# Highly Secure Communication Service Architecture using SDN Switch

Masahiro FURUKAWA*, Kouki KURODA*, Takeshi OGAWA*, Noriharu MIYAHO*

*Tokyo Denki University, 2-1200 Muzaigakuen Inzai-city,Chiba, 270-1382 Japan
E-mail: *{14jkm16@ms.dendai.ac.jp, 11jk084@ms.dendai.ac.jp, t.ogawa@mail.dendai.ac.jp, miyaho@mail.dendai.ac.jp}

***Abstract*—There is an increasing demand for secure communication services that can dynamically reflect user needs. Conventional dedicated services such as an Internet VPN or IP-VPN using IPsec and MPLS have inherent shortcomings, making it difficult for them to dynamically reflect user requirements when faced with limited network resources. It is also quite important for communication services to deal effectively with man-in-the-middle (MITM) attacks, a threat that users are certainly aware of. In this paper, we proposed a state-of-the-art SDN service architecture that can reflect user requests easily, dynamically and flexibly. We also proposed a robust network mechanism that can avoid MITM attacks and network eavesdropping by applying a new network address translation method. The most important characteristics of the method are that it can easily increase security strength without terminal host/server side encoding and decoding procedures. As the number of divisions and duplications is adjustable, it is possible to strengthen the security level according to user requests. We implemented an SDN switch to evaluate the performance of the proposed SDN architecture and verified that a secure communication service using the proposed method is realizable. We expect this method to be applied to the construction of future secure SDN services.**

***Keywords*— *SDN; Disaster Recovery; Network Architecture***

## I. INTRODUCTION

There is an increasing demand for secure communication services that can dynamically reflect user needs. Conventional dedicated services such as an Internet VPN or IP-VPN using IPsec and MPLS have inherent shortcomings, making it difficult for them to dynamically reflect user requirements when faced with limited network resources. In this paper, we propose a state-of-the-art secure communication service architecture. We have previously discussed Disaster Recovery Technology (DRT) [1] which can easily realize highly secure backup services. This architecture is useful for reflecting user requests dynamically and flexibly to provide highly secure communication services. We propose a robust network architecture achieved by combining DRT and SDN (Software-Defined Networking) [2]. It is also quite important for communication services to deal effectively with man-in-the-middle (MITM) attacks, a threat that users are certainly aware of [3]. This paper describes an efficient solution to all these problems.

## II. PROBLEMS IN THE CONVENTIONAL METHOD

The IPsec system and IP-VPN systems using MPLS have been developed as conventional secure communication services. However, users will still suffer from eavesdropping in the Internet. For example, in the case of a block cipher using IPsec, various techniques such as a related-key attack or Biclique key recovery for AES [4, 5] are possible avenues for man-in-the-middle (MITM) attacks. Since these cryptanalyses are applicable to both block ciphers and hash functions, there is a risk that all or some part of user data can be deciphered. This means that when attackers are successful in network eavesdropping on a specific path and capturing all the encrypted packets, the message may be decrypted in a usefully short time period. In general, secure IP-VPN service is costly compared with ordinary Internet service. In addition, the security level is automatically predetermined regardless of the type of communication. It is difficult to accurately reflect the user-required QoS and security levels.

To improve this situation, we propose an SDN framework using multipath support for each user packet.

## III. PROPOSED SECURE COMMUNICATION TECHNOLOGY

### *A. SDN architecture and its modification*

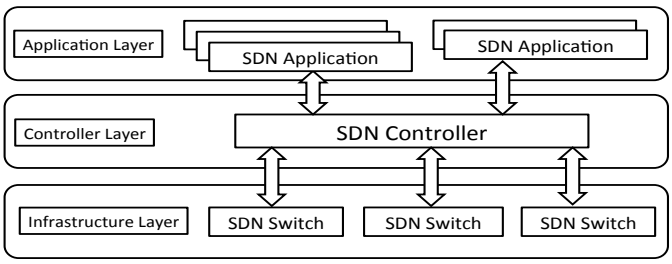


Fig.1. SDN architecture

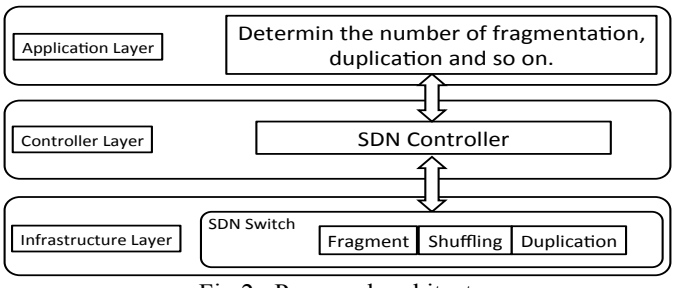


Fig.2. Proposed architecture

Fig.1 shows the general three-layer architecture of SDN, consisting of an SDN switch, SDN controller and SDN application. The SDN switch processes data packets, while the

SDN controller manages the SDN switch and selects the data processing method. Centralizing the control layer in the controller makes controlling the data layer simple and flexible. SDN software applications can dynamically control the SDN switch via the SDN controller.

The architecture of the proposed scheme is shown in Fig.2. We modified the SDN architecture by adding security-enforcing functions such as stream encryption, spatial random scrambling, fragmentation, duplication and shuffling packets inside the SDN switch by making use of disaster recovery technology [1]. The appropriate number of fragments and routing paths can be flexibly determined according to user requirements by the SDN application.

### *B. Proposed network address translation method*

We propose a network address translation method whose objective is not to notify the receiving terminal of the sending terminal's address-related information at all, in addition to avoiding MITM attacks. This method consists of two stages, as shown in Fig.3. The first stage deals with the handling mechanism at the edge-switches, in which each IP packet corresponding to the connection-oriented terminal is segmented and distributed to multiple routes via the Internet by using DRT.

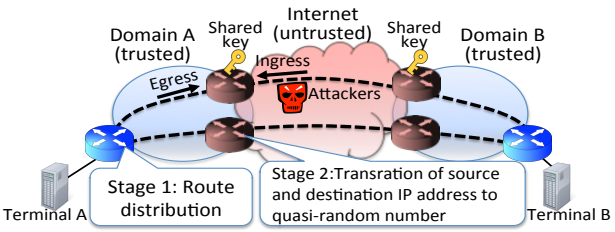


Fig.3. Proposed IP address translation mechanism

This makes it nearly impossible for MITM attackers to collect all the IP packets of a message and to decrypt the original information. In the second stage, the predetermined edge-switches connected with the destination can gather together all the related packets belonging to the specific flow, translate the destination host addresses of the egress IP packets to a quasi-random number, and translate the host addresses of ingress IP packets in reverse order to hide the IP addresses of terminals. The address translations are done with a shared key negotiated between domains A and B, which accommodate these terminals. As a result, MITM attackers cannot extract the IP packets of targeted terminals by their inherent IP addresses. Even if the attackers try an indiscriminate attack, they cannot determine which IP addresses are relevant to the target communication.

### *C. Proposed method utilizing DRT and SDN*

Encryption by DRT is performed in the following steps: spatial random scrambling of file data, subsequent random fragmentation of the file, and the duplication and encryption of each file fragment using a stream cipher code at each encryption stage. Owing to fragmentation and shuffling in the encrypted data processing, an attacker cannot decrypt the message from any of its parts or fragments. As the number of

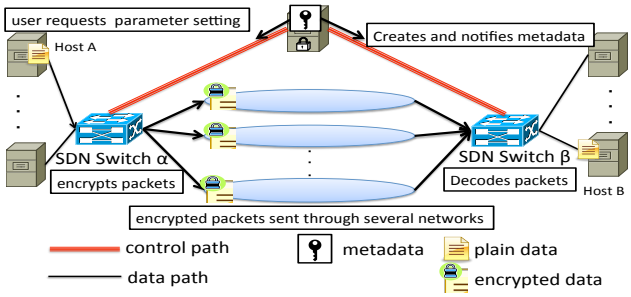


Fig.4. Physical configuration of the proposed method

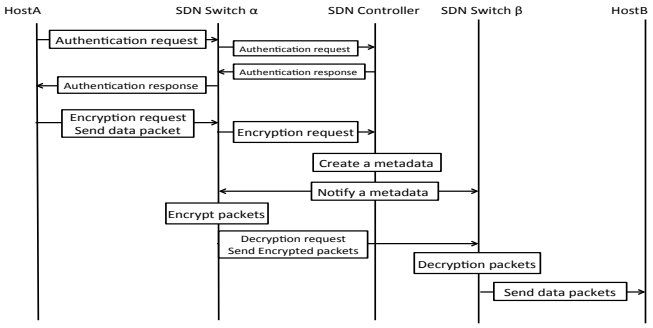


Fig.5. Processing flow when a user sends a data packet

divisions and duplications is adjustable, it is possible to strengthen the security level according to user requests.

The physical configuration of the proposed method and the corresponding processing flow are shown in Figs.4 and 5. This mechanism can provide secure communication services by combining DRT and SDN technology. SDN switch α encrypts the packet transmitted from host terminal A. The fragmented packets are randomly shuffled, and sent via different paths. There are two methods for encrypting fragments: one-by-one packet encryption, and encrypting several packets together in a batch. SDN switch β restores the received packets to the correct order and sends them to host terminal B. Attackers would need to capture and rearrange all raw packets from all the paths in the correct order. To improve communication quality, SDN switch α sends appropriate redundant packets along several paths. Service header and corresponding flags included in the user request packet in the call establishment phase show the SDN controller how to deal with the packets: authentication request, encryption request and decoding request.

SDN switch α sends a received service header to the SDN controller. The SDN controller determines the metadata that describe the encryption key, the extent of fragmentation and multiplex reproduction, and the path routes according to the user request.

The SDN controller dynamically determines the corresponding procedures, i.e. the extent of fragmentation, security level and so on.

## IV. EXPERIMENTS

### *A. Experimental system*

We implemented the proposed method by using an SDN switch and evaluated its performance by the way that packet

fragmentation and duplication influenced the end-to-end delay. We implemented SDN switches α and β in Python, and ran them in Linux on a virtual machine.

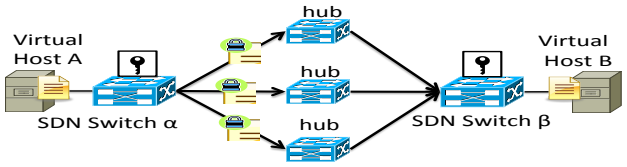


Fig.6. Configuration of the network experiment

Fig.6 shows the configuration of the test environment in a virtual network. We assumed that SDN switches α and β had metadata from the beginning and evaluated the one-by-one encryption method for each fragment in SDN switch α.

### *B. Experimental results*

We set the packet forwarding delay at 10 μs in SDN switch α to prevent packet loss in the event that more fragments required processing. In this experiment, UDP packets of 1K bytes were sent from virtual machine A to B. We measured the end-to-end delay while changing the number of fragments from 1 to 40, and multiple reproductions (packet duplication) from 1 to 3. The effect of the number of reproductions and fragments on end-to-end delay is shown in Fig.7. The processing times of switch α and switch β are shown in Fig.8 for different numbers of duplicates.

As shown in Fig.7, end-to-end delay increases in proportion to the number of fragments. When the number of fragmentations increases, the total end-to-end delay increase,

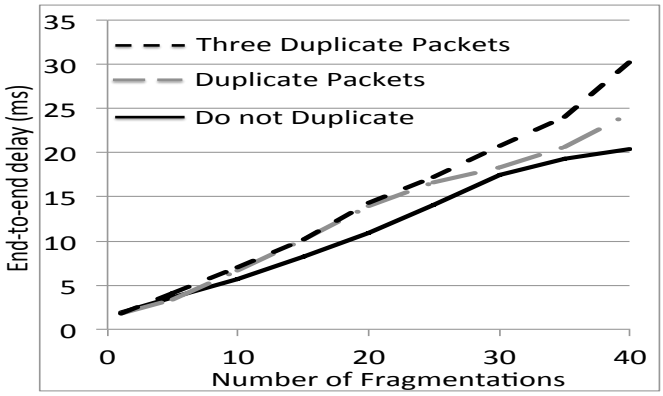


Fig.7. Measured end-to-end delay

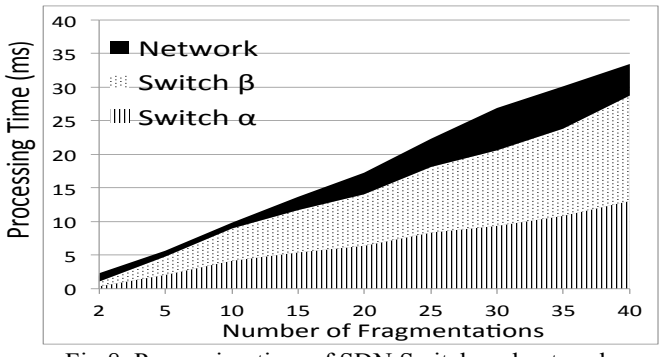


Fig.8. Processing time of SDN Switch and network

but it is not proportional to the number of fragments.

As shown in Fig.8, the switching delay inside switches α and β is approximately the same. Here delay in switch α means the elapsed time before finishing the relaying of a packet after receiving it from the source host A side, and delay in switch β means the elapsed time before finishing sending the packet to the destination B side host. This means that the encryption processing time and the decryption processing time are much the same, not dependent on the number of paths. Since the total network delay increases with the number of fragments, however, the delay in the hub switch is negligibly small.

Since the UDP communication protocol has no retransmission procedure, the acquired data err a bit on the optimistic side, but the proposed method can be effectively improved by increasing the number of reproductions in order to avoid affecting the end-to-end delay. The most important characteristics of the method are that it can easily increase security strength without terminal host/server side encoding and decoding procedures. We expect this method to be applied to the construction of future secure SDN services.

## V. Conclusion

In this paper, we proposed a state-of-the-art SDN service architecture that can reflect user requests easily, dynamically and flexibly. We also proposed a robust network mechanism that can avoid MITM attacks and network eavesdropping by applying a new network address translation method. We implemented an SDN switch to evaluate the performance of the proposed SDN architecture and verified that a secure communication service using the proposed method is realizable.

We are planning the implementation of an SDN controller that can set any kind of parameters dynamically by taking the available network resources into account.

## Acknowledgment

This work has been partially supported by a study (Issue number: 151) of the National Institute of Information and Communications Technology (NICT) of Japan.

## References

[1] N. Miyaho, S. Suzuki, Y. Ueno, K. Mori, and K. Ichihara, "Study of a Secure Backup Network Mechanism for Disaster Recovery and Practical Network Applications" IARIA Journals, vol.3, no.1, pp. 266-278, 2010.

[2] ONF (Open Networking Foundation), "Software Defined Networking: The New Norm for Network", https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf, Dec, 2014.

[3] E. Brown, B. Yuan, D. Johnson and P. Lutz, "Covert channels in the HTTP network Protocol: Channel characterization and detecting Man–in–the-Middle attacks", The Proceedings of the 5th International Conference on information warfare and security (ICIW2010), pp.56-64, April, 2010.

[4] A. Biryukov and D. Khovratovich, "Related-key cryptanalysis of the full AES-192 and AES-256," Asiacrypt 2009, LNCS, vol.5912, pp. 1–18, 2009.

[5] A. Bogdanov, D. Khovratovich and C. Rechberger, "Biclique cryptanalysis of the full AES," Asiacrypt2011, LNCS, vol.7073, pp.344–371, 2011.