An IPv6 Source Address Validation Testbed and Prototype Implementation

Jun Bi, Guang Yao and Jianping Wu
Tsinghua National Laboratory for Information Science and Technology
Network Research Center of Tsinghua University, Beijing 100084, China
junbi@tsinghua.edu.cn

***Abstract*—Since the Internet uses destination-based packet forwarding, malicious attacks have been launched using spoofed source addresses. In an effort to enhance the Internet with IP source address validation, we prototyped an implementation of the IPv6 Source Address Validation Architecture (SAVA) and conducted the evaluation on an IPv6 test-bed. This paper reports our prototype implementation and the test results, as well as the lessons and insights gained from our experimentation. Some enhanced methods are also introduced in this article.**

***Index Terms*—IPv6, Source Address Validation, Test-bed, prototype**

I. INTRODUCTION

The fundamental principles of today's Internet are best-effort and destination address based packet forwarding. The lack of verifying source address of IP packets being forwarded through a router makes it easy for the attackers to spoof a source IP address other than the accurate address of the attacking host. Designing a Source Address Validation Architecture (SAVA) is not only helpful to network security, but also helpful to network application, network management and accounting. We have implemented a SAVA prototype on an operational network, a native IPv6 backbone network of the China Next Generation Internet project, and conducted some evaluation experiments. In this paper we first describe our prototype solutions and then report our experimental results. In recent years, there have been some efforts in the research community and IETF to design mechanisms on fighting against source address spoofing, such as [1] [2] [3] [4] [5]. Our SAVA prototype implementation was inspired by some of the schemes from the proposed or existing solutions.

The rest of the paper is organized as follows: Section 2 introduces the architecture, Section 3 introduces the implemented prototype, Section 4 introduces the IPv6 SAVE test-bed, Section 5 discusses the limitation and future work, Section 6 introduces some enhanced methods, finally Section 7 concludes the paper.

Supported by China Science and Technology Supporting Project and China Next Generation Internet Project.

II. ARCHITECHTURE

A. Overview

Since the Internet is very large, it is unrealistic to expect any single IP source address validation mechanism to be universally supported. Different operators and vendors may choose to deploy/develop different mechanisms to achieve the same goal, and there needs to be different mechanisms to solve the problems at different places in the network. Furthermore, implementation bugs or configuration errors can also affect the effectiveness of the intended implementation. Therefore, our prototype implementation of SAVA is a combination of multiple coexisting and cooperating mechanisms. More specifically, we implement source IP address validation at three levels: access network source address validation; intra-AS source address validation; and inter-AS source address validation, as shown in Fig. 1.

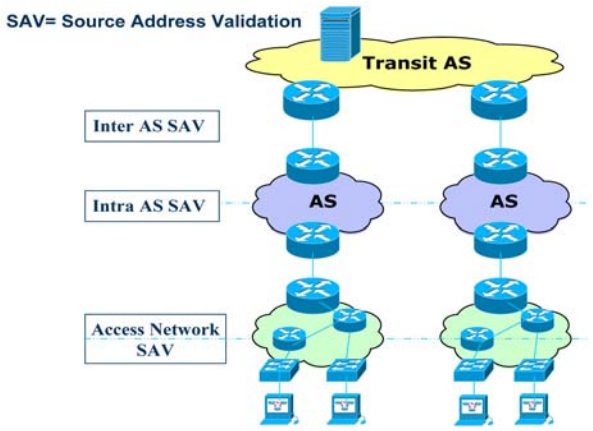

Figure 1. A Source Address Validation Architecture.

It is important to enforce IP source address validity at the access network. That is, when an IP packet is sent from a host, the routers, switches or other devices (if they implement the functions) should check to make sure that the packet carries a legally assigned source IP address. If this access network source address validation is missing, then a host may be able to spoof the source IP address which belongs to another local host.

We use the term "intra-AS source address validation" to refer to the IP source address validation at the attachment point of an access network to its provider network, which is also called the ingress point. IP source address validation at ingress points can enforce the source IP address validity at the IP prefix level, assuming that the access network owns one or more IP address blocks. This practice has been adopted as the Internet Best-Current-Practice [1] [6]. Even in the absence of the access network source address checking, this ingress checking can still prevent the hosts within one access network from spoofing IP addresses belonging to other networks.

Inter-AS IP source address validation refers to mechanisms that enforce packet source address validity at AS boundaries. The first two steps of source address validation utilize the physical connectivity of the access network and the ingress points. Because the global Internet has a mesh topology, and because different networks belong to different administrative authorities, IP source address validation at Inter-AS level becomes more challenging. Nevertheless, we believe this third level of protection is necessary to detect packets with spoofed source addresses when the first two levels of source address validation are missing or ineffective.

B. *IP Source Address Validation in the Access Network*

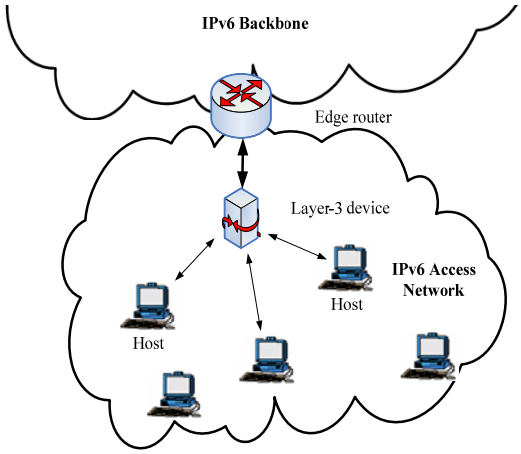

Figure 2. The deployment scenario in access network

The solution includes the following steps: (the mechanism details are described in [7])

1. When a host wants to access the Internet, it should first carry out the access authentication.
2. Then the host generates a session key S and sends it to a layer-3 device (could be the first hop router) via a key exchange mechanism. The layer-3 device binds the session key with the host's IP address.
3. When the host sends packet M to somewhere outside the access network, it generates one signature for each packet using the hash digest algorithm (e.g. MD5). Then the signature H [M||S] is carried in a new IPv6 extension header named source address validation header.
4. The layer-3 device uses the session key to authenticate the signature carried in the packet so that it can validate the source address.
5. The layer-3 device identifies the replay packets by checking whether the sequence number of the packet is increasing within the admission time window T (T is set up by timestamp mechanism).

The deployment scenario of this method is as shown in Fig. 2.

C. *IP Source Address Validation at Intra-AS/Ingress Point*

We adopted the solution of the source address validation of IP packets at ingress points described in [1] and [6]; the latter describes source address validation at the ingress points of multi-homed access networks.

D. *IP Source Address Validation at Inter-AS level*

This solution is inspired by the work [4]. The basic ideas of this light-weight signature based mechanism are as follows. For every two SAVA-compliant ASes, there is a pair of unique temporary signatures. All SAVA-compliant ASes register in a registration server and form a SAVA AS Alliance. When a packet is leaving its own AS, if the destination IP address belongs to an AS in the SAVA AS Alliance, the border router of this AS looks up the signature based on the destination AS number (derived from the packet's destination address), and tags a signature to the packet. When a packet arrives at the destination AS, if the source address of the packet belongs to an AS in the SAVA AS Alliance, the border router of the destination AS looks up the signature based on the source AS number, and the signature carried in the packet is verified and removed. This particular method uses a light-weight signature. For every packet forwarded, the signature can be put in an IPv6 hop-by-hop extension header. We can use a 128-bit shared random number as the signature, instead of using cryptographic method to generate the signature.

As shown in Fig. 4, there are three major components in the system: the Registration Server (REG), the AS Control Server (ACS), and the AS Border Router (ABR).

The Registration Server is the "center" of the Trust Alliance (TA). It maintains a member list for the TA. It performs two major functions:

1. Processes requests from the AS Control Server, to get the member list for the TA.
2. When the member list is changed, notifies each AS Control Server. Each AS deploying the method has an AS Control Server. The AS Control Server has three major functions: (1) communicates with the Registration Server to get the up-to-date member list of TA. (2) communicates with the AS Control Server in other member ASes in the TA, to exchange updates of prefix ownership information, and to exchange signatures. (3) communicates with all AS Border routers of the local AS,

to configure the processing component on the AS Border routers.

The AS Border Router does the work of adding signature to the packet at the sending AS, and the work of verifying and removing the signature at the receiving AS.

In the design of this system, in order to decrease the burden on the REG, most of the control traffic happens between ASCs.

The signature needs to be changed frequently to provide better security. However, the overhead of maintaining and exchanging signatures between AS pairs increase as the number of ASes increases. Therefore an enhanced method called APPA with automatic signature changing is proposed in section 6.

III. PROTOTYPE IMPLEMENTATION

This part will describe the implementation of SAVA prototype for inter-AS level. The prototype is a box working as an enhancement of AS border router, performing the task of source address validation. Currently, this box is named SAVD (Source Address Validation Device), and deployed beside the border router.

A SAVD is composed of a control layer and a data layer, as shown in Fig. 3. The control layer communicates with the controllers, updates the data used in filtering. And it takes charge of switching the filtering mechanisms. The data layer of SAVD checks received packets. It contains some function modules that would be necessary for some mechanism, such as key management module. The controller, mentioned above, is the entity which generates filtering rules, for instance, the AS Control server.

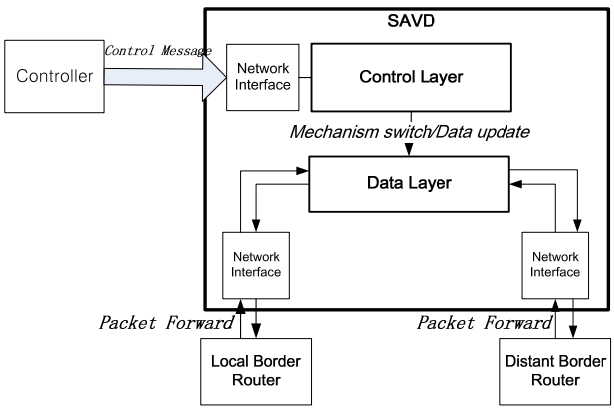

Figure 3. Structure of SAVD

A. Control layer

The control layer of SAVD receives packets from controllers and parses them to update the rule table and other data structures. The packets are from a specialized link, instead of the normal data links. Using special link can avoid using cryptographic method to authenticate the source of control message. Generally the controller doesn't have to be placed far from the SAVD, so it is not expensive to deploy a special link.

For the controllers in different mechanisms are different in function there is no restriction on the design of controller, other than it must communicate with the control layer of SAVD using special formatted packets.

There are two data structures needed to be updated: Address Mapping table and AS Number-Signature Mapping table. The former maps address to some data used in verification, and the latter maps the AS number to the signature of the AS. The details of these two data structures will be described in section 2) of part B. Fig. 4 shows the format of an Address Mapping table update packet.

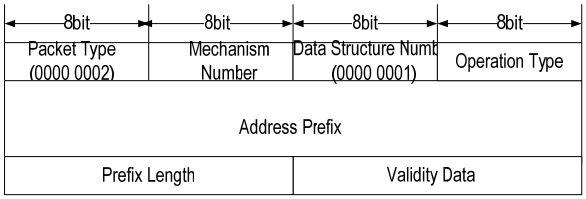

Figure 4. Format of Address Mapping table update packet

In Address Mapping table update packet, the mechanism number field tells which source validation method's data structures are to be updated. The data structure number field tells which data structure should be updated, which is 1 for Address Mapping table and 2 for the other one. Operation type field tells the action is adding a prefix to the table(0000 0001), or delete the prefix from the table(0000 0002), or update the corresponding data in the table(0000 0003). Address prefix field contains the prefix for the table with a max length of 64 bit. Prefix length field tells the actual length of the prefix. The validity data field contains the flow label for the prefix.

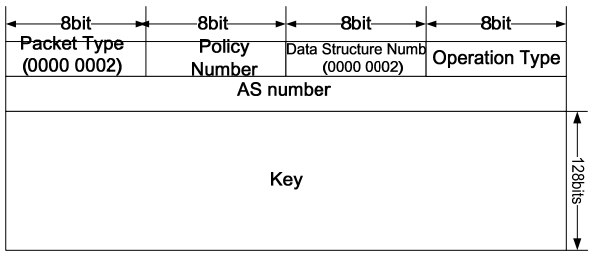

Figure 5. Format of an AS Number-Signature Mapping table update packet

Fig. 5 shows the format of an AS Number-Signature Mapping table update packet. Now the length of the key field is 128 bits in the design. The AS number family tells whose key needs update, and the key field contains the key. The operation type field tells whether to update the key(0000 0001), or remove the key(0000 0002).

B. Data layer

The data layer works according to dedicated process

flow for each mechanism. A process flow is composed by a number of basic functions and logic judgments. The basic functions are offered by function modules in data layer, and logic judgments are established separately for each mechanism. A mechanism should have two flows, one for incoming packets, and the other for outgoing packets, because generally a mechanism should perform different checks on these two kinds of packets.

The data layer of a SAVD contains a number of function modules. These modules cover basic requirements of a source address validation mechanism.

(1) Packet Transmit Module

This module takes charge of receiving and sending packets. When a packet arrives, the receiving sub-module fetches it from the network card, and inserts it into a packet queue, called Receiving Buffer Queue (RBQ). While the Sending Buffer Queue (SBQ) is not empty, the sending sub-module will always take packets from SBQ and sending them to the opposite network cards. The details about RBQ and SBQ will be described in Section i of 2).

Because SAVD is required to be inserted into network without awareness of the existing routers, the network cards in SAVD don't have IP addresses, and the packets are fetched and sent at link layer. The receiving sub-module cannot distinguish incoming packets from packets that sent out by the sending sub-module working on the same network card, as both would appear on the line without difference. We should use additional information to avoid this. Fig. 6 shows the situation. Firstly, we get the MAC address of the network card of router RA, denoted as RA_MAC. The MAC address can be learned from packets sent by A, or recorded in the configure file of SAVD manually. Then the receiving sub-module of SAVD on the inside network card (CI in Fig. 6) should filter out packets whose source MAC is not RA_MAC, and the one on the outside network card should drop packets whose source MAC is RA_MAC.

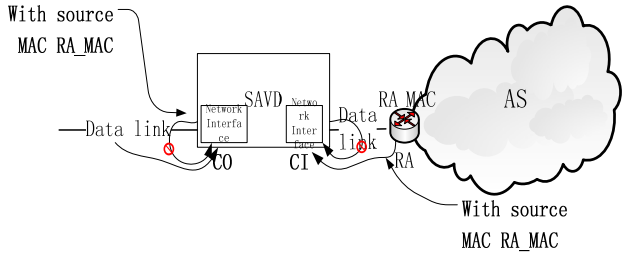

Figure 6. Filter received packets just sent by the interface itself

(2) Address Mapping Module

The Address Mapping Module deals with the initialization, update and lookup of the Address Mapping table. When a mechanism is initialized, the Address Mapping Module will generate a trie-tree, called Address Mapping table, and get to know what kind of data would be mapped to. When the control layer asks this module to add an entry to the mapping table, a piece of memory space will be allocated for the entry data, and the corresponding pointer will be added to the trie-tree. During a lookup operation, the pointer is returned and the entry data can be retrieved according to the pointer.

(3) Key Processing Module

Key Processing Module has following functions: inserting a key into a packet, checking the validation of key in a packet, and removing the key from a packet. In our design, the key is contained in a Hop-By-Hop option, with the option type number 01100. After an option with a key has been inserted, the NextHead option will be modified to recover the chain-like structure of an IPv6 packet. The key check sub-module will compare the key in a packet with the corresponding keys in AS-Key mapping table. The key remove sub-module will remove the option if any, and then recover the chain-like structure.

(3) MTU Processing Module

In many mechanisms, if a key is needed to be inserted to a packet, the packet size may be larger than the path MTU and the packet would be dropped. The ICMP error message returned by the router which fails to receive the packet, suggests a MTU that is larger than appropriate MTU by the length of the option field containing a key. Thus, this message is needed to be modified to decrease the MTU value. The MTU Processing Module checks whether the ICMP error message is caused by key insertion. If so, it will decrease the MTU value suggested and recomputed the checksum. The subtrahend is preset with the mechanism.

There are several main data structures in the data layer:

(1) SBQ and RBQ

Sending Buffer Queue and Receiving Buffer Queue are linked list of pointers, instead of real packets. All the en-queue and de-queue operations are operation on pointers.

When a packet is received, it is stored in a piece of memory space of fixed size. If the above MTU problem has been resolved, the packet should be shorter than the memory size at least by the length of a key option. We don't use the structure of mbuff, because a key may be inserted into the packet and mbuff will not be suitable for this task, especially on locating the place to insert the option. By storing the packet in a bigger and integrated space, we don't need to reallocate a space if a key is to be inserted.

(2) Address Mapping table

For any source address validation mechanism, a source address checking process is inevitably needed. In different mechanisms, source address of a packet should be mapped to different kinds of data, and the data is used to help authenticating the correctness of the source address. In the solution, the source address should be mapped to the number of AS it belongs to, and using this number we can get the correct key for the AS from an AS-Signature mapping table. Thus, a lookup table using source address, similar to a routing table, is necessary. The Address Mapping table is based on a trie-tree. The lookup result is a pointer pointing to the required data. In this way, the Address Mapping Module can map the

source address of a packet to any kind of data.

(3) AS Number-Signature Mapping table

This table is an array whose index is AS number and element stores a pointer that points to a list of zero or more key structures. This table is extensible for the length of the list and key is not solid. This would be very useful when some mechanism uses a sliding window of keys, since the number of keys can be variable.

IV. CNGI-CERNET2 SAVA TEST-BED

A. *SAVA Test-bed on CNGI-CERNET2 Infrastructure*

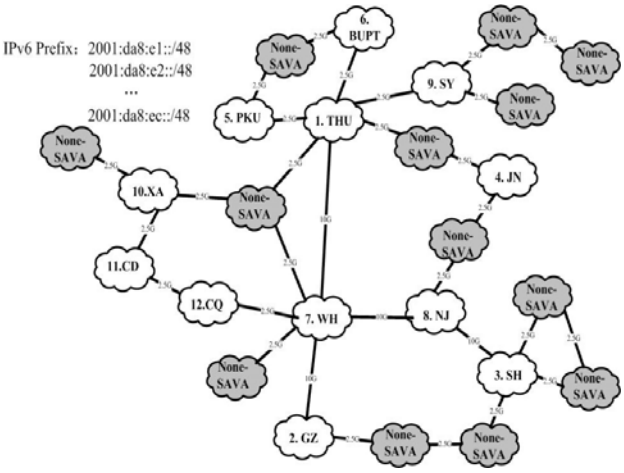

Figure 7. The CNGI-CERNET2 SAVA Test-bed

The test-bed is distributed across 12 universities connected to CNGI-CERNET2, namely Tsinghua University (THU), Peking University (PKU), Beijing University of Post and Telecommunications (BUPT), Shanghai Jiao tong University (SH), Huazhong University of Science and Technology in Wuhan (WH), Southeast University in Nanjing (NJ), and South China University of Technology in Guangzhou (GZ), Northeast University in Shenyang (SY), Xi'an Jiao tong University (XA), Shandong University in Jinan (JN), University of Electronic Science and Technology of China in Chengdu (CD) and Chongqing University (CQ). As shown in Fig. 7, the grey node denotes the none-SAVA AS.

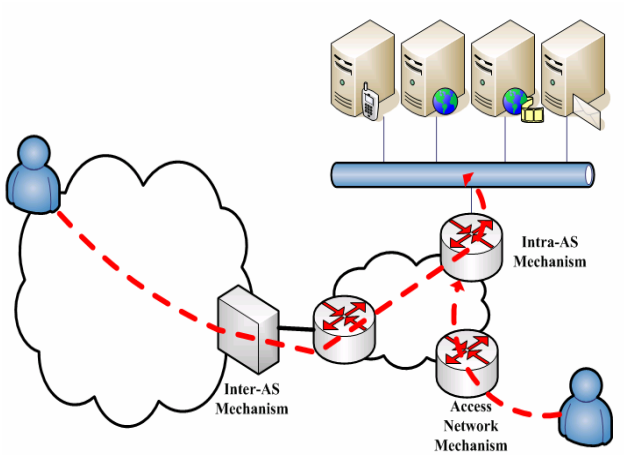

Figure 8. Applications and attack modes on the test-bed

Each of the university installations is connected to the CNGI-CERNET2 backbone through a set of inter-AS Source Address Validation prototype equipment and traffic monitoring equipment for test result display, as shown in Fig. 8.

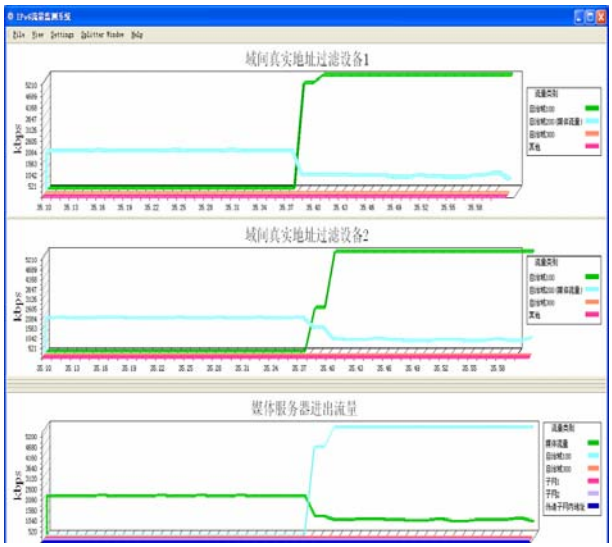

Figure 9. Flow information when an attack happens

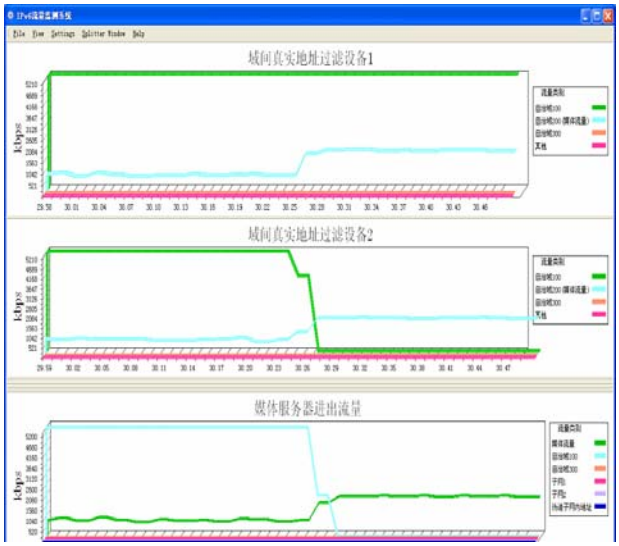

Figure 10. Flow information after we turn on SAVA mechanism

Based on this test-bed environment, we have set up some services to prove the effectiveness of SAVA mechanisms. In order to test inter-AS and intra-AS mechanisms, we set up a media server in one AS, providing video service for clients at other ASes. Then, we use an IXIA testing machine to generate packets with spoofed source addresses and launch an attack on the server. Under this attack, the performance of the server will be compromised, and the quality of video image on the client's side will drop drastically. When we turn on

the inter-AS or intra-AS mechanism, packets with spoofed source addresses will be filtered out, the media server will go back to normal, and clients can see a smooth video once again. Aside from the video quality observed by clients, we can also see the effectiveness from flow information in the network. We deploy several monitors in the test-bed, and these monitors collect flow information at different points in the network. In Fig. 9 and Fig. 10, the top row shows the flow information before the filtering device, the middle row shows the flow information after the filtering device, and the bottom row shows the flow information on the media server's link. As we can see from the figures, when we turn on SAVA mechanisms, malicious flow disappears after going through the filtering device and the media server can provide a better service.

As for access network mechanism, we set up a webpage server in one of the ASes. A computer with Windows operating system is used as the test machine. Before we turn on access network validation mechanism, the host can easily assign an address to itself and connect to the website to view the content. After the mechanism is turned on, the previously self-assigned address will stop functioning. And the host won't be able to connect with the outside network any more. Only by authenticating through a software using password can the host get a valid address, and be able to connect to the website once again.

There are also some upper-layer applications in the test-bed that use SAVA as a basis. These applications include secure email, secure BBS, secure VOIP, etc. We also carry out some experiments using these applications. The result shows that on a SAVA basis, security services can be simplified, and the level of application security can be improved very effectively.

VI. Design Limitation

There are several design limitations for the solutions deployed in CNGI-CERNET2 test-bed.

1. For the Inter-AS SAVA solution, the difficulty for guessing the signature between two AS members was discussed in [4]. It is relatively difficult and we can increase the difficulty of guess by increasing the length of the signature. In current CNGI-CERNET2 SAVA test-bed, a 128-bit signature is designed in IPv6 hop-by-hop option header. The size of the packets increases with the signatures. Because this IPv6 hop-by-hop option has to be looked at by all intervening routers, it still needs further discussion whether the IPv6 hop-by-hop option is the right tool for the task. Although the overhead is relatively low, the addition of the option and the calculation of the signature consume valuable resources on the forwarding path.

2. Given that a large fraction of current denial-of-service attacks are employing legitimate IP addresses belonging to Botnet clients, even universal deployment of better source address validation techniques would be unable to prevent these attacks. However, tracing these attacks would be easier if there could be more reliance on the validity of source address.

VII. Methods Enhancement

We have concentrated on the topic of source address validation for about 3 years. After the implementation of our initial ideas, we began to rethink our original mechanisms and the starting point. It is hard to say we have designed and implemented perfect mechanisms. The weak points of the mechanisms have been exposed in discussions and experiments, and some of them can not be remedied with ease.

However, we are still gratified for two things. First, the importance of source address validation has never been disregarded, despite of some arguments that the existence of Botnets makes attackers unnecessary to spoof source address. Source address validation brings more than eliminating spoofing DDoS attacks. Second, the 3-level architecture of source address validation has been accepted popularly. It is a firm basis for our future work.

Accompany with the implementation of the prototype, new enhanced promising mechanisms have been designed and some of them have been implemented.

A. APPA: Inter-AS level

Our current mechanism at inter-AS level can not prevent signature sniffer. This is mainly because the signature carried in packet is in plaintext and it can be sniffed easily and used by attacker. Once the signature is sniffed, the whole mechanism is disabled. Moreover, the mechanism has scalability problem, because all the AS control servers have to communicate with all others in the lifetime of a signature, which is usually one or two hours considering security problem of signature. Once the number of participants is large, the communication will cost heavy.

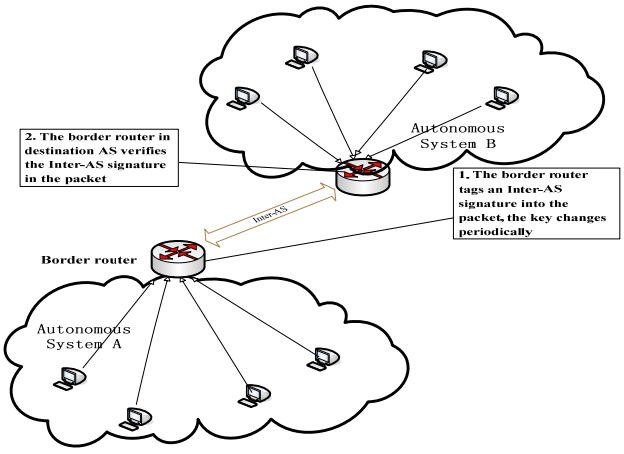

Figure 11. APPA at inter-AS level

Our solution is using self-generated signature instead of using fixed and exchanged signature. The signature is

generated by the border router itself using some pseudo random number generation algorithm, and synchronized by the time. Because the signature can be generated very fast, the effect of sniffer is weakened. And the communication between control servers is rare, so its scalability is much better.

Fig. 11 shows how this mechanism works. The detail of this mechanism is described in [8].

B. *DVF: Intra-AS level*

Currently, the mechanism at intra-AS level is actually ingress filtering. It is a simple but effective mechanism. It works at prefix granularity, but actually, we can achieve much better granularity with only little extra cost. For a border router running in intra-domain routing system, it is not hard for it to know the hops from any other router in the same domain. So it can also check the TTL value in the packet to find whether this packet has passed correct number of hops. Ingress filtering checks whether packets arrive at the correct interface, which is a direction value. The check on hops actually verifies the distance from source. The combination of the two checks can be regarded as the check on distance vector. We name this mechanism DVF (Distance Vector Filtering). Fig. 12 shows how it works.

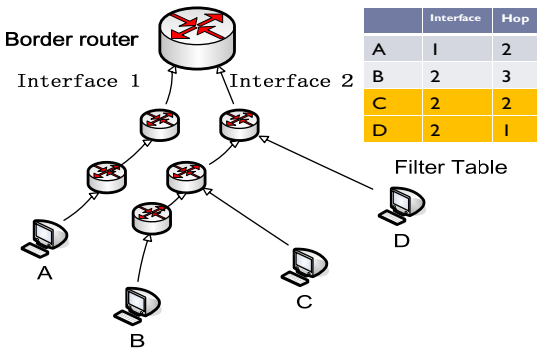

Figure 12. Filter table of DVF

C. *CSA: Access network level*

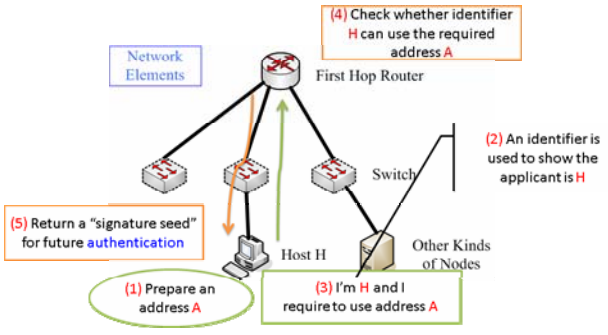

Figure 13. CSA: Address binding and seed exchange

The new mechanism for access network is designed to meet the requirement that all the address allocation methods in IPv6 should be supported by the source address validation method. Unfortunately, our previous methods only support a fixed address allocation method.

CSA (CGA based source address authentication method) is inspired by the idea of HIP (Host Identity Protocol) [9]. The main idea of CSA is described in [10]. In CSA, a host generates a self-certified identifier, and access router binds this identifier with the address allocated to it, as described in Fig. 13. A shared secret called "signature seed" is also exchanged.

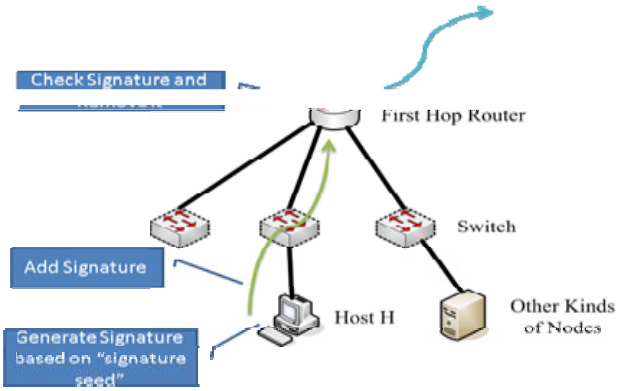

Figure 14. CSA: Signature verification.

When the host wants to send some packets out, it put a bit string generated from the signature seed into the packet. The router checks if the signature is correct corresponding to the source address. This part is shown in Fig. 14.

VIII. CONCLUSION

The early version of this paper was published in [11]. In this paper, the detail prototype implementation is introduced and the latest research progress is presented in this paper.

Several conclusions can be made from the test experience and results.

It is possible to devise a loosely-coupled and "multiple-fence" design for SAVA. This provides different granularities of authenticity of source IP addresses. It also allows different providers to use different solutions, and the coupling of components at different levels of granularity of authenticity can be loose enough to allow component substitution.

Incremental deployment is another design principle for SAVA. The tests have demonstrated that benefit is derived even when deployment is incomplete, which gives providers an incentive to be early adopters of the framework. Some DiffServ mechanism could also be considered. That is, traffic from SAVA-compliant ASes could be given a higher priority, especially when attacks are happening.

Access network source address validation is an important part of SAVA to achieve an authenticity of host

IP granularity. There are multiple access cases: local subnet in enterprise networks, residential broadband, and wireless mobile, etc. For enterprise networks, there are multiple solutions from the research and engineering community. Focusing on the appropriate framework and solutions for access network source address validation could be a valuable initial step for solving the source address spoofing problem in IETF.

SAVA must be capable of scaling to the size of the global Internet. The scalability of SAVA still needs further consideration. CNGI-CERNET2 test-bed merely provides an initial test-bed for SAVA. To study the scalability of the current solutions, we need to extend the scale of the test-bed.

REFERENCES

[1] Ferguson, P. and Senie, D., "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
[2] Park, K. and Lee, H., "On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets", SIGCOMM 2001.
[3] Li, J., Mirkovic, J., Wang, M., Reiher, P., and L. Zhang, "SAVE: Source Address Validity Enforcement Protocol", INFOCOM 2002.
[4] Bremler-Barr, A. and Levy, H., "Spoofing Prevention Method", INFOCOM 2005.
[5] Snoeren, A., et. al., "A Hash-based IP traceback", SIGCOMM 2001.
[6] Baker, F. and Savola, P., "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, 2004.
[7] Xie, L., Bi, J., and Wu, J. "An Authentication based Source Address Spoofing Prevention Method Deployed in IPv6 Edge Network", ICCS 2007.
[8] Shen, Y., Bi, J., Wu, J., and Liu, Q, “A Two-Level Source Address Spoofing Prevention based on Automatic Signature and Verification Mechanism”, the 13th IEEE Symposium on Computers and Communications (ISCC), 2008.
[9] Moskowitz, R. and P. Nikander, P. Host Identity Protocol (HIP) Architecture. Internet Engineering Task Force, May 2006. RFC 4423.
[10] Yao, G., Bi, J., “A CGA Based IP Source Address Authentication Method in IPv6 Access Network”, the 33rd IEEE Conference on Local Computer Networks (LCN), Canada, 2008.
[11] Bi, J., et. al, “An IPv6 Test-Bed Implementation for a Future Source Address Validation Architecture”, the 4th EURO-NGI Conference on Next Generation Internet Networks (NGI), Krakow, Poland, 2008.

Jun Bi received the B.S., M.S., and Ph.D. degree in computer science from Tsinghua University. Currently he is a full professor and director of Network Architecture & IPv6 Research Division, Network Research Center of Tsinghua University, Beijing, China.

Guang Yao is a Graduate student of Network Research Center of Tsinghua University.

Jianping Wu received the B.S., M.S. and Ph.D. degrees in Computer Science from Tsinghua University. Currently, he is a full professor in the Computer Science Department, Tsinghua University and director of the China Education and Research Network (CERNET).