Cryptographic Algorithm Invocation in IPsec: Guaranteeing the Communication Security in the Southbound Interface of SDN Networks

Deqiang Wang, Wan Tang(✉), Ximin Yang, and Wei Feng

College of Computer Science, South-Central Univ. for Nationalities, Wuhan 430074, China
tangwan@scuec.edu.cn

Abstract. Due to the static configuration of IPsec cryptographic algorithms, the invocation of these algorithms cannot be dynamically self-adaptable to the traffic fluctuation of software-defined networking (SDN) southbound communication. In this paper, an invocation mechanism, based on the Free-to-Add (FTA) scheme, is proposed to optimize the invocation mode of cryptographic algorithms in traditional IPsec. To balance the link security and communication performance, a feedback-based scheduling approach is designed for the controller of IPsec-applied SDN to replace flexibly and switch synchronously the IPsec cryptographic algorithms in use according to the real-time network status. The feedback information is applied to decide which appropriate algorithm(s) should be employed for the cryptographic process in a special application scenario. The validity and effectiveness of the proposed invocation mechanism are verified and evaluated on a small-scale SDN/OpenFlow platform with the deployed IPsec security gateway. The results show that the FTA-based mechanism invokes IPsec encryption algorithms consistently with the requirement for communication security in the SDN southbound interface, and the impact of the IPsec cryptographic process on the network performance will be reduced even if the network traffic fluctuates markedly.

Keywords: Communication security · Software-defined networking (SDN) IPsec · Algorithm invocation · Southbound interface (SBI)

1 Introduction

The software-defined networking (SDN) paradigm decouples the control plane from the underlying data plane and introduces network programmability and other features to promote network flexibility, adapting to the constantly changing network condition and facilitating the network's verification and deployment. However, due to less consideration of security issues in the initial design period of the SDN architecture, some new features introduced into SDN provide more convenience for the network management, but some new types of security threats consequently emerge [1–3].

The OpenFlow protocol is a widely adopted communication standard for the southbound interface (SBI) in SDN networks. While the control plane communicates with the data plane using the OpenFlow-supported instructions, the feature of

X. Liu et al. (Eds.): ChinaCom 2018, LNICST 262, pp. 583–592, 2019.
https://doi.org/10.1007/978-3-030-06161-6_57

separation between these two planes makes it insecure for the control flows when passing through exterior network links. OpenFlow thereby cooperates with the transport layer security (TLS) protocol to secure the communication between the SDN controllers and the switches (i.e. the SBI communication) [4]. Nevertheless, the TLS is too complicated in the verification and too fragile in defense of man-in-the-middle (MITM) attacks to guarantee the security, and it becomes optional instead of mandatory for OpenFlow [5]. Without the security protection of TLS, the TCP-based SBI communication is vulnerable to the tapping and forgery of control information, which makes the network more insecure and unreliable.

In recent years, some schemes have been proposed to enhance the SBI security [6–8]. These new controllers achieve better and more comprehensive security than general SDN controllers and reduce the risk from SBI. However, some risks still cannot be eliminated by the new 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 [9].

Internet protocol security (IPsec) is introduced to guarantee the security in the southbound interface of the controller and maintain secure communication between the controller and the switches in the SDN network [10, 11]. IPsec, originally developed for IPv6, can ensure the communication security in the Internet layer and does not require extra support from the controller. Meanwhile, as part of IPv6, it is in line with the current network evolution trend.

The IPsec protocol is mature in architecture but rigid in the invocation of cryptographic algorithms. Besides, the demand for customized algorithms or more algorithms supported by IPsec is urgent in various application scenarios, along with the mounting importance of network security. However, less research focuses on the flexibility of the subsequent invocation of IPsec algorithms. The rigid invocation of cryptographic algorithms in IPsec makes it hard to meet the diversified security demands of networks [12].

Furthermore, the invocation of these algorithms should consider the trade-off between the link security offered by IPsec and the communication performance of the SBI in SDN networks [11]. When IPsec is adopted to secure the communication between the SDN controllers and the switches, it is troublesome for the user to add the customized algorithm to the switches, because the vendors have limited the modification of switches, and device addition or upgrading thereby results in large costs [13]. Additionally, IPsec encryption/decryption will increase the performance consumption, even though there are certain performance requirements of SBI communication between the controller and the switches. When the traffic fluctuates, the consumption forms a bottleneck problem of communication and may amplify the variation in traffic and communication performance.

To address the above-discussed issue of IPsec algorithm invocation in SDN, a flexible FTA-based mechanism for IPsec encryption algorithm invocation will be studied in this paper. By striking a balance between IPsec-encrypted link security and communication performance, a scheme is proposed to ensure that the SDN SBI communication security has little impact on the transmission of control messages.

The rest of the paper is organized as follows. Section 2 provides an overall view of the proposed invocation mechanism of IPsec cryptographic algorithms. Section 3

demonstrates the proposal with an experiment and compares it with the native scheme in IPsec in terms of network performance. Finally, the conclusion of this paper is drawn in Sect. 4.

2 Invocation Mechanism for IPsec Cryptographic Algorithms

2.1 IPsec in SDN Architecture

Due to the separation of control and data planes, SDN controllers and switches are in different network locations. Controllers are usually high-performance hosts or servers, so the deployment of IPsec is straightforward and convenient. For most OpenFlow switches (i.e. Juniper EX4550), the vendors tend to limit their modification. That is to say, it is difficult to implement some users' customized demands, for example special security demands, locally.

Taking the above into consideration, adding a computer card or development board, for example Raspberry Pi, to OpenFlow switches can build an IPsec secure gateway. The open architecture of IPsec facilitates the addition of a new or customized cryptographic algorithm and is helpful for building a communication system with stronger closure and higher security. Moreover, IPsec can guarantee secure communication between controllers and OpenFlow switches with the IPsec gateways located in the switches. The added computer card or development board will enable optional and easily managed operation without exerting a negative impact on the configurations of OpenFlow switches, system running, data forwarding and so on.

2.2 Free-to-Add Invocation of Cryptographic Algorithms

Retaining the basic IPsec workflow, in our preliminary work, we proposed a mechanism (shown in Fig. 1) [13] titled Free-to-Add (FTA), providing flexible cryptographic algorithm addition and invocation for IPsec in SDN networks. Compared with the method in native IPsec, FTA makes the algorithm switching more adaptable and flexible, avoiding the need to rebuild the IPsec security association (SA) or modify the configuration file and restart IPsec. Besides, FTA applies a layer of user encapsulation for a special encryption algorithm to make attacks on encryption algorithms more costly and difficult.

2.3 Feedback-Based Algorithm Scheduling

In this section, we will discuss the method for scheduling IPsec cryptographic algorithms in FTA considering the requirements of both communication performance and link security.

Impact on Network Performance and System Resources. Primitively, to simplify the analysis of the impact that the cryptographic algorithm exerts on the network performance, we make three assumptions as follows:

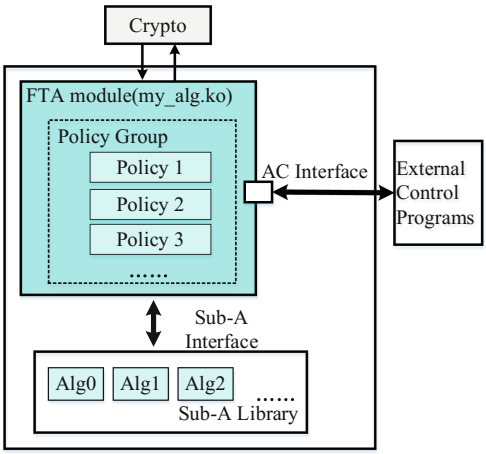


Fig. 1. Invocation process of encryption algorithms in FTA-based IPsec (AC: algorithm control, Sub-A: sub-algorithm)

Assumption 1. There are multiple algorithms with similar security strength and different system resource requirements, and the network performance can be optimized by applying these algorithms.

Assumption 2. The link bandwidth is not considered as a system resource that is required in IPsec encryption/decryption or communication.

Assumption 3. The variation of network delay is only associated with the queuing delay and processing delay of encryption/decryption, and other delays remain unchanged.

If the IPsec encryption/decryption processes quickly with few resources, the processing delay of encryption/decryption and the queuing delay of forwarding data in the link may decrease. Thus, we know that:

- IPsec encryption/decryption may degrade the communication performance. The stronger the encryption is, the greater the resource consumption and the more processing time it takes, leading to a greater impact on the performance. The changes in system resource consumption are proportional to the strength of the IPsec encryption/decryption, and the communication performance is inversely proportional to the strength.
- When the data traffic fluctuates, the offset of packet size distribution can result in throughput reduction and an increase in network latency. At this point, we can switch to a more appropriate encryption algorithm or policy to reduce the consumption of system resources and the network latency to ensure throughput. As a consequence, the impact of the IPsec encryption/decryption on the communication performance can be reduced under a guaranteed security level.

Feedback-Based Adjustment Model. The security capability of an encryption algorithm is positively related to the system consumption. Thus, an algorithm with higher

security capability corresponds to lower communication performance and a higher level of security. Due to this feature, a less-consuming algorithm should be chosen to meet a high demand for communication performance and a higher-level algorithm for a higher security demand.

Therefore, we design a feedback-based adjustment model for the security algorithms, as shown in Fig. 2, according to the trade-off between IPsec encryption and communication performance. The state of an IPsec-secured link S is denoted by a triplet (T, D, E), where T, D and E denote the throughput, network latency and level of link security, respectively. Each cryptographic algorithm or policy corresponds to a 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 change. The security level of the network link should be adjusted according to the current T, D and security demand.

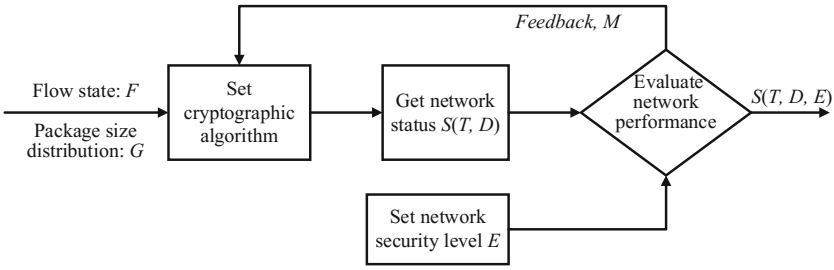


Fig. 2. Adjustment model according to the feedback

In the encryption process, an algorithm can be substituted by another algorithm with a higher security level when the security demand increases. Having set the levels of security, all the algorithms form an algorithm set A (a_1, a_2, …, a_n), $n \geq 2$, in which an algorithm with a higher index means that it can provide higher security capability. A median algorithm a_j (j = [n/2]) is selected as the initial algorithm for the SBI communication.

The algorithm to be used will be switched according to the feedback of the later communication performance evaluation. The simple adjustment model maintains a balance between encryption security and communication performance, in which the key is to adjust the proper encryption algorithm for use according to the demands of both security and performance.

Scheduling IPsec Cryptographic Algorithms. In the adjustment model presented in the previous section, the feedback information indicates which appropriate algorithm should be employed for the cryptographic process. An appropriate encryption algorithm is one that can meet the demands of communication performance and link security in a balanced way. Here, we present an evaluation model to find the trade-off between communication performance and link security:

$$M = \frac{w_e}{w_s}, \tag{1}$$

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

The weight of link security w_e is adjusted according to the network security status. Its value will increase when the risk of network security rises. With the security level of IPsec encryption algorithm elevating, the security risk keeps declining, and the security weight w_e is also reduced to the initial value 0. In terms of the weight of communication performance w_s, it is related to the link throughput, the network latency and the possibility of security risk. When an attack or security risk is detected, the link security becomes the primary goal, and w_s remains the same.

Assumption 3 indicates that the network delay is only related to the queuing delay and encryption/decryption delay. If the network latency exceeds the maximum latency of the normal link, it means that the queue congestion is very heavy. In this case, it is necessary to increase the value of w_s and schedule the encryption algorithm providing better performance to guarantee the communication performance. The weight of communication performance is calculated using (2):

$$w_s = \begin{cases} \max(1, \frac{D_n}{D_{\max}}) & (w_e = 1) \\ 1 & (w_e \neq 1) \end{cases}, \tag{2}$$

where D_n denotes the link latency obtained from the n-th sampling and D_{max} is the maximum latency when the network status is normal.

Since the processing performance of each encryption algorithm is not continuous and the link latency of the sampled network may change constantly, we set the algorithm switch criterion according to (3), where p_j is the processing performance of the j-th encryption algorithm in the algorithm set A. Note that the switch criterion is a value range instead of an exact value and the security level of the candidate algorithms for scheduling should meet the security demand.

$$\begin{cases} \textit{if } M \geq \frac{p_{j-1}}{p_j}; & \text{switch to the algorithm with a higher security level, until it is the inital algorithm} \\ \textit{if } M \leq \frac{p_{j+1}}{p_j}; & \text{switch to the algorithm providing higher process performance,} \\ & \text{until it is the algorithm with the lowest security level} \end{cases} \tag{3}$$

Hence, the IPsec encryption algorithm used for IPsec and SBI communication may be replaced by another more appropriate algorithm when the link security and communication performance change to keep the balance between the two factors.

3 Experiment and Verification

3.1 Validity of the FTA Mechanism

In our work, a small-scale testbed is built to verify the validity of the FTA-based mechanism. The topology is shown in Fig. 3, and the configuration information is given in Table 1. In this testbed, the communication between the SDN controller and the OpenFlow switches is the SBI communication of SDN, and two Raspberry Pis act as the IPsec security gateways that ultimately use IPsec to secure the communication between the SDN controller and the OpenFlow switches. The performances of FTA-based IPsec and native IPsec are compared using three cryptographic algorithms.

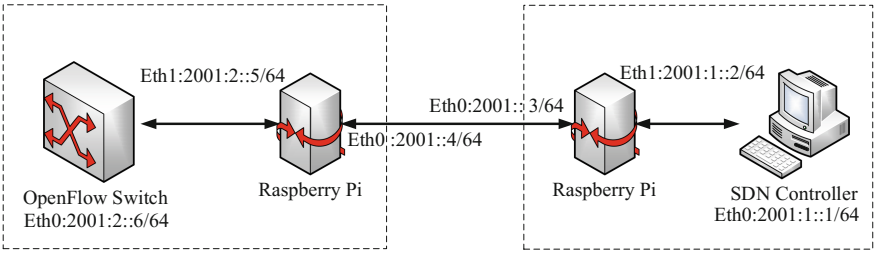


Fig. 3. Topology of the SDN testbed with IPsec gateways

Table 1. Configuration of the simulation software and hardware

SDN device	Equipment	Operating system	Hardware	Software
OpenFlow switch	Raspberry Pi 3B	Rappbian Stretch Lite	ARM Cortex-A53 1.2 GHz	OpenvSwitch 2.3.0
IPsec security gateway	Raspberry Pi 3B		Quad Core 1G RAM USB 2.0	StrongSwan5.4.0
SDN controller	DELL Inspiron 14 7000	Ubuntu 16.04.3	i5-4200H@2.8 GHz 8 G RAM, USB 2.0	OpenDaylight Beryllium SR4

From the results presented in Table 2, it can be seen that the network latencies and link throughputs are approximate in these two cases and that their variations are in the normal range. In addition, the actual bandwidths tested by *Iperf3* in both mechanisms are almost the same. In short, the use of the FTA-based mechanism has little impact on the network performance.

Table 2. 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
Request/response per second	Native IPsec	378	382	381
	FTA-based IPsec	363	394	381
Test bandwidth (Mbps)	Native IPsec	64.1	55.7	34.8
	FTA-based IPsec	63.6	53	34.7

Table 3. Parameter settings

Parameter	Value
Packet generation exception	10
Packet size	(0 ~ 1480 Byte)
Basic network latency	0.02 s
Upper limit of normal network latency	0.4 s
Algorithm processing performance	15 Kbps, 12 Kbps, 10 Kbps, 8 Kbps, 6 Kbps

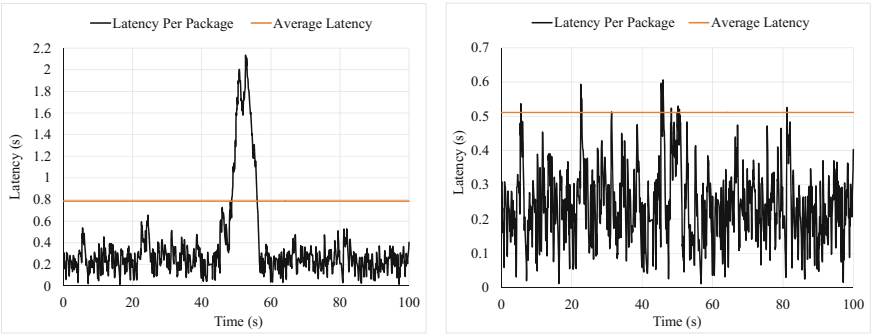
3.2 Verification of Feedback-Based Scheduling

To evaluate the availability of the feedback-based scheduling scheme, in this section, simulations are carried out using Matlab, and the network latency and link throughput are the performance metrics.

The settings of the simulation parameters are given in Table 3, in which the distribution of data packets and the sizes of the packets follow Poisson distribution and average distribution, respectively.

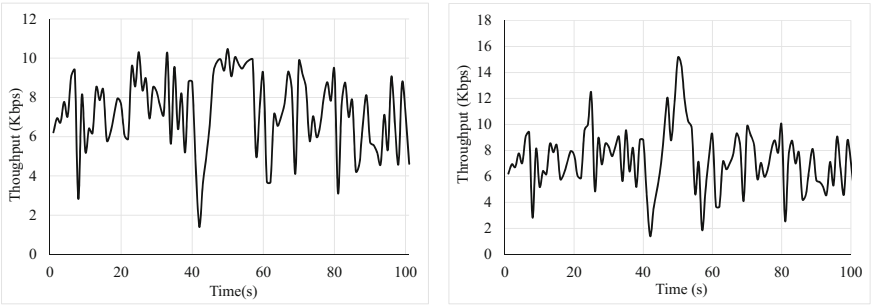
From Fig. 4(a), it is obvious that the link with the native scheduling scheme achieves an average network latency of 0.78 s and the distribution of latency is scattered with great variation. The results of the case of applying the feedback-based scheduling scheme are presented in Fig. 4 (b). The average network latency of the feedback-based case is 0.51 s. When detecting a latency exceeding the normal range of values, the algorithm will be substituted by another one to keep the peak value of latency less than 0.6 s. The results shown in Fig. 5 indicate that applying the feedback-based scheme can provide a more stable network latency with concentrated distribution and fluctuation within a narrow range.

In Fig. 5 (a), we can see that the link throughput in the native scheduling case is likely to be limited to about 10 Kbps, that is, the maximum processing performance of initial algorithms. This phenomenon is caused by the reduced processing performance in encryption/decryption nodes and results in heavy network congestion and packet loss. On the contrary, when the links apply the proposed feedback-based scheme to schedule the IPsec cryptographic algorithms, the processing performance and link throughput are improved, and the network congestion is thereby mitigated with higher throughput



(a) Native scheduling (b) Feedback-based scheduling

Fig. 4. Variations of network latency using different scheduling schemes for IPsec cryptographic algorithms



(a) Native scheduling (b) Feedback-based scheduling

Fig. 5. Variations of link throughput using different algorithm scheduling schemes for IPsec cryptographic algorithms

shown in Fig. 5 (b). In summary, the feedback-based scheduling for IPsec cryptographic algorithms can effectively switch the encryption algorithms and provides the SBI communication with a good balance between link security and communication performance.

4 Conclusion

In our work, an FTA-based mechanism considering the network feedback has been proposed to simplify the process of IPsec cryptographic algorithm addition and switching and ensure flexible and available invocation of IPsec cryptographic algorithms. The experimental results show that the feedback-based scheduling scheme enables the mechanism of algorithm invocation to invoke and switch the IPsec encryption algorithms, providing a good trade-off between the level of link security and the demand for communication performance in the SDN southbound interface. In

future work, we want to study a more adaptable invocation scheme combining the IPsec algorithms with different security levels and test it in more application scenarios and in a physical SDN.

Acknowledgement. The work described in this paper was carried out with the support of the National Natural Science Foundation of China (61772562), the China Education and Research Network (CERNT) Innovation Project (NGII20150106), and the Fundamental Research Funds for the Central Universities, South-Central University for Nationalities (CZY18014).

References

1. Kreutz, D., Ramos, F.M.V., Veríssimo, P., et al.: Software-defined networking: a comprehensive survey. Proc. IEEE **103**(1), 14–76 (2015)
2. Liyanage, M., Ahmed, I., Okwuibe, J., et al.: Enhancing security of software defined mobile networks. IEEE Access **5**, 9422–9438 (2017)
3. Chen, M., Qian, Y., Mao, S., et al.: Software-defined mobile networks security. Mobile Netw. Appl. **21**(5), 1–15 (2016)
4. OpenFlow Switch Specification v1.5.1. http://OpenFlowSwitch.org
5. Wang, M., Liu, J., Chen, J., et al.: Software defined networking: security model, threats and mechanism. J. Softw. **27**(4), 969–992 (2016)
6. Shu, Z., Wan, J., Li, D., et al.: Security in software-defined networking: Threats and countermeasures. Mobile Netw. Appl. **21**(5), 764–776 (2013)
7. Porras, P.A., Cheung, S., Fong, M.W., et al.: Securing the software-defined network control layer. In: Proceedings of Network and Distributed System Security (NDSS) Symposium, pp. 1–15. San Diego, USA (2015)
8. Scott-Hayward, S., Natarajan, S., Sezer, S.: A survey of security in software defined networks. IEEE Commun. Surv. Tutor. **18**(1), 623–654 (2016)
9. Ferguson, A.D., Guha, A., Liang, C., et al.: Participatory networking: an API for application control of SDNs. In: Proceedings of ACM Special Interest Group on Data Communication (SIGCOMM), pp. 327–338. Hong Kong, China (2013)
10. Huang, X., Yu, R., Kang, J., et al.: Software defined networking for energy harvesting Internet of Things. IEEE Internet Things J. **5**(3), 1389–1399 (2018)
11. Marin-Lopez, R., Lopez-Millan, G.: Software-defined networking (SDN)-based IPsec flow protection. I2NSF Internet-Draft. July 2018. draft-ietf-i2nsf-sdn-ipsec-flow-protection-02. https://datatracker.ietf.org/doc/draft-ietf-i2nsf-sdn-ipsec-flow-protection/
12. Al-Khatib, A.A., Hassan, R.: Impact of IPSec protocol on the performance of network real-time applications: A review. Int. J. Netw. Secur. **19**, 800–808 (2017)
13. Yang, X., Wang, D., Feng, W., Wu, J., Tang, W.: Cryptographic algorithm invocation based on software-defined everything in IPsec. Wirel. Commun. Mobile Comput. (WCMC) **2018** (2018). https://doi.org/10.1155/2018/8728424. Article ID 8728424