case, the processing time of the southbound-communication packets is obviously reduced in the three feedback modes proposed in Section IV-C3. The degree of reduction is sensitive to the number of encryption algorithms selected with weak intensity, i.e., it is dependent on the maximum encryption intensity threshold E_{max}.

If the requirements for network security are very strict, the processing latency becomes too high to meet the network requirements. When the trade-off between the network security intensity and the demand for transmission performance is not in balance, the feedback-based algorithm scheduler designed in this paper can maintain transmission performance by selecting encryption algorithms with weaker encryption.

3) DISTRIBUTION OF OPTIONAL ENCRYPTION INTENSITY

As shown by the experimental results presented in Figure 9, the algorithm scheduler guarantees the performance requirements by selecting a weak encryption intensity, and the chosen intensity is not always the weakest. If the requirements are well guaranteed, the cryptographic algorithm that is currently in use will be replaced by another one in the next adjustment period. The newly applied algorithm should also meet the security requirements and have a stronger encryption intensity in order to provide the network with superior security. Otherwise, the algorithm scheduler will select a cryptographic algorithm that offers a weaker encryption intensity on the basis of the security requirements, in order to give priority to the transmission performance. In other words, weaker encryption intensities are also selected and play a role, as shown in Figure 9(b).

In all three feedback modes, the performance of the scenarios in which the EP policy is applied is generally better than that for the GA policy. The GA policy does not always select the algorithm with the highest encryption intensity, but switches between medium-intensity algorithms. In short, the EP policy is more effective when the period of switching encryption intensity (i.e., updating cryptographic algorithm) is a key factor and cannot be concerned.

VI. CONCLUSION

In this paper, IPsec has been introduced to guarantee communication security for SDN controller and OpenFlow switches, providing a lightweight and scalable cryptographic service. Based on FTA, a mechanism for IPsec cryptographic algorithm invocation, we have designed a feedback-based scheduling scheme for algorithm invocation that takes into consideration a trade-off between the level of the link security and the requirement for communication performance. By applying this scheduling scheme, the impact of the IPsec cryptographic process on the throughput and latency can be reduced during periods in which the network traffic fluctuates markedly. The system resources can be used to process the key information and data more efficiently.

ACKNOWLEDGMENT

This article was presented in part at the ChinaCom, Chengdu, China, [30], in October 2018. The present version includes an improved performance analysis, evaluation model, and feedback-based scheduling scheme. The authors have also expanded the experiments in this version.

REFERENCES

[1] T. Alharbi and M. Portmann, "The (in)security of virtualization in software defined networks," *IEEE Access*, vol. 7, pp. 66584–66594, 2019.

[2] V. Varadharajan, K. Karmakar, U. Tupakula, and M. Hitchens, "A policy-based security architecture for software-defined networks," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 4, pp. 897–912, Apr. 2019.

[3] J. Wu, M. Dong, K. Ota, J. Li, and Z. Guan, "Big data analysis-based secure cluster management for optimized control plane in software-defined networks," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 1, pp. 27–38, Mar. 2018.

[4] S. Hares, D. Lopez, M. Zarny, C. Jacquenet, R. Kumar, and J. P. Jeong, *Interface to Network Security Functions (I2NSF): Problem Statement and Use Cases*, document RFC 8192, Jul. 2017. [Online]. Available: https://rfc-editor.org/rfc/rfc8192.txt

[5] (Mar. 2015). *OpenFlow Switch Specification Version 1.5.1*, Open Networking Foundation Std. [Online]. Available: https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-swit%ch-v1.5.1.pdf

[6] M. M. Wang, J. W. Liu, J. Chen, J. Mao, and K. F. Mao, "Software defined networking: Security model, threats and mechanism," *J. Softw.*, vol. 27, no. 4, pp. 969–992, 2016.

[7] G. Lopez-Millan, R. Marin-Lopez, and F. Pereniguez-Garcia, "Towards a standard SDN-based IPsec management framework," *Comput. Standards Interface*, vol. 66, Oct. 2019, Art. no. 103357.

[8] S. Raza, S. Duquennoy, J. Höglund, U. Roedig, and T. Voigt, "Secure communication for the Internet of Things–A comparison of link-layer security and IPsec for 6LoWPAN," *Secur. Commun. Netw.*, vol. 7, no. 12, pp. 2654–2668, Dec. 2014.

[9] Z. V. A. Roohi, S. Megha, S. M. Ghori, and H. P. Sahana, "Design of a software tool to verify the security level of cryptographic algorithm," *Int. J. Modern Eng. Res.*, vol. 6, no. 6, pp. 25–32, Jun. 2016.

[10] C. Yoon, S. Lee, H. Kang, T. Park, S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "Flow wars: Systemizing the attack surface and defenses in software-defined networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3514–3530, Dec. 2017.

[11] S. González, A. de la Oliva, X. Costa-Pérez, A. Di Giglio, F. Cavaliere, T. Deiß, X. Li, and A. Mourad, "5G-crosshaul: An SDN/NFV control and data plane architecture for the 5G integrated Fronthaul/Backhaul," *Trans. Emerg. Telecommun. Technol.*, vol. 27, no. 9, pp. 1196–1205, Sep. 2016.

[12] N. A. Aziz, T. Mantoro, M. A. Khairudin, and A. F. B. A. Murshid, "Software defined networking (SDN) and its security issues," in *Proc. Int. Conf. Comput., Eng., Design (ICCED)*, Sep. 2018, pp. 40–45.

[13] P. Porras, S. Cheung, M. Fong, K. Skinner, and V. Yegneswaran, "Securing the software defined network control layer," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2015, pp. 1–15.

[14] S. Shin, Y. Song, T. Lee, S. Lee, J. Chung, P. Porras, V. Yegneswaran, J. Noh, and B. B. Kang, "Rosemary: A robust, secure, and high-performance network operating system," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. CCS*, 2014, pp. 78–89.

[15] F. Klaedtke, G. O. Karame, R. Bifulco, and H. Cui, "Access control for SDN controllers," in *Proc. 3rd Workshop Hot Topics Softw. Defined Netw. HotSDN*, 2014, pp. 219–220.

[16] F. Giesen, F. Kohlar, and D. Stebila, "On the security of TLS renegotiation," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. CCS*, 2013, pp. 387–398.

[17] M. L. Das and N. Samdaria, "On the security of SSL/TLS-enabled applications," *Appl. Comput. Informat.*, vol. 10, nos. 1–2, pp. 68–81, Jan. 2014.

[18] K. Benton, L. J. Camp, and C. Small, "OpenFlow vulnerability assessment," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. HotSDN*, 2013, pp. 151–152.

[19] S. Hong, L. Xu, H. Wang, and G. Gu, "Poisoning network visibility in software-defined networks: New attacks and countermeasures," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2015, pp. 1–15.

[20] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD: Scalable and vigilant switch flow management in software-defined networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. CCS*, 2013, pp. 413–424.

[21] K. Seo and S. Kent, *Security Architecture for the Internet Protocol*, document RFC 4301, Karen Seo and Stephen Kent Std. 4301, Dec. 2005. [Online]. Available: https://rfc-editor.org/rfc/rfc4301.txt

[22] A. A. Al-khatib and R. Hassan, "Impact of IPSec protocol on the performance of network real-time applications: A review," *Int. J. Netw. Secur.*, vol. 20, no. 5, pp. 811–819, Sep. 2018.

[23] R. Lopez, G. Lopez-Millan, and F. Pereniguez-Garcia. (Aug. 2019). *Software-Defined Networking (SDN)-Based IPsec Flow Protection IETF*. Internet-Draft draft-ietf-i2nsf-sdn-ipsec-flow-protection-07. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-sdn-ipsec-flow-p%rotection-07

[24] N. D. Jorstad and L. T. Smith, "Cryptographic algorithm metrics," in *Proc. 20th Nat. Inf. Syst. Secur. Conf.*, Baltimore, MD, USA, Oct. 1997.

[25] K. W. Dam and H. S. Lin, *Cryptography's Role in Securing the Information Society*. Washington, DC, USA: National Academy Press, 1996.

[26] E. Barker, "Recommendation for key management—Part 1: General," National Institute of Standards and Technology, Gaithersburg, MD, USA, Tech. Rep., May 2020. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf, doi: 10.6028/NIST.SP.800-57pt1r5.

[27] X. Yang, D. Wang, W. Feng, J. Wu, and W. Tang, "Cryptographic algorithm invocation based on software-defined everything in IPsec," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–11, Jul. 2018.

[28] E. Barker, "Guideline for using cryptographic standards in the federal government cryptographic mechanisms," National Institute of Standards and Technology, Gaithersburg, MD, USA, Tech. Rep., 2020, doi: 10.6028/NIST.SP.800-175Br1.

[29] H. Rifa-Pous and J. Herrera-Joancomartí, "Computational and energy costs of cryptographic algorithms on handheld devices," *Future Internet*, vol. 3, no. 1, pp. 31–48, Feb. 2011.

[30] D. Wang, W. Tang, X. Yang, and W. Feng, "Cryptographic algorithm invocation in IPsec: Guaranteeing the communication security in the southbound interface of SDN networks," in *Proc. Commun. Netw., 13th EAI Int. Conf.*, Chengdu, China: Springer, Oct. 2019, pp. 583–592.

XIMIN YANG received the B.S. and M.S. degrees in computer application technology from South-Central University for Nationalities (SCUN), Wuhan, China, in 1994 and 2003, respectively, and the Ph.D. degree in computer architecture from the Huazhong University of Science and Technology, Wuhan, in 2010. He is currently an Associate Professor with the College of Computer Science, SCUN. His research interests include software defined networking, network security, and so on.

DEQIANG WANG received the B.S. degree from the Department of Information and Computing Science, South-Central University for Nationalities (SCUN), Wuhan, China, in 2013, and the M.S. degree in computer system architecture from SCUN, in 2018. He is currently with Sangfor Technologies Inc. His research interests include software-defined networking, network protocols, cloud security, and so on.

WAN TANG (Member, IEEE) received the B.S. and M.S. degrees in computer application technology from South-Central University for Nationalities (SCUN), Wuhan, China, in 1995 and 2001, respectively, and the Ph.D. degree in communication and information systems from Wuhan University, China, in 2009. From 2001 to 2002, she was a Visiting Scholar with the Department of Computer Engineering, Chonbuk National University, South Korea. From 2012 to 2013, she was a Visiting Scholar with the Department of Computer Science and Engineering, SUNY at Buffalo, USA. She is currently a Professor with the College of Computer Science, SCUN. Her research interests include network protocols, software defined networking, network security, and so on.

WEI FENG received the B.S. degree from the Department of Computer Science, Wuhan Institute of Technology, Wuhan, China, in 2017, and the M.S. degree in information security from South-Central University for Nationalities (SCUN), Wuhan, in 2020. His research interests include network security and software defined networking.

CUITAO ZHU received the B.S. degree from the School of Computer Science, Beijing University of Posts and Telecommunications, Beijing, China, in 1992, and the M.S. and Ph.D. degrees in electrical and electronic engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1999 and 2008, respectively. He is currently a Professor with the College of Electronics and Information Engineering, South-Central University for Nationalities, Wuhan. His research interests include 5G wireless networks, radio resource management, cognitive radio networks, and massive MIMO.