Journal of The Korea Society of Computer and Information
Vol. 22 No. 9, pp. 49-55, September 2017

www.ksci.re.kr
https://doi.org/10.9708/jksci.2017.22.09.049

A Secure Switch Migration for SDN with Role-based IBC

JunHuy Lam*, Sang-Gon Lee**, Vincentius Christian Andrianto***

Abstract

Despite the Openflow's switch migration occurs after the channel was established in secure manner (optional), the current cryptography protocol cannot prevent the insider attack as the attacker possesses a valid public/private key pair. There are methods such as the certificate revocation list (CRL) or the online certificate status protocol (OCSP) that tries to revoke the compromised certificate. However, these methods require a management system or server that introduce additional overhead for the communication. Furthermore, these methods are not able to mitigate power abuse of an insider. In this paper, we propose a role-based identity-based cryptography (RB-IBC) that integrate the identity of the node along with its role so the nodes within the network can easily mitigate any role abuse of the nodes. Besides that, by combining with IBC, it will eliminate the need of exchanging certificates and hence improve the performance in a secure channel.

▸Keyword: SDN, OpenFlow, Identity Based Cryptography, role-based IBC

I. Introduction

In the conventional network, network switch stores their flow table in its memory to perform its tasks. Openflow is the de-facto southbound protocol of Software-Defined Network (SDN) that tries to allow the SDN controller to manipulate the flow table of the switches from the control plane. By doing so, it centralizes and simplifies the network management. However, this introduce the new communication channel in SDN - southbound communication.

Southbound communication is used by the SDN controller to issue the commands to the network switches such as the installation or removal of flow rules, the mastership of the network switches and etc.

One of the concerns raised in this southbound communication is the network switch migration in which the SDN controller has the authority to migrate the network switch that was under the control of another controller to its control. If the southbound communication is not secured, any device can actually try to issue this command to the network switches and gain control of the network.

Therefore, it is essential to secure the communication channel in order to prevent the attacker to learn or even manipulate the network topology.

This is an improved version of our previous work [1] that described the application of identity-based cryptography (IBC) on distributed SDN. In this paper, we proposed the role-based IBC to mitigate the role abuse of the devices. To the best of our knowledge, this is the first application of the role-based access control (RBAC)

• First Author: JunHuy Lam, Corresponding Author: Sang-Gon Lee
*JunHuy Lam (timljh@gmail.com), Dept. of Ubiquitous IT, Dongseo University
**Sang-Gon Lee (nok60@dongseo.ac.kr), Div. of Computer Engineering, Dongseo University
***Vincentius Christian Andrianto (vincent.ch93@gmail.com), Dept. of Ubiquitous IT, Dongseo University

• Received: 2017. 05. 12, Revised: 2017. 06. 12, Accepted: 2017. 08. 28.
• This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grant Number: 2014R1A1A2060021).

of IBC on SDN. The rest of the paper is organized as follow, Section II explains the background knowledge of switch migration, IBC and the related works. Section III describes the role-based IBC (RB-IBC) and the discussions can be found in Section IV. The paper ends with the conclusions in Section V.

II. Background

1. Switch migration

As mentioned in the OpenFlow specifications [2], a switch migration is initiated by controller by sending the OFPT_ROLE_REQUEST command to the switch. There are three possible roles of a controller to a switch, namely, OFPCR_ROLE_MASTER(Master), OFPCR_ROLE_EQUA(Equal) and OFPCR_ROLE_SLAVE (slave).

The default role of a controller to the switch is Equal. The controller can request for the role change to either master or slave from then. The master and equal roles both have full access to the switch. By default, the controller with the role equal or master can receive all of the switch asynchronous messages which include the packet_in, flow removed messages. Besides that, they can also send controller-switch commands to modify the state of the switch.

The only difference between the master and equal role is that there is only 1 master for a switch at a time but multiple controllers with the equal role can exists for a switch. Due to this criteria, whenever a controller requests for its role to be changed to master for a particular switch, the switch needs to send the OFPT_ROLE_STATUS message to its current master controller to inform it about this role change. Then, the current master controller will be switched to a slave controller.

A slave controller has read-only access to the switch and do not receive switch asynchronous messages except the port status message. Besides that, it also cannot execute controller-switch commands with the data query commands being the exceptions, this includes OFPT_ROLE_REQUEST, OFPT_SET_ASYNC, OFPT_MULTIPART_REQUEST messages.

Typically, a switch migration is required under a few circumstances as stated below,

- Congested controller
- Controller down time
- Change of topology

When the controller is overloaded, the switch migration will be triggered to move one or more network switches under its control to a controller which is less busy. This is part of the load balancing process to distribute the load between the controllers in a distributed SDN network.

The second scenario where a switch migration will happen is when the master controller of the network switch is down. An online controller will be notified of the controller that went down and initiate the switch migration process by sending a OFPT_ROLE_REQUEST command to the switch for the master role.

The third scenario is when the network administrator changes the network topology, it can be for optimization, segregation or any other purpose that the network administrator deems fit for such changes. In this case, the network administrator will trigger the switch migration manually through the controllers.

As per the OpenFlow specification, the southbound communication can be secured by Transport Layer Security (TLS). By doing so, it can prevent the attacker from manipulating the network switches through this communication channel. However, even if the channel is secured with TLS, the network is not able to mitigate the identity abuse by the node.

For example, if the attacker managed to obtain a public/private key pair of any device within the network even if this compromised device is not controller, it can disguise itself as a controller by using the stolen key. The network is not able to mitigate this attack and allows the attacker to establish a “secure” channel despite its role abuse. Hence, a compromised switch or even a compromised host will compromise the entire SDN network with the conventional security.

2. Identity-based Cryptography (IBC)

Shamir introduced an identity-based signature scheme [3] in 1984and it became the basis of IBC research thereafter. His idea of IBC was then implemented by Sakai et al. [4] and Boneh et al. [5] for the encryption scheme with pairing in the year 2000 and 2001 respectively.

Similar to the Public Key Cryptography (PKC) of TLS, IBC requires a Trusted Authority (TA) to act as a Private Key Generator (PKG) that generates keys for the users. In the SDN environment, controllers can also act as PKGs for the switches that are located within its domain.

In PKC, CA is used to generate the public and private key pairs whereas the PKG of IBC generates only the private keys. In IBC, public keys will be derived from the identity of the user; in this case, the user's identity can be in the form of the Media Access Control (MAC) address or any other network identities of the controllers and switches.

With IBC, the users, or in this case, the controllers, switches or data stores, do not need to store every single public key of every user in the domain it communicates to or obtain a particular public key from the TA on demand because it will be able to derive it from the identity information. This in turn saves storage space and network bandwidth that otherwise can decrease the network performance or translate into high system setup costs.

Smart [6], whose research was based on the implementation of Boneh and Franklin, initiated the usage of IBC in key establishment protocol. Chen et al. [7] improved on Smart's protocol by solving the key escrow problem, allowing the communication between users of multiple TAs and providing forward secrecy.

3. Role-Based Access Control (RBAC)

As described by F.F. David et al. [8], RBAC policies are managed in terms of users, subjects, roles, role hierarchies, operations and protected objects. Network administrator will have the capability to place constraints on the jurisdiction and operation that can be executed by each user by assigning the role to each user.

Certain roles might have overlapping authorities and operations and hence role hierarchies can be useful by creating new roles by inheriting from the top role hierarchy of such role. This works similar with the inheritance in object-oriented programming. For example, in SDN network, basic operation such as reading flow information should be allowed on most nodes and hence a parent role with reading access of flow information can be added at the top of the hierarchy and inherited by derived roles such as the controller, network switches, network applications and etc.

4. Related Works

J.S. Park et al. [9] proposed to secure the web with RBAC, it organizes the roles to reflect the organization's lines of authority and responsibility. They also secure the cookies with authentication, integrity and confidentiality so it can verify the owner of the cookies, protect cookies from unauthorized modification and having its content revealed to unauthorized entity. However, these access control is not applicable for the network management.

D. Nali et al. [10] proposed to use a mediated IBC to support RBAC. The advantage of their proposal is that it allows online user revocation but this can also be a disadvantage especially for the IBC protocol because it introduces an additional communication overhead for the user to retrieve the revoked user frequently.

Besides that, this method also tries to integrate the identity information as part of the "public key"of the mediated IBC that created an additional step for the user to retrieve the other part of identity with the role information from the mediator. This in turn creates the additional communication overhead and reduces the performance significantly.

Since network performance is not that important for the human-based user, this method can provide an efficient role-based management with revocation ability. However, this method is not suitable for the device's roles management where network performance is one of the most important issue to consider.

S. Shin et al. [11] proposed AVANT-GUARD to mitigate the scanning and denial-of-service (DoS) attacks by adding intelligence back to the data plane or they called it the connection migration that allow the data plane to differentiate the sources that will complete a TCP connection from sources that do not. Besides that, they also introduced actuating triggers that automatically insert flow rules when the network is under duress.

They also proved that their proposal is able to mitigate such attacks with minimal impact on the network performance. However, this proposal violates the plane separation of SDN where the data plane should be stripped off its intelligence.

III. Role-based IBC

Our proposal, the role-based IBC (RB-IBC) protocol makes use of the IBC protocol that uses the identity information to replace the certificate/public key. This is an improved version of our previous work [1] that mitigate the possible threat of southbound communication. In this protocol, a PKG will issues the private key for a particular identity. Our proposal modifies the way the PKG generates the private key by inserting the role to the identity information prior the private key generation. Hence, the private key generated by the PKG to the devices will be integrated with the role information.

Any device that tries to perform any action that is not within its role's jurisdiction will not be able to complete the handshake process during the session establishment. This is simply because the attacker does not have the private key that is needed to decrypt the FINISHED handshake message and respond accordingly.

Without being able to complete the handshake, it mitigates the attack that abuse its identity information to perform any activities that is not within its jurisdiction. Unlike the CRL, OCSP or mediator-based solution, this method neither introduce an additional server to distribute this information nor overhead for the channel establishment.

In SDN environment, there are four roles information that are publicly available for the "public key" derivation along with the identity information and they are listed as below,

- Network applications
- SDN Controllers
- Network switches
- Hosts

1. IBC Key Establishment

The IBC key establishment protocol is based uponour previous work [9] and added the RBAC to mitigate the role abuse of the nodes. The system setup and protocol are illustrated as below.

1.1 System setup

Supposing there are two PKGs,PKG_1 and PKG_2, that generate the private keys for the controllers of the SDN. Each has a public/private key pair, $(P, s_1P \in G_1, s_1 \in_R Z_q^*)$ and $(P, s_2P \in G_1, s_2 \in_R Z_q^*)$ respectively, where P and G_1 have been globally agreed on.

Controller A, $controller_A$, is registered under PKG_1 with its private key, $S_A = s_1 Q_A$ where $Q_A = H_1(controller_A{}'s\,ID|RBAC_{controller})$. Controller B, $controller_B$, is registered under PKG_2 with $S_B = s_2 Q_B$ where $Q_B = H_1(controller_B{}'s\,ID|RBAC_{controller})$. The "public key" will be derived by using the H_1 function of the concatenation of the controller's identity and $RBAC_{controller}$: the role information of a controller that is publicly available. H_1 is a cryptographic hash function, $H_1 : \{0,1\}^* \rightarrow G_1$.

Since the PKG is the only entity that is able to derive the private key to the controller, it is not possible for the adversary to imitate a controller with a stolen private key which does not belong to a controller.

1.2 Key establishment

If controller A wants to communicate with controller B, the RB-IBC will be initiated to establish the shared session keys. Controller A and B each picks a nonce at random, a and $b \in_R Z_q^*$ and computes $T_A = aP$, $W_A = aP_{pub,crlr2}$ and $T_B = bP$, $W_B = bP_{pub,crlr1}$, respectively, where $P_{pub,ctrl1} = s_1P$ and $P_{pub,ctrl2} = s_2P$. These computed values will then be exchanged between the two controllers.

At the end of the protocol, controller A computes the shared key, $K_{AB} = \hat{e}(S_{A,} T_B)\hat{e}(Q_{B,} W_A)$, and controller B computes the shared key,$K_{BA} = \hat{e}(S_{B,} T_A)\hat{e}(Q_{A,} W_B)$.

$$\therefore K_{BA} = K_{AB} \tag{1}$$

Then, the shared session key can be generated by hashing the key, $SK = h_2(K_{AB})$, where h_2 is a secure hash function for the purpose of key derivation.

IV. Security analysis and discussions

In the current SDN southbound communication, TLS secure channel is only an optional feature in OpenFlow. Network administrators also tend to avoid using it since most SDN network is deployed for internal use only. By implementing TLS for the southbound communication, it impacts the performance as well as the troublesome certificate management. IBC manages to ease the certificate management but it still affects the performance like any other secure channel. However, the expected performance degradation should be much lower than TLS as it does not require to exchange certificates between the communicating devices.

In conventional switch migration, a controller will issue the OFPT_ROLE_REQUEST command to a switch and migrate its role on the switch to Master/Equal in order to gain control of the switch. However, this mechanism can be exploited even if the network is secured with TLS. The adversary can attack any device within the network (even if it is not a controller since controller will be well protected as it is the core of a SDN network) and use the stolen key pairs to initiate connection as a disguised controller.

This role abuse renders the protection to the controller useless as an attack to any device within the network can compromise the entire network when the adversary uses the credential, disguise as a controller and then gain control of the entire network. Figure 1 illustrates the attacks on the management plane or data plane to obtain the key (This scenario assumes that the network administrator protects the control plane and it will be more difficult to penetrate the control plane).

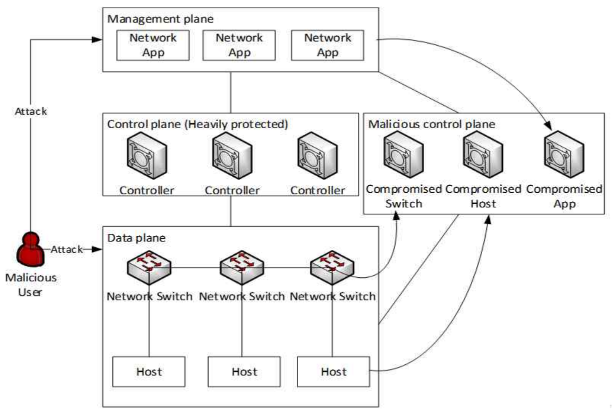

Fig. 1. Possible attack on the management plane or data plane to take control of the instances and disguise as controllers

As illustrated in Figure 1, even though the nodes within the management plane, control plane and data plane appears to be different; In most cases, they are practically a compute node that runs different software (network application software, SDN controller software, virtual network switch software or simply a compute node host). Hence, the malicious user can simply run a SDN controller software on any compromised node regardless of its existing role and disguised itself as a controller.

Therefore, securing the core of SDN, the control plane alone is insufficient to secure the network as a whole. The instances in the management plane or data plane cannot differentiate the genuine control plane from the malicious control plane that was formed with the compromised compute nodes. This allows the attacker to manipulate or take control of the network with the malicious control plane. Integrating the role-based information to the security protocol can however allow such limited protection to securely protect the entire network as it can prevent the role abuse of the compute nodes to disguise itself as a controller. It also simplifies the entire protection mechanism as it can emphasize the security on the control plane alone.

The RB-IBC is not perfect in solving the security issues in SDN. It can mitigate this role abuse of the devices that a knowledgeable adversary can exploit. By integrating the role information into the key pairs, the adversary cannot use the stolen key pairs of any devices other than a controller to initiate a connection as a controller. Hence, the protection of the control plane is still intact as long as the controllers are well protected. Compromise of any data plane devices do not expose the control plane to the attacker.

Despite the advantages of RB-IBC, it also comes with some shortcomings. It is not able to revoke the any node without having the PKG being online. There is also a delay for the nodes to receive the revoked list of devices from the online PKG that the adversary can exploit. These are the two remaining issues to be resolved by RB-IBC.

In order to allow the revocation to take place swiftly, the identity information has to be unique. The Media Access Control (MAC) address is one of the best example. One might argue that the adversary can spoof the MAC address and disguise as one of the node within the network.However, even if the adversary is able to get the other devices within the network to initiate a connection, it cannot complete the handshake as it does not possess the private key needed to decrypt the messages.

Lastly, the performance degradation is expected just like any other security protocols. However, it should be much lower compared to TLS as the certificate exchanges are not required in IBC. Besides that, the overhead of the role information is minimal because the additional length of the information were offset by hashing to the same length.

However, in most SDN use cases, these are not actually the main concerns as they are mostly use in a closed/internal network. Even the roleabuse can be rare but possible and hence, we tried to minimize the risk of the network with the proposal of RB-IBC.

V. Conclusion

Our proposed RB-IBC is able to mitigate the role abuse of the node or the attacker that tries to use the stolen private key of the other nodes (even if it is not a controller) within the network and disguise as a new controller. This in turns allow the network switches to migrate in a more secure manner by knowing the migration target has the actual role of a controller.

Besides that, by combining the RBAC with IBC, it also simplifies the certificate management and the client; In this case, the network switches are not required to retrieve the role information from a management server as a disguised controller is not able to decrypt the message because it does not have the private key that was integrated with the role of a controller ($RBAC_{controller}$).

The communicating devices also do not require to exchange the certificate information because IBC can derive it from the identity information while TLS requires the certificate for the identity derivation. Hence, it will improve the performance as compared with TLS.

In a nutshell, the RB-IBC is able to mitigate the role abuse of the nodes even when their private keys are stolen. However, the protocol is not able to revoke such nodes until the devices obtain the latest revoked nodes from the PKG. This delay might be abused by the adversary but not for long as the devices will be obtaining this information after a preset interval by the network administrator or even immediately if the network allows.

REFERENCES

[1] J.H. Lam, S.G. Lee, H.J. Lee, et al. “Securing Distributed SDN with IBC”. 2015 Seventh IEEE International Conference on Ubiquitous and Future Networks (ICUFN), pp 921-925. July 2015.

[2] N. Mckeown, T. Anderson, H. Balakrishnan, et al. “OpenFlow”. ACM SIGCOMM Computer Communication Review, 38(2), pp 69-74. April 2008, New York, USA. doi:10.1145/1355734.1355746

[3] A. Shamir, “Identity-Based Cryptosystems and Signature Schemes”, Proceedings of CRYPTO ’84, Section I, pp 47-53, 1985, doi:10.1007/3-540-39568-7_5.

[4] Ryuichi Sakai, K. Ohgishi, Masao Kasahara, “Cryptosystems based on pairing”, Symposium on Cryptography and Information Security 2000 (SCIS 2000), Okinawa, Japan, Jan 26-28, 2000.

[5] Dan Boneh and Matthew Franklin, “Identity-Based Encryption from the Weil Pairing”, The Proceedings of 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-23, 2001. doi: 10.1007/3-540-44647-8_13.

[6] N.P. Smart, “An Identity based Authenticated Key Agreement Protocol based on the Weil Pairing”, Electronics Letters Volume 38, Issue 13, pp 630-632, 20 June 2002. doi:10.1049/el:20020387.

[7] Liqun Chen and Caroline Kudla, “Identity Based Authenticated key Agreement Protocols from Pairings”, The Proceedings of 16th IEEE Computer Security Foundations Workshop, 30 June - 2 July 2003, doi:10.1109/CSFW.2003.1212715.

[8] David F. Ferraiolo, Janet A. Cugini, and D. Richard Kuhn, "Role-based access control (RBAC): Features and motivations." The Proceedings of 11th Annual Computer Security Application Conference. pp 241-248. December 11, 1995.

[9] J.S. Park, R. Sandhu and G.J. Ahn, “Role-based Access Control on the Web”, ACM Transactions on Information and System Security (TISSEC), 4(1), pp 37-71, Feb 2001, New York, USA. doi: 10.1145/383775.383777

[10] D. Nali, C. Adams, A. Miri. “Using Mediated Identity-Based Cryptography to Support Role-Based Access Control”. International Conference on Information Security 2004. Lecture Notes in Computer Science (LNCS), 3225, pp 245-256, Springer, Berlin, Heidelberg. doi: 10.1007/978-3-540-30144-8_21

[11] S. Shin, V. Yegneswaran, P. Porras, et al. “AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks”. Proceedings of the 2013 ACM SIGSAC conference on Computer & Communications Security. pp. 413-424, 04-08 Nov, 2013, Berlin, Germany. doi: 10.1145/2508859.2516684

Authors

JunHuy Lam received the B.E. degree in computer engineering from the Multimedia University, Cyberjaya, Malaysia, in 2010, and the M.Sc. degree in ubiquitous IT from Dongseo University, Busan, South Korea, in 2012. In 2012, he joined GHL Systems Berhad, Kuala Lumpur, Malaysia as Software Engineer, and involved in the software development of the credit card and contact-less card payment systems. Since 2014, he joined the Department of Ubiquitous IT, Dongseo University as a PhD candidate. His current research interests include software defined network, algorithm, and network security.

Sang-Gon Lee received the B.S., M.S. and Ph.D. degrees in Electronics Engineering from Kyungpook National University, Korea, in 1986, 1988 and 2003, respectively Dr. Sang-Gon Lee joined the faculty of the Department of Electronic Communication at Changshin University, Changwon, Korea, in 1997 and is currently a Professor in the Division of Computer Engineering, Dongseo University. He is interested in cryptography, design and analysis of cryptographic protocols, network security, software defined network, block chain.

Vincentius Christian Andrianto received the B.S. degree in Electrical Engineering from Petra Christian University, Indonesia in 2015 and M.S. degree in Computer and Information Engineering from Dongseo University, Korea, in 2017. Mr. Andrianto is interested in Internet of Things, software defined network, network security, and artificial intelligence.