          A Source Address Validation Architecture (SAVA) Testbed
                        and Deployment Experience

Status of This Memo

Abstract

   Because the Internet forwards packets according to the IP destination
   address, packet forwarding typically takes place without inspection
   of the source address and malicious attacks have been launched using
   spoofed source addresses. In an effort to enhance the Internet with
   IP source address validation, a prototype implementation of the IP
   Source Address Validation Architecture (SAVA) was created and an
   evaluation was conducted on an IPv6 network. This document reports
   on the prototype implementation and the test results, as well as the
   lessons and insights gained from experimentation.

Table of Contents

1. Introduction

   By design, the Internet forwards data packets solely based on the
   destination IP address. The source IP address is not checked during
   the forwarding process in most cases. This makes it easy for
   malicious hosts to spoof the source address of the IP packet. We
   believe that it would be useful to enforce the validity of the source
   IP address for all the packets being forwarded.

   Enforcing the source IP address validity would help us achieve the
   following goals:

   o Since packets which carry spoofed source addresses would not be
      forwarded, it would be impossible to launch network attacks that
      are enabled by using spoofed source addresses and more difficult
      to successfully carry out attacks enhanced or strengthened by the
      use of spoofed source addresses.

   o Being able to assume that all packet source addresses are correct
      would allow traceback to be accomplished accurately and with
      confidence. This would benefit network diagnosis, management,
      accounting, and applications.

   As part of the effort in developing a Source Address Validation
   Architecture (SAVA), we implemented a SAVA prototype and deployed the
   prototype in 12 ASes in an operational network as part of China Next
   Generation Internet (CNGI) Project [Wu07]. We conducted evaluation
   experiments. In this document, we first describe the prototype
   solutions and then report experimental results. We hope that this
   document can provide useful insights to those interested in the
   subject, and can serve as an initial input to future IETF effort in
   this area.

   In recent years, there have been a number of research and engineering
   efforts to design IP source address validation mechanisms, such as
   [RFC2827], [Park01], [Li02], [Brem05], and [Snoe01]. Our SAVA
   prototype implementation was inspired by some of the schemes from the
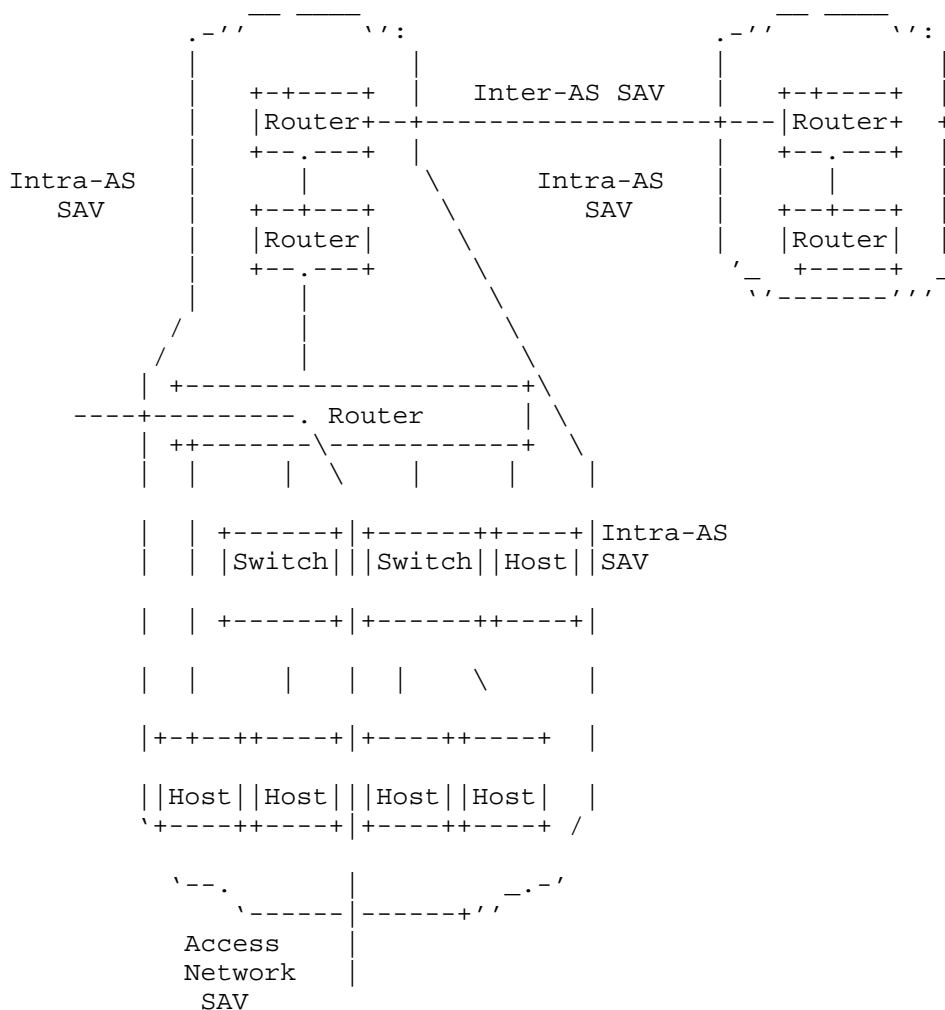   proposed or existing solutions.

   The prototype implementation and experimental results presented in
   this report serve only as an input, and by no means preempt any
   solution development that may be carried out by future IETF effort.
   Indeed, the presented solutions are experimental approaches and have
   a number of limitations as discussed in Sections 5 and 6.

2. A Prototype SAVA Implementation

2.1. Solution Overview

   A multiple-fence solution is proposed in this document. That is,
   there are multiple points in the network at which the validity of a
   packet's source address can be checked. This is because in the
   current single-fence model where source address validity is
   essentially checked only at ingress to the network, deployment has
   been inadequate to the point that there is always sufficient
   opportunity to mount attacks based on spoofed source addresses, and
   it seems likely that this condition will continue in the foreseeable
   future. A multiple-fence solution will allow "holes" in deployment
   to be covered and validity of the source address to be evaluated with
   increased confidence across the whole Internet. The assumption here
   is that when validity checking is not universal, it is still
   worthwhile to increase the confidence in the validity of source
   addresses and to reduce the opportunities to mount a source address
   spoofing attack.

   Furthermore, the architecture allows for multiple independent and
   loosely-coupled checking mechanisms. The motivation for this is that
   in the Internet at 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 end, and there need to be
   different mechanisms to solve the problem at different places in the
   network. Furthermore, implementation bugs or configuration errors
   could potentially render an implementation ineffective. Therefore,
   our prototype SAVA implementation 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
   Figure 1. The system details can be found in [Wu07].

```
                   __  ____                       __  ____
              .-''     ''':                  .-''     ''':
              |            |                 |            |
              |  +-+----+  |  Inter-AS SAV   |  +-+----+  |
              |  |Router+--+-----------------+---|Router+  +
              |  +--.---+  |                 |  +--.---+  |
    Intra-AS  |     |       \    Intra-AS    |     |      |
      SAV     |  +--+---+    \     SAV       |  +--+---+  |
              |  |Router|     \             |  |Router|  |
              |  +--.---+      \            '_  +-----+  _
              |     |           \             `'-------'''
              |    /            |             \
              |   /             |              \
              | +--------------------+\
            ----+---------. Router    | \
              | ++-------\-----------+  \
              | |   |  |  \   |      |   |
              |    |      \   |      |   |
              | | +------+|+------++----+|Intra-AS
              | | |Switch|||Switch||Host||SAV
              | | +------+|+------++----+|
              | |    |    |  |   |   \    |
              |+-+--++----+|+----++----+  |
              ||Host||Host|||Host||Host|  |
              '+----++----+|+----++----+ /
                `--.       |        _.-'
                   `------|------+''
                 Access   |
                 Network  |
                   SAV
```

        Key: SAV - Source Address Validation

                      Figure 1: Solution Overview

   This document divides source address validation into three different
   classes of solutions:

   1. Access network. This prevents a host in a network from spoofing
       the address of another host in the same network segment. This
       enables host-granularity of protection compared to Intra-AS
       prevention. See Section 2.2 for details.

   2. Intra-AS. When the edge router of an access network performs
       source address validation (e.g., using [RFC2827] and [RFC3704]),
       hosts are prevented from spoofing an arbitrary address, but
       unless access network SAV is employed, they may be able to spoof
       an address of a host in the same network segment. In a
       degenerate case, when a router connects a single host, the host
       can't spoof any address.

   3. Inter-AS. Mechanisms that enforce packet source address
       correctness at AS boundaries. Because the global Internet has a
       mesh topology, and because different networks belong to different
       administrative authorities, IP source address validation at the
       Inter-AS level is 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.

   In the following sections, we describe the specific mechanisms
   implemented at each of the three levels in detail.

2.2. IP Source Address Validation in the Access Network

   At the access network level, the solution ensures the host inside the
   access network cannot use the source address of another host. The
   host address should be a valid address assigned to the host
   statically or dynamically. The solution implemented in the
   experiment provides such a function for Ethernet networks. A layer-3
   source address validation architecture device (SAVA Device) for the
   access network (the device can be a function inside the Customer
   Premises Equipment (CPE) router or a separate device) is deployed at
   the exit of the access network. Source address validation
   architecture agents (SAVA Agents) are deployed inside the access
   network. (In fact, these agents could be a function inside the first
   hop router/switch connected to the hosts.) A set of protocols was
   designed for communication between the host, SAVA Agent, and SAVA
   Device. Only a packet originating from the host that was assigned
   that particular source address may pass through the SAVA Agent and
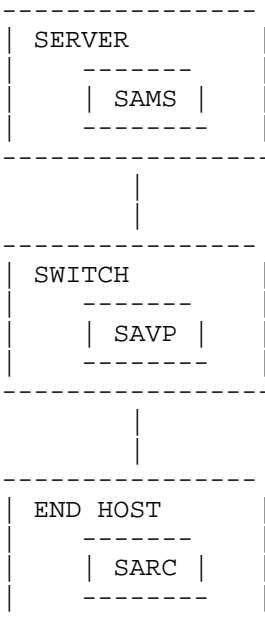   SAVA Device.

   Two possible deployment variants exist; we will call them Variant A
   and Variant B. In Variant A, an agent is mandatory and each host is
   attached to the agent on a dedicated physical port. In Variant B,
   hosts are required to perform network access authentication and
   generate key material needed to protect each packet. In this
   variant, the agent is optional.

The key function of Variant A is to create a dynamic binding between
a switch port and valid source IP address, or a binding between Media
Access Control (MAC) address, source IP address, and switch port. In
the prototype, this is established by having hosts employ a new
address configuration protocol that the switch is capable of
tracking.

Note: In a production environment, the approach in the prototype
   would not be sufficient due to reasons discussed in Section 5.

In Variant A, there are three main participants: Source Address
Request Client (SARC) on the host, Source Address Validation Proxy
(SAVP) on the switch, and Source Address Management Server (SAMS). as
shown in Figure 2. The solution follows the basic steps below:

1. The SARC on the end host sends an IP address request. The SAVP
    on the switch relays this request to the SAMS and records the MAC
    address and incoming port. If the address has already been
    predetermined by the end host, the end host still needs to put
    that address in the request message for verification by SAMS.

2. After the SAMS receives the IP address request, it then allocates
    a source address for that SARC based on the address allocation
    and management policy of the access network, it stores the
    allocation of the IP address in the SAMS history database for
    traceback, then sends response message containing the allocated
    address to the SARC.

3. After the SAVP on the access switch receives the response, it
    binds the IP address and the former stored MAC address of the
    request message with the switch port on the binding table. Then,
    it forwards the issued address to SARC on the end host.

4. The access switch begins to filter packets sent from the end
    host. Packets which do not conform to the tuple (IP address,
    Switch Port) are discarded.

```
               ----------------
              | SERVER         |
              |   -------      |
              |  | SAMS |      |
              |   -------      |
               ----------------
                      |
                      |
               ----------------
              | SWITCH         |
              |   -------      |
              |  | SAVP |      |
              |   -------      |
               ----------------
                      |
                      |
               ----------------
              | END HOST       |
              |   -------      |
              |  | SARC |      |
              |   -------      |
               ----------------


    Key:     SARC - Source Address Request Client
             SAVP - Source Address Validation Proxy
             SAMS - Source Address Management Sever
```

Figure 2: Binding-Based IP Source Address Validation
in the Access Network

The main idea of Variant B is to employ key material from network
access authentication for some additional validation process. A
session key is derived for each host connecting to the network, and
each packet sent by the host has cryptographic protection that
employs this session key. After establishing which host the packet
comes from, it again becomes possible to track whether the addresses
allocated to the host match those used by the host. The mechanism
details can be found in [XBW07], but the process follows these basic
steps:

1. When a host wants to establish connectivity, it needs to perform
    network access authentication.

2. The network access devices provide the SAVA Agent (often co-
    located) a session key S. This key is further distributed to the
    SAVA Device. The SAVA Device binds the session key and the
    host's IP address.

   3. When the host sends packet M to somewhere outside the access
       network, either the host or the SAVA Agent needs to generate a
       message authentication code for each using key S and packet M.
        In the prototype, the message authentication code is carried in
       an experimental IPv6 extension header.

   4. The SAVA Device uses the session key to authenticate the
       signature carried in the packet so that it can validate the
       source address.

   In our testbed, we implemented and tested both solutions. The
   switch-based solution has better performance, but the switches in the
   access network would need to be upgraded (usually the number of
   switches in an access network is large). The signature-based
   solution could be deployed between the host and the exit router, but
   it has some extra cost in inserting and validating the signature.

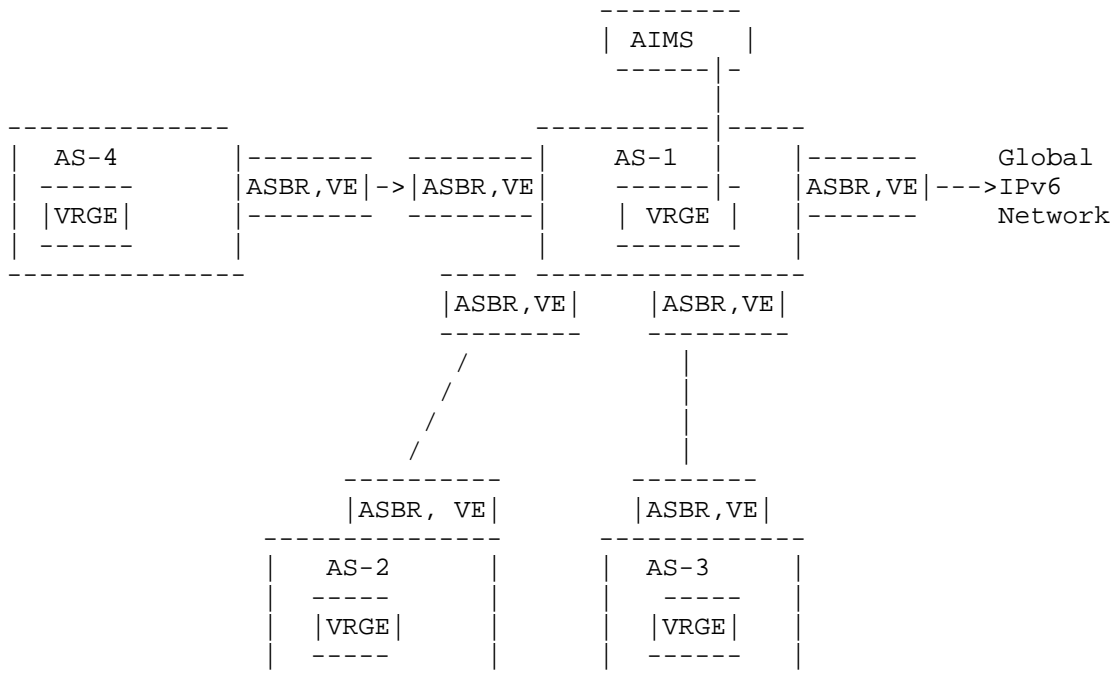2.3. IP Source Address Validation at Intra-AS/Ingress Point

   We adopted the solution of the source address validation of IP
   packets at ingress points described in [RFC2827] and [RFC3704]; the
   latter describes source address validation at the ingress points of
   multi-homed access networks.

2.4. IP Source Address Validation in the Inter-AS Case (Neighboring AS)

   Our design for the Inter-AS Source Address Validation included the
   following characteristics: It should cooperate among different ASes
   with different administrative authorities and different interests.
   It should be lightweight enough to support high throughput and not to
   influence forwarding efficiency.

   The inter-AS level of SAVA can be classified into two sub-cases:

   o Two SAVA-compliant ASes exchanging traffic are directly connected;

   o Two SAVA-compliant ASes are separated by one or more intervening,
      non-SAVA-compliant providers.

```
                                ---------
                               |  AIMS   |
                                ------|-
                                      |
        --------------            ----------|-----
       |  AS-4        |          |          |     |-------         Global
       | ------       |--------  --------|  AS-1   |     |ASBR,VE|--->IPv6
       ||VRGE|        |ASBR,VE|->|ASBR,VE| ------|-      |-------    Network
       | ------       |--------  --------|  | VRGE  |    |
       | ------       |          |        | -------      |
        --------------            ----- ----------------
                        |ASBR,VE|   |ASBR,VE|
                         ---------    ---------
                           /            |
                          /             |
                         /              |
                        /               |
                  ----------        --------
                 |ASBR, VE|        |ASBR,VE|
          --------------           -------------
         |      AS-2    |         |      AS-3   |
         | -----        |         | -----       |
         ||VRGE|        |         ||VRGE|       |
         | -----        |         | ------      |
          --------------           -------------
```

         Key: AIMS - AS-IPv6 prefix Mapping Server
                  ASBR - AS Border Router
                  VE   - Validating Engine
                  VR   - Validation Rule
                  VRGE - Validation Rule Generating Engine

                 Figure 3: Inter-ISP (Neighboring AS) Solution

   Two ASes that exchange traffic have a customer-to-provider, provider-
   to-customer, peer-to-peer, or sibling-to-sibling relationship. In a
   customer-to-provider or provider-to-customer relationship, the
   customer typically belongs to a smaller administrative domain that
   pays a larger administrative domain for access to the rest of
   Internet. The provider is an AS that belongs to the larger
   administrative domain. In a peer-to-peer relationship, the two peers
   typically belong to administrative domains of comparable size and
   find it mutually advantageous to exchange traffic between their
   respective customers. Two ASes have a sibling-to-sibling
   relationship if they belong to the same administrative domain or to
   administrative domains that have a mutual-transit agreement.

An AS-relation-based mechanism is used for neighboring SAVA-compliant
ASes. The basic ideas of this AS-relation-based mechanism are as
follows. It builds a VR table that associates each incoming
interface of a router with a set of valid source address blocks, and
then uses it to filter spoofed packets.

In the solution implemented on the testbed, the solution for the
validation of IPv6 prefixes is separated into three functional
modules: The Validation Rule Generating Engine (VRGE), the Validation
Engine (VE), and the AS-IPv6 prefix Mapping Server (AIMS).
Validation rules that are generated by the VRGE are expressed as IPv6
address prefixes.

The VRGE generates validation rules that are derived according to
Table 1, and each AS has a VRGE. The VE loads validation rules
generated by VRGE to filter packets passed between ASes (in the case
of Figure 3, from neighboring ASes into AS-1). In the SAVA testbed,
the VE is implemented as a simulated layer-2 device on a Linux-based
machine inserted into the data path just outside each ASBR interface
that faces a neighboring AS. In a real-world implementation, it
would probably be implemented as a packet-filtering set on the ASBR.
The AS-IPv6 prefix mapping server is also implemented on a Linux
machine and derives a mapping between an IPv6 prefix and the AS
number of that prefix.

\Export To \	Own Address	Customer's Address	Sibling's Address	Provider's Address	Peer's Address
Provider	Y	Y	Y		
Customer	Y	Y	Y	Y	Y
Peer	Y	Y	Y		
Sibling	Y	Y	Y	Y	Y

Table 1: AS-Relation-Based Inter-AS Filtering

Different ASes exchange and transmit VR information using the AS-
Relation-Based Export Rules in the VRGE. As per Table 1, an AS
exports the address prefixes of itself, its customers, its providers,
its siblings, and its peers to its customers and siblings as valid
prefixes, while it only exports the address prefixes of itself, its
customers, and its siblings to its providers and peers as valid
prefixes. With the support of the AS-IPv6 prefix mapping server,
only AS numbers of valid address prefixes are transferred between
ASes, and the AS number is mapped to address prefixes at the VRGE.

Only changes of AS relation and changes of IP address prefixes
belonging to an AS require the generation of VR updates.

The procedure's principal steps are as follows (starting from AS-1 in
Figure 3):

1. When the VRGE has initialized, it reads its neighboring SAVA-
    compliant AS table and establishes connections to all the VEs in
    its own AS.

2. The VRGE initiates a VR renewal. According to its export table,
    it sends its own originated VR to VRGEs of neighboring ASes. In
    this process, VRs are expressed as AS numbers.

3. When a VRGE receives a new VR from its neighbor, it uses its own
    export table to decide whether it should accept the VR and, if it
    accepts a VR, whether or not it should re-export the VR to other
    neighboring ASes.

4. If the VRGE accepts a VR, it uses the AIMS to transform the AS-
    expressed VR into an IPv6 prefix-expressed VR.

5. The VRGE pushes the VR to all the VEs in its AS.

The VEs use these prefix-based VRs to validate the source IP
addresses of incoming packets.

2.5. IP Source Address Validation in the Inter-AS Case
      (Non-Neighboring AS)

In the case where two ASes do not exchange packets directly, it is
not possible to deploy a solution like that described in the previous
section. However, it is highly desirable for non-neighboring ISPs to
be able to form a trust alliance such that packets leaving one AS
will be recognized by the other and inherit the validation status
they possessed on leaving the first AS. There is more than one way
to do this. For the SAVA experiments to date, an authentication tag
method has been used. This solution is inspired by the work of
[Brem05].

The key elements of this lightweight authentication tag based
mechanism are as follows: For each pair of SAVA-compliant ASes, there
is a pair of unique temporary authentication tags. All SAVA-
compliant ASes together 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 edge router of this AS looks up the
authentication tag using the destination AS number as the key, and
adds an authentication tag 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 edge router of the destination AS
searches its table for the authentication tag using the source AS
number as the key, and the authentication tag carried in the packet
is verified and removed. As suggested by its name, this particular
method uses a lightweight authentication tag. For every packet
forwarded, the authentication tag can be put in an IPv6 hop-by-hop
extension header. It is reasonable to use a 128-bit shared random
number as the authentication tag to save the processing overhead
brought by using a cryptographic method to generate the
authentication tag.

The benefit of this scheme compared to merely turning on local
address validation (such as RFC 2827) is as follows: when local
address validation is employed within a group of networks, it is
assured that their networks do not send spoofed packets. But other
networks may still do this. With the above scheme, however, this
capability is eliminated. If someone outside the alliance spoofs a
packet using a source address from someone within the alliance, the
members of the alliance refuse to accept such a packet.

```
                                 +-----+
                   .-------------+ REG |----------------.
                   |             +-----+                |
                   |                                    |
              ,-----+--------                   ,------+-------
            ,'      `|       `.                ,'     ` |     `.
           /         |         \              /         |       \
          /          |          \            /          |        \
         ;    +--'--+      +----+            +----+    +-----+      ;
         |    | ASC +------+ASBR|            |ASBR+-----+ ASC |      |
         :    +--.--+      +----+`           +----+    +--+--+      :
          \      |_____/            _____|         /
           \     |             /              \         |        /
            `.   |           ,'                `.       |      ,'
             `.  '------------'                  '------------'
               '------------'                      '------------'
              AS-1                                  AS-2
```
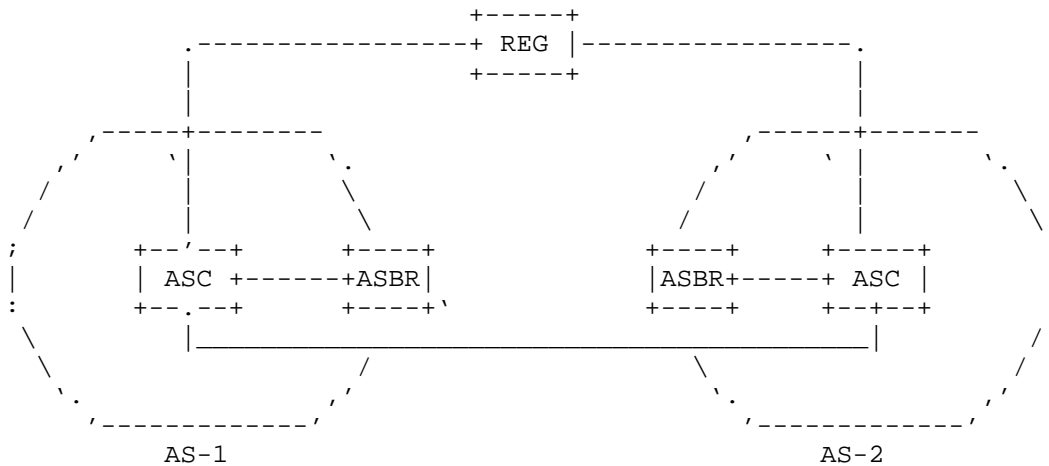
            Key: REG - Registration Server
                   ASC - AS Control Server
                   ASBR - AS Border Router

           Figure 4: Inter-AS (Non-Neighboring AS) Solution

There are three major components in the system: the Registration
Server (REG), the AS Control Server (ASC), and the AS Border Router
(ASBR).

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

o Processes requests from the AS Control Server, to get the member
   list for the TA.

o Notifies each AS Control Server when the member list is changed.

Each AS deploying the method has an AS Control Server. The AS
Control Server has three major functions:

o Communicates with the Registration Server, to get the up-to-date
   member list of TA.

o Communicates with the AS Control Server in other member ASes in
   the TA, to exchange updates of prefix ownership information and to
   exchange authentication tags.

o 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 the authentication tag
to the packet at the sending AS, and the work of verifying and
removing the authentication tag at the destination 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 authentication tag needs to be changed periodically. Although
the overhead of maintaining and exchanging authentication tags
between AS pairs is O(N) from the point of view of one AS, rather
than O(N^2), the traffic and processing overhead do increase as the
number of ASes increases. Therefore, an automatic authentication tag
refresh mechanism is utilized in this solution. In this mechanism,
each peer runs the same algorithm to automatically generate an
authentication tag sequence. Then the authentication tag in packets
can be changed automatically with high frequency. To enhance the
security, a seed is used for the algorithm to generate an
authentication tag sequence robust against guessing. Thus, the peers
need only to negotiate and change the seed at very low frequency.
This lowers the overhead associated with frequently negotiating and
changing the authentication tag while maintaining acceptable
security.

Since the authentication tag is put in an IPv6 hop-by-hop extension
header, the MTU issues should be considered. Currently we have two
solutions to this problem. Neither of the solutions is perfect, but

they are both feasible. One possible way is to set the MTU at the
ASBR to be 1280 bytes, which is the minimum MTU for the IPv6. Thus,
there should be no ICMP "Packet Too Big" message received from the
downstream gateways. The disadvantage of this solution is that it
doesn't make good use of the available MTU. The other possible way
is to let the ASBR catch all incoming ICMP "Packet Too Big" messages,
and decrease the value in the MTU field before forwarding it into the
AS. The advantage of this solution is that it can make good use of
the available MTU. But such processing of ICMP packets at the ASBR
may create a target for a denial-of-service (DoS) attack.

Because the authentication tag is validated at the border router of
the destination AS, not the destination host, the destination options
header is not chosen to carry the authentication tag.

Authentication tag management is a critical issue. Our work focused
on two points: tag negotiation and tag refresh. The tag negotiation
happens between the ASCs of a pair of ASes in the SAVA AS Alliance.
Considering the issue of synchronization and the incentive of
enabling SAVA, receiver-driven tag negotiation is suggested. It
gives more decision power to the receiver AS rather than the sender
AS. With a receiver-driven scheme, the receiver AS can decide the
policies of tag management. The packets tagged with old
authentication tags should not be allowed indefinitely. Rather,
after having negotiated a new tag, the old tag should be set to be
invalid after a period of time. The length of this period is a
parameter that will control how long the old tag will be valid after
the new tag has been assigned. In the experiment, we used five
seconds.

The trust alliance is intended to be established dynamically (join
and quit), but in this testbed we needed to confirm off-line the
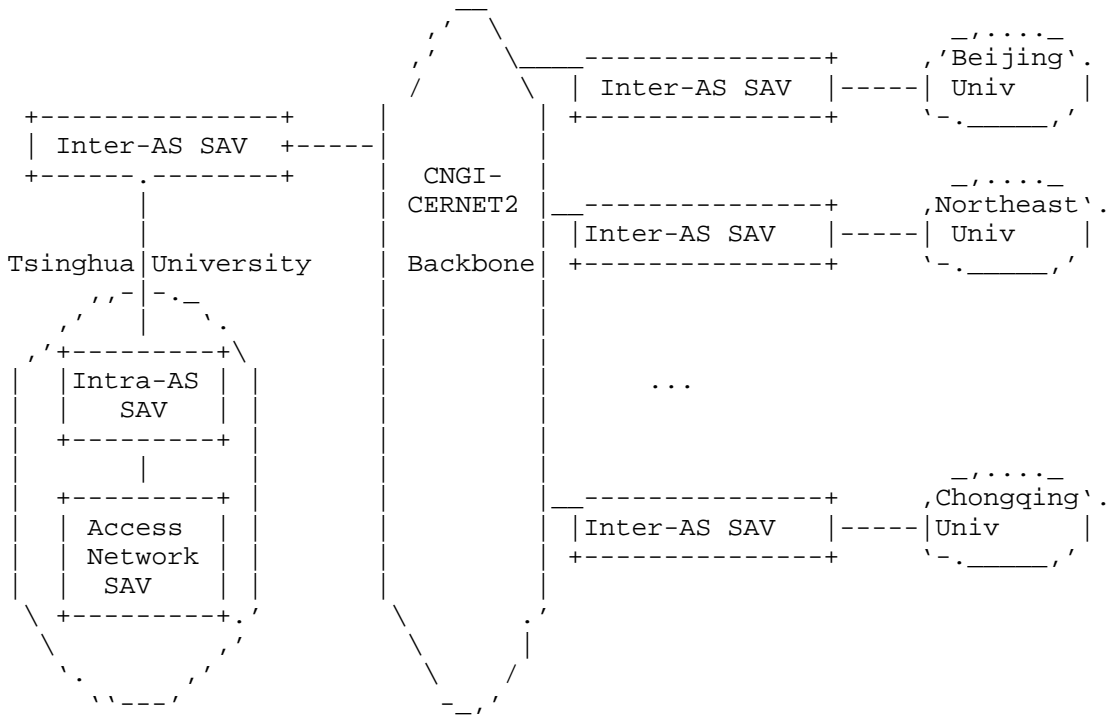initial trust among alliance members.

3. SAVA Testbed

3.1. CNGI-CERNET2

The prototypes of our solutions for SAVA are implemented and tested
on CNGI-CERNET2. CNGI-CERNET2 is one of the China Next Generation
Internet (CNGI) backbones, operated by the China Education and
Research Network (CERNET). CNGI-CERNET2 connects 25 core nodes
distributed in 20 cities in China at speeds of 2.5-10 Gb/s. The
CNGI-CERNET2 backbones are IPv6-only networks rather than being a
mixed IPv4/IPv6 infrastructure. Only some Customer Premises Networks
(CPNs) are dual-stacked. The CNGI-CERNET2 backbones, CNGI-CERNET2
CPNs, and CNGI-6IX all have globally unique AS numbers. Thus a
multi-AS testbed environment is provided.

3.2. SAVA Testbed on CNGI-CERNET2 Infrastructure

   It is intended that eventually the SAVA testbed will be implemented
   directly on the CNGI-CERNET2 backbone, but in the early stages the
   testbed has been implemented across 12 universities connected to
   CNGI-CERNET2. First, this is because some of the algorithms need to
   be implemented in the testbed routers themselves, and to date they
   have not been implemented on any of the commercial routers forming
   the CNGI-CERNET2 backbone. Second, since CNGI-CERNET2 is an
   operational backbone, any new protocols and networking techniques
   need to be tested in a non-disruptive way.

```
                               __
                            ,’    \                        _,...._
                          ,’       \____--------------+    ,’Beijing‘.
                         /          \    | Inter-AS SAV  |-----| Univ     |
    +---------------+     |           |    +--------------+    ‘-.____,’
    | Inter-AS SAV  +-----|           |
    +------.--------+     |  CNGI-    |                        _,...._
           |             | CERNET2   |__--------------+    ,Northeast‘.
           |             |           |  |Inter-AS SAV  |-----| Univ     |
    Tsinghua|University  | Backbone|    +--------------+    ‘-.____,’
        ,,-|-._          |           |
      ,’   |    ‘.       |           |
    ,’+---------+\       |           |
    | |Intra-AS | |      |           |      ...
    | |  SAV    | |      |           |
    | +---------+ |      |           |
    |     |       |      |           |                        _,...._
    | +---------+ |      |           |__--------------+    ,Chongqing‘.
    | | Access  | |      |           |  |Inter-AS SAV  |-----|Univ      |
    | | Network | |      |           |  +--------------+    ‘-.____,’
    | |  SAV    | |      |           |
    \ +---------+.’      \          .’
     \        ,’          \         |
      ‘.    ,’             \        /
        ‘‘---’             -_,’
```

              Key: SAV - Source Address Validation

                Figure 5: CNGI-CERNET2 SAVA Testbed

   In any case, the testbed is fully capable of functional testing of
   solutions for all parts of SAVA. This includes testing procedures
   for ensuring the validity of IPv6 source addresses in the access
   network, in packets sent from the access network to an IPv6 service
   provider, in packets sent within one service provider’s network, in

packets sent between neighboring service providers, and in packets
sent between service providers separated by an intervening transit
network.

The testbed is distributed across 12 universities connected to CNGI-
CERNET2.

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.

Each university deployed one AS. Six universities deployed all parts
of the solution and are hence fully-featured, with validation at the
inter-AS, intra-AS, and access network levels all able to be tested.
In addition, a suite of applications that could be subject to
spoofing attacks or that can be subverted to carry out spoofing
attacks were installed on a variety of servers. Two solutions for
the access network were deployed.

4. Test Experience and Results

   The solutions outlined in section 2 were implemented on the testbed
   described in section 3. Successful testing of all solutions was been
   carried out, as detailed in the following sections.

4.1. Test Scenarios

   For each of the test scenarios, we tested many cases. Taking the
   Inter-AS (non-neighboring AS) SAVA solution test as an example, we
   classified the test cases into three classes: normal class, dynamic
   class, and anti-spoofing class.

   1. For normal class, there are three cases: Adding authentication
       tag Test, Removing authentication tag Test, and Forwarding
       packets with valid source address.

   2. For dynamic class, there are four cases: Updating the
       authentication tag between ASes, The protection for a newly
       joined member AS, Adding address space, and Deleting address
       space.

   3. For anti-spoofing class, there is one case: Filtering of packets
       with forged IP addresses.

   As is shown in Figure 5, we have "multiple-fence" design for our SAVA
   testbed. If source address validation is deployed in the access
   network, we can get a host granularity validation. If source address

validation is deployed at the intra-AS level, we can guarantee that
the packets sent from this point have a correct IP prefix. If source
address validation is deployed at the inter-AS level, we can
guarantee that the packets sent from this point are from the correct
AS.

4.2. Test Results

   1. The test results are consistent with the expected ones. For an
       AS that has fully-featured SAVA deployment with validation at the
       inter-AS, intra-AS, and access network levels, packets that do
       not hold an authenticated source address will not be forwarded in
       the network. As a result, it is not possible to launch network
       attacks with spoofed source addresses. Moreover, the traffic in
       the network can be traced back accurately.

   2. For the Inter-AS (non-neighboring AS) SAVA solution, during the
       period of authentication tag update, the old and the new
       authentication tags are both valid for source address validation;
       thus, there is no packet loss.

   3. For the Inter-AS (non-neighboring AS) SAVA solution, the
       validation function is implemented in software on a device
       running Linux, which simulates the source address validation
       functions of a router interface. It is a layer-2 device because
       it needs to be transparent to the router interface. During the
       test, when the devices were connected directly, normal line-rate
       forwarding was achieved. When the devices were connected with
       routers from another vendor, only a very limited forwarding speed
       was achieved. The reason is that the authentication tags are
       added on the IPv6 hop-by-hop option header, and many current
       routers can handle the hop-by-hop options only at a limited rate.

5. Limitations and Issues

   There are several issues both within this overall problem area and
   with the particular approach taken in the experiment.

5.1. General Issues

   There is a long-standing debate about whether the lack of universal
   deployment of source address validation is a technical issue that
   needs a technical solution, or if mere further deployment of existing
   tools (such as RFC 2827) would be a more cost effective way to
   improve the situation. Further deployment efforts of this tool have
   proved to be slow, however. Some of the solutions prototyped in this
   experiment allow a group of network operators to have additional
   protection for their networks while waiting for universal deployment

of simpler tools in the rest of the Internet. This allows them to
prevent spoofing attacks that the simple tools alone would not be
able to prevent, even if already deployed within the group.

Similarly, since a large fraction of current denial-of-service
attacks can be launched by 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 given that
there would be more reliance on the validity of source address.

There is also a question about the optimal placement of the source
address validation checks. The simplest model is placing the checks
on the border of a network. Such RFC 2827-style checks are more
widely deployed than full checks ensuring that all addresses within
the network are correct. It can be argued that it is sufficient to
provide such coarse granularity checks, because this makes it at
least possible to find the responsible network administrators.
However, depending on the type of network in question, those
administrators may or may not find it easy to track the specific
offending machines or users. It is obviously required that the
administrators have a way to trace offending equipment or users --
even if the network does not block spoofed packets in real-time.

New technology for address validation would also face a number of
deployment barriers. For instance, all current technology can be
locally and independently applied. A system that requires global
operation (such as the Inter-AS validation mechanism) would require
significant coordination, deployment synchronization, configuration,
key setup, and other issues, given the number of ASes.

Similarly, deploying host-based access network address validation
mechanisms requires host changes, and can generally be done only when
the network owners are in control of all hosts. Even then, the
changing availability of the host for all types of products and
platforms would likely prevent universal deployment even within a
single network.

There may be also be significant costs involved in some of these
solutions. For instance, in an environment where access network
authentication is normally not required, employing an authentication-
based access network address validation would require deployment of
equipment capable of this authentication as well as credentials
distribution for all devices. Such undertaking is typically only
initiated after careful evaluation of the costs and benefits
involved.

Finally, all the presented solutions have issues in situations that
go beyond a simple model of a host connecting to a network via the
same single interface at all times. Multihoming, failover, and some
forms of mobility or wireless solutions may collide with the
requirements of source address validation. In general, dynamic
changes to the attachment of hosts and topology of the routing
infrastructure are something that would have to be handled in a
production environment.

5.2. Security Issues

The security vs. scalability of the authentication tags in the
Inter-AS (non-neighboring AS) SAVA solution presents a difficult
tradeoff. Some analysis about the difficulty of guessing the
authentication tag between two AS members was discussed in [Brem05].
It is relatively difficult, even with using a random number as an
"authentication tag". The difficulty of guessing can be increased by
increasing the length of the authentication tag.

In any case, the random number approach is definitely vulnerable to
attackers who are on the path between the two ASes.

On the other hand, using an actual cryptographic hash in the packets
will cause a significant increase in the amount of effort needed to
forward a packet. In general, addition of the option and the
calculation of the authentication tag consume valuable resources on
the forwarding path. This resource usage comes on top of everything
else that modern routers need to do at ever increasing line speeds.
It is far from clear that the costs are worth the benefits.

5.3. Protocol Details

In the current CNGI-CERNET2 SAVA testbed, a 128-bit authentication
tag is placed in an IPv6 hop-by-hop option header. The size of the
packets increases with the authentication tags. This by itself is
expected to be acceptable, if the network administrator feels that
the additional protection is needed. The size increases may result
in an MTU issue, and we found a way to resolve it in the testbed.
Since an IPv6 hop-by-hop option header was chosen, the option header
has to be examined by all intervening routers. While in theory this
should pose no concern, the test results show that many current
routers handle hop-by-hop options with a much reduced throughput
compared to normal traffic.

The Inter-AS (neighboring AS) SAVA solution is based on the AS
relation; thus, it may not synchronize with the dynamics of route
changes very quickly and it may cause false positives. Currently,

CNGI-CERNET2 is a relatively stable network, and this method works
well in the testbed. We will further study the impact of false
positives in an unstable network.

The access network address validation solution is merely one option
among many. Solutions appear to depend highly on the chosen link
technology and network architecture. For instance, source address
validation on point-to-point links is easy and has generally been
supported by implementations for years. Validation in shared
networks has been more problematic, but is increasing in importance
given the use of Ethernet technology across administrative boundaries
(such as in DSL). In any case, the prototyped solution has a number
of limitations, including the decision to use a new address
configuration protocol. In a production environment, a solution that
is suitable for all IPv6 address assignment mechanisms would be
needed.

6. Conclusion

Several conclusions can be drawn from the experiment.

First, the experiment is a proof that a prototype can be built that
is deployable on loosely-coupled domains of test networks in a
limited scale and "multiple-fence" design for source address
validation. The solution allows different validation granularities,
and also allows different providers to use different solutions. The
coupling of components at different levels of granularity can be
loose enough to allow component substitution.

Incremental deployment is another design principle that was used in
the experiment. The tests have demonstrated that benefit is derived
even when deployment is incomplete, thus giving providers an
incentive to be early adopters.

The experiment also provided a proof of concept for the switch-based
local subnet validation, network authentication based validation,
filter-based Inter-AS validation, and authentication tag-based
Inter-AS validation mechanisms. The client host and network
equipment need to be modified and some new servers should be
deployed.

Nevertheless, as discussed in the previous section, there are a
number of limitations, issues, and questions in the prototype designs
and the overall source address validation space.

It is our hope that some of the experiences will help vendors and the
Internet standards community in these efforts. Future work in this
space should attempt to answer some of the issues raised in
Section 5. Some of the key issues going forward include:

o Scalability questions and per-packet operations.

o Protocol design issues, such as integration to existing address
   allocation mechanisms, use of hop-by-hop headers, etc.

o Cost vs. benefit questions. These may be ultimately answered only
   by actually employing some of these technologies in production
   networks.

o Trust establishment issue and study of false positives.

o Deployability considerations, e.g. modifiability of switches,
   hosts, etc.

7. Security Considerations

The purpose of the document is to report experimental results. Some
security considerations of the solution mechanisms of the testbed are
mentioned in the document, but are not the main problem to be
described in this document.

8. Acknowledgements

This experiment was conducted among 12 universities -- namely,
Tsinghua University, Beijing University, Beijing University of Post
and Telecommunications, Shanghai Jiaotong University, Huazhong
University of Science and Technology in Wuhan, Southeast University
in Nanjing, South China University of Technology in Guangzhou,
Northeast University in Shenyang, Xi'an Jiaotong University, Shandong
University in Jinan, University of Electronic Science and Technology
of China in Chengdu, and Chongqing University. The authors would
like to thank everyone involved in this effort in these universities.

The authors would like to thank Jari Arkko, Lixia Zhang, and Pekka
Savola for their detailed review comments on this document, and thank
Paul Ferguson and Ron Bonica for their valuable advice on the
solution development and the testbed implementation.

9. References

9.1. Normative References

   [RFC2827]   Ferguson, P. and D. Senie, "Network Ingress Filtering:
               Defeating Denial of Service Attacks which employ IP Source
               Address Spoofing", BCP 38, RFC 2827, May 2000.

   [RFC3704]   Baker, F. and P. Savola, "Ingress Filtering for Multihomed
               Networks", BCP 84, RFC 3704, March 2004.

9.2. Informative References

   [Brem05]   Bremler-Barr, A. and H. Levy, "Spoofing Prevention
               Method", INFOCOM 2005.

   [Li02]   Li,, J., Mirkovic, J., Wang, M., Reiher, P., and L.
               Zhang, "SAVE: Source Address Validity Enforcement
               Protocol", INFOCOM 2002.

   [Park01]   Park, K. and H. Lee, "On the effectiveness of route-based
               packet filtering for distributed DoS attack prevention in
               power-law internets", SIGCOMM 2001.

   [Snoe01]   Snoeren, A., Partridge, C., Sanchez, L., and C. Jones, "A
               Hash-based IP traceback", SIGCOMM 2001.

   [Wu07]   Wu, J., Ren, G., and X. Li, "Source Address Validation:
               Architecture and Protocol Design", ICNP 2007.

   [XBW07]   Xie, L., Bi, J., and J. Wu, "An Authentication based
               Source Address Spoofing Prevention Method Deployed in IPv6
               Edge Network", ICCS 2007.

Authors' Addresses

    Jianping Wu
    Tsinghua University
    Computer Science, Tsinghua University
    Beijing 100084
    China
    EMail: jianping@cernet.edu.cn

    Jun Bi
    Tsinghua University
    Network Research Center, Tsinghua University
    Beijing 100084
    China
    EMail: junbi@cernet.edu.cn

    Xing Li
    Tsinghua University
    Electronic Engineering, Tsinghua University
    Beijing 100084
    China
    EMail: xing@cernet.edu.cn

    Gang Ren
    Tsinghua University
    Computer Science, Tsinghua University
    Beijing 100084
    China
    EMail: rg03@mails.tsinghua.edu.cn

    Ke Xu
    Tsinghua University
    Computer Science, Tsinghua University
    Beijing 100084
    China
    EMail: xuke@csnet1.cs.tsinghua.edu.cn

    Mark I. Williams
    Juniper Networks
    Suite 1508, W3 Tower, Oriental Plaza, 1 East Chang'An Ave
    Dong Cheng District, Beijing 100738
    China
    EMail: miw@juniper.net