A System for Source Address Authentication and Anonymous Communication Based on SDN

Ming Li*, Ranran Shan and Yong Tang
School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, Sichuan, China
Email: limingdashu@163.com

Abstract. Based on the idea of SDN (Software Defined Network) management and control separation and the development convenience provided by the programmable interface, we conduct the research on the security during the communication security, and protection for anonymity: (1) For the problem of communication security, we design two source address authentication schemes to detect whether the source host is legal. (2) For the problem of anonymity protection, a series of changes of the IP address in data packets by intermediate SDN switches make it difficult for the attacker to obtain information from both parties in communication. And to ensure that the address generated after the execution of those changes can uniquely identify the current data stream, we proposed the Improved-MAGA algorithm based on MAGA (M-Address Generation Algorithm). (3) For the shortest path routing algorithm, which maybe result in hot-links problems, the measurement link bandwidth utilization is added as a weight vector in the routing process to solve the problem of excessive link load caused by the default routing algorithm.

1. Introduction
The network structure of TCP/IP is divided into 4 layers, and each layer must decapsulate the data message. And the step of encapsulating relevant information into IP header is flawed, for that, after the source host fills in its IP address into the source address field of its IP header, both the host and the router won’t check the authenticity of the source IP address, and later forwarding operations are executed only based on the destination IP address. Attackers can easily exploit this flaw. For example, an attacker fills in a fake source IP address and impersonates someone else to send out illegal messages, and then TCP/IP protocol stack forwards the illegal messages to the target host.

As a new type of network architecture, SDN was originally proposed by the Clean Slate research team at Stanford University [1], but its concept was formally proposed by Professor Mckeown in 2009 [2]. Its main idea is to separate the data forwarding function of the switch in the traditional network from the logic control function. In SDN, there is a data forwarding plane with a logic control function, which mainly processes data messages according to operation instructions from the control plane. Moreover, the data forwarding plane and the control plane communicate with each other through the OpenFlow [3] protocol.

2. Dynamic Routing Algorithm Based on Bandwidth Utilization
The design of the routing algorithm based on the dynamic selection of bandwidth utilization can meet the following two requirements.
2.1. Specify Bandwidth When Creating Networks
The algorithm first determines a path, and then sends a flow entry to all switches passing through the path, and configures a speed limit queue at the data outgoing port of the edge switch which determines the maximum bandwidth of the port egress and responses to initial demand. At this time, the bandwidth rate of the data stream must be less than or equal to the specified bandwidth requirement to output traffic.

2.2. Do Not Specify Bandwidth When Creating Networks
In such a network, bandwidth fluctuations and data flow are unstable.

The related important definitions involved in the routing algorithm based on dynamic selection of bandwidth utilization include the followings:

- Link bandwidth utilization: The value of link bandwidth utilization is equal to the ratio of the bandwidth used by the link between any two adjacent nodes to the total link bandwidth.
- Path bandwidth utilization: The value of path bandwidth utilization is equal to the maximum link bandwidth utilization, which in the path.

Let variable $\Delta t$ donates the measured time interval. Variable $U_i$ donates the average bandwidth of the i-th link, then the i-th link’s average bandwidth during the time $t$~$\Delta t$ is:

$$U_i(t + \Delta t) = \frac{E_i(t + \Delta t) - E_i(t)}{\Delta t}$$

(1)

$$L_i(t + \Delta t) = F_i(t) - U_i(t + \Delta t)$$

(2)

The path bandwidth utilization of each path is the highest utilization rate of a certain link bandwidth in this path, which is, $P = \text{Max (} u_l)$, and then find the path p from P with the lowest path bandwidth utilization from all s to d paths.

3. Research on Source Address Authentication

3.1. Get <MAC, IP, PORT> Triplet
The principle of binding triplet information based on SDN is shown in figure 1. When the SDN network is initialized, the controller first obtains all information of the MAC and IP address of the connected host and the switch port of the connected host, binds the obtained data in advance and stores it in the binding configuration table [4]. Then the controller configures the pre-processed triple binding rule <MAC, IP, PORT> in the filter rules of the switch.

![Figure 1. Bind <MAC, IP, PORT>](image-url)
3.2. Centralized Calculation of Path Filtering Scheme (CPF)
First, the controller obtains the topology information of the current network through the link discovery module and the topology management module. Then, the obtained topology information is calculated according to the routing algorithm and the IP and MAC change algorithm designed by the controller to obtain the \(<m_{\text{src}}_{\text{IP}}, m_{\text{dst}}_{\text{IP}}, \text{inPort, outPort}>\) quadruple filter rules, and store the information in the filter configuration table [5].

3.3. MAGA
SDN architecture can make data packets forwarding more flexible. The controller can install routing rules into switches in advance and switches can modify the data packet header to hide the real participants of the data flow to achieve anonymous communication.

When two or more flows are forwarded through the same port on the switch, an address conflict may occur, a route between two m-flows may conflict, or an m-flow and a non-m-flow, which will produce error messages. Figure 2 shows three address conflict scenarios.

(a) The packets’ addresses of the two flows f1 and f2 become the same on the same switch, as shown in figure 2a.
(b) The flow f1 packets’ address is modified to be the same as another flow f2 on the same switch, as shown in figure 2b.
(c) The packets’ addresses of the two flows f1 and f2 are the same before reaching the same switch, but the switch will not change the addresses of the two flows, as shown in figure 2c.

The root cause of the address conflict is that m-flow uses variable addresses during communication. Therefore, m-flow may occupy non-m-flow addresses, or two m-flows may be modified to the same address and used at the same time. In order to avoid the address conflicts mentioned above, a conflict avoidance mechanism MAGA is designed.

For example, a simple \(f(x, y, z)\) is constructed as follows:

\[
\begin{align*}
  f(x, y, z) &= \left[(x \oplus A_4)A_4 \right] \oplus \left[(x \oplus A_4)A_4 \right] \oplus \left[(y \oplus B_0)B_0 \right] \oplus \left[(y \oplus B_0)B_0 \right] \oplus \left[(z \oplus C_0)C_0 \right] \\
  &= (x \oplus A_4)A_4 \oplus (x \oplus A_4)A_4 \oplus (y \oplus B_0)B_0 \oplus (y \oplus B_0)B_0 \oplus (z \oplus C_0)C_0
\end{align*}
\] (3)

Then the inverse function of the variable z is obtained as follows:

\[
\begin{align*}
  f^{-1}(x, y, z) &= V \oplus \left[(x \oplus A_4)A_4 \right] \oplus \left[(x \oplus A_4)A_4 \right] \oplus \left[(y \oplus B_0)B_0 \right] \oplus \left[(y \oplus B_0)B_0 \right] \oplus \left[(z \oplus C_0)C_0 \right] \\
  &= (x \oplus A_4)A_4 \oplus (x \oplus A_4)A_4 \oplus (y \oplus B_0)B_0 \oplus (y \oplus B_0)B_0 \oplus (z \oplus C_0)C_0
\end{align*}
\] (4)

\(A_0, A_1, A_2, A_3, B_0, B_1, B_2, B_3, C_0, C_1\) are parameters, and these parameters may be different for different MNs to construct different hash functions.
3.4. Limitations and Improvements of MAGA

This paper analyzes some of the limitations of the MAGA [6] and summarizes the following points:

- In MAGA, the algorithm for determining routing information is the default Dijkstra routing algorithm that comes with the controller.
- In the MAGA algorithm, in order to avoid address conflicts after changing the address of the changing node, all data flows entering the SDN network have adopted the method of changing the address and adding MPLS labels, which will sacrifice part of the performance of the system. Therefore, in this paper, m-flow and non-m-flow are operated separately.

3.5. Improved-MAGA

The specific implementation steps of the improved MAGA algorithm are divided into the following 9 steps.

- SDN network initialization. SDN controller obtains the entire network topology and the addresses of all hosts, including IP addresses and MAC addresses, and saves a mapping table of all IP addresses and MAC addresses;
- Establish anonymous communication channels. First, the responder needs to register the service name with the SDN controller, and bind the service name to the managed IP and MAC address mapping table, and add the virtual IP and real corresponding to the real IP on the bound data Virtual MAC corresponding to MAC;
- Enter the monitoring state, the initiator sends the service name (or IP address) of the responder to the SDN controller. If the initiator specifies the IP address of the responder, it is non-m-flow. If the request specifies the service name, the initiator and the responder can be anonymous at the same time and perform the next step;
- Calculate the route. The controller calculates routing information through a routing algorithm based on the specified responder address or the responder address resolved from the service name, and selects two edge switches on the determined path for the m-flow data flow as two conversion nodes, which modify the IP address and MAC address in the data packet;
- Communicate anonymously. During communication with the responder, the initiator sends a network message to the ingress address or receives a network message from the ingress address;
- After the anonymous channel is established, SDN controller can update the forwarding path and the conversion node of the anonymous channel;
- Close the anonymous channel. The initiator (or responder) sends a shutdown notification to SDN controller, and the controller deletes the anonymous channel information.

4. Design and Implementation of the System

The overall architecture of SDN-based source address authentication and anonymous communication system is shown in figure 3.

![Figure 3. Overall system architecture diagram.](image)
After the controller realizes the design of the above-mentioned modules, when it receives a packet for anonymous communication, it will transfer the packet to the application function module for processing, and then configure the corresponding forwarding rules to the switch according to the processing result.

5. Simulation and Test

5.1. Simulation Environment

Combine the tools used in this paper to build a schematic diagram of the experimental environment shown in figure 4. We use floodlight [6] as the SDN controller.

![Figure 4. Schematic diagram of the experimental environment.](image)

5.2. Function Test

After successful communication, enter the following command to check the configured flow entry information on the switch. The analysis shows that only S1, S3, and S4 realize the deployment of flow entries in all switches.

When using the routing algorithm based on path bandwidth utilization, after entering the command to view the switch. From the flow entry information delivered, it is found that only switches S1, S2, and S4 are successfully installed.

After adding the source address authentication module designed in this article, use Wireshark to capture the link on H4-H3. The result is shown in figure 5. It can be seen from the figure that no forged data packets with a source address of 10.0.0.1 and a destination address of 10.0.0.3 have been caught on the H4-H3 link. It can be seen that the source address authentication module has filtered out the data packets sent by H4 to falsify the IP address of H1 to H3. The source address authentication module designed in this paper has been effectively verified, and this part of the test is completed.

![Figure 5. H4-H3 link packet capture result diagram.](image)
Therefore, it can be seen from the flow entry entries issued on the switch that the Improved-MAGA algorithm used in the anonymous communication process changes the IP and MAC addresses in the data packets, and successfully deploys the flow entries on SDN switches. At this point, the entire system test is over.

6. Conclusion
This paper takes an anonymous communication algorithm based on SDN environment as the core and assists with source address authentication technology to design a system based on SDN for source address authentication and anonymous communication. This system aims to prevent and detect the IP spoofing of the source host in the network, and to hide the real participants of the communication, complete the anonymous communication, and realize the non-association of the session. The main research of this paper is as follows: (1) Two schemes to achieve source address authentication design. (2) A dynamic routing algorithm based on link bandwidth is proposed. (3) It proposes the implementation of changing IP and MAC algorithms for anonymous data streams. (4) Bind the source IP and destination IP to the non-anonymous data stream.

Acknowledgments
This work was supported by Science & Technology Department of Sichuan Province Project No. 2019ZDYF0111.

References
[1] Stanford University Clean Slate Program 2006 http://cleanslate.stanford.edu/.
[2] McKeown N 2009 Software-defined networking Proc. of the INFOCOM Keynote http://infocom2009.ieee-infocom.org/keynotes.html.
[3] Mckeown N, Anderson T, Balakrishnan H, et al. 2008 OpenFlow: enabling innovation in campus networks ACM SIGCOMM Computer Communication Review 38 (2) 44-95.
[4] Gao W, Sun Y, Fu Q, et al. 2018 ARP poisoning prevention in internet of things 2018 9th International Conference on Information Technology in Medicine and Education (ITME) (IEEE Computer Society).
[5] Zhu T, Feng D, Hua Y, et al. 2016 MIC: An efficient anonymous communication system in data center networks International Conference on Parallel Processing (IEEE).
[6] Floodlight 2014 http://www.projectfloodlight.org/floodlight/.