TOWARDS A SDN-BASED INTEGRATED ARCHITECTURE FOR MITIGATING IP SPOOFING ATTACK

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

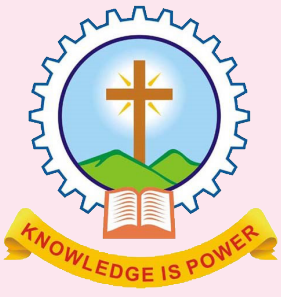
COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

ABHINAV T K



Department of Computer Science & Engineering
Mar Athanasius College Of Engineering Kothamangalam

TOWARDS A SDN-BASED INTEGRATED ARCHITECTURE FOR MITIGATING IP SPOOFING ATTACK

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

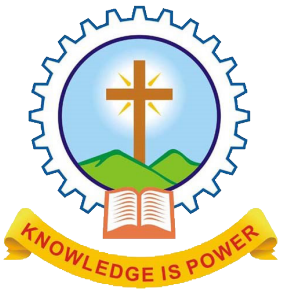
COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

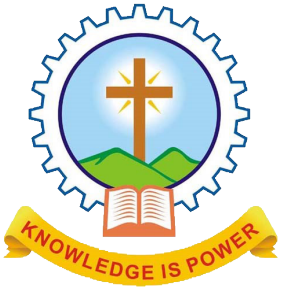
Submitted By

ABHINAV T K



Department of Computer Science & Engineering
Mar Athanasius College Of Engineering Kothamangalam

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAR ATHANASIUS COLLEGE OF ENGINEERING
KOTHAMANGALAM



CERTIFICATE

This is to certify that the report entitled **TOWARDS A SDN-BASED INTEGRATED ARCHITECTURE FOR MITIGATING IP SPOOFING ATTACK** *submitted by Mr. ABHINAV TK, Reg. No. MAC15CS003 towards partial fulfilment of the requirement for the award of Degree of Bachelor of Technology in Computer science and Engineering Engineering from APJ Abdul Kalam Technological University for the year December 2018 is a bonafide record of the work carried out by him under our supervision and guidance.*

.....	.....	.....
Prof. Joby George	**Prof. Neethu Subash**	**Dr.Surekha Mariam Varghese**
Faculty Guide	***Faculty Guide***	***Head of the Department***

Date:

Dept. Seal

ACKNOWLEDGEMENT

First and foremost, I sincerely thank the God Almighty for his grace for the successful and timely completion of the seminar.

I express my sincere gratitude and thanks to Dr. Solly George, Principal and Dr. Surekha Mariam Varghese, Head Of the Department for providing the necessary facilities and their encouragement and support.

I owe special thanks to the staff-in-charge Prof. Joby George , Prof. Neetha Subash and Prof. Joby Anu Mathew for their corrections, suggestions and sincere efforts to co-ordinate the seminar under a tight schedule.

I express my sincere thanks to staff members in the Department of Computer Science and Engineering who have taken sincere efforts in helping me to conduct this seminar.

Finally, I would like to acknowledge the heartfelt efforts, comments, criticisms, co-operation and tremendous support given to me by my dear friends during the preparation of the seminar and also during the presentation without whose support this work would have been all the more difficult to accomplish.

ABSTRACT

Current Internet packet delivery only relies on packet's destination IP address and forwarding devices neglect the validation of packet's IP source address, it makes attackers can leverage this flaw to launch attacks with forged IP source address so as to meet their vicious purposes and avoid to be tracked. The solutions to mitigate this threat and enhance Internet accountability are faced with some issues hard to cope with, like low filtering rates, high deployment cost. With the central control and edge response pattern of software defined networking (SDN) architecture, an integrated IP source address validation architecture covering both intra- and inter-domain areas and effectively lower SDN devices deployment cost, while achieve desirable control granularities is proposed. Within autonomous system (AS), it relies on an SDN incremental deployment scheme which can achieve IP prefix level validation granularity with minimum SDN devices deployment. While among ASes, it sets up border server and establishes a vouch mechanism between allied ASes for signing outbound packets so as to achieve AS-level validation granularity. The intra-domain scheme can get beyond 90% filtering rates with only 10% deployment in average, while the inter-domain scheme can get high filtering rates with low system cost and less storage usage.

Contents

List of Figures

LIST OF ABBREVIATION

SDN	Software Defined Networking
ISAVA	Integrated IP source address validation architecture
AS	Autonomous system
ACL	Access Control List
SAVSH	Source Address Validation for SDN Hybrid network
PSVM	Packet signing and verification
TAENP	Trust Alliance Establishment Negotiation protocol
NOS	Network Operation System
AM	Adjacency matrix
MTU	Maximum transmission unit
uRPF	Unicast reverse path forwarding

Introduction

IP source address spoofing or IP spoofing attack, it refers to attackers release packets with forged IP source addresses so that they can conceal their real identities and launch attacks, e.g., reflect network traffics to flood victim hosts. Once suffering such attack, it is hard for victim to trace back to perpetrators and identify their real identities, which severely compromises Internet accountability indeed. From the perspective of technique, IP spoofing threat is derived from the design that Internet packet forwarding in routers only relies on packet's destination IP address, but neglects the validation of packet's IP source address to verify sender authenticity. Taking this vulnerability, attackers can launch serious attacks against specified targets, and as a matter of fact, most of attack directly related with this volubility, i.e., TCP-SYN [1] flooding , DDoS and Smurf [3].

Despite anti-IP spoofing has been studied extensively in the past decade, however, feasible and integrated solutions that cover both of intra-domain and inter-domain scopes still under the way of research. As a matter of fact, the IP spoofing phenomena in Internet did not improve much in the last few years. According to the Center for Applied Internet Data Analysis (CAIDA)'s statistics , by end of October 2017, the spoofable address space, prefix, and AS have up to 26.7%, 33.6% and 34.1%, respectively. Also, the global cyber-security event recording proves that the number of IP spoofing and related attacks has sharply increased in last few years . More than that, the annual report regarding Chinese Internet security status confirms that the new IP spoofing-related attack means, such as Distributed Reflection Denial of Service (DRDoS), DNS request reflection, Network Time Protocol (NTP) synchronization reflection, are thrived and abused to target at many high-value websites.

In order to mitigate this threat, many intra-domain or inter-domain solutions have been proposed. The former mainly solve the issue within Autonomous System (AS), while the latter cover the area between ASes. In Detail, solutions within domain for anti-IP-spoofing can be categorized into packet filtering, address encryption and protocol modification. Packet filtering is a common practice for anti-spoofing in many domain networks, e.g., configuring Access Control List (ACL) rules onto intra-domain routers or switches so that they can drop packets with unexpected/illegal IP source addresses or prefixes. Inter-domain solutions mainly concentrate on three directions: end-based, end-to-end and path-based filtering. End based filtering method specifies AS border device drops the inbound packets with source addresses belong to the local AS and the outbound packets whose source addresses belong to other domain. End-to-end filtering idea establishes a connection between two ASes' border devices and ignores ASes

along the path. And each source-destination pair ASes shares a secret key so that the source AS can tag the packets header to destination AS and the destination AS can verify the packets authenticity. Path based filtering proposal verify the packets by the paths they flow through as the attacker usually cannot manipulate the forwarding path.

As Software Defined Networking (SDN) owns the capacity of global topological view and central control pattern, it has gained much attention from both academic and industrial communities in recent years. With SDN, traditional ACL can be interpreted by flow rules in SDN-supported switches (or SDN switches), which can be issued by logically centralized controllers based on network real-time situations. Thus, SDN offers us an opportunity to solve the IP spoofing issue and overcome defections existing in traditional solutions.

Inspired by this motivation, an SDN-based Integrated IP Source Address Validation Architecture (ISAVA) which can cover both intra and inter-domain areas and effectively lower SDN devices deployment cost but achieve desirable control granularities is proposed here. Specifically, within Autonomous System (AS), ISAVA propose an SDN incremental deployment plan which can achieve IP prefix (subnet)-level validation granularity with minimum SDN devices deployment. Thus the most exciting advantage of this plan is that it can gain the maximal IP source address validation effect by deploying the minimal SDN switches into traditional networks, which can keep existing network assets to the maximal degree that can promote system incremental deployment.

Among ASes, ISAVA sets up SDN controller in each AS's border and establishes a vouch mechanism between allied AS controllers for outbound packets so as to achieve AS-level validation granularity. What's more, since the topological information of each AS is partly visible to other allied ASes, and the packet's original source address is replaced by its SDN border controller's IP address, our inter-domain solution also can get trade-off between privacy and security. Finally, through our conducted experiments, we confirm that ISAVA intra-domain scheme can get at least 90% filtering rates with only 10% deployment in average, while ISAVA inter-domain scheme can get high filtering rates with low system cost and less storage usage.

Related Works

The works on mitigating the IP spoofing problem can be broadly classified to intra-domain and inter-domain. Some of the related works that are available for both these domains are discussed here

2.1 Intra-domain solutions

Intra-domain solutions try to solve the issue within an Autonomous System (AS). Solutions within a domain for anti-IP-spoofing can be categorized as follows

2.1.1 IP source address filtering

Depending on the acting positions, filtering solutions can be divided into three types: ingress-, egress- and router-based filtering, which checks packet legitimacy in router's ingress ports, egress ports and internal modules, respectively. The unicast Reverse Path Forwarding (uRPF) [**?**] is a deployable ingress filtering solution, which was advocated by Cisco and applied to its products. When uRPF function is enabled, for every packet, router's ingress port first looks up its Forwarding Information Base (FIB) with packet's IP source address, so that it can verify the packet's legality based on whether the forwarding port matches the current ingress port or not. However, uRPF is proprietary mechanism and it is hard to cope with the situations when both the victim and the attacker are in the same direction, routing asymmetry and etc.

2.1.2 IP source address encryption

In order to authenticate communication correspondents, some researchers give their solutions from the angle of replacing the IP source address with the encrypted one. For example, Cryptographically Generated Addresses (CGA) and Accountable Internet Protocol (AIP) encrypt IP source address with the asymmetric key cryptography so that keys sharing both ends can verify each other. But such designs need extra secure key agreement protocols because key generation and public-key distribution are accomplished by individual hosts without Certificate Authority (CA), which is non-suitable for large-scale networks. To address this issue, TrueIP takes IP source address as the public key and utilizes the Identity Based Cryptography (IBC) to produce the private key, so that correspondents can verify the authenticity of each other directly without public-key acquirements.

2.1.3 Protocol and host-stack redesign

SPM and Base solve the IP-spoofing problem by leveraging some rarely used fields (e.g., ToS) in the IP header and replacing them with customized tags. But this design may disturb other special applications (e.g., Quality of Service). Host IdentityProtocol (HIP) sets up a new layer named Host Identity (HI) in the middle of IP and transportation layers. It obtains reliable host identities through asymmetrically encrypting the HI data. But in the meantime, it complicates system implementation as it has to modify client's host-stack. More importantly, it needs to install a DNS-like system to resolve the mapping relationship between HI and IP addresses. Therefore, the largest overhead comes from their implementation and deployment.

2.1.4 SDN-based source address validation

Virtual source Address Validation Edge (VAVE) protects users under the SAVI switch being spoofed by other users within the same domain. To do so, VAVE establishes an IP source address protection zone comprising by all of Layer3 (L3) OpenFlow switches (OF-switch) and L2 SAVI switches. And the legacy network assets are posited outside this zone. Thus, any flows originated from the legacy switches and passed through this zone will be redirected to the controller to verify their IP source addresses authenticity, except that matching rules explicitly exist in the boundaries of the OF-switch. O-CPF is anti-IP-spoofing with the granularity of subnet prefixes in intra-domain. It leverages the SDN controller to compute the forwarding path for each prefix pair and tries to upgrade domain routers to accommodate OpenFlow specification. By doing this, the OF-routers can check the validity for each packet passing byand drop illegal ones via issued rules from controller.

2.2 Inter-domain solutions

Intra-domain solutions try to solve the issue between Autonomous Systems (AS). Solutions between domains for anti-IP-spoofing can be categorized as follows

2.2.1 End-based source address filtering

The main idea of end-based filtering method is to drops the inbound packets whose IP source addresses belong to the local domain or the IP destination addresses do not belong to local. So it is usually used in domain boundary devices. CatchIt [6] proposes a way of validating inter-domain packets authenticity by enabling inter-domain routing system cooperation via an intelligent routing choice notification mechanism. However, CatchIt still has the

implementation issue, thus it is hard to deploy and that hampers its promotion as well.

2.2.2 Path-based source address filtering

The Path-based filtering method verifies the forwarding path to identify the spoofing packets since attackers may modify packets' IP source addresses but they cannot manipulate packets' forwarding path in general. There are some work concentrating on path-based verification, like DPF and IDPF . DPF associates each source AS to a set of valid upstream ASes so that it can validate packets attribution by check their incoming AS. Theoretically, DPF can be very effective in inter-domain IP spoofing scenario, and it is extensively studied by other path based defense proposals. However, its filtering accuracy relies on the filtering sets' being complete and accurate. By overcoming DPF's drawback, The IDPF constructs filters set by inferring feasible paths for every source AS. Detailly, it constructs feasible path set by analyzing routing entries or route export rules.

2.2.3 End-to-end (E2E)-based source address validation

The E2E-based filtering is on the basis of corporation between two ends (ASes) or two ASes composed alliance. Each alliance pair shares a secret key or establishes one particular communication protocol. Some studies, like SAVE , SPM , Passport [4] , DIA and APPA , are the typical representatives. Spoofing Prevention Method (SPM) [7] associates a unique temporal key with every AS pair and add tags into packets travel between the two ASes, so that receiver AS border routers can verify packets' authenticity and remove tags. DIA forms inter-domain anti-spoofing alliances. By adding and verifying MAC message in the packet, so the ASes belong to the party can identity the fake packets. APPA is a signature-based IP source address prevention method. It takes advantage of an automatically synchronizing state machine to exchange generate secrete state code (password) in a fixed time interval, and generate signatures for outbound packets based on current password.

The Proposed System

3.1 Formalized description

Assuming in a domain network named D, the network topology can be denoted as: $Topo_D = G(V, E)$, where V is the node/router set and E is the links between routers/edge collection. Also, we use the set $HOST_D = \{H_1, H_2, ...H_S\}$, $USER_D = \{U_1, U_2, ..U_Q\}$ and $IP_D = \{IP_1, IP_2, ..IP_M\}$ to represent the collection of host, user and IP address in domain D, where S, Q and M are the total number of hosts, number of users and number of IP address, respectively. Further, according to the IP packet's format, we can describe a IP packet as $packet = \{version, length, IP_{src}, IP_{dst}, data..\}$, items in which represent packet fields, such as packet version, length, IP source address, IP destination address, upper layer data and etc. Thus, all packet collection that source from domain D can be denoted as $Packet_D = \{packet | IP_{src} \in IP_D\}$. In order to get the packet reliability, we believe that system needs to achieve both IP source address credibility and user credibility. The former one refers to every host has its own IP address or vice versa, which can be describe as $Host_D \leftrightarrow IP_D$. While the latter one states each packet's IP source address should be consistent with packet's true sender, which can be expressed as $IP_D \leftrightarrow User_D$. Thus, packet reliability can be denoted as $Host_D \leftrightarrow IP_D \leftrightarrow User_D$, which means elements in the end host set, IP address set and user identity set has a unique mapping relationship.

3.2 Threat model

Based on the attacker's location, we category the IP spoofing scenarios as three types: host-based attack, router-based attack and flow-based attack, as shown in Fig 3.1.

3.2.1 Host-based attack

Attackers forge packets with specified or random IP source address in IP header so as to launch attack and shift responsibility to other innocent people. This type of attack is very common in current Internet and it evolves a lot of versions till now, such as DDoS, reflected amplification DDoS and etc.

3.2.2 Router-based attacks

Attackers may leverage routers' or key routing devices' vulnerability to take over their control privilege or even modify forwarding function so that attackers can pollute flowing

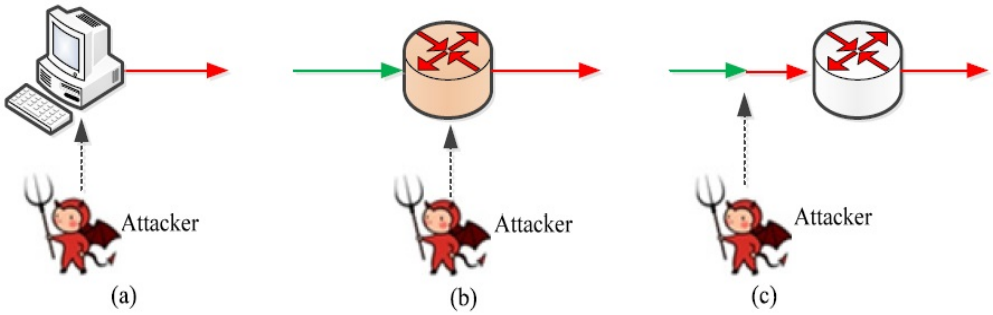


Fig. 3.1: IP source address spoofing scenarios. (a) Host-based attack. (b) Router-based attack. (c) Flow-based attack.

through packets with false IP source address. Compare to the host-based attack, this kind of attack is much harder since network admins will pose enhanced security protection to these devices.

3.2.3 Flow-based attacks

This kind of attack also knows as man-in-the middle attack, which refers to attackers posit half-way of packets flow through and conquer some key routing devices (e.g., wireless access point) so that they can capture, alternate and then replay them with forged IP source address to meet their vicious purpose.

Considering last two attack tricks are relative rare, we mainly consider the first threat only.

Autonomous system

On the Internet, an autonomous system (AS) is the unit of router policy, either a single network or a group of networks that is controlled by a common network administrator (or group of administrators) on behalf of a single administrative entity (such as a university, a business enterprise, or a business division). An autonomous system is also sometimes referred to as a routing domain. An autonomous system is assigned a globally unique number, sometimes called an Autonomous System Number (ASN).

Networks within an autonomous system communicate routing information to each other using an Interior Gateway Protocol (IGP). An autonomous system shares routing information with other autonomous systems using the Border Gateway Protocol (BGP). Previously, the Exterior Gateway Protocol (EGP) was used. In the future, the BGP is expected to be replaced with the OSI Inter-Domain Routing Protocol (IDRP).

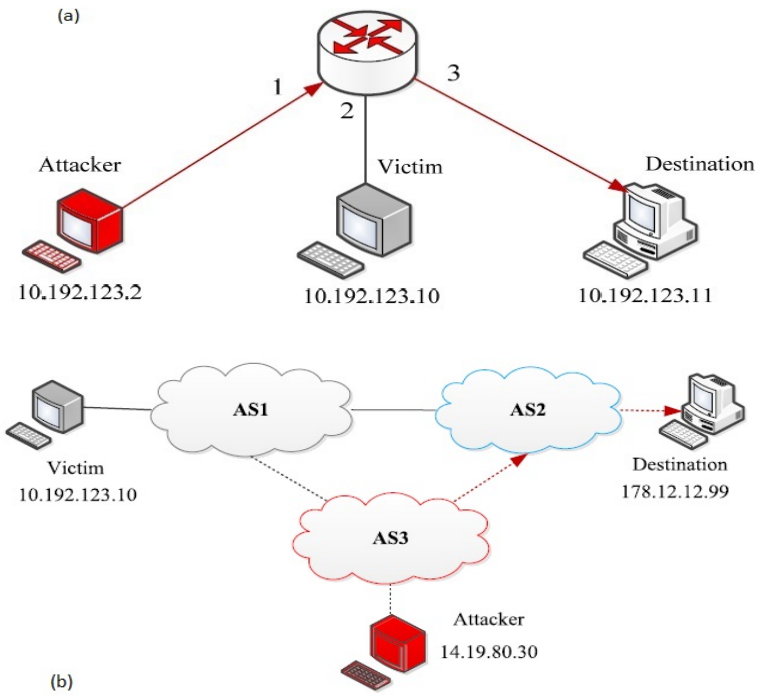


Fig. 3.2: The illustration of (a) intra-domain (b) inter-domain spoofing attack scenario.

3.3 Attack-scenarios

According to the locations of spoofing source host and spoofing packets' destination, we summary the two attack scenarios that are intra-domain and inter-domain spoofing.

3.3.1 Intra-domain spoofing

Intra-domain spoofing means both attacker and destination host are in the same domain, so the checkpoint in the domain border cannot effect and filter forged packets. Taking the scenario in Fig 3.2 (a) as example, the spoofing host 10.192.123.2 pretends to be the host 10.192.123.10 and conduct an attack to victim host 10.192.123.11. If the router has not deployed any anti-spoofing measures, the attack could be harmed to destination host and pretending host both. Thus, this threat reminds us that anti-spoofing measures have to impose to intra-domain area, instead of domain border only.

3.3.2 Inter-domain spoofing

As various management policies exist in different ASes and routing flapping phenomenon happens occasionally between ASes, attacker could posit in any ASes and launch

attack without traceback risk, which indicates deploying anti-IP-spoofing solution in inter-domain area is much difficulty than intra-domain area. For example, as Fig. 3.2 (b) depicted, the spoofing host 14.19.80.30 in AS3 can emit packets with victim's address 10.192.123.10 and attack host 178.12.12.99 in AS2. Even worse, if attacker and victim are in the same direction, e.g., BGP routing flapping makes AS 3 needs to traverse AS 1 to reach AS 2, it will hard to distinguish normal flows from vicious flows in the destination AS or end-host.

SDN- based Integrated IP Source Address Validation Architecture (ISAVA) with the help of SDN's centralized control capability tries to realize the goal of mitigating IP Spoofing attacks. It can cover both intra- and inter-domain dimensions and effectively lower SDN devices deployment cost but owns desirable control granularities. Specifically, within AS area, ISAVA supports an SDN incremental deployment scheme which can achieve IP prefix (subnet)-level packet validation granularity with minimum SDN devices deployment. While among ASes, our architecture provides a packet signing and verification mechanism to achieve AS-level packet validation granularity. As Fig.3.5 shows, ISAVA relies on SDN controller in each AS that acts as a center intelligent brain to control all information and guides the inter-domain modules (IDM) and intra-domain module (DM) to meet requirements. The two part of modules work together and response to domain area and inter-domain area packet verification, respectively. Specifically, the IDM communicates with peer module to exchange key information, e.g., encryption key, topology information, and then guides SDN-enabled AS border device to sign outbound packets or verify packets' sig- natures for allied ASes. As DM, it computes intra-domain checkpoints node based on real-time substrate network topology, then generates filtering rules and distributes onto these checkpoint nodes so as to filtering spoofing packets within domain.

3.3.3 Software defined networking

Software-defined networking (SDN) is an architecture that aims to make networks agile and flexible. The goal of SDN is to improve network control by enabling enterprises and service providers to respond quickly to changing business requirements.

In a software-defined network, a network engineer or administrator can shape traffic from a centralized control console without having to touch individual switches in the network. The centralized SDN controller directs the switches to deliver network services wherever they're needed, regardless of the specific connections between a server and devices.

This process is a move away from traditional network architecture, in which individual network devices make traffic decisions based on their configured routing tables.

SDN architecture

A typical representation of SDN architecture comprises three layers: the application layer, the control layer and the infrastructure layer.

The application layer, not surprisingly, contains the typical network applications or functions organizations use, which can include intrusion detection systems, load balancing or firewalls. Where a traditional network would use a specialized appliance, such as a firewall or load balancer, a software-defined network replaces the appliance with an application that uses the controller to manage data plane behavior. The control layer represents the centralized SDN controller software that acts as the brain of the software-defined network. This controller resides on a server and manages policies and the flow of traffic throughout the network.

The infrastructure layer is made up of the physical switches in the network. These three layers communicate using respective northbound and southbound application programming interfaces (APIs). For example, applications talk to the controller through its northbound interface, while the controller and switches communicate using southbound interfaces, such as OpenFlow although other protocols exist.

How SDN works

SDN encompasses several types of technologies, including functional separation, network virtualization and automation through programmability.

Originally, SDN technology focused solely on separation of the network control plane from the data plane. While the control plane makes decisions about how packets should flow through the network, the data plane actually moves packets from place to place.

In a classic SDN scenario, a packet arrives at a network switch, and rules built into the switch's proprietary firmware tell the switch where to forward the packet. These packet-handling rules are sent to the switch from the centralized controller.

The switch also known as a data plane device queries the controller for guidance as needed, and it provides the controller with information about traffic it handles. The switch sends every packet going to the same destination along the same path and treats all the packets the exact same way.

Software-defined networking uses an operation mode that is sometimes called adaptive or dynamic, in which a switch issues a route request to a controller for a packet that does not have a specific route. This process is separate from adaptive routing, which issues route requests through routers and algorithms based on the network topology, not through a controller.

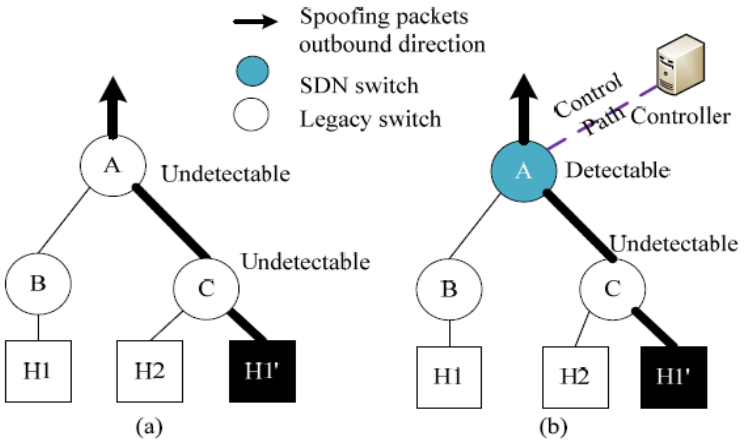


Fig. 3.3: The illustration of (a) intra-domain (b) inter-domain spoofing attack scenario.

3.4 Intra-domain solution

SAVSH (Source Address Validation for SDN Hybrid network) [8] is our intra-domain proposal for filtering spoofing packets within domain. Its aim is to prevent the packets with forged IP source address to leave out domain or attack hosts within domain. As SDN technology has achieved great success and accepted by lots of networks, now most networks has partially deployed or considered to deploy SDN devices to meet their diversified purposes. Under such circumstances, the goal of SAVSH is to take advantage of SDN patterns to maximally filter spoofing packets but with minimal SDN devices deployment. In other words, SAVSH aims to obtain the best trade-off between filtering accuracy and deployment overhead. As illustrated in the topology of Fig.3.3 (a), allthe nodes in the unprotected tradition network are unable to detect the spoofing flows originated by vicious host H1' which spoofs legitimate host H1's IP source address. With SAVSH design in Fig. 3.3 (b), we replace node A with a SDN device and takes it as a checkpoint to perform IP address filtering function.

Certainly, the node A still needs to be deployed some rules to perform filtering function. The rules are defined like pair hImport, SIP, DIP, Actioni. The "Import" item in the pair states the device port through which packets enters the device, then the SIP and DIP items are the source address and destination address separately, and the last item Action could be output(forward to appropriate port), drop and other options.

Although the above scenario is not complex, the main challenges come from three aspects:

(1) locate deployment nodes (checkpoints) and prioritize them

(2) design controller application to distribute appropriate rules onto these SDN nodes

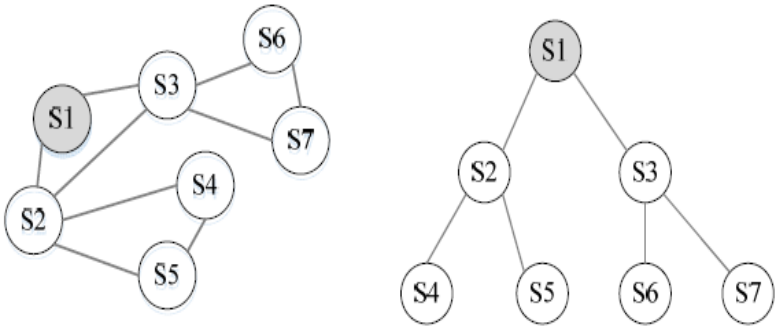


Fig. 3.4: Export based sink tree conversion (assuming S1 as the border router).

(3) adapt to network dynamics.

3.4.1 Converting topology into sink-tree

In order to accurately find the deployment nodes, we first need to convert complex intra-domain network topology into a simple export-based sink-tree, which takes the domain border router as the root and other L3 switches/routers as nodes. To do that, we assume that: (1) multiple links (e.g., port trunk) bet- ween two nodes are treated as one; (2) we do not take link bandwidth or quality into consideration (we argue that this assumption is reasonable since most domain networks usually take hop-count as link quality. Otherwise, we can assign links with corresponding weights in the following topological matrix); (3) we only focus on the single-homing scenarios (Actually, for multi-homing cases, we can take the nearest concentration border router as the root node to apply our proposal). In such sink-tree, each leaf node can follow the shortest path to reach the root and intra-domain other nodes. Next we treat the initial topology as a directed acyclic graph (DAG) with the border router as root node. Then we can get a $N * N$ (N is the total number of all nodes) adjacent topological matrix according to the link connection relationship between nodes. The value of the matrix can be assigned as the following:

Eventually, we can take this matrix as the parameter of the Dijkstra algorithm to shape this tree, and Fig.3.4 shows an example of this idea.

Dijksta algorithm

Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph.For a given source node in the graph, the algorithm finds the shortest path between that node and every other. It can also be used for finding the shortest paths from a single node to a single destination node by stopping the algorithm once the shortest path to the destination node has been determined. For example, if the nodes of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities. As a result, the shortest path algorithm is widely used in network routing protocols, most notably IS-IS (Intermediate System to Intermediate System) and Open Shortest Path First (OSPF). It is also employed as a subroutine in other algorithms such as Johnson's.

3.4.2 Locating SDN deployment nodes

Technically, border routers and all of the L3 access switches deployment can address the IP source address spoofing issue. However, considering most networks are still traditional net- works with legacy devices, the biggest concern of them is how to lower down the front-end investment and achieve this anti-spoofing purpose. Thus, within the legacy and SDN device hybrid network, minimizing SDN device deployment ratio but satisfying the desirable IP prefix-level anti-spoofing coverage ratio in the same time becomes our optimal goal.

SAVSH first introduces the notations pc_i which repre- sents the prefixes collection issued by node i and its subtree nodes, while pc_{all} notates the collection of all prefixes within domain. Once one node is replaced by an SDN switch, its utility can be expressed by the number of prefix pairs that transit through the node, which indicates all possible paths from source prefix to destination. Thus this utility not only includes the valid prefix pairs within the subtree with root node i, but also contains the prefix pair combinations inside and outside this subtree. Nevertheless, the whole formulation should meet user's predefined requirement λ, and this constraint expresses the ratio of unspoofable prefix pair as a whole.

$$
\begin{aligned}
& & \forall pc_s, pc_t \in Child[i], s \neq t : u_i = pc_s \cdot pc_t + pc_i \cdot \overline{pc_i} \\
min\ \ \alpha = \frac{\Sigma_{i=1}^{N} \sigma_i}{N} & & \forall pc_s, pc_t \in V, s \neq t : \lambda = \frac{\Sigma_{i=1}^{N} distinct(\mu_i \cdot \sigma_i)}{pc_s \cdot pc_t} \\
& & \alpha \in [0,1], \lambda \in [0,1]
\end{aligned}
\tag{3.1}
$$

However, such optimization goal with multiple constraints is a problem of integral linear

Notation	Meaning
AM(v,e)	Matrix,the adjacency matrix of topology with node set V and link set E
ST(v,e')	Matrix, sink-tree converted by topology matrix, and the nodes are indexed from 1
N	Integer,total number of nodes, equal to —V—
pc_i	Array, set of IP prefixes covered by node i (set of prefixes in the subtree with root node i, which includes all IP prefixes in this subtree and node i
pc_{all}	Array, total set ofIP prefixes within domain
σ_i	Binary, indicates whether node i is an SDN node or not
child[i]	Integer Array, set of child nodes with root node i
β_i	Ratio,proportion of unspoofable IP prefix pair when node i is an SDN node
α	Ratio, proportion of SDN nodes in all nodes
u_i	Integer, utility/prefix pair that can be checked by node i
λ	Ratio, requirement of unspoofable IP prefix rate in total
distinct()	Function, used to eliminate overlapped utilities when multiples nodes are selected for deployment

Table 3.1: Key notations in SASSH formulation.

Algorithm 1 SDN Deployment Nodes Selection Algorithm

Input AM, N*N topology adjacent matrix;
Output α , proportion of SDN nodes in all nodes;

1: **for** i = 1 to N **do**
2: **for** j=1 to N **do**
3: $u_{all} += ||pc_i||.||pc_j||$
4: **end for**
5: **end for**
6: **for** i=1 to N **do**
$\beta_i = u_i/u_{all}$
7: **end for**
8: $sort(\beta)$
9: **for** $i = 1 to N \lambda_{temp} < \lambda$ **do**
$\lambda_{temp} += Distinct(\beta_i)$
10: **end for**
11: $\alpha = i/N$
12: **return** α

programming, which is proved to be an NP-hard problem that cannot be solved in a mathematic way. Alternatively, we consider adopting a heuristic algorithm to locate SDN nodes, whose details are shown in Algorithm 1.With this algorithm, we first calculate the utility of prefix pair coverage and sort them in a decreased order for each node. Then we sum and eliminate the overlapped utilities from the first node to the last one until the utility requirement is satisfied.

3.4.3 Distributing filtering rule generation

After we locate the SDN nodes and replace them with SDN switches, rules for each individual SDN switch should be gen- erated from the controller and deployed onto them. Controller generates corresponding rules for each SDN node according to following three steps: (1) Relying on the generated sink- tree, SAVSH sorts out all legal prefixes in all of its downlink ports and aggregates prefixes in the same port as much as possible; (2) System organizes all possible prefix pairs between these prefixes in different downlink ports, and then forms corresponding forwarding rules; (3) Besides, system needs to produce forbidden rules to block illegal prefix pair to get through. Thus once spoofing packets reach these SDN checkpoints, they will be matched with these defined rules and executed by related actions.

3.4.4 Coping with network dynamics

How to deal with network dynamics is an important issue that matters solution's success, since topology changes would affect the shape of the sink-tree and the rule for SDN nodes. Unfortunately, sink-tree reshaping and rule recalculation will incur relative large latency than other procedures. To address this problem, we take the proactive and reactive combined way to cope with it. That is, for one link or one node failure situation, the system calculates new tree and store related rules into database in advance, so that system can directly distribute them if one of such cases happens. While for the rest of situations, system has to recalculate in time because multiple links and nodes failure combined situations are too complex to simulate. As the issue of rules redistribution would cause packets loss in the air, it is beyond the agenda of this paper and many studies (e.g., zUpdate) have focused on this issue.

3.5 Inter-domain solution

As we depicted in the Fig.3.5, the inter-domain module mainly consists four components. From above to top they are: the trust alliance establishment negotiation protocol (TAENP), east-west bridge, keyshare and packet sign and validation mechanism (PSVM).

Given each pair of SDN controllers, TAENP responses to communicate with them to exchange system fundamental information, such as peer identity verification, domain IP address list, leader AS election in each pair AS and etc. Based on TAENP, east-west bridge can exchange domain abstract network view with peer ASes so as to meet high-level requirement, e.g., path-based packet verification. Then the keyshare component has two very important functions, time synchronization and encryption key exchange with fixed intervals between allied ASes. Lastly, the PSVM relies on shared keys to tag packets and forward to allied ASes, or verify legitimacy and remove tags for the packets from allied ASes as well.

3.5.1 Trust alliance establishment negotiation protocol

TAENP is use to establish alliance relationship and nego- tiation some information between SDN controllers in differ- ent allied AS. In the first beginning, it will connect peer AS controller and identify peer's identity. When authenticated each other, two peers will form a pair and they will exchange some key information, for example, AS number, range list of domain IP address, for other component to perform their function. Besides that, they will elect a leader for initiatively launch connection in the subsequent interactions. The follow- ing simple algorithm decides how to select the leader in a pair.

$$Peer_{leader} = \begin{cases} max(AS1, AS2) & \textit{where AS1 + AS2 is odd} \\ min(AS1, AS2) & \textit{where AS1 + AS2 is even} \end{cases} \tag{3.2}$$

Where AS1 and AS2 are the AS numbers of two peers, and their sum decides leader peer selection. If the two allied peers are two networks instead of two domains, then we can assign their AS number with value 0.

3.5.2 Network view and secret key sharing

In the meantime, allied peers maybe need to exchange topology view for inter-domain innovations, e.g. cross-domain multicast application. However, since different domains have different policies and their admins usually not willing to expose full topology information but partial or abstract net- work topology views to their peers due to commercial benefits and security reasons. With the help of EW-Bridge, we can abstract physical topology as a virtual network view with only virtual links and nodes (e.g., a small network in domain can be abstract as a router node), so that we can achieve our purpose in a privacy and security manner.

Another important function in our system is the key sharing component for packets sign-

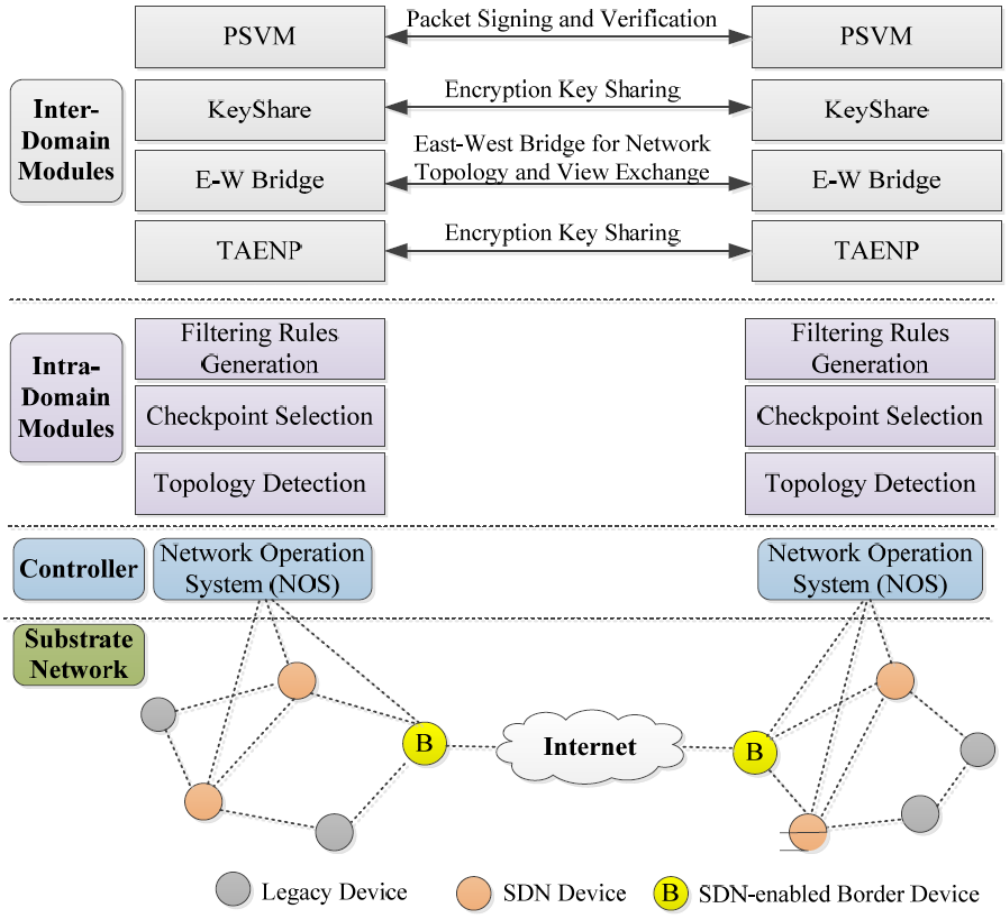


Fig. 3.5: The logical diagram of system architecture (In each allied AS, there is a SDN controller that includes domain and inter-domain modules. The former one exploits global topology and computes checkpoints' location and distribute SDN rules onto these checkpoints so as to filter spoofing flows within domain area; Differently, the latter module communicates peer module between allied ASes so as to verdict inbound packets' legitimacy or sign signature for outbound packets to peer AS, this function is performed by SDN border device with yellow circle).

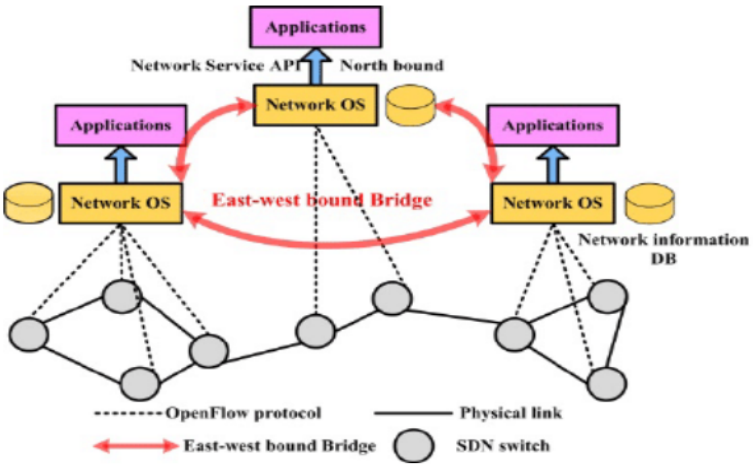


Fig. 3.6: The illustration of (a) intra-domain (b) inter-domain spoofing attack scenario.

ing between allied AS. Thus symmetric encryption key will be securely shared between each pair of peers with the help of Diffie-Hellman algorithm. Besides that, network time protocol (NTP) will facilitate to synchronize two peers' time and keep secret key update with a fixed time interval, which can defer attacker to perform replay attack or brutal force attack to analyze secret key in a short time.

East-West bridge

Large networks are always partitioned into several small networks when deploying software defined networks (SDN), and a dedicated network operating system (NOS) is deployed for each network. Each NOS has the local network view. However, to route data packets in an entire network, a global network is required. Thus, a high performance East-West Bridge with full mesh connection is proposed in this paper for heterogeneous NOSes to exchange network views in enterprise, data center, and intra-domain networks.

Diffie-Hellman

DiffieHellman key exchange (DH) is a method of securely exchanging cryptographic keys over a public channel and was one of the first public-key protocols as originally conceptualized by Ralph Merkle and named after Whitfield Diffie and Martin Hellman. DH is one of the earliest practical examples of public key exchange implemented within the field of cryptography.

Traditionally, secure encrypted communication between two parties required that they first exchange keys by some secure physical channel, such as paper key lists transported by a

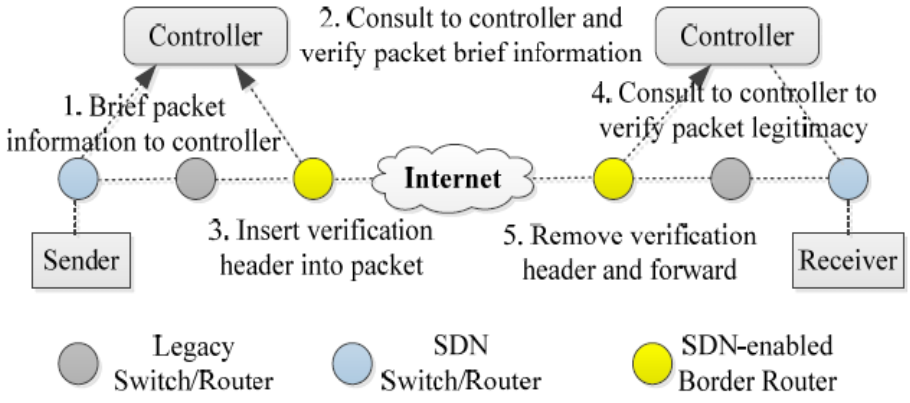


Fig. 3.7: The workflow of packet signing and verification.

trusted courier. The DiffieHellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

Network Time Protocol(NTP)

The Network Time Protocol (NTP) is a networking protocol for clock synchronization between computer systems over packet-switched, variable-latency data networks. In operation since before 1985, NTP is one of the oldest Internet protocols in current use.

NTP is intended to synchronize all participating computers to within a few milliseconds of Coordinated Universal Time (UTC). It uses the intersection algorithm, a modified version of Marzullo's algorithm, to select accurate time servers and is designed to mitigate the effects of variable network latency. NTP can usually maintain time to within tens of milliseconds over the public Internet, and can achieve better than one millisecond accuracy in local area networks under ideal conditions. Asymmetric routes and network congestion can cause errors of 100 ms or more.

The protocol is usually described in terms of a client-server model, but can as easily be used in peer-to-peer relationships where both peers consider the other to be a potential time source.Implementations send and receive timestamps using the User Datagram Protocol (UDP) on port number 123.They can also use broadcasting or multicasting, where clients passively listen to time updates after an initial round-trip calibrating exchange.

3.5.3 Packet signing and verification

As we stated, in order to verify packets authenticity between two allied ASes, packets travelling between two allied peers will be tagged signatures by source peer and verified by desti- nation. Detailly, as illustrated in Fig.3.12, once a packet released to network, its first-hop SDN device will brief controller with packets' hash result. When the packet arrives to the border of network, the SDN-enabled domain border router will consult to its controller to verify packet's brief information. Once positive result returned, the border routers will insert an extra header we named as IGuarantee header into the packet. Also, when packet arrives to destination AS, peer SDN-enabled border router will take shared key and re-compute the sig- nature so that it can compare the two signatures and verify the packet's authenticity. For inbound legitimate packets, the border router will remove the IGuarantee header and forward to next-hop. Thus the whole processes are transparent to end users.

As packet's brief generation, as illustrated in the following algorithm, we take the first-hop SDN router's IP address, packet's IP header and body parts, the three components as parameters of hash algorithm (e.g., SHA-256) so as to avoid packet's key data to be modified in the route.:

$$Brief(P) = H(IP_{SDN-router}||P_{body}||P_{IP-header})$$

3.5.4 IGuarantee header

In order to compatible with IPv4 option field that cannot exceed 40 bytes limitation, we design IGuarantee header's fields as Fig.3.8 shown. This design is derived from following considerations.

1. **Option Type:** we assign the value "00011111" for this new type of option header. According to the IPv6 specification, the first two bits indicate "skip over this option and continue processing the header," while the third bit means "option data does not change en-route." The other five con- secutive bits with customized value 1 is for device identification and processing convenience. Certainly, this value needs to be approved from the Internet Assigned Numbers Authority (IANA);

2. **Option Length:** this field indicates the while option length. Currently, our header length is fixed size of 40 bytes, but 8 bits can maximally hold 64 bytes data in this header.

3. **Version:** this field is for identify different option ver- sion when this header has multiple updates.

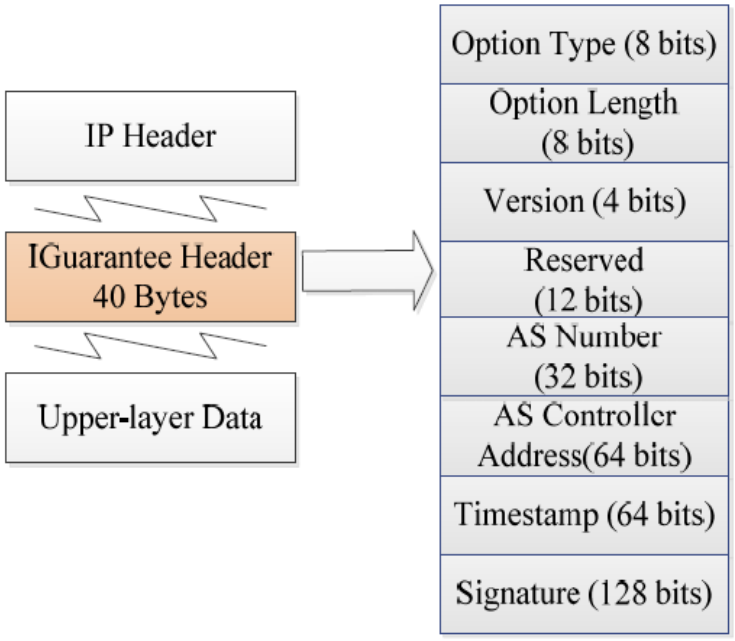


Fig. 3.8: IGuarantee header format.

4. **Reserved:** filed is reserved for future consideration.

5. **AS number:** 32 bits space is for compatible with new version AS number in IPv6 networks that is up to 32 bits.

6. **AS controller Address:** field is for hold domain AS controller's IP address, which can facilities peer AS to verify packets.

7. **Timestamp:** is aim to prevent reply attack, since receiving domain can drop inbound packets with outdated timestamp.

8. **Signature:** field is use to hold packet's signature so that end domain can re-calculate and compare it so as to decide packet's legitimacy. The signature is generated with a hash and encryption combined way:

$$Signature(P) = HMAC(K \oplus opad|(K \oplus ipad|M))$$

The K is the key shared by both ends in a pair, and HMAC is the encryption hash function (e.g., HMAC-MD5). Then the opad and ipad are the outside and inside padding for HMAC function. Last M is the message, which includes IGuarantee header, upper-layer data, and IP source and IP destination fields in IP header as well.

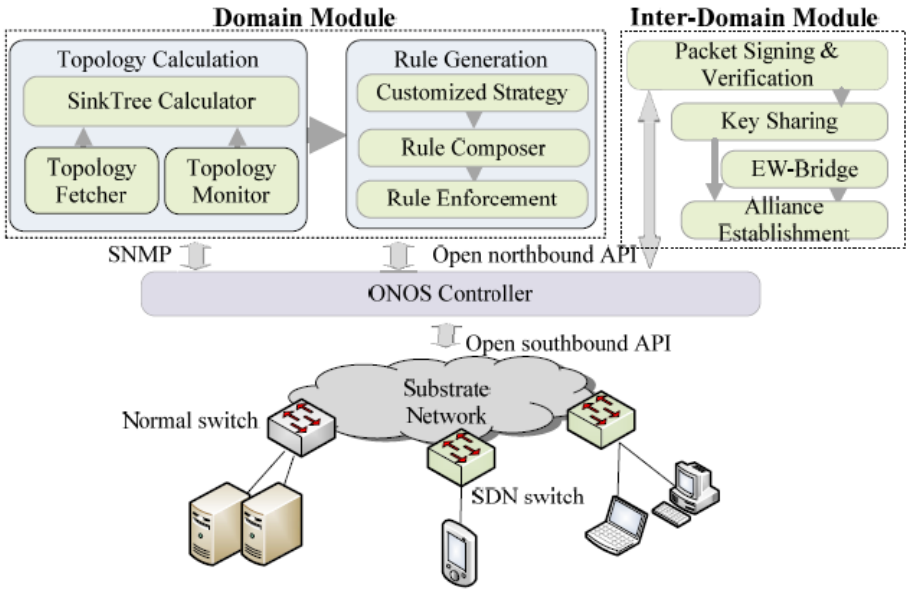


Fig. 3.9: System prototype modules in SDN controller.

3.5.5 Discussion

Packet Fragmentation issue

Since our proposal needs to add extra header into packets head to allied ASes, it will enlarge packets' size slightly. However, Maximum Transmission Unit (MTU) does not allow oversized packets to be transmitted. In IPv4 networks, we can arrange a layer 3 device outside SDN-enabled bor- der router so that it can perform packets' fragmentation function for oversized packets. While in IPv6 networks, we can decrease the MTU value in the MTU announcement if necessary.

System security issue

Considering system feasibility and performance, we take symmetric encryption algorithm and shared key to generate signature for packets between each pair of allied ASes. Thus, the key's security is the biggest concern in system security. First of all, we utilize Diffie-Hellman algorithm to distribute encryption key with a security manner. In the meantime, in order to avoid attacker perform sniff and replay attack, we setup the timestamp field and update key in every fixed interval that makes attacker cannot decryption the key in a short time.

Computing overhead issue

Indeed, computing signature and adding new header into a normal packet will incur extra overhead. However, since these operations are very common and they can be implemented by hardware, thus it can be achieve full line-rate speed.

Conclusion

Despite anti-IP spoofing has been studied extensively in the past decade, however, feasible and integrated solutions that cover both of intra-domain and inter-domain scopes still under the way of research. An integrated IP spoofing validating solution named ISASA for both intra-domain and inter-domain scenarios is presented as a solution to this problem. The intra-domain part scheme first computes key network nodes and takes SDN switches to replace traditional devices in these nodes, so that it can gain a balance between fake packets filtering rate and deploy- ment cost. Further, taking advantage of SDN pattern, filtering rules can be generated and distributed by central controller based on network real-time topology. In the meanwhile, the inter-domain part scheme proposes a time-synchronized packet signature signing and verification protocol between AS alliances. Through the established allied relationship, two ASes can exchange secret key, network abstract view and other information. Eventually, packets shuttle between the two ASes will be tagged signature header and removed after they have been verified in the destination AS. The implemented system prototype, and conducted experiments prove ISASA poses desirable performance. Based on some new research, there is a need to enhance the system architecture design and joint with network equipment manufacturer, so that related products can be released onto market and apply them into real network scenarios.

REFERENCES

[1] W. M. Eddy, ”Defenses against TCP SYN flooding attacks,” Internet Protocol J., vol. 9, no. 4, pp. 2-16, 2006.

[2] D. Senie and P. Ferguson, Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing, document RFC 2267, 1998.

[3] The Indian Computer Emergency Response Team. CERT Advisory Smurf IP Denial-of-Service Attacks. Accessed: Dec. 6, 2017. [Online]. Available: http://www/cert.org/advisories/CA-1998-01.html

[4] X. Liu, A. Li, X. Yang, and D.Wetherall, ”Passport: Secure and adoptable source authentication,” in Proc. NSDI, vol. 8. 2008, pp. 365-378.

[5] J. Bi, B. Liu, J. Wu, and Y. Shen, ”Preventing IP source address spoofing: Atwo-level, state machine-based method,” Tsinghua Sci. Technol., vol. 14, no. 4, pp. 413-422, Aug. 2009.

[6] T. Peng, C. Leckie, and K. Ramamohanarao, ”Protection from distributed denial of service attacks using history-based IP filtering,” in Proc.

[7] A. Bremler-Barr and H. Levy, ”Spoofing prevention method,” in Proc. INFOCOM, 24th Annu. Joint Conf. IEEE Comput. Commun. Soc., vol. 1. Mar. 2005, pp. 536-547.

[8] G. Chen, G. Hu, Y. Jiang, and C. Zhang, –SAVSH: IP source address validation for SDN hybrid networks,” in Proc. IEEE Symp. Comput. Commun. (ISCC), Jun. 2016, pp. 409-414.